SpyGlass® Power Verify
Rules Reference Guide

Version N-2017.12-SP2, June 2018

SYNOPSYS®

Synopsys, Inc.
690 E. Middlefield Road
Mountain View, CA 94043
www.synopsys.com

Report an Error

The SpyGlass Technical Publications team welcomes your feedback and suggestions on this publication. Please provide specific feedback and, if possible, attach a snapshot. Send your feedback to *spyglass_support@synopsys.com*.

Contents

Preface

About This Book

TThe SpyGlass® Power Verify Rules Reference Guide describes the rules that handle the structural, architectural, and system level issues of a design to reduce power.

Contents of This Book

The SpyGlass Power Verify Rules Reference Guide consists of the following chapters:

Chapter	Describes...
Using the Rules in the SpyGlass Power Verify Solution	Contains information on the rule parameters and constraints used with this product. In addition, contains information on the reports generated.
Rules in SpyGlass Power Verify	Contains information on the product rules and messages generated by each rule. In addition, debugging information and examples are provided.
Appendix: CPF Commands	Contains information on the supported CPF commands.
Appendix: UPF Commands	Contains information on the supported CPF commands.
Example of Using CPF / UPF Commands	Contains an example of using CPF / UPF commands.
Appendix: List of Rules With Applicable Design Stages	Contains list of rules along with their respective applicable design stages.
Appendix: SGDC Constraints	Provides links to SGDC constraints referenced by this product.

Typographical Conventions

This document uses the following typographical conventions:

To indicate	Convention Used
Program code	`OUT <= IN;`
Object names	`OUT`
Variables representing objects names	*<sig-name>*
Message	Active low signal name '<sig-name>' must end with _X.
Message location	OUT <= IN;
Reworked example with message removed	OUT_X <= IN;
Important Information	**NOTE:** This rule...

The following table describes the syntax used in this document:

Syntax	Description
[ ] (Square brackets)	An optional entry
{ } (Curly braces)	An entry that can be specified once or multiple times
| (Vertical bar)	A list of choices out of which you can choose one
. . . (Horizontal ellipsis)	Other options that you can specify

Using the Rules in the SpyGlass Power Verify Solution

This section provides information on using the SpyGlass Power Verify Solution. The information is organized in the following sections:

- *Introduction to SpyGlass Power Verify*
- *Running the SpyGlass Power Verify Solution*
- *Evaluating Results in the SpyGlass Power Verify Solution*
- *Parameters in the SpyGlass Power Verify Solution*
- *SpyGlass Power Verify Reports*

Introduction to SpyGlass Power Verify

The SpyGlass® Power Verify solution is an option to the SpyGlass Predictive Analyzer® and contains a variety of low power usage related rules.

Objective of the SpyGlass Power Verify Solution

The objective of the SpyGlass Power Verify solution is to guide design towards lower power design closure. This applies to the following stages of design:

- RTL
- Netlist
- PG Netlist

Low Power Methodology

Low power design involves successfully placing a design methodology that may include:

- Multiple voltage domains
- Special techniques to deal with leakage power
  - Power shut-off (PSO)
  - Use of Multi-threshold standard cell (HVT/SVT/LVT)
  - Back-biasing concept
  - Multiple power domains
  - State retention power gating (SRPG)
  - Frequency scaling dynamic biasing
  - Dynamic source biasing
  - Fine grain voltage scaling
- A host of techniques that can be applied to the construction of power-aware design.

Impact of the SpyGlass Power Verify Solution

The SpyGlass Power Verify solution helps guide the development of a design power-intent (UPF/CPF) such that:

- It incorporates low power design techniques

- Ensures adherence to new low power design methodologies arising from advances in technology.

The SpyGlass Power Verify solution has a collection of rules for specific purposes, which includes rules associated with certain power standard or lowpower design requirement. The product can be extended thus allowing you to develop and manage customized groupings of rules more easily.

Low Power Design

The low-power designs are the electronics designs that have been designed to use less electric power.

The ability to address potential low power design issues early in the design cycle is critical to achieve high productivity in the design process. We get more optimized designs as we address these issues during the RTL code development and improve the efficiency for the rest of the tool flow being used in the design process. In addition, there is a lot to be gained from having a truly good quality RTL code for the current as well as future implementations of these systems.

The main goals of analyzing such issues using predictive analysis include creation of a system that addresses RTL design using policies that guide the design process efficiently towards design goals under given design constraints.

This is important in the context of low power in that every low power design is different and tends to have its own interesting set of issues to solve.

Running the SpyGlass Power Verify Solution

The SpyGlass Power Verify solution can be run in the following two ways:

- Using the Tcl Shell Interface

  Refer to the Tcl Shell Interface User Guide
- Using Atrenta Console Graphical User Interface (GUI)

  Refer to the *Atrenta Console Reference Guide*

This section is organized in the following sub sections:

Prerequisites for Running SpyGlass Power Verify

Ensure you have performed the following:

- Read the design successfully into SpyGlass. Refer to the *Setting up a Design* section of the *Atrenta Console User Guide* for more details. You should ensure the analyzed design contains a minimum of unintended black boxes.
- Include appropriate technology library (*.lib*) files in your analysis. Refer to the *Specifying Functionality Information through .lib File in the Atrenta Console User Guide*. You should have access to all *.lib* files for PVT corners and Vt options you need to include in your analysis.
  The following is an example of a *.lib* file:

```
library (example) {
     cell (AND2X1) {
          pg_pin(VDD) {pg_type : primary_power; }
          pg_pin(VSS) {pg_type : primary_ground; }
```

```
            pin(A) {direction : input ; }
            pin(B) {direction : input ; }
           pin(Y) {direction : output; function : “A&B” ; }
        }
    }
```

- Read the following sections:
  - *Using Technology Library*
  - *Supported Files in the SpyGlass Power Verify Solution*
- (Optional) If you are using power format files (UPF or CPF) to specify power management, you should have access to those files corresponding to the same RTL you plan to analyze.

Using Technology Library

Some rules of the SpyGlass Power Verify solution require you to specify certain information using the attributes defined in the technology library.

The following table lists supported library attributes:

Attribute	Description	Syntax
always_on	When specified for a pin, this attribute identifies the pin as driven by an always-on signal. When specified for a cell, this attribute identifies the cell as an always-on cell.	`always_on : always_on_pin | always_on_cell`
input_voltage_range	Specifies the allowed voltage range of the level shifter input pin under all possible operating conditions (defined across multiple libraries).	`input_voltage_range (lower_bound, upper_bound)`
is_level_shifter	Specifies whether a cell is a level shifter cell or not.	`is_level_shifter : <true | false>`
is_isolated	Pin has been isolated internally	`is_isolated : <true | false>`
is_isolation_cell	Specifies whether a cell is an isolation cell or not.	`is_isolation_cell : <true | false>`

Attribute	Description	Syntax
isolation_cell_enable_pin	Specifies the enable input pin on an isolation cell.	`isolation_cell_enable_pin : <true | false>`
level_shifter_type	Specifies the voltage conversion type that is supported. Valid values are: LH (Low to High), HL (High to Low), HL_LH (High to Low and Low to High)	`level_shifter_type : level_shifter_type_value`
output_voltage_range	Specifies the allowed voltage range of the level shifter output pin under all possible operating conditions (defined across multiple libraries).	`output_voltage_range (lower_bound, upper_bound)`
pg_function	Specifies the switched pin of a Power Switch cell. (written in pin scope).	`pg_function : "VDDG"`

Attribute	Description	Syntax

pg_pin	Specifies power and ground pins. The pg_pin group contains *pg_type* attribute, listed below.	cell (namestring){
		  pg_pin (pg_pin_name){
		  pg_type : value ;
		  }
		}

pg_type	Specifies the type of power and ground pin. pg_type can have following values: - **primary_power:** Used to specify primary power pin of cell - **primary_ground:** Used to specify primary ground pin of cell - **backup_power:** Used to specify backup power pin of cell - **backup_ground:** Used to specify backup ground pin of cell - **internal_power:** Used to specify internal power pin of cell - **internal_ground:** Used to specify internal ground pin of cell - **nwell:** Used to specify bias power pin of cell - **pwell:** Used to specify bias ground pin of cell	
related_ground_pin	Specifies Ground pin associated with the functional pin	related_ground_pin : ground_pin_name
related_power_pin	Specifies Power pin associated with the functional pin	related_power_pin : power_pin_name

Attribute	Description	Syntax
retention_cell	Identifies a retention cell.	`retention_cell : retention_cell_name`
retention_pin	Specifies the retention pins of a retention cell. The attribute defines the following information: pin class (save | restore | save_restore), to perform the respective action on the state of the retention cell, disable_value (0 | 1), defining the value of the retention pin when the cell works in normal mode.	`retention_pin (pin_class, disable_value)`
std_cell_main_rail	When this attribute is set to true, the power and ground pin is used to determine which side of the voltage boundary the power and ground pin is connected.	`std_cell_main_rail : true | false ;`
switch_cell_type	Specifies the Power Switch cells (written in cell scope).	`switch_cell_type : coarse_grain;`
switch_function	Specifies the condition when the switch is turned off by the input switch_pin.	`switch_function : "<function_string>"`
switch_pin	Specifies the input enable pin of a Power Switch cell (written in pin scope).	`switch_pin : true;`
voltage_map	Specifies cell-level pg_pin groups **NOTE:** For a level shifter, if the *input_voltage_range*/ *output_voltage_ range* attribute is not specified, the input/output range is inferred from the *voltage_map* attribute.	`voltage_map (voltage_nameid, voltage_valuefloat) ;`
voltage_value	Specifies a voltage value	

Supported Files in the SpyGlass Power Verify Solution

The SpyGlass Power Verify solution can work with the following types of source files:

RTL description files with or without library files

You can specify the RTL (Verilog and/or VHDL) description files in the normal manner.

Gate-level netlist files and their associated library files

You can precompile your gate libraries using the SpyGlass Library Compiler and specify them to SpyGlass along with the gate-level netlist files in the normal manner.

Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)

The design data may be represented in physical Verilog, DEF, or another layout format, which can be converted to DEF. The file formats LEF and PLIB contain sufficient supply pin information (required during post-route verification). Also, PLIB files can be easily converted into LEF and vice-versa. In some cases, it is possible to convert DEF files into physical Verilog and vice-versa.

The library data can be either in PLIB or LEF files. The PLIB or LEF files are specified along with liberty files to the SpyGlass library compiler. When you execute the SpyGlass library compiler, you need to specify the name of the library file in the project file using:

```
read_file -type gateslib
```

To enable processing of post-layout netlist files, in the project file specify:

```
set_option enable_pgnetlist yes
```

To ensure that SpyGlass uses the supply pin information, it is recommended to compile all the library information into an sglib file.

SpyGlass parses the specified PLIB files and reports issues, if any, through the PLIBSTX* and PLIBWRN* built-in rules. Also, each macro in the PLIB files must also be defined in the associated LIB files.

You can also specify the physical macro definitions using the *pg_cell* constraint in lieu of the physical library files (PLIB). Thus, you can specify all required physical macro definitions in PLIB files only, using the `pg_cell` constraint, or a mix of both. The macro definitions in the

PLIB files are read first and missing macros, if any, are searched in the `pg_cell` constraint.

DEF files and their associated LEF files

To setup designs containing LEF and DEF data, you need to compile LEF data into an sglib file, and then specify the DEF file name to SpyGlass by using the following project file command:

```
read_file –type def <file-name>
```

To specify a LEF file, use the following project file command:

```
read_file –type lef <file-name>
```

NOTE: *It is recommended that you specify the top-level of the design hierarchy by using the set_option top <top-design> project file command when specifying multiple DEF files.*

You can also specify the physical macro definitions using the *pg_cell* constraint in lieu of the LEF files. Thus, you can specify all required physical macro definitions in LEF files only, using the `pg_cell` constraint only, or a mix of both. The macro definitions in the LEF files are read first and missing macros, if any are searched in the `pg_cell` constraint.

Using Constraints in the SpyGlass Power Verify Solution

Constraints are specified to:

- Provide additional design information that is not apparent in RTL
- Restrict SpyGlass analysis to a set of design objects

Some rules of the SpyGlass Power Verify solution also require you to specify constraints. You can specify these constraints in any of the following ways:

- In the SpyGlass Design Constraints file.
- In the Common Power Format (CPF). If you have power intent specified in the CPF format file, you can provide the CPF file using the `power_data` constraint in the SGDC file, as shown below:

  ```
  power_data -format CPF -file <file-name-list>
  ```

  For details of CPF commands support by the SpyGlass Power Verify

solution, refer to the *Supported CPF Commands* section.

- In the Unified Power Format (UPF). If you have power intent specified in the UPF format file, you can provide the UPF file using the `power_data` constraint in the SGDC file, as shown below:

  ```
  power_data -format UPF -file <file-name-list>
  ```

  For details of UPF commands support by the SpyGlass Power Verify solution, refer to the *Supported UPF Commands* section.

SpyGlass Design Constraints (SGDC) provides additional design information that is not apparent in an RTL. Refer to the *Appendix: SGDC Constraints* section for details.

NOTE: *For a SpyGlass run, you should use only one of the methods given above to provide power intent. However, if both CPF and UPF formats, are specified, the LP_POWERDATA_CHECK rule reports a fatal violation.*

NOTE: *For the CPF or UPF flow, SpyGlass also honors the following SGDC commands:*

- *activity*
- *always_on_buffer*
- *assume_path*
- *assertion_signal*
- *cell_hookup*
- *cell_pin_info*
- *cell_tie_class*
- *clock*
- *multivt_lib*
- *non_pd_inputcells*
- *power_data*
- *power_down*
- *power_down_sequence*
- *pg_cell*
- *pg_pins_naming*
- *ram_instance*
- *ram_switch*
- *set_case_analysis*

- *special_cell*
- *switchoff_wrapper_instance*

Using Commands in the SpyGlass Power Verify Solution

Supported CPF Commands

The SpyGlass Power Verify solution supports the following CPF commands:

Voltage/Power Domain Related Commands

- *create_power_domain*
- *update_power_domain*

Supply Related Commands

- *create_power_nets*
- *create_ground_nets*
- *create_global_connection*

Power Mode Related Commands

- *create_power_mode*
- *create_nominal_condition*

Isolation Related Commands

- *define_isolation_cell*
- *create_isolation_rule*
- *update_level_shifter_rules*

Level Shifter Related Commands

- *define_level_shifter_cell*
- *create_level_shifter_rule*
- *update_level_shifter_rules*

Power Switch Related Commands

- *define_power_switch_cell*
- *create_power_switch_rule*

- *update_power_switch_rule*

Retention Cell Related Commands

- *define_state_retention_cell*
- *create_state_retention_rule*

Always On Cell Related Commands

- *define_always_on_cell*

Other Commands

- *set_design*
- *end_design*
- *set_macro_model*
- *end_macro_model*
- *set_power_mode_control_group*
- *end_power_mode_control_group*
- *get_parameter*
- *include*
- *define_related_power_pins*

Commands Not Affecting the Rules in the SpyGlass Power Verify Solution

```
create_analysis_view
create_bias_net
create_mode_transition
create_operating_corner
define_library_set
identify_always_on_driver
identify_power_logic
set_power_target
set_switching_activity
set_floating_ports
set_input_voltage_tolerance
set_wire_feedthrough_ports
```

For details, refer to the *Supported CPF Commands* section.

Supported UPF Commands

The SpyGlass Power Verify solution supports the following UPF commands:

Voltage Domain Related Commands

- *create_power_domain*
- *add_domain_elements*
- *set_domain_supply_net*

Supply Related Commands

- *create_supply_port*
- *create_supply_net*
- *connect_supply_net*
- *set_pin_related_supply*
- *set_pin_related_supply*

Isolation Related Commands

- *set_isolation*
- *map_isolation_cell*
- *set_isolation_control*

Commands to Specify Level Shifters

- *map_level_shifter_cell*
- *set_level_shifter*

Retention Cell Related Commands

- *set_retention*
- *set_retention_control*
- *set_retention_control*

Commands to Specify Power States

- *create_pst*
- *add_pst_state*

Commands to Specify Power Switches

- *create_power_switch*

- *map_power_switch*

Commands Not Affecting the Rules in the SpyGlass Power Verify Solution

```
set_power_switch
merge_power_domains
get_supply_net
bind_checker
create_hdl2upf_vct
create_upf2hdl_vct
name_format
```

For details, refer to the *Supported UPF Commands* section.

Using Parameters in the SpyGlass Power Verify Solution

Parameters are required by the SpyGlass Power Verify solution if you need to customize the behavior of the rules.

For example, if a rule does not assume an all off state for an IP when the SoC supply is off, you can specify the relevant parameter to make the rule assume an all off state for an IP when the SoC supply is off.

Specifying Parameters

You can set parameters in both Atrenta Console and Tcl by using the following syntax:

```
set_parameter <parameter_name> <parameter_value>
```

For details of the parameters that can be used by the rules in the SpyGlass Power Verify solution, refer to the *Parameters in the SpyGlass Power Verify Solution* section.

Using SpyGlass Power Verify Goals

A goal is a pre-packaged set of rules that detects specific types of design issues. For example, the *power_verif_rtl* goal contains checks to be performed on the RTL. When you run a goal after specifying design files in Atrenta Console all rules of that goal are run. Once the goal run is complete, appropriate violation messages are reported to indicate design

issues. For more details about Goals, refer to the *Introducing Goals* section in the *Atrenta Console User Guide*.

The following SpyGlass Power Verify goals are available:

Goals	Purpose
power_best_practice	Checks the RTL and finds inefficient structures from the power standpoint. This includes arithmetic structures having glitches, tristate buses, and different types of FSM structures.
power_verif_postroute	For performing post-route checks
power_verif_postsynth	For performing post-synthesis checks
power_verif_rtl	For performing RTL checks
power_verif_noninstr	For performing non instrumented RTL checks
power_verif_abstract	For generating abstract power model of a block in.lib or UPF format

Using the Project Files

You can also run your goals through the saved project files (*.prj*). You can navigate through your directory structure to locate the required goal file.

The following is an example of a *.prj* file:

```
##Read HDL and library (design) Data
read_file -type sglib lsi_10k.sglib
read_file -type hdl test1.v
read_file -type hdl test_sup.v
read_file -type sgdc format.sgdc

##Common Options Section
set_option enable_save_restore no
set_option projectwdir $::env(SPYGLASS_OUTDIR)
set_option language_mode verilog
set_option active_methodology $SPYGLASS_HOME/GuideWare2.0
block/rtl_handoff

##Goal Setup Section
```

```
define_goal test_goal -policy { lowpower } {
        ###### Goal Options ######
        set_goal_option rules LP_DECOMPILE_CONSTR
        set_goal_option rules LPSVM08
        ###### Parameters #######
        set_parameter lp_skip_buf 0
}
set test_goal_to_run test_goal
```

For more details on using the project files, refer to the *SpyGlass Console Reference Guide*.

Using Rule Mnemonics

Rule mnemonics refer to the custom rule names assigned to the SpyGlass rule. These custom rule names are more descriptive and convey additional details about a rule.
To know more about using the rule mnemonics, refer to the *Using Rule Mnemonics* section of the *SpyGlass Console Reference Guide.*

The following table provides a list of rules along with their respective mnemonics provided by the SpyGlass Power Verify solution and applicable messages.

Mnemonics	Rule Name	Messages
LpObjectSelfExistsnot	*checkUPF_existence*	All messages
LpIsoStrategyMissing	*LPISO04A*	All messages
LpIsoStrategyIncorrectOption	*LPISO04B*	All messages
LpIsoStrategyIncorrectSupply	*LPISO04C*	All messages
LpIsoStrategyIncorrectMap	*LPISO04D*	All messages
LpIsoStrategyRedundantAlLpath	*LPISO05A*	All messages
LpIsoStrategyRedundantOnepath	*LPISO05B*	All messages
LpLsPwrpinMissing	*LPPLIB04*	LPPLIB04_1, LPPLIB04_11

Mnemonics	Rule Name	Messages
LpLsPwrpinConnflaw	*LPPLIB04*	LPPLIB04_2, LPPLIB04_3, LPPLIB04_4, LPPLIB04_6, LPPLIB04_7, LPPLIB04_8, LPPLIB04_9
LpLsPinConnflaw	*LPPLIB04*	LPPLIB04_5, LPPLIB04_10
LpSupplyPwrpinConnflaw	*LPPLIB06*	All messages
LpLsGndpinMissing	*LPPLIB07*	LPPLIB07_5,
LpLsGndpinConnflaw	*LPPLIB07*	LPPLIB07_1, LPPLIB07_2, LPPLIB07_3, LPPLIB07_4, LPPLIB07_6, LPPLIB07_7, LPPLIB07_8, LPPLIB07_9, LPPLIB07_10
LpSupplyGndpinConnflaw	*LPPLIB15*	All messages
LpBiasPwrpinConnflaw	*LPPLIB18A*	All messages
LpBiasGndpinConnflaw	*LPPLIB18B*	All messages
LpBiasNetRelativeoff	*LPPLIB19A*	All messages
LpIsoCellMissingAtpdout	*LPSVM08A*	LPSVM08A_1, LPSVM08A_2, LPSVM08A_4
LpIsoCellIncorrectAtpdout	*LPSVM08A*	LPSVM08A_3, LPSVM08A_4, LPSVM08A_5, LPSVM08A_6, LPSVM08A_7, LPSVM08A_8, LPSVM08A_9
LpIsoCellClampflawAtpdout	*LPSVM09*	All messages
LpIsoCellClampflawAtpdin	*LPSVM47*	All messages
LpIsoCellMissingAtpdin	*LPSVM60*	LPSVM60_1, LPSVM60_2
LpIsoCellIncorrectAtpdin	*LPSVM60*	LPSVM60_3, LPSVM60_4, LPSVM60_5, LPSVM60_6, LPSVM60_7, VLPSVM60_8, LPSVM60_9
LpDomainTopSpecflaw	*UPF_lowpower 03*	All messages

Mnemonics	Rule Name	Messages
LpPstStateDuplicate	*UPF_lowpower04*	All messages
LpIsoControlSpecflaw	*UPF_lowpower05*	UPF_lowpower05_1, UPF_lowpower05_2, UPF_lowpower05_6
LpIsoStrategyMapflaw	*UPF_lowpower05*	UPF_lowpower05_3, UPF_lowpower05_4
LpIsoStrategyClampflaw	*UPF_lowpower05*	UPF_lowpower05_5
LpPswStrategyMapflaw	*UPF_lowpower07*	UPF_lowpower07_1, UPF_lowpower07_2
LpPswStrategySpecflaw	*UPF_lowpower07*	UPF_lowpower07_3, UPF_lowpower07_4
LpSupplyPorttopActiveoff	*UPF_lowpower09*	UPF_lowpower09_1
LpPswPortActiveoff	*UPF_lowpower09*	UPF_lowpower09_2
LpSupplyNetDrivenwrong	*UPF_lowpower09*	UPF_lowpower09_3, UPF_lowpower09_4, UPF_lowpower09_5, UPF_lowpower09_7
LpSupplySelfUnused	*UPF_lowpower09*	UPF_lowpower09_6
LpSignalPinTieflawPersupplyrelated	*LPPLIB14*	LPPLIB14_1, LPPLIB14_8
LpSignalPinTieflawPertiespec	*LPPLIB14*	LPPLIB14_2, LPPLIB14_3, LPPLIB14_4, LPPLIB14_5, LPPLIB14_6, LPPLIB14_7
LpPswPinConnflaw	*LPPLIB17*	LPPLIB17_1, LPPLIB17_2, LPPLIB17_3, LPPLIB17_4, LPPLIB17_5, LPPLIB17_6, LPPLIB17_8, LPPLIB17_9, LPPLIB17_10, LPPLIB17_13
LpPswSupplyMismatchInout	*LPPLIB17*	LPPLIB17_7

Mnemonics	Rule Name	Messages
LpPswCellIncorrectLoc	*LPPLIB17*	LPPLIB17_11, LPPLIB17_12
LpPswCellMissing	*LPPLIB17*	LPPLIB17_14
LpPstStateMismatch	*UPF_lowpower08*	UPF_lowpower08_1
LpPstStateUndefined	*UPF_lowpower08*	UPF_lowpower08_2
LpPstStateMergeflaw	*UPF_lowpower08*	UPF_lowpower08_3, UPF_lowpower08_6
LpPstCmdSpecflaw	*UPF_lowpower08*	UPF_lowpower08_8, UPF_lowpower08_9, UPF_lowpower08_10, UPF_lowpower08_11, UPF_lowpower08_12
LpSupplyPinMulticonn	*UPF_lowpower08*	UPF_lowpower08_13, UPF_lowpower08_17
LpPstStateIgnored	*UPF_lowpower08*	UPF_lowpower08_14, UPF_lowpower08_16, UPF_lowpower08_19
LpSupplySetConnflaw	*UPF_lowpower08*	UPF_lowpower08_18, UPF_lowpower08_20, UPF_lowpower08_21
LpIsoControlSenseflaw	*UPF_lowpower10*	All messages
LpAnyStrategyElementSpecflaw	*UPF_lowpower12*	All messages
LpPswPortstateMismatchAtports	*UPF_lowpower13*	All messages
LpPstSupplyUnused	*UPF_lowpower14*	UPF_lowpower14_1
LpPstPortstateUnused	*UPF_lowpower14*	UPF_lowpower14_2

Mnemonics	Rule Name	Messages
LpSupplyMultipininstMissingConn	*UPF_lowpower 15*	All messages
LpBiasNetRelativeoffPerspec	*UPF_lowpower 16*	All messages

Evaluating Results in the SpyGlass Power Verify Solution

The SpyGlass Power Verify solution generates various reports, schematics and other data that you can use to evaluate your design.

This section is organized in the following sub sections:

- *Viewing Reports*
- *Viewing Schematics*
- *Using the Power Intent View Window*

Viewing Reports

The data related to the issues found in the design by SpyGlass Power Verify is written to the respective reports files. You can cross-check this data in order to rectify issues in your design.

Reports can be viewed by selecting the relevant option from the *Reports* > *Lowpower* menu.

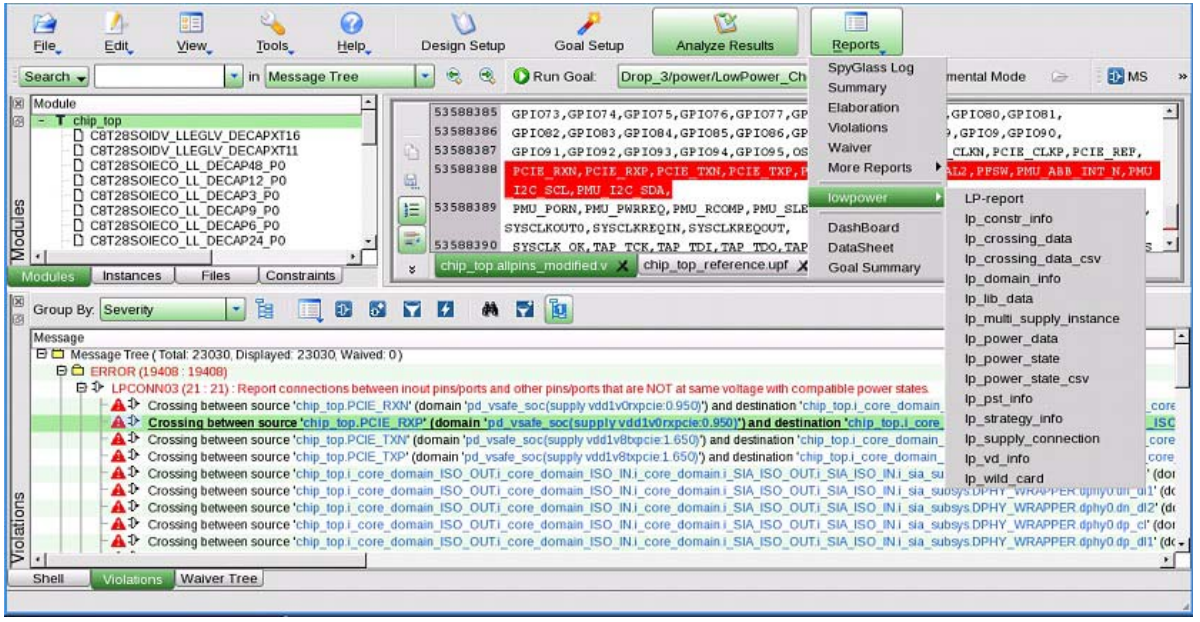


FIGURE 1. Viewing reports

For more details on viewing the reports, refer to the *Atrenta Console Reference Guide.*

For the details of each report that is generated by the SpyGlass Power Verify solution, refer to the *SpyGlass Power Verify Reports* section.

Viewing Schematics

The SpyGlass Power Verify solution provides a mechanism through which you can view the graphical representation of the reported violations.

There are two types if of schematics that can be viewed:

- Modular schematic

  The Modular Schematic window displays a hierarchical schematic layout of the modules analyzed by SpyGlass. This window displays data based on the violation message selected or a module selected from the Module

View Page.

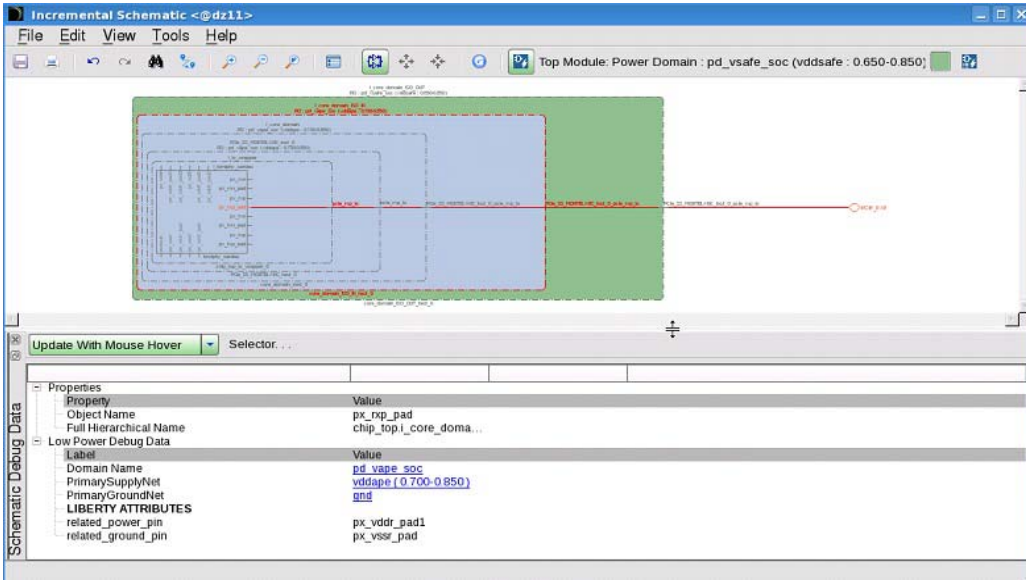


FIGURE 2. Viewing schematics

For more details on viewing the modular schematics, refer to the The Modular Schematic Window in the *Atrenta Console Reference Guide.*

- Incremental schematic

The incremental schematic window displays the selected portions of (flattened) design schematic across hierarchical boundaries.

For more details on viewing the incremental schematics, refer to the *The Incremental Schematic Window* section in the *Atrenta Console Reference Guide.*

Using the Power Intent View Window

The SpyGlass Power Verify solution provides the *Power Intent View* window to view the power intent of the design. The purpose of this window is to

show the power domain information of each hierarchy in design in graphical format.

Use the Power Intent View window to:

- Check power domains assigned to instances in a design.
- View power intent of the design.

To open the window, select *Tools -> Power Intent View* menu option.

The following figure shows the *Power Intent View* window:

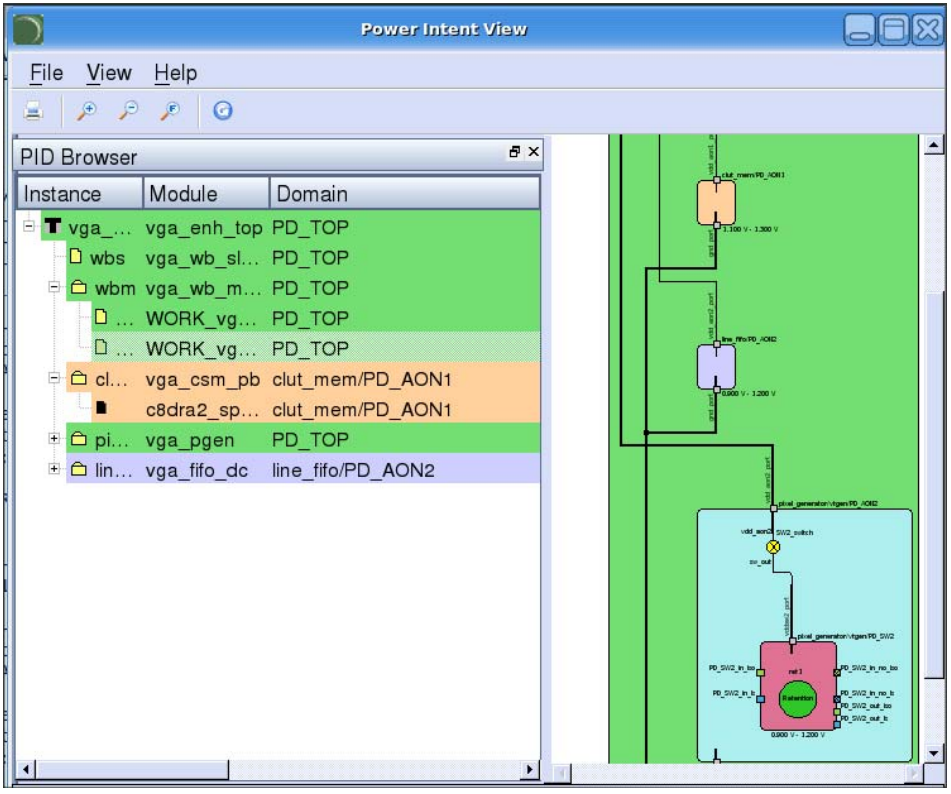


FIGURE 3. Power Intent View window

You can also click the  icon, on the top of the Violation Browser to open the Power Intent View window.

The Power Intent View window is divided into the following functionality:

- *PID Browser Section*
- *Power Intent Hierarchy Section*
- *The Supply Net Relationship Table Widget*

PID Browser Section

In this section, you can view power domains assigned to instances in the design hierarchy, as shown in the following figure:

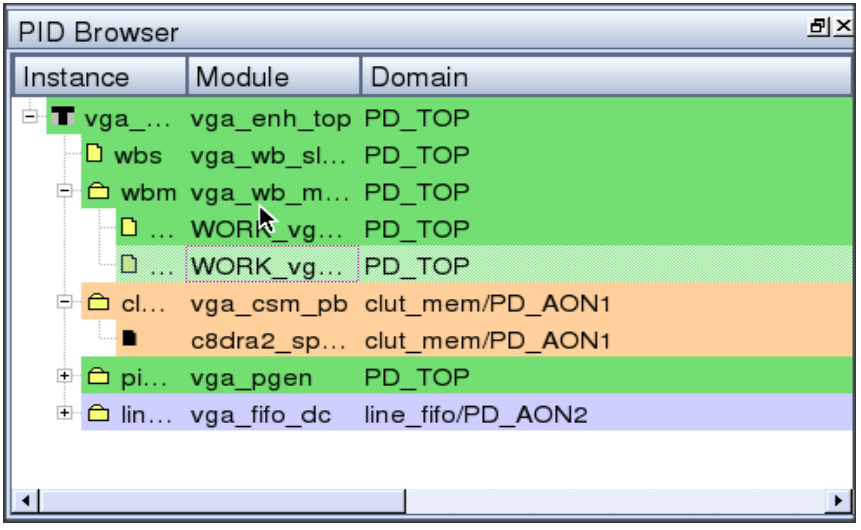


FIGURE 4. PID browser section

Each instance is highlighted in the color of the assigned power domain. This color is same as the color of power domains appearing in the *Power Intent Hierarchy Section*.

To hide this section, deselect the *View -> Toolbars -> PID Browser* option of the *Power Intent View* window.

When you double-click an instance in the PID Browser section, the design file appears in the Files section, as shown in the following figure:

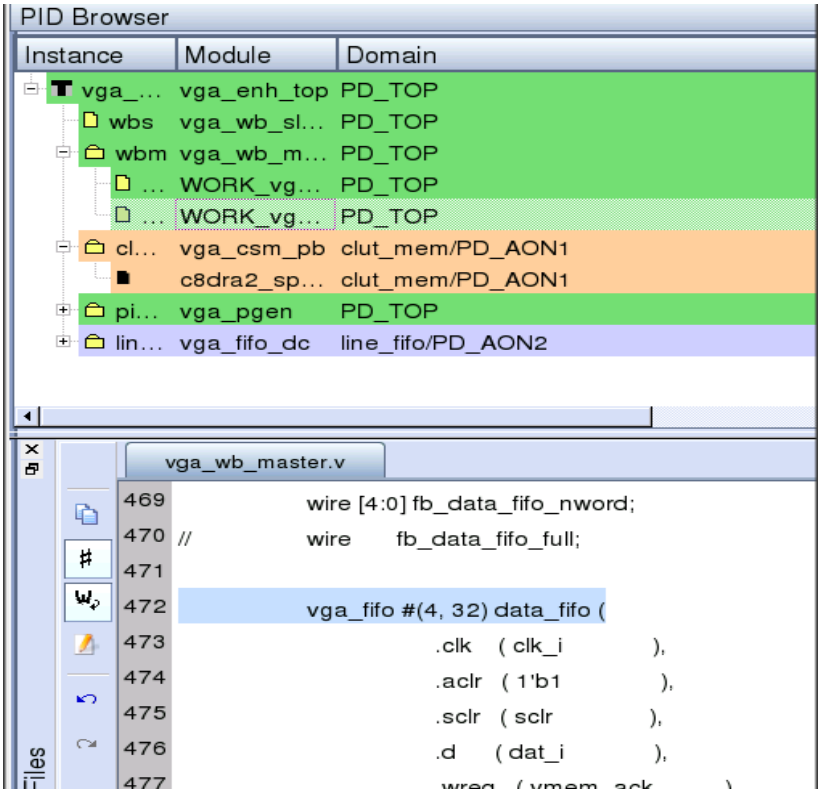


FIGURE 5. PID browser

Power Intent Hierarchy Section

This section shows the power intent of your design in a graphical format. The power intent view works only with a UPF file.

The following figure shows the Power Intent Hierarchy section:

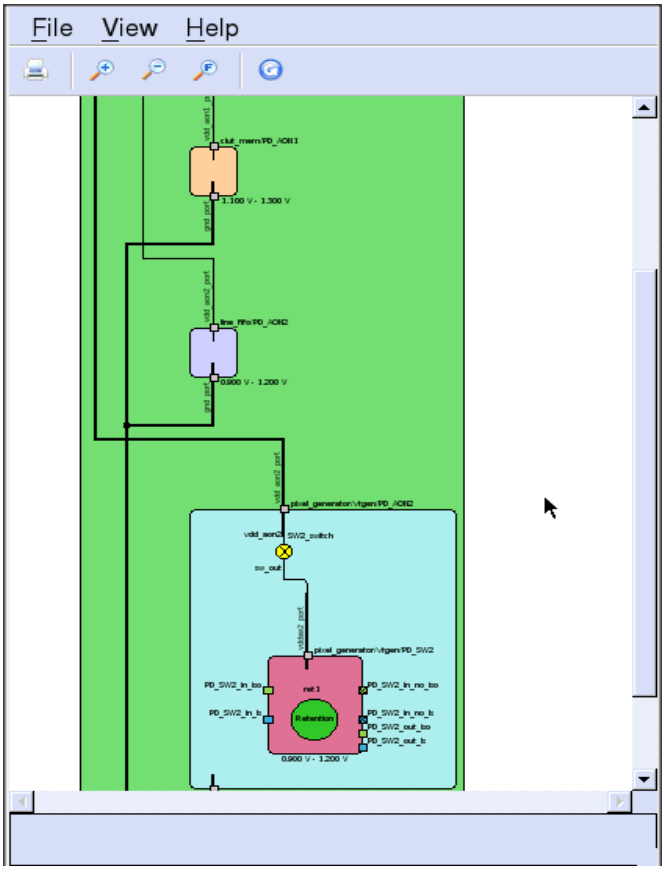


FIGURE 6. Power intent hierarchy section

In the above section:

- Each power domain appears in a different color.
- Isolation, level shifter, retention, power switch, and supply rails appear for each power domain.
- All power domains are organized as per the specified hierarchy.

Information Displayed for a Power Domain Object

SpyGlass Power Verify displays the following information for a power domain:

- *Supply Ports*
- *Supply Nets*
- *Isolation and Level Shifter Logic*
- *Power Switches*
- *Retention*

NOTE: *The input boxes appear on the left side and the output boxes appear on the right side of a power domain.*

Supply Ports

Ports are displayed on the top and bottom of a power domain with connected power and ground nets.

Based on different situations, following details are displayed for a power domain.

A port appears on the bottom of a domain if there is a ground rail connected. Otherwise, the port is displayed on the top. A primary rail appears thick and bold.

The following figure displays a power domain:

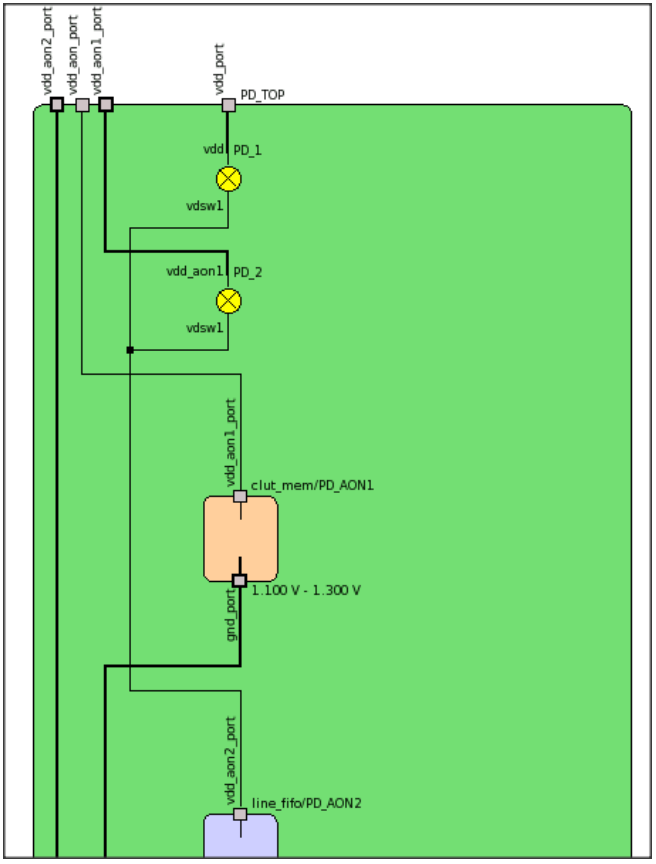


FIGURE 7. Supply ports

Supply Nets

All the rails that are given in the power intent are shown here, whether they are connected to power ports, ground ports, or power switch ports.

The following figure shows a portion of a power domain with the rail information:

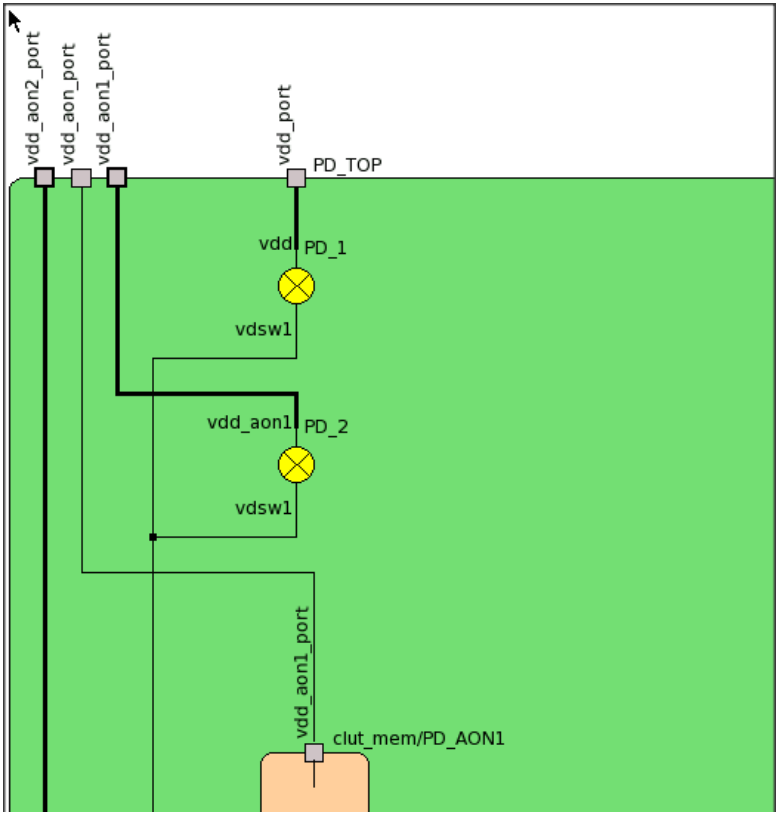


FIGURE 8. Supply nets

Isolation and Level Shifter Logic

The following table describes various situations in which different boxes for isolation and level shifter logic appear on a power domain:

TABLE 1. **Input and Output Boxes for Isolation and Level Shifter Logic**

Box Name	Description
	**Input:** This box appears on the input side (left) of a power domain if an isolation logic applies to input or both (that is input and output). In this case, the *Applies_To* field for that isolation logic is set to *input*. **Output:** This box appears on the output side (right) of a power domain if an isolation logic applies to output or both (that is input and output). In this case, the *Applies_To* field for that isolation logic is blank.
	This box appears on the input side (left) of a power domain if a level shifter is present on input or both (that is input and output). In this case, the *Applies_To* field for that logic is set to *input*.
	This box appears on the output side (right) of a power domain if a level shifter is present on output or both (that is input and output). In this case, the *Applies_To* field for that logic is blank.
	This circle indicates the presence of a retention logic on a power domain. It appears in the center of the power domain.
	**Input:** This box appears on the input side (left) of a power domain if an isolation logic with -no_isolation specified applies to the input or both (that is input and output). In this case, the *Applies_To* field for that isolation logic is set to *input*. **Output:** This box appears on the output side (right) of a power domain if an isolation logic with -no_isolation specified applies to output or both (that is input and output). In this case, the *Applies_To* field for that isolation logic is blank.
	**Input:** This box appears on the input side (left) of a power domain if an isolation logic with -no_shift specified applies to the input or both (that is input and output). In this case, the *Applies_To* field for that isolation logic is set to *input*. **Output:** This box appears on the output (right) side of a power domain if an isolation logic with -no_shift specified applies to output or both (that is input and output). In this case, the *Applies_To* field for that isolation logic is blank.

The following figure displays a power domain with the above information:

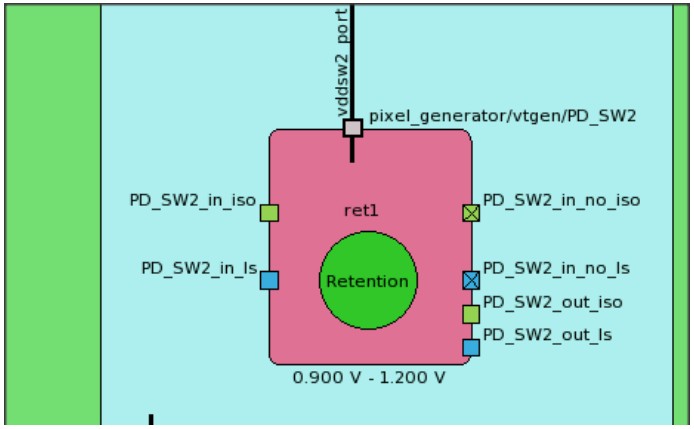


FIGURE 9. Input and Output Boxes for Isolation and Level Shifter Logic

Power Switches

The following figure shows power switch of a power domain:

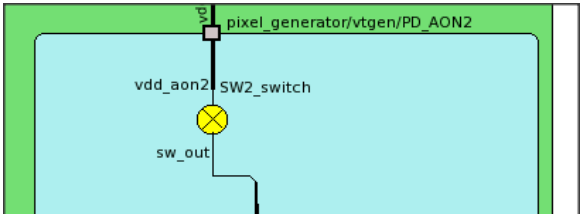


FIGURE 10. Power switches

Retention

The following figure shows the retention specified on a power domain:

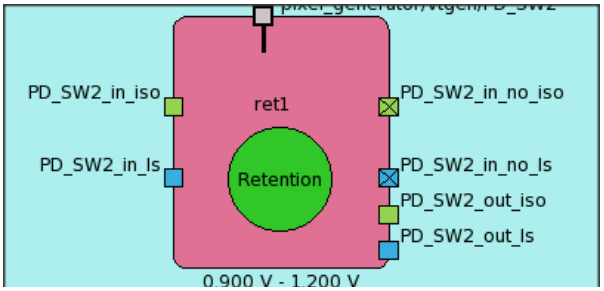


FIGURE 11. Retention

Isolation and Level Shifters

Isolation and level shifter strategies on a power domain are respectively shown with green and blue boxes.

Elements Specified with -no_isolation or -no_shift

If a power domain contain elements that are not covered by any isolation or level shifter logic having -no_isolation or -no_shift. Such elements are identified by a box with a cross.

The following figure shows power domains containing such elements:

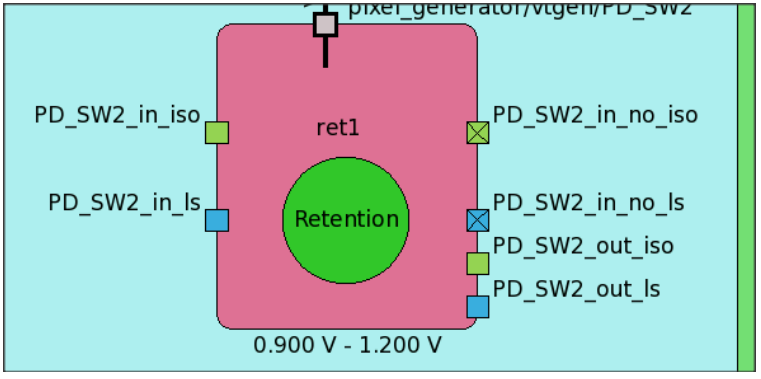


FIGURE 12. Power domains not covered by any isolation or level shifter logic

Viewing the UPF File

To view the UPF file of a power intent, such as isolation logic or a power switch, click on that object in the schematic.

The corresponding UPF file appears in the *Files* section, as shown in the following figure:

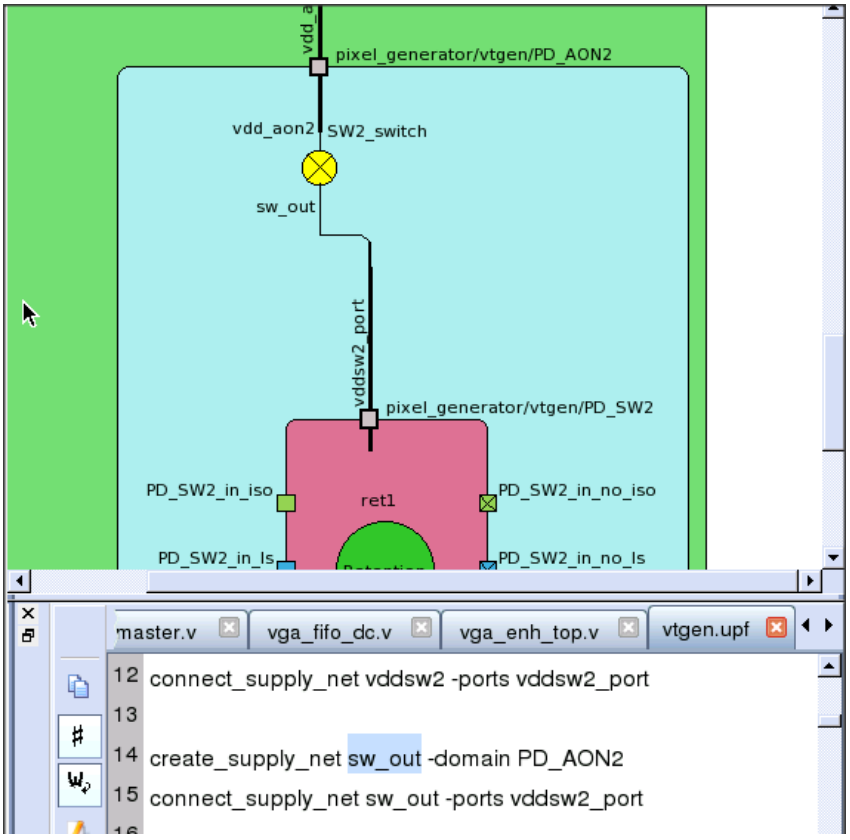


FIGURE 13. Viewing the UPF file

NOTE: *To show or hide the Files window, select or deselect the View -> Toolbars -> Files option of the Power Intent View window.*

Viewing Object Properties

You can view the object properties, such as object name and object type in the Properties window, as shown in the following figure:

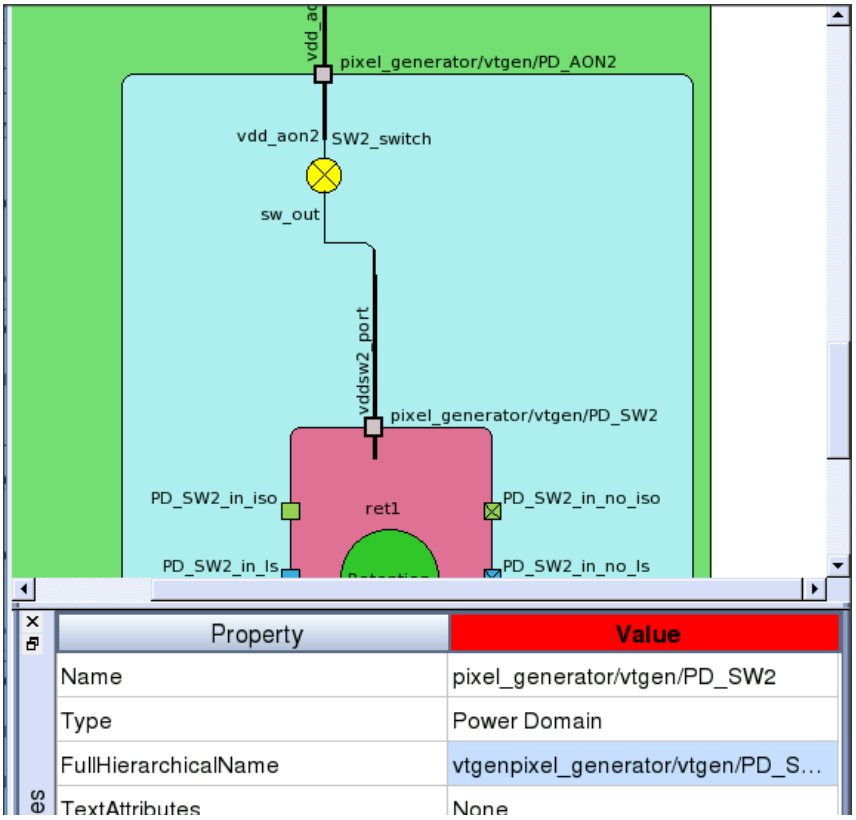


FIGURE 14. Viewing object properties

NOTE: *To show or hide the Properties window, select or deselect the View -> Toolbars -> Properties option of the Power Intent View window.*

Viewing Mini Map

You can select a portion in the schematic to view the details of that portion.

This way you can focus on the relevant area in the schematic while debugging.

To view a particular portion in the schematic, move the cursor over the required area in the Minimap window, as shown in the following figure:

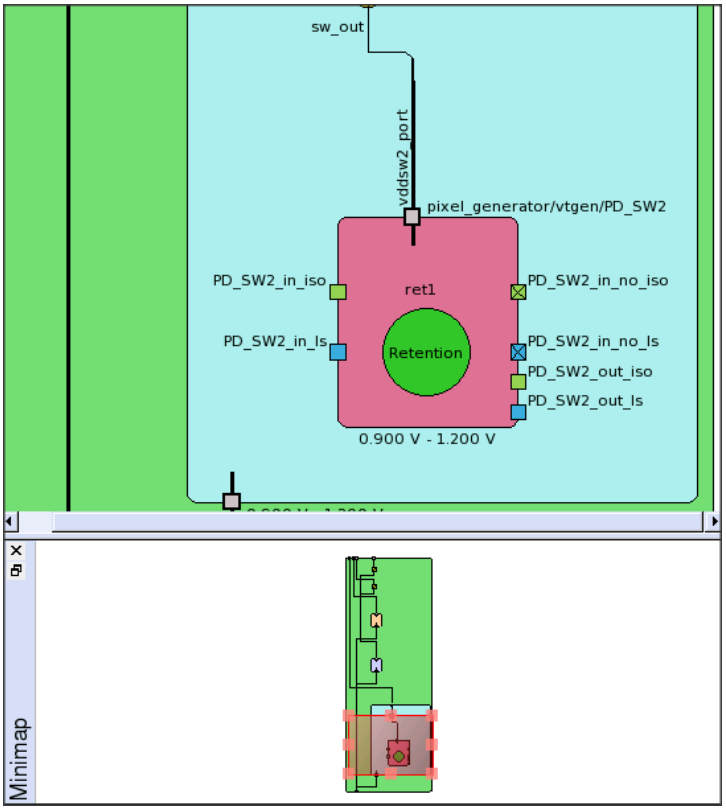


FIGURE 15. Viewing mini map

To show or hide the Minimap window, select or deselect the View -> Toolbars -> Minimap option of the Power Intent View window.

Search for Instances

You can search for instances using the **Find** button as shown in the following figure:

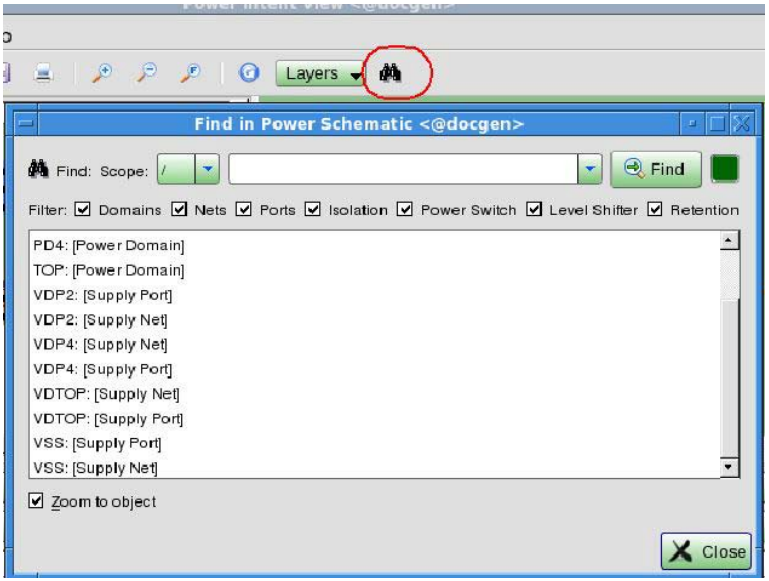


FIGURE 16. Searching for instances

Domain or strategy selected in PIH Schematic is back-referenced to corresponding command in the UPF file as shown in the following figure:

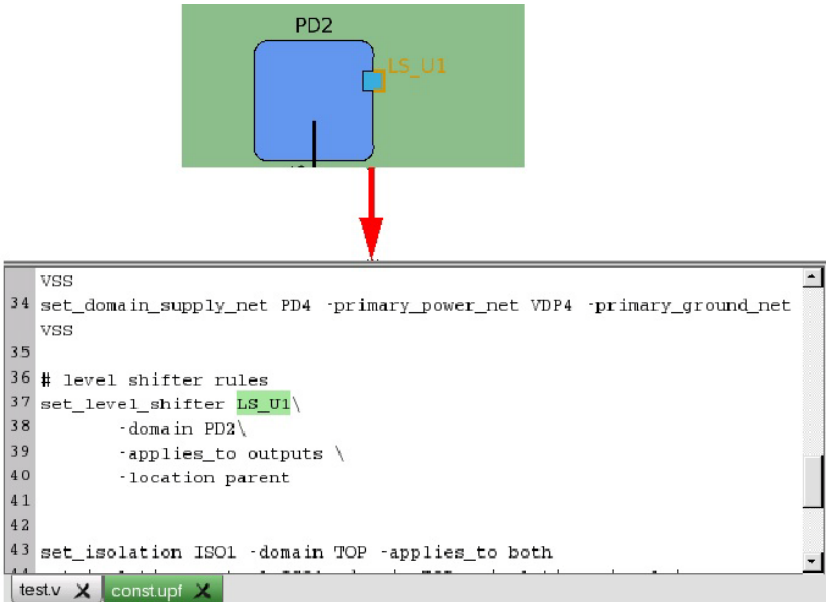


FIGURE 17. PIH Schematic back-referenced

The Supply Net Relationship Table Widget

The Supply Net Relationship table displays the relationship between the two supplies. By default, when a user double clicks on a violation message in message browser, the supply net relationship window is automatically updated with information related to the two supply nets which are being referred to in the violation message. In case user wants to view relationship between any supply nets of his choice, do the following:

- Open the Power Intent view
- Select the **Supply Net Relationship** tab
- Select Scope 1 and Net 1 from the respective drop-down menu
- Select Scope 2 and Net 2 from the respective drop-down menu

Then the relationship table displays the information in two columns, Property and Value.

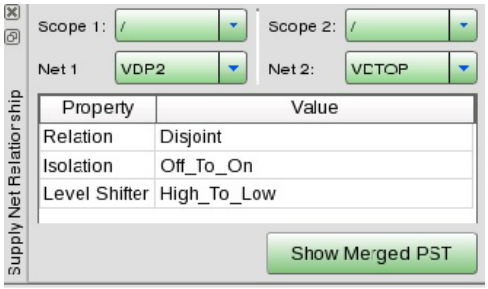


FIGURE 18. Supply Net Relationship widget

For each property there can be any one of the four relationship values:

Property	Value	Description
Hierarchical	Common	Both nets are connected
	Disjoint	Nets are not connected to each other
	Parent-Child	The second net is connected to switched output of first net
	Child-Parent	The first net is connected to switched output of second net
Isolation	None	No requirement
	On-to-Off	The first net is on and the second net is off in the PST
	Off-to-On	The first net is off and the second net is on in the PST
	Both	A net is on while the other is off in the PST
Level Shifter	None	No requirement
	High-to-Low	First net is at a higher voltage than the second net in the PST
	Low-to-High	First net is at a lower voltage than the second net in the PST
	Both	A net is at a relatively higher voltage than the other in the PST

The Power State Table (PST) Widget

The Power/Port State Table (PST) widget provides the supply value and names of the Power States for a circuit. You can determine the number of power states in the circuit and examine the correctness of each state.

Viewing the PST widget:

- Click on the ***Show Merged PST*** button, provided at the bottom of the *Supply Net Relationship* window. This opens up the merged PST for the two nets visible in the relationship window.
- Click on the ***Show PST*** button, provided at the bottom of the merged PST view. This opens the Power/Port State Table widget displaying power state table for all supply nets available in current scope in the design.

The **Power State Table** displays the Power States horizontally and the Supplies vertically. Each cell in this table displays the value of the supply state.
States requiring level shifting are highlighted in blue color, while the ones requiring isolation are shown in green color. The two supply nets which are being compared are also highlighted separately in the PST widget.

The following figure displays a consolidated view of the Supply Net Relationship and PST widgets:

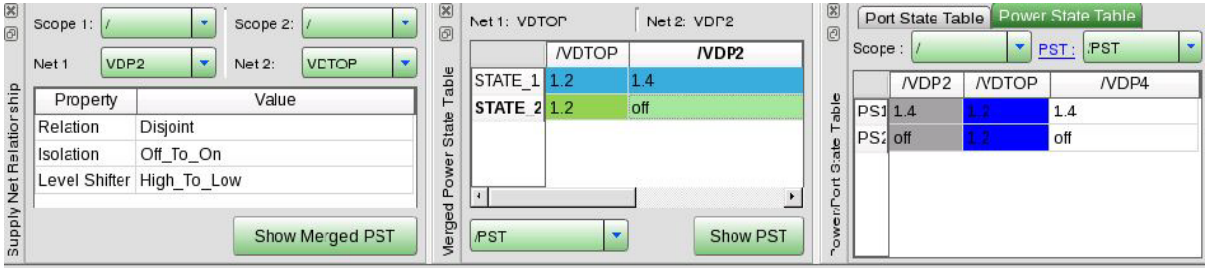


FIGURE 19. PST widget

Parameters in the SpyGlass Power Verify Solution

This section provides detailed information on the SpyGlass Power Verify product rule parameters.

You can set these parameters in both Atrenta Console and Tcl by using the following syntax:

```
set_parameter <parameter_name> <parameter_value>
```

For more information on setting the parameters, refer to the *SpyGlass Tcl Interface User Guide* and *Atrenta Console User Guide*.

lp_abstraction_config_file

Specifies the configuration file to be used for generating the abstraction model by the *PV_Abstract01* rule.

By default, this parameter is not set. Set this parameter to path of the configuration file to be used for generating the abstraction model.

Used by	*PV_Abstract01*
Options	lib, upf
Default	`null`
Example	
Console/Tcl-based usage	`set_parameter lp_abstraction_config_file ../SypGlaSS_Config`
Usage in goal/source files	`-lp_abstraction_config_file=../SypGlaSS_Config`

lp_allow_check_name_format

Specifies whether to report violations for prefix/suffix convention mismatches of isolation/level shifter instance names to corresponding strategies.

By default, this parameter is set to `no` and the *LPSVM08A*, *LPSVM60*, *LPSVM04B* rules do not report a violation if isolation/level shifter instance

name does not match with the name prefix/suffix specified in corresponding strategy.

Set this parameter to `yes` to report violations for such cases.

Used by	*LPSVM08A*, *LPSVM60*, *LPSVM04B*
Options	0, 1, no, or yes
Default	`0`
Example	
Console/Tcl-based usage	`set_parameter lp_allow_check_name_format yes`
Usage in goal/ source files	`-lp_allow_check_name_format=yes`

lp_allow_force_shift_in_UPF1

Specifies whether to enable the `-force_shift` argument of the *set_level_shifter* command in UPF 1.0.

By default, the `lp_allow_force_shift_in_UPF1` parameter is set to `no` and the `-force_shift` argument is enabled in UPF 2.0 only.

Set this parameter to `yes` to enable `-force_shift` argument of the `set_level_shifter` command in UPF 1.0 as well.

Used by	All level shifter rules in the SpyGlass Power Verify solution
Options are	0, 1, no, or yes
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_allow_force_shift_in_UPF1 yes`
Usage in goal/ source files	`-lp_allow_force_shift_in_UPF1=yes`

lp_allow_iso_sig_from_equivalent_domain

Specifies whether to avoid violations for equivalent domains.

By default, this parameter is set as `no`, and the *LPSVM12A* rule reports violations for equivalent domains.

Set this parameter to `yes` to avoid violations for such cases.

Used by	*LPSVM12A*
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_allow_iso_sig_from_equivalent_domain yes`
Usage in goal/ source files	`-lp_allow_iso_sig_from_equivalent_domain=yes`

lp_allow_parent_scope_supply

Specifies whether to allow supply nets created in parent scope to be used in the child scope without providing full hierarchical names.

By default, this parameter is set as `0`, so if you use a supply net in a scope other than the one in which it is created, the UPF parser flags an error.

Set this parameter to allow supply nets created in the parent scope to be used in the child scope without specifying full hierarchical name.

Used by	All rules in the SpyGlass Power Verify solution in the UPF flow
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_allow_parent_scope_supply 1`
Usage in goal/ source files	`-lp_allow_parent_scope_supply=1`

lp_allow_set_design_attr_in_UPF1

Specifies whether to allow the set_design_attrbute command in the UPF 1.0 version.

By default, this parameter is set as no, and the set_design_attrbute command is not allowed in the UPF 1.0 version.

Set this parameter to yes to allow the set_design_attrbute command in the UPF 1.0 version.

Used by	All rules in the SpyGlass Power Verify solution in the UPF flow
Options	0, 1, no, or yes
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_allow_set_design_attr_in_UPF1 yes`
Usage in goal/ source files	`-lp_allow_set_design_attr_in_UPF1=yes`

lp_allow_UPF_21_commands_options

Specifies whether to allow UPF 2.1 commands in UPF file, written in the UPF 2.0 version.

By default, this parameter is set as no. Set this parameter to yes to allow the UPF 2.1 commands in UPF file, written in the UPF 2.0 version.

Currently, the following UPF2.1 commands are supported only when the lp_allow_UPF_21_commands_options parameter is set to yes:

- set_repeater
- set_equivalent
- begin_power_model
- end_power_model
- apply_power_model

The following UPF2.1 options are supported by existing commands when this parameter is set to yes:

- <domain-name> is allowed in –source/-sink option of the set_isolation command:
  - set_isolation –source <domain-name> -sink <domain-name>
- <supply-set> is allowed in –source/-sink option of the set_level_shifter command:
  - set_level_shifter –source <supply-set> -sink <supply-set>
- use_equivalence is supported in the set_isolation, set_level_shifter, and set_repeater commands:
  - set_isolation –use_equivalence
  - set_level_shifter –use_equivalence
  - set_repeater –use_equivalence
- is_macro_cell is supported in the set_design_attibute command.
- is_leaf_cell is supported in the set_design_attibute command.
- Option –applies_to is allowed with –source/-sink options in the set_isolation command
  - set_isolation –applies_to <inputs|outputs|both> –source/-sink
- Option -model is mandatory with -related_power_ports, -related_ground_ports, or -related_bias_ports arguments.
- Option [–domain | -supply] is supported in the add_power_state command
- Option –reconnect is supported in the connect_logic_net command

Used by	All rules in the SpyGlass Power Verify solution in the UPF flow
Options	0, 1, no, or yes
Default	`no`
Example	
Console/Tcl-based usage	`set_parameter lp_allow_UPF_21_commands_options yes`
Usage in goal/ source files	`-lp_allow_UPF_21_commands_options=yes`

lp_allow_UPF_30_commands_options

Specifies whether to allow UPF 3.0 commands in UPF file, written in the UPF 2.0 version.

By default, this parameter is set as `no`. Set this parameter to `yes` to allow the UPF 3.0 commands in UPF file, written in the UPF 2.0 version.

The `lp_allow_UPF_30_commands_options` parameter makes the *connect_supply_net* command as optional for electrically equivalent nets specified in the *set_equivalent* command.

Used by	All rules in the SpyGlass Power Verify solution in the UPF flow
Options	0, 1, no, or yes
Default	`no`
Example	
Console/Tcl-based usage	`set_parameter lp_allow_UPF_30_commands_options 0`
Usage in goal/ source files	`-lp_allow_UPF_30_commands_options=0`

lp_allow_wildcard_in_UPF_2

Specifies whether the *checkUPF_existence* rule reports `Fatal` messages for wildcards used in the UPF commands, apart from the `find_object`

command.

By default this parameter is set to 1 and this rule does not report `Fatal` messages for such cases. Set this parameter to 0 to report `Fatal` messages for wildcards used in the UPF commands, apart from the `find_object` command.

This parameter is used only for UPF version 2.0.

Used by	*checkUPF_existence*
Options	0, 1, no, or yes
Default	`1`
Example	
Console/Tcl-based usage	`set_parameter lp_allow_wildcard_in_UPF_2 0`
Usage in goal/ source files	`-lp_allow_wildcard_in_UPF_2=0`

lp_allow_pin_alias

Specifies whether to allow pin aliases in the *create_power_switch* command, instead of the actual pins in the library cells.

By default, this parameter is set to `no`. This indicates that the *UPF_lowpower07* rule reports a violation message when the pin specified in the *create_power_switch* command is not present in the cell specified in the *map_power_switch* command.

Set this parameter to `yes` to allow pin aliases in the *create_power_switch* command, instead of the actual pins in library cells.

Used by	*UPF_lowpower07*
Options	0, 1, no, or yes
Default	`0`
Example	

Console/Tcl-based usage	`set_parameter lp_allow_pin_alias 1`
Usage in goal/ source files	`-lp_allow_pin_alias=1`

lp_apply_strategy_on_pwr_gnd

Specifies whether to apply isolation/level shifter strategy on power/ground ports.

By default, this parameter is set as `yes` and the isolation/level shifter strategies are applied on power/ground ports.

Set this parameter to `no` to not to apply these strategies.

Used by	All rules in the SpyGlass Power Verify solution in the UPF flow
Options	0, 1, no, or yes
Default	`yes`
Example	
Console/Tcl-based usage	`set_parameter lp_apply_strategy_on_pwr_gnd no`
Usage in goal/ source files	`-lp_apply_strategy_on_pwr_gnd=no`

lp_block_abstraction_dir

Specifies the absolute path of the directory to be created for generated SoC abstractions.

By default, this parameter is set to `spyglass_reports/lowpower/abstract_view/`.

Set this parameter to a desired directory path/name to be created for the generated SoC abstractions.

Used by	*PV_Abstract01*
Options	string

Default	spyglass_reports/lowpower/abstract_view/
Example	
Console/Tcl-based usage	`set_parameter lp_block_abstraction_dir /aview`
Usage in goal/ source files	`-lp_block_abstraction_dir=/aview`

lp_bus_driver

Specify the space-separated list of bus keepers or special bus driver module names for the *LPBUS01* and *LPBUS03* rule respectively.

Used by	*LPBUS01*, *LPBUS03*
Options	Space separated list of string
Default	none
Example	
Console/Tcl-based usage	`set_parameter lp_bus_driver "module_1 module_2"`
Usage in goal/ source files	`-lp_bus_driver="module_1 module_2"`

lp_bus_receiver

Specify the space-separated list of special bus receiver module names for the *LPBUS03* rule.

Used by	*LPBUS03*
Options are	Space separated list of string
Default	none
Example	

Console/Tcl-based usage	`set_parameter lp_bus_receiver "module_1 module_2"`
Usage in goal/ source files	`-lp_bus_receiver="module_1 module_2"`

lp_check_all_ps_in_daisy_chain

Specifies whether the *LPPSW02* rule checks relatively less-on buffers in a power switch enable signal path with respect to all the power switches in the chain.

By default, this parameter is set to `0` and most-on power switch is picked from the chain and all the buffers are validated against that power switch only.

When you set this parameter to `1`, all the buffers in the path are validated to be more-on with respect to each power switch in the chain.

Used by	*LPPSW02*
Options are	0, 1, no, or yes
Default	`0`
Example	
Console/Tcl-based usage	`set_parameter lp_check_all_ps_in_daisy_chain 1`
Usage in goal/ source files	`-lp_check_all_ps_in_daisy_chain=1`

lp_check_aob_on_control

Specifies whether the *LPSVM56B* rule flags always-on buffers (specified using the *always_on_buffer* constraint) not present in the path of control (sleep/save/restore) signal.

By default, the `lp_check_aob_on_control` rule parameter is not set and the `LPSVM56B` rule does not report the always-on buffer instances not present in the path of sleep/restore/save signal.

Used by	*LPSVM56B*
Options are	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_check_aob_on_control 1`
Usage in goal/ source files	`-lp_check_aob_on_control=1`

lp_check_aon_buffer

Specifies whether to check AON buffers.

Used by	*LPERC03A*
Options are	0, 1, no, or yes
Default	1
Example	
Console/Tcl-based usage	`set_parameter lp_check_aon_buffer 0`
Usage in goal/ source files	`-lp_check_aon_buffer=0`

lp_check_control_path_of_fine_grain_psw

Specifies whether the *LPPSW02* and *LPPSW01* rules report violations for fine-grain power switches.

By default, the `lp_check_control_path_of_fine_grain_psw` parameter is set to `yes` and the rules report violations for fine-grain power switches.

Set this parameter to `no` to not report such violations.

Used by	*LPPSW01*, *LPPSW02*
Options	0, 1, no, or yes

Default	yes
Example	
Console/Tcl-based usage	set_parameter lp_check_control_path_of_fine_grain_psw no
Usage in goal/source files	-lp_check_control_path_of_fine_grain_psw=no

lp_check_csn_on_internal_pg_pin

Specifies whether the *UPF_lowpower15* rule reports the missing connect_supply_net command on an internal power/ground pin.

By default, the lp_check_csn_on_internal_pg_pin parameter is set to yes and the *UPF_lowpower15 rule* reports the missing connect_supply_net command on the internal power/ground pin for multi-supply cell.

Set this parameter to no to not to report such violations.

Used by	*UPF_lowpower15*
Options	0, 1, no, or yes
Default	yes
Example	
Console/Tcl-based usage	set_parameter lp_check_csn_on_internal_pg_pin no
Usage in goal/source files	-lp_check_csn_on_internal_pg_pin=no

lp_check_data_pin_of_els

Specifies whether the *LPERC01B* and *LPERC02B* rules check data pins of the enable level shifter cells.

By default, the lp_check_data_pin_of_els parameter is set to no. Set this parameter to yes to check data pins of the enable level shifter cells.

Used by	*LPERC01B*, *LPERC02B*
Options	0, 1, no, or yes
Default	`yes`
Example	
Console/Tcl-based usage	`set_parameter lp_check_data_pin_of_els yes`
Usage in goal/ source files	`-lp_check_data_pin_of_els=yes`

lp_check_domain_equivalence

Specifies whether *LPCONN07A* and *LPCONN07B* rules check for domain equivalence.

By default, the `lp_check_domain_equivalence` parameter is set as `0` and the *LPCONN07A* and *LPCONN07B* rules check if the domains are equal based on their primary supplies.

Set this parameter to `1` to enable the *LPCONN07A* and *LPCONN07B rules* to check buffers that are existing in the same domain while checking for feed through paths.

Used by	*LPCONN07A*, *LPCONN07B*
Options	0, 1, no, or yes
Default	`0`
Example	
Console/Tcl-based usage	`set_parameter lp_check_domain_equivalence 0`
Usage in goal/ source files	`-lp_check_domain_equivalence=0`

lp_check_driver_receiver_supply

Specifies whether to check the missing driver_supply/receiver_supply options of the *set_port_attribute* command.

By default, the `lp_check_driver_receiver_supply` parameter is

set to no and the *UPF_lowpower20* rule does not report the missing driver_supply/receiver_supply for a top level port.

Set this parameter to yes to report a top level port with missing driver_supply/receiver_supply options of the *set_port_attribute* command.

Used by	*UPF_lowpower20*
Options are	0, 1, no, or yes
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_check_driver_receiver_supply yes`
Usage in goal/ source files	`-lp_check_driver_receiver_supply=yes`

lp_check_dual_enable_psw

Specifies whether the *LPPLIB17* checks the dual enable power switch based on the library specification and not on the order of occurrence of the -control_port argument of the create_power_switch command.

By default, the lp_check_dual_enable_psw parameter is set to no and the rule considers first -control_port argument for first enable pin and second -control_port argument for the second enable pin. So the default behavior is based on the order of the -control_port argument.

Set this parameter to yes to check the connection of control pins of dual enable power switch based on the control net, given in the -control_port argument of the create_power_switch command, against the port name coming from library.

This parameter removes the dependency of order and the correct -control_port argument is decided based on the port name coming from library.

Used by	*LPPLIB17*
Options	0, 1, no, or yes

Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_check_dual_enable_psw yes`
Usage in goal/ source files	`-lp_check_dual_enable_psw=yes`

lp_check_equally_on_supplies_for_aon_pins

Specifies whether the *LPPLIB06* rule should consider equally-on supplies as more-on supplies.

By default, this parameter is set to no and the *LPPLIB06* rule does not consider equally-on supplies as more-on supplies.

Set this parameter to yes to consider equally-on supplies as more-on supplies in the LPPLIB06 rule.

Used by	*LPPLIB06*
Options are	0, 1, no, or yes
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_check_equally_on_supplies_for_aon_pins yes`
Usage in goal/ source files	`-lp_check_equally_on_supplies_for_aon_pins=yes`

lp_check_equivalent_domain_for_main_rail

Specifies whether the *LPLSH08* rule checks the equivalency of source/ destination domain and level shifter cell domain.

By default, this parameter is set to yes and the rule checks if the source/ destination domain and domain in which level shifter cell will be inferred are equivalent. These domains should be similar, voltage wise, and level shifter domain should not be less on than source/destination domain.

Set this parameter to no to exactly match the source/destination domain

and domain in which level shifter cell will be inferred.

Used by	*LPLSH08*
Options are	0, 1, no, or yes
Default	`1`
Example	
Console/Tcl-based usage	`set_parameter lp_check_equivalent_domain_for_main_rail 0`
Usage in goal/ source files	`-lp_check_equivalent_domain_for_main_rail=0`

lp_check_internal_pg_pin_connection

Specifies whether the *LPPLIB06* and *LPPLIB15* rules check the PG connections of internal PG pin whose direction is output.

By default, this parameter is set to `no` and the PG connections of PG pin with `pg_type internal_power/internal_ground` are not checked.

Set this parameter to `yes` to check the PG connections of internal PG pin whose direction is output.

Used by	*LPPLIB06*, *LPPLIB15*
Options are	0, 1, no, or yes
Default	`no`
Example	
Console/Tcl-based usage	`set_parameter lp_check_internal_pg_pin_connection yes`
Usage in goal/ source files	`-lp_check_internal_pg_pin_connection=yes`

lp_check_ls_strategy_presence

Specifies whether *LPCONN07A* and *LPCONN07B* rules check for the presence of level shifter strategy on any of the ports on the domain boundary from source to destination.

By default, the `lp_check_ls_strategy_presence` parameter is set as `0` and the *LPCONN07A* and *LPCONN07B* rules do not check for the presence of level shifter strategy on any of the ports on the domain boundary from source to destination.

Set this parameter to `1` to enable the *LPCONN07A* and *LPCONN07B rules* to check for the presence of level shifter strategy on any of the ports on the domain boundary from source to destination.

Used by	*LPCONN07A*, *LPCONN07B*
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_check_ls_strategy_presence 0`
Usage in goal/ source files	`-lp_check_ls_strategy_presence=0`

lp_check_ls_supply_at_rtl

Specifies whether the *LPPLIB04* rule checks for inconsistency between PG pins of level shifter cells and source/sink domain primary supplies in a partial PG netlist design.

By default, the `lp_check_ls_supply_at_rtl` parameter is set as `0` and the *LPPLIB04* rule does not check for the level shifters that are connected to improper supply for partial PG netlist.

Set this parameter to 1 to check for inconsistency between PG pins of level shifters and source/sink domain primary supplies.

Used by	*LPPLIB04*
Options	0, 1, no, or yes
Default	0

Example	
Console/Tcl-based usage	`set_parameter lp_check_ls_supply_at_rtl 1`
Usage in goal/ source files	`-lp_check_ls_supply_at_rtl=1`

lp_check_enable_pin

Specifies whether the *LPSVM53* rule flags for the enable pins of the isolation cells or clamp level shifters.

By default, the `lp_check_enable_pin` parameter is set as `0` and the *LPSVM53* rule does not check the enable pin of the isolation cells or clamp level shifters.

Used by	*LPSVM53*
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_check_enable_pin 1`
Usage in goal/ source files	`-lp_check_enable_pin=1`

lp_check_feedthrough_path

Specifies whether the *LPSVM52* rule flags clock nets that have only always-on buffers (specified using the *always_on_buffer* constraint) in their feedthrough path.

By default, the `lp_check_feedthrough_path` rule parameter is set and the LPSVM52 rule ignores clock nets that have only always-on buffers (specified using the `always_on_buffer` constraint) in their feedthrough path. Unset the `lp_check_feedthrough_path` rule parameter to have the LPSVM52 rule flag such clock nets also.

Used by	*LPSVM52*
Options	0, 1, no, or yes
Default	1
Example	
Console/Tcl-based usage	`set_parameter lp_check_feedthrough_path 0`
Usage in goal/ source files	`-lp_check_feedthrough_path=0`

lp_check_hier

Specifies whether the *LPSVM37* rule checks the complete hierarchy under the specified region or just the specified region without traversing the hierarchy.

By default, the `lp_check_hier` rule parameter is set to 1 and the LPSVM37 rule checks the complete hierarchy. Set the `lp_check_hier` rule parameter to `0` to restrict the checking to the specified level only.

Used by	*LPSVM37*
Options	0, 1, no, or yes
Default	1
Example	
Console/Tcl-based usage	`set_parameter lp_check_hier 0`
Usage in goal/ source files	`-lp_check_hier=0`

lp_check_iso_hier

Causes the *LPSVM55* rule to flag isolation cell instances when they are not present exactly at the specified instance hierarchy.

By default, the `lp_check_iso_hier` rule parameter is not set and the LPSVM55 rule does not flag isolation cell instances when they are not present exactly at the specified instance hierarchy. Set the `lp_check_hier` rule parameter to flag isolation cell instances not present exactly at the specified instance hierarchy.

Used by	*LPSVM55*
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_check_iso_hier 0`
Usage in goal/ source files	`-lp_check_iso_hier=0`

lp_check_isocell

By default, the lp_check_isocell rule parameter is set and the LPSVM49 rule will not flag for the net crossings from the power domain gate to proper isolation gate.

Unset this parameter to enable the LPSVM49 violations for the net crossings from the power domain gate to the proper isolation gate.

Used by	*LPSVM49*
Options	0, 1, no, or yes
Default	1
Example	
Console/Tcl-based usage	`set_parameter lp_check_isocell 0`
Usage in goal/source files	`-lp_check_isocell=0`

lp_check_map_iso_cmd

By default, the lp_check_map_iso_cmd parameter is set to 1 and the *UPF_lowpower05* rule does not ignore the missing map_isolation_cell command corresponding to a set_isolation command in the UPF.

Set the lp_check_map_iso_cmd parameter to 0 to ignore the missing map_isolation_cell command corresponding to a set_isolation command in the UPF.

Used by	*UPF_lowpower05*
Options	0, 1, no, or yes
Default	1
Example	
Console/Tcl-based usage	`set_parameter lp_check_map_iso_cmd 0`
Usage in goal/source files	`-lp_check_map_iso_cmd=0`

lp_check_map_ls_cmd

By default, the `lp_check_map_ls_cmd` parameter is set to `1` and the *UPF_lowpower06* rule does not ignore the missing `map_level_shifter_cell` command corresponding to a `set_level_shifter` command in the UPF.

Set the `lp_check_map_ls_cmd` parameter to `0` to ignore the missing `map_level_shifter_cell` command corresponding to a `set_level_shifter` command in the UPF.

Used by	*UPF_lowpower06*
Options	0, 1, no, or yes
Default	`1`
Example	
Console/Tcl-based usage	`set_parameter lp_check_map_ls_cmd 0`
Usage in goal/ source files	`-lp_check_map_ls_cmd=0`

lp_check_name_based_conflict

Specifies if the *LPSUP03* rule reports a violation if supplies specified in the `set_related_supply_net` command and the `connect_supply_net` commands have different names.

By default, the parameter is set to `no` and the *LPSUP03* rule does not report a violation for such name mismatches.

Used by	*LPSUP03*
Options	no, yes
Default	`no`
Example	

Console/Tcl-based usage	`set_parameter lp_check_name_based_conflict yes`
Usage in goal/ source files	`-lp_check_name_based_conflict=yes`

lp_check_ps_loc

Causes the *LPPLIB17* rule to check for power switch location.

By default, the `lp_check_ps_loc` parameter is not set and the `LPPLIB17` rule will not flag for the power switches which are located in a domain other than their switching power domains.

Used by	*LPPLIB17*
Options	0, 1, no, or yes
Default	`0`
Example	
Console/Tcl-based usage	`set_parameter lp_check_ps_loc 0`
Usage in goal/ source files	`-lp_check_ps_loc=0`

lp_check_pwr_gnd_to_macro_without_prd

Specifies if the rules should ignore crossings between supply to macro and pad cells without domain boundary.

By default, the `lp_check_pwr_gnd_to_macro_without_prd` parameter is set to `yes` and rules do not ignore crossings between supply to macro and pad cells without domain boundary.

Set this parameter to `no` to ignore crossings between tie-high/low and macro cell (`is_macro_cell: true`)/pad cell (`pad_cell: true`), without domain boundary. Level shifter and isolation checking for these crossings at the RTL and Netlist levels will be ignored.

Used by	*LPLSH05A*, *LPLSH05B*, *LPSVM04A*, *LPSVM04B*, *LPSVM04C*, *LPISO03A* , *LPISO03B*, *LPISO04A*, *LPSVM08B*, *LPSVM08C*, *LPSVM22*
Options	0, 1, no, or yes
Default	`yes`
Example	
Console/Tcl-based usage	`set_parameter lp_check_pwr_gnd_to_macro_without_prd no`
Usage in goal/ source files	`-lp_check_pwr_gnd_to_macro_without_prd=no`

lp_check_pass_gate_on_pwr_gnd

Enables the *LPCONN06* rule to check crossings involving supply nets.

By default, the `lp_check_pass_gate_on_pwr_gnd` parameter is not set and the *LPCONN06* rule does not check crossings involving supply nets (1'b1/1'b0 and UPF supply nets).

Set this parameter to `1` to check such supply nets.

Used by	*LPCONN06*
Options	0, 1, no, or yes
Default	`0`
Example	
Console/Tcl-based usage	`set_parameter lp_check_pass_gate_on_pwr_gnd 1`
Usage in goal/ source files	`-lp_check_pass_gate_on_pwr_gnd=1`

lp_check_relative_aon_buffers

Specifies whether to check relative always-on buffers in the *LPSVM53* rule.

By default, the `lp_check_relative_aon_buffers` parameter is not set and the *LPSVM53* rule does not check for relative always-on buffers.

Set this parameter to 1 to enable the *LPSVM53* rule to check for relative always-on buffers.

Used by	*LPSVM53*
Options are	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_check_relative_aon_buffers 1`
Usage in goal/ source files	`-lp_check_relative_aon_buffers=1`

lp_check_same_biasnet

Set this parameter to 1 to check for the following:

- *LPPLIB18A*: Use to check if the connection of the power pin and the related bias power pin are the same.
- *LPPLIB18B*: Use to check if the connection of the ground pin and the related bias ground pin are the same.

Used by	*LPPLIB18A*, *LPPLIB18B*
Options are	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_check_same_biasnet 1`
Usage in goal/ source files	`-lp_check_same_biasnet=1`

lp_check_same_srsn_supply

Enables the *LPPLIB14* rule to report a violation for mismatch between supply net connected to the input signal pin (verilog connection) and

related supply, through `set_related_supply_net` or `connect_supply_net` UPF command.

By default, the `lp_check_same_srsn_supply` parameter is set to `no` and the *LPPLIB14* rule does not report such cases. Set the value of this parameter to `yes` to report these cases.

Used by	*LPPLIB14*
Options	0, 1, no, or yes
Default	`no`
Example	
Console/Tcl-based usage	`set_parameter lp_check_same_srsn_supply yes`
Usage in goal/ source files	`-lp_check_same_srsn_supply=yes`

lp_check_same_voltage_path

Causes the *LPLSH04* rule to report crossings from a domain to another domain where there is no voltage change.

By default, the `lp_check_same_voltage_path` parameter is not set and the *LPLSH04* rule does not report such crossings.

Used by	*LPLSH04*
Options	0, 1, no, or yes
Default	`0`
Example	
Console/Tcl-based usage	`set_parameter lp_check_same_voltage_path 0`
Usage in goal/ source files	`-lp_check_same_voltage_path=0`

lp_check_src_driver_supply_name

Causes the *LPERC01A* rule to check if the supply nets of driver and receiver

are the same.

By default, the lp_check_src_driver_supply_name parameter is set to 0 and the *LPERC01A* rule does not check if the supply nets of driver and receiver are the same.

Set the value of this parameter to `1` or `yes` to report a violation when the supply nets of driver and receiver are different.

Used by	*LPERC01A*
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_check_src_driver_supply_name 1`
Usage in goal/ source files	`-lp_check_src_driver_supply_name=1`

lp_check_valid_iso

Causes the *LPSVM08A* and *LPSVM60* rules to check whether the isolation cell is a NAND/NOR/AND/OR gate by functionality.

By default, the `lp_check_valid_iso` parameter is not set and the LPSVM08A rule only checks for a valid isolation cell.

When you set the `lp_check_valid_iso` rule parameter, the LPSVM08 rule further checks that the isolation cell is a NAND/NOR/AND/OR gate by functionality. If the isolation cell is found to be a gate other than the allowed gate types, the isolation cell is assumed to be an invalid isolation cell.

Used by	*LPSVM08A*, *LPSVM60*
Options	0, 1, no, or yes
Default	0
Example	

Console/Tcl-based usage	`set_parameter lp_check_valid_iso 0`
Usage in goal/ source files	`-lp_check_valid_iso=0`

lp_check_with_output

Causes the *LPSVM24* rule to flag when the enable of a level shifter lies in a different voltage domain from that of the instance or port to which the output of the level shifter is connected.

By default, the `lp_check_with_output` rule parameter is not set and the LPSVM24 rule does not flag such cases.

Used by	*LPSVM24*
Options	`0` or `1`
Default	`0`
Example	
Console/Tcl-based usage	`set_parameter lp_check_with_output 0`
Usage in goal/ source files	`-lp_check_with_output=0`

lp_complex_iso_logic

Specifies whether the complex isolation logic should be recognized by the *LPSVM08*, *LPSVM09*, *LPSVM10*, and *LPSVM51* rules.

By default, this rule parameter is set to `0` and these rules assume that the isolation cells are simple cells comprising of a single gate, only. Set the rule parameter to 1 to have these rules assume that isolation cells are complex cells.

Used by	*LPSVM08*, *LPSVM09*, *LPSVM10*, *LPSVM51*
Options	0, 1, no, or yes

Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_complex_iso_logic 0`
Usage in goal/ source files	`-lp_complex_iso_logic=0`

lp_complex_level_shifter

Specifies whether any complex level-shifter design units should be recognized by the *LPSVM04A*, *LPSVM04B*, *LPSVM04C*, *LPSVM04E*, *LPSVM04E*, and *LPSVM17* rules.

By default, the `lp_complex_level_shifter` rule parameter is not set and these rules assume that all level-shifters are simple cells comprising a single gate.

Used by	*LPSVM04A*, *LPSVM04B*, *LPSVM04C*, *LPSVM04E*, *LPSVM04E*, *LPSVM17*
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_complex_level_shifter 0`
Usage in goal/ source files	`-lp_complex_level_shifter=0`

lp_constant_els_as_iso_cell

Specifies whether an enable level shifter with enable pin tied to constant value should be considered as an isolation cell or not.

By default, the enable level shifter with enable pin tied to constant value is not considered as an isolation cell.

Set the `lp_constant_els_as_iso_cell` parameter to `1` to consider this type of cell as an isolation cell also.

Used by	All rules in the SpyGlass Power Verify solution
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_constant_els_as_iso_cell 1`
Usage in goal/ source files	`-lp_constant_els_as_iso_cell=1`

lp_consider_inverted_output

Specifies if the *LPCONN04C* rule should consider inverted outputs NAND, NOR, XNOR cells as inverter plus logic.

By default, the parameter is set to `no`. Set this parameter to `yes` to enable the *LPCONN04C* rule consider inverted outputs NAND, NOR, XNOR cells as inverter plus logic. This impacts the inverter count in the path.

Used by	*LPCONN04C*
Options	no, yes
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_consider_inverted_output yes`
Usage in goal/ source files	`-lp_consider_inverted_output=yes`

lp_csn_on_ground_pg_pin

Specifies whether to consider the *connect_supply_net* UPF command for the ground pin.

By default the value of the `lp_csn_on_ground_pg_pin` parameter is set to `0`, *connect_supply_net* is not considered for the ground pin. Set this parameter to `1` to consider the *connect_supply_net* command for the PG

ground pin.

Used by	All rules in the SpyGlass Power Verify solution
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_csn_on_ground_pg_pin 1`
Usage in goal/ source files	`-lp_csn_on_ground_pg_pin=1`

lp_disable_lib_attr_read

Specifies whether the power related library attributes are read from library or not. By default, this parameter is set to 0 and the power related library attribute are read from the library.

If you want to disable the data to be auto-inferred from the library or you want to provide the information by SGDC/CPF/UPF only, set `lp_disable_lib_attr_read` rule parameter to 1.

Used by	All rules in the SpyGlass Power Verify solution
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_disable_lib_attr_read 0`
Usage in goal/ source files	`-lp_disable_lib_attr_read=0`

lp_dump_ls_in_multi_supply_rpt

Specifies whether level shifter information be provided in the *lp_multi_supply_instance* report.

Used by	*UPF_lowpower15*
Options	0, 1, no, or yes
Default	1 or yes
Example	
Console/Tcl-based usage	`set_parameter lp_dump_ls_in_multi_supply_rpt 0`
Usage in goal/ source files	`-lp_dump_ls_in_multi_supply_rpt=0`

lp_dump_ps_in_multi_supply_rpt

Specifies whether power switch information be provided in the *lp_multi_supply_instance* report for the *UPF_lowpower15* rule.

By default this parameter is set as 1 or yes and the *UPF_lowpower15* rule provides power switch information.

Set this parameter as 0 to enable the *UPF_lowpower15* rule to not to provide the power switch information in the *lp_multi_supply_instance* report.

Used by	*UPF_lowpower15*
Options	0, 1, no, or yes
Default	1 or yes
Example	
Console/Tcl-based usage	`set_parameter lp_dump_ps_in_multi_supply_rpt 0`
Usage in goal/ source files	`-lp_dump_ps_in_multi_supply_rpt=0`

lp_dump_scope_supply_in_domain_info

Specifies if the information on the scope supplies are included in the lp_domain_info.rpt report. These supplies are available in a domain because they are declared in same scope as domain.

By default this parameter is set to no and information on the scope

supplies are included in the lp_domain_info.rpt report. Set this parameter to yes to include such information on scope supplies.

Used by	*LP_DECOMPILE_CONSTR*
Options	no, or yes
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_dump_scope_supply_in_domain_info yes`
Usage in goal/ source files	`-lp_dump_scope_supply_in_domain_info=yes`

lp_dump_unrelated_macro_cell_pgpin_in_multi_supply_rpt

Specifies whether the *UPF_lowpower15* rule should check dangling PG pins of macro cells that are not related to any signal pin in.

By default this parameter is set to no and the *UPF_lowpower15* rule does not report unconnected PG pins of macro cells that are not related to data pins.

Set this parameter as yes to report such cases.

Used by	*UPF_lowpower15*
Options	0, 1, no, or yes
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_dump_unrelated_macro_cell_pgpin_in_multi_supply_rpt yes`
Usage in goal/ source files	`-lp_dump_unrelated_macro_cell_pgpin_in_multi_supply_rpt=yes`

lp_enable_buf_check

Enables you to check for signals that are not buffered by a relatively-AON power. Set the value to 1 to check such signals.

Used by	*LPPSW04*
Options	`0`, `1`, `no`, or `yes`
Default	`0` or `no`
Example	
Console/Tcl-based usage	`set_parameter lp_enable_buf_check 1`
Usage in goal/ source files	`-lp_enable_buf_check=0`

lp_find_objects_match_count

When the `find_objects` command matches more than 1000 results, the wildcards are not expanded in-line.

Default value is 1000. If the SpyGlass hangs during the execution of the *LP_POWERDATA_INFO* and *LP_POWERDATA_READ* rules and the `find_objects` command used in the UPF file, you can try reducing the value of this parameter so that the wildcards are not expanded in-line.

Used by	*LP_POWERDATA_INFO*, *LP_POWERDATA_READ*
Options	`Any positive integer`
Default	`1000`
Example	
Console/Tcl-based usage	`set_parameter lp_find_objects_match_count 101`
Usage in goal/ source files	`-lp_find_objects_match_count=101`

lp_flag_iso_cell_in_crossing

Specifies if the *LPISO04A* rule should report information related to isolation cells present in a crossings.

By default this parameter is set to `no` and the *LPISO04A* rule does not report the information related to isolation cells present in a crossings.

Set this parameter to `yes` to report information related to isolation cells present in crossings.

Used by	*LPISO04A*
Options	0, 1, no, or yes
Default	`no`
Example	
Console/Tcl-based usage	`set_parameter lp_flag_iso_cell_in_crossing yes`
Usage in goal/ source files	`-lp_flag_iso_cell_in_crossing=yes`

lp_flag_iso_ls_strategy_on_domain_sub_hier

Specifies if the *UPF_lowpower12* rule should report the isolation/level shifter strategy specified on domain sub-hierarchy.

By default this parameter is set to `no`, and this parameter does not report a violation for the isolation/level shifter strategy specified on domain sub-hierarchy.

Set this parameter to `yes` to report violation for such cases.

Used by	*UPF_lowpower12*
Options	0, 1, no, or yes
Default	`no`
Example	
Console/Tcl-based usage	`set_parameter lp_flag_iso_ls_strategy_on_domain_sub_hier yes`
Usage in goal/ source files	`-lp_flag_iso_ls_strategy_on_domain_sub_hier=yes`

lp_flag_iso_ls_crossing

Specifies if the *LP_MULTI_DOMAIN_CROSSING_CHECK* rule should report violation for intermediate domain irrespective of isolation/level shifter requirement.

By default, this parameter is set to `yes` and the intermediate domain is treated as different domain if there is isolation/level shifter requirement at the both sides, that is, source to intermediate domain and intermediate domain to destination domain. Set the value of this parameter to `no` to report violation for intermediate domain irrespective of isolation/level shifter requirement.

Used by	*LP_MULTI_DOMAIN_CROSSING_CHECK*
Options	0, 1, no, or yes
Default	`yes`
Example	
Console/Tcl-based usage	`set_parameter lp_flag_iso_ls_crossing no`
Usage in goal/ source files	`-lp_flag_iso_ls_crossing=no`

lp_flag_ls_strategy_for_multi_fanout

Specifies if the *LPLSH05A* and *LPLSH05B* rules should report invalid level shifter strategy at the source.

By default this parameter is set to `no` and the above rules do not report violation for common level shifter strategy applied on source, which is going to a mixed fanout, as the strategy cannot be valid for both fanouts at the same time.

Set this parameter to `yes` to report violation for such cases.

Used by	*LPLSH05A*, *LPLSH05B*
Options	0, 1, no, or yes
Default	`no`

Example	
Console/Tcl-based usage	`set_parameter lp_flag_ls_strategy_for_multi_fanout yes`
Usage in goal/ source files	`-lp_flag_ls_strategy_for_multi_fanout=yes`

lp_flag_missing_wildcard_in_set_retention

Specifies if the *checkUPF_existence* rule should report vector signals, mentioned in the elements list of the `set_retention` command, that are not specified with a wildcard.

By default this parameter is set to `no`, and this parameter does not report a violation if any vector signal, when specified in the `elements` field of the `set_retention` command, is not specified with a wildcard.

Set this parameter to `yes` to report violation for such cases.

Used by	*checkUPF_existence*
Options	0, 1, no, or yes
Default	`no`
Example	
Console/Tcl-based usage	`set_parameter lp_flag_missing_wildcard_in_set_retention yes`
Usage in goal/ source files	`-lp_flag_missing_wildcard_in_set_retention=yes`

lp_flag_missing_power_switch_lib

Specifies if the *checkUPF_existence* rule should report violations at the RTL/ Netlist level for missing library cells as specified in the `map_power_switch` command.

By default, this parameter is set to `no` and violations are not reported at the RTL/Netlist level for missing library cells as specified in the `map_power_switch` command.

Set this parameter to `yes` to report violations in such cases.

Used by	*checkUPF_existence*
Options	0, 1, no, or yes
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_flag_missing_power_switch_lib yes`
Usage in goal/ source files	`-lp_flag_missing_power_switch_lib=yes`

lp_flag_missing_timing_arc_on_pin

Specifies if the *LPPSW03* and the *LPPSW04* rules should report missing timing arc for pins found in enable signal path of power switches.

By default the parameter is set to no. Set this parameter to yes to enable the rules report missing timing arc for pins found in enable signal path of power switches.

Used by	*LPPSW03*, *LPPSW04*
Options	yes or no
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_flag_missing_timing_arc_on_pin yes`
Usage in goal/ source files	`-lp_flag_missing_timing_arc_on_pin=yes`

lp_flag_multi_elements_in_create_power_domain

Specifies if the *UPFSEM_44* rule should report violations for the multiple elements specified in the *create_power_domain* command.

By default, this parameter is set to no.

Set this parameter to yes to report if multiple elements have been

specified in the `create_power_domain` UPF command.

Used by	*UPFSEM_44*
Options	0, 1, no, or yes
Default	`no`
Example	
Console/Tcl-based usage	`set_parameter lp_flag_multi_elements_in_create_power_domain yes`
Usage in goal/ source files	`-lp_flag_multi_elements_in_create_power_domain=yes`

lp_flag_pd_outputs

Specifies whether the power domain rules (*LPSVM08A*, *LPSVM08B*, *LPSVM08C*, *LPISO04A*, *LPISO04C*, *LPISO06A*, *LPSVM09*, and *LPSVM10* rules) flag outputs of power domain that are directly connected to primary ports.

By default, these rules flag the outputs of power domain that are directly connected to primary output ports. Set the `lp_flag_pd_outputs` parameter to ignore such outputs:

Used by	*LPSVM08A*, *LPSVM08B*, *LPSVM08C*, *LPISO04A*, *LPISO04C*, *LPISO06A*, *LPSVM09*, *LPSVM10*
Options	0, 1, no, or yes
Default	`1`
Example	
Console/Tcl-based usage	`set_parameter lp_flag_pd_outputs 0`
Usage in goal/ source files	`-lp_flag_pd_outputs=0`

lp_flag_rtlc_cell

Specifies if the *LP_MULTI_DOMAIN_CROSSING_CHECK* rule should report

violation for SpyGlass generated buffers/inverters.

By default, this parameter is set to `no`. Set the value of this parameter to `yes` to report violation for SpyGlass generated buffers/inverters.

Used by	*LP_MULTI_DOMAIN_CROSSING_CHECK*
Options	0, 1, no, or yes
Default	`no`
Example	
Console/Tcl-based usage	`set_parameter lp_flag_rtlc_cell yes`
Usage in goal/ source files	`-lp_flag_rtlc_cell=yes`

lp_flag_unconnected_nets

Specifies how the *LPSVM08A*, *LPSVM08B*, *LPSVM08C*, *LPISO04A*, *LPISO04C*, *LPISO06A*, *LPLSH03*, *LPLSH04*, *LPSVM04A*, *LPSVM04B*, *LPSVM04C*, *LPSVM04E*, *LPSVM04E*, *LPSVM09*, *LPSVM10*, *LPSVM60*, *LPPLIB04*, *LPPLIB05*, *LPLSH07*, *LPPLIB07*, and *LP_MULTI_DOMAIN_CROSSING_CHECK* rules process unconnected nets.

By default, the `lp_flag_unconnected_nets` rule parameter is not set and the LPSVM04A and LPSVM04B rules ignore unconnected nets at voltage crossings, the LPSVM08A, LPSVM08B, LPSVM08C, LPSVM09, and LPSVM10 rules ignore unconnected outputs of power domain, and the LPPLIB04, LPPLIB05, and LPPLIB07 rules ignore level-shifters with unconnected outputs.

Set the `lp_flag_unconnected_nets` rule parameter to have these rules consider the objects ignored by default.

Used by	*LPSVM08A*, *LPSVM08B*, *LPSVM08C*, *LPISO04A*, *LPISO04C*, *LPISO06A*, *LPLSH03*, *LPLSH04*, *LPSVM04A*, *LPSVM04B*, *LPSVM04C*, *LPSVM04E*, *LPSVM04E*, *LPSVM09*, *LPSVM10*, *LPSVM60*, *LPPLIB04*, *LPPLIB05*, *LPPLIB07*, *LPLSH07*, *LP_MULTI_DOMAIN_CROSSING_CHECK*
Options	0, 1, no, or yes

Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_flag_unconnected_nets 0`
Usage in goal/source files	`-lp_flag_unconnected_nets=0`

lp_flag_undriven_nets

Specifies how the *LPSVM08A*, *LPSVM08B*, *LPSVM08C*, *LPISO04A*, *LPISO04C*, *LPISO06A*, *LPLSH03*, *LPLSH04*, *LPSVM04A*, *LPSVM04B*, *LPSVM04C*, *LPSVM04E*, *LPSVM04E*, *LPSVM09*, *LPSVM10*, *LPSVM47*, *LPSVM60*, *LPPLIB04*, *LPPLIB05*, *LPPLIB07*, and *LP_MULTI_DOMAIN_CROSSING_CHECK* rules process un-driven nets.

By default, the `lp_flag_undriven_nets` parameter is not set and the LPSVM04A and LPSVM04B rules ignore undriven nets at voltage crossings, the LPSVM08A, LPSVM08B, LPSVM08C, LPSVM09, and LPSVM10 rules ignore undriven outputs of power domain, and the LPPLIB04, LPPLIB05, and LPPLIB07 rules ignore level-shifters with undriven inputs.

Set the `lp_flag_undriven_nets` parameter to have these rules consider the objects ignored by default.

Used by	*LPSVM08A*, *LPSVM08B*, *LPSVM08C*, *LPISO04A*, *LPISO04C*, *LPISO06A*, *LPLSH03*, *LPLSH04*, *LPSVM04A*, *LPSVM04B*, *LPSVM04C*, *LPSVM04E*, *LPSVM04E*, *LPPLIB04*, *LPPLIB05*, *LPPLIB07*, *LPSVM09*, *LPSVM10*, *LPSVM47*, *LPSVM60*, *LP_MULTI_DOMAIN_CROSSING_CHECK*
Options	0, 1, no, or yes
Default	0
Example	

Console/Tcl-based usage	`set_parameter lp_flag_undriven_nets 0`
Usage in goal/ source files	`-lp_flag_undriven_nets=0`

lp_flag_violation_on_antenna_cell

Specifies whether the *LPERC01*, *LPERC02*, *LPERC03*, *LPERC04* rules should report violations on antenna cells.

By default, this parameter is set to `no`, and these rules do not report violations on antenna cells.

Set the value of this parameter to `yes` to perform rule checking on antenna cells for above mentioned rules.

Used by	*LPERC02*, *LPERC02*, *LPERC03*, *LPERC04*
Options	0, 1, no, or yes
Default	`no`
Example	
Console/Tcl-based usage	`set_parameter lp_flag_violation_on_antenna_cell yes`
Usage in goal/ source files	`-lp_flag_violation_on_antenna_cell=yes`

lp_gen_block_abstraction_format

Specifies if the *PV_Abstract01* rule generates block power model in the LIB or UPF format.

By default, this parameter is set to `lib`. Set this parameter to `upf` to generate the UPF file. Only one of the formats, either `upf` or `lib`, can be generated during single execution.

Used by	*PV_Abstract01*
Options	lib, upf
Default	`lib`
Example	
Console/Tcl-based usage	`set_parameter lp_gen_block_abstraction_format upf`
Usage in goal/ source files	`-lp_gen_block_abstraction_format=lib`

lp_gen_block_abstraction_in_bit_blasted_way

Specifies if the *PV_Abstract01* rule generates vector ports in LIB file in bit-blasted format or not. By default, this parameter is set to `0`. Set this parameter to `1` to enable bit-blasting.

Used by	*PV_Abstract01*
Options	0, 1, yes, no
Default	`0`
Example	
Console/Tcl-based usage	`set_parameter lp_gen_block_abstraction_in_bit_blasted_way 1`
Usage in goal/ source files	`-lp_gen_block_abstraction_in_bit_blasted_way=1`

lp_generate_missing_csn_violation

Specifies whether the *UPF_lowpower15* rule reports violations instead of generating report.

By default, this parameter is set to no. Set the `lp_generate_missing_csn_violation` parameter to yes to enable the UPF_lowpower15 rule to report violations instead of generating report.

Used by	*UPF_lowpower15*
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_generate_missing_csn_violation 1`
Usage in goal/source files	`-lp_generate_missing_csn_violation=1`

lp_genvcdfile

Specifies the name of the VCD file that is to be translated to SpyGlass activity format by the *LPSVM42* and *LPSVM43* rules.

You can specify only a single VCD file name or a space-separated list of VCD file names enclosed in double quotes.

The mandatory LPCheckVCD rule checks the specified VCD files for syntax errors. The VCD file with syntax errors is ignored and the corresponding rules are not run.

Used by	*LPSVM42*, *LPSVM43*
Options	Space separated file names
Default	none
Example	
Console/Tcl-based usage	`set_parameter lp_genvcdfile "vcd_file_name"`
Usage in goal/source files	`-lp_genvcdfile="vcd_file_name"`

lp_hdlin_enable_upf_compatible_naming

Enables the *checkUPF_existence* rule to support complex `struct` naming in System Verilog.

By default, this parameter is set to `0`.

Set this parameter to `1` to support `struct` naming in System Verilog with

"." to access the member variable of the struct.

Used by	*checkUPF_existence*
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_hdlin_enable_upf_compatible_naming 1`
Usage in goal/source files	`-lp_hdlin_enable_upf_compatible_naming=1`

lp_hdlin_sv_union_member_naming

Enables the *checkUPF_existence* rule to support complex union naming in System Verilog.

By default, this parameter is set to 0.

Set this parameter to 1 (along with the *lp_hdlin_enable_upf_compatible_naming* parameter also set to 1), to support union naming in System Verilog with "." to access the member variable of the union.

Used by	*checkUPF_existence*
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_hdlin_sv_union_member_naming 1`
Usage in goal/source files	`-lp_hdlin_sv_union_member_naming=1`

lp_ignore_b2b_ls

Specifies whether level shifters connected back to back in the design are to be checked for improper usage by the rule *LPSVM04C*.

By default, the `lp_ignore_b2b_ls` rule parameter is not set and the LPSVM04C rule reports all the level shifters connected back to back in the design. Set the `lp_ignore_b2b_ls` rule parameter to have LPSVM04A, LPSVM04B and LPSVM04C rules check on such level shifters and flag only, if they are improperly placed.

Used by	*LPSVM04C*
Options	0, 1, no, or yes
Default	`0`
Example	
Console/Tcl-based usage	`set_parameter lp_ignore_b2b_ls 1`
Usage in goal/source files	`-lp_ignore_b2b_ls=1`

lp_ignore_input_isolation_for_inout

Specifies if the *LPCONN03* rule should ignore checking of input isolation (AON to gated crossing) for inout crossings.

By default, the parameter is set to `false`. Set this parameter to `true` to enable the rule ignore checking of input isolation (AON to gated crossing) for inout crossings.

Used by	*LPCONN03*
Options	true, false
Default	`false`
Example	
Console/Tcl-based usage	`set_parameter lp_ignore_input_isolation_for_inout true`
Usage in goal/source files	`-lp_ignore_input_isolation_for_inout=true`

lp_ignore_isocell

Specifies whether the *LPSVM54* rule ignores checking that isolation cell exists in power domain.

By default, the `lp_ignore_isocell` rule parameter is not set and the LPSVM54 rule checks whether isolation cell exists in power domain.

Set the `lp_ignore_isocell` rule parameter to have the LPSVM54 rule ignore the isolation cell.

Used by	*LPSVM54*
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_ignore_isocell 1`
Usage in goal/ source files	`-lp_ignore_isocell=1`

lp_ignore_same_voltage_error

Specifies whether the *LPPLIB04*, *LPPLIB07*, *LPPLIB08* or *LPSVM49* rule should ignore checking the supplies having different name but same value for a level shifter.

You can set this parameter to ignore violations of the rule LPPLIB04, LPPLIB07, LPPLIB08 and LPSVM49 for the supplies having different name but same value. By default, the parameter is set to 0, which means name based checking is done for supplies by the associated rules.

Used by	*LPPLIB04*, *LPPLIB07*, *LPPLIB08*, *LPSVM49*
Options are	0, 1, no, or yes
Default	0

Example	
Console/Tcl-based usage	`set_parameter lp_ignore_same_voltage_error 1`
Usage in goal/ source files	`-lp_ignore_same_voltage_error=1`

lp_ignore_seqelem_retencell

Specifies whether the sequential elements or the retention cells should be ignored while checking the sneak leakage paths by the *LPSVM36* rule.

By default, the `lp_ignore_seqelem_retencell` parameter is not set and sequential elements and retention cell instances are checked.

The sequential elements and retention cells have the feedback path and hence can lead to a sneak leakage path.

Used by	*LPSVM36*
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_ignore_seqelem_retencell 1`
Usage in goal/ source files	`-lp_ignore_seqelem_retencell=1`

lp_ignore_SGDC_rules

Specifies the space-separated name list of *Constraints Checking Rules* that are to be ignored while running any/all rules from the SpyGlass Power Verify solution.

By default, all constraints checking rules are run.

To waive setup related rules-run, you can use the `ignorerules` project

file command. For example:

```
set_option ignorerules { rule-names }
set_goal_option ignorerules { rule-names }
```

Options	Space separated list of rule names
Default	none
Example	
Console/Tcl-based usage	`set_parameter lp_ignore_SGDC_rules "SGDC_lowpower_44"`
Usage in goal/source files	`-lp_ignore_SGDC_rules="SGDC_lowpower_44"`

lp_ignore_tc_commands

Specifies list of SDC constraints to be ignored by the *LP_SDC_PARSE_DEBUG* rule.

Set the value of this parameter as the name of the file that contains the list of SDC constraints to be ignored.

Used by	*LP_SDC_PARSE_DEBUG*
Options	file name
Default	null
Example	
Console/Tcl-based usage	`set_parameter lp_ignore_tc_commands ignored_commands.txt`
Usage in goal/source files	`-lp_ignore_tc_commands=ignored_commands.txt`

lp_is_common_ground

Specifies whether the level shifters have only one ground pin as checked by the *LPPLIB07* rule.

By default, the `lp_is_common_ground` is set and the LPPLIB07 rule assumes that all level shifters have only one ground pin.

Used by	*LPPLIB07*
Options	0, 1, no, or yes
Default	1
Example	
Console/Tcl-based usage	`set_parameter lp_is_common_ground 1`
Usage in goal/ source files	`-lp_is_common_ground=1`

lp_is_gate_level

Specifies the design type as RTL or gate-level netlist for the *LPSVM23* rule.

By default, the `lp_is_gate_level` rule parameter is set to 0 and the LPSVM23 rule assumes that the design is an RTL design. Set the `lp_is_gate_level` rule parameter to 1 when the design is a gate-level netlist.

Used by	*LPSVM23*
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_is_gate_level 1`
Usage in goal/ source files	`-lp_is_gate_level=1`

lp_islc

Sets whether all signal names are to be converted to lowercase during translation by the *LPSVM42* and *LPSVM43* rules.

By default, the `lp_islc` rule parameter is set to `0` and the names are

retained as they are during translation. Set the `lp_islc` rule parameter to `1` to convert the names to lowercase during translation.

Used by	*LPSVM42*, *LPSVM43*
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_islc 1`
Usage in goal/ source files	`-lp_islc=1`

lp_iso_module_name

Specifies the file name prefix for the Verilog files containing the isolation cell instantiations as generated by the *LPSVM23* rule.

By default, the `lp_iso_module_name` rule parameter is not set and the file name is iso_module_for_*<pd-name>*.v without any prefix.

Used by	*LPSVM23*
Options	<String>
Default	none
Example	
Console/Tcl-based usage	`set_parameter lp_iso_module_name "iso_island_pd1"`
Usage in goal/ source files	`-lp_iso_module_name=iso_island_pd1`

lp_isologic_in_pd

When set to `0`, the `lp_isologic_in_pd` rule parameter disables power domain rules *LPSVM08A*, *LPSVM09*, *LPSVM10*, *LPSVM48*, and *LPSVM60* to check

for the isolation logic inside a power domain (just before the crossing). In this case, the rules consider only the isolation cells outside the power domain boundary. The isolation cells that are in the same power domain that they are isolating are assumed to be inside the power domain boundary. The isolation cells that are in a voltage/power domain other than the power domain being isolated are assumed to be outside the power domain boundary.

In addition, the *LPSVM48* rule considers only the input-side isolation cells (specified with the `input_isocell` constraint) outside the power domain boundary at input of power domain.

By default, the `lp_isologic_in_pd` parameter is set to `1` and these rules consider the isolation cells both inside and outside the power domain boundary.

Used by	*LPSVM08A*, *LPSVM09*, *LPSVM10*, *LPSVM48*, *LPSVM60*
Options	0, 1, no, or yes
Default	`1`
Example	
Console/Tcl-based usage	`set_parameter lp_isologic_in_pd 0`
Usage in goal/ source files	`-lp_isologic_in_pd=0`

lp_match_location_by_domain

Specifies whether the isolation cell and level shifter cell be matched based on hierarchy or domain. This parameter is applicable only in the UPF flow.

By default, the isolation cell and level shifter cell locations are matched based on domain. To match these locations based on hierarchy, set the value of this parameter to `no`.

A hierarchy is considered to be the immediate child of a parent module.

To further understand the use of the `lp_match_location_by_domain` parameter, consider following example:

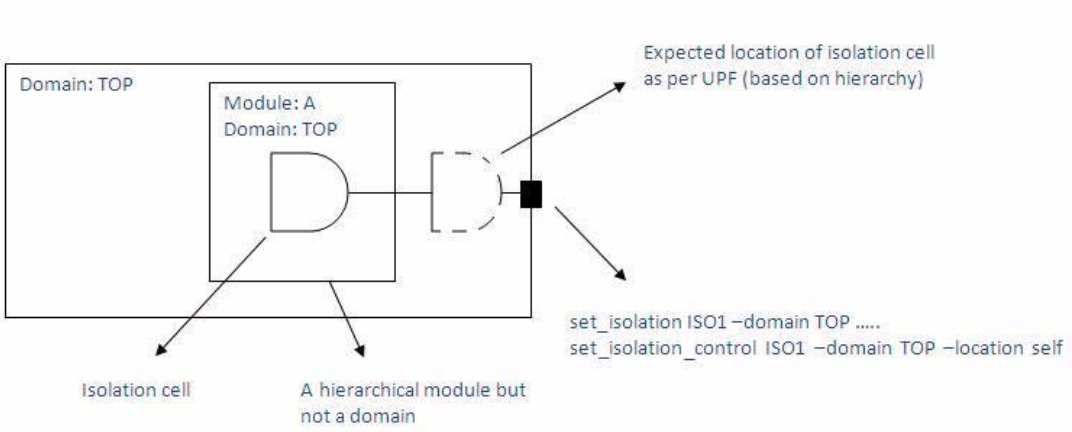


FIGURE 20. Use of the lp_match_location_by_domain parameter

- An isolation strategy (ISO1) is written at the output port of top domain (TOP).
- Isolation location specified in the isolation strategy (ISO1) is self.
- Isolation cell is not placed strictly inside the top module. The cell is placed inside a module A that was instantiated inside TOP. Module A is not another domain but just another level of hierarchy inside TOP.
- By default, this parameter is set to `yes`, and it is not an ERROR to place an isolation cell inside hierarchical module A and no violation is reported, as the location is matched on the basis of domain.
- If this parameter is set to `no`, then isolation location is matched on the basis of hierarchy and a violation is reported in this scenario.

Used by	*LPSVM08A*, *LPSVM60*, *LPSVM04D*
Options	0, 1, no, or yes
Default	`yes`
Example	

Console/Tcl-based usage	`set_parameter lp_match_location_by_domain yes`
Usage in goal/ source files	`-lp_match_location_by_domain=1`

lp_match_iso_cell_type_with_clamp_value

Set this parameter to `yes` to enable the *UPF_lowpower05* rule to report a violation if:

- An AND gate is not specified in the `map_isolation_cell` command and `clamp_value` is given as `0` or no `clamp_value` is specified in the corresponding `set_isolation` command.
- An OR cell is not specified in the `map_isolation_cell` command and `clamp_value` is given as `1` in the corresponding `set_isolation` command.

By default, the `lp_match_iso_cell_type_with_clamp_value` parameter is set to `no` and this violation is not reported.

Used by	*UPF_lowpower05*
Options	0, 1, no, or yes
Default	`no`
Example	
Console/Tcl-based usage	`set_parameter lp_match_iso_cell_type_with_clamp_value yes`
Usage in goal/ source files	`-lp_match_iso_cell_type_with_clamp_value=yes`

lp_max_blackbox_percentage

Specifies the maximum allowed ratio of blackbox elements with respect to all elements for the *LP_BLACKBOX_CHECK* rule.

By default, the this parameter is set to `100`.

Set the value of this parameter to any positive integer to report a violation

if the ratio of blackbox elements with respect to all elements in the design exceeds this value.

Used by	*LP_BLACKBOX_CHECK*
Options	<Positive integer number>
Default	100
Example	
Console/Tcl-based usage	`set_parameter lp_max_blackbox_percentage 50`
Usage in goal/ source files	`-lp_max_blackbox_percentage=50`

lp_max_pins_viol

Specifies the maximum number of pins to be reported for one net by the *LPSVM49* rule.

By default, the LPSVM49 rule reports up to ten pins for a net.

Used by	*LPSVM49*
Options	<Positive integer number>
Default	10
Example	
Console/Tcl-based usage	`set_parameter lp_max_pins_viol 20`
Usage in goal/ source files	`-lp_max_pins_viol=20`

lp_max_psw_fanout_count

Specifies the maximum number of fan-outs that a power switch can have in a daisy chain path.

Used by	*LPPSW04*
Options	<Positive integer number>

Default	1
Example	
Console/Tcl-based usage	`set_parameter lp_max_psw_fanout_count 3`
Usage in goal/ source files	`-lp_max_psw_fanout_count=3`

lp_max_retention_cell_percentage

Specifies the limit for the percentage of retention cells with respect to sequential elements.

By default, the value of the `lp_max_retention_cell_percentage` parameter is set to `50`. Set the value of this parameter to any positive integer to report a violation if the ratio of retention cells with respect to sequential elements exceeds the set value.

Used by	*LPRET01*
Options	<positive-integer>
Default	`50`
Example	
Console/Tcl-based usage	`set_parameter lp_max_retention_cell_percentage 50`
Usage in goal/ source files	`-lp_max_retention_cell_percentage=50`

lp_max_viol_count

Specifies the maximum number of total rule messages to be reported for *LPSVM38*, *LPPLIB04*, *LPPLIB05*, *LPPLIB06*, *LPPLIB07*, *LPPLIB08*, *LPPLIB10*, *LPPLIB13*, *LPPLIB14*, *LPPLIB15*, and *LPPLIB16* rules.

By default, the `lp_max_viol_count` rule parameter is set to `1000` and only the first 1000 rule messages of these rules are reported. You can set the `lp_max_viol_count` rule parameter to any positive integer number.

Used by	*LPSVM38*, *LPPLIB04*, *LPPLIB05*, *LPPLIB06*, *LPPLIB07*, *LPPLIB08*, *LPPLIB10*, *LPPLIB13*, *LPPLIB14*, *LPPLIB15*, *LPPLIB16*
Options	<Positive integer number>
Default	1000
Example	
Console/Tcl-based usage	`set_parameter lp_max_viol_count 500`
Usage in goal/ source files	`-lp_max_viol_count=500`

lp_min_number_of_bits_for_latching

Specifies the minimum number of data bits in a power hungry circuit for the circuit to be flagged for latching recommendation by the *LPXFM01* rule.

Used by	*LPXFM01*
Options	<Positive integer number>
Default	16
Example	
Console/Tcl-based usage	`set_parameter lp_min_number_of_bits_for_latching 10`
Usage in goal/ source files	`-lp_min_number_of_bits_for_latching=10`

lp_min_size_of_comparator

Specifies the minimum number of data bits in a comparator circuit for the circuit to be flagged for pre-computation recommendation by the *LPXFM06* rule.

Used by	*LPXFM06*
Options are	<Positive integer number>

Default	32
Example	
Console/Tcl-based usage	`set_parameter lp_min_size_of_comparator 25`
Usage in goal/ source files	`-lp_min_size_of_comparator=25`

lp_no_report

Specifies whether report named lp_wild_card is created that has the expansions of wildcards specified in the SpyGlass Design Constraints.

By default, the `lp_no_report` rule parameter is not set and the report is generated by the mandatory `LpWildCardMatchReport` rule.

Used by	*LpWildCardMatchReport*
Options are	0, 1, no, or yes
Default	`no`
Example	
Console/Tcl-based usage	`set_parameter lp_no_report yes`
Usage in goal/ source files	`-lp_no_report=1`

lp_output_pg_pin_as_supply_port

Use this parameter to assume the output PG Pin of the library cell as a supply port in the *UPF_lowpower02* and *UPF_lowpower09* rules.

By default this parameter is set to `no` and it does not assume the output PG Pin of the library cell as a supply port in UPF.

In the *UPF_lowower09* rule, PG pins with direction output that have pg_type constraint as `internal_power/internal_ground` (with or without `add_port_state`) or `primary_power/primary_ground` (with `add_port_state`) constraint are considered as output PG pins.

Used by	*UPF_lowpower02*, *UPF_lowpower09*
Options are	0, 1, no, or yes
Default	`no`
Example	

Console/Tcl-based usage	`set_parameter lp_output_pg_pin_as_supply_port yes`
Usage in goal/ source files	`-lp_output_pg_pin_as_supply_port=1`

lp_pg_direction_as_supply_port

Defines the directions of PG pins as valid drivers for supply nets.

By default this parameter is not set to any value.

Set the value of this parameter to `output` or `inout` or both, for the directions of the PG pins that need to be considered as valid drivers of supply nets.

For example:

```
set_parameter lp_pg_direction_as_supply_port "output,inout"
```

Here, all the PG pins whose directions are either `output` or `inout` will be considered as valid drivers of supply nets.

Used by	*UPF_lowpower02*, *UPF_lowpower09*
Options are	inout, output
Default	`none`
Example	
Console/Tcl-based usage	`set_parameter lp_pg_direction_as_supply_port output`
Usage in goal/ source files	`-lp_pg_direction_as_supply_port=output`

lp_pg_type_as_supply_port

Defines the types PG pins as valid drivers for supply nets.

By default this parameter is not set to any value.

Set the value of this parameter to `internal_power` or `internal_ground` or both, for the types of PG pins that need to be considered as valid drivers of supply net.

For example:

```
set_parameter lp_pg_type_as_supply_port
"internal_power,internal_ground"
```

Here, all the PG pin whose direction is internal and type is either `internal_power` or `internal_ground` will be considered as valid drivers of supply nets.

Used by	*UPF_lowpower09*
Options are	internal_power, internal_ground
Default	`none`
Example	
Console/Tcl-based usage	`set_parameter lp_pg_type_as_supply_port internal_ground`
Usage in goal/ source files	`-lp_pg_type_as_supply_port=internal_ground`

lp_pst_merge_no_caching

Specifies whether the *UPF_lowpower08* rule should use the PST merging flow without caching of merged PSTs.

By default the parameter is set to `no` and caching of PST is used.

Set this parameter to `yes` to use the PST merging flow without caching of the merged PSTs.

NOTE: *USe this parameter in case if the runtime of the LP_CROSSING_DATA rule is very long.*

Used by	*UPF_lowpower08*
Options are	0, 1, no, or yes
Default	`no`
Example	
Console/Tcl-based usage	`set_parameter lp_pst_merge_no_caching yes`
Usage in goal/ source files	`-lp_pst_merge_no_caching=yes`

lp_pst_merge_new

Specifies whether to use the new PST merging flow by the *UPF_lowpower08* rule.

By default this parameter is set to `yes` and it uses the enhanced PST merging flow. This optimizes merging while preserving all the information needed to correctly evaluate all domain crossings for isolation, level-shifting and other such considerations.

Set the value of this parameter to `no` to use the old PST merging flow.

NOTE: *Using the old PST merging flow may increase the runtime and memory usage.*

Used by	*UPF_lowpower08*
Options are	0, 1, no, or yes
Default	`yes`
Example	
Console/Tcl-based usage	`set_parameter lp_pst_merge_new no`
Usage in goal/ source files	`-lp_pst_merge_new=no`

lp_read_powerdata_at_setup

Specifies whether UPF commands should be parsed at the setup stage of SpyGlass or after completing setup.

By default this parameter is set to `0` and UPF command parsing takes place after completing the setup stage. This invokes the *LP_POWERDATA_INFO* rule.

Set the `lp_read_powerdata_at_setup` parameter to `1` to enable the UPF command parsing at the setup stage. This invokes the *LP_POWERDATA_READ* instead.

Used by	*LP_POWERDATA_READ*
Options are	0, 1, no, or yes
Default	`0`

Example	
Console/Tcl-based usage	`set_parameter lp_read_powerdata_at_setup 1`
Usage in goal/ source files	`-lp_read_powerdata_at_setup=1`

lp_relative_aon_checking

Use this parameter to turn on the relatively always-on checking.

Used by	*LP_SPECIAL_PIN_CONNECTION*, *LPSVM53*
Options are	0, 1, no, or yes
Default	`no`
Example	
Console/Tcl-based usage	`set_parameter lp_relative_aon_checking yes`
Usage in goal/ source files	`-lp_relative_aon_checking=1`

lp_report_all_wildcards

For the `LpWildCardMatchReport` rule, the `lp_report_all_wildcards` rule parameter enables the report generation of matches found for the `-instname`, `-portname`, and `-netname` arguments with wildcards of the *voltage_domain* constraint. By default, the rule parameter is set to `0` and the report does not contain matches found for the `-instname`, `-portname`, and `-netname` arguments.

For the `SGDC_lowpower75` rule, the `lp_report_all_wildcards` rule parameter enables the violation message for each instance/module matched with the `-instance`/`-module` argument of the *pin_voltage* constraint for which pin specified with the `-names` argument does not exist. By default, the instance/module name with a wildcard is given in the

violation message.

Used by	*LpWildCardMatchReport*, *SGDC_lowpower75*
Options	0, 1, no, or yes
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_report_all_wildcards yes`
Usage in goal/ source files	`-lp_report_all_wildcards=1`

lp_report_domain_crossing_on_iso_signal

Specifies if the *LPSVM08A* rule should check the isolation signals in the SGDC flow.

By default, this parameter is set to `0`. Set the value of the `lp_report_domain_crossing_on_iso_signal` parameter to `1` to enable the *LPSVM08A* rule to check isolation signals in the SGDC flow.

Used by	*LPSVM08A*
Options are	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_report_domain_crossing_on_iso_signal 1`
Usage in goal/ source files	`-lp_report_domain_crossing_on_iso_signal=1`

lp_report_upf_command_in_source_file

Specifies if the rule *UPFSEM_40* should report the UPF command used in sourced TCL file.

By default, this parameter is set to `no`. Set the value of the `lp_report_upf_command_in_source_file` parameter to `yes` to report the UPF command used in sourced TCL file.

Used by	*UPFSEM_40*
Options are	0, 1, no, or yes
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_report_upf_command_in_source_file yes`
Usage in goal/ source files	`-lp_report_upf_command_in_source_file=yes`

lp_report_missing_ack_port_option

Specifies if the *UPFSEM_41* rule should report the `ack_port` argument in the `create_power_switch` command.

By default, this parameter is set to `no`. Set the value of the `lp_report_missing_ack_port_option` parameter to `yes` to report the `ack_port` argument in the `create_power_switch` command.

Used by	*UPFSEM_41*
Options are	0, 1, no, or yes
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_report_missing_ack_port_option yes`
Usage in goal/ source files	`-lp_report_missing_ack_port_option=yes`

lp_report_drc_blockgroup_for_missing_csn

Specifies if the *UPF_lowpower15* rule should report the *drc_blockgroup* section in the *lp_multi_supply_instance* report.

By default, this parameter is set to `yes`. Set the value of the `lp_report_drc_blockgroup_for_missing_csn` parameter to `no` to not include the *drc_blockgroup* section in the generated `lp_multi_supply_instance.rpt` report.

Used by	*UPF_lowpower15*
Options are	0, 1, no, or yes
Default	yes
Example	
Console/Tcl-based usage	`set_parameter lp_report_drc_blockgroup_for_missing_csn no`
Usage in goal/ source files	`-lp_report_drc_blockgroup_for_missing_csn=no`

lp_report_port_state_on_non_supply_port

Specifies whether the *UPFSEM_29* should report the *add_port_state* command specification on non-supply ports that are not already created with the *create_supply_port* command.

By default, the `lp_report_port_state_on_non_supply_port` parameter is set to `no`.

Set this parameter to `yes` to report the specification of the `add_port_state` command on non-supply ports that are not already created with the `create_supply_port` command.

Used by	*UPFSEM_29*
Options are	0, 1, no, or yes
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_report_port_state_on_non_supply_port yes`
Usage in goal/ source files	`-lp_report_port_state_on_non_supply_port=yes`

lp_set_abstract_power_model_name

Specifies the desired name for the power model.

By default, the power model name is set to `upf_<module-name>`. Set the value of this parameter to desired power model name generated by the *PV_Abstract01* rule.

Used by	*PV_Abstract01*
Options are	any string
Default	`null`
Example	
Console/Tcl-based usage	`set_parameter lp_set_abstract_power_model_name upf_pm1`
Usage in goal/ source files	`-lp_set_abstract_power_model_name=upf_pm1`

lp_set_design_stage

Specifies the desired design stage for running the rule.

By default, the parameter is set to `rtl`. Set the value of the `lp_set_design_stage` parameter to `netlist` or `pg_netlist` to specify the desired design stage.

Used by	All rules in the SpyGlass Power Verify solution
Options are	rtl, netlist, pg_netlist
Default	`rtl`
Example	
Console/Tcl-based usage	`set_parameter lp_set_design_stage netlist`
Usage in goal/ source files	`-lp_set_design_stage=netlist`

lp_set_sim_val_x

Enables the *LPSVM09*, *LPSVM10*, *LPSVM28*, *LPSVM31*, *LPSVM47*, *Connection Rules*, and *LPPLIB17* rules (that use LE for simulation) to propagate the value for set_case_analysis and other control signals (such as isolation signal, save/restore signal) for the logic that lies in the power domain.

By default, the parameter is set to `1` and the value for `set_case_analysis` and other control signals will not be propagated, if the combination logic lies in power domain.

Used by	*LPSVM09*, *LPSVM10*, *LPSVM28*, *LPSVM31*, *LPSVM47*, *Connection Rules*, and *LPPLIB17*
Options are	0, 1, no, or yes
Default	`1`
Example	
Console/Tcl-based usage	`set_parameter lp_set_sim_val_x 0`
Usage in goal/ source files	`-lp_set_sim_val_x=1`

lp_single_supply

Causes the *LPPLIB04* rule to check that the supply pin of single supply level shifter is connected to the output side supply. By default, the parameter is set to `high` and the rule checks the supply pin of single supply level shifter is connected to the higher side supply connected to input or output side.

Used by	*LPPLIB04*
Options	`output`, `high`
Default	`high`
Example	
Console/Tcl-based usage	`set_parameter lp_single_supply output`
Usage in goal/ source files	`-lp_single_supply=output`

lp_set_upf_attributes

Specifies if SpyGlass should display additional UPF attributes in the Debug

Data Window section of the schematic.

By Default, this parameter is set to `1` and SpyGlass displays additional UPF attributes in the Debug Data Window section of the schematic.

Set this parameter to `0` if you do not want the additional UPF attributes displayed in the Debug Data Window section of the schematic.

Used by	All rules in the SpyGlass Power Verify solution
Options are	0, 1, no, or yes
Default	`1`
Example	
Console/Tcl-based usage	`set_parameter lp_set_upf_attributes 0`
Usage in goal/ source files	`-lp_set_upf_attributes=0`

lp_sig_viol_count

Specifies the maximum rule message count for each violating signal as reported by the *LPSVM40* and *LPPLIB11* rules.

By default, the `lp_sig_viol_count` rule parameter is set to 50 and these rules flag only the first 50 rule messages for each signal. You can set the `lp_sig_viol_count` parameter to any positive integer number.

Used by	*LPSVM40*, *LPPLIB11*
Options are	<Positive integer number>
Default	`50`
Example	
Console/Tcl-based usage	`set_parameter lp_sid_viol_count 100`
Usage in goal/ source files	`-lp_sid_viol_count=100`

lp_skip_aon_buf

Specifies whether SpyGlass skips always-on/relatively-on buffers to find an isolation crossing.

By default, this parameter is set to `1` and SpyGlass skips always-on/ relatively-on buffers, in the following scenarios:

- When always-on buffer form the library is in the path
- Normal buffer is present in the always-on domain
- Normal buffer is present in the switchable domain but is more on or equally on with respect to its receiver.

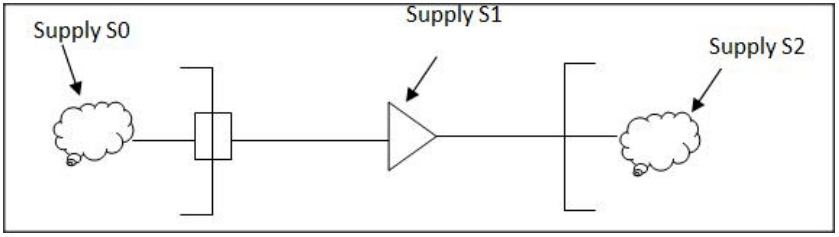


FIGURE 21. Buffers

In the above figure, the supply S1 is more or equally on that the supply s2.

Set the value of the `lp_skip_aon_buf` parameters to `0` to enable SpyGlass to consider always-on buffers for the above cases.

Used by	All rules in the SpyGlass Power Verify solution
Options	0, 1, no, or yes
Default	`1`
Example	
Console/Tcl-based usage	`set_parameter lp_skip_aon_buf 0`
Usage in goal/ source files	`-lp_skip_aon_buf=0`

lp_skip_blackbox_checking

Specifies whether the rule checking for black boxes should be skipped.

By default this parameter is set to 0 and rule checking is performed for black boxes.

Set this parameter to 1 to skip checking for black boxes. Modules that are specified with stop project file command will also be considered as black-boxes for following rules.

Used by	*LPSVM04A*, *LPSVM04C*, *LPSVM04D*, *LPSVM04E*, *LPSVM08A*, *LPSVM08B*, *LPSVM08C*, *LPISO03A* , *LPISO03B*, *LPLSH03*, *LPLSH04*, *LPLSH05A*, *LPLSH05B*, *LPLSH06*, *LPSVM22*, *LPISO04A*, *LPISO04B*, *LPISO04C*, *LPISO05A*, *LPISO05B*, *LPSVM60*, *UPF_lowpower21*
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_skip_blackbox_checking 1`
Usage in goal/ source files	`-lp_skip_blackbox_checking=1`

lp_skip_buf

Specifies whether the SpyGlass-generated buffers are considered during rule checking by the *LPCONN04A*, *LPCONN04E*, *LPISO01*, *LPISO02*, *LPSVM08A*, *LPSVM08B*, *LPSVM08C*, *LPISO04A*, *LPISO04C*, *LPISO06A*, *LPPLIB04*, *LPLSH03*, *LPLSH04*, *LPPSW01*, *LPPSW02*, *LPSVM04A*, *LPSVM04B*, *LPSVM04C*, *LPSVM04D*, *LPSVM04E*, *LPSVM09*, *LPSVM10*, *LPSVM12A*, *LPSVM22*, *LPSVM24*, *LPSVM26*, *LPSVM47*, *LPSVM54*, *LPSVM60*, *LPTIE01*, *LPCONN06*, *LPLSH07*, *LP_SPECIAL_PIN_CONNECTION*, and *LPTIE02* rules.

By default, this parameter is set to 1 and above rules skip such buffers. Set this parameter to 0 to consider SpyGlass-generated buffers during rule checking.

Used by	*LPCONN04A*, *LPISO01*, *LPISO02*, *LPSVM08A*, *LPSVM08B*, *LPSVM08C*, *LPISO04A*, *LPISO04C*, *LPISO06A*, *LPPLIB04*, *LPLSH03*, *LPLSH04*, *LPPSW01*, *LPPSW02*, *LPSVM04A*, *LPSVM04B*, *LPSVM04C*, *LPSVM04E*, *LPSVM09*, *LPSVM10*, *LPSVM12A*, *LPSVM22*, *LPSVM24*, *LPSVM26*, *LPSVM47*, *LPSVM54*, *LPSVM60*, *LPTIE01*, *LPTIE02*, *LPCONN06*, *LPLSH07*, *LP_SPECIAL_PIN_CONNECTION*
Options	0, 1, no, or yes
Default	`1`
Example	
Console/Tcl-based usage	`set_parameter lp_skip_buf 0`
Usage in goal/ source files	`-lp_skip_buf=0`

lp_skip_existence_check_for_resolve_parallel_net

Specifies whether the *checkUPF_existence* rule skips the existence check for supply nets created with `-resolve parallel` UPF strategy.

By default this parameter is set to `no` and the *checkUPF_existence* reports violation for the supply nets created with `-resolve parallel` UPF strategy.

Set the value of this parameter to `yes` to skip the existence check for such supply nets.

Used by	*checkUPF_existence*
Options	0, 1, no, or yes
Default	`no`
Example	
Console/Tcl-based usage	`set_parameter lp_skip_existence_check_for_resolve_parallel_net yes`
Usage in goal/ source files	`-lp_skip_existence_check_for_resolve_parallel_net=yes`

lp_skip_combo_cell_for_reference_toplevel_isolation_signal

Specifies whether the *LPSVM22* and *LPISO01* rules should skip combinational cells whose output pin is related to signal input pin.

By default this parameter is set to `0`.

Set this parameter to `1` to skip combinational cells whose output pin is related to signal input pin to get correct source of control signal. This parameter is used only to identify the correct reference top-level isolation signal.

Used by	*LPSVM22*, *LPISO01*
Options	0, 1, no, or yes
Default	`0`
Example	
Console/Tcl-based usage	`set_parameter lp_skip_combo_cell_for_reference_toplevel_isolation_signal 1`
Usage in goal/ source files	`- lp_skip_combo_cell_for_reference_toplevel_isolation_signal=1`

lp_skip_existence_check_for_virtual_supply_net_without_srsn

Specifies whether the *checkUPF_existence* rule should perform existence checking for virtual supplies that are not used in the `set_related_supply_net` command.

By default, this parameter is not set and existence checking is only skipped for those virtual supplies that are used in the `set_related_supply_net` command.

Set this parameter to `yes` to skip existence check for virtual supplies that are not used in the `set_related_supply_net` command.

Used by	*checkUPF_existence*
Options	0, 1, no, or yes
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_skip_existence_check_for_virtual_supply_net_without_srsn yes`
Usage in goal/ source files	`-lp_skip_existence_check_for_virtual_supply_net_without_srsn=yes`

lp_skip_feedthrough_buffer

Specifies whether the instances of feedthrough buffers are skipped by the *LPSVM08B* and *LPISO04A* rules.

A buffer whose supply is more on or equally on with respect to the supply of destination node is considered as feedthrough buffer.

By default this parameter is set to 0 and these rules consider such instances of feedthrough buffers.

Used by	*LPSVM08B*, *LPISO04A*
Options	0, 1, no, or yes
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_skip_feedthrough_buffer 1`
Usage in goal/ source files	`-lp_skip_feedthrough_buffer=1`

lp_skip_iso_check_on_ground

Specifies whether to report isolation checks on ground nets.

By default, the `lp_skip_iso_check_on_ground` parameter is set to `no` and it flags missing isolation strategy and missing isolation cells on the

ground nets.

Set this parameter to `yes` to not report missing isolation strategy and missing isolation cells on the ground nets.

Used by	*LPISO04A*, *LPISO05A*, *LPISO05B*, *LPSVM08B*, *LPSVM08C*, *LPISO03A* , *LPISO03B*, *LPSVM60*
Options	0, 1, no, or yes
Default	`no`
Example	
Console/Tcl-based usage	`set_parameter lp_skip_iso_check_on_ground yes`
Usage in goal/ source files	`-lp_skip_iso_check_on_ground=yes`

lp_skip_lib_buf

Specifies whether the instances of library buffers are skipped, while checking for crossings, by the *LPSVM04A*, *LPSVM04B* and *LPSVM08A* rules.

By default this parameter is set to `0` and these rules consider such instances of library buffers.

Used by	*LPSVM04A*, *LPSVM04B*, *LPSVM08A*, *LPISO04A*, *LPISO04B*, *LPISO04C*, *LPISO05A*, *LPISO05B*
Options	0, 1, no, or yes
Default	`0`
Example	
Console/Tcl-based usage	`set_parameter lp_skip_lib_buf 1`
Usage in goal/ source files	`-lp_skip_lib_buf=1`

lp_skip_related_pgpin_checking_for_analog_pin

Specifies whether the *LPLIB_check02* rule skips checking for analog pins.

By default, this parameter is set as `yes` and the *LPLIB_check02* does not check related_power_pin/related_ground_pin for analog pins.

Set this parameter as `no` to perform checking for analog pins.

Used by	*LPLIB_check02*
Options	0, 1, no, or yes
Default	`yes`
Example	
Console/Tcl-based usage	`set_parameter lp_skip_related_pgpin_checking_for_analog_pin no`
Usage in goal/ source files	`-lp_skip_related_pgpin_checking_for_analog_pin=no`

lp_skip_same_src_supply_buf

Specifies if the *LPSVM04A*, *LPSVM08A*, *LPSVM22*, *LPISO05A* rules skip buffers and inverters between strategy node and isolation cell which are of same supply as source but are LESS ON w.r.t. destination. This parameter will impact the creation of crossings.

By default this parameter is set to `no`. Set this parameter to yes to enable these rules skip such buffers and inverters.

Used by	*LPSVM04A*, *LPSVM08A*, *LPSVM22*, *LPISO05A*
Options	no, or yes
Default	`no`
Example	
Console/Tcl-based usage	`set_parameter lp_skip_same_src_supply_buf yes`
Usage in goal/ source files	`-lp_skip_same_src_supply_buf=yes`

lp_skip_pwr_gnd

Specifies whether the *LPSVM08A*, *LPSVM08B*, *LPSVM08C*, *LPISO04A*, *LPISO04C*, *LPISO06A*, *LPLSH03*, *LPLSH04*, *LPSVM04A*, *LPSVM04B*, *LPSVM04C*, *LPSVM04E*, *LPSVM04E*, *LPSVM09*, *LPSVM10*, *LPSVM22*, *LPSVM47*, *LPSVM60*, and *LP_MULTI_DOMAIN_CROSSING_CHECK* rules check nets connected to power/ground supply.

By default, the `lp_skip_pwr_gnd` rule parameter is set and such nets are ignored.

Used by	*LPSVM08A*, *LPSVM08B*, *LPSVM08C*, *LPISO04A*, *LPISO04C*, *LPISO06A*, *LPLSH03*, *LPLSH04*, *LPSVM04A*, *LPSVM04B*, *LPSVM04C*, *LPSVM04E*, *LPSVM04E*, *LPSVM09*, *LPSVM10*, *LPSVM22*, *LPSVM47*, *LPSVM60*, *LP_MULTI_DOMAIN_CROSSING_CHECK*
Options	0, 1, no, or yes
Default	`1`
Example	
Console/Tcl-based usage	`set_parameter lp_skip_pwr_gnd 0`
Usage in goal/source files	`-lp_skip_pwr_gnd=0`

lp_support_highconn_lowconn

Specifies whether the level-shifter/isolation strategy should be applied on the `highconn`/`lowconn` ports.

By default this parameter is set to `no` and level-shifter/isolation strategy is not applied on the `highconn`/`lowconn` ports. Set this parameter to `yes` to apply such strategy.

Used by	All rules in the SpyGlass Power Verify solution in the UPF flow
Options	0, 1, no, or yes
Default	`no`

Example	
Console/Tcl-based usage	`set_parameter lp_support_highconn_lowconn yes`
Usage in goal/ source files	`-lp_support_highconn_lowconn=yes`

lp_support_ldo_net

Specifies whether the SpyGlass Power Verify solution should support low dropout (LDO) voltage regulator nets.

By default, this parameter is set to `no` and the LDO nets are not supported.

Suppose, a supply net VDD is created in UPF and the `add_port_state` command is written on this supply net, but no `create_supply_port` command is written for this port state.

In this case, the SpyGlass Power Verify solution creates an implicit port VDD. The supply net VDD is connected to another scope's supply ports, but implicitly created port is not connected to any supply net.

This supply net is mentioned in scope level PST. Supply port gets preference over supply net (both having the same name) while inferring PST. So, implicitly created supply port gets preference in PST and due to the lack of connectivity of this port, an error in PST merging is observed. The supply nets on which port state can be written are known as LDO nets. Set the `lp_support_ldo_net` parameter to `yes` to support such LDO nets.

For example:

```
create_supply_net VDD
add_port_state VDD –state {on1 .8}
connect_supply_net VDD –ports {inst1/VDD inst2/VDD}
```

In this example, the supply net VDD is an LDO net as there is no `create_supply_port VDD` command mentioned.

After setting the `lp_support_ldo_net` parameter to `yes`, the supply net VDD is inferred correctly in the PST.

Used by	All rules in the SpyGlass Power Verify solution
Options	0, 1, no, or yes
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_support_ldo_net yes`
Usage in goal/ source files	`-lp_support_ldo_net=yes`

lp_support_unified_naming_style

Specifies whether to support SDC style hierarchical names for objects inside generate block in UPF using `define_name_rules/ change_name` SDC command, in the SpyGlass Power Verify (PV) flow.

By default this parameter is set to `no`, so if you specify SDC style hierarchical names for objects inside generate block in UPF using `define_name_rules/change_name` SDC command, it will not be supported.

Set this parameter to `yes` to support SDC style hierarchical names for objects inside generate block in UPF using `define_name_rules/ change_name` SDC command. An example is given below.

Consider the flowing Verilog snippet:

```
module TOP();
test1 inst1();
endmodule
module test1();

generate
    for (i=0;i<3;++i)
       begin:
         test3 inst1();
       end
endgenerate
endmodule
module test3 (input in, output out);
```

```
AND and1(); ? Lib cell
endmodule
```

Consider the flowing UPF snippet:

```
create_supply_net VDD3
.....
connect_supply_net VDD3 -ports {inst1/genblk1[0]/inst1/and1/
VDD}
```

Consider the flowing SDC snippet:

```
define_name_rules verilog_out2 -allowed "a-zA-Z0-9_\[\]" -
replacement_char "/"
change_names -rules verilog_out2 -hierarchy
```

In the above example, the hierarchical name (up to AND cell) generated/ supported in SpyGlass is `inst1/\genblk1[0].inst/and1` (Note the dot after genblk[0].inst).

However the name specified in UPF is `inst1/genblk[0]/inst/and1`.

To support this name (specified in the UPF) in SpyGlass PV flow, you need to specify the `define_name_rules` and `change_names` commands in the SDC. When this SDC file is passed to SpyGlass, SpyGlass is able to find the object (`inst1/genblk[0]/inst/and1`) in the design. However this support is not present by default. Set the `lp_support_unified_naming_style` parameter to `yes` to enable SpyGlass to recognize this name.

Currently, the following UPF commands are supported:

- create_power_domain
- connect_supply_net
- set_related_supply_net
- set_port_attributes
- set_retention
- create_power_switch
- set_pin_related_supply
- set_scope

- load_upf
- set_isolation
- set_level_shifter

Used by	All rules in the SpyGlass Power Verify solution in the UPF flow
Options	0, 1, no, or yes
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_support_unified_naming_style yes`
Usage in goal/ source files	`-lp_support_unified_naming_style=yes`

lp_switch_selective_out

By default, the *LPXFM01* rule flags a power-hungry circuit if its output is being used selectively and its inputs are not latched. Set the `lp_switch_selective_out` rule parameter to flag all power-hungry circuits if their inputs are not latched irrespective of the usage type of their outputs.

Used by	*LPXFM01*
Options	0, 1, no, or yes
Default	1
Example	
Console/Tcl-based usage	`set_parameter lp_switch_selective_out 0`
Usage in goal/ source files	`-lp_switch_selective_out=0`

lp_treat_equivalent_psw_output_supplies_different

Specifies if the *UPF_lowpower08* rule consider output supply of power switches with same control signal and same input as equivalent.

By default, the parameter is set to `no`. Set this parameter to `yes` to enable the UPF_lowpower08 rule to not consider output supply of power switches with same control signal and same input as equivalent.

Used by	*UPF_lowpower08*
Options	0, 1, yes, no
Default	`0`
Example	
Console/Tcl-based usage	`set_parameter lp_treat_equivalent_psw_output_supplies_different 1`
Usage in goal/ source files	`-lp_treat_equivalent_psw_output_supplies_different =1`

lp_use_equivalence_in_pst

Specifies that the *UPF_lowpower08* rule considers equivalent nets as connected during PST merging.

By default the parameter is set to no and the *UPF_lowpower08* rule does not use the equivalence information from the *set_equivalent* command in the PST.

Set this parameter to yes to use the equivalence information from the *set_equivalent* command in the PST.

Since the merged PST will be different when this parameter is enabled, violation reporting for the isolation and level shifter rules would also be impacted. Some violations for missing strategies may not get reported anymore or you may see new violations for redundant strategies.

Used by	*UPF_lowpower08*
Options	yes, no
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_use_equivalence_in_pst yes`
Usage in goal/ source files	`-lp_use_equivalence_in_pst=yes`

lp_use_inferred_clocks

Specifies that *LPCONN05* rules use the auto-generated clock information in addition to any user-defined clocks, which are specified using the *clock* constraint in the SGDC file. However, this is not a recommended methodology.

It is recommended to specify primary clock and reset sources in SGDC file to have better QoR and less noise.

Used by	*LPCONN05*
Options	yes, no
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_use_inferred_clocks yes`
Usage in goal/ source files	`-lp_use_inferred_clocks=yes`

lp_use_inferred_resets

Specifies that *LPCONN05* rules use the auto-generated reset information in addition to any user-defined reset(s), which are specified using the *reset*

constraint in the SGDC file. However, this is not a recommended methodology.

It is recommended to specify primary clock and reset sources in SGDC file to have better QoR and less noise.

Used by	*LPCONN05*
Options	yes, no
Default	no
Example	
Console/Tcl-based usage	`set_parameter lp_use_inferred_resets yes`
Usage in goal/ source files	`-lp_use_inferred_resets=yes`

lp_use_voltage_map_value

Specifies whether the voltage value provided through the `voltage_map` library pin attribute should be considered or not.

By default, this parameter is set to `0`. If you set the value of the `lp_use_voltage_map_value` parameter is set to `1`, the voltage value provided for the `voltage_map` attribute of source pin of a crossing is considered while calculating the level shifter requirement.

Used by	*LPLSH03*, *LPLSH04*, *LPLSH05A*, *LPLSH05B*, *LPLSH06*, *LPLSH07*, *LPSVM04A*, *LPSVM04B*
Options	0, 1, yes, no
Default	0
Example	
Console/Tcl-based usage	`set_parameter lp_use_voltage_map_value 1`
Usage in goal/ source files	`-lp_use_voltage_map_value=1`

lp_vcdendtime

Specifies the end time (in the same time unit as in the VCD file) for the VCD file analysis for the *LPSVM42* and *LPSVM43* rules.

By default, `lp_vcdendtime` is set to a negative number (`-1`) that indicates that the end time as specified in vcd file should be used.

Used by	*LPSVM42*, *LPSVM43*
Options	<Positive integer number>
Default	`-1`
Example	
Console/Tcl-based usage	`set_parameter lp_vcdendtime 50000`
Usage in goal/ source files	`-lp_vcdendtime=50000`

lp_vcdminclk

Specifies the minimum time period of the clock to be used for calculation of signals in the VCD to SpyGlass Design Constraints translation by the *LPSVM42* and *LPSVM43* rules.

By default, the `lp_vcdminclk` rule parameter is set to `-1` and these rules use the VCD clock with minimum time period for reference. You can set the value of the `lp_vcdminclk` rule parameter to any positive integer number. Then, these rules consider only those clocks with time period more than the specified value.

Used by	*LPSVM42*, *LPSVM43*
Options	<Positive integer number>
Default	`-1`
Example	
Console/Tcl-based usage	`set_parameter lp_vcdminclk 8`
Usage in goal/ source files	`-lp_vcdminclk=8`

lp_vcdstarttime

Specifies the start time (in the same time unit as in the VCD file) for the vcd file analysis for the *LPSVM42* and *LPSVM43* rules.

By default, `lp_vcdstarttime` is set to a negative number (-1) that indicates that the start time as specified in VCD file should be used.

Used by	*LPSVM42*, *LPSVM43*
Options	<Positive integer number>
Default	`-1`
Example	
Console/Tcl-based usage	`set_parameter lp_vcdstarttime 1000`
Usage in goal/ source files	`-lp_vcdstarttime=1000`

lp_vcdtopname

Specifies the hierarchical instance name of the top module in the VCD (testbench) file specified using the *lp_genvcdfile* rule parameter for the *LPSVM42* and *LPSVM43* rules.

The mandatory LPCheckVCD rule checks for existence of the specified instance in the VCD file. If the specified instance is not found, the VCD file is ignored and the corresponding rules are not run.

Used by	*LPSVM42*, *LPSVM43*
Options	<String>
Default	`none`
Example	

Console-based usage	`set_parameter lp_vcdtopname "testbench.U1"`
Usage in goal/ source files	`-lp_vcdtopname="testbench.U1"`

lp_wildcard_report_count

Specifies the maximum number of desired matched entries to be used for the `-instname` and `-netname` arguments in the report generated by *LpWildCardMatchReport* rule.

By default, the `lp_wildcard_report_count` parameter is set to 30.

Used by	*LpWildCardMatchReport*
Options	<Positive integer number>
Default	`30`
Example	
Console/Tcl-based usage	`set_parameter lp_wildcard_report_count 20`
Usage in goal/ source files	`-lp_wildcard_report_count=20`

lp_write_sgdc

By default, SpyGlass generates a SpyGlass Design Constraints file named auto_activity.sgdc that contains activity constraints based on the VCD file (specified using the *lp_genvcdfile* parameter). Thus, you can use the generated SpyGlass Design Constraints file in place of the VCD file in the subsequent runs.

Unset the `lp_write_sgdc` parameter to skip generating the file.

Used by	*LPSVM42*, *LPSVM43*
Options are	<Positive integer number>
Default	1
Example	
Console/Tcl-based usage	`set_parameter lp_write_sgdc 0`
Usage in goal/ source files	`-lp_write_sgdc=0`

SpyGlass Power Verify Reports

The SpyGlass Power Verify solution generates rule-specific reports that can be opened from the **Reports** menu in Atrenta Console.

The following SpyGlass Power Verify solution reports are available:

Report Name	Description
lp_assertion_info	This report contains information on the OVL assertions.
lp_autofix_info	This report contains information on auto insertion of level shifters.
lp_constr_info	This report contains information on voltage domain specific design information provided using SGDC/UPF/CPF files.
lp_cons_req	This report lists missing constraints that are required by the rules.
lp_crossing_data	This report contains domain crossing data information as inferred by SpyGlass using the SGDC/CPF/UPF files and library files.
lp_domain_info	This report contains domain related information for the domain crossings.
lp_lib_data	This report contains information from the library files as interpreted by the SpyGlass Power Verify solution.
lp_multivt_perblock	This report contains the instantiation information about VT type library cells in each hierarchy.
lp_multivtreport	This report contains the instantiation information about VT type library cells in the complete design.
lp_multi_domain_crossing_check	This report contains information of all the signals that are going to more than two domains.
lp_nested_domain_info	This report contains information related to nested domains for each domain.
lp_power_data	This report contains information about user-specified CPF/UPF commands parsed by SpyGlass.
lp_power_state	This report shows the power state information specified in the power intent SGDC/CPF/UPF files.
lp_psw_info	This report contains the list of valid power switch strategies defined in the UPF file.

Report Name	Description
lp_ret_info	This report contains the list of valid retention strategies defined in the UPF file.
lp_retention_cell_list	This report lists the retention instances found in the design.
lp_srsn_info	This report lists the ports on which correct supply information using the set_related_supply_net command is specified
lp_supply_connection	This report displays the instance path name, followed by the power pin and corresponding supply net connection.
lp_vd_info	This report provides level-shift, isolation and power-switch information for domains in the design.
lp_wild_card	This report contains the matches found for the cell names specified in the SGDC constraints with wildcards (* or ?).
LP-report	This report is generated when you specify the `set_option report LP-report` command in the project file.
lp_strategy_info	This report contains the list of valid isolation and/or level shifter strategies defined in the UPF file. In addition, this report contains the expanded list of elements to which that strategy is applied.
lp_multi_supply_instance	This report lists the multi-supply instances that do not have an associated connect_supply_net command.
lp_special_pin_connection	The report shows the primary port connection to special pins.

lp_assertion_info

Report Generated by the LPSVM58 Rule

The *LPSVM58* generates a report named lp_assertion_info containing OVL assertions.

A sample report is as follows:

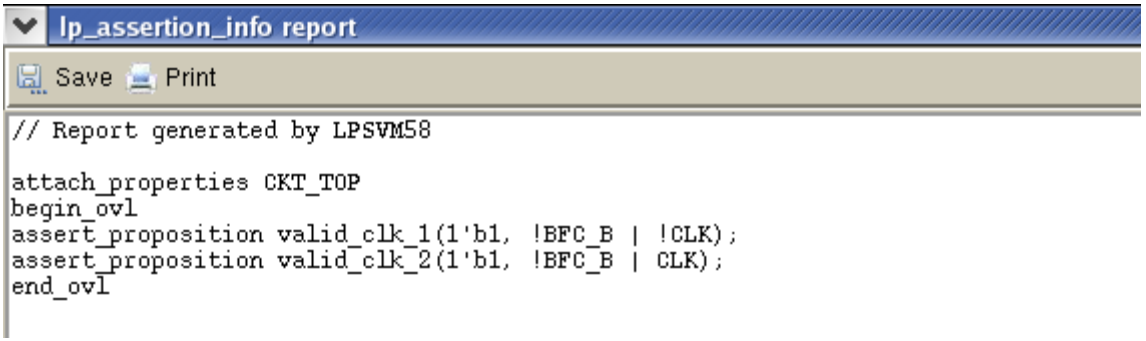


FIGURE 22. Sample lp_assertion_info report

lp_autofix_info

The *LPSVM30* rule generates a report named lp_autofix_info that contains information about auto insertion of level shifters.

A sample report is as follows:

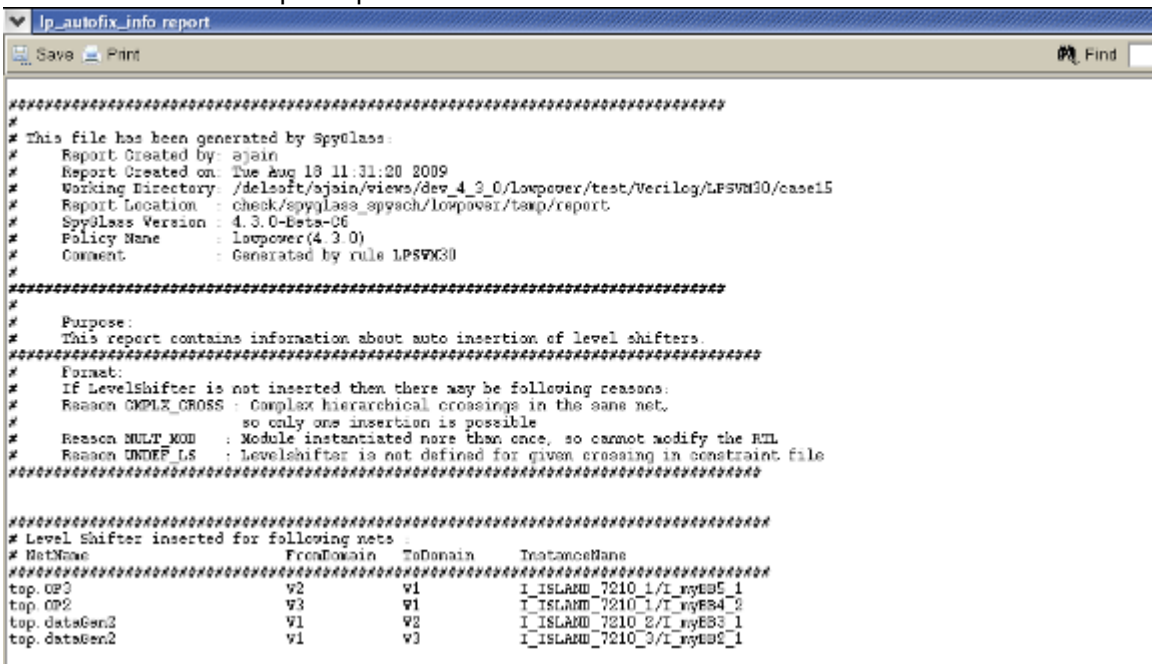


FIGURE 23. Sample lp_autofix_info report

The report shows the following details:

- List of nets for which no level shifters could be inserted along with domain names and the reason. If level shifter is not inserted then any of the following reasons is reported:
  - Reason (CMPLX_CROSS): Complex hierarchical crossings in the same net, so only one insertion is possible
  - Reason (MULT_MOD): Module instantiated more than once, so cannot modify the RTL
  - Reason (UNDEF_LS): Level shifter is not defined for given crossing in the constraint file
- List of nets for which level shifter instances are inserted along with domain names, level shifter design unit and instance names, and whether enable pin is available.

lp_constr_info

Report Generated by the LP_DECOMPILE_CONSTR Rule

The *LP_DECOMPILE_CONSTR* rule generates a report named lp_constr_info that contains voltage domain specific design information provided using SGDC/UPF/CPF files. This report also contains the information related to special cells like Level Shifter Cells, Isolation Cells, Retention Cells and Always-On Buffers provided using CPF/UPF/SGDC power format files or inferred from the library information.

NOTE: *When you specify the voltage domain design units with wildcards, their information is not provided in the* lp_constr_info *report.*

The lp_constr_info report contains the following sections:

- Information on voltage domain in the design:

  The voltage domain information is provided using the `voltage_domain` SGDC command or the `create_power_domain` CPF/UPF command. This section also contains information such as domain value, whether domain is an always-on domain or not, associated isolation and input side isolation signals with respective values, and associated supply rails. A sample of this section of the report looks like the following:

```
################################################################################

Always-On Domain: 'V2'    value: 1.50 volts    instance: 'Top.u'

Always-On Domain: 'V1'    value: 1.20 volts    module: 'Top'

################################################################################
```

FIGURE 24. Information on voltage domain

- Information about level shifters in the design

  This section contains information about the 'Level Shifter' cells with source and destination voltage domains. This section also provides information about the input pin, output pin, enable pin(s), input supply pin, output supply pin and location of the domain, where this cell can be

placed. A sample of this section of the report looks like the following:

```
################################################################################
Level shifter cell:'BASIC_LS'     inTerm:'A'  outTerm: 'Y'

Level shifter cell:'BASIC_LS_EN' inTerm:'A'  outTerm: 'Y' enableTerm(s):'E'

Level shifter cell:'lsV1_V2'(from,to): ('V1','V2') inTerm: 'A'  outTerm:'Y'
enableTerm(s): 'E'    location: 'V2'
################################################################################
```

FIGURE 25. Information about level shifters in the design

- Information about isolation cells in the design

  This section contains information about the 'Isolation' cells. The information about isolation cells is provided using the `isolation_cell` SGDC command or the `define_isolation_cell` CPF command. The cells specified in the library with attribute *is_isolation_cell* set as `true` are also considered as isolation cells. This section also contains information about input, output and enable pin(s) of the isolation cell. A sample of this section of the report looks like the following:

```
################################################################################
Isolation cell(s): 'BASIC_LS_EN'  input_pin : 'A'  output_pin : 'Y'  enable_pin : 'E'

Isolation cell(s): 'MY LS EN'     output pin : 'Y'
################################################################################
```

FIGURE 26. Information about isolation cells in the design

- Information about input isolation cells in the design

  This section contains information about the 'Input Isolation' cells. The information about isolation cells is provided using the `input_isocell` SGDC command or the `define_isolation_cell` CPF command. The cells specified in the library with attribute *is_isolation_cell* set as `true` are also considered as input isolation cells. This section also

contains information about input, output and enable pin(s) of the input isolation cell. A sample of this section of the report looks like the following:

```
#################################################################################

Inisolation cell : 'BASIC_LS_EN'    input_pin : 'A'    enable_pin : 'E'

Inisolation cell : 'MY_LS_EN'
#################################################################################
```

FIGURE 27. Information about input isolation cells in the design

- Information about always-on cells in the design

  This section contains information about the 'Always-on' cells. The information about always-on cells is provided using the `always_on_cell` SGDC command or `the define_always_on_cell` CPF command. A sample of this section of the report looks like the following:

```
#################################################################################

   aoncells : AN*      locate : BOTH

   aoncells : AN?      locate : AON

#################################################################################
```

FIGURE 28. Information about always-on cells in the design

- Information about always-on buffer cells in the design

  This section contains information about the 'Always-on-buffer' cells. The information about always-on cells is provided using the `always_on_buffer` SGDC command or the `define_always_on_cell` CPF command. The cells specified in the library with the *always_on* attribute set as `true` are also considered as always-on cells. This section contains always-on cells with the functionality of a buffer or inverter. A sample of this section of the report

looks like the following:

```
################################################################################
  name : 'AON_BUF'

  name : 'AON_BUF_2'

  name : 'RET DFF'    vddcPin : 'VDDC'
################################################################################
```

FIGURE 29. Information about always-on buffer cells in the design

- Information about state retention cells in the design

  This section contains information about the 'State Retention' cells. The information about state retention cells is provided using the *retention_cell* SGDC command or the `define_state_retention_cell` CPF command. The cells specified in the library with the *retention_cell* attribute set as `true` are also considered as retention cells. A sample of this section of the report looks like the following:

```
################################################################################
  retencell : 'FD1'    qTerm :'QN'   'Q'   clk : 'CP'   clkval: '1'

################################################################################
```

FIGURE 30. Information about state retention cells in the design

- Information about retention cells location in the design

  This section contains information about hierarchy information about retention cell location. The information about retention cells location is provided using the *retention_instance* SGDC command or the `-instances` argument of the `create_state_retention_cell`CPF command or the `-domain` and `-elements` arguments of `the set_retention` UPF command.

A sample of this section of the report looks like the following:

```
##########################################################################
  Hierarchical instance names : 'top.inst1'
##########################################################################
```

FIGURE 31. Information about retention cells location in the design

- Information from `pin_voltage` SGDC command

  This section contains information about pin voltage provided with the `pin_voltage` SGDC command. A sample of this section of the report looks like the following:

```
##########################################################################################
 Instance : 'top.ram_inst'  Module : 'SRAM1'  Pin : 'QT[0]'  VoltageDomain : 'MYVD'

 Instance : 'top.ram_inst'  Module : 'SRAM1'  Pin : 'QT[1]'  VoltageDomain : 'MYVD'

 ##########################################################################################
```

FIGURE 32. Information from pin_voltage SGDC command

FIGURE 33. Information about the supply rails in the design

This section contains information about the 'supply' rails defined in the design. This section contains information about supply rails provided with the `supply` SGDC command or the `create_power_nets/create_ground_nets` CPF command or the `create_supply_net` UPF command. A sample of this section of the report looks like the following:

```
##########################################################################
  supply : 'VDD'     value : '0.00'     alwayson : '1'

  supply : 'VSS'     value : '1.00'     alwayson : '1'
##########################################################################
```

FIGURE 34. Information about the supply rails in the design

- Information about the power switch in the design

  This section contains information about the 'Power switch' cells in the design. This section contains information about supply rails provided with the `power_switch` SGDC command or the `define_power_switch_cell` CPF command or the `create_power_switch` UPF command. A sample of this section of the report looks like the following:

```
##########################################################################

 name : 'PWRSW'  pwroutpin : 'VDDO'  pwrinpin : 'VDDI'  enablepin : 'ON'

##########################################################################
```

FIGURE 35. Information about the power switch in the design

- Information about the crossing that needs to be ignored

  This section contains information about the crossings that need to be ignored between two domains, where isolation cells are not required between the domains. This information is provided using `ignore_crossing` SGDC command.

  The information about crossings to be ignored can also be inferred from valid power states information for the design. The power state information is provided using the `power_state` SGDC command or the `create_power_mode` CPF command or the `create_pst` UPF command.

  A sample of this section of the report looks like the following:

```
##########################################################################

  Ignore domain crossing  from: 'V2'    to: 'V1'

##########################################################################
```

FIGURE 36. Information about the crossing that needs to be ignored

- Information about the always-on pins

  This section contains information about 'Always-On' pins. This section

contains information about the input pin(s) that must always be derived by a always-on cell. The information used in this section is provided with the `always_on_pin` SGDC command or the `-always_on_pins` argument of the `define_level_shifter_cell`, `define_isolation_cell`, and `define_state_retention_cell` CPF command. The input pins specified in the library with the *always_on* attribute set as `true` are also considered as always-on pins.

A sample of this section of the report looks like the following:

```
#################################################################################

     cell : 'aop_v1_v2'     pins: 'in'

     cell : 'aop_v2_v1'     pins: 'out'

#################################################################################
```

FIGURE 37. Information about the always-on pins

lp_cons_req

The *SGDC_lowpower_RuleReq* rule generates the lp_cons_req report. The report lists missing constraints that are required by rules run with SpyGlass

Power Verify solution. A sample report is as follows:

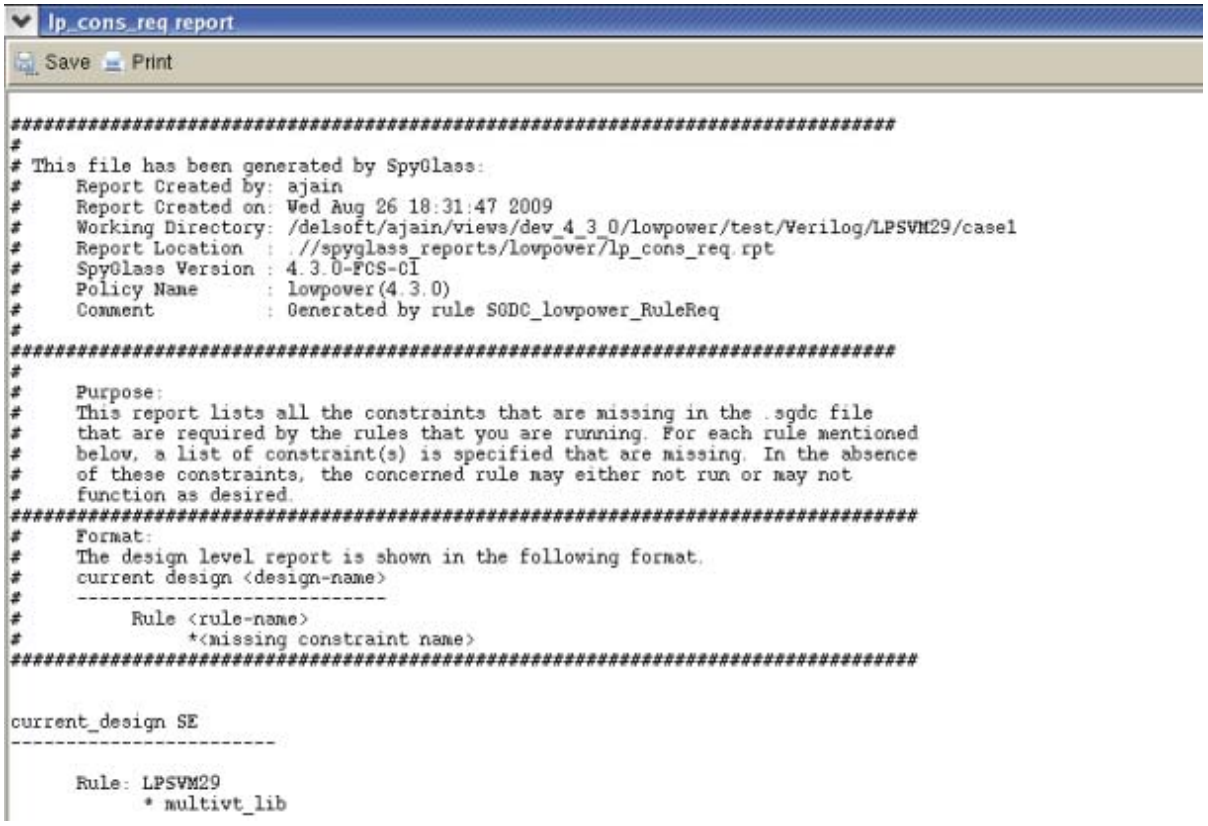


FIGURE 38. Sample lp_cons_req report

lp_crossing_data

The *LP_CROSSING_DATA* rule generates a report that contains domain crossing data information as inferred by SpyGlass using the SGDC/CPF/UPF files and library files. These reports are generated in the `spyglass_reports/lowpower` directory.

This report is generated in two variants.

- *lp_crossing_data.csv*

- *lp_crossing_data.rpt*

lp_crossing_data.csv

You can view the lp_crossing_data.csv report in spreadsheet viewer by double clicking on the violation message.

The following are sample snapshots of the lp_crossing_data.csv report.

NOTE: *The snapshots shown here are not complete. For the full details of the spreadsheet report, see the table given after these snapshots.*

A Input Domain	B Input Domain Voltage Range (Lower Bound)	C Input Domain Voltage Range (Upper Bound)	D Output Domain
line_fifo/PD_AON2	0.9	0.9	PD_TOP
...erator/vtgen/PD_AON2	0.9	1.2	PD_TOP
...nerator/vtgen/PD_SW2	0.9	1.2	PD_TOP
PD_TOP	1	1	...m/PD_AON1
clut_mem/PD_AON1	1.1	1.3	PD_TOP

F Output Domain Voltage Range (Upper Bound)	G Level Shifter(Type Required)	H Level Shifter Cells available	I Isolation(Required)	J Cross Prob
1.3	Low to High	None	Not Required	Show Relationship
1	High to Low	None	Not Required	Show Relationship
1	Low to High	LS01; LS02; LS03;	Not Required	Show Relationship
1	Both	None	Not Required	Show Relationship
1	Both	None	Output Isolation	Show Relationship

FIGURE 39. Sample lp_crossing_data report

The following table contains the details of the information provided in this

spreadsheet report:

Column Name	Information Provided
Input Domain	Hierarchical name of the source domain
Input Domain Voltage Range (Lower Bound)	Lower bound of the voltage range of the source domain
Input Domain Voltage Range (Upper Bound)	Upper bound of the voltage range of the source domain
Output Domain	Hierarchical name of the destination domain
Output Domain Voltage Range (Lower Bound)	Lower bound of the voltage range of the destination domain
Output Domain Voltage Range (Upper Bound)	Upper bound of the voltage range of the destination domain
Level Shifter (Type Required)	Type of level shifter required for the crossing
Level Shifter Cells Available	List of level shifter cells available for the crossing in the library
Isolation (Required)	Type of isolation required for the crossing
Cross Probe	Click on the Show Relationship hyper link to view the Supply Net Relationship widget. This widget displays the relationship between the primary supplies of the source and destination domains, and isolation and level shifter requirements.

lp_crossing_data.rpt

The lp_crossing_data.rpt report shows the following information:

- Input Domain and its Voltage Range: Shown as `<domain-name> (min voltage-max voltage)`
- Output Domain and its Voltage Range: Shown as `<domain-name> (min voltage-max voltage)`
- Type of level shifting required: Shown as one of `LH | HL | LH_HL | None`
- Exclude Pins: Shows the list of pins that does not need level shifting between the specified domains

- Allowed Level Shifters: Shows the list of user specified level shifters for the specified crossing
- Available Level Shifters: Shows the list of level shifters cells in libraries that can be placed on this crossing
- LS Location: Shows the list of allowed domains, where level shifter should be placed on the specified crossing

A sample lp_crossing_data report is as follows:

FIGURE 40. Sample lp_crossing_data report

lp_domain_info

The *lp_domain_info* report contains information related to domains for the domain crossings and the number of instances.

A sample *lp_domain_info* report is as follows:

```
################################################################################
# This section contains information about voltage domains in design
# The information can be given from SGDC using command voltage_domain
# or command create_power_domain in CPF and UPF
# This section also contains information such as domain value, whether
domain
# is Always-On or not, associated isolation signals with their isolation
value,
# associated input side isolation signals with respective isolation
values
################################################################################
Domain:PD2
Type:On-Off
Domain SupplyList:
VDD2
VSS
Primary Power Net : VDD2
Primary Ground  Net : VSS
Number of flop instances : 14
Number of gates : 21
Number of flat instances : 35
Number of flat instances (cumulative) : 50

Domain:PD1
Type:On-Off
Domain SupplyList:
VDD1
VSS
Primary Power Net : VDD1
Primary Ground  Net : VSS
Number of flop instances : 14
Number of gates : 21
Number of flat instances : 35
Number of flat instances (cumulative) : 50

Domain:TOP
Type:Always-On
Domain SupplyList:
```

```
VDD
VSS
Primary Power Net : VDD
Primary Ground  Net : VSS
Number of flop instances : 2
Number of gates : 3
Number of flat instances : 5
Number of flat instances (cumulative) : 75

######################### Dumping Instances List#########################
'PD2' instances:
  instance: 'top.instA1.instB1'
'PD1' instances:
  instance: 'top.instA2.instB1'
'PD2' instances:
  instance: 'top.instA2'
'PD1' instances:
  instance: 'top.instA1'
'TOP' modules:
  module: 'top'
```

lp_lib_data

The *LP_LIB_DATA* rule generates a report named lp_lib_data that contains information from the library files as interpreted by the SpyGlass Power Verify solution. The report shows comprehensive library information with respect to Level Shifter, Isolation, Always-on, Retention, Multiple Supply, Power Switch, and Tie cells.

NOTE: *For all the sections in the* lp_lib_data *report, if relevant information is specified in both, the library file as well as CPF file, the concluding information is merged and reported. However, in case of any conflict, the information provided with CPF command gets precedence and is used.*

The lp_lib_data report contains the following sections:

- **Information about level shifter cells**

This section contains information about 'level shifter' cells specified in the library. This section reports cells from the library specified with the attribute *is_level_shifter* set as `true` or library cells specified with the `define_level_shifter_cell` CPF command in the CPF flow.

A sample of this section of the report looks like the following:

```
#################################################################################

Cell       |Type |Input     |Output    |Input|Output|Enable|Input |Output|Ground|AON
Name       |     |VoltRange|VoltRange|Pin  |Pin   |Pins  |Supply|Supply|Supply|Pins
-------------------------------------------------------------------------------
ISO_EN_BIR|HL_LH|0.80,1.20|0.80,1.20|A    |Y     |E     |VDDA  |VDDY  |VSS   |A E
MY_ISO    |     |          |          |     |Q     |      |      |VDD   |VSS   |

#################################################################################
```

FIGURE 41. Information about level shifter cells

■ Information about Isolation cells

This section contains information about 'Isolation' cells present in library. This section reports cells from the library specified with the attribute *is_isolation_cell* or library cells specified with the `define_isolation_cell` CPF command in the CPF flow.

A sample of this section of the report looks like the following:

```
#################################################################################

Cell           |Cell  |Input|Output|Enable|Enable|Power    |Ground|Always-on
Name           |Type  |Pin  |Pin   |Pin   |Level |Supply   |Supply|Pins
-------------------------------------------------------------------------------
BASIC_ISO      |      |A    |Y     |      |      |         |VSS   |A
BASIC_ISO_EN   |AND   |A    |Y     |E     |HIGH  |VDDA     |VSS   |E
ISO_AND        |AND   |     |Y     |      |      |VDD      |VSS   |
ISO_AND_EN     |AND   |B    |Y     |A     |HIGH  |         |      |
ISO_EN_BIR     |AND   |A    |Y     |E     |HIGH  |VDDY     |VSS   |A E
ISO_TRI_0      |      |A    |Y     |OE    |LOW   |         |      |
ISO_TRI_1      |      |A    |Y     |OE    |HIGH  |VDD1     |VSS   |

#################################################################################
```

FIGURE 42. Information about Isolation cells

■ Information about Always-on cells

This section contains information about the 'Always-on' cells. This

section reports cells from the library specified with the attribute *always_on* applied at cell level or cells with at least one power pin specified with *pg_type* as `backup_power`. The always-on cells with functionality buffer or inverter are also treated as Always-On Buffers.

A sample of this section of the report looks like the following:

```
#########################################################################################

Cell            |Cell       |Always-on |Switched  |Always-on |Switched  |Always-on
Name            |Type       |Power Pin |Power Pin |Ground Pin|Ground Pin|Pins
-----------------------------------------------------------------------------------------
Always_On       |BUFFER     |          |VDDC      |          |VSS       |

#########################################################################################
```

FIGURE 43. Information about Always-on cells

Information about Retention cells

This section contains information about 'Retention' cells present in library. This section reports cells from the library specified with the attribute *retention_cell* or library cells specified with the `define_state_retention_cell` CPF command in the CPF flow. This section also displays active values of Save and Restore pins. A sample of this section of the report looks like the following:

```
#########################################################################################

Cell     |Cell   |Always-On |Switched |Save     |Restore |Clk     |QTerm |Input |AON
Name     |Type   |PwrPin    |PwrPin   |Pin:Val  |Pin:Val |Pin:Val|      |      |Pins
-----------------------------------------------------------------------------------------
R_F      |true   |VDDC      |VDD      |SAVE   :0|RST   :1|CLK:1  |Q     |D     |SAVE RST CL

#########################################################################################
```

FIGURE 44. Information about Retention cells

Information about Multiple Supply cells

This section contains information about the 'Multiple Supply' cells. This section reports cells from the library with more than one power and ground pins specified with the *pg_pin* group attribute.

This section also provides information of the related signal pins to that supply pin. A sample of this section of the report looks like the

following:

```
##########################################################(
Cell Name                      | Supply Pins (Type) | Related Signal Pins¶
---------------------------------------------------------------------¶
HS45_LS_RSDFPQNTX17_CNHL  |gndi (primary_ground)  |CP D QN TE TI TQ(
                          |gndo (backup_ground)  |SLEEP(
                          |vddi (primary_power)  |CP D QN TE TI TQ(
                          |vddo (backup_power)   |SLEEP(
```

FIGURE 45. Information about Multiple Supply cells

- **Information about Power Switch cells**

  This section contains information about the 'Power Switch' cells in the design. A sample of this section of the report looks like the following:

```
#############################################################################
Cell Name |Cell  |Cell     |Input    |Switched |Input    |Switched |Output |Input
          |Type  |Function|pwr pin  |pwr pin  |gnd pin  |gnd pin  |Enable |Enable
-----------------------------------------------------------------------------
HEADBUF16_X1M_A9TR
          HEADER|BUFFER    |VDDG    |VDD      |VSS      |         |SLEEPOUT|SLEEP

#############################################################################
```

FIGURE 46. Information about Power Switch cells

- **Information about Tie cells**

  This section contains information about the 'Tie' cells inferred from the provided library. A sample of this section of the report looks like the following:

```
#############################################################################
Cell Name      |Cell TIE |Always On|Switched|Switched|Always-on|Always-on
               |Function |        |pwr pin |gnd pin |pwr pin  |gnd pin
-----------------------------------------------------------------------------
TIEHI_X1M_A9TH |TIE-HIGH |NO      |VDD     |VSS     |         |
TIELO_X1M_A9TH |TIE-LOW  |YES     |VDD     |VSS     |         |
#############################################################################
```

FIGURE 47. Information about Tie cells

- **Information on Isolation cells with the dont_use attribute**

```
#############################################################################
# This section contains List of 'Isolation Cells with dont_use
attribute' present in library.
#
#############################################################################
Cell Name
-----------------------------------------------------------------------
iso_cell1
iso_cell2
```

- **Information on Retention cells with the dont_use attribute**

```
#############################################################################
# This section contains List of 'Retention Cells with dont_use
attribute'  present in library.
#
#############################################################################
Cell Name
-----------------------------------------------------------------------
Retention_cell1
Retention_cell2
```

- **Information on Power Switch cells with the dont_use attribute**

```
#############################################################################
# This section contains List of 'Power Switch Cells with dont_use
attribute' present in library.
#
#############################################################################
Cell Name
-----------------------------------------------------------------------
```

```
power_switch1
power_switch2
```

lp_multivtreport

Report Generated by the LPSVM29 Rule

The *LPSVM29* rule generates a report named lp_multivtreport. The report contains the instantiation information about VT type library cells in the complete design. The lp_multivtreport report displays the cells composition of different vt type libraries, provided as either LIB or SGLIB, to SpyGlass command-line. You can also provide the library with the constraint `multivt_lib`.

A sample report is as follows:

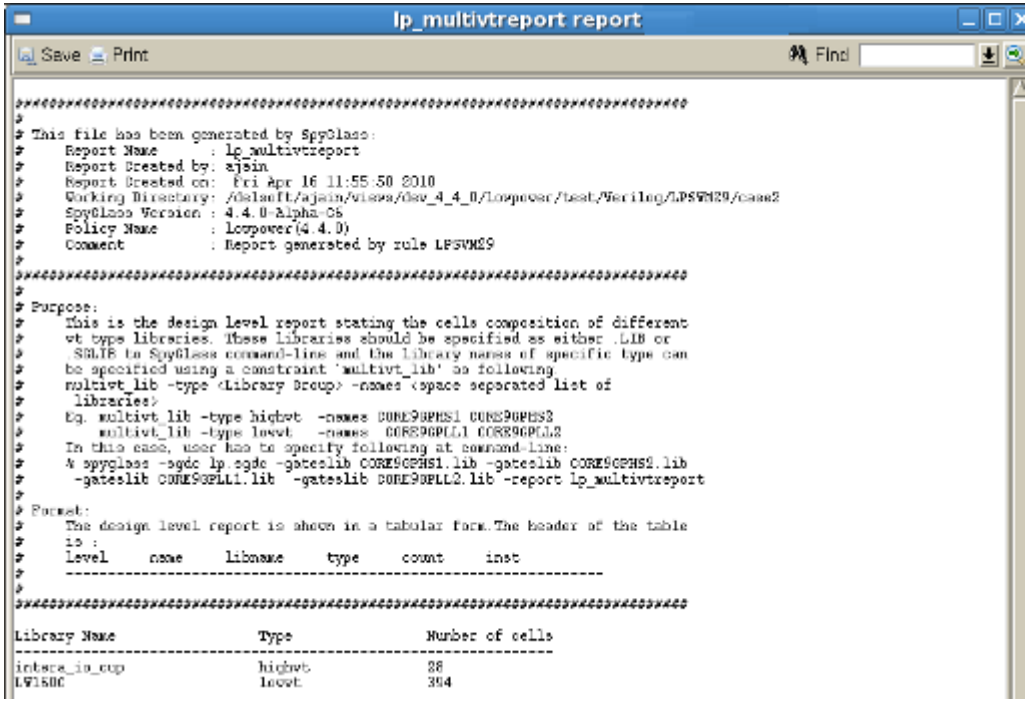


FIGURE 48. Sample lp_multivtreport report

lp_multivt_perblock

Report Generated by the LPSVM29 Rule

The *LPSVM29* rule generates a report named lp_multivt_perblock that has the instantiation information about VT type library cells in each hierarchy. These libraries can be provided as either LIB or SGLIB, to SpyGlass command-line or specified with the constraint `multivt_lib` as follows:

```
multivt_lib -type <Library Group> -names <space separated
list of libraries>
```

A sample report is as follows:

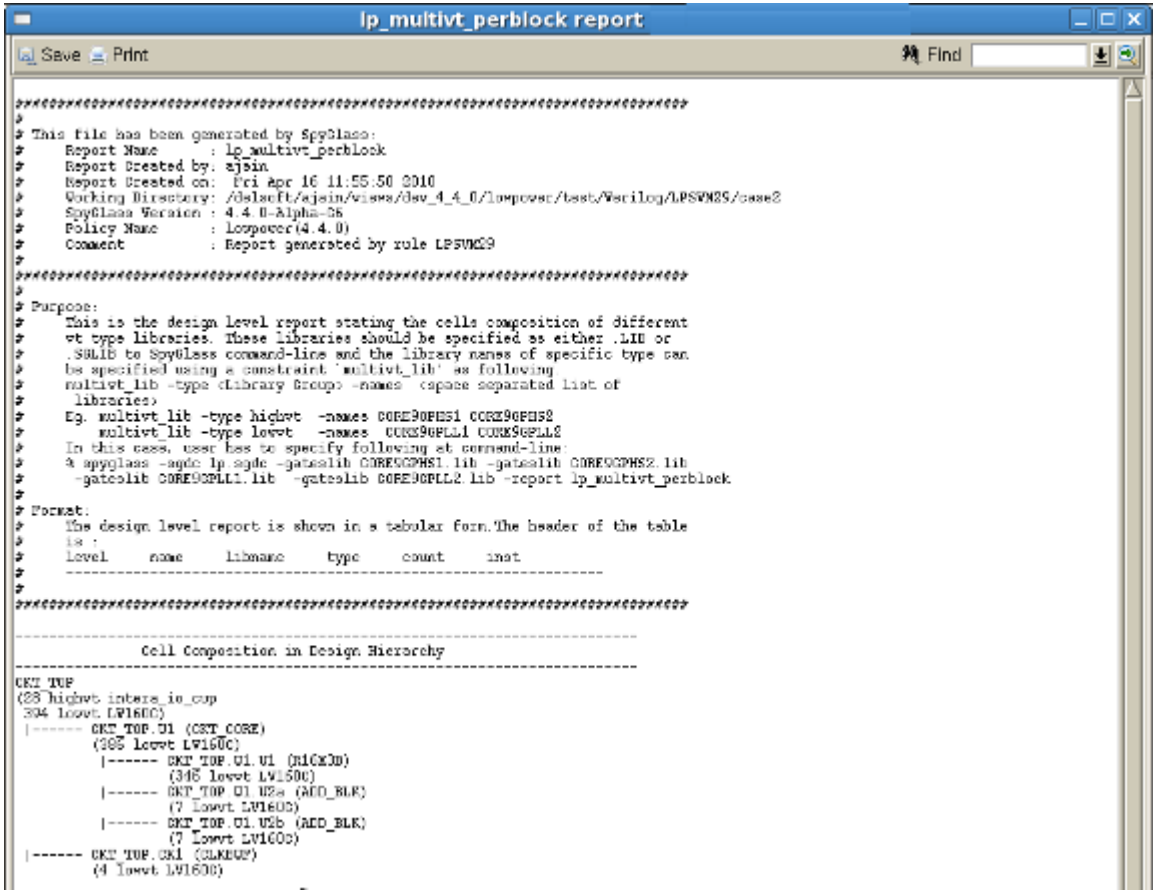


FIGURE 49. Sample lp_multivt_perblock report

lp_multi_domain_crossing_check

Report Generated by the LP_MULTI_DOMAIN_CROSSING_CHECK Rule

The *LP_MULTI_DOMAIN_CROSSING_CHECK* rule generates a report named lp_multi_domain_crossing_check.csv that has information of all the signals that are going to more than two domains.

The `lp_multi_domain_crossing_check.csv` report is generated in the `spyglass_reports/lowpower` directory. This spreadsheet report is also available in SpyGlass GUI.

A sample spreadsheet report is as follows:

	C	D	E	F	G	H	I	
	Source Domain Name	Source Supply Name	Destination Name	Destination Domain Name	Destination Supply Name	Intermediate cell Name	Intermediate Domain Name	Interme
1	TOP	VDD	top.inst2.df1.D	PD2	VDD2	top.inst1.bf3.A	PD1	VDD1

FIGURE 50. Sample lp_multi_domain_crossing_check.csv report

lp_nested_domain_info

The `lp_nested_domain_info` report contains information related to nested domains for each domain.

A sample `lp_nested_domain_info` report is as follows:

```
#################################################################
# This section contains information about nested voltage
# domains in design

# The information can be given from create_power_domain in
# UPF. The domain hierachy is taken from design not from
# power specification. Gate count is taken with inverters,
# buffers ect..
#################################################################
Domain :: TOP
```

```
Nested Domains ::
PD1
PD2

Domain :: PD1
Nested Domains ::

Domain :: PD2
Nested Domains ::
PD3

Domain :: PD3
Nested Domains ::
```

multi_domain_crossing_check

Report Generated by the LP_MULTI_DOMAIN_CROSSING_CHECK Rule

The *LP_MULTI_DOMAIN_CROSSING_CHECK* rule generates a report named lp_multi_domain_crossing_check.csv that has information of all the signals that are going to more than two domains.

The `lp_multi_domain_crossing_check.csv` report is generated in the `spyglass_reports/lowpower` directory. This spreadsheet report is also available in SpyGlass GUI.

A sample spreadsheet report is as follows:

	C	D	E	F	G	H	I	
	Source Domain Name	Source Supply Name	Destination Name	Destination Domain Name	Destination Supply Name	Intermediate cell Name	Intermediate Domain Name	Interme
1	TOP	VDD	top.inst2.df1.D	PD2	VDD2	top.inst1.bf3.A	PD1	VDD1

FIGURE 51. Sample lp_multi_domain_crossing_check.csv report

lp_power_data

Report Generated by the LP_POWERDATA_INFO Rule

The *LP_POWERDATA_INFO* rule generates a report named lp_power_data that contains information about user-specified CPF/UPF commands parsed by SpyGlass.

A sample report is as follows:

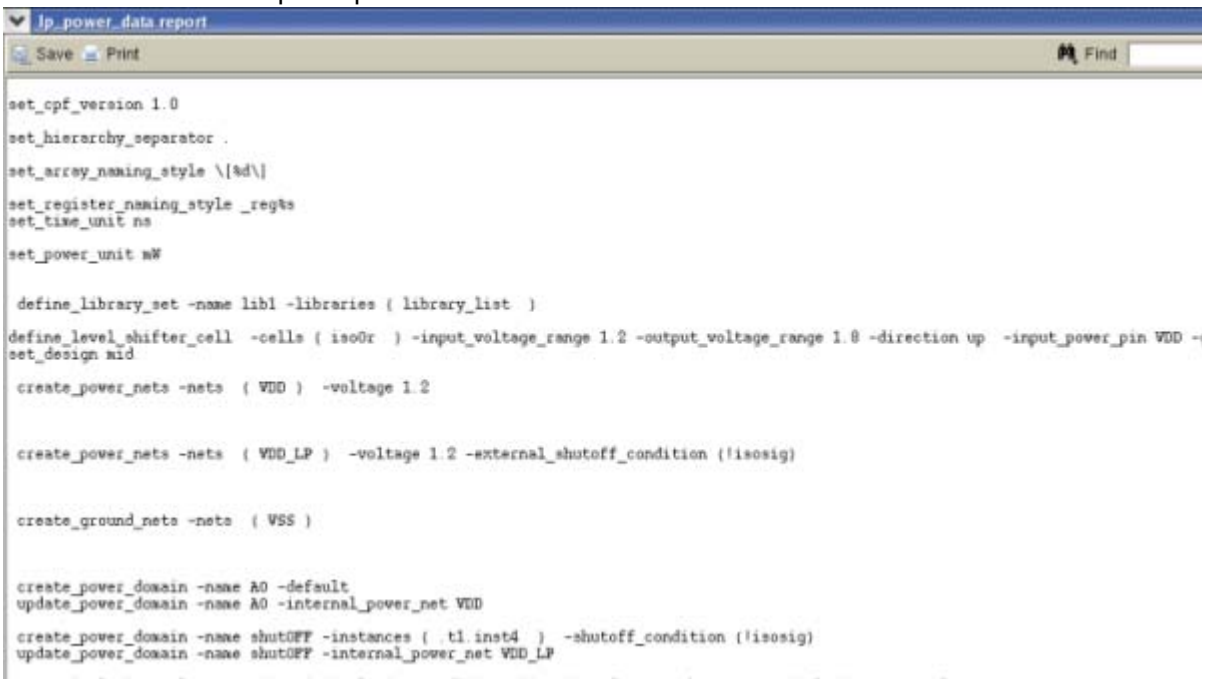


FIGURE 52. Sample lp_power_data report

lp_power_state

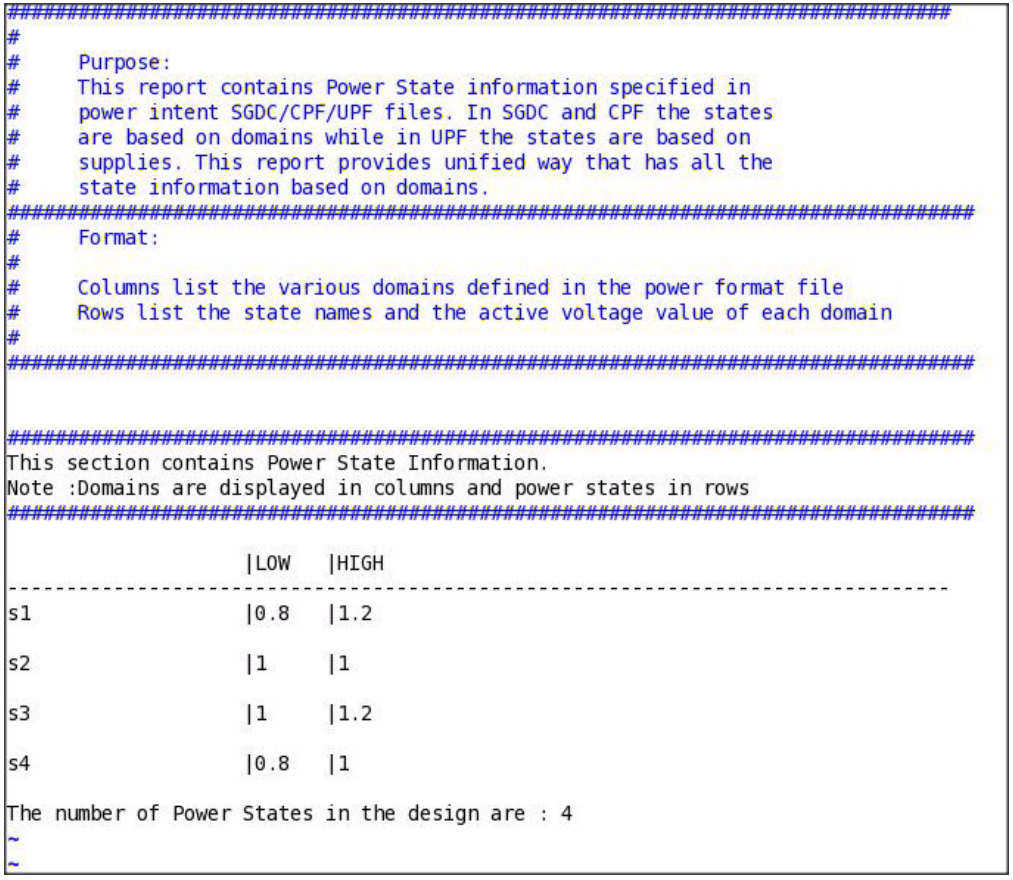
The *LP_CROSSING_DATA* rule generates a report named lp_power_state showing the power state information specified in the power intent SGDC/CPF/UPF files.

The lp_power_state report is shown in a tabular format, where columns list the domain names defined in various power states and rows list the power state names. The report displays active voltage value of each domain for all power states.

This report also displays the number of power states possible in the design.

NOTE: *This report is also generated in CSV format.*

A sample lp_power_state report is as follows:



```
################################################################################
#
#     Purpose:
#     This report contains Power State information specified in
#     power intent SGDC/CPF/UPF files. In SGDC and CPF the states
#     are based on domains while in UPF the states are based on
#     supplies. This report provides unified way that has all the
#     state information based on domains.
################################################################################
#     Format:
#
#     Columns list the various domains defined in the power format file
#     Rows list the state names and the active voltage value of each domain
#
################################################################################


################################################################################
This section contains Power State Information.
Note :Domains are displayed in columns and power states in rows
################################################################################

                    |LOW   |HIGH
--------------------------------------------------------------------------------
s1                  |0.8   |1.2

s2                  |1     |1

s3                  |1     |1.2

s4                  |0.8   |1

The number of Power States in the design are : 4
~
~
```

FIGURE 53. Sample lp_power_state report

lp_pst_merge_message

The *UPF_lowpower08* rule generates a report named lp_pst_merge_message showing the PST states which could not be merged with any PST. *Figure 54* shows a sample of the lp_pst_merge_message report.

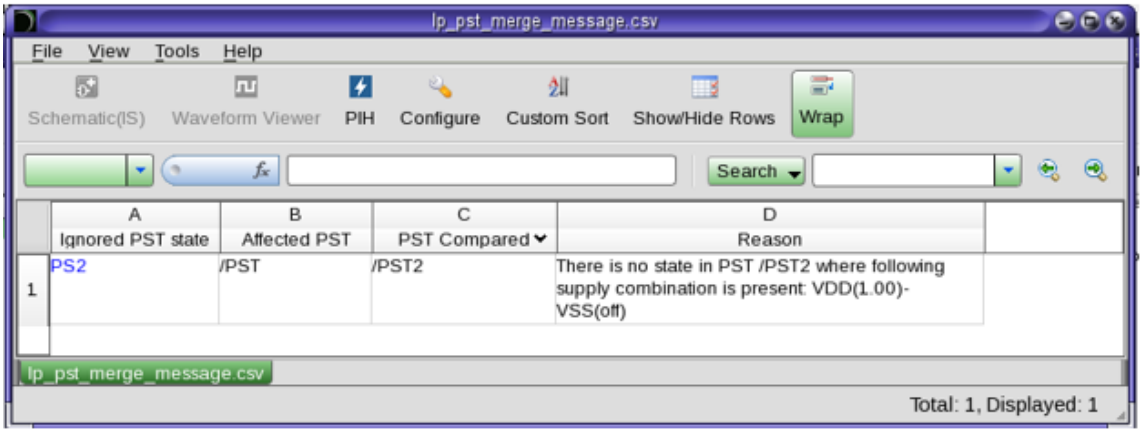


FIGURE 54. Sample lp_pst_merge_message report

The *Table 2* describes the columns in the report.

TABLE 2 Columns in the lp_pst_merge_message report

Column	Description
Ignore PST state	The PST state which ignored in the process of pst merge
Affected PST	The PST in which ignored state is present
PST compared	The PST which does not contain the supply state combination of the ignored PST state
Reason	The detailed reason

lp_psw_info

The *LPPLIB17* rule generates a report named *lp_psw_info.* This report contains the list of valid power switch strategies defined in the UPF file. In addition, this report contains the expanded list of elements to which that strategy is applied.

After specifying multiple general and element specific strategies on a design element, the SpyGlass Power Verify solution uniquifies these strategies and applies only one strategy on a design element. You can use

this report to identify the strategy applied by the SpyGlass Power Verify solution on that design element.

A sample lp_psw_info report is as follows:

```
#################################################################
# Purpose:
# This report contains the list of Power Switch Strategies
  defined in
# UPF along with the expanded list of elements  to which that
  strategy is applied.
#################################################################
PSW Strategy :/ps_rule1
Element List :
top/l5/l1

PSW Strategy :/ps_rule2
Element List :
top/l5/l1
```

lp_ret_info

The *LPRET01* rule generates a report named *lp_ret_info. This* report contains the list of valid retention strategies defined in the UPF file. In addition, this report contains the expanded list of elements to which that strategy is applied.

After specifying multiple general and element specific strategies on a design element, the SpyGlass Power Verify solution uniquifies these strategies and applies only one strategy on a design element. You can use this report to identify the strategy applied by the SpyGlass Power Verify solution on that design element.

A sample *lp_ret_info* report is as follows:

```
#################################################################
#     Purpose:
#     This report contains the list of Retention Strategies
      defined in
#     UPF along with the expanded list of elements  to which
      that strategy is applied.
```

```
#################################################################
RET Strategy :/ret3
Element List :
top/inst1/inst2/inst3/inst5/ret9
top/inst1/inst2/inst3/inst5/ret10
top/inst1/inst2/inst4/inst6/ret9
top/inst1/ret1
top/inst1/ret2
top/ret0
```

lp_related_pin_data_for_consistency_check

The *UPF_lowpower26* rule generates report named `lp_related_pin_data_for_consistency_check.csv`.

This report contains information about the interpretation of the related supplies from UPF and lib.

The report is generated in the `spyglass_reports/lowpower` directory. This spreadsheet report is also available in SpyGlass GUI.

The following image displays an example of the `lp_related_pin_data_for_consistency_check` report in CSV format, generated by the *UPF_lowpower26* rule:

NOTE: *For the full details of the csv report, see the table given after this snapshot.*

	A PIN NAME	B RELATED POWER SUPPLY IN LIB	C RELATED POWER SUPPLY IN UPF	D POWER SUPPLY USED FR
1	top.inst1.IO1	vddA	vddA	DOMAIN SUPPLY
2	top.inst1.IO2	vddA	vddA	DOMAIN SUPPLY
3	top.inst1.IO3	vddB	vddB	SRSN
4	top.inst1.W	vddA	vddA	DOMAIN SUPPLY
5	top.inst1.X	vddB	vddB	SRSN
6	top.inst1.Y	vddB	vddA	DOMAIN SUPPLY
7	top.inst1.Z	vddA	vddA	DOMAIN SUPPLY
8	top.inst1.Z1	vddB_int	vddA	DOMAIN SUPPLY
9	top.inst1.Z2	vddB	vddB	SRSN
10	top.inst2.IO1	vddA	vddA	DOMAIN SUPPLY
11	top.inst2.IO2	vddA	vddA	DOMAIN SUPPLY
12	top.inst2.IO3	vddB	vddB	SRSN
13	top.inst2.W	vddA	vddA	DOMAIN SUPPLY
14	top.inst2.X	vddB	vddB	SRSN
15	top.inst2.Y	vddB	vddA	DOMAIN SUPPLY
16	top.inst2.Z	vddA	vddA	DOMAIN SUPPLY
17	top.inst2.Z1	vddB_int	vddA	DOMAIN SUPPLY
18	top.inst2.Z2	vddB	vddB	SRSN

FIGURE 55. Sample snapshot of the lp_related_pin_data_for_consistency_check report.

The following table contains the details of the information provided in this spreadsheet report:

Column Name	Information Provided
PIN NAME	Name of the pin checked
RELATED POWER SUPPLY IN LIB	Related power supply used from LIB
RELATED POWER SUPPLY IN UPF	Related power supply used from UPF
POWER SUPPLY USED FROM	Source of power supply in UPF
RELATED GROUND SUPPLY IN LIB	Related ground supply used from LIB

Column Name	Information Provided
RELATED GROUND SUPPLY IN UPF	Related ground supply used from UPF
GROUND SUPPLY USED FROM	Source of ground supply in UPF

lp_retention_cell_list

The LP *LPRET01* rule generates a report named `lp_retention_cell_list`. This report lists the retention instances found in the design.

A sample lp_retention_cell_list report is as follows:

```
#################################################################
####################
#
# This file has been generated by SpyGlass:
#     Report Created by: deeptanshus
#     Report Created on: Tue Jul 30 16:30:42 2013
#     Working Directory: /delsoft/deeptanshus2/views/LPRET01
lowpower/test/PV_TestCases/FT_2013/AUG_FT/5104-Patch/78083/
case2
#     Report Location  : ./Work/Linux4/test_reports/lowpower/
lp_retention_cell_list.rpt
#     SpyGlass Version : 5.1.0.4
#     Policy Name      : lowpower(5.1.0.4)
#     Comment          : Generated by rule LPRET01
#
#################################################################
####################
# Purpose:
# This report contains count of retention cells found by
LPRET01
#################################################################
####################
```

```
Total Number of Flops/Latches/Retention Cells : 12
Total Number of Retention Cells on which Retention Strategy
is Applied : 3
List of Retention Cells on which Retention Strategy is
Applied :
============================================================
=====
top.inst1.inst2.ret2
top.inst1.ret1
top.ret3
```

lp_srsn_info

The *UPF_lowpower11* rule generates a report named `lp_srsn_info`. This report lists the ports on which correct supply information using the *set_related_supply_net* command is specified.

A sample `lp_srsn_info` report is as follows:

```
#############################################################
#########################
#     Purpose:
#     This report contains the list of ports for which
#     'set_related_supply_net' command is written correctly
#############################################################
#########################


[ const.upf : 32 ]
         top.my_levelshifter.in2
```

lp_supply_connection

The *LP_CROSSING_DATA* rule generates a report named lp_supply_connection showing the supply connection information specified in the UPF power intent files.

The lp_supply_connection report displays the instance path name, followed by the power pin and corresponding supply net connection.

A sample lp_supply_connection report is as follows:

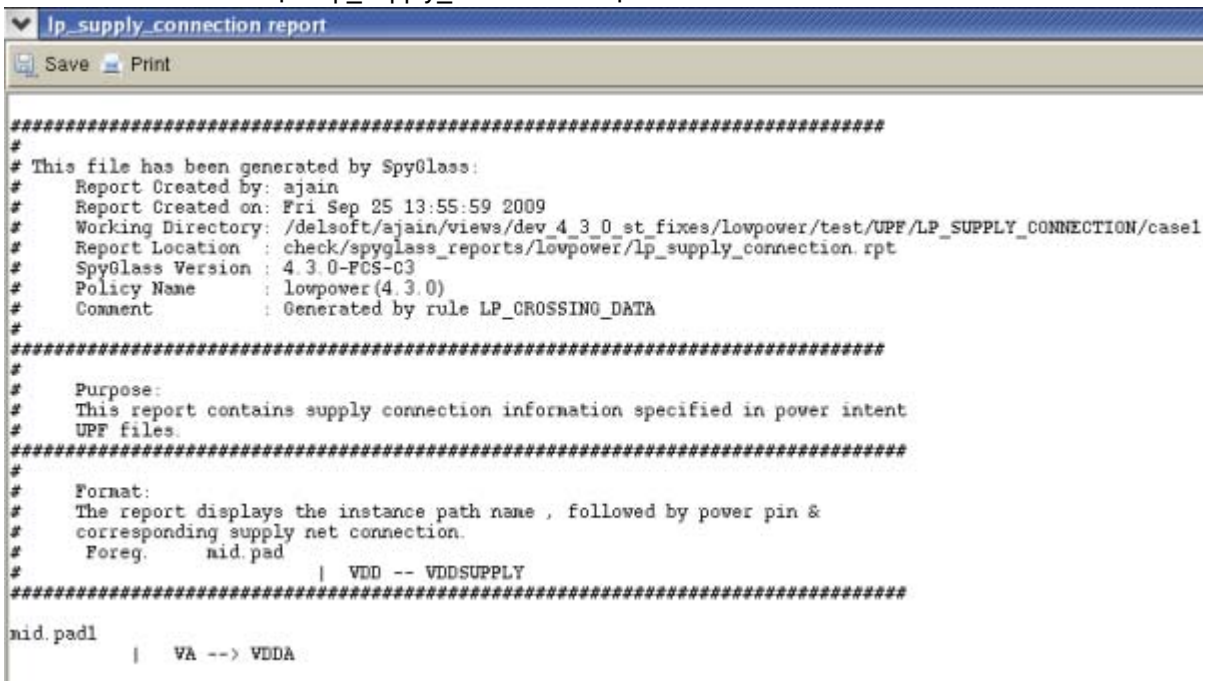


FIGURE 56. Sample lp_supply_connection report

This report also shows the related supply net information specified with the `set_related_supply_net` UPF command, displaying the instance name or top design, followed by the signal pin and the corresponding

supply net connection as follows.

```
################################################################################
#
#     Purpose:
#     This report contains related supply net information specified in power intent
#     UPF files , as specified through the command set_related_supply_net. This
#     command overrides the information specified through connect_supply_net.
################################################################################
#
#     Format:
#     The report displays the instance name or top design, followed by signal pin &
#     corresponding supply net connection.
#      Foreg.     mid.pad
#                          |  SIG1 --> VDDSUPPLY
################################################################################

mid.pad1

            | P2 --> VDDB
```

FIGURE 57. Supply net connection information

lp_vd_info

The lp_vd_info report is generated by the following rules:

Report Generated by the LPSVM04 Rule

The *LPSVM04A* rule reports missing level shifters at voltage domain

crossings. A sample lp_vd_info report for the *LPSVM04A* rule is as follows:

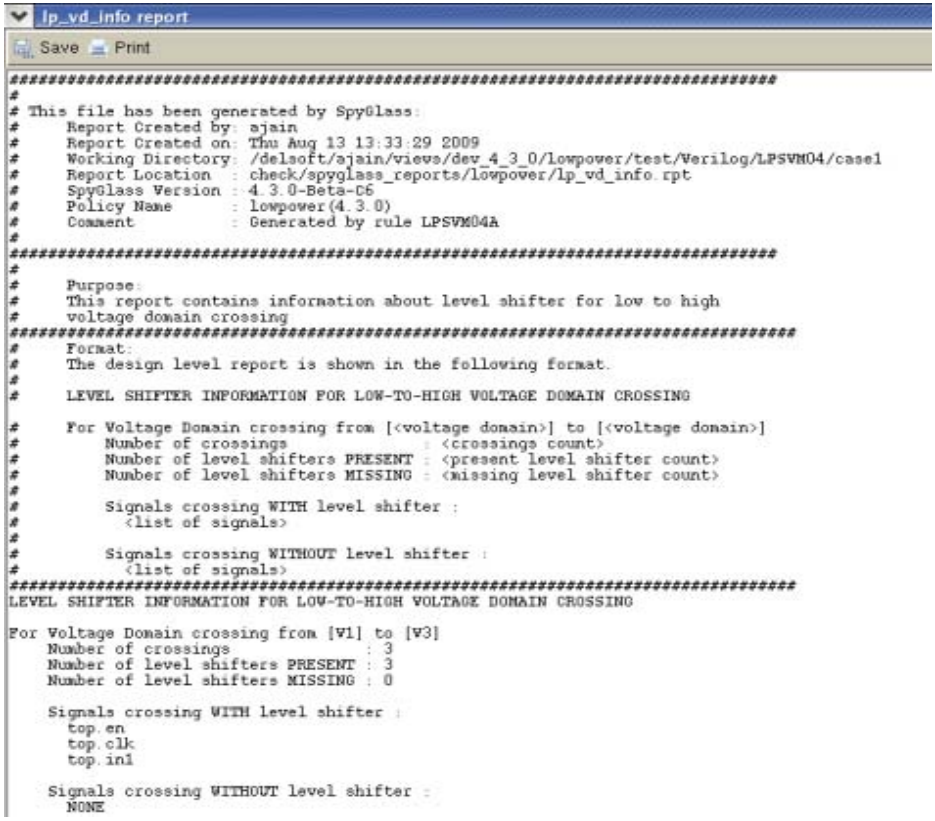


FIGURE 58. Sample lp_vd_info report generated by the LPSVM04 rule

The report shows level shifter information for low to high voltage domain crossing. The report displays the following details:

- Number of crossings
- Number of existing and missing level shifters
- Name of signals with and without level shifters

Report Generated by the LPSVM10 Rule

The *LPSVM10* rule reports the steady state values at the output of the power domain. The report contains the following details:

- Name of the isolation signal and their value
- Signal name and steady-state value at the output of power domain.

A sample lp_vd_info report is as follows:

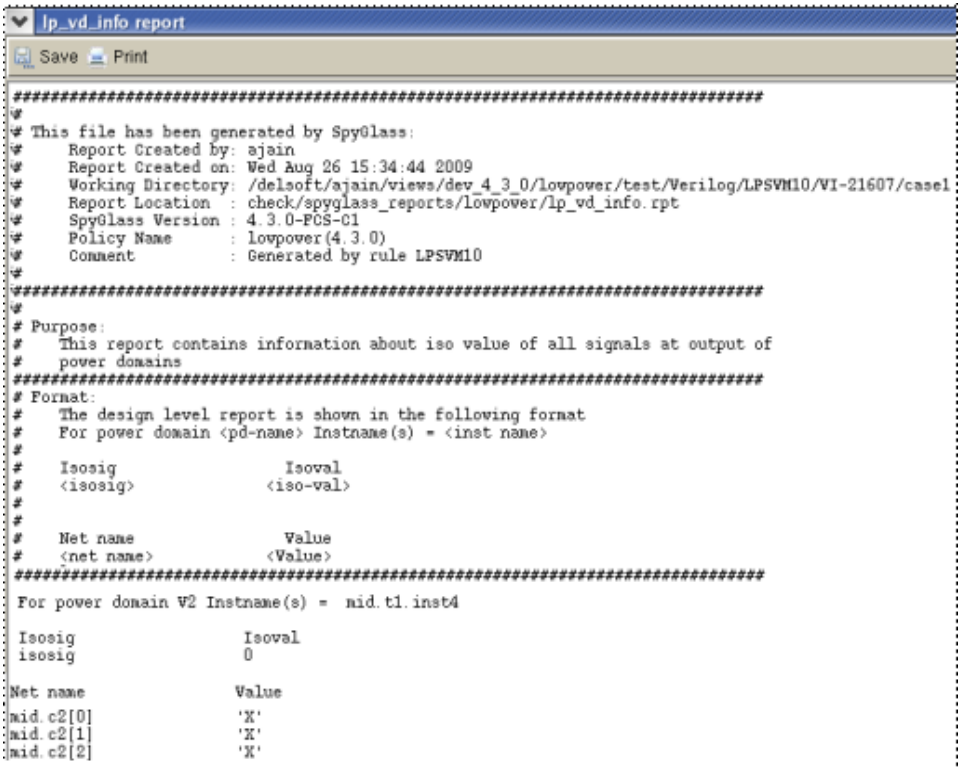


FIGURE 59. Sample lp_vd_info report generated by the LPSVM10 rule

Report Generated by the LPSVM23 Rule

The *LPSVM23* generates information about the missing isolation logic for

power domains in the design. A sample lp_vd_info report is as follows:

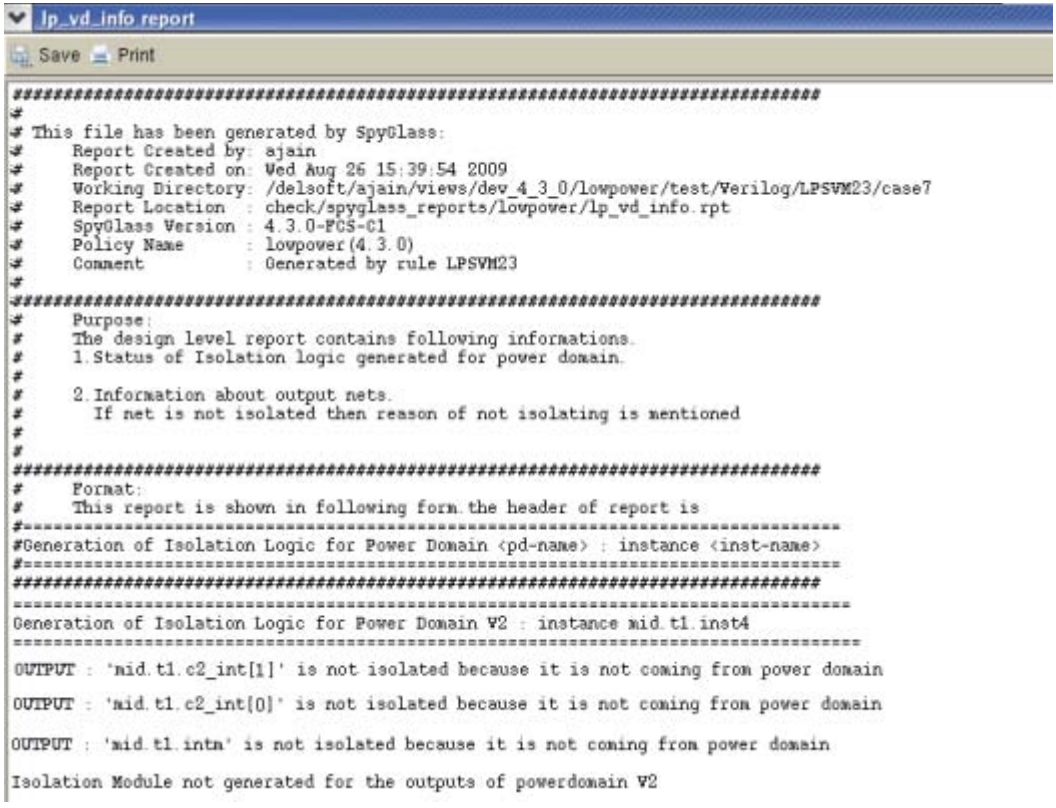


FIGURE 60. Sample lp_vd_info report generated by the LPSVM23 rule

The report displays the following details:

- Status of isolation logic: If isolation logic is not generated for a power domain, then the report states the reason why isolation logic has not been generated.
- Isolation signal: If no valid isolation signal is specified for instance `<instance-name>` through the `-isosig` field, then `dummy_iso_signal_blocking` is used to generate the isolation logic. The isolation signal and instance name should be in the same hierarchy.

- Output nets: If net is not isolated then reason of not isolating is mentioned

Report Generated by the LPSVM45 Rule

The *LPSVM45* rule reports the total number of power switches (by type) that are connected to each enable port of each power domain in a report.

A sample lp_vd_info report is as follows:

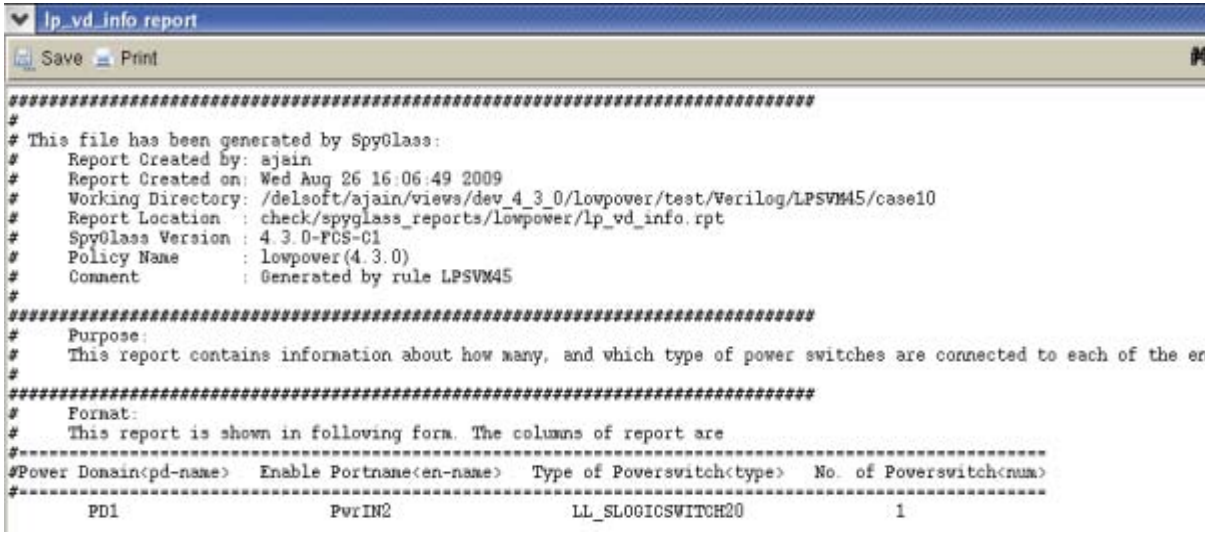


FIGURE 61. Sample lp_vd_info report generated by the LPSVM45 rule

The report contains the following details:

- Number and type of power switches connected to each of the enable ports on each power domain.

Report Generated by the LPSVM46 Rule

The *LPSVM46* rule reports the following information:

- Source cell driving each RAM switch enable pin and the instance name of the RAM switch
- Number of RAM instances in each power domain

A sample lp_vd_info report is as follows:

lp_vd_info report

Save Print

```
##############################################################################
#
# This file has been generated by SpyGlass:
#     Report Created by: ajain
#     Report Created on: Wed Aug 26 16:10:25 2009
#     Working Directory: /delsoft/ajain/views/dev_4_3_0/lowpower/test/Verilog/LPSVM46/case1
#     Report Location  : check/spyglass_reports/lowpower/lp_vd_info.rpt
#     SpyGlass Version : 4.3.0-PCS-C1
#     Policy Name      : lowpower(4.3.0)
#     Comment          : Generated by rule LPSVM46
#
##############################################################################
#     Purpose:
#     This report contains information about following :
#     1. Source instance/port driving each RAM switch instance enable pin and the instance name.
#     2. All RAMs with RAM instances on PD basis.
#
##############################################################################
#     Format for 1(above):
#     This report is shown in following form. The columns of report are
#=============================================================================================================
# Switch-MasterName        Switch-InstanceName                  AON   Driver-MasterName      Driver-Instan
#=============================================================================================================
LL_SLOGICSWITCH20              top.mid_inst.switch2           0       INT_BUF          top.inst2.inst
LL_SLOGICSWITCH20              top.mid_inst.Pswitch1          0       INT_BUF          top.inst1.inst

##############################################################################
#     Format for 2(above):
#     This report is shown in following form. The columns of report are
#=============================================================================
# Power-DomainName        Memory-MasterName         Memory-InstanceName
#=============================================================================
       PD1                     INT_PD          top.mid_inst.term_inst1
       PD1                     INT_PD          top.mid_inst.term_inst2
```

FIGURE 62. Sample lp_vd_info report generated by the LPSVM46 rule

Report Generated by the LPPLIB12 Rule

The *LPPLIB12* rule reports each RAM switch/RAM instance pair with RAM switch enable pin source cell name.

A sample lp_vd_info report is as follows:

lp_vd_info report

Save Print

```
################################################################################
#
# This file has been generated by SpyGlass:
#     Report Created by: ajain
#     Report Created on: Wed Aug 26 16:13:59 2009
#     Working Directory: /delsoft/ajain/views/dev_4_3_0/lowpower/test/Verilog/LPPLIB12/case10
#     Report Location  : .//spyglass_reports/lowpower/lp_vd_info.rpt
#     SpyGlass Version : 4.3.0-FCS-C1
#     Policy Name      : lowpower(4.3.0)
#     Comment          : Generated by rule LPPLIB12
#
################################################################################
#     Purpose:
#     This report contains information about following :
#     1. Source instance/port driving each RAM switch instance enable pin and the instance name.
#     2. All RAMs with RAM instances on PD basis.
#
################################################################################
#     Format for 1(above):
#     This report is shown in following form. The columns of report are
#=============================================================================================================
# Mem-Master      Mem-Instance  Sw-Master            Sw-Instance            AON     Sw-Driver-Master      S
#=============================================================================================================

-                     -       LL_SLOGICSWITCH20    top.mid_inst.Pswitch1    0       -       -

-                     -       LL_SLOGICSWITCH10    top.mid_inst.Pswitch2    1       -       in1
```

FIGURE 63. Sample lp_vd_info report generated by the LPPLIB12 rule

The report contains the following details:

- Source instance/port driving each RAM switch instance enable pin and the instance name.
- All RAMs with RAM instances on power domain basis.

The LPPLIB12 rule also generates the lp_ram_switch_info report. This report contains the information about RAM switches in the design.

Report Generated by the vdPDInfo Rule

The *vdPDInfo* rule prints the voltage domain and power domain information for each design unit.

A sample lp_vd_info report is as follows:

lp_vd_info report

Save Print

```
################################################################################
#
# This file has been generated by SpyGlass:
#     Report Created by: ajain
#     Report Created on: Wed Aug 26 16:33:23 2009
#     Working Directory: /delsoft/ajain/views/dev_4_3_0/lowpower/test/Verilog/vdPDInfo/case1
#     Report Location  : .//spyglass_reports/lowpower/lp_vd_info.rpt
#     SpyGlass Version : 4.3.0-PCS-CI
#     Policy Name      : lowpower(4.3.0)
#     Comment          : Generated by rule vdPDInfo
#
################################################################################

################################################################################
#
# This report contains following information about design
# VOLTAGE AND POWER DOMAIN COUNT
# GATE-COUNT INFORMATION FOR THE VOLTAGE DOMAINS
# and states about all instances belonging to each voltage domain and power domain

################################################################################
# Format:
#     The design level report is shown in following format.
#     VOLTAGE AND POWER DOMAIN COUNT
#     Design <design name> has <voltage-domain count> Voltage Domains and <power-domain count> Power Domains
#     INSTANCES BELONGING TO VOLTAGE/POWER DOMAIN
#     Voltage Domain <voltage-domain name> Instances
#     GATE-COUNT INFORMATION FOR THE VOLTAGE DOMAINS
#     Voltage Domain <voltage-domain name>     GateCount : <Gate-count>
#     Percentage of Design <design-percent>
################################################################################

VOLTAGE AND POWER DOMAIN COUNT

Design (CKT_TOP) has [0] Voltage Domains and [1] Power Domains.

INSTANCES BELONGING TO VOLTAGE/POWER DOMAINS

In Voltage Domain [V1],
      Instances : (CKT_TOP)

GATE-COUNT INFORMATION FOR THE VOLTAGE DOMAINS (Presence of Black-Box will give inaccurate results)

For Voltage Domain [V1],
      GateCount : 14
      Percentage of Design : 0.19
```

FIGURE 64. Sample lp_vd_info report generated by the vdPDInfo rule

The report contains the following details:

- Voltage and power domain count
- Gate count information for voltage domains

- Lists all instances belonging to each voltage and power domain

Report Generated by the LPPLIB17 Rule

The *LPPLIB17* rule reports information about power switches on switched supply and their enable. A sample lp_vd_info report for LPPLIB17 rule is as follows:

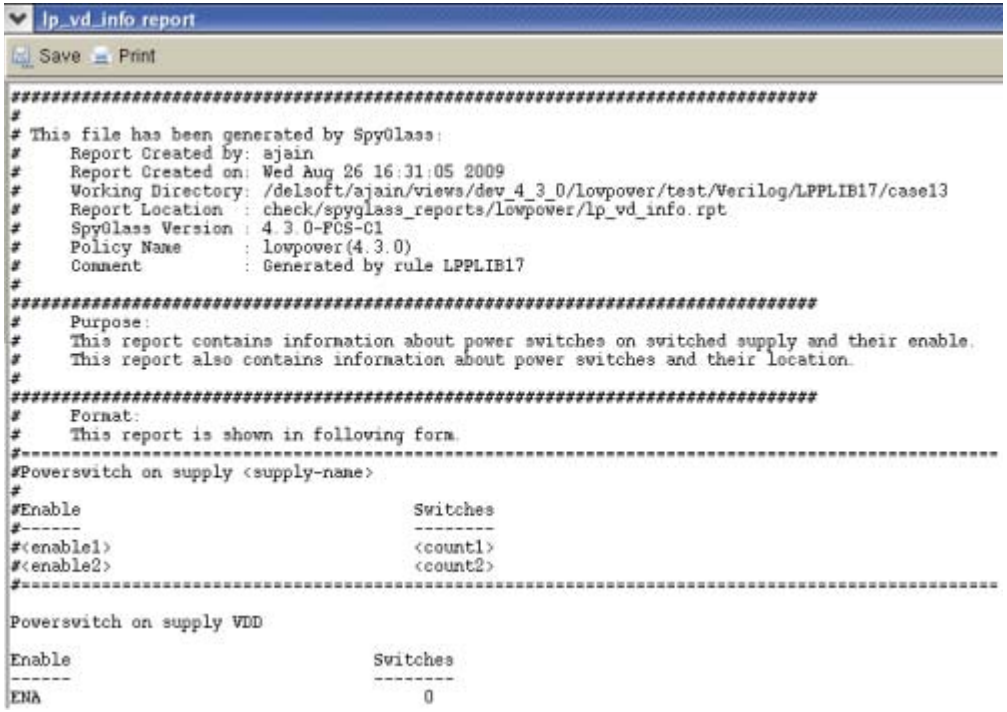


FIGURE 65. Sample lp_vd_info report generated by the LPPLIB17 rule

The report displays the following details (if the LPPLIB17 rule is run):

- Power switches on switched supply and their enable.
- Power switches and their location.

lp_wild_card

Report Generated by the LpWildCardMatchReport Rule

The *LpWildCardMatchReport* rule provides the matches found for the cell names specified in the SGDC constraints with wildcards (* or ?) in the lp_wild_card report.

A sample lp_wild_card report is as follows:

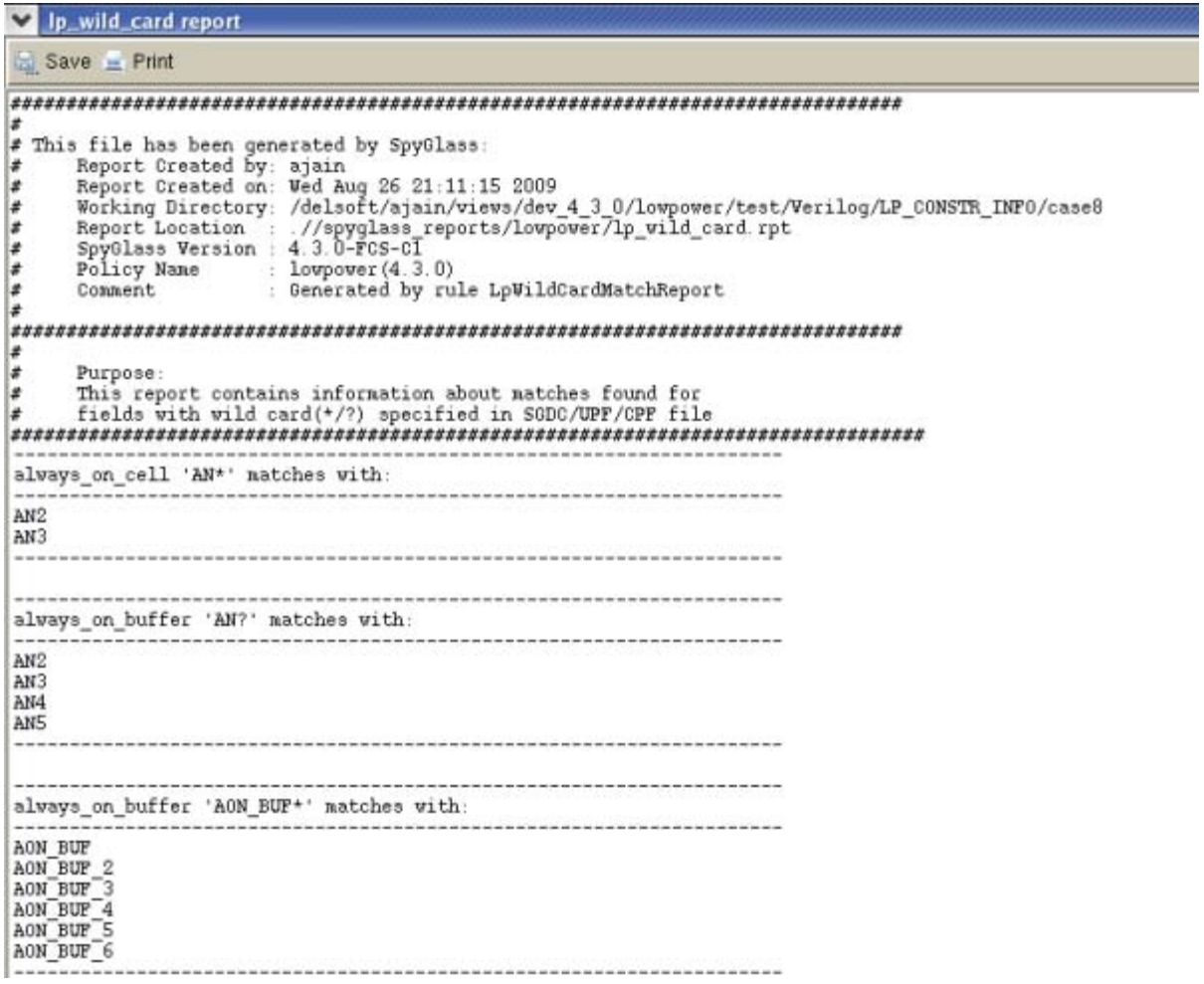


FIGURE 66. Sample lp_wild_cardreport

The LP-report Report

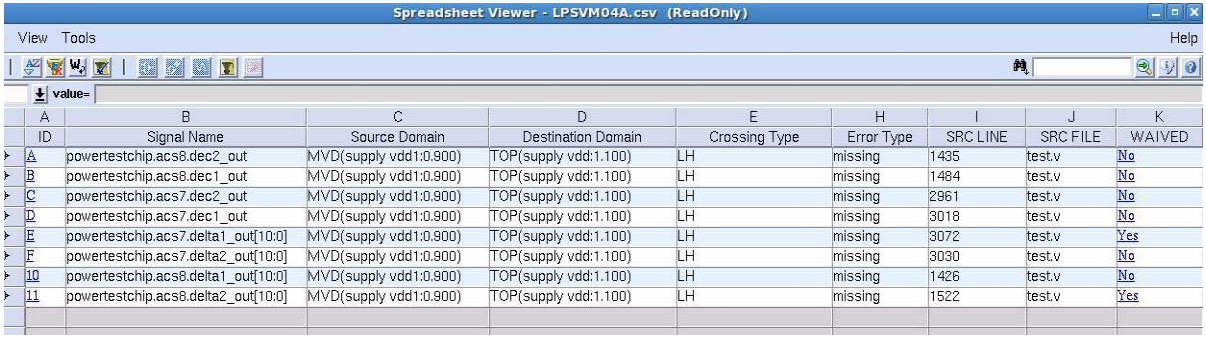
LP-report is generated in SpyGlass standard format (LP-report.rpt) when you specify `set_option report LP-report` in the project file, the following comma-separated data files are created. The following table states the rules that generate this report.

Rule	File Name	Description
LPSVM60	LPSVM60.csv	Contains messages of LPSVM60 in csv form
LPSVM22	LPSVM22.csv	Contains messages of LPSVM22 in csv form
LPSVM08A	LPSVM08A.csv	Contains messages of LPSVM08A in csv form
LPSVM04A	LPSVM04A.csv	Contains messages of LPSVM04A in csv form
LPSVM04B	LPSVM04B.csv	Contains messages of LPSVM04B in csv form
LPSVM04C	LPSVM04C.csv	Contains messages of LPSVM04C in csv form
LPSVM04D	LPSVM04D.csv	Contains messages of LPSVM04D in csv form
LPSVM04E	LPSVM04E.csv	Contains messages of LPSVM04E in csv form
LPLSH04	LPLSH04.csv	Contains messages of LPSVM04E in csv form
LPLSH03	LPLSH03.csv	Contains messages of LPLSH03 in csv form
LPISO05A	LPISO05A.csv	Contains messages of LPISO05A in csv form
LPISO05B	LPISO05B.csv	Contains messages of LPISO05B in csv form
LPISO04A	LPISO04A.csv	Contains messages of LPISO04A in csv form
LPISO04B	LPISO04B.csv	Contains messages of LPISO04B in csv form
LPISO04C	LPISO04C.csv	Contains messages of LPISO04C in csv form
LPSVM08B	LPSVM08B.csv	Contains messages of LPSVM08B in csv form
LPSVM08C	LPSVM08C.csv	Contains messages of LPSVM08C in csv form
LP_ISO_REPORT	lp_isolation_spreadsheet.csv	Contains isolation related information in csv form
LP_LSH_REPORT	lp_levelshifter_spreadsheet.csv	Contains level shifter related information in csv form

NOTE: *When the related information is not available, the corresponding comma-separated data file is not generated.*

The following image displays an example of the csv file is generated by the

LPSVM04A rule.



	A	B	C	D	E	H	I	J	K
	ID	Signal Name	Source Domain	Destination Domain	Crossing Type	Error Type	SRC LINE	SRC FILE	WAIVED
▸	A	powertestchip.acs8.dec2_out	MVD(supply vdd1:0.900)	TOP(supply vdd:1.100)	LH	missing	1435	test.v	No
▸	B	powertestchip.acs8.dec1_out	MVD(supply vdd1:0.900)	TOP(supply vdd:1.100)	LH	missing	1484	test.v	No
▸	C	powertestchip.acs7.dec2_out	MVD(supply vdd1:0.900)	TOP(supply vdd:1.100)	LH	missing	2961	test.v	No
▸	D	powertestchip.acs7.dec1_out	MVD(supply vdd1:0.900)	TOP(supply vdd:1.100)	LH	missing	3018	test.v	No
▸	E	powertestchip.acs7.delta1_out[10:0]	MVD(supply vdd1:0.900)	TOP(supply vdd:1.100)	LH	missing	3072	test.v	Yes
▸	F	powertestchip.acs7.delta2_out[10:0]	MVD(supply vdd1:0.900)	TOP(supply vdd:1.100)	LH	missing	3030	test.v	No
▸	10	powertestchip.acs8.delta1_out[10:0]	MVD(supply vdd1:0.900)	TOP(supply vdd:1.100)	LH	missing	1426	test.v	No
▸	11	powertestchip.acs8.delta2_out[10:0]	MVD(supply vdd1:0.900)	TOP(supply vdd:1.100)	LH	missing	1522	test.v	Yes

FIGURE 67. Sample LP-report report

lp_strategy_info

The lp_strategy_info report contains the list of valid isolation and/or level shifter strategies defined in the UPF file. In addition, this report contains the expanded list of elements to which that strategy is applied.

After specifying multiple general and element specific strategies on a design element, the SpyGlass Power Verify solution uniquifies these strategies and applies only one strategy on a design element. You can use this report to identify the strategy applied by the SpyGlass Power Verify solution on that design element.

NOTE: *This report is generated only in the UPF flow.*

```
##################################################
#    Purpose:
#    This report contains the list of Strategies (Isolation/Level Shifter)
#    defined in UPF along with the expanded list of elements to which
#    that strategy is applied.
##################################################
##################################################
##################################################
#                  Isolation Strategies defined in UPF                  #
##################################################
--------------------------------------------------
Strategy Name: ISO_TOP
Element List:
TOP/out1[0]
TOP/out1[1]
TOP/out2[0]
TOP/out2[1]
--------------------------------------------------
Strategy Name: ISO_A_OUT
Element List:
TOP/u1/out1[0]
TOP/u1/out1[1]
##################################################
#                Level Shifter Strategies defined in UPF                #
##################################################
--------------------------------------------------
Strategy Name: shift_down
Element List:
TOP/l1/out1
```

lp_multi_supply_instance

The lp_multi_supply_instance report lists the multi-supply instances that do not have an associated *connect_supply_net* command. This report is generated after you run the *UPF_lowpower15* rule. Use this report to ensure every multi-supply cells are handled correctly in the UPF file by associating them with *connect_supply_net* command. Therefore, the behavior of the design is as desired.

A sample report is as follows:

```
#################################################################
# Purpose:
# This report contains multi supply instances with missing
# connect_supply_net # command.
#################################################################

#################################################################
# Section I: Cells with "is_macro: true" attribute
#################################################################

#################################################################
# Section II: Always On Cells
#################################################################
#Master cell = HS45_LS_ONBFISOX18; Power Domain =  TOP
 connect_supply_net <supply_net> -ports top/inst2/gndo
 connect_supply_net <supply_net> -ports top/inst2/vddo
#Master cell = HS45_LS_ONBFISOX9; Power Domain =  TOP
 connect_supply_net <supply_net> -ports top/sph_c3883/gndo

#################################################################
# Section III: Cells which do not have "drc_blockgroup:
# stdcell" set
#################################################################
```

lp_special_pin_connection

The lp_special_pin_connection report is generated by the *LP_SPECIAL_PIN_CONNECTION* rule. The report shows the primary port connection to special pins.

A sample report is as follows:

```
#############################################################################
#     Purpose:
#     This report shows the primary port connection to special pins.
#############################################################################
#
```

```
#     Format:
#     The design level report is shown in the following information.
#
#     Primary Port/Net        : <port-name/net-name>
#
#     Cell                    : <cell-name>
#
#     Pin                     : <pin-name>
#
#     Domain                  : <domain-name>
#
##############################################################################
Port/Net Name    | CellName   | PinName          | Domains
 Port : top.en1  | PS1        | SLEEP            | VD1
 Port : top.en1  | PS2        | SLEEP            | VD2
 Net  : top.w2   | PS3        | SLEEP            | VD1
 No connection   | PS1        | top.psw0.SLEEP1 | VD1
```

Rules in SpyGlass Power Verify

Rule Severity Classes

The rules in the SpyGlass Power Verify solution have been classified under the SpyGlass pre-defined rule severity classes as follows:

Rule Severity Class	Contains the Rule Severity Labels
Fatal	Fatal
Recommended, Mandatory, Prohibited	Warning
Information	Info

See the *SpyGlass Console Reference Guide* for more information about SpyGlass pre-defined rule severity classes.

Rule Categories

The rules in the SpyGlass Power Verify solution cover different aspects of the power verification of a design. These rules have been organized in rule groups based on their usage or type.

Refer to the *Appendix: List of Rules With Applicable Design Stages* to see the design stages applicable to each rule.

The SpyGlass Power Verify solution provides the following rule types:

- *Level Shifter Rules*
- *Isolation Logic Rules*
- *Always-on Logic Rules*
- *State Retention Rules*
- *Connection Rules*
- *Supply Rules*
- *Automatic Fix Rules*
- *Fine Grain Power Gate (MTCMOS) Rules*
- *Best Design Practices*
- *Special Purpose Rules*
- *Detailed Reporting Rules*
- *Constraints Checking Rules*
- *Electrical Checks Rules*
- *CPF Check Rules*
- *UPF Check Rules*
- *Debug Rules*
- *SoC Abstraction Rules*

NOTE: *Currently, all the LPFSM, LPGLT, LPBUS, and LPXFM rules are not run by default, in the SpyGlass Power Verify solution. Therefore, you need to explicitly run these checks as per your requirements. Refer to the Best Design Practices section for detailed description of these rules.*

Level Shifter Rules

The Level Shifter group of rules is as follows:

Rule	Reports
LPPLIB04	Level shifters which are connected to improper supply signals
LPPLIB05	Level shifters where the power supply of a connected cell is different from the corresponding power supply of the level shifter
LPPLIB07	Level shifters with improper ground connections
LPPLIB08	Level shifters where the gate feeding the enable terminal of the level shifter is operating at the lower of the two voltage domains of the level shifter
LPLSH01	Level shifter cell not defined properly
LPLSH02	Level shifter cell, where the input/output voltage range does not match with the voltage range of the source/destination domains, respectively
LPLSH03	Excluded signal of power domain crossing that requires a level shifter
LPLSH04	Crossing signal, where a level shifter strategy is given but level shifter is not required
LPLSH05	Reports missing and incorrect level shifter strategies for voltage difference crossings
LPLSH05A	Reports missing and incorrect level shifter strategies for low-to-high voltage difference crossings
LPLSH05B	Reports missing and incorrect level shifter strategies for high to low voltage difference crossings
LPLSH06	Checks level shifter compatibility with main rail of cell
LPLSH07	Checks the voltage range of level shifter cells with reference to source and destination domains
LPLSH08	Checks the level shifter cell location with respect to the std_cell_main_rail attribute at the RTL level
LPSVM04	Level-shifters at voltage domain crossings
LPSVM04A	Missing level shifter from a voltage domain crossing from lower voltage domain to higher voltage domain
LPSVM04B	Missing level shifter from a voltage domain crossing from higher voltage domain to lower voltage domain

Rule	Reports
LPSVM04C	Inadvertently used level shifters
LPSVM04D	Incorrectly placed level shifters based on power format files specification
LPSVM04E	Incorrectly placed level shifters based on *std_cell_main_rail* library attribute specification.
LPSVM17	Multiple instances of the same level-shifter
LPSVM24	Level shifters where the enable pin does not belong to an always-on voltage domain
LPLIB_check 04	Checks input/output voltage range of level-shifter cells

LPPLIB04

Reports level shifters which are connected to improper supply signals

When to Use

In power managed designs, each voltage domain and each power domain requires a separate supply net. Therefore, it is important to make sure that each gate in the post-route design is connected to the correct supply. Use this rule for supply net checking in the Post-layout design phase.

This rule is recommended for use with *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)* and *DEF files and their associated LEF files*.

Description

The *LPPLIB04* rule checks for the supply connection for level shifters in the Post-layout design phase. The following list contains the criteria for level shifter supply connection:

- Level shifters must be connected to two supply voltages, one on each side.
  - The supply net connected on the input supply pin of level shifter should be same as the supply net that is connected to the supply pin of the instance on the source side.
  - The supply net connected on the output supply pin of level shifter should be same as the supply net that is connected to the supply pin of the instance on the destination side.
- If there is a single power pin, it should be connected to the higher of the two supply values.
- The rule also checks for power pins other than input supply pin and output supply pin. They should be connected to the supply net corresponding to the location of level shifter.

NOTE: *A cell with the is_level_shifter library attribute set to* `true` *is recognized as a level shifter cell.*

Prerequisites

Specify the following information before running this rule:

- Information of level shifter and its input supply pin and output supply pin
- Power domain information and its associated supply net information

Reported Information

The *LPPLIB04* rule reports the level shifter instances where:

- They are not connected to the same power supply signals as those of the connected cells.

  The connected cells exclude the instances in which the corresponding master cell definition is not found in the PLIB/LEF files, the master cell definition does not have any power pin, or the master cell has been defined as a level shifter.
- The single power pin is not connected to the higher of the two supply values of the connected cells.
- The level shifter cell has two power pins but either one or both are not a signal net instead of a supply (power) rails.
- The input pin is undriven, provided the *lp_flag_undriven_nets* parameter is set.
- The output pin is unconnected, provided the *lp_flag_unconnected_nets* parameter is set.
- Incorrect supply net connected with the input and output supply pin.
- Unconnected or undriven nets are present at level shifter input/output.

By default, the *LPPLIB04* rule checks that the supply pin of single supply level shifter is connected to the higher side supply connected to input or output side.

NOTE: *The LPPLIB07 rule performs similar checks for ground signals.*

Language

Verilog, VHDL, DEF

Rule Exceptions

This rule does not check for bias pins. The *LPPLIB18* rule checks for bias pins.

Parameter(s)

- *lp_flag_undriven_nets*: Default value is `0`. Set this parameter to `1` to consider undriven nets which are ignored by default. Other possible values are `no` and `yes`.
- *lp_flag_unconnected_nets*: Default value is `0`. Set this parameter to `1` to consider unconnected nets which are ignored by default. Other possible values are `no` and `yes`.
- *lp_ignore_same_voltage_error*: Default value is `0`. Set this parameter to 1 to ignore checking supplies having different names but the same value. Other possible values are `no` and `yes`.
- *lp_skip_buf*: Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.
- *lp_single_supply*: Default value is `high`. Set this parameter to `output` to check that the supply pin of single supply level shifter is connected to the output side.
- *lp_max_viol_count*: Default value is `1000`. Set this parameter to a positive integer number to specify the maximum number of violations that should be reported by the rule.
- *lp_check_ls_supply_at_rtl*: Default value is `0`. Set this parameter to 1 to checks for inconsistency between PG pins of level shifter cells and source/sink domain primary supplies in a partial PG netlist design.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage/ power domains in the design.
- *levelshifter* (Mandatory): Use this constraint to specify the names of design units to be used as level shifters.
- *supply* (Mandatory): Use this constraint to specify the supply and ground port names for all LPPLIB rules.
- *pg_cell* (Optional): Use this constraint to specify the names of power/ ground pins for cells in the input netlist, which are missing from the library cells defined in the PLIB/LIB/LEF libraries.

CPF Commands

- *create_power_nets* (Mandatory)
- *create_ground_nets* (Mandatory)
- *create_power_domain* (Mandatory)
- *update_power_domain* (Mandatory)
- *create_level_shifter_rule* (Mandatory)
- *define_level_shifter_cell* (Mandatory)
- *update_level_shifter_rules* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_level_shifter* (Mandatory)
- *map_level_shifter_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when the pin <*pin-name*> of level shifter <*ls-name*> instantiated as instance <*inst-name*> is missing from the associated PLIB files:

[LPPLIB04_1][Recommended] Pin '<pin-name>' is missing in PLIB files for macro <ls-name> instantiated as '<inst-name>'

For information on debugging, click *How to Debug and Fix*.

Message 2

The following message appears when the supply pin <*pin-name*> of level

shifter <*ls-name*> instantiated as instance <*inst-name*> is not connected to a supply port:

[LPPLIB04_2][Recommended] Pin '<pin-name>' of macro <ls-name> instantiated as '<inst-name>' is not connected to supply port

For information on debugging, click *How to Debug and Fix*.

Message 3

The following message appears when the input-side supply pin <*in-pin-name*> of the level shifter <*ls-name*> instantiated as instance <*inst-name*> is connected to a power supply signal different from the power supply signal of the cell instance at the input:

[LPPLIB04_3][Recommended] Incorrect Power Connection for in-supply terminal '<in-pin-name>' (supply <supply-name>:<value>) of Level Shifter (<ls-name>) instantiated as '<inst-name>'w.r.t. source <terminal/portname> (supply <supply-name>)

For information on debugging, click *How to Debug and Fix*.

Message 4

The following message appears when the output-side supply pin <*out-pin-name*> of the level shifter <*ls-name*> instantiated as instance <*inst-name*> is connected to a power supply signal does not match the power supply signal of the cell instance at the output:

[LPPLIB04_4][Recommended] Incorrect Power Connection for out-supply terminal '<out-pin-name>' (supply <supply-name>:<value>) of Level Shifter (<ls-name>) instantiated as '<inst-name>'w.r.t. source <terminal/portname> (supply <supply-name>)

For information on debugging, click *How to Debug and Fix*.

Message 5

The following message appears when the <*term-type*> (input or output) terminal of level shifter <*ls-name*> instantiated as instance <*inst-name*> is tied to constant <value> (0 or 1):

[LPPLIB04_5][Recommended] <term-type> terminal <term-name> of Level Shifter <*ls-name*> instantiated as <*inst-name*> is tied to constant <value>.

For information on debugging, click *How to Debug and Fix*.

Message 6

The following message appears when <term-type> (input or output) terminal <term-name2> of level shifter <*ls-name*> instantiated as instance <*inst-name*> is connected to different supply <supply-name2> other than supply <supply-name1> connected to <term-name1> terminal of <supply-term-type> (in-supply/out-supply):

[LPPLIB04_6][Recommended] Mismatch in Power Connection for <supply-term-type> terminal <term-name1> (connected to <supply-name1>) and <term-type> terminal <term-name2> (connected to <supply-name2>) of Level Shifter (<ls-name>) instantiated as <inst-name>

For information on debugging, click *How to Debug and Fix*.

Message 7

The following message appears when the power connection for the single supply terminal <*term-name*> of the level shifter <*ls-name*> instantiated as <*inst-name*> is incorrect (lower of the two supply values):

[LPPLIB04_7][Recommended] Incorrect Power Connection for single supply terminal '<term-name>' (supply <supply-name>:<value>) of Level Shifter (<ls-name>) instantiated as '<inst-name>'w.r.t. source/destination <terminal/portname> supply (<supply-name>:<value>)

For information on debugging, click *How to Debug and Fix*.

Message 8

The following message appears when the biased power pin <*bias-pin-name*> of the level shifter <*ls-name*> instantiated as instance <inst-name> is connected to a wrong supply <*supply-name1*>:

[LPPLIB04_8][Recommended] Power pin '<bias-pin-name>' of Level Shifter (<ls-name>) instantiated as '<inst-name>' is connected to supply '<supply-name1>' instead of '<supply-name2>'

For information on debugging, click *How to Debug and Fix*.

Message 9

The following message appears when <*pin-type*> pin of the level shifter

<ls-name> instantiated as instance *<inst-name>* is connected to ground supply *<gnd-supply-name>*:

[LPPLIB04_9][Recommended] Power pin '<pin-name>' of Level Shifter (<ls-name>) instantiated as '<inst-name>' is connected to ground supply '<gnd-supply-name>'

Where, *<pin-type>* can be input side supply pin, output side supply pin, or bias power pin.

Message 10

The following message appears when unconnected or undriven nets are present at the level shifter *<ls-name>* input/output:

[LPPLIB04_10][Recommended] Incorrect connection for '<in/out/single>'-supply terminal '<term-name>' of Level Shifter (ls-name) instantiated as '<inst-name>'.

For information on debugging, click *How to Debug and Fix*.

Message 11

An informational message appears when the violation count of this rule exceeds the limit set by the *lp_max_viol_count* parameter. Refer to *Message 5* for the message and, for debugging information, refer to *How to Debug and Fix*.

Potential Issues

The input or output supply pin of the level shifter is either unconnected or connected to a signal net or connected to an incorrect supply net.

If the supply name is not stated, which is possible in SGDC format, only the value of the supply is stated in Messages 3, 4, and 7.

Consequences of Not Fixing

The level shifter cell may not work if the voltage level of the supply nets connected to the input and output supply pin is incorrect.

How to Debug and Fix

The violation message states that either the input or the output power supply pin of a level shifter is hanging or is connected to the wrong supply net. The message for an incorrect connection appears when there is a mismatch between the supply net connected to the input or output power pin of a level shifter and the power pin of the instance connected to its input or output signal pin, respectively.

The violation message is reported at the place where the level shifter with incorrect power connection is instantiated.

For a graphical view of the violation, double-click the message and then click **Incremental Schematic**. Turn on Power View. The schematic shows the following:

- the power pin of the level shifter and its connection to supply net (with value)
- the connection from the input or output pin of the level shifter to the connected instance
- the power pin of the instance connected to the level shifter and its connection to supply net (with value)

To resolve this violation, ensure that the input/output power pins of all the instances are properly connected to the power supply net associated with the source/sink instance.

Example Code and/or Schematic

The SpyGlass Design Constraints file is as follows:

```
current_design top
supply -name VDD12 -value 1.2
supply -name VDD10 -value 1.0
supply -name GND -value 0.0
supply -name GNDA -value 0.0
voltage_domain -name Vtop -value 1.2 -modname top -supplyname
VDD12 GND
voltage_domain -name VA -value 1.0 -instname top.ua \
    -supplyname VDD10 GND -portname i1 i2
levelshifter -name LS10_12 -from VA -to Vtop -inTerm I -
outTerm O \
    -inSupplyTerm V10 -outSupplyTerm V12
```

The following violation appears when the input-side supply pin of a level shifter is connected to a power supply signal VDD with voltage value 1.6. This voltage value is different from the power supply signal VDDEXT of the input cell instance BH1_BUF:

```
Incorrect Power Connection for in-supply terminal 'VDD' of
Level Shifter (R120_ASLSIL120X160) instantiated as 'top.m1.LS1'
```

The incremental schematic is displayed as shown below:

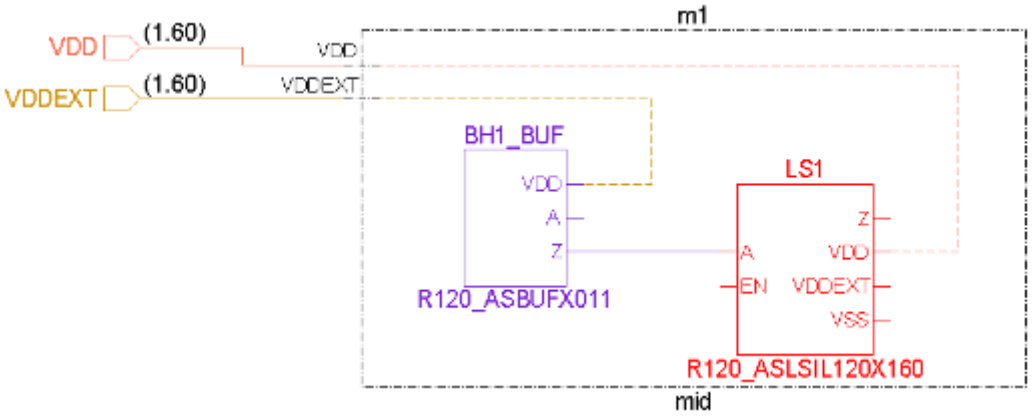


FIGURE 1. Incremental schematic

Default Severity Label

Recommended

Rule Group

LayoutPowerConnectivityRules

Reports and Related Files

- *lp_constr_info*: Reports information of the power domain and associated supplies.
- *lp_supply_connection*: Reports information of the supply pins connection to the supply net.

LPPLIB05

Reports cells connected to level shifters that have incorrect power connections

When to Use

Use this rule for:

- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPPLIB05* rule reports level shifters where the power supply of a connected cell is different from the corresponding power supply of the level shifter.

The *LPPLIB05* rule reports the following:

- The level shifter instances where the power supply of the previous cell connected to the level shifter is the same as the source power supply specified for the level shifter.
- The level shifter instances where the power supply of the next cell connected to the level shifter is the same as the destination power supply specified for the level shifter.
- The level shifter instances where the input pin is not driven, provided the *lp_flag_undriven_nets* parameter is set.
- The level shifter instances where the output pin is not connected, provided the *lp_flag_unconnected_nets* parameter is set.

Prerequisites

The *LPPLIB05* rule requires you to specify the following:

- Level shifters using the *levelshifter* constraint with the `inTerm` and `enableTerm` arguments specifying the input terminal and the enable terminal, respectively
- Supply and Ground port names using the *supply* constraint
- Supply and Ground port names using the *supply* constraint, for each voltage domain using the `supplyname` argument of the *voltage_domain* constraint

- (Optional) Power/Ground pins using the *pg_cell* constraint to provide details of Power/Ground pins for the cells that are not in the PLIB/LEF files

Rule Exceptions

The *LPPLIB05* rule cannot be run in the UPF/CPF power format because this rule is a duplicate of the LPSVM04 rule. The LPSVM04 rule has support for the UPF/CPF power format.

In addition, while inferring the connected cells, the *LPPLIB05* rule ignores instances where the corresponding master cell definition is not found in the PLIB/LEF files, the master cell definition does not have any power pins, or the master cell has been defined as a level shifter.

Languages

Verilog, VHDL

Parameter(s)

- *lp_flag_undriven_nets*: Default value is 0. Set the *lp_flag_undriven_nets* parameter to 1 report the input pins that are not driven.
- *lp_flag_unconnected_nets*: Default value is 0. Set the *lp_flag_unconnected_nets* parameter to 1 consider output pins that are not connected.
- *lp_max_viol_count*: Default value is 1000. Set this parameter to a positive integer number to specify the maximum number of violations that should be reported by the rule.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/power domains in the design.
- *levelshifter* (Mandatory): Use to specify the names of design units to be used as level shifters.
- *supply* (Mandatory): Use to specify the supply and ground port names for all the LPPLIB rules.

- *pg_cell* (Optional): Use to specify the names of power/ground pins for cells present in the input netlist, which are missing from the PLIB/LIB/LEF libraries.

Messages and Suggested fix

Message 1

The following message appears when the voltage domain <*vd-name*> of the previous cell connected to instance <*inst-name*> of the level shifter <*ls-name*> does not match any of the voltage domain(s) specified using the `from` argument in the *levelshifter* constraint defining this level shifter:

[LPPLIB05_1][RECOMMENDED] Incorrect 'from' domain(s) for Level Shifter (<lsname>) instantiated in '<inst-name>' shifting from Voltage Domain <vd-name>

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the voltage domain of the previous cell connected to instance <*inst-name*> of the level shifter <*ls-name*> matches one of the voltage domain(s) specified using the `from` argument in the *levelshifter* constraint defining this level shifter, but the voltage domain <*vd-name*> of the next cell connected does not match any of the voltage domain(s) <*vd-name-list*> specified using the corresponding `to` argument:

[LPPLIB05_2][RECOMMENDED] Incorrect 'to' domain(s) <vd-name-list> for Level Shifter (<lsname>) instantiated in '<inst-name>' shifting to Voltage Domain <vd-name>

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when the voltage domain of the previous cell connected to instance <*inst-name*> of the level shifter <*ls-name*> does not match one of the voltage domains specified using the `from` argument in the *levelshifter* constraint defining this level shifter, and the voltage domain of the next cell does not match one of the voltage domains <*destination-vd-name*> specified using the `to` argument:

[LPPLIB05_3][RECOMMENDED] Incorrect 'from' and 'to' domain for

```
Level Shifter <lsname> instantiated as '<inst-name>' shifting
from voltage domain ‘<source-vd-name>' to Voltage Domain
'<destination-vd-name>
```

For debugging information, click *How to Debug and Fix*.

Message 4

An informational message appears when the violation count of this rule exceeds the limit set by the *lp_max_viol_count* parameter. Refer to *Message 5* for the message and, for debugging information, refer to *How to Debug and Fix*.

Potential Issues

The violation messages explicitly states the potential issues.

Consequences of Not Fixing

The consequences of not fixing are as follows:

- **Message 1**: As the voltage domain at the source side of the level shifter differs from the voltage domain provided in the `from` field of the level shifter strategy, the level shifter strategy becomes redundant and of no use.
- **Message 2**: As the voltage domain at the destination side of the level shifter differs from the voltage domain provided in the `to` field of the level shifter strategy, the level shifter strategy becomes redundant and of no use.
- **Message 3**: As the voltage domain at the source and destination side of the level shifter differs from the voltage domain provided in the `from` and `to` fields of the level shifter strategy, the level shifter strategy becomes redundant and of no use.

How to Debug and Fix

The violation is reported at the place where the level shifter is instantiated with incorrect power connections.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. The schematic shows the following information:

To fix these violation messages, ensure the:

- voltage domain of the previous cell and the next cell connected to a level shifter matches with any of the voltage domain(s) specified using the `from` and the `to` argument of the *levelshifter* constraint, which defines the level shifter respectively.
- input pins are driven
- output pins are connected

Example Code and/or Schematic

In this example, a violation occurs when the voltage domain `'V1'` of the previous cell connected to the level shifter '`R120_ASLSIL120X160`' does not match any of the voltage domain(s) specified using the `from` argument in the `levelshifter` constraint defining this level shifter. The following message appears:

```
Incorrect 'from' domain for Level Shifter (R120_ASLSIL120X160)
instantiated as 'top.m1.LS1' shifting from Voltage Domain 'V1'
```

The schematic is as follows:

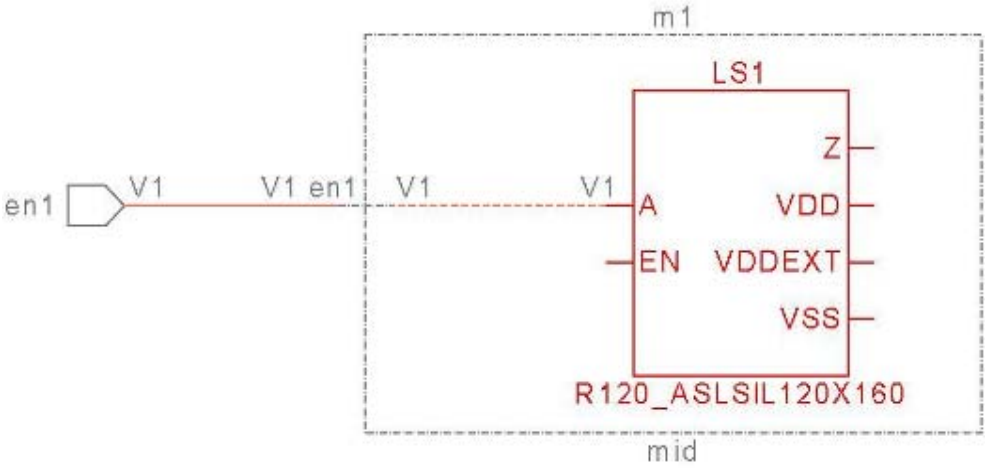


FIGURE 2. Incremental schematic

Default Severity Label

Recommended

Rule Group

Level Shifter

Reports and Related Files

None

LPPLIB07

Reports improper ground connection of the level shifters

When to Use

Use this rule for:

- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPPLIB07* rule reports a violation if the input/output terminal of a level shifter is connected to a ground supply that does not match with the ground connection of the ground terminal(s) of the level shifter.

For level shifters with a single ground pin, this rule reports a violation if the ground pin of the level shifter is neither connected to the ground rail of the previous cell (input cell) nor connected to the ground rail of the next cell (output cell).

This rule also checks for level shifter bias ground pins.

Prerequisites

To use the *LPPLIB07* rule, you need to specify:

- Level shifters using the *levelshifter* constraint with the `inTerm` and `enableTerm` arguments specifying the input terminal and the enable terminal respectively
- Ground port names using the *supply* constraint
- Ground port names for each voltage domain using the `supplyname` argument of the *voltage_domain* constraint
- (Optional) Power/Ground pins using the *pg_cell* constraint to provide details of Power/Ground pins for the cells not in the specified PLIB/LEF files

Rule Exceptions

The *LPPLIB07* rule ignores instances where master cell definition is not found in the PLIB/LEF files, the master cell definition does not have any ground pins, or the master cell defined as a level shifter. Then, this rule

checks the connections of both ground pins.

The *LPPLIB07* rule does not perform the checks on power signals.

Language

Verilog, VHDL

Parameters

- *lp_is_common_ground*: Default value is `1`. Set this value to `0` to indicate that the level shifters have two ground pins.
- *lp_flag_undriven_nets*: Default value is `0`. Set this value to `1` to check the input pins that are not driven.
- *lp_flag_unconnected_nets*: Default value is 0. Set this value to `1` to check the unconnected output pins.
- *lp_ignore_same_voltage_error*: Default value is `0`. Set this value to `1` to ignore checking the supplies having different name but same value.
- *lp_max_viol_count*: Default value is `1000`. Set this parameter to a positive integer number to specify the maximum number of violations that should be reported by the rule.\

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/power domains.
- *levelshifter* (Mandatory): Use to specify the names of design units to be used as level shifters.
- *supply* (Mandatory): Use to specify the supply and ground port names for all the *LPPLIB* rules.
- *pg_cell* (Optional): Use to specify the names of power/ground pins for cells pin the input netlist, and missing from the respective PLIB/LIB/LEF libraries.

CPF Commands

- *create_power_nets* (Mandatory)
- *create_ground_nets* (Mandatory)
- *create_power_domain* (Mandatory)

- *update_power_domain* (Mandatory)
- *create_level_shifter_rule* (Mandatory)
- *define_level_shifter_cell* (Mandatory)
- *update_level_shifter_rules* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_level_shifter* (Mandatory)
- *map_level_shifter_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when the ground pin <*pin-name*> of the level shifter <*ls-name*> is connected to the power supply <*pwr-sup-name*>:

[LPPLIB07_8][RECOMMENDED] Ground pin '<pin-name>' of Level Shifter (<ls-name>) instantiated as '<inst-name>' is connected to power supply '<pwr-sup-name>'

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the level shifter <*ls-name*> instantiated as instance <*inst-name*> is connected to a different ground signal from the ground signal of the next cell:

[LPPLIB07_3][RECOMMENDED] Incorrect Ground Connection for Ground Pin of destination domain of Level Shifter (<ls-name>)

instantiated as '<inst-name>'

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when the level shifter <*ls-name*> instantiated as instance <*inst-name*> is connected to a different ground signal from the ground signal of the previous cell:

[LPPLIB07_4][RECOMMENDED] Incorrect Ground Connection for Ground Pin of source domain of Level Shifter (<ls-name>) instantiated as '<inst-name>'

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears when the level shifter <*ls-name*> instantiated as instance <*inst-name*> is connected to a different ground signal from the ground signals of both the previous cell and the next cell:

[LPPLIB07_5][RECOMMENDED] Incorrect Ground Connection for Ground Pins of source domain and destination domain of Level Shifter (<lsname>) instantiated as '<inst-name>'

For debugging information, click *How to Debug and Fix*.

Message 5

The following message appears when the level shifter master macro <*duname*> of instance <*inst-name*> does not have two ground pins:

[LPPLIB07_1][RECOMMENDED] Missing ground pin(s) in PLIB/LEF files for macro <duname> instantiated as '<inst-name>'

For debugging information, click *How to Debug and Fix*.

Message 6

The following message appears when the ground pin of the level shifter instance <*inst-name*> (master macro <*duname*>) is not connected:

[LPPLIB07_2][RECOMMENDED] Missing ground pin(s) connections for macro <duname> instantiated as '<inst-name>'

For debugging information, click *How to Debug and Fix*.

Message 7

The following message appears when the ground pin <*pin-name*> of level shifter <*cell-name*> instance <*inst-name*> is not connected to any supply (ground) rail:

[LPPLIB07_6][RECOMMENDED] Ground pin '<pin-name>' of level shifter '<inst-name>(<cell-name>)' is not connected to supply net

For debugging information, click *How to Debug and Fix*.

Message 8

The following message appears when the <*term-type*> (input/output) terminal <*term-name*> (connected to supply <*supply-name*>) of level shifter <*ls-name*> instantiated as instance <*inst-name*> does not match with the ground terminal(s) connection <*gnd-conn-list*> of the level shifter:

[LPPLIB07_7][RECOMMENDED] Mismatch in Ground Connection for <term-type> terminal '<term-name>' (connected to '<supply-name>') and ground terminal(s) <gnd-conn-list> of Level Shifter (<ls-name>) instantiated as '<inst-name>'

<*gnd-conn-list*> can be expanded as:

- "<*gnd-name1*> (connected to '<*gnd-supply1*>')", when the level shifter has a single ground terminal
- "<*gnd-name1*> (connected to '<*gnd-supply1*>') and <*gnd-name2*> (connected to '<*gnd-supply2*>')", when the level shifter has two ground terminals

For debugging information, click *How to Debug and Fix*.

Message 9

The following message appears when the biased ground pin <*bias-pin-name*> of the level shifter <*ls-name*> instantiated as instance <*inst-name*> is connected to a wrong supply <*supply-name1*>:

[RECOMMENDED] Ground pin '<bias-pin-name>' of Level Shifter (<ls-name>) instantiated as '<inst-name>' is connected to supply '<supply-name1>' instead of '<supply-name2>'

For debugging information, click *How to Debug and Fix*.

Message 10

An informational message appears when the violation count of this rule exceeds the limit set by the *lp_max_viol_count* parameter. Refer to *Message 5* for the message and, for debugging information, refer to *How to Debug and Fix*.

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The consequences of not fixing are:

- **Message 1**: Since the ground pin is connected to the power supply, it is a design flaw.
- **Message 2**: The ground pins of the level shifter and the destination cell are not connected to the same ground supply. This is a design flaw.
- **Message 3**: The ground pins of the source cell and the level shifter are not connected to the same ground supply. This is a design flaw.
- **Message 4**: The ground pins of the source cell and the destination cell are not connected to the same ground supply. This is a design flaw.
- **Message 5**: The ground pins of level shifter cells are missing in PLIB/LEF files. This is a design flaw.
- **Message 6**: The ground pins of level shifter cells are left unconnected. This is a design flaw.
- **Message 7**: The ground pins of level shifter cells are not connected to any ground supply rail. This is a design flaw.
- **Message 8**: The ground pins of level shifter cells are not connected to the correct ground supply rail. This is a design flaw.
- **Message 9**: The biased ground pin of level shifter cells is connected to the wrong ground supply rail. This is a design flaw.

How to Debug and Fix

The violation is reported at the place where the level shifter is instantiated with incorrect/missing ground connections.

The message indicates the input or output ground supply pin of a level shifter is either hanging or connected to the wrong supply net. The message for incorrect connection appears when there is mismatch between the supply net connected to input/output ground pin of a level shifter and

ground pin of the instance connected to its input/output signal pin, respectively.

View the incremental schematic of the violation message. Turn on the Power View. The schematic shows:

- the ground pin of level shifter and its connection to ground supply net
- the connection from the input/output pin of level shifter to the connected instance
- the ground pin of the instance connected to the level-shifter and its connection to ground supply net

To view a sample of the schematic generated by this rule message, refer to the Example Code and/or Schematic section.

To fix these violation messages,

- Check that the level shifter is not connected to a ground signal different from the ground signal of the next cell or the previous cell.
- Check that the level shifter master macro should have two ground pins (if the *lp_is_common_ground* parameter is set as 0) or the ground pins should be connected to a supply (ground) rail.
- Check the input pins that are not driven.
- Check the unconnected output pins.

Example Code and /or Schematic

In this example, a violation occurs when the level shifter master macro 'R120_ASLSIL120X160' of instance 'top.m1.BH1_BUF6' does not have correct ground connections. The following message appears:

```
Incorrect Ground Connection for Ground Pins of source domain
and destination domain of Level Shifter (R120_ASLSIL120X160)
instantiated as 'top.m1.BH1_BUF6'
```

The schematic is as follows:

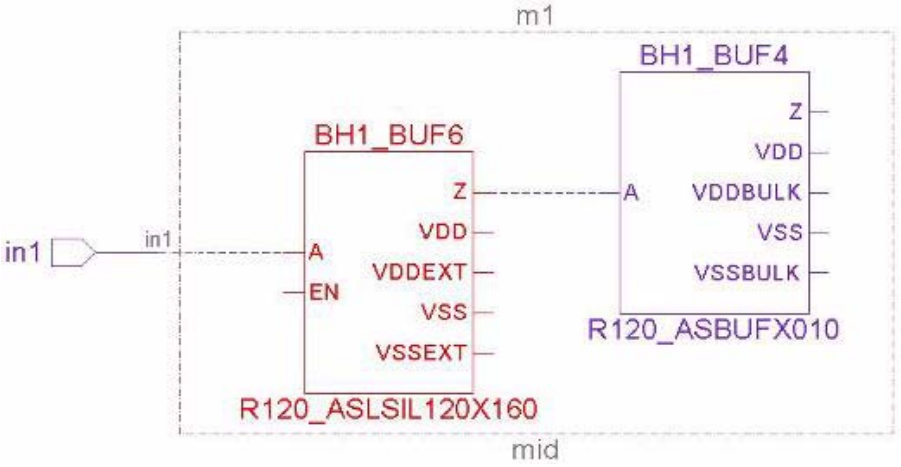


FIGURE 3. Incremental schematic

The schematic shows the following:

- Instances of level shifters where the ground pin is not connected to either of the ground rails.
- Instances of level shifters where the master definition is not found in the specified PLIB/LEF files
- Instances of level shifters where the master cell does not have two ground pins, when the *lp_is_common_ground* parameter is set to 0.
- Instances of level shifters where the master cell has two ground pins but one or both pins are not connected to supply (ground) rails.
- Instances of level shifters where the input pin is not driven provided the *lp_flag_undriven_nets* parameter is set.
- Instances of level shifters where the output pin is not connected when the *lp_flag_unconnected_nets* parameter is set.
- Instances of level shifters where the ground pin is not connected.

Default Severity Label

Recommended

Rule Group

Level Shifter

Reports and Related Files

None

LPPLIB08

Reports gates of enable signal of level shifters, supplied with a different voltage domain

When to Use

Use this rule for:

- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The LPPLIB08 rule reports level shifters, where the gate feeding the enable terminal is operating at a voltage domain different from the voltage domain of the level shifter enable pin.

Prerequisites

The *LPPLIB08* rule requires to specify the following:

- Level shifters using the *levelshifter* constraint with the `inTerm` and `-enableTerm` arguments specifying the input terminal and the enable terminal respectively.
- Ground port names using the *supply* constraint.
- Ground port names (specified using the *supply* constraint) for each voltage domain using the `supplyname` argument of the *voltage_domain* constraint.
- (Optional) Power/Ground pins using the *pg_cell* constraint to provide details of Power/Ground pins for the cells that are not in the specified PLIB/LEF files.

Language

Verilog, VHDL

Parameters

- *lp_ignore_same_voltage_error*: Default value is `0`. Set this value to `1` to ignore checking the supplies having different name but same value.

- *lp_max_viol_count*: Default value is 1000. Set this parameter to a positive integer number to specify the maximum number of violations that should be reported by the rule.

Constraints

- *voltage_domain* (Mandatory): Use to specify the voltage/power domains and its information is used by *SP_01* rule.
- *levelshifter* (Mandatory): Use to specify the names of design units to be used as level shifters.
- *supply* (Mandatory): Use to specify the supply and ground port names for all the *LPPLIB* rules.
- *pg_cell* (Optional): Use to specify the names of power/ground pins for cells pin the input netlist, and missing from the respective PLIB/LIB/LEF libraries.

Messages and Suggested Fix

Message 1

An informational message appears when the violation count of this rule exceeds the limit set by the *lp_max_viol_count* parameter. Refer to *Message 5* for the message and, for debugging information, refer to *How to Debug and Fix*.

Message 2

The following message appears when gate connected to the enable pin <*pin-name*> of the level shifter <*ls-name*> instantiated as instance <*inst-name*> has voltage domain <*en-domain*> instead of <*domain*>:

[LPPLIB08_2][RECOMMENDED] Gate connected to enable Pin '<pin-name>' of Level Shifter (<ls-name>) instantiated as '<inst-name>' has voltage domain '<en-domain>'(<en-voltage-value>) instead of '<domain>'(<voltage-value>)

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not Fixing

The enable pin of the level shifter does not belong to the same domain as

that of its related power pin. This is a design issue.

How to Debug and Fix

The violation is reported at the place where the level shifter is instantiated with enable signal connected to voltage domain different from that of its related power pin.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. The schematic shows the following information:

To fix this violation message, check that the gates of enable pin of level shifters are supplied with the same voltage domain as its related power pin.

Example Code and /or Schematic

Example 1

Consider the following example,

```
voltage_domain -name V1 -value 1.2 -instname \ mid.t1.inst1
supplyname VDD VSS

voltage_domain -name V2 -value 1.6 -instname \ mid.t1.inst4
supplyname VDDEXT VSSEXT

supply -name VDD -value 1.2

supply -name VDDEXT -value 1.6

supply -name VSS -value 0

supply -name VSSEXT -value 0'
```

Level shifters needs to be specified for voltage domains, `V1` and `V2` using the *levelshifter* constraint. The name of the enable terminal must be specified using the `enableTerm` argument and the name of the input signal terminal with the `-inTerm` argument, as follows:

```
levelshifter -name R120_ASLSIL120X160 -from VDD1 -to VDDEXT -
inTerm A -enableTerm EN'
```

Example 2

A violation occurs when the enable pin of the level shifter `R120_ASLSIL120X160` instantiated as instance `'top.m1.LS1'` has

voltage domain V1 (1.20) instead of V2 (1.60). The following message appears:

```
Gate connected to enable Pin 'EN' of Level Shifter
(R120_ASLSIL120X160) instantiated as 'top.m1.LS1' has voltage
domain 'V1' (1.20) instead of 'V2' (1.60)
```

The schematic is as follows:

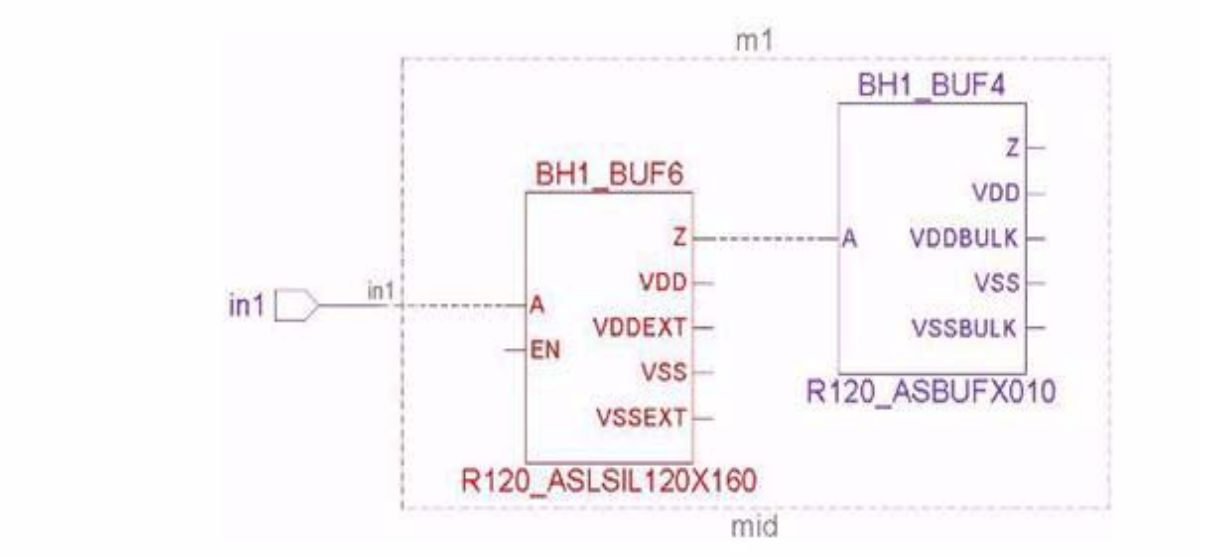


FIGURE 4. Incremental schematic

Default Severity Label

Recommended

Rule Group

Level Shifter

Reports and Related Files

None

LPLSH01

Reports level shifter cells that are not defined properly

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPLSH01* rule reports a violation if the *is_level_shifter* library attribute is not set for any level shifter cell used with the *levelshifter* / *map_level_shifter_cell* command. In the CPF flow, this rule reports a violation if any level shifter cell is not defined with the *define_level_shifter_cell* command.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/power domains in the design.
- *levelshifter* (Mandatory): Use to specify the names of design units to be used as level shifters.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_level_shifter_rule* (Mandatory)
- *create_nominal_condition* (Mandatory)

- *create_power_mode* (Mandatory)
- *update_level_shifter_rules* (Optional)
- *define_level_shifter_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_level_shifter* (Mandatory)
- *map_level_shifter_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when the level shifter cell *<cellname>* does not have the *is_level_shifter* library attribute set:

```
[LPLSH01_1][WARNING] Cell '<cell-name>' specified as level
shifter, does not have 'is_level_shifter' attribute in the
library cell description
```

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears in the CPF flow when the level shifter cell *<cell-name>* is not defined with the *define_level_shifter_cell* command:

```
[LPLSH01_2][WARNING] Cell '<cell-name>' specified in '-cells'
option of 'update_level_shifter_rules' command, is not
specified with 'define_level_shifter_cell' command
```

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The consequences of not fixing the violations are as follows:

Message 1: The level shifter cell specified in the power intent does not have the *is_level_shifter* attribute set in the library. This makes the constraint redundant.

Message 2: In the CPF flow, the level shifter cell specified as in the '-cells' argument of the *update_level_shifter_rules* constraint has not been specified by using the *define_level_shifter_cell* constraint in the power intent. This makes the constraint redundant.

How to Debug and Fix

The violation is reported at the place in power intent file, where the level shifter (not properly defined) is used.

To fix these violations, define level shifter cells in your design with the *is_level_shifter* library attribute or the *define_level_shifter_cell* command in the CPF flow.

Example and /or Schematic

UPF

```
set_level_shifter LS1 -domain UA -applies_to outputs
map_level_shifter_cell LS1 -lib_cells BASIC_LS -domain UA
```

LIBRARY

```
cell (BASIC_LS) {
   level_shifter_type : LH;
   input_voltage_range (0.8, 0.6);
   output_voltage_range (1.0, 0.9);
   pg_pin (VDDA) { pg_type : backup_power; }
   pg_pin (VDDY) { pg_type : primary_power; }
   pg_pin (VSS) { pg_type : primary_ground; }
   pin (A) { direction : input; related_power_pin : VDDA; }
   pin (Y) { direction : output; function : "A";
   related_power_pin : VDDY; }
}
```

As the attribute *is_level_shifter* is missing. A violation is reported.

Default Severity Label

Warning

Rule Group

Level Shifter

Reports and Related Files

None

LPLSH02

Reports voltage range mismatch between level shifter cell and source/destination domains

When to Use

NOTE: *The LPLSH02 rule will be deprecated in future release. It is recommended to use the LPLSH07 rule instead.*

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPLSH02* rule reports a mismatch between the input/output voltage range of a level shifter cell with the voltage range of the source/destination domain, respectively.

This rule checks the following:

- **For UPF**: When an invalid level shifter is specified in *map_level_shifter_cell* only.
- **For CPF**: When a level shifter strategy is specified without mapping it to any cell in the library. For UPF, use the *LPLSH07* rule.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/power domains and its information is used by *SP_01* rule.

- *levelshifter* (Mandatory): Use to specify the names of design units to be used as level shifters.

CPF Commands

- *create_power_domain* (Mandatory): Use to create a power domain.
- *create_level_shifter_rule* (Mandatory): Use to create a rule for adding a level shifter.
- *create_nominal_condition* (Mandatory): Use to infer voltage values for domains in a power state.
- *create_power_mode* (Mandatory): Use to create a power mode.
- *update_level_shifter_rules* (Optional): Use to update the details of a level shifter rules.
- *define_level_shifter_cell* (Mandatory): Use to identify the library cells in the .lib files that can be used as level shifter cells.

UPF Commands

- *create_power_domain* (Mandatory): Use to create power domain.
- *create_supply_port* (Mandatory): Use to create supply port.
- *add_port_state* (Mandatory): Use to state a port.
- *create_supply_net* (Mandatory): Use to create a supply net.
- *connect_supply_net* (Mandatory): Use to connect a supply net with a supply port.
- *set_domain_supply_net* (Mandatory): Use to specify a primary power and ground supply nets for a power domain.
- *set_level_shifter* (Mandatory): Use to specifies a level shifter strategy.
- *create_pst* (Optional): Use to create a Power State Table (PST) by using a specific order of supply nets.
- *add_pst_state* (Optional): Use to define the states of each of the supply nets for one possible state of the design.

Messages and Suggested Fix

Message 1

The following message appears for an invalid level shifter:

```
[LPLSH02_1][WARNING] Cell '<cell-name>[voltage-range]'
```

specified as level shifter from domain '<src-domain>[voltage-range]' to domain '<dest-domain>[voltage-range]' is not valid : <reason>

Where, <reason> can be one of the following:

- Input and output voltage range of level shifter does not match with the voltage range of the from and to domains
- Input voltage range of level shifter does not match with the voltage range of the from domain
- Output voltage range of level shifter does not match with the voltage range of the to domain

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

The consequences of not fixing the violations are as follows:

- The voltage value of the to/destination domain of the level shifter should lie in the voltage value range specified in the library.
- The voltage value of the from/source domain of the level shifter should lie in the voltage value range specified in the library.

If either of these conditions is not met, there is a design flaw.

How to Debug and Fix

The violation is reported at the place, where the level shifter (with incorrect specification) is specified.

To fix the violation, check that the input/output voltage range of a level shifter cell matches with the voltage range of source/destination domain, respectively.

Message 2

The following message appears when you specify a level shifter strategy, without mapping it to any cell, in CPF and SGDC:

[LPLSH02_2][WARNING] There is no valid level shifter in the library whose input/output voltage range lies between the voltage range from domain <src-domain> to domain <dest-domain>

Potential Issues

For CPF, if you did not use the *update_level_shifter_rules* command to specify a level shifter strategy, the rule checks the existence of any level shifter present in `availableLSCellList` that can be used between the voltage domains. If no level shifter is identified, this violation message is reported.

Consequences of not Fixing

There is no level shifter provided in the library whose input and output voltage ranges match the voltage values of the source and destination domains.

How to Debug and Fix

The violation is reported at the place where the level shifter strategy is specified using the *create_level_shifter_rule* command in CPF.

To fix this violation, ensure that the library specified has at least one level shifter cell that can be used between the specified voltage domains.

Example Code and/or Schematic

In this example, the level shifter is placed between the domains `HIGH` and `LOW`, at least one of which have the voltage value not in the ranges specified in the library. Therefore, the *LPLSH02* rule reports violations.

```
set_design TOP


set_cpf_version 1.0e
# power domain definitions
create_power_domain -name LOW  -instances .low_domain -
default
create_power_domain -name HIGH -instances .high_domain

create_power_nets -nets vddh -voltage 1.2
create_power_nets -nets vddl -voltage 0.8
create_ground_nets -nets vss -voltage 0.0

update_power_domain -name LOW -primary_power_net vddl -
primary_ground_net vss
update_power_domain -name HIGH -primary_power_net vddh -
primary_ground_net vss
```

```
# level shifter rules
create_level_shifter_rule -name LS1 -from HIGH -exclude {
.high_domain.out1 .high_domain.out2 }
update_level_shifter_rules -names LS1  -cells { ls1 ls2 } -
location to


create_level_shifter_rule -name LS2 -to HIGH -exclude {
.high_domain.in1 .high_domain.in2 }
update_level_shifter_rules -names LS2  -cells ls3  -location
from


create_level_shifter_rule -name LS3 -to HIGH -exclude {
.high_domain.in1 .high_domain.in2 }


#Incorrect: CPF input/output voltage range not accepted/
understood.
#Incorrect/Correct????: LS type(LH) taken in lp_lib_data,
wrong wrt voltage-range, correct wrt direction
define_level_shifter_cell -cells {ls1} -always_on_pins {A E
Y} -input_voltage_range {0.2 0.22} -output_voltage_range
{0.3 0.33} -direction up -valid_location either


#Incorrect: Pin D does not exist but is shown in lp_lib-data
define_level_shifter_cell -cells {ls2} -input_voltage_range
1.2 -output_voltage_range 0.3 -direction down -enable {A} -
valid_location to -always_on_pins {D}


define_level_shifter_cell -cells {ls3} -input_voltage_range
0.2 -output_voltage_range 1.2 -direction bidir -enable {E} -
valid_location from


define_level_shifter_cell -cells {LOW} -input_voltage_range
0.2 -output_voltage_range 1.2
define_level_shifter_cell -cells {HIGH} -input_voltage_range
```

```
0.2 -output_voltage_range 1.2
define_level_shifter_cell -cells {BASIC_BUF} -
input_voltage_range 0.2 -output_voltage_range 1.2
```

This rule reports the following violation messages:

[WARNING] Cell 'ls1[10.000:11.000-1.300:1.500]' specified as level shifter from domain 'HIGH[1.200]' to domain 'LOW[0.800]' is not valid : input and output voltage range of level shifter does not match with the voltage range of 'from' and 'to' domains

[WARNING] Cell 'ls2[1.200-0.300]' specified as level shifter from domain 'HIGH[1.200]' to domain 'LOW[0.800]' is not valid : output voltage range of level shifter does not match with the voltage range of 'to' domain

[WARNING] Cell 'ls3[0.200-1.200]' specified as level shifter from domain 'LOW[0.800]' to domain 'HIGH[1.200]' is not valid : input voltage range of level shifter does not match with the voltage range of 'from' domain

[WARNING] There is no valid level shifter in the library whose input/output voltage range lies between the voltage range from domain 'LOW' to domain 'HIGH'

To resolve the above violation messages, specify the voltage value within the range.

Default Severity Label

Warning

Rule Group

Level Shifter

Reports and Related Files

None

LPLSH03

Reports excluded signals having voltage difference which requires level shifter

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPLSH03* rule reports if a level shifter specified as excluded in the CPF/UPF power format files is required at a port/terminal of a domain crossing.

You can specify in the CPF/UPF flow whether to exclude a level shifter using either of the following options:

- The `-exclude` argument of the *create_level_shifter_rule* CPF command on a port/terminal of a domain crossing.
- The `-no_shift` argument of the *set_level_shifter* UPF command.

NOTE: *If the set_port_attribute -repeater_supply or set_repeater –repeater_supply_set is specified on a port in the UPF, then this rule assumes the presence of an implicit repeater at RTL and performs the checking accordingly.*

Language

Verilog, VHD**L**

Parameter(s)

- *lp_flag_undriven_nets*: Default value is `0`. Set this value to `1` to check the input pins that are not driven.
- *lp_flag_unconnected_nets*: Default value is `0`. Set this value to `1` to check the output pins that are not connected.

- *lp_skip_buf*: Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.
- *lp_skip_pwr_gnd*: Default value is `1`. Set this value to `0` to consider nets connected to the power/ground supply.
- *lp_use_voltage_map_value*: Default value is `0`. If you set the value of this parameter to `1`, the voltage value provided for the `voltage_map` attribute of source pin of a crossing is considered while calculating the level shifter requirement.
- *lp_skip_blackbox_checking*: Default value is `0`. Set the parameter to `1` to skip checking for black boxes.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/power domains and its information is used by *SP_01* rule.
- *levelshifter* (Mandatory): Use to specify the names of design units to be used as level shifters.

CPF Commands

- *create_power_domain* (Mandatory): Use to create a power domain.
- *create_level_shifter_rule* (Mandatory): Use to create a rule for adding a level shifter.
- *create_nominal_condition* (Mandatory): Use to infer voltage values for domains in a power state.
- *create_power_mode* (Mandatory): Use to create a power mode.
- *update_level_shifter_rules* (Optional): Use to update the details of a level shifter rules.
- *define_level_shifter_cell* (Mandatory): Use to identify the library cells in the .lib files that can be used as level shifter cells.

UPF Commands

- *create_power_domain* (Mandatory): Use to create power domain.
- *create_supply_port* (Mandatory): Use to create supply port.
- *add_port_state* (Mandatory): Use to state a port.

- *create_supply_net* (Mandatory): Use to create a supply net.
- *connect_supply_net* (Mandatory): Use to connect a supply net with a supply port.
- *set_domain_supply_net* (Mandatory): Use to specify a primary power and ground supply nets for a power domain.
- *set_level_shifter* (Mandatory): Use to specify a level shifter strategy.
- *map_level_shifter_cell* (Optional): Use to map a particular level shifter strategy to a library cell or range of library cells.
- *create_pst* (Optional): Use to create a Power State Table (PST) by using a specific order of supply nets.
- *add_pst_state* (Optional): Use to define the states of each of the supply nets for one possible state of the design.
- *set_repeater* (Optional)

Messages and Suggested Fix

Message 1

The following message appears if required <*lsh-type*> level shifter is given with the `no_shift` option in the UPF format.

[LPLSH03_1][WARNING] Crossing signal from '<src-domain>' (domain '<domain1>') to '<dest-domain>' (domain '<domain2>') require level shifter of type '<lsh-type>',is given '-no_shift' option in strategy <strategy-name>

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears if required <*lsh-type*> level shifter is given with the exclude option in CPF format:

[LPLSH03_2][WARNING] Crossing signal from '<src-domain>' (domain '<domain1>') to '<dest-domain>' (domain '<domain2>') require level shifter of type '<lsh-type>',is given '-exclude' option

For debugging information, click *How to Debug and Fix*.

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of not Fixing

Message 1: The strategy specified at the domain boundary in UPF flow has `-no_shift` option, even though, a level shifter is required in the crossing.

Message 2: The strategy specified at the domain boundary ports in CPF flow has `-exclude` option, even though, a level shifter is required in the crossing.

How to Debug and Fix

The violation is reported at the first place where the signal is used. This message supports UPF cross-probing. Therefore, in addition to the design file, the UPF files are displayed in the **File** tab.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

The *LPLSH03* rule also generates a file named LPLSH03.csv. This file contains all the messages generated by the *LPLSH03* rule. To view the spreadsheet, in the Message Tree window, click the Open Spreadsheet option in the right-click menu of the rule.

To fix the violations, ensure that a level shifter is not required at the signals excluded in the power intent files.

For **Message 1,** double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

Example Code and/or Schematic

Example 1

Consider the following example. A violation occurs when `no_shift` is specified for the signal `top.u2_inst.out1`:

```
Crossing signal from 'LOW(supply VDDL:0.95)' to 'HIGH(supply
VDDH:1.20)' require level shifter of type 'LH',is given '-
no_shift' option in strategy 'LS1'
```

The schematic is as follows:

FIGURE 5. Incremental schematic

Here, a level shifter is required in the crossing between the domains ‘LOW’ and ‘HIGH’, whereas in the power intent, the `-no_shift` option has been provided. Therefore, this rule reports a violation.

Example 2

Consider the following example. The following violation occurs when the `-exclude` option is specified for the `TOP.high_domain.out2` signal even when the crossing requires level shifting:

```
Crossing signal from 'TOP.high_domain.out2' (domain
'HIGH(supply vddh:1.200)') to 'TOP.out2' (domain 'LOW(supply
vddl:0.800)') require level shifter of type 'HL',is given '-
exclude' option
```

The schematic is as follows:

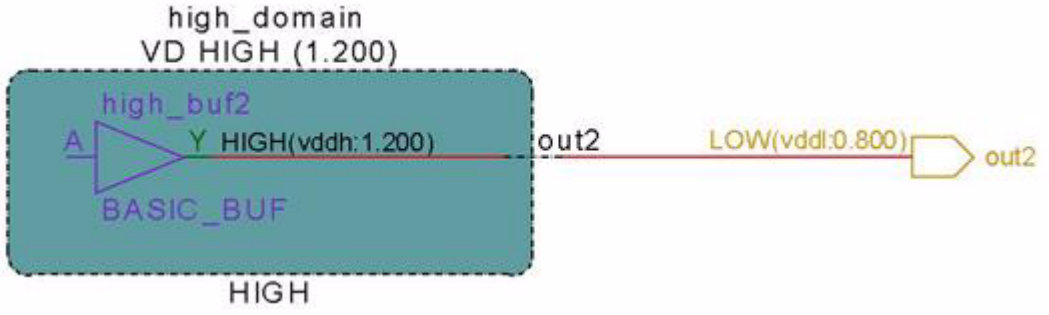


FIGURE 6. Incremental schematic

In the above schematic, a level shifter is required in the crossing between the `HIGH` and `LOW` domains, whereas in the power intent, the `-exclude` option has been provided. Therefore this rule reports the violation.

Default Severity Label

Warning

Rule Group

`Level Shifter`

Reports and Related Files

None

LPLSH04

Reports redundant level shifter strategy specified on a crossing, where either a level shifter is not required or more than one sufficient level shifter strategies are specified

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPLSH04* rule reports when a level shifter strategy is specified for a crossing, but a level shifter is not required at the domain crossing.

When there are multiple strategies on the path such that one of them is sufficient for the crossing, this rule reports other strategies as redundant strategies. All strategies which are applicable for the crossing, except the first applicable strategy, is reported as redundant.

NOTE: *If the set_port_attribute -repeater_supply or set_repeater –repeater_supply_set is specified on a port in the UPF, then this rule assumes the presence of an implicit repeater at RTL and performs the checking accordingly.*

Rule Exceptions

This rule does not run in CPF format.

Language

Verilog, VHDL

Parameter(s)

- *lp_flag_undriven_nets*: Default value is `0`. Set this value to `1` to check input pins that are not driven.
- *lp_flag_unconnected_nets*: Default value is `0`. Set this value to `1` to check output pins that are not connected.

- *lp_skip_buf*: Default value is 1 and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to 0 to consider SpyGlass-generated buffers during rule checking.
- *lp_skip_pwr_gnd*: Default value is 1. Set this value to 0 to consider nets connected to power/ground supply.
- *lp_check_same_voltage_path*: Default value is 0. By default, this rule does not report crossings from a domain to another domain where there is no voltage change. To report such crossings, set the lp_check_same_voltage_path to 1.
- *lp_use_voltage_map_value*: Default value is 0. If you set the value of this parameter to 1, the voltage value provided for the voltage_map attribute of source pin of a crossing is considered while calculating the level shifter requirement.
- *lp_skip_blackbox_checking*: Default value is 0. Set the parameter to 1 to skip checking for black boxes.

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory): Use to create power domain.
- *create_supply_port* (Mandatory): Use to create supply port.
- *add_port_state* (Mandatory): Use to state a port.
- *create_supply_net* (Mandatory): Use to create a supply net.
- *connect_supply_net* (Mandatory): Use to connect a supply net with a supply port.
- *set_domain_supply_net* (Mandatory): Use to specify a primary power and ground supply nets for a power domain.
- *set_level_shifter* (Mandatory): Use to specify a level shifter strategy.
- *map_level_shifter_cell* (Optional): Use to map a particular level shifter strategy to a library cell or range of library cells.
- *create_pst* (Optional): Use to create a Power State Table (PST) by using a specific order of supply nets.
- *add_pst_state* (Optional): Use to define the states of each of the supply nets for one possible state of the design.

- *set_repeater* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when a redundant strategy is present:

```
[LPLSH04_1][WARNING] Redundant strategy <strategy-name> present
in crossing from '<source>' (domain 'source-domain') to
'<destination>' (domain '<destination-domain>') : <reason>
```

The reason can be:

- No voltage difference between source and destination
- Valid strategy <strategy2-name> already present

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when an level shifter strategy *<strategy-name>* is specified for a crossing and either the source, destination, or both is an inout port/pin:

```
[LPLSH04_2][WARNING] Level shifter strategy '<strategy-name>'
specified at <source|destination> for a crossing between source
<source-hiername>'(domain '<domain-and-supply-name>') and
destination '<destinationhier-name>' (domain '<domain-and-
supply-name>')  is ignored as <source is |destination is |both
sides are> an inout <port|pin>
```

For debugging information, click *How to Debug and Fix*.

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

The reported strategy is redundant. Therefore, the UPF is not optimal.

How to Debug and Fix

The violation is reported at the place where the signal with a redundant level shifter strategy is used in the design file. This message supports UPF cross-probing. Therefore, in addition to the design file, the UPF files are displayed in the **File** tab.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

The *LPLSH04* rule also generates a file named LPLSH04.csv. This file contains all the messages generated by the *LPLSH04* rule. To view the spreadsheet, in the Message Tree window, click the Open Spreadsheet option in the right-click menu of the rule.

To fix the violations, remove the redundant strategies reported from the power intent files.

Double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

Example Code and/or Schematic

Consider the following example. A violation occurs when a level shifter strategy is specified on the source side of the crossing going from `top.u2_inst.out1` (HIGH) to `top.out1` (LOW):

```
[WARNING] Redundant strategy 'ls1' present in crossing from
'top.u2_inst.out1' (domain 'HIGH(supply VDDH:1.20)') to
'top.out1' (domain 'LOW(supply VDDL:1.20)') : No voltage
difference between source and destination
```

The schematic is as follows:

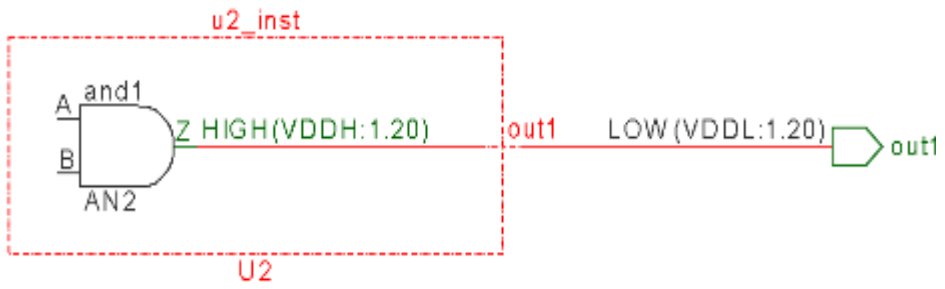


FIGURE 7. Incremental schematic

Here, the level shifter strategy has been provided at the domain boundary ‘u2_inst.out1’. As the voltage value of both the domains ‘HIGH’ and ‘LOW’

is the same, there is no requirement of a level shifter. Therefore, this rule reports a violation.

Default Severity Label

Warning

Rule Group

`Level Shifter`

Reports and Related Files

None

LPLSH05

Reports missing and incorrect level shifter strategies for voltage difference crossings

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The LPLSH05 rule checks for missing and incorrect level shifter strategies for voltage difference crossings.

This rule has the following subrules:

- *LPLSH05A*: Reports missing and incorrect level shifter strategies for low to high voltage difference crossings.
- *LPLSH05B*: Reports missing and incorrect level shifter strategies for high to low voltage difference crossings

NOTE: *If the set_port_attribute -repeater_supply or set_repeater –repeater_supply_set is specified on a port in the UPF, then this rule assumes the presence of an implicit repeater at RTL and performs the checking accordingly.*

LPLSH05A

Reports missing and incorrect level shifter strategies for low-to-high voltage difference crossings

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPLSH05A* rule reports a violation when a low-to-high level shifter strategy is not defined in crossings that have the source and sink functioning on different voltages.

Languages

Verilog, VHDL

Parameter(s)

- *lp_skip_buf*: Default value is `1`. Set the value to 0 to consider buffers generated during design synthesis.
- *lp_skip_pwr_gnd*: Default value is `1`. Set the value to 0 to consider nets connected to power/ground supply.
- *lp_flag_undriven_nets*: Default value is 0. Set the parameter to 1 to consider nets that are not driven at voltage crossings.
- *lp_flag_unconnected_nets*: Default value is 0. Set the parameter to 1 to consider unconnected nets at voltage crossings.
- *lp_flag_ls_strategy_for_multi_fanout*: Default value is `no`. Set the parameter to `yes` to report violation for common level shifter strategy applied on source, which is going to a mixed fanout.
- *lp_use_voltage_map_value*: Default value is `0`. If you set the value of this parameter to `1`, the voltage value provided for the `voltage_map`

attribute of source pin of a crossing is considered while calculating the level shifter requirement.

- *lp_check_pwr_gnd_to_macro_without_prd*: Default value is `yes`. Set this parameter to `no` to ignore crossings between tie-high/low and macro cell (`is_macro_cell: true`)/pad cell (`pad_cell: true`), without domain boundary. Level shifter and isolation checking for these crossings at the RTL and Netlist levels will be ignored.
- *lp_skip_blackbox_checking*: Default value is `0`. Set the parameter to `1` to skip checking for black boxes.

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_level_shifter* (Mandatory)
- *map_level_shifter_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)
- *set_repeater* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when you do not specify a level shifter on either source or destination domain boundary ports:

[LPLSH05A_1][WARNING] Crossing signal from low voltage '<src-name>'(domain '<src-domain>') to high voltage '<sink-name>'(domain '<dest-domain>') needs level shifter but does not have strategy of type '<ls-type>' on either source or destination domain boundary ports

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when you do not specify a level shifter and no boundary port is present in the path of the crossing signal:

[LPLSH05A_2][WARNING] Crossing signal from low voltage '<src-name>'(domain '<src-domain>') to high voltage '<dest-name>'(domain '<dest-domain>') needs level shifter but no domain boundary port is present in the path

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when you do not specify a level shifter and an incorrect type of strategy is defined:

[LPLSH05A_3][WARNING] Crossing signal from low voltage '<src-name>'(domain '<src-domain>') to high voltage '<dest-name>'(domain '<dest-domain>') needs level shifter but has strategy '<strategy-name>' with incorrect type '<ls-type>'

This message supports UPF cross-probing. Therefore, in addition to the design file, the UPF files are displayed in the **File** tab.

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears when you do not specify a level shifter and a strategy cannot be defined because all domain boundary ports in the path of the crossing signal are of the `inout` type:

[LPLSH05A_4][WARNING] Crossing signal from low voltage '<src-name>'(domain '<src-domain>') to high voltage '<dest-name>'(domain '<dest-domain>') needs level shifter but strategy cannot be defined as all domain boundary port(s) in the path is (are) of inout type

For debugging information, click *How to Debug and Fix*.

Message 5

The following message appears when crossing signal from low voltage to high voltage has invalid level shifter strategy specified at source for multiple fanouts with different domains:

[LPLSH05A_5][WARNING] Crossing signal from low voltage '<src-

name>'(domain ‘<domain-name>’) to high voltage '<destination1-name>’(domain '<domain-name>'), ‘<destination2-name>’(domain ‘<domain-name>’) has invalid level shifter strategy '<strategy-name>’ specified at source

This message is reported only when the *lp_flag_ls_strategy_for_multi_fanout* parameter is set to `yes`.

For debugging information, click *How to Debug and Fix*.

Potential Issues

The violation messages appear because:

- **Message 1:** The source and destination domain are working in different voltage levels but there is no level shifter strategy specified on either source or destination domain boundary ports.
- **Message 2:** The source domain to destination domain crossing needs a level shifter but no domain boundary port is present in the path where a level shifter strategy can be specified.
- **Message 3:** Incorrect level shifter strategy is specified in the voltage domain crossing.
- **Message 4**: The source domain to destination domain crossing needs a level shifter but all domain boundary port(s) in the path is (are) of inout type.
- **Message 5**: The strategy specified at source cannot be valid for multiple fanouts with different domains, at the same time.

Consequences of Not Fixing

Not resolving these violation messages can lead to a design failure.

How to Debug and Fix

The violation is reported at the place where the source signal of the crossing is used in the design file. Message 3 supports UPF cross-probing. Therefore, in addition to the design file, the UPF files are displayed in the **File** tab.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

The *LPLSH05* rule also generates a file named LPLSH05.csv. This file

contains all the messages generated by the *LPLSH05* rule. To view the spreadsheet, in the Message Tree window, click the Open Spreadsheet option in the right-click menu of the rule.

To fix these violations, provide a level shifter strategy/rule matching with the domain crossing. You should add the *set_level_shifter* UPF command with the appropriate level shifter cell provided with the *map_level_shifter_cell* UPF command, or change the type of crossing for the strategy.

Double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

Example Code and /or Schematic

Example 1

This example illustrates when *Message 1* and *Message 3* are reported. Consider the following UPF file snippet.

```
# power domain definitions

create_power_domain top_domain

create_power_domain VD_domain -elements { /insttop_VD }

create_power_domain PD_domain -elements { /insttop_PD }


create_supply_port top_supply -domain top_domain -direction
in

create_supply_port PD_supply -domain top_domain -direction
in

create_supply_port VD_supply -domain top_domain -direction
in


add_port_state top_supply -state {top_state 1.0}

add_port_state VD_supply -state {VD_state 1.1}

add_port_state PD_supply -state {PD_state 1.2} -state
{off_state off}
```

```
set_domain_supply_net top_domain -primary_power_net
top_supply -primary_ground_net VSS

set_domain_supply_net VD_domain -primary_power_net VD_supply
-primary_ground_net VSS

set_domain_supply_net PD_domain -primary_power_net PD_supply
-primary_ground_net VSS


# level shifter rules

set_level_shifter LS1 -domain PD_domain -applies_to outputs -
elements {/insttop_PD/iso2} -rule low_to_high
```

The violation messages reported by the LPLSH05A rule are:

```
[WARNING] Crossing signal from low voltage
'top.insttop_VD.out1'(domain 'VD_domain(supply
VD_supply:1.100)') to high voltage 'top.insttop_PD.in1'(domain
'PD_domain(supply PD_supply:0.900-1.200)') needs level shifter
but has strategy 'LS1' with incorrect type 'low_to_high'

[WARNING] Crossing signal from low voltage 'top.in1'(domain
'top_domain(supply top_supply:1.000)') to high voltage
'top.insttop_VD.in1'(domain 'VD_domain(supply
VD_supply:1.100)') needs level shifter but does not have
strategy of type 'low_to_high' on either source or destination
domain boundary ports
```

The schematic generated is as follows:

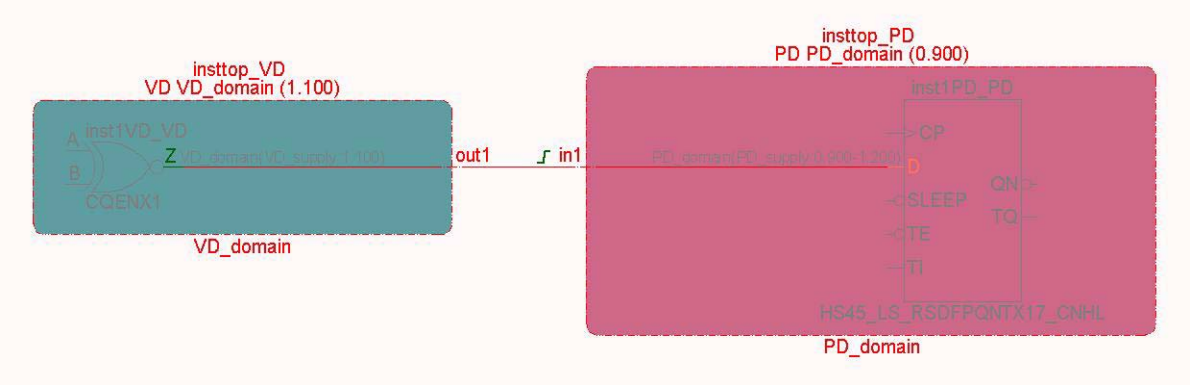


FIGURE 8. Incremental schematic

To fix these violations, provide a low-to-high level shifter strategy/rule matching with the domain crossing.

Example 2

Consider the following UPF file snippet.

```
set_level_shifter LS1 -domain top -applies_to inputs -rule
low_to_high -location parent
```

The following violation message is reported by the *LPLSH05A* rule because level shifter strategy `LS1` has been specified at source `top.in1` cannot be valid for both the destinations at the same time:

```
Crossing signal from low voltage 'top.in1'(domain 'top(supply
Vtop:1.100)') to high voltage 'top.e2.rtlc_I2.in1'(domain
'VD2(supply V_VD    2:1.300)'), 'top.e1.rtlc_I2.in1'(domain
'VD1(supply V_VD1:1.200)') has invalid level shifter strategy
'LS1' specified at source
```

The schematic generated is as follows:

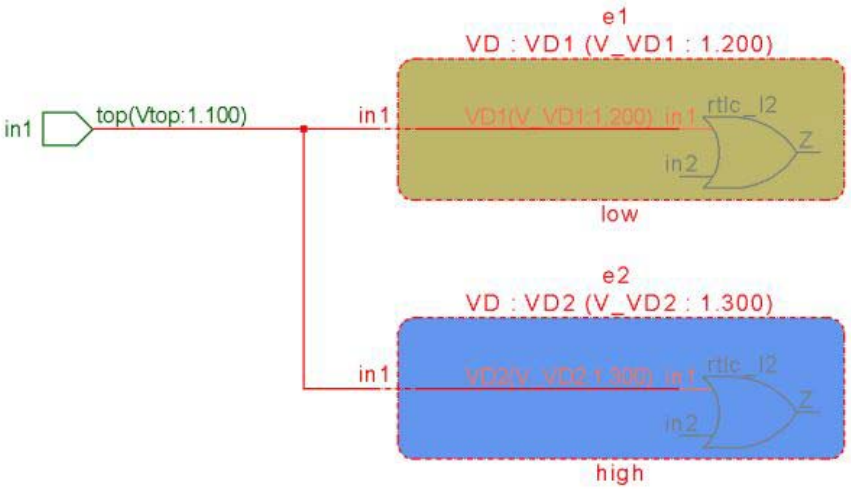


FIGURE 9. Incremental schematic

To fix this violation, provide level shifter strategies at the destinations.

Default Severity Label

Warning

Rule Group

`Level Shifter`

Reports and Related Files

None

LPLSH05B

Reports missing and incorrect level shifter strategies for high to low voltage difference crossings

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPLSH05B* rule reports a violation when a high-to-low level shifter strategy is not defined in crossings that have the source and sink functioning on different voltages.

Languages

Verilog, VHDL

Parameter(s)

- *lp_skip_buf*: Default value is `1`. Set the value to 0 to consider buffers generated during design synthesis.
- *lp_skip_pwr_gnd*: Default value is `1`. Set the value to 0 to consider nets connected to power/ground supply.
- *lp_flag_undriven_nets*: Default value is 0. Set the parameter to 1 to consider nets that are not driven at voltage crossings.
- *lp_flag_unconnected_nets*: Default value is 0. Set the parameter to 1 to consider unconnected nets at voltage crossings.
- *lp_flag_ls_strategy_for_multi_fanout*: Default value is `no`. Set the parameter to `yes` to report violation for common level shifter strategy applied on source, which is going to a mixed fanout.
- *lp_use_voltage_map_value*: Default value is `0`. If you set the value of this parameter to `1`, the voltage value provided for the `voltage_map`

attribute of source pin of a crossing is considered while calculating the level shifter requirement.

- *lp_check_pwr_gnd_to_macro_without_prd*: Default value is `yes`. Set this parameter to `no` to ignore crossings between tie-high/low and macro cell (`is_macro_cell: true`)/pad cell (`pad_cell: true`), without domain boundary. Level shifter and isolation checking for these crossings at the RTL and Netlist levels will be ignored.
- *lp_skip_blackbox_checking*: Default value is `0`. Set the parameter to `1` to skip checking for black boxes.

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_level_shifter* (Mandatory)
- *map_level_shifter_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)
- *set_repeater* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when you do not specify a level shifter on either source or destination domain boundary ports:

[LPLSH05B_1][WARNING] Crossing signal from high voltage '<src-name>'(domain '<src-domain>') to low voltage '<sink-name>'(domain '<dest-domain>') needs level shifter but does not have strategy of type '<ls-type>' on either source or destination domain boundary ports

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when you do not specify a level shifter and no boundary port is present in the path of the crossing signal:

[LPLSH05B_2][WARNING] Crossing signal from high voltage '<src-name>'(domain '<src-domain>') to low voltage '<dest-name>'(domain '<dest-domain>') needs level shifter but no domain boundary port is present in the path

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when you do not specify a level shifter and an incorrect type of strategy is defined:

[LPLSH05B_3][WARNING] Crossing signal from high voltage '<src-name>'(domain '<src-domain>') to low voltage '<dest-name>'(domain '<dest-domain>') needs level shifter but has strategy '<strategy-name>' with incorrect type '<ls-type>'

This message supports UPF cross-probing. Therefore, in addition to the design file, the UPF files are displayed in the **File** tab.

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears when you do not specify a level shifter and a strategy cannot be defined because all domain boundary ports in the path of the crossing signal are of the `inout` type:

[LPLSH05B_4][WARNING] Crossing signal from high voltage '<src-name>'(domain '<src-domain>') to low voltage '<dest-name>'(domain '<dest-domain>') needs level shifter but strategy cannot be defined as all domain boundary port(s) in the path is (are) of inout type

For debugging information, click *How to Debug and Fix*.

Message 5

The following message appears when crossing signal from high voltage to low voltage has invalid level shifter strategy specified at source for multiple fanouts with different domains:

[LPLSH05A_5][WARNING] Crossing signal from high voltage '<src-

name>'(domain ‘<domain-name>’) to low voltage '<destination1-name>’(domain '<domain-name>'), ‘<destination2-name>’(domain ‘<domain-name>’) has invalid level shifter strategy '<strategy-name>’ specified at source

This message is reported only when the *lp_flag_ls_strategy_for_multi_fanout* parameter is set to `yes`.

For debugging information, click *How to Debug and Fix*.

Potential Issues

The violation messages appear because:

- **Message 1:** The source and destination domain are working in different voltage levels but there is no level shifter strategy specified on either source or destination domain boundary ports.
- **Message 2:** The source domain to destination domain crossing needs a level shifter but no domain boundary port is present in the path where a level shifter strategy can be specified.
- **Message 3:** Incorrect level shifter strategy is specified in the voltage domain crossing.
- **Message 4**: The source domain to destination domain crossing needs a level shifter but all domain boundary port(s) in the path is (are) of inout type.
- **Message 5**: The strategy specified at source cannot be valid for multiple fanouts with different domains, at the same time.

Consequences of Not Fixing

Not resolving these violation messages can lead to a design failure.

How to Debug and Fix

The violation is reported at the place where the source signal of the crossing is used in the design file. Message 3 supports UPF cross-probing. Therefore, in addition to the design file, the UPF files are displayed in the **File** tab.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

The *LPLSH05* rule also generates a file named LPLSH05.csv. This file contains all the messages generated by the *LPLSH05* rule. To view the

spreadsheet, in the Message Tree window, click the Open Spreadsheet option in the right-click menu of the rule.

To fix these violations, provide a level shifter strategy/rule matching with the domain crossing. You should add the *set_level_shifter* command with the appropriate level shifter cell provided with the *map_level_shifter_cell* command, or change the type of crossing for the strategy.

Double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

Example Code and /or Schematic

Example 1

This example illustrates when *Message 1* and *Message 3* are reported. Consider the following UPF file snippet.

```
# power domain definitions
create_power_domain TOP -include_scope
create_power_domain LOW -elements u1_inst
create_power_domain HIGH -elements u2_inst

create_supply_port VDDH
add_port_state VDDH -state {on_state 1.4}

create_supply_port VDDL
add_port_state VDDL -state {on_state 1.2}

create_supply_port VTOP
add_port_state VTOP -state {on_state 1.3}

set_domain_supply_net LOW  -primary_power_net VDDL -
primary_ground_net VSS
set_domain_supply_net HIGH -primary_power_net VDDH -
```

```
primary_ground_net VSS

set_domain_supply_net TOP -primary_power_net VTOP -
primary_ground_net VSS


# level shifter rules

set_level_shifter LOW_LS -domain LOW  -applies_to outputs -
location parent -rule high_to_low

map_level_shifter_cell LOW_LS -lib_cells BASIC_LS_EXPR -
domain LOW
```

The violation messages reported by the *LPLSH05B* rule are:

```
[WARNING] Crossing signal from high voltage
'top.u1_inst.out[1:0]'(domain 'LOW(supply VDDL:1.200)') to low
voltage 'top.u2_inst.in[1:0]'(domain 'HIGH(supply VDDH:1.400)')
needs level shifter but has strategy 'LOW_LS' with incorrect
type 'high_to_low'


[WARNING] Crossing signal from high voltage
'top.u2_inst.out1'(domain 'HIGH(supply VDDH:1.400)') to low
voltage 'top.out1'(domain 'TOP(supply VTOP:1.300)') needs level
shifter but does not have strategy of type 'high_to_low' on
either source or destination domain boundary ports
```

The schematic generated is as follows:

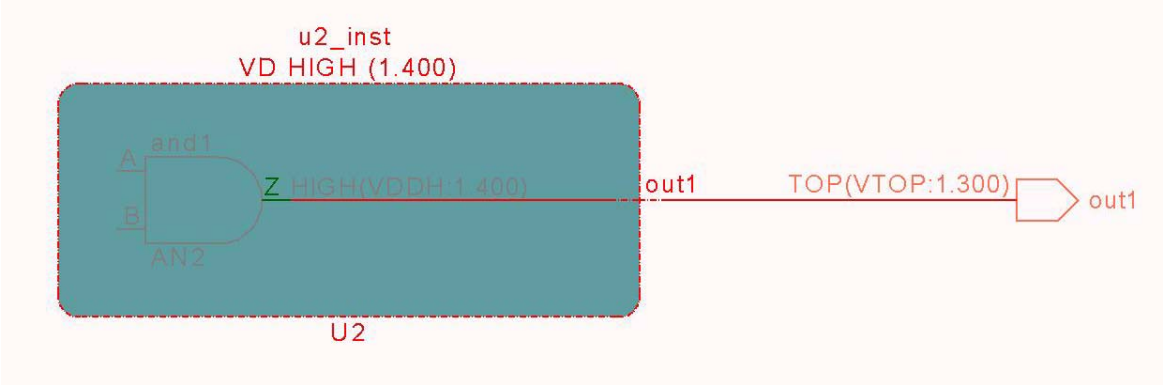


FIGURE 10. Incremental schematic

To fix these violations, provide a high-to-low level shifter strategy/rule matching with the domain crossing.

Example 2

Consider the following UPF file snippet.

```
set_level_shifter LS1 -domain top -applies_to inputs -rule
high_to_low -location parent
```

The following violation message is reported by the *LPLSH05B* rule because level shifter strategy `LS1` has been specified at source `top.in1` cannot be valid for both the destinations at the same time:

```
Crossing signal from high voltage 'top.in1'(domain 'top(supply
Vtop:1.100)') to low voltage 'top.e2.rtlc_I2.in1'(domain
'VD2(supply V_VD2:0.900)'), 'top.e1.rtlc_I2.in1    '(domain
'VD1(supply V_VD1:0.800)') has invalid level shifter strategy
'LS1' specified at source
```

The schematic generated is as follows:

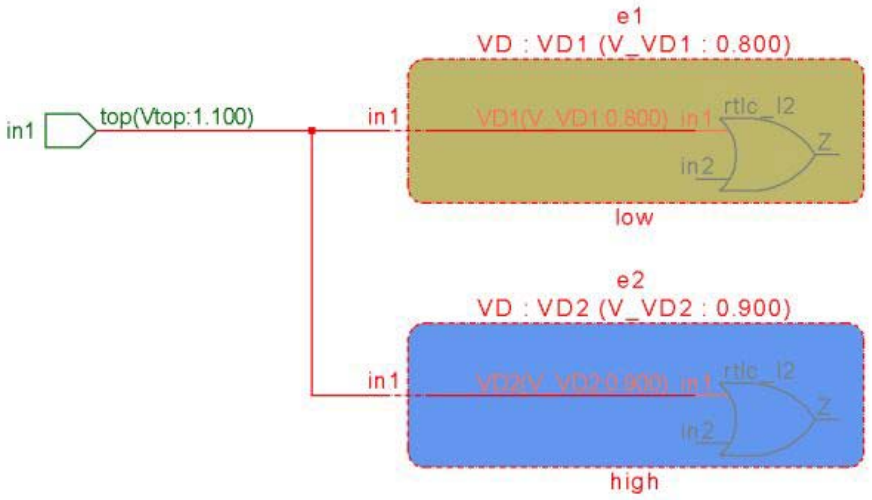


FIGURE 11. Incremental schematic

To fix this violation, provide level shifter strategies at the destinations.

Default Severity Label

Warning

Rule Group

Level Shifter

Reports and Related Files

None

LPLSH06

Checks level shifter compatibility with main rail of cell

NOTE: *This rule will be deprecated in a future release. Use the LPLSH08 rule, instead.*

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPLSH06* rule reports a violation message if the location of the level shifter specified in the *set_level_shifter* command is not compatible with the main rail of the cell. All the level shifter strategies which place level shifter to the left of the domain boundary, where the strategy is written, must contain only those level shifters specified in the *map_level_shifter_cell* command that have the main rail related to the (input) data pin of the level shifter.

Similarly, all the level shifter strategies which place level shifters to the right of the domain boundary, where the strategy is written, must contain only those level shifters in the *map_level_shifter_cell* command that have main rail related to output pin of the level shifter.

In addition, this rule reports a violation message if a *map_level_shifter_cell* or *map_isolation_cell* command contains conflicting level shifter specifications. For example, the commands contain a mix of level shifter cells where some cells relate the main rail to the input data pin while other cells relate it to the output pin.

Rule Exceptions

This rule does not perform any check on a level shifter strategy that has:

- The `-location` field set to automatic.
- The *set_isolation* command has a `-no_shift` field.
- Missing level shifter names.

Languages

Verilog, VHDL

Parameter(s)

lp_use_voltage_map_value: Default value is `0`. If you set the value of this parameter to `1`, the voltage value provided for the `voltage_map` attribute of source pin of a crossing is considered while calculating the level shifter requirement.

- *lp_skip_blackbox_checking*: Default value is `0`. Set the parameter to `1` to skip checking for black boxes.

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_level_shifter* (Mandatory)
- *map_isolation_cell* (Optional)
- *map_level_shifter_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when different types of level shifters are specified in the same *map_level_shifter_cell* or *map_isolation_cell* command:

[LPLSH06_1][ERROR] Different types of level shifters '<level-shifter-cells>' are specified in same <map_level_shifter_cell | map_isolation_cell> command '<cmd-name>'

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the location specified is not compatible with the main rail:

```
[LPLSH06_2][ERROR] Level Shifter cannot be implemented.
Location '<level-shifter-location>' specified in
set_level_shifter command '<cmd-name>' is not compatible with
cells specified in <map_level_shifter_cell |
map_isolation_cell> where main rail is related to <input |
output> data pin
```

Potential Issues

The violation messages appear because:

- **Message 1:** The violation message explicitly states the potential issue.
- **Message 2:** This violation message appears because the location specified in strategy is not compatible with cells specified in strategy.

Consequences of Not Fixing

Not resolving these violation messages can lead to a design failure.

How to Debug and Fix

The UPF command is highlighted in the Atrenta Console GUI. For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix these violations, perform the following:

- **Message 1**: Ensure the same types of level shifters are specified in the same *map_level_shifter_cell* or *map_isolation_cell* commands.
- **Message 2**: Ensure that correct level shifter cells compatible with location specified in strategy are specified in the same *map_level_shifter_cell* or *map_isolation_cell* commands.

Example Code and /or Schematic

Example 1

This example illustrates a circumstance in which Message 1 appears. Suppose, you have specified the following UPF commands.

```
set_level_shifter LS1 -domain LOW \
                      -applies_to outputs \
                      -rule low_to_high \
                      -location self \
                      -elements {/low_domain/out2}
set_isolation ISO -domain LOW -applies_to outputs -
isolation_power_net VDD
set_isolation_control ISO -domain LOW -isolation_signal iso
map_isolation_cell ISO -domain LOW -lib_cells { MY_LS_EN
BASIC_LS_EN BASIC_BUF}
```

The library specification is as follows.

```
cell (MY_LS_EN) {
    is_level_shifter : true;
    pg_pin (VDDY) { pg_type : backup_power;
std_cell_main_rail:true;}
    pin (A) { direction : input; related_power_pin : VDDA; }
    pin (E) { direction : input; level_shifter_enable_pin :
true; related_power_pin : VDDA;}
    pin (Y) { direction : output; function : "A&E";
related_power_pin : VDDY; }
     }


cell (BASIC_LS_EN) {
    is_level_shifter : true;
    pg_pin (VDDA) { pg_type :
backup_power;std_cell_main_rail:true; }
    pin (A) { direction : input; related_power_pin : VDDA;
level_shifter_data_pin:true;}
    pin (E) { direction : input; level_shifter_enable_pin :
true; related_power_pin : VDDA;}
```

```
    pin (Y) { direction : output; function : "A&E";
related_power_pin : VDDY; }
   }
```

For the MY_LS_EN cell, *std_cell_main_rail* relates to Y (output). While, BASIC_LS_EN, *std_cell_main_rail* relates to A (input). Since this is a mismatch, Message 1 is reported.

Example 2

This example illustrates the circumstances in which Message 2 appears. Suppose, you have specified the following UPF commands.

```
set_level_shifter LS3 -domain LOW \
                      -applies_to outputs \
                      -rule low_to_high \
                      -location parent  \
                      -elements {/low_domain/out3}


map_level_shifter_cell LS3 -domain LOW \
                           -lib_cells { BASIC_LS }
```

The library specification is as follows.

```
cell (BASIC_LS) {
      is_level_shifter : true;
      pg_pin (VDDA) { pg_type : primary_power;
std_cell_main_rail:true;}
      pin (A) { direction : input; related_power_pin :
VDDA;level_shifter_data_pin:true; }
      pin (Y) { direction : output; function : "A";
related_power_pin : VDDY; }
   }
```

Message 2 is reported because a cell with a main rail that relates to the input pin. To resolve Message 2, a cell with a main rail that relates to the output pin should be specified.

Example 3

This example illustrates the incremental schematic that is generated. The schematic highlights the crossing along which the *LPLSH06* rule reports a violation message.

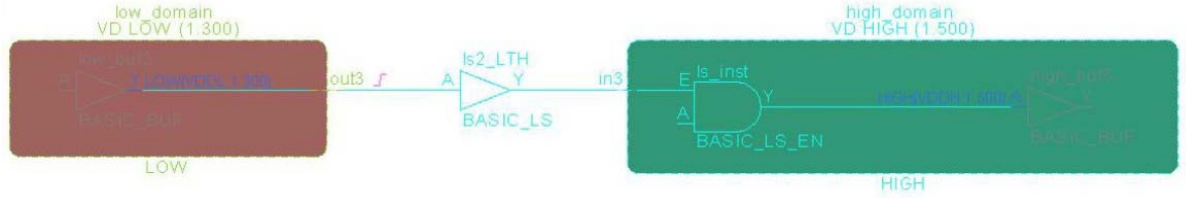


FIGURE 12. Incremental schematic

Default Severity Label

Error

Rule Group

`Level Shifter`

Reports and Related Files

None

LPLSH07

Checks the voltage range of level shifter cells with reference to source and destination domains

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPLSH07* rule reports a violation message in following scenarios (UPF only):

- No level shifter cell is provided as an input to SpyGlass.
- Level shifter cells are provided, but none of the level shifter cells has an input/output voltage range that matches with the voltage range of source/destination domains.

Languages

Verilog, VHDL

Parameter(s)

- *lp_skip_buf*: Default value is `1`. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking in the path between voltage domains while checking for the correctness of the level shifters.
- *lp_skip_pwr_gnd*: Default value is `1`. Set the parameter to `0` to process nets connected to power/ground supply while checking for the correctness of the level shifters.
- *lp_flag_undriven_nets*: Default value is `0`. Set the parameter to `1` to consider nets that are not driven at voltage crossings.
- *lp_flag_unconnected_nets*: Default value is `0`. Set the parameter to `1` to consider unconnected nets at voltage crossings.

- *lp_use_voltage_map_value*: Default value is `0`. If you set the value of this parameter to `1`, the voltage value provided for the `voltage_map` attribute of source pin of a crossing is considered while calculating the level shifter requirement.

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_level_shifter* (Mandatory)
- *map_level_shifter_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when the library does not contain a valid level shifter:

[LPLSH07_1][WARNING] There is no valid level shifter in the library whose input/output voltage range lies between the voltage range from domain '<src-domain>' to domain '<dest-domain>

Potential Issues

Liberty files containing level shifter cells are either not updated or not in sync with design.

Consequences of Not Fixing

An implementation tool will not be able to find and insert correct level shifters for the crossing where voltage shifting is required. Therefore, voltage shifting will not be done.

How to Debug and Fix

To fix this violation, update the library with level shifter cells that have a matching voltage range as specified in the UPF file.

Message 2

The following message appears when the level shifter specified in the `map_level_shifter_cell` command for a strategy is not within the range of voltage value of *add_port_state* command for the source and destination domain crossings:

[LPLSH07_2][**WARNING**] The LS cell defined in the command map_level_shifter_cell for strategy <strategy-name> is not within the range of add_port_state voltage values for the crossing of source domain <source-domain> destination domain <destination-domain>

Potential Issues

The input/output voltage ranges of the library cell(s) defined in the map_level_shifter_cell command(s) for the level shifter strategy are not compatible with the voltage values defined in *add_port_state* values.

Consequences of Not Fixing

An implementation tool will not be able to find and insert correct level shifters for the crossing where voltage shifting is required. Therefore, voltage shifting will not be done.

How to Debug and Fix

To fix this violation, specify a library cell in the map_level_shifter_cell command that is compatible with the *add_port_state* voltage ranges.

Example Code and /or Schematic

This example illustrates when the *LPLSH07* rule reports a violation message. In this example, there are four level shifter cells in the library.

Cell	Input Voltage Range	Output Voltage Range
LS1	0.9000-1.2600	0.9000-1.2600
LS2	0.9000-1.2600	0.9000-1.2600
LS3	0.9000-1.2600	0.9000-1.2600
LS4	0.9000-1.2600	0.9000-1.2600

The crossing is shown in following schematic. The voltage of the source is 0.8, which does not match with input voltage range of the any of the level shifters provided. Therefore, the LPLSH07 rule reports a violation. You can resolve this violation by providing correct level shifter cells in library.

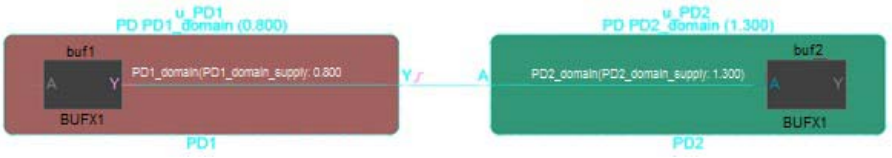


FIGURE 13. Incremental schematic

Default Severity Label

Warning

Rule Group

Level Shifter

Reports and Related Files

None

LPLSH08

Checks the level shifter cell location with respect to the std_cell_main_rail attribute at the RTL level

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

This rule reports a violation if the location of level shifter specified in the *set_level_shifter* command is not compatible with the main rail of the cell specified in the *map_level_shifter_cell* or *map_isolation_cell* command.

This rule also reports a violation if a *map_level_shifter_cell* or *map_isolation_cell* command contains a mix of level shifters cells where some cells have main rail related to input data pin of the cell while other cells have main rail related to output pin of the cell.

Languages

Verilog, VHDL

Parameter(s)

lp_check_equivalent_domain_for_main_rail: Default value is `yes`. Set this parameter to `no` to exactly match the source/destination domain and domain in which level shifter cell will be inferred.

Constraint(s)

UPF Commands

- *set_level_shifter* (Mandatory)
- *map_level_shifter_cell* (Optional)
- *map_isolation_cell* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when level shifters with different types of `std_cell_main_rail` attribute are specified in the same *map_level_shifter_cell* or *map_isolation_cell* command:

[LPLSH08_1][ERROR] Level shifters with different types of std_cell_main_rail '<cell-list>' are specified in same <map_level_shifter_cell/ map_isolation_cell> command '<command-name>'

Message 2

The following message appears when:

- The `std_cell_main_rail` attribute is written on input supply pin.
  - Domain in which level shifter cell will be inferred is not exactly the same or equivalent to the domain of source. Equivalent, here, implies that both domains should be similar (voltage wise) and also domain of level shifter cell should not be less on with respect to the source.
- The `std_cell_main_rail` attribute is written on output supply pin.
  - Domain in which level shifter cell will be inferred is not exactly the same or equivalent to the domain of destination. Equivalent, here, implies that both domains should be similar (voltage wise) and also domain of level shifter cell should not be less on with respect to the destination.

[LPLSH08_2][ERROR] Cell '<ls-cell-name>' specified in <map_level_shifter_cell | map_isolation_cell> '<strategy-name>' will be incorrectly connected in <source/destination/parent> location (domain '<ls-cell-domain-and-supply>') between source '<source-name>' (<source-domain-and-supply>) and destination '<destination-name>' (destination-domain-and-supply). std_cell_main_rail is specified on <input | output> pin of cell

Potential Issues

The violation messages appear because:

- **Message 1:** The violation message explicitly states the potential issue.
- **Message 2:** This violation message appears because the location specified in strategy is not compatible with cells specified in strategy.

Consequences of Not Fixing

Not resolving these violation messages can lead to a design failure.

How to Debug and Fix

The UPF command is highlighted in the Atrenta Console GUI. For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix these violations, perform the following:

- **Message 1**: Ensure the same types of level shifters are specified in the same *map_level_shifter_cell* or *map_isolation_cell* commands.
- **Message 2**: Ensure that correct level shifter cells compatible with location specified in strategy are specified in the same *map_level_shifter_cell* or *map_isolation_cell* commands.

Example Code and /or Schematic

Example 1

This example illustrates a situation in which Message 1 appears. Suppose, you have specified the following UPF commands.

```
set_level_shifter LS1 -domain LOW \
                      -applies_to outputs \
                      -rule low_to_high \
                      -location self \
                      -elements {/low_domain/out2}
map_level_shifter_cell LS1 -domain LOW -lib_cells { MY_LS_EN
BASIC_LS_EN }
```

The library specification is as follows.

```
cell (MY_LS_EN) {
    is_level_shifter : true;
    pg_pin (VDDY) { pg_type : backup_power;
std_cell_main_rail:true;}
    pin (A) { direction : input; related_power_pin : VDDA; }
```

```
    pin (E) { direction : input; level_shifter_enable_pin :
true; related_power_pin : VDDA;}
    pin (Y) { direction : output; function : "A&E";
related_power_pin : VDDY; }
     }


cell (BASIC_LS_EN) {
    is_level_shifter : true;
    pg_pin (VDDA) { pg_type :
backup_power;std_cell_main_rail:true; }
    pin (A) { direction : input; related_power_pin : VDDA;
level_shifter_data_pin:true;}
    pin (E) { direction : input; level_shifter_enable_pin :
true; related_power_pin : VDDA;}
    pin (Y) { direction : output; function : "A&E";
related_power_pin : VDDY; }
   }
```

For the MY_LS_EN cell, *std_cell_main_rail* relates to Y (output). While, BASIC_LS_EN, *std_cell_main_rail* relates to A (input). Since this is a mismatch, the following message is reported:

```
Level shifters with different types of std_cell_main_rail
'BASIC_LS_EN MY_LS_EN' are specified in same
map_level_shifter_cell command 'LS1'
```

Example 2

Consider the following example:

```
set_level_shifter LS2 -domain LOW \
                       -applies_to outputs \
                       -rule low_to_high \
                      -location fanout  \
                      -elements {/low_domain/out1}
```

```
map_level_shifter_cell LS2 -domain LOW -lib_cells { BASIC_LS
}
```

The library specification is as follows.

```
cell (BASIC_LS) {
       is_level_shifter : true;
      level_shifter_type : LH;
       input_voltage_range (1.2, 1.8);
      output_voltage_range (0.8, 1.5);
       pg_pin (VDDA) { pg_type : primary_power;
std_cell_main_rail:true;}
       pg_pin (VDDY) { pg_type : primary_power; }
       pg_pin (VSS) { pg_type : primary_ground; }
       pin (A) { direction : input; related_power_pin :
VDDA;level_shifter_data_pin:true; }
       pin (Y) { direction : output; function : "A";
related_power_pin : VDDY; }
    }
```

For the above example the *LPLSH08* rule reports the following violation message because the domain `HIGH`, in which level shifter cell is inferred, is not equivalent to the source domain `LOW`. As the `std_cell_main_rail` attribute has been specified for the input pin, the level shifter cell domain `HIGH` should have been equivalent to the source domain `LOW`.

```
Cell 'BASIC_LS' specified in map_level_shifter_cell 'LS2' will
be incorrectly connected in destination location (domain
'HIGH(supply VDDH:1.500)') between source
'TOP.low_domain.low_buf1.Y' (domain 'LOW(supply VDDL:1.300)')
and destination 'TOP.high_domain.high_buf1.A' (domain
'HIGH(supply VDDH:1.500)'). std_cell_main_rail is specified on
input pin of cell
```

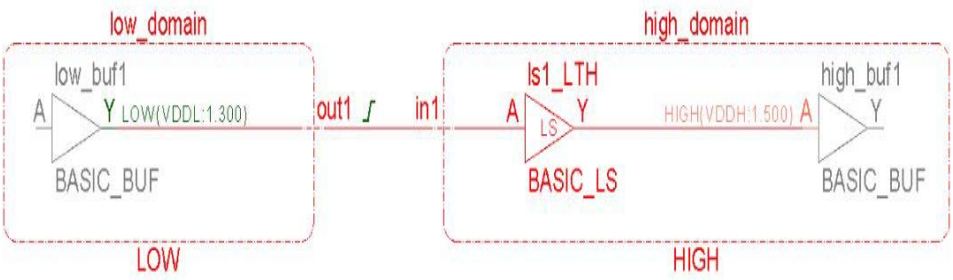


FIGURE 14. Incremental schematic

Default Severity Label

Error

Rule Group

`Level Shifter`

Reports and Related Files

None

LPSVM04

Checks for presence of level shifters in all voltage domain crossings

Language

Verilog, VHDL

Rule Description

The LPSVM04 rule flags the entire voltage domain crossing that does not use a level shifter. The rule checks whether the correct level shifter is present in a path from one voltage domain to the other.

The LPSVM04 rule is divided into four sub-rules: *LPSVM04A*, *LPSVM04B*, *LPSVM04C*, *LPSVM04D*, and *LPSVM04E*. When you select the LPSVM04 rule, all these rules are run.

NOTE: *A cell with the is_level_shifter attribute set as* true *in the library, is recognized as a level shifter cell. For details of the library attributes, refer to Using Constraints in the SpyGlass Power Verify Solution section.*

NOTE: *The LPSVM04 rule can also check for level shifters with an acceptable voltage range. Therefore, in case of Dynamic Voltage Scaling, the range of design voltages should be specified with power states, and the range of level shifter cell voltages should be specified in the library. In the CPF flow, range of level shifters can also be specified using the* define_level_shifter_cell *CPF command.*

Example

The subsequent section shows an example for defining voltage range for domains with power states in all three formats SGDC, CPF and UPF.

In the example, 2 domain (PD1 operates on 1.2V and PD2 operates over range of 0.8-1.2V) and 3 power states (PS1, PS2 and PS3) are defined such that:
For PS1, PD1 and PD2, both operate at 1.2 volts.
For PS2, PD1 operates at 1.2 and PD2 at 0.8 volts.
For PS3, PD2 is turned off (zero volts).

Inference from the provided information:

A level shifter of type HL should be present on the crossing from PD1 to PD2 and a level shifter of type LH should be present on the crossing from PD2 to PD1.

Following section provides detailed syntax for each format.

- Specifying power states in SGDC

```
voltage_domain -name PD1 -value 1.2 -modname top
voltage_domain -name PD2 -value 1.2 0.0 \
-instname top.U1
power_state -name PS1 -domain PD1@1.2 PD2@1.2
power_state -name PS2 -domain PD1@1.2 PD2@0.8
power_state -name PS3 -domain PD1@1.2 PD2@0.0
```

- Specifying power states in CPF

```
create_power_domain -name PD1 -default
create_power_domain -name PD2 -instances u1
create_nominal_condition -name High -voltage 1.2
create_nominal_condition -name Low -voltage 0.8
create_nominal_condition -name Off -voltage 0
create_power_mode -name PS1 \
  -domain_conditions {PD1@High PD2@High}
create_power_mode -name PS2 \
  -domain_conditions {PD1@High PD2@Low}
create_power_mode -name PS3 \
  -domain_conditions {PD1@High PD2@Off}
```

- Specifying power states in UPF

```
create_power_domain PD1
create_power_domain PD2 -elements u1
create_supply_net VDD1
create_supply_net VDD2
create_supply_net VSS
set_domain_supply_net PD1 -primary_power_net VDD1 \
-primary_ground_net VSS
```

```
set_domain_supply_net PD2 -primary_power_net VDD2 \
-primary_ground_net VSS

add_port_state VDD1 -state {HV 1.2}

add_port_state VDD2
-state {-state {High 1.2}\
-state {Low 0.8}\
-state {Off off}

create_pst dvfs_pst -supplies {VDD1 VDD2}

add_pst_state PS1 -pst dvfs_pst \
  -state {HV High}

add_pst_state PS2 -pst dvfs_pst \
  -state {HV Low}

add_pst_state PS3 -pst dvfs_pst -state {HV Off}
```

LPSVM04A

Reports level shifters in voltage domain crossings from lower voltage domain to higher voltage domain

When to Use

To check for level shifters in a multiple power design. This rule is applicable to all design phases after the insertion of level shifters. Refer to:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM04A* rule reports voltage domain crossings from a lower voltage domain to a higher voltage domain where the corresponding level shifter is not used. In addition, this rule reports signals of incorrect voltage domains connected to a level shifter specified for a voltage domain crossing from lower to higher voltage domain.

For UPF 2.0, the rule also reports if the level shifter cell name does not match the `name_prefix/name_suffix` specified in the `set_level_shifter` command.

Prerequisites

The information about power domains and their operating voltage are specified and the information about the level shifter is available in either CPF, UPF or SGDC format.

Rule Exceptions

The *LPSVM04A* rule skips the domain crossing check for the enable signal of the level shifter if the name of the enable terminal of the level shifter is specified using the `-enableTerm` argument of the *levelshifter* constraint. For example:

```
levelshifter -name LS_1_1_20 -from V1 -to V2
  -enableTerm EN
```

Language

Verilog, VHDL, DEF

Parameter(s)

- *lp_skip_pwr_gnd*: Default value is 1. Set the parameter to 0 to process nets connected to power/ground supply while checking for the correctness of the level shifters.
- *lp_skip_buf*: Default value is 1. Set the parameter to 0 to consider SpyGlass-generated buffers during rule checking in the path between voltage domains while checking for the correctness of the level shifters.
- *lp_skip_lib_buf*: Default value is 0. Set the parameter to 1 to skip the instances of library buffers while checking for crossings.
- *lp_complex_level_shifter*: Default value is 0. Set the parameter to 1 to recognize complex level-shifters also.
- *lp_flag_undriven_nets*: Default value is 0. Set the parameter to 1 to consider nets that are not driven at voltage crossings.
- *lp_flag_unconnected_nets*: Default value is 0. Set the parameter to 1 to consider unconnected nets at voltage crossings.
- *lp_use_voltage_map_value*: Default value is 0. If you set the value of this parameter to 1, the voltage value provided for the voltage_map attribute of source pin of a crossing is considered while calculating the level shifter requirement.
- *lp_check_pwr_gnd_to_macro_without_prd*: Default value is yes. Set this parameter to no to ignore crossings between tie-high/low and macro cell (is_macro_cell: true)/pad cell (pad_cell: true), without domain boundary. Level shifter and isolation checking for these crossings at the RTL and Netlist levels will be ignored.
- *lp_skip_blackbox_checking*: Default value is 0. Set the parameter to 1 to skip checking for black boxes.
- *lp_skip_same_src_supply_buf*: Default value is no. Set this parameter to yes to enable the rule skip buffers and inverters between strategy node and isolation cell which are of same supply as source but are LESS ON w.r.t. destination.

Constraint(s)

SDGC

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage/ power domains in the design.
- *levelshifter* (Mandatory): Use this constraint to specify the names of design units to be used as level shifters.
- *power_state* (Optional): Use this constraint to specify the combinations of domain states which can exist at the same time during a design operation.
- *pin_voltage* (Optional): Use this constraint to specify the voltage or the power domain in the design for primary ports and pins of design units or instances.
- *antenna_cell* (Optional): Use this constraint to specify the antennae protection cells (diode cells) which need to be ignored by the rules in the SpyGlass Power Verify solution.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_level_shifter_rule* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *update_level_shifter_rules* (Optional)
- *define_level_shifter_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_level_shifter* (Mandatory)

- *map_level_shifter_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when a signal <*sig-name*> going from source <*src*> to destination <*dest*> does not have an instance of the corresponding level shifter of type <type1> (LH or BOTH) in its path:

```
[LPSVM04A_1][WARNING] Signal '<sig-name>' is going from source
<src> (domain '<domain-supply1>') to destination <dest> (domain
'<domain-supply2>'), must contain level shifter of type
'<type1>'
```

NOTE: *In CPF and SGDC flow, the domain information (*<vd1>,<vd2>*) shows the domain name and its voltage value. However, in UPF flow, the domain information shows the domain name along with the associated supply name and supply value.*

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when a signal <*sig-name*> going from source <*src*> to destination <*dest*> is connected to incorrect level shifter <*ls-name*> (<*inst-name*>):

```
[LPSVM04A_2][WARNING] Signal '<sig-name>' is going from source
<src> (domain '<domain-supply1>') to destination <dest> (domain
'<domain-supply2>' is connected to incorrect level shifter
'<ls-name> (<inst-name>)'.<reason>
```

Where, <*reason*> is either of the following:

- Level shifter cell is not present in the *map_level_shifter_cell* command <*command-name*>.
- Level shifter cell is of H_L type but crossing is of L_H type.
- Voltage range of the level shifter <*LS-voltage-range*> does not match the voltage range of the crossing.
- As <*input/output*> LS supply <*supply-name*> is OFF with respect to the <*source/destination*> supply <*supply-name*>

- As source supply <*supply-name*> is OFF with respect to the input LS supply<*LS-supply*>
- As there is voltage difference between <*source/destination*> supply <*supply*> and the <*input/output*> LS supply <*LS-supply*>

NOTE: *The <reason> is reported with the UPF format only.*

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when a signal <*sig-name*> is going from source <*src*> to destination <*dest*> and the path has a pair of incorrect level shifters:

```
[LPSVM04A_3][WARNING] Signal '<sig-name>' is going from source
<src> (domain '<domain-supply1>') to destination <dest> (domain
'<domain-supply2>' does not have correct level shifters
```

Message 4

The following message appears in the UPF2.0 flow, when for a signal <*sig-name*> going from source <*src*> to destination <*dest*>, the level shifter instance name <*ls-inst-name*> does not match the prefix/suffix convention specified in the corresponding strategy:

```
[LPSVM04A_4][WARNING] For signal '<sig-name>' going from source
'<src>' (domain '<domain-supply1>') to destination '<dest>'
(domain '<domain-supply2>'), the level shifter instance name
'<ls-inst-name>' does not match the name <name> specified in
corresponding strategy '<strategy-name>'
```

Potential Issues

The rule reports issues, such as missing level shifter that needs voltage scaling or a level shifter with incorrect voltage level placed at a crossing.

Consequences of Not Fixing

The design would fail if the correct level shifter is not present on crossings that need voltage scaling.

Message 4 only shows that the naming prefix/suffix convention has not been followed and does not cause the design to fail.

How to Debug and Fix

The message states that the connection from low to high voltage either has an incorrect level-shifter or a missing level-shifter. In addition, it shows details of the source voltage domain, destination voltage domain, and their voltage value. The violation is reported at the first place where the signal is used.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Turn on the Power View. The schematic highlights the following:

- Signal path going from lower voltage domain to a higher voltage domain having no level shifter.
- Signal path going from one voltage domain to another voltage domain and the path has a pair of incorrect level shifters.

Refer to the *Example Code and/or Schematic* section to view a sample schematic and to understand this rule.

The *LPSVM04A* rule also generates a file named LPSVM04A.csv. This file contains all the messages generated by the *LPSVM04A* rule. To view the spreadsheet, in the Message Tree window, click the Open Spreadsheet option in the right-click menu of the rule.

Refer to the *Reports and Related Files* section for information on the reports generated by this rule.

To resolve this violation, perform the following:

- Insert missing level shifter in the signal path between low voltage domain to high voltage domain.
- Recheck from/to voltage domain of incorrectly used level shifters.

For, **Message 4** make sure level shifter instance name corresponds to the prefix/suffix convention specified in the corresponding strategy of the specified port, in the UPF.

For **Messages 1** and **3**, double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

Example Code and/or Schematic

Example 1

Consider the following example for level shifter checking on a simple

design. This example illustrates a crossing between V1 and V2.

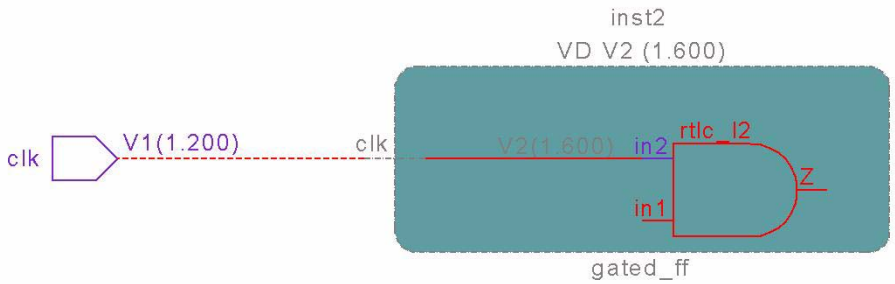


FIGURE 15. Incremental schematic showing crossings

To check this design, the hierarchical instance names of the voltage domains and their voltage values must be defined. The top level is assigned with `-modname`, while the other domains are assigned with `-instname` and their full hierarchical path names. The following SGDC file captures this information:

```
current_design top
voltagedomain  -name V1 -value 1.2 -modname top
voltagedomain -name V2 -value 1.6 -instname top.inst2
```

After running the analysis, the following messages appear:

```
[WARNING] Signal 'top.clk' is going from source 'top.clk'
(domain 'V1(voltage level 1.200)') to destination
'top.inst2.rtlc_I2.in2' (domain 'V2(voltage level 1.600)'),
must contain level shifter of type 'LH'
```

Example 2

Consider the following example. A violation occurs when signal `top.w3` going from lower voltage domain `V1` to a higher voltage domain `V3` does not have an instance of the corresponding level shifter in its path. The following message appears:

```
[WARNING] Signal 'top.w3[0]' is going from source
'top.inst1.q_reg.Q' (domain 'V1(voltage level 1.20)') to
destination 'top.inst2.rtlc_I1.A' (domain 'V3(voltage level
1.80)'), must contain level shifter of type 'LH'
```

The incremental schematic is displayed as shown in figure below.

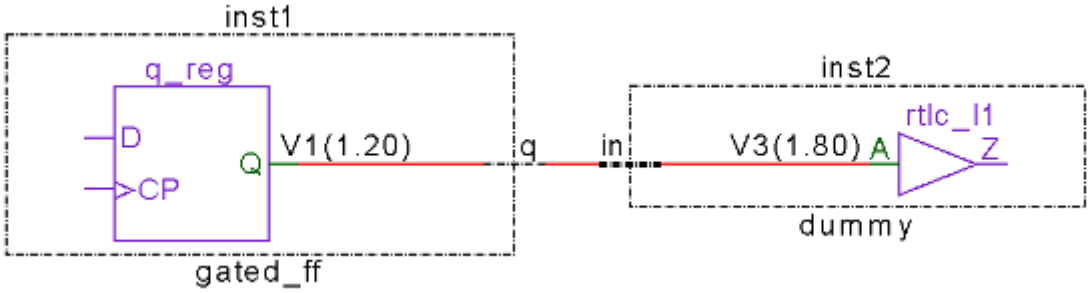


FIGURE 16. Incremental schematic

The schematic shows the following:

- the connection between the two domains
- the source voltage domain and its value
- the destination voltage domain and its value
- (Optional) the (incorrect) level shifter is highlighted in a different color.

Example 3

Consider the following UPF snippet:

```
set_level_shifter LS1 -domain HIGH1 -applies_to both
map_level_shifter_cell LS1 -domain HIGH1 -lib_cells
{BASIC_LS_EN}
```

A violation occurs when signal `TOP.low_domain.out1`, going from lower voltage domain `LOW(TOP.low_domain.low_buf1.Y)` to a higher voltage domain `HIGH1(TOP.high_domain1.high_buf3.A)`, has an incorrect level shifter `TOP.high_domain1.ls1_LTH (MY_LS_EN)` instantiated in it's path because the cell `MY_LS_EN` is not present in the cell list given in the *map_level_shifter_cell* command, in the above UPF. The following message appears:

[WARNING] Signal 'TOP.low_domain.out1' going from source 'TOP.low_domain.low_buf1.Y' (domain 'LOW(supply VDDL:1.000)') to destination 'TOP.high_domain1.high_buf3.A' (domain 'HIGH1(supply VDDH1:1.200)') is connected to incorrect level shifter 'TOP.high_domain1.ls1_LTH (MY_LS_EN)'. Level shifter cell is not present in map_level_shifter_cell command 'LS1'

The incremental schematic is displayed as shown in figure below:

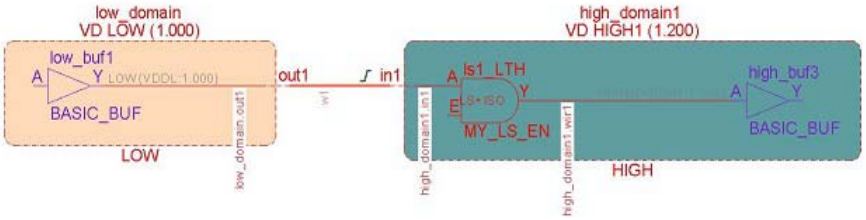


FIGURE 17. Incremental schematic

Example 4

Consider the following example. A violation occurs when signal `top.inst2.w1`, going from lower voltage domain `VD1(top.inst2.and1.Y)` to a higher voltage domain `VD2(top.inst2.inst3.and1.A)`, has two incorrect level shifters `top.inst2.ls1` and `top.inst2.ls11`, instantiated in its path. The following message appears:

[WARNING] Signal 'top.inst2.w1' going from source 'top.inst2.and1.Y' (domain 'VD1(supply VDD1:0.600)') to destination 'top.inst2.inst3.and1.A' (domain 'VD2(supply VDD2:0.800)') does not have correct level shifters

The incremental schematic is displayed as shown in figure below:

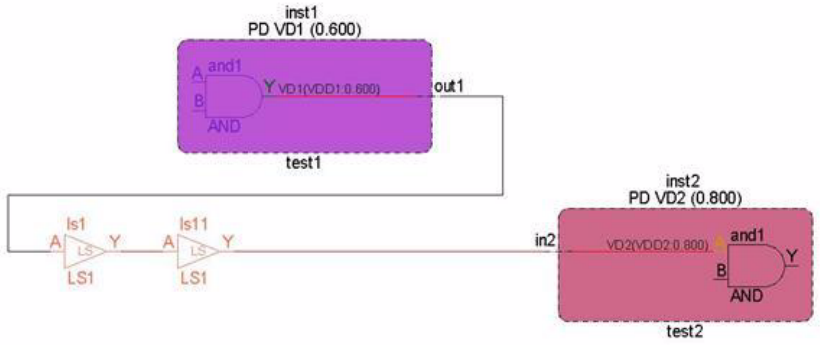


FIGURE 18. Incremental schematic

Example 5

Consider the following UPF snippet:

```
upf_version 2.0
 power domain definitions
create_power_domain top
create_power_domain shutOFF -elements { inst1 }

# supply nets
create_supply_port VDD
add_port_state VDD -state {default 1.2}
create_supply_net  VDD -domain top

create_supply_port VDD_LP
add_port_state VDD_LP -state {default 1.3}
create_supply_net  VDD_LP -domain shutOFF


connect_supply_net VDD -ports VDD
connect_supply_net VDD_LP -ports VDD_LP

create_supply_port VSS
create_supply_net  VSS -domain top
```

```
connect_supply_net VSS -ports VSS
add_port_state VSS -state {default 0.0}

set_domain_supply_net top -primary_power_net VDD
primary_ground_net VSS
set_domain_supply_net shutOFF -primary_power_net VDD_LP
primary_ground_net VSS

# isolation rules
set_level_shifter ls_r1 -domain shutOFF  -applies_to both
location parent -name_prefix ABC -name_suffix XYZ
map_level_shifter_cell ls_r1 -domain shutOFF -lib_cells
BASIC_LS

set_level_shifter ls_r2 -domain shutOFF  -elements { inst1
in1 } -location parent
map_level_shifter_cell ls_r2 -domain shutOFF -lib_cells
BASIC_LS_EN
```

For the above example, the rule reports the following violation message because for the signal `top.in_wir2` going from source `top.in_wir2` to destination `top.inst1.buf_inst2.A`, the level shifter instance name `top.inst1._ABC_XYZ_` does not match the prefix/suffix `prefix('ABC')/suffix('XYZ')` specified in corresponding strategy `ls_r1`:

```
[WARNING] For signal 'top.in_wir2' going from source
'top.in_wir2' (domain 'top(supply VDD:1.200)') to destination
'top.inst1.buf_inst2.A' (domain 'shutOFF(supply
VDD_LP:1.300)'), the level shifter instance name
'top.inst1._ABC_XYZ_' does not match the name prefix('ABC')/
suffix('XYZ') specified in corresponding strategy 'ls_r1'
```

The incremental schematic is displayed as shown in figure below.

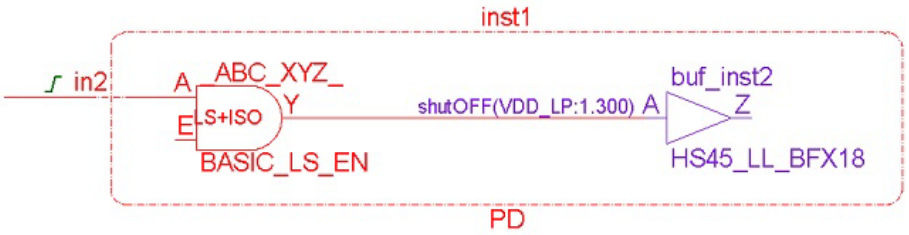


FIGURE 19. Incremental schematic

Default Severity Label

Warning

Rule Group

Voltage_Domain_Rules

Reports and Related Files

- *lp_vd_info*: Shows level shifter information for low to high voltage domain crossing.
- *lp_crossing_data*: Shows the valid level shifter cell for the combination of source and destination voltage domains.

LPSVM04B

Reports level shifters in voltage domain crossings from higher voltage domain to lower voltage domain.

When to Use

To check for level shifters in a multiple power design. This rule is applicable to all design phases after the insertion of level shifters. Refer to:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM04B* reports voltage domain crossings from a higher voltage domain to a lower voltage domain where the corresponding level shifter is not used. In addition, this rule reports signals of incorrect voltage domains connected to a level shifter specified for a voltage domain crossing from higher to lower voltage domain.

For UPF 2.0, the rule also reports if the level shifter cell name does not match the `name_prefix/name_suffix` specified in the `set_level_shifter` command.

Prerequisites

The information about power domains and their operating voltage are specified and the information about the level shifter is available in either CPF, UPF or SGDC format.

Rule Exceptions

The *LPSVM04B* rule skips domain crossing check for the enable signal of the level shifter if the name of the enable terminal of the level shifter is specified using the `-enableTerm` argument of the *levelshifter* constraint. For example:

```
levelshifter -name LS_1_1_20 -from V1 -to V2
  -enableTerm EN
```

Language

Verilog, VHDL. DEF

Parameter(s)

- *lp_skip_pwr_gnd*: Default value is 1. Set the parameter to 0 to process nets connected to power/ground supply while checking for the correctness of the level shifters.
- *lp_skip_buf*: Default value is 1. Set the parameter to 0 to consider SpyGlass-generated buffers during rule checking in the path between voltage domains while checking for the correctness of the level shifters.
- *lp_complex_level_shifter*: Default value is 0. Set the parameter to 1 to recognize complex level-shifters.
- *lp_flag_undriven_nets*: Default value is 0. Set the parameter to 1 to consider nets that are not driven at voltage crossings.
- *lp_flag_unconnected_nets*: Default value is 0. Set the parameter to 1 to consider unconnected nets at voltage crossings.
- *lp_ignore_b2b_ls*: Default value is 0. Set the parameter to 1 to enable the rule to check level shifters connected back to back in the design and report a violation only if they are improperly placed.
- *lp_skip_lib_buf*: Default value is 0. Set the parameter to 1 to skip the instances of library buffers while checking for crossings.
- *lp_use_voltage_map_value*: Default value is 0. If you set the value of this parameter to 1, the voltage value provided for the voltage_map attribute of source pin of a crossing is considered while calculating the level shifter requirement.
- *lp_allow_check_name_format*: Default value is no and this rule does not report a violation if isolation/level shifter instance name does not match with the name prefix/suffix specified in corresponding strategy. Set this parameter to yes to report violations for such cases.
- *lp_check_pwr_gnd_to_macro_without_prd*: Default value is yes. Set this parameter to no to ignore crossings between tie-high/low and macro cell (is_macro_cell: true)/pad cell (pad_cell: true), without domain boundary. Level shifter and isolation checking for these crossings at the RTL and Netlist levels will be ignored.

Constraint(s)

SDGC

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage/ power domains in the design.
- *levelshifter* (Mandatory): Use this constraint to specify the names of design units to be used as level shifters.
- *power_state* (Optional): Use this constraint to specify the combinations of domain states which can exist at the same time during a design operation.
- *pin_voltage* (Optional): Use this constraint to specify the voltage or the power domain in the design for primary ports and pins of design units or instances.
- *antenna_cell* (Optional): Use this constraint to specify the antennae protection cells (diode cells) which need to be ignored by the rules in the SpyGlass Power Verify solution.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_level_shifter_rule* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *update_level_shifter_rules* (Optional)
- *define_level_shifter_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_level_shifter* (Mandatory)

- *map_level_shifter_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when the signal <*sig-name*> traverses from the source <*src*> to the destination <*dest*> does not have an instance of the corresponding level shifter in its path:

[LPSVM04B_1][WARNING] Signal '<sig-name>' is going from source <src> (domain '<domain-supply1>') to destination <dest> (domain '<domain-supply2>', must contain level shifter of type 'HL'

NOTE: *In CPF and SGDC flow, the domain information (*<vd1>,<vd2>*) shows the domain name and its voltage value. However, in UPF flow, the domain information shows the domain name along with the associated supply name and supply value.*

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the signal <*sig-name*> traversing from the source <*src*> to the destination <*dest*> is connected to the incorrect level shifter <*ls-name*> (<*inst-name*>):

[LPSVM04B_2][WARNING] Signal '<sig-name>' is going from source <src> (domain '<domain-supply1>') to destination <dest> (domain '<domain-supply2>' is connected to incorrect level shifter '<ls-name> (<inst-name>)'.<reason>

Where, <*reason*> is either of the following:

- Level shifter cell is not present in the *map_level_shifter_cell* command <*command-name*>.
- Level shifter cell is of L_H type but crossing is of H_L type.
- Voltage range of the level shifter <*LS-voltage-range*> does not match the voltage range of the crossing.
- As <*input/output*> LS supply <*supply-name*> is OFF with respect to the <*source/destination*> supply <*supply-name*>

- As source supply <*supply-name*> is OFF with respect to the input LS supply<*LS-supply*>
- As there is voltage difference between <*source/destination*> supply <*supply*> and the <*input/output*> LS supply <*LS-supply*>

NOTE: *The* <*reason*> *is reported with the UPF format only.*

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when a signal <*sig-name*> is traversing from the source <*src*> to the destination <*dest*> and the path has a pair of incorrect level shifters:

[LPSVM04B_3][WARNING] Signal '<sig-name>' is going from source <src> (domain '<domain-supply1>') to destination <dest> (domain '<domain-supply2>' does not have correct level shifters

Message 4

The following message appears when for a signal <*sig-name*> going from source <*src*> to destination <*dest*>, the level shifter instance <*ls-inst-name*> does not match the prefix/suffix convention specified in the corresponding strategy. In UPF 2.0, prefix/suffix convention can be applied using the `–prefix/-suffix` argument of the `set_level_shifter` command. In UPF 1.0, the prefix/suffix convention can be applied using the `name_format` command. Also, in UPF 1.0, the checking of the `name_format` command is done using the *lp_allow_check_name_format* parameter:

[LPSVM04B_4][WARNING] For signal '<sig-name>' going from source '<src>' (domain '<domain-supply1>') to destination '<dest>' (domain '<domain-supply2>'), the level shifter instance name '<ls-inst-name>' does not match the name <name> specified in corresponding strategy '<strategy-name>'

Potential Issues

The rule reports issues, such as missing level shifter that needs voltage scaling or a level shifter with incorrect voltage level placed at a crossing.

Consequences of Not Fixing

The design would fail if the correct level shifter is not present on crossings that need voltage scaling.

Message 4 only shows that the naming prefix/suffix convention has not been followed and does not cause the design to fail.

How to Debug and Fix

The message states that the connection from high to low voltage either has an incorrect level-shifter or a missing level-shifter. In addition, it shows details of the source voltage domain, destination voltage domain, and their voltage value.

The violation is reported at the first place where the signal is used.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Turn on the Power View. The schematic highlights the following:

- Signal path going from higher voltage domain to a lower voltage domain having no level shifter.
- Signal path going from one voltage domain to another voltage domain and the path has a pair of incorrect level shifters.

Refer to the *Example Code and/or Schematic* section to view a sample schematic and to understand this rule.

The *LPSVM04B* rule also generates a file named LPSVM04B.csv. This file contains all the messages generated by the *LPSVM04B* rule. To view the spreadsheet, in the Message Tree window, click the Open Spreadsheet option in the right-click menu of the rule.

Refer to the *Reports and Related Files* section for information on the reports generated by this rule.

To resolve this violation, perform the following:

- Insert missing level shifter in the signal path between high voltage domain to low voltage domain.
- Recheck from/to voltage domain of incorrectly used level shifters.

Double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

For **Message 4**, make sure level shifter instance name corresponds to the prefix/suffix convention specified in the corresponding strategy of the specified port, in the UPF.

Example Code and/or Schematic

Example 1

Consider the following example for level shifter checking on a simple design. This example illustrates a crossing between V2 and V1.

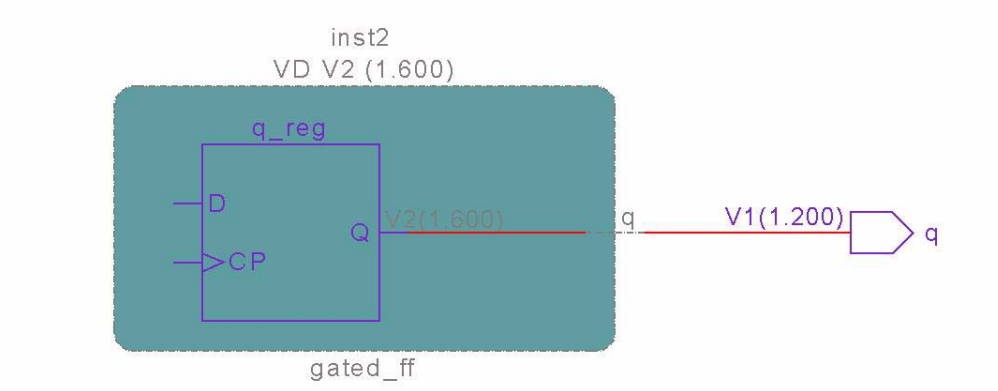


FIGURE 20. Incremental schematic showing crossings

To check this design, the hierarchical instance names of the voltage domains and their voltage values must be defined. The top level is assigned with `-modname`, while the other domains are assigned with `-instname` and their full hierarchical path names. In addition, the cell names and pins of the level shifters must be defined. The following SGDC file captures this information:

```
current_design top
voltagedomain  -name V1 -value 1.2 -modname top
voltagedomain -name V3 -value 1.8 -instname top.inst1
voltagedomain -name V2 -value 1.6 -instname top.inst2
levelshifter -name ls -from V1 -to V3 -inTerm in -outTerm out
levelshifter -name ls -from V1 -to V2 -inTerm in -outTerm out
levelshifter -name ls -from V2 -to V1 -inTerm in -outTerm out
levelshifter -name ls -from V2 -to V3 -inTerm in -outTerm out
```

After running the analysis, the following messages appear:

```
[WARNING] Signal 'top.inst2.q' is going from source
'top.inst2.q_reg.Q' (domain 'V2(voltage level 1.600)') to
destination 'top.q' (domain 'V1(voltage level 1.200)'), must
contain level shifter of type 'HL'
```

Example 2

Consider the following example. A violation occurs when a signal `top.inst1.out` going from higher voltage domain `VD1` to a lower voltage domain `VD2` does not have an instance of the corresponding level shifter in its path:

```
[WARNING] Signal 'top.inst1.out' is going from source
'top.inst1.and1.Y' (domain 'VD1(supply VDD1:1.200)') to
destination 'top.inst3.and1.B' (domain 'VD2(supply
VDD2:0.900)'), must contain level shifter of type 'HL'
```

The incremental schematic is shown below.

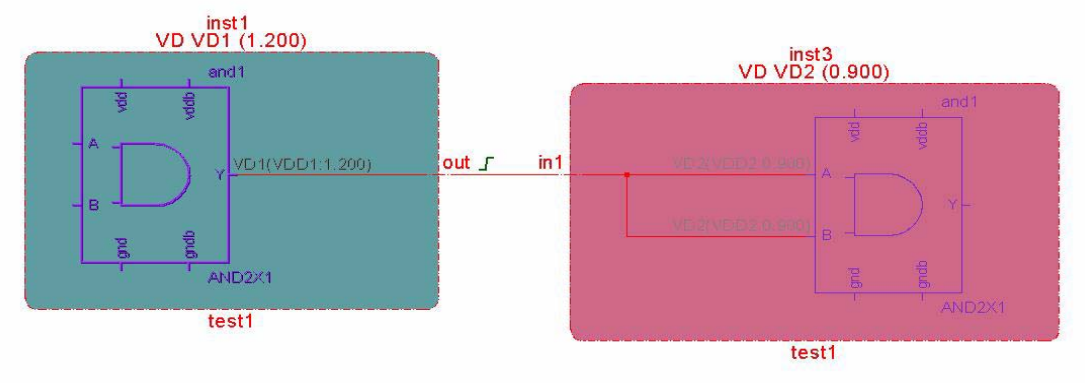


FIGURE 21. Incremental schematic

The schematic for the message shows the following:

- the connection between the two domains
- the source voltage domain and its value

- the destination voltage domain and its value
- (Optional) the (incorrect) level shifter is highlighted in a different color.

Example 3

Consider the following example. A violation occurs when signal `top.inst2.w1`, going from a higher voltage domain `VD2(top.inst2.and1.Y)` to a lower voltage domain `VD3(top.inst2.inst3.and1.A)`, has two incorrect level shifters, `top.inst2.ls3 and top.inst2.ls4`, instantiated in its path. The following message appears:

```
[WARNING] Signal 'top.inst2.w1' going from source
'top.inst2.and1.Y' (domain 'VD2(supply VDD2:0.800)') to
destination 'top.inst2.inst3.and1.A' (domain 'VD3(supply VDD3
0.600)') does not have correct level shifters.
```

The incremental schematic is shown below:

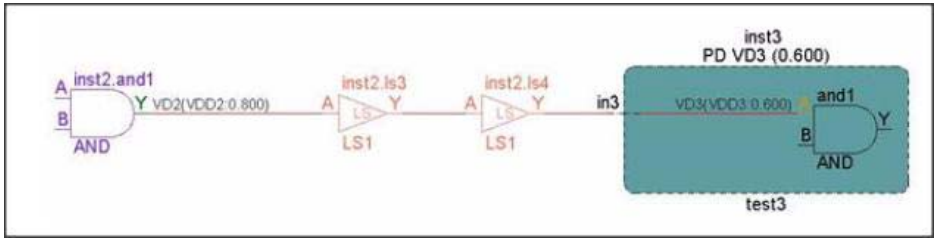


FIGURE 22. Incremental schematic

Example 4

Consider the following library file snippet.

```
cell (BASIC_LS_EN) {
      is_level_shifter : true;
      level_shifter_type : HL;
      input_voltage_range (1.0, 1.2);
      output_voltage_range (1.0, 1.2);
      pg_pin (VDDA) { pg_type : backup_power; }
      pg_pin (VDDY) { pg_type : primary_power; }
```

```
        pg_pin (VSS) { pg_type : primary_ground; }
        pin (A) { direction : input; related_power_pin : VDDA;
    }
        pin (E) { direction : input; level_shifter_enable_pin
        :true; related_power_pin : VDDA;}
        pin (Y) { direction : output; function : "A&E";
    related_power_pin : VDDY; }
}
```

A violation occurs when signal `TOP.high_domain2.wir3`, going from a higher voltage domain `HIGH2(TOP.high_domain2.high_buf3.Y)` to a lower voltage domain `LOW(TOP.out3)`, has incorrect level shifter, `TOP.high_domain2.ls1_HTL (BASIC_LS_EN)`, instantiated in its path because voltage range of level shifter given in library file above does not match with the voltage range of crossing. The following message appears:

```
[WARNING] Signal 'TOP.high_domain2.wir3' going from source
'TOP.high_domain2.high_buf3.Y' (domain 'HIGH2(supply
VDDH2:1.500)') to destination 'TOP.out3' (domain 'LOW(supply
VDDL:1.000)') is connected to incorrect level shifter
'TOP.high_domain2.ls1_HTL (BASIC_LS_EN)'. Voltage range of the
level shifter '(1.000,1.200-1.000,1.200)' does not match with
voltage range of the crossing.
```

The incremental schematic is shown below:

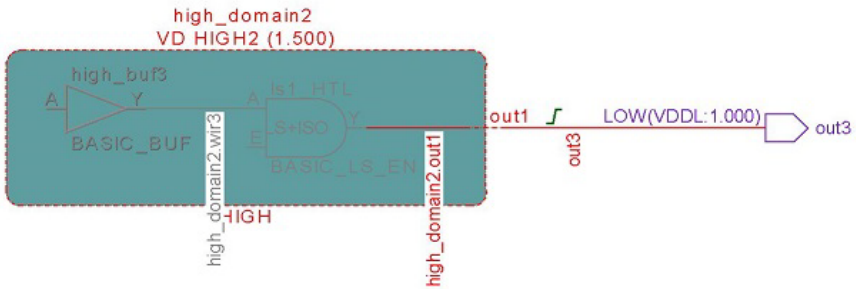


FIGURE 23. Incremental schematic

Example 5

Consider the following UPF snippet:

```
upf_version 2.0
# power domain definitions

create_power_domain shutOFF -elements { inst1 }

# supply nets
create_supply_port VDD
add_port_state VDD -state {default 1.2}
create_supply_net  VDD -domain top

create_supply_port VDD_LP
add_port_state VDD_LP -state {default 1.3}
create_supply_net  VDD_LP -domain shutOFF


connect_supply_net VDD -ports VDD
connect_supply_net VDD_LP -ports VDD_LP

create_supply_port VSS
create_supply_net  VSS -domain top
connect_supply_net VSS -ports VSS
add_port_state VSS -state {default 0.0}

set_domain_supply_net top -primary_power_net VDD
primary_ground_net VSS
set_domain_supply_net shutOFF -primary_power_net VDD_LP
primary_ground_net VSS

# isolation rules
set_level_shifter ls_r1 -domain shutOFF  -applies_to both
location self -name_prefix ABC -name_suffix XYZ
map_level_shifter_cell ls_r1 -domain shutOFF -lib_cells
BASIC_LS

set_level_shifter ls_r2 -domain shutOFF  -elements { inst1
```

```
out1 } -location self -name_prefix ABC
map_level_shifter_cell ls_r2 -domain shutOFF -lib_cells  {
BASIC_LS_EN BASIC_LS }
```

For the above example, the rule reports the following violation message because for the signal `top.inst1.w2` going from source `top.inst1.buf_inst2.Z` to destination `top.buf_inst4.A`, the level shifter instance name `top.inst1.ABC_XYZ_` does not match the suffix name `suffix('XYZ')` specified in corresponding strategy `ls_r1`:

```
[WARNING] For signal 'top.inst1.w2' going from source
'top.inst1.buf_inst2.Z' (domain 'shutOFF(supply VDD_LP:1.300)')
to destination 'top.buf_inst4.A' (domain 'top(supply
VDD:1.200)'), the level shifter instance name
'top.inst1.ABC_XYZ_' does not match the name suffix('XYZ')
specified in corresponding strategy 'ls_r1'
```

The incremental schematic is displayed as shown in figure below.

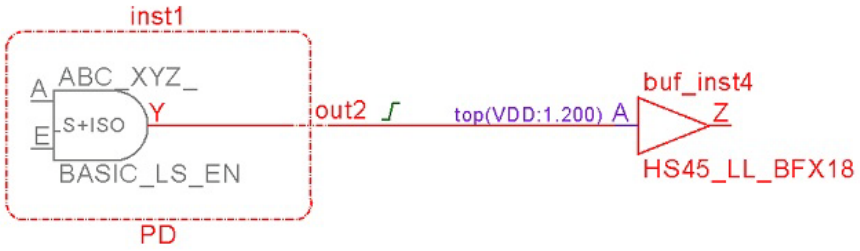


FIGURE 24. Incremental schematic

Default Severity Label

Warning

Rule Group

Voltage_Domain_Rules

Reports and Related Files

- *lp_vd_info*: Shows level shifter information for low to high voltage domain crossing.
- *lp_crossing_data*: Shows the valid level shifter cell for the combination of source and destination voltage domains.

LPSVM04C

Reports inadvertently used level shifters in the design

When to Use

To check for level shifters in a multiple power design. This rule is applicable to all design phases after the insertion of level shifters. Refer to:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM04C* rule flags inadvertently used level shifters.

If a level shifter has instances/ports lying in same voltage domain at its input and output, the level shifter is considered inadvertently used. This rule flags such level shifters where they are not required but are used inadvertently.

Prerequisites

The information about power domains and their operating voltage are specified and the information about the level shifter is available in either CPF, UPF or SGDC format.

Language

Verilog, VHDL, DEF

Parameter(s)

- *lp_skip_pwr_gnd*: Default value is `1`. Set the parameter to `0` to process nets connected to power/ground supply while checking for the correctness of the level shifters.
- *lp_skip_buf*: Default value is `1`. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking in the path between voltage domains while checking for the correctness of the level shifters.
- *lp_complex_level_shifter*: Default value is `0`. Set the parameter to `1` to have the *LPSVM04C* rule recognize complex level-shifters also.

- *lp_flag_undriven_nets*: Default value is `0`. Set the parameter to `1` to have the *LPSVM04C* rule consider nets that are not driven at voltage crossings.
- *lp_flag_unconnected_nets*: Default value is `0`. Set the parameter to `1` to have the *LPSVM04C* rule consider unconnected nets at voltage crossings.
- *lp_ignore_b2b_ls*: Default value is `0`. Set the parameter to `1` to enable the rule to check level shifters connected back to back in the design and report a violation only if they are improperly placed.
- *lp_check_pwr_gnd_to_macro_without_prd*: Default value is `yes`. Set this parameter to `no` to ignore crossings between tie-high/low and macro cell (`is_macro_cell: true`)/pad cell (`pad_cell: true`), without domain boundary. Level shifter and isolation checking for these crossings at the RTL and Netlist levels will be ignored.
- *lp_skip_blackbox_checking*: Default value is `0`. Set the parameter to `1` to skip checking for black boxes.

Constraint(s)

SDGC

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage/power domains in the design.
- *levelshifter* (Mandatory): Use this constraint to specify the names of design units to be used as level shifters.
- *pin_voltage* (Optional): Use this constraint to specify the voltage or the power domain in the design for primary ports and pins of design units or instances.
- *antenna_cell* (Optional): Use this constraint to specify the antennae protection cells (diode cells) which need to be ignored by the rules in the SpyGlass Power Verify solution.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_level_shifter_rule* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)

- *update_level_shifter_rules* (Optional)
- *define_level_shifter_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_level_shifter* (Mandatory)
- *map_level_shifter_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when the signal <*sig-name*> traverses from the source <*src*> to the destination <*dest*> and the path has a pair of out-of-place level shifters:

[LPSVM04C_1][WARNING] Signal '<sig-name>' is going from source <src> (domain '<domain-supply1>') to destination <dest> (domain '<domain-supply2>' has out of place level shifters

NOTE: *In CPF and SGDC flow, the domain information (*<vd1>,<vd2>*) shows the domain name and its voltage value. However, in UPF flow, the domain information shows the domain name along with the associated supply name and supply value.*

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the level shifter <*ls-name*> (<*inst-name*>) shifting from the source <*src*> to the destination <*dest*> is used out-of-place:

[LPSVM04C_2][WARNING] Level shifter '<ls-name>(<inst-name>)' shifting from source <src> (domain '<domain-supply1>') to destination <dest> (domain '<domain-supply2>' is used out of place

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when the *lp_ignore_b2b_ls* parameter is set to 0 and the level shifter <ls-name1> (<inst-name1>) is driving another level shifter <ls-name2> (<inst-name2>) (back-to-back connected):

[LPSVM04C_3][WARNING] Level shifter '<ls-name1>(<inst-name1>)' is driven by another level shifter '<ls-name2>(<inst-name2>)'

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears when the level shifter <ls-cell-name> is present at the excluded signal <sig-name> specified with <arg-name> (-exclude/-no_shift) argument of *create_level_shifter_rule*/ *set_level_shifter* CPF/UPF command, respectively:

[LPSVM04C_4][WARNING] Signal '<sig-name>' going from source <src> (domain '<domain-supply1>') to destination <dest> (domain '<domain-supply2>' has '<arg-name>' specified and contains a level shifter '<ls-inst-name> (<ls-cell-name>)'

Potential Issues

There is a level shifter at a crossing, which does not need voltage scaling.

Consequences of Not Fixing

The design would fail if the correct level shifter is not present on crossings that need voltage scaling.

How to Debug and Fix

The message states that the connection from same voltage level either has an incorrect level-shifter or an inadvertently (out of place) used level-shifter. In addition, it shows details of the source voltage domain, destination voltage domain and their voltage value. The violation is reported at the first place where the signal is used.

For a graphical view of the violation, double-click the message and click the

Incremental Schematic button. Turn on the Power View. The schematic highlights the signal path traversing from one voltage domain to another voltage domain, having out of place level shifters.

Refer to the *Example Code and/or Schematic* section to view a sample schematic and to understand this rule.

The *LPSVM04C* rule also generates a file named LPSVM04C.csv. This file contains all the messages generated by the *LPSVM04C* rule. To view the spreadsheet, in the Message Tree window, click the Open Spreadsheet option in the right-click menu of the rule.

Refer to the *Reports and Related Files* section for information on the reports generated by this rule.

To resolve this violation, check out-of-place level shifters used from one voltage domain to another voltage domain.

Example Code and/or Schematic

Example 1

Consider the following example. The following message appears when a level shifter `top.ls1`, shifting from voltage domain `VD1` to voltage domain `VD2`, has been used out-of-place:

```
[WARNING] Level shifter 'top.ls1 (BASIC_LS)' shifting from
source 'top.inst1.and1.Y' (domain 'VD1(supply VDD1:1.200)') to
destination 'top.inst3.and1.A' (domain 'VD2(supply
VDD2:1.200)') is used out of place
```

The incremental schematic is displayed as shown in figure below:

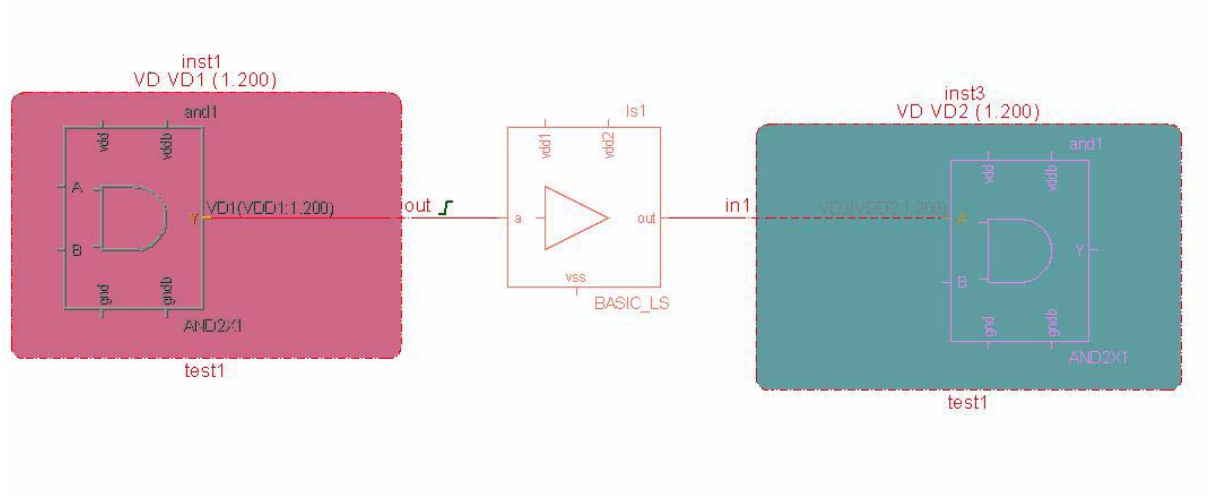


FIGURE 25. Incremental schematic

The schematic for the message shows the following:

- the connection between the two domains
- the source voltage domain and its value
- the destination voltage domain (same as source voltage domain) and its value
- the (incorrect) level shifter is highlighted in a different color.

Example 2

Consider the following example where two out-of-place level shifters, `top.I1_L_A.G3` and `top.I1_L_B.G1`, are placed back to back, and the *lp_ignore_b2b_ls* parameter is set to `1`. The following message appears:

```
[WARNING] Signal 'top.I1_L_A.G2_Y' going from source
'top.I1_L_A.G2.Y' (domain 'V2(voltage level 0.700)') to
destination 'top.I1_L_B.G2.A' (domain 'V2(voltage level
0.700)') has out of place level shifters.
```

The incremental schematic is displayed as shown in figure below:

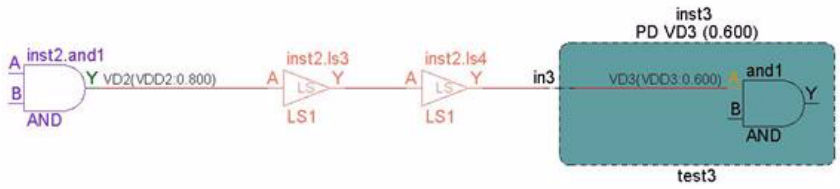


FIGURE 26. Incremental schematic

Example 3

Consider the following example where two level shifters, `top.inst2.ls3` and `top.inst2.ls4`, are placed back to back, and the *lp_ignore_b2b_ls* parameter is set to `0`. The following message appears:

```
[WARNING] Level shifter 'top.inst2.ls4' ( LS1 ) is driven by
another level shifter 'top.inst2.ls3' ( LS1 )
```

The incremental schematic is displayed as shown in figure below:

FIGURE 27. Incremental schematic

Example 4

Consider the following UPF snippet:

```
set_level_shifter NO_LS -domain HIGH -elements { /
high_domain/in1} -no_shift
```

In the above snippet, level shifter `top.inst2.ls3` is placed on the

signal crossing element `top.high_domain.in1`. The following violation is reported because the `-no_shift` option is specified on this particular element:

```
[WARNING] Signal 'TOP.low_domain.out3' going from source
'TOP.low_domain.low_buf3.Y' (domain 'LOW(supply VDDL:0.800)')
to destination 'TOP.high_domain.high_buf3.A' (domain
'HIGH(supply VDDH:1.200)') has '-no_shift' specified and
contains a level shifter 'TOP.high_domain.ls1_LTH (MY_LS_EN)'
```

The incremental schematic is displayed as shown in figure below:

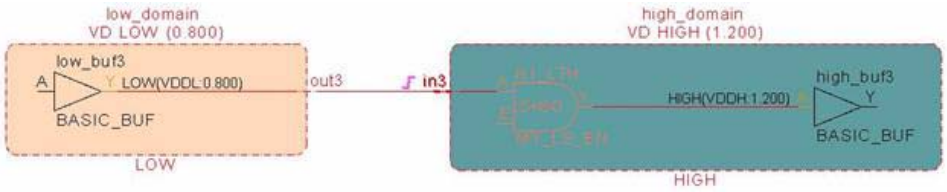


FIGURE 28. Incremental schematic

Default Severity Label

Warning

Rule Group

Voltage_Domain_Rules

Reports and Related Files

- *lp_crossing_data*: Shows that no level shifter cell is required for the combination of source and destination voltage domains. The report shows the following:
  - the `Level Shifter Type required` field shows `None`.
  - the list of pins which are specified as excluded pins.

LPSVM04D

Checks location of level shifters based on information specified in power format files

When to Use

To check for level shifters in a multiple power design. This rule is applicable to all design phases after the insertion of level shifters. Refer to:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM04D* rule reports level shifters that are instantiated in a voltage domain other than their specified voltage domain.

This rule requires you to specify the voltage domains using the *voltage_domain* constraint and the level shifters (design units) between a pair of voltage domains using the *levelshifter* constraint (having the `-locate` argument to specify the voltage domain for the level shifter being defined) in a SpyGlass Design Constraints file. Then, the *LPSVM04D* rule flags those level shifter instances that are not in the voltage domain specified with the `-locate` argument of the *levelshifter* constraint defining the master level shifter design unit.

For example, in the following specification all instances of level shifter `LS_1_1_20` are supposed to be in the voltage domain `VD`.

```
levelshifter -name LS_1_1_20 -from V1 -to V2 -locate VD
```

The *LPSVM04D* rule reports any instantiation of `LS_1_1_20` lying in any other voltage domain.

NOTE: *A cell with the is_level_shifter attribute set as* `true` *in the library, is recognized as a level shifter cell.*

Prerequisites

The information about power domains and their operating voltage are specified and the information about the level shifter is available in either CPF, UPF or SGDC format.

Language

Verilog, VHDL, DEF

Parameter(s)

- *lp_skip_pwr_gnd*: Default value is `1`. Set the parameter to `0` to process nets connected to power/ground supply while checking for the correctness of the level shifters.
- *lp_skip_buf*: Default value is `1`. Set the parameter to `0` to consider SpyGlass-generated buffers in the path between voltage domains while checking for the correctness of the level shifters.
- *lp_complex_level_shifter*: Default value is `0`. Set the parameter to `1` to have the rule recognize complex level-shifters also.
- *lp_flag_undriven_nets*: Default value is `0`. Set the parameter to `1` to have the rule consider nets that are not driven at voltage crossings.
- *lp_flag_unconnected_nets*: Default value is `0`. Set the parameter to `1` to have the rule consider unconnected nets at voltage crossings.
- *lp_match_location_by_domain*: Default value is `yes` and the isolation cell and level shifter cell locations are matched based on domain. To match these locations based on hierarchy, set the value of this parameter to `no`.
- *lp_skip_blackbox_checking*: Default value is `0`. Set the parameter to `1` to skip checking for black boxes.

Constraint(s)

SDGC

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage/ power domains in the design.
- *levelshifter* (Mandatory): Use this constraint to specify the names of design units to be used as level shifters.
- *pin_voltage* (Optional): Use this constraint to specify the voltage or the power domain in the design for primary ports and pins of design units or instances.

CPF Commands

- *create_power_domain* (Mandatory)

- *create_level_shifter_rule* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *update_level_shifter_rules* (Optional)
- *define_level_shifter_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_level_shifter* (Mandatory)
- *map_level_shifter_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when the level shifter <*ls-name*> (<*inst-name*>) at crossing from the source <src> to destination <dest> is present in an invalid location:

[LPSVM04D_1][WARNING] Level shifter '<ls-name>(<inst-name>)' at crossing going from source <src> (domain '<domain-supply1>') to destination <dest> (domain '<domain-supply2>' is not present in valid location

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appear in the CPF flow, when a location <loc> is specified with the -within_hierarchy option of the *update_level_shifter_rules* command for the level shifter cell <ls-

name>(<inst-name>), which is at the crossing from source <src> to destination <dest>, present at an invalid location:

[LPSVM04D_2][WARNING] Level shifter '<ls-name>(<inst-name>)' at crossing from source <src> (domain '<domain-supply1>') to destination <dest> (domain '<domain-supply2>' is not present in valid location <loc>

Potential Issues

The level shifter is placed in an incorrect location instead of the specified location.

Consequences of Not Fixing

The supply connection of the level shifter may fail due to unavailability of the correct supply. This could lead to design failure.

How to Debug and Fix

The message states the location of the level-shifter is incorrect. In addition, it also shows details of the source voltage domain, destination voltage domain and their voltage value. The violation is reported at the first place where the level shifter is instantiated.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Turn on the Power View. The schematic highlights the level shifter that is in a voltage domain other than the specified voltage domain.

Refer to the *Example Code and/or Schematic* section to view a sample schematic and to understand this rule.

The *LPSVM04D* rule also generates a file named LPSVM04D.csv. This file contains all the messages generated by the *LPSVM04D* rule. To view the spreadsheet, in the Message Tree window, click the Open Spreadsheet option in the right-click menu of the rule.

Refer to the *Reports and Related Files* section for information on the reports generated by this rule.

To resolve this violation, relocate the level shifter to the specified voltage domain.

Example Code and/or Schematic

Consider the following example. The following message appears when the level shifter BASIC_LS is not correctly placed.

```
[WARNING] Level shifter 'top.ls1 (BASIC_LS)' at crossing from
source 'top.inst1.and1.Y' (domain 'VD1(supply VDD1:0.900)') to
destination 'top.inst3.and1.A' (domain 'VD2(supply
VDD2:1.200)') is not present in valid location
```

The incremental schematic is displayed as shown in figure below.

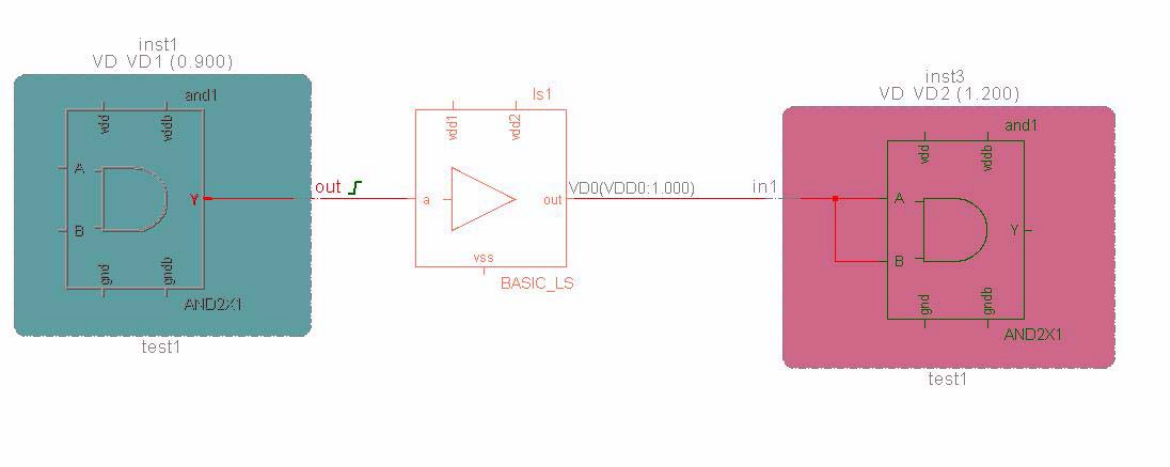


FIGURE 29. Incremental schematic

The schematic for the message shows the following:

- the voltage domain of the incorrect level shifter
- the (incorrect) level shifter is highlighted in a different color.

Default Severity Label

Warning

Rule Group

Voltage_Domain_Rules

Reports and Related Files

- *lp_crossing_data*: Shows the valid level shifter location for this combination of source and destination voltage domains. The `Level shifter location` field shows the correct location for the highlighted level-shifter.

LPSVM04E

Checks location of level shifters based on main supply rail specified with 'std_cell_main_rail' attribute

When to Use

To check for level shifters in a multiple power design. This rule is applicable to all design phases after the insertion of level shifters. Refer to:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM04E* rule reports all level shifters in the design that are placed in incorrect domain based on the *std_cell_main_rail* attribute. This rule checks for the location of the level shifter only when the *std_cell_main_rail* library attribute is specified. For details of the library attributes, refer to *Using Constraints in the SpyGlass Power Verify Solution* section.

The *LPSVM04E* rule reports a violation:

- when a level shifter with the *std_cell_main_rail* attribute specified on the output supply pin is placed in a domain not matching with the destination/sink domain.
- when a level shifter with the *std_cell_main_rail* attribute specified on the input supply pin is placed in a domain not matching with the source/driving domain.
- for incorrectly placed level shifters.

Prerequisites

The information about power domains and their operating voltage are specified and the information about the level shifter is available in either CPF, UPF or SGDC format.

Language

Verilog, VHDL, DEF

Parameter(s)

- *lp_skip_pwr_gnd*: Default value is 1. Set the parameter to 0 to process nets connected to power/ground supply while checking for the correctness of the level shifters.
- *lp_skip_buf*: Default value is 1. Set the parameter to 0 to consider SpyGlass-generated buffers during rule checking in the path between voltage domains while checking for the correctness of the level shifters.
- *lp_complex_level_shifter*: Default value is 0. Set the parameter to 1 to have the rule recognize complex level-shifters also.
- *lp_flag_undriven_nets*: Default value is 0. Set the parameter to 1 to have the rule consider nets that are not driven nets at voltage crossings.
- *lp_flag_unconnected_nets*: Default value is 0. Set the parameter to 1 to have the rule consider unconnected nets at voltage crossings.
- *lp_skip_blackbox_checking*: Default value is 0. Set the parameter to 1 to skip checking for black boxes.

Constraint(s)

SDGC

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage/ power domains in the design.
- *levelshifter* (Mandatory): Use this constraint to specify the names of design units to be used as level shifters.
- *pin_voltage* (Optional): Use this constraint to specify the voltage or the power domain in the design for primary ports and pins of design units or instances.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_level_shifter_rule* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *update_level_shifter_rules* (Optional)
- *define_level_shifter_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_level_shifter* (Mandatory)
- *map_level_shifter_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

The following message appears when the level shifter <*ls-name*> (<*inst-name*>) crossing from source domain <src-dom> to destination domain <des-dom> is present in an incorrect domain <dom>:

[LPSVM04E_1][WARNING] Level shifter '<ls-name>(<inst-name>)' crossing from source <src> (domain '<domain-supply1>') to destination <dest> (domain '<domain-supply2>' is incorrectly placed in domain '<dom>'

Potential Issues

The level shifter is placed in an incorrect location instead of the specified location.

Consequences of Not Fixing

The supply connection of the level shifter may fail due to unavailability of the correct supply. This could lead to design failure.

How to Debug and Fix

The message states that the level shifter is present in an incorrect domain. It shows the details, such as cell name and instance name, of the level shifter. In addition, it states details of the source voltage domain, destination voltage domain, and the domain in which the level shifter is present with their voltage values. The message appears when the level shifter is placed in the destination voltage domain, but is placed in the

source voltage domain, or vice versa.

The violation is reported at the first place where the level shifter is instantiated.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Turn on the Power View. The schematic highlights incorrectly placed level shifter.

Refer to the *Example Code and/or Schematic* section to view a sample schematic and to understand this rule.

The *LPSVM04E* rule also generates a file named LPSVM04E.csv. This file contains all the messages generated by the *LPSVM04E* rule. To view the spreadsheet, in the Message Tree window, click the Open Spreadsheet option in the right-click menu of the rule.

Refer to the *Reports and Related Files* section for information on the reports generated by this rule.

To resolve this violation, place the:

- Level shifter with an output pin specified with the *std_cell_main_rail* attribute in a domain matching with the destination/sink domain.
- Level shifter with an input pin specified with the *std_cell_main_rail* attribute in a domain matching with source/driving domain.

Example Code and/or Schematic

Consider the following example. Level shifter `TOP.ls2_LTH (BASIC_LS_EN)` crossing from source domain `HIGH(supply VDDH:1.10)` to destination domain `LOW(supply VDDL:0.90)` is present in an incorrect domain `TOP(supply VDD:1.00)`:

```
[WARNING] Level shifter 'TOP.ls2_LTH (BASIC_LS_EN)' crossing
from source 'top.high_domain.low_buf3.Y' (domain 'HIGH(supply
VDDH:1.10)') to destination 'top.low_domain.high_buf3.A'
(domain 'LOW(supply VDDL:0.90)') is incorrectly placed in
domain 'TOP(supply VDD:1.00)'
```

The incremental schematic is displayed as shown in figure below.

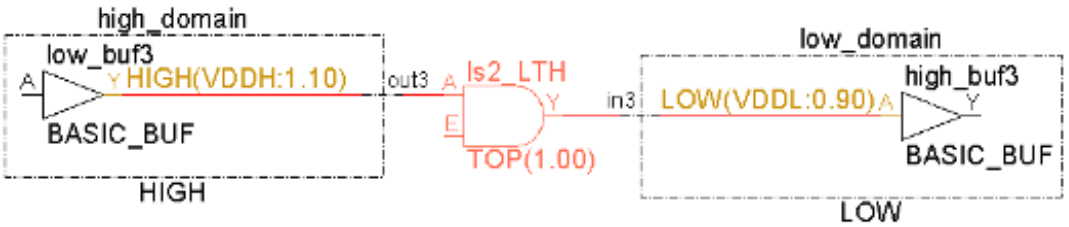


FIGURE 30. Incremental schematic

The schematic for the message shows the following:

- the connection between the two domains
- the source voltage domain and its value
- the destination voltage domain and its value
- the (incorrect) level shifter is highlighted in a different color
- the voltage domain of the incorrect level shifter

Default Severity Label

Warning

Rule Group

Voltage_Domain_Rules

Reports and Related Files

- *lp_lib_data*: Shows the valid domain for this level shifter cell. From this report, read the `input supply pin` and `output supply pin` fields for the level shifter cell. One of these fields will have an asterisk (`*`).

  If the output supply pin is shown as an asterisk, consider placing the level shifter in a domain matching with the destination/sink domain. If the input supply pin is shown as an asterisk, consider placing the level shifter in a domain matching with the source/driving domain.

LPSVM17

Reports multiple instances of the same level shifter

When to Use

NOTE: *This rule will be deprecated in a future release.*

Use the *LPSVM17 rule* for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM17* rule reports multiple instances of the same level shifter for a single domain crossing.

Prerequisites

The *LPSVM17* rule requires that the voltage domains are specified using the *voltage_domain* constraint and the level shifters (design units) between a pair of voltage domains using the *levelshifter* constraint in an SGDC file.

Rule Exceptions

The *LPSVM17* rule does not flag cases when an incorrect level shifter is used in one or more voltage domain crossings between two voltage domains while all other crossings between these voltage domains have a correct level shifter.

Language

Verilog, VHDL

Parameter(s)

- *lp_complex_level_shifter*: Default value is `0`. Set the value to`1` to recognize complex level-shifters.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage domain in the design.
- *levelshifter* (Mandatory): Use to specify the level shifter in the design.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_level_shifter_rule* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *update_level_shifter_rules* (Optional)
- *define_level_shifter_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_level_shifter* (Mandatory)
- *map_level_shifter_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

The following message appears, when multiple instance of the level shifter design unit <*ls-du-name*> is found for the same input <*sig-name*>:

[LPSVM17_1][INFO] Multiple instances of Level Shifter '<ls-du-name>'found in the design for input signal '<sig-name>'

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

This is just an informational message that highlights multiple instances specified for the same level shifter for the same domain crossing.

How to Debug and Fix

The violation is reported at a place where the level shifter, which has multiple instances, is instantiated.

The *LPSVM17* rule is a multi-line highlighting rule. When you click on the *LPSVM17* rule messages in Atrenta Console, multiple source window tabs are opened, each displaying and highlighting the source code for a rule violating instance.

To fix the violation, check the level shifters with multiple instances for an input signal.

Example Code and/or Schematic

The *LPSVM17* rule highlights the level shifter with multiple instances for the same input signal.

Consider the following example. A violation occurs when multiple instance of the level shifter `'lshfter_V1_V2'` are found in the design for the same input `'top.in'`. The following message appears:

[INFO] Multiple instances of Level Shifter 'lshfter_V1_V2' found in the design for input signal 'top.in'

The schematic generated is as follows:

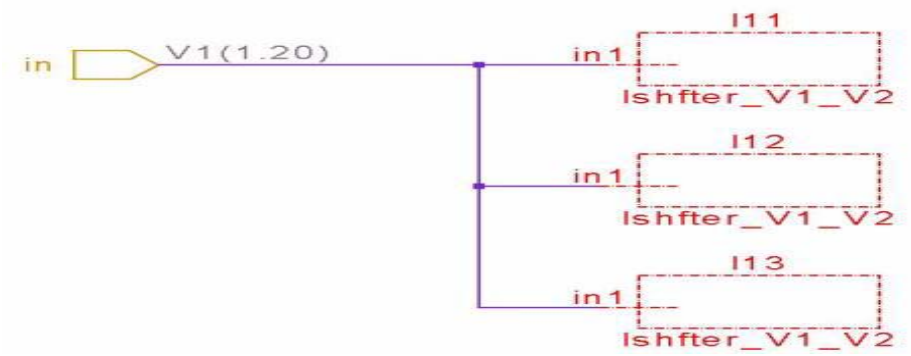


FIGURE 31. Incremental schematic

There are ‘three’ instances of the level shifter ‘lshfter_V1_V2’ in the design for the same domain crossing from voltage domain ‘V1’ to ‘V2’. This rule reports the same.

Default Severity Label

Info

Rule Group

Level Shifter

Reports and Related Files

None

LPSVM24

Reports enable of a level shifter that does not belong to an always-on domain

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM24* rule reports the following cases:

- The enable pin is connected to an instance that is not in an always-on domain.
- The enable pin is connected to an instance of an always-on domain but the voltage of the enable pin voltage domain and the voltage domain of the instance/port connected to the output pin of the level shifter are not equal when the *lp_check_with_output* parameter is set.

Prerequisites

The *LPSVM24* rule requires to specify the voltage domains using the *voltage_domain* constraint and the level shifters (design units) between a pair of voltage domains and their respective enable pins using the *levelshifter* constraint with the `-enableTerm` argument in an SGDC file.

Language

Verilog, VHDL

Parameter(s)

- *lp_check_with_output*: Default value is `0`. Set this value to `1` to check the enable of a level shifter in a different voltage domain from that of the instance or port to which the output of the level shifter is connected.
- *lp_skip_buf*: Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/power domains and its information is used by *SP_01* rule.
- *levelshifter* (Mandatory): Use to specify the names of design units to be used as level shifters.

CPF Commands

- *create_power_domain* (Mandatory): Use to create a power domain.
- *create_level_shifter_rule* (Mandatory): Use to create a rule for adding a level shifter.
- *create_nominal_condition* (Mandatory): Use to infer voltage values for domains in a power state.
- *create_power_mode* (Mandatory): Use to create a power mode.
- *update_level_shifter_rules* (Optional): Use to update the details of a level shifter rules.
- *define_level_shifter_cell* (Mandatory): Use to identify the library cells in the .lib files that can be used as level shifter cells.
- *define_power_clamp_cell* (Optional): Use to specify the list of diode cells used for power clamp control.

UPF Commands

- *create_power_domain* (Mandatory): Use to create power domain.
- *create_supply_port* (Mandatory): Use to create supply port.
- *add_port_state* (Mandatory): Use to state a port.
- *create_supply_net* (Mandatory): Use to create a supply net.
- *connect_supply_net* (Mandatory): Use to connect a supply net with a supply port.
- *set_domain_supply_net* (Mandatory): Use to specify a primary power and ground supply nets for a power domain.
- *set_level_shifter* (Mandatory): Use to specify a level shifter strategy.
- *map_level_shifter_cell* (Optional): Use to map a particular level shifter strategy to a library cell or range of library cells.

- *create_pst* (Optional): Use to create a Power State Table (PST) by using a specific order of supply nets.
- *add_pst_state* (Optional): Use to define the states of each of the supply nets for one possible state of the design.

Messages and Suggested Fix

Message 1

The following message appears when the enable pin of level shifter <*ls-du-name*> belongs to the voltage domain <*vd1*> that is not an always-on voltage domain:

[LPSVM24_1][RECOMMENDED] Enable (voltage domain '<vd1>') of the level shifter '<ls-du-name>' is not in the always-on domain

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the voltage <*num1*> of the voltage domain <*vd1*> of enable pin of level shifter <*ls-du-name*> is not equal to the voltage <*num2*> of the voltage domain <*vd2*> (voltage <*num2*>) of the voltage domain of the instance/port connected to the output of the level-shifter:

[LPSVM24_2][RECOMMENDED] Enable of the level shifter '<ls-du-name>' coming from voltage domain '<vd1>'(value '<num1>') does not match with the output voltage domain '<vd2>'(value '<num2>')

For debugging information, click *How to Debug and Fix*.

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not Fixing

The consequences of not fixing the violations are as follows:

Message 1: The enable pin of the level shifter doesn't lie in an alwayson domain. This is a design flaw.

Message 2: The enable pin of the level shifter lies in a domain different from the domain at the destination side of the level shifter when parameter

lp_check_with_output = 1 is provided. This is a design flaw.

How to Debug and Fix

The violation is reported at a level shifter cell instance having enable pin not connected to an always-on domain or coming from voltage domain does not match with the output voltage domain (if parameter *lp_check_with_output* set to 1).

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To resolve these violation messages, ensure that the:

- enable of level shifters are connected to always-on domains.
- enable of level shifters lie in the same voltage domain of the instance or port to which the output of the level shifter is connected.

Example Code and/or Schematic

The *LPSVM24* rule highlights the enable of level shifter which does not lie in always-on domain. In the following example, a violation occurs when enable 'E' of level shifter `top.ls` lie in the switchable domain 'PD'.

```
Enable(voltage domain 'PD') of the level shifter 'top.ls' does
not lie in the alwayson domain
```

The schematic generated is as follows:

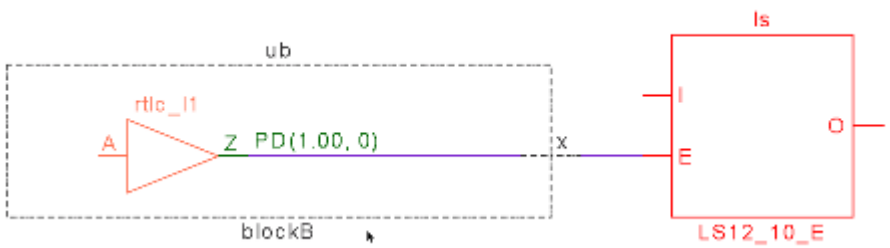


FIGURE 32. Incremental schematic

Here, the enable pin of the level shifter is being driven by a net coming from a power domain ‘PD’ which is not an always on domain.

Default Severity Label

Recommended

Rule Group

Level Shifter

Reports and Related Files

None

LPLIB_check04

Checks input/output voltage range of level-shifter cells

When to Use

This rule is to be used to report undefined or incorrect input/output voltage range declarations for level-shifter cells used in the design.

Description

While defining level-shifter cells in library files, the `input_voltage_range` and `output_voltage_range` values need to be provided too. These values represent the voltage range in which the input/output ports of the cell can be operated.
Each voltage range should be defined with two values, the lower bound and the upper bound.

This can be defined at cell level (applies to all pins in cell) or at pin level (applies to that pin only).

The pin level definition gets precedence over cell-level definition when both are available.

This rule generates violations in the following scenarios:

- When input/output voltage range is not defined for level-shifter cells
- When input/output voltage range is defined with single value instead of the both, lower-bound and upper-bound values
- When input/output voltage range is defined with more than two values

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

Message 1

The following message appears when input/output voltage range is not defined for level-shifter cells:

[LPLIB_check04_1][FATAL] <Input | Output> voltage range is not defined for level shifter cell '<cell-name>'

Message 2

The following message appears when input/output voltage range is defined with single value instead of both lower-bound and upper-bound values:

[LPLIB_check04_2][FATAL] <Input | Output> voltage range defined for level shifter cell '<cell-name>' has only one value. It should contain both lower bound and upper bound

Message 3

The following message appears when input/output voltage range is defined with more than two values:

[LPLIB_check04_3][FATAL] <Input | Output> voltage range defined for level shifter cell '<cell_name>' contains more than 2 values. It should contain lower bound and upper bound only

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

SpyGlass requires you to resolve this violation.

How to Debug and Fix

The signal pin and cell stated in the violation message are highlighted in the Atrenta Console GUI.

To fix these violations:

Message 1: Make sure that the `output_voltage_range` attribute is added either at the cell level or pin level.

Message 2: Make sure that both lower-bound and upper-bound values are provided, instead of a single value.

Message 3: Make sure that the voltage range is defined with two values only.

Example Code and/or Schematic

Example 1

Consider the following example:

```
cell (LS1) {
      is_level_shifter : true;
      level_shifter_type : HL;
      input_voltage_range (0.9, 1.2);
      pg_pin (VDD) { pg_type : primary_power; }
      pg_pin (VSS) { pg_type : primary_ground; }
      pin (A) { direction : input; related_power_pin : VDDC;
related_ground_pin : VSS; }
      pin (Y) { direction : output; function : "A";
related_power_pin : VDDC; related_ground_pin : VSS; }
   }
```

In the above example, the *LPLIB_check04* rule reports the following violation message because the `output_voltage_range` attribute is missing for the cell `LS1`.

```
Output voltage range is not defined for level shifter cell
'LS1'
```

Example 2

Consider the following example:

```
cell (LS2) {
      is_level_shifter : true;
      level_shifter_type : HL;
      pg_pin (VDD) { pg_type : backup_power; }
      pg_pin (VDDC) { pg_type : primary_power; }
      pg_pin (VSS) { pg_type : primary_ground; }
      pin (A) { direction : input; related_power_pin : VDDC;
related_ground_pin : VSS; input_voltage_range (0.9);  }
      pin (Y) { direction : output; function : "A";
related_power_pin : VDDC; related_ground_pin : VSS;
output_voltage_range (0.7 0.8); }
   }
```

In the above example, the *LPLIB_check04* rule reports the following violation message because the `input_voltage_range` attribute is defined with only one value.

```
Input voltage range defined for level shifter cell 'LS2' has
only one value. It should contain both lower bound and upper
bound
```

Example 3

Consider the following example:

```
cell (LS3) {
      is_level_shifter : true;
      level_shifter_type : HL;
      pg_pin (VDD) { pg_type : backup_power; }
      pg_pin (VDDC) { pg_type : primary_power; }
      pg_pin (VSS) { pg_type : primary_ground; }
      pin (A) { direction : input; related_power_pin : VDDC;
related_ground_pin : VSS; input_voltage_range (0.9 1.1 1.2);
}
      pin (Y) { direction : output; function : "A";
related_power_pin : VDDC; related_ground_pin : VSS;
output_voltage_range (0.7 0.8); }
   }
```

In the above example, the *LPLIB_check04* rule reports the following violation message because the `input_voltage_range` attribute is defined with three values.

```
Input voltage range defined for level shifter cell 'LS3'
contains more than 2 values. It should contain lower bound and
upper bound only
```

Default Severity Label

Fatal

Rule Group

`Isolation Logic`

Reports and Related Files

None

Isolation Logic Rules

The Isolation Logic group of rules is as follows:

Rule	Description
LPISO01	Reports unused isolation cells in the design
LPISO02	Checks isolation cell location in the design on the basis of its supply pins
LPISO03	Checks the presence of isolation cell at output terminals of power-domain.
LPISO03A	Reports missing isolation cell at output terminals of power domain, not having any isolation strategy.
LPISO03B	Reports non-existence of isolation cell at excluded output terminals of power domain.
LPISO04	Reports missing and incorrect isolation strategies
LPISO04A	Reports missing isolation strategy at power domain output ports
LPISO04B	Reports incorrect isolation strategy at power domain output ports
LPISO04C	Checks isolation strategy for incorrect isolation_power_net or isolation_ground_net
LPISO04D	Reports standard isolation cell that cannot be implemented
LPISO05	Checks for redundant isolation strategy at power domain output ports
LPISO05A	Checks for cases where isolation is not required for all paths
LPISO05B	Checks for cases where isolation is needed for at least one path but is not required for other paths
LPISO06	Checks if enable pin of isolation cell is tied to isolating value or not
LPISO06A	Reports an error when the enable pin of an isolation cell is not tied to an isolating value
LPISO06B	Reports an informational message when the enable pin of an isolation cell is tied to an isolating value
LPISO07	Checks if the clamp value provided in isolation strategy is in sync with the constant value reaching at the element specified in the strategy

Rule	Description
LPSVM08	Checks for the presence of correct isolation cells in all power domain output crossings
LPSVM08A	Checks for correct isolation logic of various power domains in the design
LPSVM08B	Reports missing isolation cell at output terminals of power domain, not having any isolation strategy
LPSVM08C	Reports non-existence of isolation cell at excluded output terminals of power domain
LPSVM09	Checks if power domain output attains the specified value in power down condition
LPSVM10	Displays values on the output of power domain under shutdown condition
LPSVM12	Isolation signal should be a state signal and in always-on domain
LPSVM12A	Checks that the isolation signal always comes from an always-on domain
LPSVM12B	Ensures that the isolation signal is always a state signal
LPSVM15	Reports power domains that have multiple enable signals
LPSVM22	Reports inadvertently used isolation cells in the design
LPSVM26	Checks whether logic cells in the power domain in feedthrough between two voltage domain crossings exist
LPSVM28	Checks if the net attains the specified value in power down condition
LPSVM31	Reports isolation cells that do not receive a steady state value in the power down condition
LPSVM47	Reports power domain inputs that do not attain the specified value in power down condition
LPSVM48	Reports cells other than isolation cells used at the input-side and output-side of the power domain
LPSVM50	Reports incorrect instances at the input stage of a power domain
LPSVM51	Reports isolation cells that are not transparent in power up mode
LPSVM52	Reports clock nets going into the power domain are feeding an instance of another domain

Rule	Description
LPSVM55	Reports isolation cells not present at a specific location in the hierarchy
LPSVM60	Reports power domain inputs that are not isolated
LPLIB_check01	Reports isolation cells that have ambiguous or undefined enable pins
LPLIB_check02	Checks for missing *related_power_pin*/*related_ground_pin* attribute in library for signal pin
LPLIB_check03	Checks the isolation_enable_condition attribute in library for a signal pin

LPISO01

Reports unused isolation cells in the design

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPISO01* rule performs the following connectivity related checks on the isolation cells used in the design and signals on which the isolation strategy is present:

- The input pin of the isolation cell is not driven or is unconnected.
- The enable pin of the isolation cell is unconnected.
- The output pin of the isolation cell is unloaded or unconnected.
- The signal, on which the isolation strategy is present, is undriven
- The signal, on which the isolation strategy is present, is unloaded

A cell with the *is_isolation_cell* attribute set as `true` in the library is recognized as an isolation cell.

NOTE: *A level shifter with an enable pin (clamp level shifter cell) that is tied to a constant value is not treated as an isolation cell.*

Prerequisites

By default, the *LPISO01* rule is not run while running the SpyGlass Power Verify solution. To enable this rule, select the *LPISO01* rule in the Atrenta Console GUI.

Language

VHDL, Verilog

Parameter(s):

- *lp_disable_lib_attr_read*: Default value is 0. This signifies that the *LPISO01* rule reads the power related library attributes from the library. Set the value to 1 to provide the information from SGDC, UPF, or CPF files.
- *lp_skip_buf:* Default value is 1 and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to 0 to consider SpyGlass-generated buffers during rule checking.
- *lp_skip_combo_cell_for_reference_toplevel_isolation_signal*: Default value is 0. Set this parameter to 1 to skip combinational cells whose output pin is related to signal input pin to get correct source of control signal.

Constraint(s):

SGDC

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage domains in the design.
- *isolation_cell* (Optional): Use to specify the isolation cells in power domains.
- *input_isocell* (Optional): Use to specify the isolation cells at inputs of a power domain.
- *reference_toplevel_isolation_signal* (Optional): Use this constraint to specify the reference top-level isolation signal at the SoC level.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_isolation_rule* (Mandatory)
- *update_isolation_rules* (Optional)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_isolation_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)

- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_isolation* (Mandatory)
- *set_isolation_control* (Mandatory)
- *map_isolation_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when the isolation cell <*iso-cellname*> instantiated as <*inst-name*> has an unconnected enable pin <*pin-name*>:

[LPISO01_1][WARNING] Isolation cell '<iso-cell-name>' instantiated as '<inst-name>' has unconnected enable pin '<pin-name>'

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the isolation cell <*iso-cellname*> instantiated as <*inst-name*> has an unconnected input pin <*pin-name*>:

[LPISO01_2][WARNING] Isolation cell '<iso-cell-name>' instantiated as '<inst-name>' has unconnected input pin '<pin-name>'

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when the isolation cell <*iso-cellname*> instantiated as <*inst-name*> has an undriven input pin <*pin-name*>:

[LPISO01_3][WARNING] Isolation cell '<iso-cell-name>'

instantiated as '<inst-name>' has undriven input pin '<pin-name>'

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears when the isolation cell *<iso-cellname>* instantiated as *<inst-name>* has an unconnected output pin *<pin-name>*:

[LPISO01_4][WARNING] Isolation cell '<iso-cell-name>' instantiated as '<inst-name>' has unconnected output pin '<pin-name>'

For debugging information, click *How to Debug and Fix*.

Message 5

The following message appears when the isolation cell *<iso-cellname>* instantiated as *<inst-name>* has an unloaded output pin *<pin-name>*:

[LPISO01_5][WARNING] Isolation cell '<iso-cell-name>' instantiated as '<inst-name>' has unloaded output pin '<pin-name>'

For debugging information, click *How to Debug and Fix*.

Message 6

The following message appears when the signal, on which isolation strategy is present, is undriven:

[LPISO01_6][WARNING] Signal '<signal-name>' present in the path of isolation strategy '<isolation-strategy>' is undriven

For debugging information, click *How to Debug and Fix*.

Message 7

The following message appears when the signal, on which isolation strategy is present, is unloaded:

[LPISO01_7][WARNING] Signal '<signal-name>' present in the path of isolation strategy '<isolation-strategy>' is unloaded

For debugging information, click *How to Debug and Fix*.

Message 8

The following message appears if no reference top-level isolation signal is found while traversing isolation enable signal for an isolation strategy specified on a domain boundary:

```
[LPISO01_8][ERROR] Signal ‘<signal-name>’ reaching isolation control signal ‘<isolation-signal-name>’ defined for isolation strategy ‘<isolation-strategy-name>’ is not defined as a reference top level isolation signal
```

For debugging information, click *How to Debug and Fix*.

Message 9

The following message appears if a reference top-level isolation signal is found while traversing isolation enable signal, and the domain boundary on which the isolation strategy was applied is driven by an instance that is powered by a supply other than the one mapped using the *reference_toplevel_isolation_signal* constraint:

```
[LPISO01_9][ERROR] Supply ‘<supply-name-from-constraint>’ specified for reference top level isolation signal ‘<signal-name>’ does not match the source supply ‘<source-supply-name>’ for domain boundary port ‘<name-of-domain-boundary-port>’, on which isolation strategy ‘<strategy-name>’ is applicable
```

For debugging information, click *How to Debug and Fix*.

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The following lists the consequences of not fixing:

- **Message 1**: The enable pin is left unconnected. This results in a design failure because the isolation cell will never get enabled.
- **Message 2**: The input pin is left unconnected. This makes the isolation cell useless in the design.
- **Message 3**: The input pin is connected to an undriven net. This makes the isolation cell useless in the design.
- **Message 4**: The output pin is left unconnected. This makes the isolation cell useless in the design.

- **Message 5**: The output pin is connected to a hanging net, therefore it is unloaded. This makes the isolation cell useless in the design.
- **Message 6**: The signal, on which isolation strategy is present, is undriven. So, the isolation cell inserted corresponding to this strategy will be useless.
- **Message 7**: The signal, on which isolation strategy is present, is unloaded. So, the isolation cell inserted corresponding to this strategy will be useless.
- **Message 8**: If no reference isolation signal is reaching the isolation signal defined for a isolation strategy, the corresponding isolation cell that will be placed after instrumentation will fail to provide isolation. As a result, undefined values will be propagated at the output of the domain in power down conditions.
- **Message 9**: If incorrect isolation signal is reaching the isolation signal defined for a isolation strategy, the corresponding isolation cell that will be placed after instrumentation will fail to provide correct isolation. As a result, undefined values can be propagated at the output of the domain in power down conditions.

How to Debug and Fix

The violation is reported at the RTL level where isolation cells are yet to be inserted and reported at the first instance of the unused isolation cell.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To resolve these violation messages, ensure that:

- The enable pin of Isolation cell is connected.
- Input pin of Isolation cell is connected and accordingly driven
- Output pin of Isolation cell is connected and accordingly loaded
- The undriven signal is connected and accordingly driven
- The unloaded signal is connected and accordingly loaded
- The correct reference top level isolation signal is driving the isolation enable signal defined in the UPF
- The supply mentioned in the *reference_toplevel_isolation_signal* constraint matches with the source supply, on the path of an isolation strategy

Example Code and/or Schematic

Example 1

Consider the following example. A violation occurs when the output pin 'Y' of the isolation cell 'ISO_AND_EN' is unconnected:

```
Isolation cell 'ISO_AND_EN' instantiated as
'TOP.i_dss_iso.ISO1' has unconnected output pin 'Y'
```

The schematic is shown as follows:

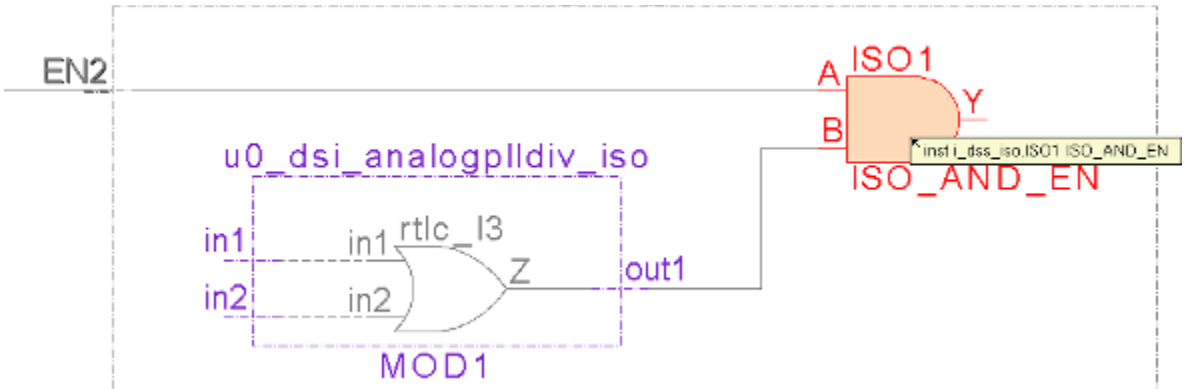


FIGURE 33. Incremental schematic

The *schematic* highlights the isolation cell, where the output pin is unconnected.

Example 2

Consider the following example. The output pin Y, on which isolation strategy is present, is unloaded.

```
Signal 'TOP.inst1.low_buf1.Y' present in the path of isolation
strategy 'ISO1' is unloaded
```

The schematic is shown as follows:

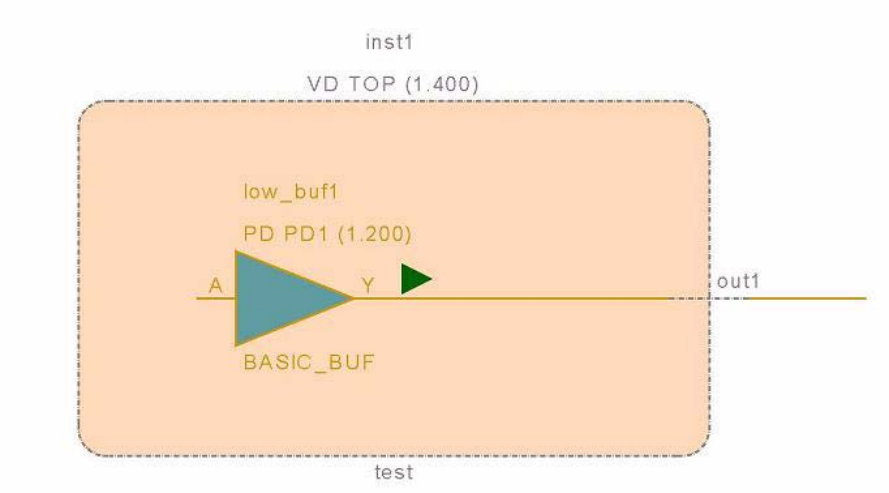


FIGURE 34. Incremental schematic

Example 3

Consider the following example. The input pin A, on which isolation strategy is present, is undriven.

```
Signal 'TOP.inst1.low_buf1.A' present in the path of isolation
strategy 'ISO1' is undriven
```

The schematic is shown as follows:

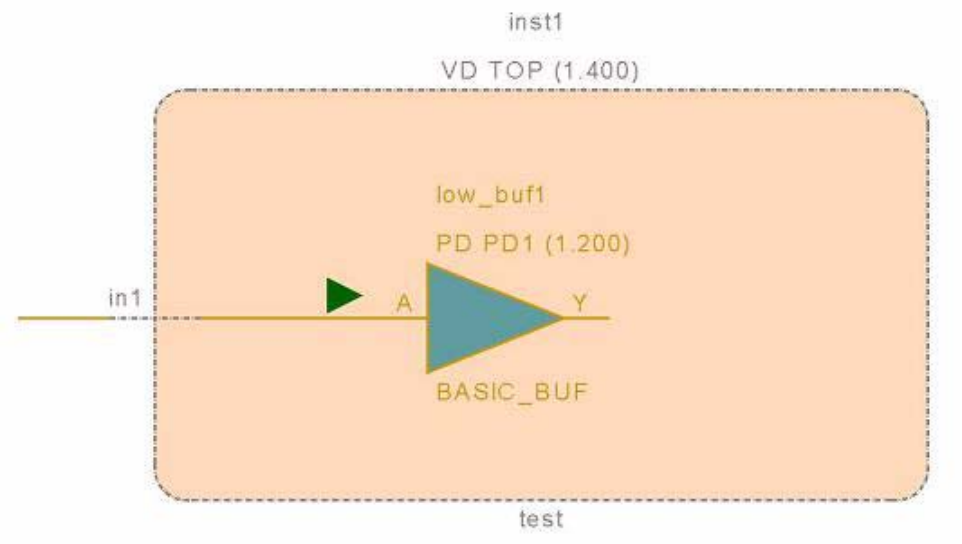


FIGURE 35. Incremental schematic

Example 4

Consider the following example.

SGDC:

```
current_design top
power_data -format upf -file const.upf

reference_toplevel_isolation_signal -name top.iso1 -supply
blk1/VDD1
```

UPF:

```
upf_version 2.0
```

```
set_design_top top
...
set_isolation iso1 -domain PD1 -elements {blk3/inst1/out1} -
isolation_signal iso -isolation_supply_set ss -clamp_value 0
-isolation_sense low
```

In the above example, since no reference isolation signal is reaching the `top.iso` signal, the following violation message is reported:

```
Signal 'top.iso' reaching isolation control signal 'top.iso'
defined for isolation strategy 'iso1' is not defined as a
reference top level isolation signal
```

The schematic is shown as follows:

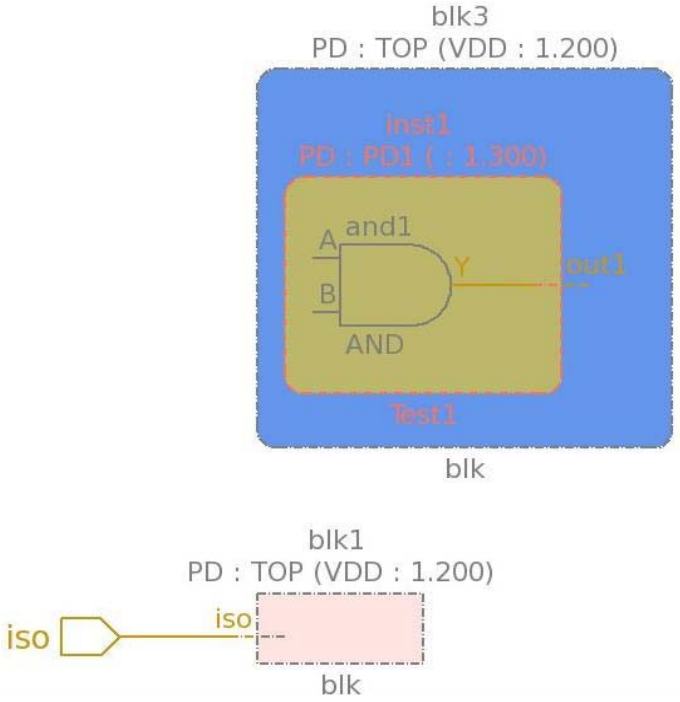


FIGURE 36. Incremental schematic

Example 5

Consider the following example.

SGDC:

```
current_design top
power_data -format upf -file const.upf
reference_toplevel_isolation_signal -name top.iso -supply
blk1/VDD1
```

UPF:

```
upf_version 2.0

set_design_top top
...
set_isolation iso1 -domain PD1 -elements {blk3/inst2/in1} -
isolation_signal iso -isolation_supply_set ss -clamp_value 0
-isolation_sense low
```

In the above example, since the supply provided in the *reference_toplevel_isolation_signal* SGDC constraint does not match with the supply associated with of the driver of the node where isolation strategy is applied, the following violation message is reported:

```
Supply '/blk1/VDD1' specified for reference top level isolation
signal 'top.iso' does not match the source supply '/blk3/VDD1'
for domain boundary port 'top.blk3.inst2.in1', on which
isolation strategy 'iso1' is applicable
```

The schematic is shown as follows:

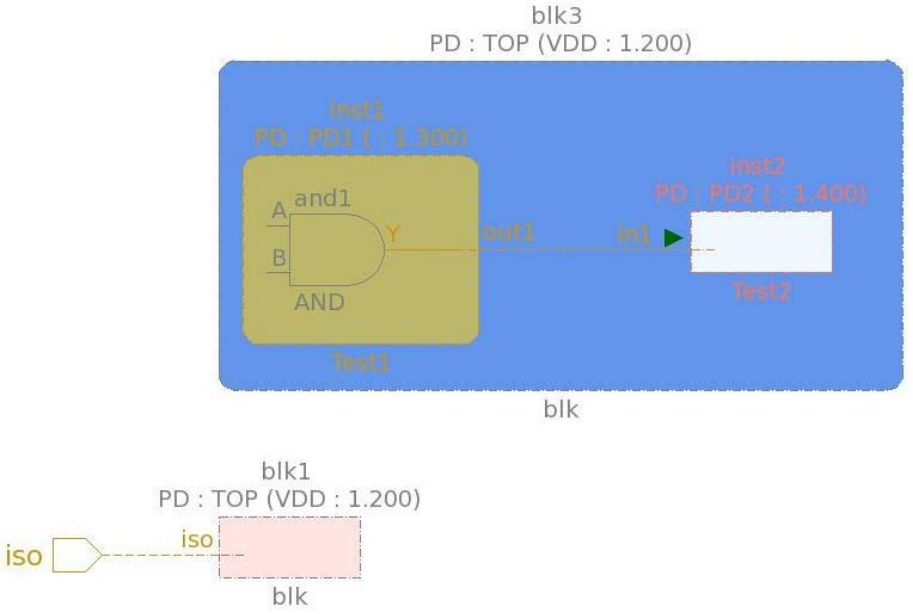


FIGURE 37. Incremental schematic

Default Severity Label

Warning, Error

Rule Group

Isolation Logic

Reports and Related Files

None

LPISO02

Checks isolation cell location in the design on the basis of its supply pins

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPISO02* checks the location of the isolation cells used in the design.

Normal isolation cells, which have primary power pins, should be placed in the relatively always-on domain only because in a relatively off domain, they will also be without power when the domain is powered off. Whereas, for the dual-rail isolation cells, which have backup as well as primary power pins, it is inefficient to place them in the always-on domain because the backup power/ground pins requires special routing.

A cell with *is_isolation_cell* attribute set as `true` in the library is recognized as an isolation cell.

Prerequisites

By default, the *LPISO02* rule is not run while running the SpyGlass Power Verify solution. To enable this rule, select the *LPISO02* rule in the Atrenta Console GUI.

Rule Exceptions

The *LPISO02* rule ignores:

- The isolation cells highlighted by the *LPSVM22* rule as out of place.

Language

VHDL, Verilog

Parameter(s)

- *lp_disable_lib_attr_read*: Default value is `0`. This signifies that the *LPISO02* rule reads the power related library attributes from the library. Set the value to `1` to provide the information from SGDC, UPF, or CPF files.
- *lp_skip_buf:* Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage domains in the design.
- *isolation_cell* (Optional): Use to specify the isolation cells in power domains.
- *input_isocell* (Optional): Use to specify the isolation cells at inputs of a power domain.
- *power_state* (Optional): Use to specify the combinations of domain states which can exist at the same time during a design operation.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_isolation_rule* (Mandatory)
- *update_isolation_rules* (Optional)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_isolation_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)

- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_isolation* (Mandatory)
- *set_isolation_control* (Mandatory)
- *map_isolation_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when the dual-rail isolation cell *<iso-cellname>* instantiated as *<inst-name>* is present in the relatively always-on domain *<dom-name>*:

[LPISO02_1][WARNING] Dual rail isolation cell '<iso-cell-name>'('<inst-name>') should not be present in relatively always on domain '<dom-name>'

Potential Issues

The violation message explicitly state the potential issues.

Consequences of Not Fixing

There is no need to place dual rail isolation cells in relatively always ON domains. They will add to power consumption in the design. They should be replaced by normal isolation Cells.

How to Debug and Fix

To fix this violation, ensure that the normal isolation cells are placed in the relatively always-on domain.

Message 2

The following message appears when the crossing between the source and destination has the normal isolation cell *<iso-cell-name>*(*<inst-name>)* present in relatively off domain *<domain-name>*:

[LPISO02_2][WARNING] Crossing between source '<source-name>'('<source-domain>') and destination '<destination-name>'('<destination-domain>') has normal isolation cell '<iso-

```
cell-name>'('<inst-name>') present in relatively off domain
'<domain-name>'
```

Potential Issues

The violation message explicitly state the potential issues.

Consequences of Not Fixing

The consequences of not fixing these violations is as follows:

- There is no need to place dual rail isolation cells in relatively always ON domains. They will add to power consumption in the design. They should be replaced by normal isolation Cells.
- It is irrelevant to place normal isolations cells in relatively OFF domain because they will be OFF when isolation is needed. This leads to design failure. They should be replaced by dual rail isolation cells.

How to Debug and Fix

The violation is reported at the first place where the incorrectly placed isolation cell is instantiated.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. The schematic highlights the incorrectly placed isolation cell in the design. Refer to the Example Code and/or Schematic section for an example.

To fix this violation message, ensure that:

- **Message 1**: The normal isolation cells are placed in the relatively always-on domain.
- **Message 2**: The dual-rail isolation cells are placed in the switched domain.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_supply_port VDDH
add_port_state VDDH -state {active 1.2} -state {off_state
off}
create_supply_net  VDDH -domain HIGH
connect_supply_net VDDH -ports VDDH
create_supply_port VDDL
add_port_state VDDL -state {active 0.8}
```

```
create_supply_net  VDDL -domain LOW
connect_supply_net VDDL -ports VDDL
set_domain_supply_net LOW  -primary_power_net VDDL -
primary_ground_net VSS
set_domain_supply_net HIGH -primary_power_net VDDH -
primary_ground_net VSS
```

For the above example, the *LPISO02* rule reports the following violation message:

```
Crossing between source 'TOP.low_domain.low_buf1.Y'('LOW(supply
VDDL:0.800)') and destination 'TOP.out2'('LOW(supply
VDDL:0.800)') has normal isolation cell
'ISO_AND_EN'('TOP.high_domain.iso2') present in relatively off
domain 'HIGH(supply VDDH:1.200)'
```

The schematic is as follows:

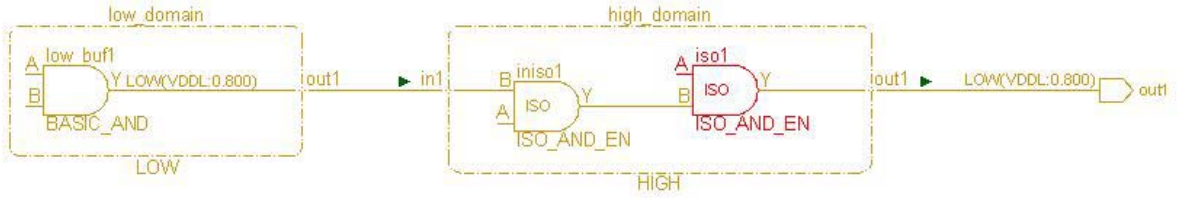


FIGURE 38. Incremental schematic

Default Severity Label

Warning

Rule Group

Isolation Logic

Reports and Related Files

None

LPISO03

Checks the presence of isolation cell at output terminals of power-domain

NOTE: *The LPISO03 will be deprecated in a future release.It is recommended to use the LPSVM08B and LPSVM08C rules instead.*

Language

Verilog, VHDL

Rule Description

The LPISO03 rule flags if an isolation cell is not present at the output crossing of a power domain.

The LPISO03 rule is divided into two sub-rules - *LPISO03A* and *LPISO03B*. When you select the LPISO03 rule, both these rules are run.

The set of constraints and parameters specified below are used by both the sub-rules (LPISO03A and LPISO03B).

Constraints

- (Mandatory) *voltage_domain*: The LPISO03 rule requires you to specify the power domains with the voltage_domain constraint and the name and value of the isolation signal using the -isosig argument and -isoval argument respectively in a SpyGlass Design Constraints file. When a power domain is shut-off, the outputs of the block should be isolated through isolation logic.
- (Optional) *ignore_crossing*: To specify the power domain-to-voltage domain crossings and power domain-to-power domain crossings to be ignored with the ignore_crossing constraint.
- (Optional) *power_state*: You can specify the power domain relationships and automatically find all the ignore_crossing constraints with the power_state constraint.
- (Optional) *assume_path*
- (Optional) *always_on_buffer*: You can specify the names of always-on buffers using the always_on_buffer constraint.

- (Optional) *set_case_analysis*: You can specify the case analysis conditions used by the LPISO03 rule using set_case_analysis constraint.

Rule Parameters

- By default, the LPISO03 rule ignores outputs, which are not driven, of power domains. Set the *lp_flag_undriven_nets* rule parameter to have the LPISO03 rule consider such outputs.
- By default, the LPISO03 rule ignores unconnected outputs of power domains. Set the *lp_flag_unconnected_nets* rule parameter to have the LPISO03 rule consider such outputs.
- By default, the LPISO03 rule flags the outputs of power domain that are directly connected to primary ports. Set the *lp_flag_pd_outputs* rule parameter to 0 to ignore such outputs.
- By default, the *lp_skip_buf* parameter is set to 1 and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to 0 to consider SpyGlass-generated buffers during rule checking.
- By default, the LPISO03 rule ignores nets connected to power/ground supply. Set the *lp_skip_pwr_gnd* rule parameter to 0 to process such nets also.
- *lp_skip_aon_buf*: Default value is 1. Set the parameter to 0 to not skip always-on buffers to find an isolation crossing.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_always_on_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)

- *create_supply_set* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *create_pst* (Optional)
- *add_pst_state* (Optional)
- *map_isolation_cell* (Optional)

Recommended for

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

LPISO03A

Reports missing isolation cell at output terminals of power domain, not having any isolation strategy

When to Use

The LPISO03A will be deprecated in a future release. It is recommended to use the *LPSVM08B* rule instead.

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPISO03A* rule reports for signal crossings, which have neither an isolation cell nor an isolation strategy.

Prerequisites

The *LPISO03A* rule does not run while running the SpyGlass Power Verify policy. To enable this rule, select the *LPISO03A* rule in the Atrenta Console GUI.

Language

Verilog, VHDL

Parameter(s)

- *lp_skip_feedthrough_buffer*: Default value is `0`. Set the value to `1` to skip feedthrough buffers.
- *lp_skip_iso_check_on_ground*: Default value is `no`. Set this parameter to `yes` to not report missing isolation strategy and missing isolation cells on the ground nets.
- *lp_skip_blackbox_checking*: Default value is `0`. Set the parameter to `1` to skip checking for black boxes.
- *lp_skip_aon_buf*: Default value is 1. Set the parameter to 0 to not skip always-on buffers to find an isolation crossing.

- *lp_check_pwr_gnd_to_macro_without_prd*: Default value is `yes`. Set this parameter to `no` to ignore crossings between tie-high/low and macro cell (`is_macro_cell: true`)/pad cell (`pad_cell: true`), without domain boundary. Level shifter and isolation checking for these crossings at the RTL and Netlist levels will be ignored.
- Refer to the *LPISO03* rule for the list of other parameters.

Constraint(s)

Refer to the *LPISO03* rule for the list of constraints.

Messages and Suggested Fix

The following message appears when an isolation cell is not found at the output terminal <*term-name*> of the power domain <*domain-info*>:

[LPISO03A_1][WARNING] Output signal '<signal-name>' going from source '<source-name>' (power domain '<pd-name>') to destination '<dest-name>' (domain 'dom-name') does not have any isolation cell placed in its path but need isolation

If the destination pin has been used in the `isolation_enable_condition` attribute as shown below, the violation message is changed to reflect that:

```
pin(d) {
  is_isolated : true;
  isolation_enable_condition : “en1 * en2”;
}
```

In the above example, the pin en1 (supply S2) is driven by buffer that is powered by supply S1. S1 is less on than S2, therefore output isolation is required. This scenario is reported in the following violation message:

[LPISO03A_2][WARNING] Output signal '<signal-name>' going from source '<source-name>' (power domain '<pd-name>') to destination '<dest-name>' (domain '<domain-name>') needs isolation. Destination pin is used in 'isolation_enable_condition' attribute

In CPF and SGDC flow, the domain information (<*domain-info*>) shows the domain name and its voltage value. However, in UPF flow, the domain

information shows the domain name along with the associated supply name and supply value.

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not Fixing

If no isolation strategy or isolation cell is provided, the crossing will result in design failure.

How to Debug and Fix

The violation is reported at the first place where the output signal of power domain is used.

Double-click the message and click the **Incremental Schematic** button. The schematic shows the output terminal and the power domain. Refer to the Example Code and/or Schematic section for an example.

To fix the violation, ensure that a valid isolation strategy is present at the power domain boundary or an isolation cell is present at the output terminals of the power domain, which does not have any isolation strategy.

Double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

Example Code and/or Schematic

In the following example, a violation occurs when no isolation cell is present at the output terminal `top.pd_inst.out2` of the power domain *shutOFF* and no isolation strategy is specified for `top.pd_inst.out2`:

```
Output signal 'top.pd_inst.out2' going from source
'top.pd_inst.buf2.Y' (power domain 'shutOFF(supply
VDD_LP:1.20)') to destination 'top.out2' (domain 'top(supply
VDD:1.20)') does not have any isolation cell placed in its path
but need isolation
```

The schematic is as follows:

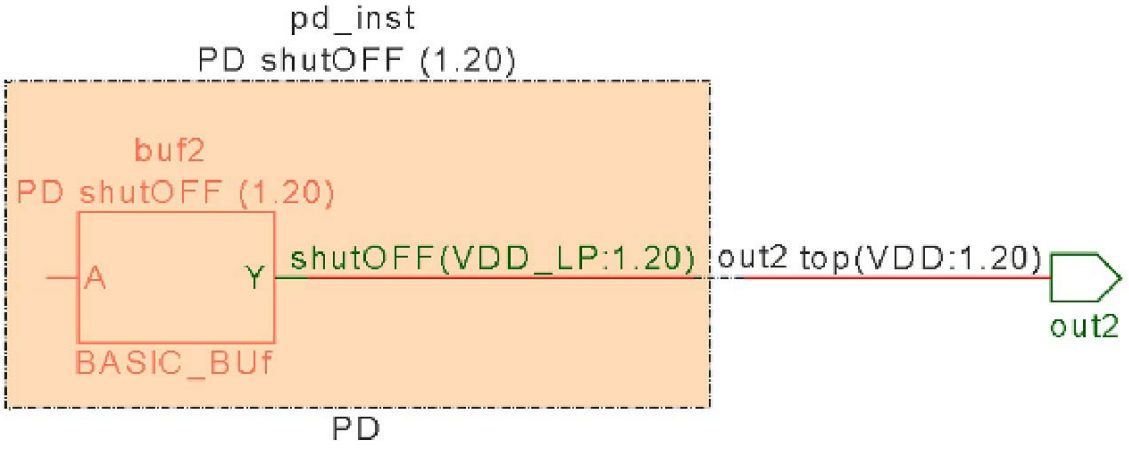


FIGURE 39. Incremental schematic

Default Severity Label

Warning

Rule Group

Isolation Logic

Reports and Related Files

None

LPISO03B

Reports non-existence of isolation cell at excluded output terminals of power domain

When to Use

The LPISO03B rule will be deprecated in a future release. It is recommended to use the *LPSVM08C* rule instead.

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPISO03B* rule reports when a signal crossing from a power domain output to another domain is defined as excluded by specifying `no_isolation` at the power domain boundary, but needs isolation, and it does not have an isolation cell present.

Prerequisites

The *LPISO03B* rule does not run while running the SpyGlass Power Verify policy. To enable this rule, select the *LPISO03B* rule in the Atrenta Console GUI.

Language

Verilog, VHDL

Parameter(s)

- *lp_skip_iso_check_on_ground*: Default value is `no`. Set this parameter to `yes` to not report missing isolation strategy and missing isolation cells on the ground nets.
- *lp_skip_blackbox_checking*: Default value is `0`. Set the parameter to `1` to skip checking for black boxes.
- *lp_skip_aon_buf*: Default value is 1. Set the parameter to 0 to not skip always-on buffers to find an isolation crossing.

- *lp_check_pwr_gnd_to_macro_without_prd*: Default value is `yes`. Set this parameter to `no` to ignore crossings between tie-high/low and macro cell (`is_macro_cell: true`)/pad cell (`pad_cell: true`), without domain boundary. Level shifter and isolation checking for these crossings at the RTL and Netlist levels will be ignored.
- Refer to the *LPISO03* rule for the list of parameters.

Constraint(s)

Refer to the *LPISO03* rule for the list of constraints.

Messages and Suggested Fix

The following message appears when an isolation cell is not found at the excluded output terminal *<term-name>* of the power-domain *<domain-info>*:

[LPISO03B_1][INFO] Excluded output '<term-name>' going from source '<source-name>' (power domain '<pd-name>') to destination '<dest-name>' (domain '<dom-name>') does not have isolation cell placed in its path but need isolation

In CPF and SGDC flow, the domain information (*<domain-info>*) shows the domain name and its voltage value. However, in UPF flow, the domain information shows the domain name along with the associated supply name and supply value.

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not Fixing

The strategy provided at the power domain boundary has `no_isolation` specified, even though isolation is needed and an isolation cell should be placed. This results in design failure.

How to Debug and Fix

The violation is reported at the first place where the output signal of power domain is used.

Double-click the message and click the **Incremental Schematic** button. The schematic shows the output terminal and the power domain. Refer to the Example Code and/or Schematic section for an example.

To fix the violation, ensure that a valid isolation strategy without `no_isolation` is specified at the relevant power domain boundary or an isolation cell is present at the excluded output terminal of the power domain.

Double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

Example Code and/or Schematic

In the following example, a violation occurs when no isolation cell is present at the output terminal `top.pd_inst.out3` of the power domain *shutOFF* and `top.pd_inst.out3` is defined as excluded:

```
Excluded output 'top.pd_inst.out3' going from source
'top.pd_inst.buf3.Y' (power domain 'shutOFF(supply
VDD_LP:1.20)') to destination 'top.out3' (domain 'top(supply
VDD:1.20)') does not have isolation cell placed in its path but
need isolation
```

The schematic is as follows:

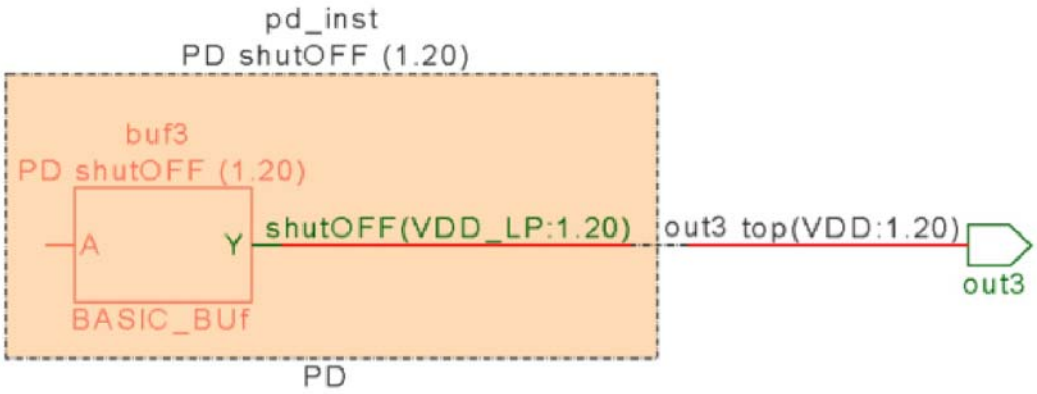


FIGURE 40. Incremental schematic

Default Severity Label

Info

Rule Group

Isolation Logic

Reports and Related Files

None

LPISO04

Reports missing and incorrect isolation strategies

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPISO04* rule reports and checks missing and incorrect isolation strategies specified on domain boundaries.

This rule has the following subrules:

- *LPISO04A*: Reports missing isolation strategy at power domain output ports.
- *LPISO04B*: Reports incorrect isolation strategy at power domain output ports.
- *LPISO04C*: Checks isolation strategy for incorrect isolation_power_net or isolation_ground_net.
- *LPISO04D*: Reports standard isolation cell that cannot be implemented.

NOTE: *If the set_port_attribute -repeater_supply or set_repeater –repeater_supply_set is specified on a port in the UPF, then this rule assumes the presence of an implicit repeater at RTL and performs the checking accordingly.*

LPISO04A

Reports missing isolation strategy at power domain output ports

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPISO04A* rule reports a violation for missing isolation strategy at domain boundaries lying in a power domain crossing. An isolation strategy can be present either at source side domain boundary or at destination side domain boundary. This rule also checks crossings with no domain boundary present.

This rule does not report a violation if the *is_isolated* attribute is set at the destination side of the power domain crossing.

The rule also reports information of the isolation cell present in a crossing if the *lp_flag_iso_cell_in_crossing* parameter is enabled.

Language

Verilog, VHDL

Parameter(s)

- *lp_flag_undriven_nets*: Default value is `0`. Set the value to `1` to consider outputs of power domains that are not driven.
- *lp_flag_unconnected_nets*: Default value is `0`. Set the value to `1` to consider unconnected outputs of power domains.
- *lp_skip_buf*: Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.

- *lp_skip_pwr_gnd*: Default value is 1. Set the value to 0 to consider nets connected to power/ground supply.
- *lp_flag_pd_outputs*: Default value is 1. Set the value to 0 to ignore the outputs of power domains that are directly connected to primary ports.
- *lp_skip_feedthrough_buffer*: Default value is 0. Set the value to 1 to skip feedthrough buffers.
- *lp_skip_iso_check_on_ground*: Default value is no. Set this parameter to yes to not report missing isolation strategy and missing isolation cells on the ground nets.
- *lp_check_pwr_gnd_to_macro_without_prd*: Default value is yes. Set this parameter to no to ignore crossings between tie-high/low and macro cell (is_macro_cell: true)/pad cell (pad_cell: true), without domain boundary. Level shifter and isolation checking for these crossings at the RTL and Netlist levels will be ignored.
- *lp_skip_aon_buf*: Default value is 1. Set the parameter to 0 to not skip always-on buffers to find an isolation crossing.
- *lp_flag_iso_cell_in_crossing*: Default value is no. Set this parameter to yes to get information related to isolation cells, if an isolation cell is present in the crossing even though there is no strategy.
- *lp_skip_blackbox_checking*: Default value is 0. Set the parameter to 1 to skip checking for black boxes.
- *lp_skip_lib_buf*: Default value is 0. Set the parameter to 1 to skip the instances of library buffers while checking for crossings.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the power domains with the *voltage_domain* constraint and the name and value of the isolation signal using the isosig and isoval arguments in a SpyGlass Design Constraints file.
- *ignore_crossing* (Optional): Use to specify the power domain to voltage domain crossings and power domain to power domain crossings to be ignored with the *ignore_crossing* constraint.

- *power_state* (Optional): Use to specify the power domain relationships and automatically find all the *ignore_crossing* constraints with the *power_state* constraint.
- *assume_path* (Optional): Use to specify the paths that exist between the input pins and the output pins of black boxes.
- *always_on_buffer* (Optional): Use to specify the names of always-on buffers using the *always_on_buffer* constraint.
- *set_case_analysis* (Optional): Use to specify the case analysis conditions.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_always_on_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *map_isolation_cell* (Mandatory)
- *create_pst* (Mandatory)
- *add_pst_state* (Mandatory)
- *set_repeater* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when no isolation strategy is specified for the signal crossing from <*port-name1*> of <*dom-name1*> to <*port-name2*> of <*dom-name2*>:

[LPISO04A_1][ERROR] Crossing from source <port-name1> (domain '<dom-name1>') to destination <port-name2> (domain '<dom-name2>') does not have any isolation strategy specified

When the *lp_flag_iso_cell_in_crossing* parameter is set, the following message is reported if an isolation cell is present in the crossing:

[LPISO04A_7][ERROR] Crossing from source <port-name1> (domain '<dom-name1>') to destination <port-name2> (domain '<dom-name2>') has an isolation cell '<cell-hier-name (LIB CELL NAME)>' in '<source/destination/parent >' location with no isolation strategy specified

If the destination pin has been used in the `isolation_enable_condition` attribute as shown below, the violation message is changed to reflect that:

```
pin(d) {
  is_isolated : true;
  isolation_enable_condition : "en1 * en2";
}
```

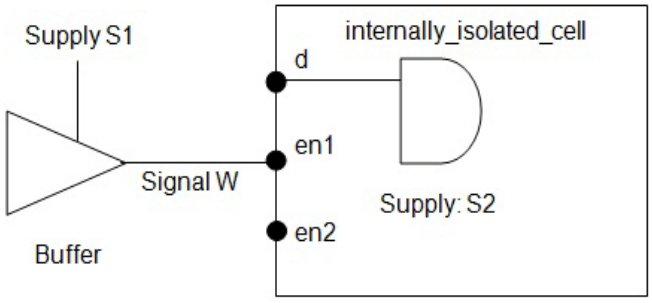


FIGURE 41. Isolation strategy

In the above example, the pin en1 (supply S2) is driven by buffer that is powered by supply S1. S1 is less on than S2, therefore output isolation is required. This scenario is reported in the following violation message:

[LPISO04A_4][ERROR] Crossing from source <port-name1> (domain '<dom-name1>') to destination <port-name2> (domain '<dom-

name2>') does not have any isolation strategy specified. Destination pin is used in ‘isolation_enable_condition’ attribute

When the *lp_flag_iso_cell_in_crossing* parameter is set, the following message is reported if an isolation cell is present in the crossing:

[LPISO04A_10][ERROR] Crossing from source <port-name1> (domain '<dom-name1>') to destination <port-name2> (domain '<dom-name2>') has an isolation cell ’<cell-hier-name (LIB CELL NAME)>’ in ‘<source/destination/parent >’ location with no isolation strategy specified. Destination pin is used in 'isolation_enable_condition' attribute

For debugging informations, click *How to Debug and Fix*.

Message 2

The following message appears when no isolation strategy is specified and a domain boundary is not present in the path of the signal crossing from <*port-name1*> of <*dom-name1*> to <*port-name2*> of <*dom-name2*>:

[LPISO04A_2][ERROR] Crossing from source <port-name1> (domain '<dom-name1>') to destination <port-name2> (domain '<dom-name2>') needs isolation but no domain boundary port is present in the path.

When the *lp_flag_iso_cell_in_crossing* parameter is set, the following message is reported if an isolation cell is present in the crossing:

[LPISO04A_8][ERROR] Crossing from source <port-name1> (domain '<dom-name1>') to destination <port-name2> (domain '<dom-name2>') has an isolation cell ’<cell-hier-name (LIB CELL NAME)>’ in ‘<source/destination/parent >’ location. Crossing needs isolation but strategy cannot be defined as all domain boundary ports in the path are of inout type

If the destination pin has been used in the `isolation_enable_condition` attribute, the violation message is changed, as follows:

[LPISO04A_5][ERROR] Crossing from source <port-name1> (domain '<dom-name1>') to destination <port-name2> (domain '<dom-name2>') needs isolation but no domain boundary port is present in the path. Destination pin is used in

‘isolation_enable_condition’ attribute

Refer to the example and diagram given in the *Message 1*, above, to know more about this scenario.

When the *lp_flag_iso_cell_in_crossing* parameter is set, the following message is reported if an isolation cell is present in the crossing:

[LPISO04A_11][ERROR] Crossing from source <port-name1> (domain '<dom-name1>') to destination <port-name2> (domain '<dom-name2>') has an isolation cell ’<cell-hier-name (LIB CELL NAME)>’ in ‘<source/destination/parent >’ location. Crossing needs isolation but strategy cannot be defined as all domain boundary ports in the path are of inout type. Destination pin is used in 'isolation_enable_condition' attribute

For debugging informations, click *How to Debug and Fix*.

Message 3

The following message appears when no isolation strategy is specified because all domain boundary port(s) in the path is (are) of inout type:

[LPISO04A_3][ERROR] Crossing from source '<source-hier-name>' (domain '<domain-and-supply-name>') to destination '<destination-hier-name>' (domain '<domain-and-supply-name>') needs isolation but strategy cannot be defined as all domain boundary port(s) in the path is(are) of inout type

When the *lp_flag_iso_cell_in_crossing* parameter is set, the following message is reported if an isolation cell is present in the crossing:

[LPISO04A_9][ERROR] Crossing from source <port-name1> (domain '<dom-name1>') to destination <port-name2> (domain '<dom-name2>') has an isolation cell ’<cell-hier-name (LIB CELL NAME)>’ in ‘<source/destination/parent >’ location. Crossing needs isolation but no domain boundary port is present in the path

If the destination pin has been used in the `isolation_enable_condition` attribute, the violation message is changed, as follows:

[LPISO04A_6][ERROR] Crossing from source '<source-hier-name>' (domain '<domain-and-supply-name>') to destination '<destination-hier-name>' (domain '<domain-and-supply-name>') needs isolation but strategy cannot be defined as all domain

boundary port(s) in the path is(are) of inout type. Destination pin is used in ‘isolation_enable_condition’ attribute

Refer to the example and diagram given in the *Message 1*, above, to know more about this scenario.

When the *lp_flag_iso_cell_in_crossing* parameter is set, the following message is reported if an isolation cell is present in the crossing:

[LPISO04A_12][ERROR] Crossing from source <port-name1> (domain '<dom-name1>') to destination <port-name2> (domain '<dom-name2>') has an isolation cell ’<cell-hier-name (LIB CELL NAME)>’ in ‘<source/destination/parent >’ location. Crossing needs isolation but no domain boundary port is present in the path. Destination pin is used in 'isolation_enable_condition' attribute

For debugging informations, click *How to Debug and Fix*.

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not Fixing

If isolation strategy is missing at a power domain crossing where it is required, the power intent given by the user is incomplete and will result in design failure.

How to Debug and Fix

The violation is reported at the first place where the output signal of power domain is used.

Double-click the message and click the **Incremental Schematic** button. The schematic shows the signal crossing where no isolation strategy is specified. This rule supports IS Abstraction between start and end points and abstracts all logic between these points, except level shifter cells. Refer to the *Abstraction between Start and End Points* topic in the *Atrenta Console Reference Guide*. Refer to the Example Code and/or Schematic section for an example.

The *LPISO04A* rule also generates a file named LPISO04A.csv. This file contains all the messages generated by the *LPISO04A* rule. To view the spreadsheet, in the Message Tree window, click the **Open Spreadsheet** option in the right-click menu of the rule.

There should be a correct isolation strategy present for power domain boundaries in power domain crossings. To fix these violations, ensure that an isolation strategy is present at all power domain boundaries in crossings in your design.

Double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

Example Code and/or Schematic

Example 1

Consider the following UPF snippet:

```
create_power_domain TOP -include_scope
create_power_domain PD2 -elements u1_inst/u2_inst
create_power_domain PD4 -elements u3_inst/u4_inst
create_supply_port VDP2
add_port_state VDP2 -state {V2 1.4}  -state {VDP2_off off}
create_supply_net  VDP2 -domain PD2
connect_supply_net VDP2 -ports VDP2
create_supply_port VDP4
add_port_state VDP4 -state {V4 1.4}  -state {VDP4_off off}
create_supply_net  VDP4 -domain PD4
connect_supply_net VDP4 -ports VDP4
set_domain_supply_net PD2 -primary_power_net VDP2 -
primary_ground_net VSS
set_domain_supply_net PD4 -primary_power_net VDP4 -
primary_ground_net VSS
```

For the above example, the following violation is reported because no isolation cell strategy is specified for the signal crossing from `top.u1_inst.out1` to `top.u2_inst.in1`:

```
[ERROR] Crossing from source top.u1_inst.out1 (domain
'PD1(supply VDP1:0.950)') to destination top.u2_inst.in1
(domain 'PD2(supply VDP2:1.100)') does not have any isolation
strategy specified
```

The incremental schematic is displayed as follows. This is without IS abstraction.

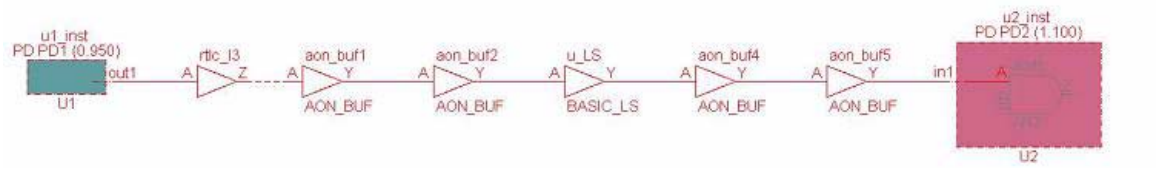


FIGURE 42. Incremental schematic without IS abstraction

This is with IS abstraction.

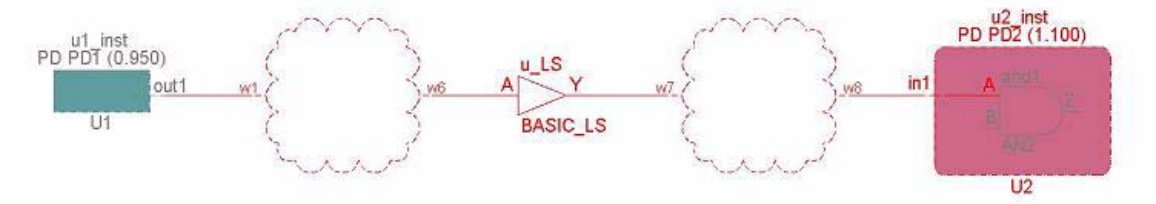


FIGURE 43. Incremental schematic with IS abstraction

Example 2

Consider the following UPF snippet:

Verilog:

```
module TOP(EN,fw_en, VDD, VSS);
input EN,fw_en, VDD, VSS
MACRO3 EBB(.PWREN(EN),.Y(w1), .Z(w2), .VDD(VDD), .VSS(VSS));
HIGH VNN_logic(.in(w1), .in_2(w2), .out(w4), .VDD(VDD),
.VSS(VSS));
    HIGH VNN_logic2(.in(w4), .in_2(w2), .out(), .VDD(VDD),
.VSS(VSS));
endmodule

module HIGH(in, in_2, EN, out, VDD, VSS);
input in, in_2, VDD, VSS, EN;
output out;
ISO1 iso_cell(.D(in_2),.EN(EN),.Y(w),.VDD(VDD),.VSS(VSS));
        BASIC_AND vnn_logic(.A(in),.B(w),.Y(out), .VDD(VDD),
```

```
.VSS(VSS));
Endmodule
```

UPF:

```
create_power_domain VD1 -include_scope
create_power_domain VD2 -elements { VNN_logic }

create_supply_port VDD
add_port_state VDD -state {on1 1.15} -state {off1 off}
create_supply_net VDD -domain VD1
create_supply_net VDD -domain VD2 -reuse
connect_supply_net VDD -ports { VDD EBB/VDD }
create_supply_port VDD1
add_port_state VDD1 -state {on1 1.15}
create_supply_net VDD1 -domain VD1
connect_supply_net VDD1 -ports {VDD1}

set_domain_supply_net VD1 -primary_power_net VDD -
primary_ground_net VSS
set_domain_supply_net VD2 -primary_power_net VDD1 -
primary_ground_net VSS
```

For the above example, the following violation is reported because no isolation strategy is specified for the signal crossing from `TOP.EBB.Z` to `TOP.VNN_logic.in_2`, whereas isolation cell, `TOP.VNN_logic.iso_cell` is inserted to isolate the crossing:

[ERROR] Crossing from source TOP.EBB.Z (domain 'VD1(supply VDD:1.150 and VSS)') to destination TOP.VNN_logic.in_2 (domain 'VD2(supply VDD1:1.150)') has an isolation cell 'TOP.VNN_logic.iso_cell (ISO1)' in 'destination' location with no isolation strategy specified

The incremental schematic is displayed as follows:

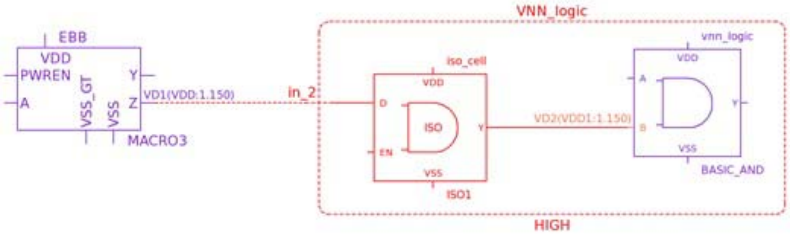


FIGURE 44. Incremental schematic

Example 3

Consider the following UPF snippet:

Verilog:

```
module top (input in,en1,iso,output out1,out2,out3);
Test1 inst1 (.in1(in),.out1(w),.out2(w1));
Test2 inst2 (.in1(),.in2(),.out1(out3));
endmodule

module Test1 (input in1,in2,en1,en2,en3,inout en4, output
out1,out2);
AND and1 (.A(in1),.B(in2),.Y(out1));
AND and2 (.A(in1),.B(in2),.Y(out2));
endmodule

module Test2 (input in1,in2, output out1);
AND and1 (.A(in1),.B(in2),.Y(w1));
AND and2 (.A(in1),.B(in2),.Y(w2));
ISO1 iso1(.D(w1), .EN(), .Y(w3));
AND9 and3
(.A(),.B(),.Y(out1),.enin1(w3),.enin2(w2),.enino1(w1),.enino
```

```
2(w2),.enout1(),.enout2());
endmodule
```

UPF:

```
upf_version 2.0
set_design_top top
create_power_domain TOP -include_scope
create_power_domain PD1 -elements {inst1}
create_power_domain PD2 -elements {inst2}
create_supply_port VDD
create_supply_net  VDD
connect_supply_net VDD -ports {VDD}
create_supply_port VDD1 -domain PD1
create_supply_net  VDD1 -domain PD1
connect_supply_net VDD1 -ports {VDD1 inst2/and1/VDD inst2/
and2/VDD}
create_supply_port VDD2 -domain PD2
create_supply_net  VDD2 -domain PD2
connect_supply_net VDD2 -ports {VDD2}
create_supply_port VSS
create_supply_net VSS
create_supply_net VSS -domain PD1 -reuse
create_supply_net VSS -domain PD2 -reuse
connect_supply_net VSS -ports {VSS}


set_domain_supply_net TOP -primary_power_net VDD -
primary_ground_net VSS
set_domain_supply_net PD1 -primary_power_net VDD1 -
primary_ground_net VSS
```

```
set_domain_supply_net PD2 -primary_power_net VDD2 -
primary_ground_net VSS

add_port_state VDD -state {on1 1.0}
add_port_state VDD1 -state {on1 1.1} -state {off1 off}
add_port_state VDD2 -state {on1 1.0}
add_port_state VSS -state {def 0.00}

create_pst ps1 -supplies {VDD VDD1 VDD2 VSS}
add_pst_state s1 -pst ps1 -state {on1 on1 on1 def }
add_pst_state s2 -pst ps1 -state {on1 off1 on1 def }
```

The following is the snapshot of the AND9 cell:

```
cell (AND9) {
      pg_pin (VDD) { pg_type : primary_power; }
      pg_pin (VDDC) { pg_type : backup_power; }
      pg_pin (VSS) { pg_type : primary_ground; }
      pin (enin1) { direction : input;related_power_pin: VDD; related_ground_pin: VSS; }
      pin (enin2) { direction : input;related_power_pin: VDD; related_ground_pin: VSS; }
      pin (enino1) { direction : inout;related_power_pin: VDD; related_ground_pin: VSS; }
      pin (enino2) { direction : inout;related_power_pin: VDD; related_ground_pin: VSS; }
      pin (enout1) { direction : output;related_power_pin: VDD; related_ground_pin: VSS; }
      pin (enout2) { direction : output;related_power_pin: VDD; related_ground_pin: VSS; }
      pin (A) { direction : input;related_power_pin: VDD; related_ground_pin: VSS;is_isolated:true;isolation_
      enable_condition:enin1&enino1; }
      pin (B) { direction : inout;related_power_pin: VDD; related_ground_pin: VSS;is_isolated:true;isolation_
      enable_condition:enin1&enino1;  }
      pin (Y) { direction : output; function : "A&B";related_power_pin: VDD;related_ground_pin: VSS;
      is_isolated:true;isolation_enable_condition:enin1&enino1;   }
   }
```

FIGURE 45. Library view of the AND9 cell

For the above scenario, the following violation is reported because isolation is required from `top.inst2.and1.Y` to `top.inst2.and3.enin1`, but there is no domain boundary between source and destination. Also, the destination pin is used in isolation enable condition for pins A, B and Y of the AND9 cell. See the library description for the AND9 cell in the *Figure 45*:

[ERROR] Crossing from source top.inst2.and1.Y (domain 'PD2(supply VDD1:1.100)') to destination top.inst2.and3.enin1 (domain 'PD2(supply VDD2:1.000)') has an isolation cell 'top.inst2.iso1 (ISO1)' in 'source' location. Crossing needs isolation but no domain boundary port is present in the path. Destination pin is used in 'isolation_enable_condition' attribute

The incremental schematic is displayed as follows:

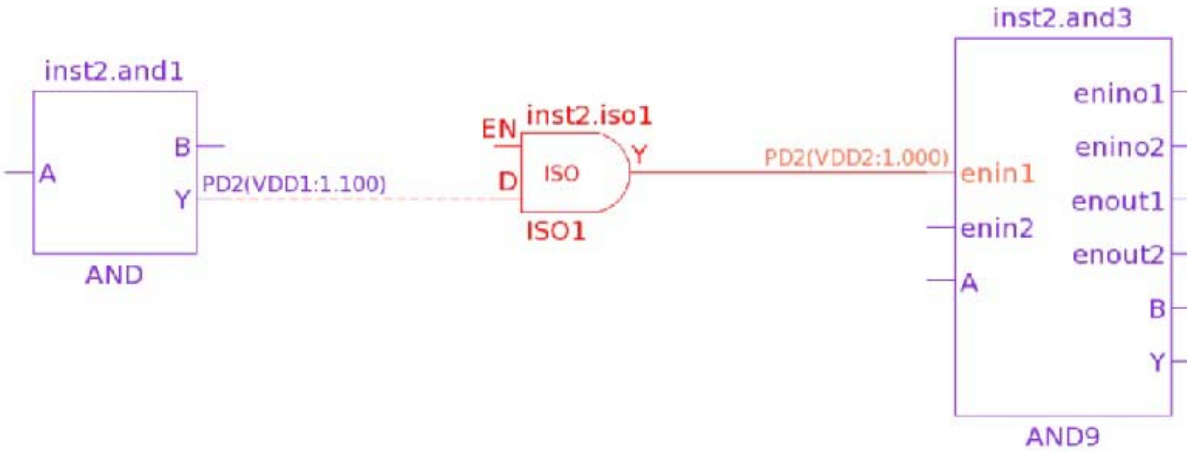


FIGURE 46. Incremental schematic

Default Severity Label

Error

Rule Group

Isolation Logic

Reports and Related Files

None

LPISO04B

Reports incorrect isolation strategy at power domain output ports

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPISO04B* rule reports a violation for a crossing if an incorrect isolation strategy is specified. An isolation strategy can be present at either power domain boundary ports.

This rule does not report a violation if the *is_isolated* attribute is set at the destination side of the power domain crossing.

Language

Verilog, VHDL

Parameter(s)

- *lp_flag_undriven_nets*: Default value is `0`. Set the value to `1` to consider outputs of power domains that are not driven.
- *lp_flag_unconnected_nets*: Default value is `0`. Set the value to `1` to consider unconnected outputs of power domains.
- *lp_skip_buf*: Default value is `1`. Set the value to `0` to consider buffers generated during design synthesis.
- *lp_skip_pwr_gnd*: Default value is `1`. Set the value to `0` to consider nets connected to power/ground supply.
- *lp_flag_pd_outputs*: Default value is `1`. Set the value to `0` to ignore the outputs of power domains that are directly connected to primary ports.
- *lp_skip_aon_buf*: Default value is 1. Set the parameter to 0 to not skip always-on buffers to find an isolation crossing.

- *lp_skip_blackbox_checking*: Default value is `0`. Set the parameter to `1` to skip checking for black boxes.
- *lp_skip_lib_buf*: Default value is `0`. Set the parameter to `1` to skip the instances of library buffers while checking for crossings.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the power domains with the *voltage_domain* constraint and the name and value of the isolation signal using the `isosig` and `isoval` arguments in a SpyGlass Design Constraints file.
- *ignore_crossing* (Optional): Use to specify the power domain to voltage domain crossings and power domain to power domain crossings to be ignored with the *ignore_crossing* constraint.
- *power_state* (Optional): Use to specify the power domain relationships and automatically find all the *ignore_crossing* constraints with the *power_state* constraint.
- *assume_path* (Optional): Use to specify the paths that exist between the input pins and the output pins of black boxes.
- *always_on_buffer* (Optional): Use to specify the names of always-on buffers using the *always_on_buffer* constraint.
- *set_case_analysis* (Optional): Use to specify the case analysis conditions.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_always_on_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)

- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *create_pst* (Optional)
- *add_pst_state* (Optional)
- *map_isolation_cell* (Optional)
- *set_repeater* (Optional)

Messages and Suggested Fix

The following message appears when `-no_isolation` option is specified for the <port-name3> of the signal crossing from <*port-name1*> of <*dom-name1*> to <*port-name2*> of <*dom-name2*>, which requires isolation:

[LPISO04B_1][WARNING] Crossing from source <port-name1> (domain '<dom-name1>') to destination <port-name2> (domain '<dom-name2>') needs isolation, has -no_isolation '<iso-strategy-name>' specified for '<port-name3>'

Potential Issues

This rule reports in following situations:

- Strategy at the destination has the `no_isolation` argument specified and source side strategy is not available.
- Strategy at source has `no_isolation` argument specified and destination side strategy is not available.
- Both the strategies on source and destination have `no_isolation` argument specified.

Consequences of not Fixing

The strategy specified is redundant because the `no_isolation` argument has been used, even though, isolation is required for the power domain crossing. If not fixed, this will lead to design failure.

How to Debug and Fix

Double-click the message and click the **Incremental Schematic** button. The schematic shows the signal crossing where isolation is needed and an

isolation strategy with the no_isolation argument is specified. Refer to the Example Code and/or Schematic section for an example.

The *LPISO04B* rule also generates a file named LPISO04B.csv. This file contains all the messages generated by the *LPISO04B* rule. To view the spreadsheet, in the Message Tree window, click the Open Spreadsheet option in the right-click menu of the rule.

There should be a correct isolation strategy present for power domain boundaries in crossings. To fix the violation, ensure that a correct isolation strategy is specified for all power domain boundaries in your design.

Double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_domain TOP -include_scope
create_power_domain PD2 -elements { u2_inst Pint }
create_supply_port VDTOP
add_port_state VDTOP -state {V12 1.2}
create_supply_net  VDTOP -domain TOP
connect_supply_net VDTOP -ports VDTOP
create_supply_port VDP2
add_port_state VDP2 -state {V32 1.1}  -state {off_state off}
create_supply_net  VDP2 -domain PD2
connect_supply_net VDP2 -ports VDP2
set_domain_supply_net TOP -primary_power_net VDTOP -
primary_ground_net VSS
set_domain_supply_net PD2  -primary_power_net VDP2 -
primary_ground_net VSS
set_isolation ISO3 -domain PD2   -no_isolation  -elements {
u2_inst/out1   u2_inst/out2  Pint/D[0] Pint/D[1] Pint/D[2] }
```

For the above example, the following violation is reported because the no_isolation parameter is specified for the signal crossing from top.u2_inst.out1 of PD2 to top.out1 of TOP:

```
Crossing from source top.u2_inst.out1[2:0] (domain 'PD2') to
destination top.out1[0:2] (domain 'TOP') needs isolation, has -
```

no_isolation specified for

The schematic is as follows:

Default Severity Label

Warning

Rule Group

Isolation Logic

Reports and Related Files

None

LPISO04C

Checks isolation strategy for incorrect isolation_power_net or isolation_ground_net

When to Use

Use this rule to check the validity of isolation power net and ground net specified for an isolation strategy. This is to ensure that the isolation logic corresponding to this strategy functions correctly.

This rule is applicable to all design phases after the insertion of isolation.

Refer to:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The `LPISO04C` rule reports a violation if the isolation strategy does not ensure the following conditions as per PST:

- **For output isolation**: The isolation cell should be relatively-on as per the sink or destination. The isolation strategy specified for output isolation on a domain boundary should have both `isolation_power_net` and `isolation_ground_net` such that they are either relatively-on or equivalent to the power/ground net of the sink/destination.
- **For input isolation**: The isolation cell should be relatively-on as per the source. The isolation strategy specified for input isolation on a domain boundary should have both `isolation_power_net` and `isolation_ground_net` such that they are either relatively-on or equivalent to the power/ground net of the source.

Parameter(s)

- *lp_flag_undriven_nets*: Set the value to 1 to have the rule consider outputs of power domains that are not driven.

- *lp_flag_unconnected_nets*: Set the value to 1 to have the rule consider unconnected outputs of power domains.
- *lp_skip_buf*: Default value is 1 and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to 0 to consider SpyGlass-generated buffers during rule checking.
- *lp_skip_pwr_gnd*: Set the value to 0 to have the rule consider nets connected to power/ground supply.
- *lp_flag_pd_outputs*: Set the value to 0 to ignore the outputs of power domain that are directly connected to primary ports.
- *lp_skip_aon_buf*: Default value is 1. Set the parameter to 0 to not skip always-on buffers to find an isolation crossing.
- *lp_skip_blackbox_checking*: Default value is 0. Set the parameter to 1 to skip checking for black boxes.
- *lp_skip_lib_buf*: Default value is 0. Set the parameter to 1 to skip the instances of library buffers while checking for crossings.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the power domains.
- *ignore_crossing* (Optional): Use to specify the power domain-to-voltage domain crossings and power domain-to-power domain crossings to be ignored.
- *power_state* (Optional): Use to specify the legal combinations of domain states (voltage values), that is those combinations of domain states that can exist at the same time during operation of the design.
- *assume_path* (Optional): Use to specify the paths that exist between the input and output pins of black boxes.
- *always_on_buffer* (Optional): Specify always-on buffers.
- *set_case_analysis* (Optional): Specify the case analysis conditions to be used.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_nominal_condition* (Mandatory)

- *create_power_mode* (Mandatory)
- *define_always_on_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_isolation* (Mandatory)
- *create_pst* (Mandatory)
- *add_pst_state* (Mandatory)
- *set_repeater* (Optional)

Messages and Suggested Fix

The following message appears when you have specified an incorrect isolation power net `<net-name>` on crossing from `<port-name1>` of `<dom-name1>` to `<port-name2>` of `<dom-name2>`, which requires isolation `<iso-name>`:

[LPISO04C_1][WARNING] Incorrect isolation <power | ground> net '<net-name>' specified with strategy '<iso-strategy-name>' on crossing from source <port-name1> (domain '<dom-name1>') to destination <port-name2> (domain '<dom-name2>') that needs <input | output> isolation

Potential Issues

The isolation cell is powered by the power net and ground net specified in the associated isolation strategy. This violation message appears in either of the following situations:

- **For output isolation**: The isolation cell is relatively-off as per the sink or destination.

- **For input isolation**: The isolation cell is relatively-off as per the source.

Consequences of Not Fixing

Using an incorrect power net or ground net can lead to excess crowbar currents and incorrect design behavior.

How to Debug and Suggested Fix

The *LPISO04C* rule also generates a file named LPISO04C.csv. This file contains all the messages generated by the *LPISO04C* rule. To view the spreadsheet, in the Message Tree window, click the Open Spreadsheet option in the right-click menu of the rule.

Double-click the violation message to view the UPF file. The corresponding strategy with the incorrect `isolation_ground_net` or `isolation_power_net` is highlighted.

For output isolation, change `isolation_ground_net` or `isolation_power_net` such that they are relatively-on with respect to the sink or destination domain.

For input isolation, change `isolation_ground_net` or `isolation_power_net` such that they are relatively-on with respect to the source domain.

Double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_domain TOP  -include_scope
create_power_domain HIGH -elements /high_domain
create_supply_port VDDT
add_port_state VDDT -state {active 1.4} -state {off_state
off}
create_supply_net  VDDT -domain TOP
connect_supply_net VDDT -ports VDDT
create_supply_port VDDH
add_port_state VDDH -state {active 1.2} -state {off_state
```

```
off}
create_supply_net  VDDH -domain HIGH
connect_supply_net VDDH -ports VDDH
set_domain_supply_net HIGH -primary_power_net VDDH -
primary_ground_net VSS
set_domain_supply_net TOP -primary_power_net VDDT -
primary_ground_net VSS
set_isolation ISO2 -domain TOP  -elements { out2 out1 } -
isolation_power_net VDDH
```

For the above example, the following violation message is reported because an output isolation is required from the source (`TOP.low_domain.low_buf2.Y`) to the sink (`TOP.out1`) and the strategy is specified with the power net that is relatively-off with respect to the sink:

```
Incorrect isolation power net 'VDDH' specified with strategy
'ISO2' on crossing from source TOP.low_domain.low_buf2.Y
(domain 'LOW(supply VDDL:0.800)') to destination TOP.out1
(domain 'TOP(supply VDDT:1.400)') that needs output isolation,
test.v, 4
```

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. The incremental schematic is displayed as shown in figure below:

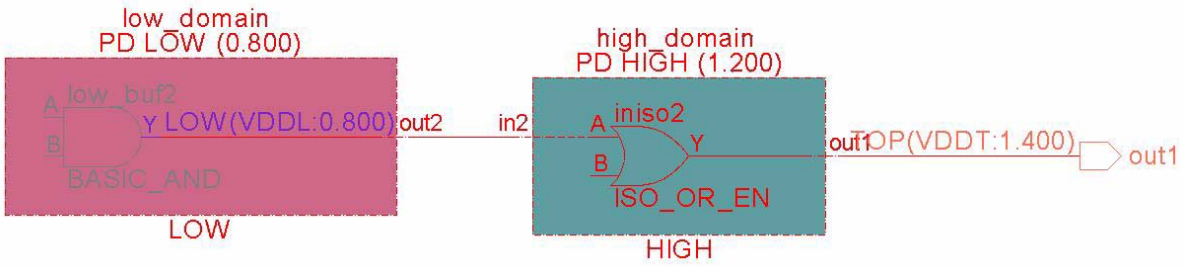


FIGURE 47. Incremental schematic

Default Severity Label

Warning

Rule Group

Isolation Logic

Reports and Related Files

None

LPISO04D

Reports standard isolation cell that cannot be implemented

When to Use

Use this rule to check the validity of isolation power net specified for an isolation strategy. This is to ensure that the isolation logic corresponding to this strategy functions correctly.

Refer to:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPISO04D* rule reports a violation message if an isolation power net is not declared as the primary power net of the domain in which the isolation cell will be inferred.

Rule Exceptions

The *LPISO04D* rule does not perform the check if:

1. `-no_isolation` is specified in the *set_isolation* command.
2. `-location` is set as `automatic` in the *set_isolation* command.
3. *map_isolation_cell* is not specified.
4. *map_isolation_cell* is specified, but it does not contain any standard isolation cells in the `-lib_cells` field.

Parameter(s)

- *lp_flag_undriven_nets*: Set the value to 1 to have the rule consider outputs of power domains that are not driven.
- *lp_flag_unconnected_nets*: Set the value to 1 to have the rule consider unconnected outputs of power domains.
- *lp_skip_buf*: Set the value to 0 to have the rule consider buffers generated during design synthesis.

- *lp_skip_pwr_gnd*: Set the value to 0 to have the rule consider nets connected to power/ground supply.
- *lp_flag_pd_outputs*: Set the value to 0 to ignore the outputs of power domain that are directly connected to primary ports.
- *lp_skip_aon_buf*: Default value is 1. Set the parameter to 0 to not skip always-on buffers to find an isolation crossing.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the power domains.
- *ignore_crossing* (Optional): Use to specify the power domain-to-voltage domain crossings and power domain-to-power domain crossings to be ignored.
- *power_state* (Optional): Use to specify the legal combinations of domain states (voltage values), that is those combinations of domain states that can exist at the same time during operation of the design.
- *assume_path* (Optional): Use to specify the paths that exist between the input and output pins of black boxes.
- *always_on_buffer* (Optional): Specify always-on buffers.
- *set_case_analysis* (Optional): Specify the case analysis conditions to be used.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_always_on_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)

- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_isolation* (Mandatory)
- *create_pst* (Mandatory)
- *add_pst_state* (Mandatory)
- *set_repeater* (Optional)

Messages and Suggested Fix

The following message appears a standard isolation cell cannot be implemented:

[LPISO04D_1][ERROR] Standard Isolation cell (determined by map_isolation_cell) cannot be implemented. Isolation power net '<iso-power-net-name>' used in set_isolation strategy 'iso-strategy-name>' is not the primary supply of domain '<domain-name(<supply-name>)' in which standard isolation cells '<standard-iso-list>' will be inferred for this strategy

Potential Issues

If *map_isolation_cell* is specified and it contains standard isolation cells, the isolation power net must be the default primary supply of the domain in which the standard isolation cell will be inferred.

Consequences of Not Fixing

The isolation cell cannot be implemented in the domain since the isolation cell supply pin is connected to the main supply rail.

How to Debug and Suggested Fix

The *set_isolation* UPF command is highlighted in the Atrenta Console GUI.

For a graphical view of the violation, double-click the message and click the Incremental Schematic button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, change `isolation_power_net` to the primary power supply of the domain in which isolation cell will be inferred.

Example Code and/or Schematic

Consider the following UPF file snippet where the –

`isolation_power_net` of domain A (VB) specified in *set_isolation* is not the same as `-primary_power_net` of domain A (VA) specified in the *set_domain_supply_net* command.

```
set_domain_supply_net A -primary_power_net VA -
primary_ground_net VSS

set_isolation ISO1 -domain A -isolation_power_net VB -
location self

set_isolation_control ISO1 -domain A -isolation_signal iso

map_isolation_cell ISO1 -domain A -lib_cells {ISO_AND
BASIC_AND}
```

The violation message generated is as follows.

```
Standard Isolation cell (determined by map_isolation_cell)
cannot be implemented.Isolation power net 'VB' used in
set_isolation strategy 'ISO1' is not the primary supply of
domain 'A(VA)' in which standard isolation cells 'ISO_AND
BASIC_AND' will be inferred for this strategy
```

The schematic generated is as follows.

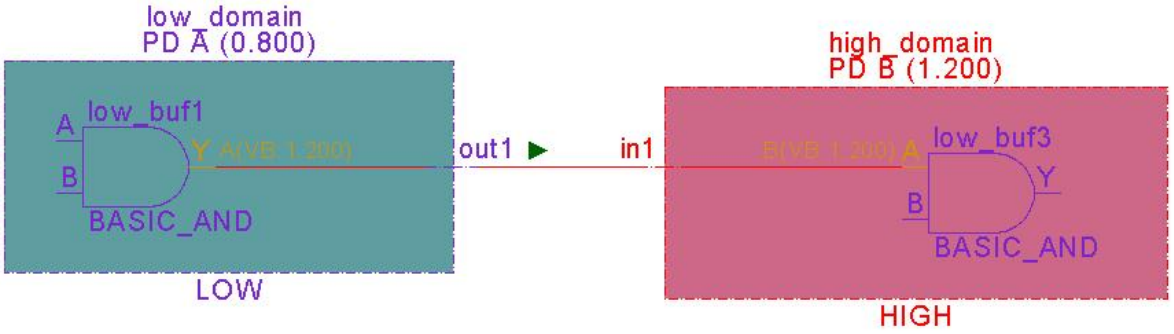


FIGURE 48. Incremental schematic

The schematic highlights the crossing, with both source and destination, involving the element where the *set_isolation* strategy is written.

To fix this violation, specify the following:

```
set_isolation ISO1 -domain A -isolation_power_net VA -
location self
```

Default Severity Label

Error

Rule Group

Isolation Logic

Reports and Related Files

None

LPISO05

Checks for redundant isolation strategy at power domain output ports

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPISO05 rule* reports a violation for a crossing where isolation is not needed but a valid isolation strategy is specified. An isolation strategy can be present at any power domain boundary port. This rule has the following subrules, which are based on two types of crossings:

- *LPISO05A*: Flags cases where isolation is not required for all paths.
- *LPISO05B*: Flags cases where isolation is needed for at least one path but is not required for other paths.

NOTE: *If the set_port_attribute -repeater_supply or set_repeater –repeater_supply_set is specified on a port in the UPF, then this rule assumes the presence of an implicit repeater at RTL and performs the checking accordingly.*

LPISO05A

Checks for cases where isolation is not required for all paths

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPISO05A rule* reports a violation for a crossing where isolation is not needed but a valid isolation strategy is specified. An isolation strategy can be present at any power domain boundary port. This rule flags cases where isolation is redundant for all paths.

Language

Verilog, VHDL

Parameter(s)

- *lp_flag_undriven_nets*: Default value is `0`. Set the value to 1 to consider outputs of power domains that are not driven.
- *lp_flag_unconnected_nets*: Default value is `0`. Set the value to 1 to consider unconnected outputs of power domains.
- *lp_skip_buf*: Default value is `1`. Set the value to 0 to consider buffers generated during design synthesis.
- *lp_skip_pwr_gnd*: Default value is `1`. Set the value to 0 to consider nets connected to power/ground supply.
- *lp_flag_pd_outputs*: Default value is `1`. Set the value to 0 to ignore the outputs of power domain that are directly connected to primary ports.
- *lp_skip_iso_check_on_ground*: Default value is `no`. Set this parameter to `yes` to not report missing isolation strategy and missing isolation cells on the ground nets.

- *lp_skip_aon_buf*: Default value is 1. Set the parameter to 0 to not skip always-on buffers to find an isolation crossing.
- *lp_skip_blackbox_checking*: Default value is `0`. Set the parameter to `1` to skip checking for black boxes.
- *lp_skip_same_src_supply_buf*: Default value is `no`. Set this parameter to `yes` to enable the rule skip buffers and inverters between strategy node and isolation cell which are of same supply as source but are LESS ON w.r.t. destination.
- *lp_skip_lib_buf*: Default value is `0`. Set the parameter to `1` to skip the instances of library buffers while checking for crossings.

Constraint(s)

CPF Commands

- *create_power_domain* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_always_on_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *create_pst* (Optional)
- *add_pst_state* (Optional)
- *map_isolation_cell* (Optional)
- *set_repeater* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when an isolation strategy is defined for of the signal crossing from *<port-name1>* of *<dom-name1>* to *<port-name2>* of *<dom-name2>,* which does not require isolation:

[LPISO05A_1][WARNING] Crossing from source <port-name1> (domain '<dom-name1>') to destination <port-name2> (domain '<dom-name2>') does not need isolation, has isolation strategy '<iso-strategy-name>' specified for '<dom-boundary>'

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the *is_isolated* attribute is set at the destination or source side and the strategy specified is redundant:

[LPISO05A_2][WARNING] Crossing from source <port-name1> (domain '<dom-name1>') to destination <port-name2> (domain '<dom-name2>') does not need isolation, has 'is_isolated' attribute set at the <destination/source> side and has isolation strategy '<iso-strategy-name>' specified for '<dom-boundary>'

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when an isolation strategy *<strategy-name>* is specified for a crossing and either the source, destination, or both is an inout port/pin:

[LPISO05A_3][WARNING] Crossing between source'<source-hier-name>' (domain '<domain-and-supply-name>') and destination '<destination-hier-name>' (domain '<domain-and-supply-name>') doesn't need isolation, has isolation strategy '<strategy-name>' specified for '<dom-boundary>' which is ignored as <source is |destination is |both sides are> an inout port

For debugging information, click *How to Debug and Fix*.

Potential Issues

For Messages 1 and 2, the strategy mentioned on the power domain boundary is redundant and is reported by this rule.

For Message 3, the strategy cannot be specified on an inout port/pin.

Consequences of Not Fixing

An isolation strategy present at a crossing that does not need to be isolated is redundant and can lead to an additional isolation cell later on.

How to Debug and Fix

Double-click the message and click the **Incremental Schematic** button. The schematic shows the crossing that does not require isolation. This rule supports IS Abstraction between start and end points and abstracts all logic between these points, except level shifter cells. Refer to the *Abstraction between Start and End Points topic* in the *Atrenta Console Reference Guide*. Refer to the Example Code and/or Schematic section for an example.

The *LPISO05A* rule also generates a file named LPISO05A.csv. This file contains all the messages generated by the *LPISO05A* rule. To view the spreadsheet, in the Message Tree window, click the **Open Spreadsheet** option in the right-click menu of the rule.

To fix these violations, ensure that a correct isolation strategy is specified for all power domain boundary ports in your design.

Double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_domain PD1  -elements {inst1 inst2}
create_power_domain PD2 -elements {inst3}
create_supply_port VDD1
add_port_state VDD1 -state {active 1.2}
create_supply_net  VDD1 -domain PD1
connect_supply_net VDD1 -ports VDD1
create_supply_port VDD2
add_port_state VDD2 -state {active 0.8}
create_supply_net  VDD2 -domain PD2
connect_supply_net VDD2 -ports VDD2

set_domain_supply_net PD2 -primary_power_net VDD2 -
```

```
primary_ground_net VSS
set_domain_supply_net PD1  -primary_power_net VDD1 -
primary_ground_net VSS
set_isolation ISO1 -domain PD1  -applies_to outputs -
isolation_power_net VDD -isolation_ground_net VSS -
clamp_value 0
```

For the above example, the following violation is reported because isolation is not needed for both the crossings (inst1 to inst2 and inst1 to inst3) and a strategy has been provided at the power domain boundary (inst.out1). Here, the strategy is redundant for all paths:

```
Crossing from source TOP.inst1.low_buf1.Y (domain 'PD1(supply
VDD1:1.200)') to destination TOP.inst3.low_buf1.A (domain
'PD2(supply VDD2:0.800)') does not need isolation, has
isolation strategy 'ISO1' specified for 'TOP.inst1.out1'
```

The schematic is as follows:

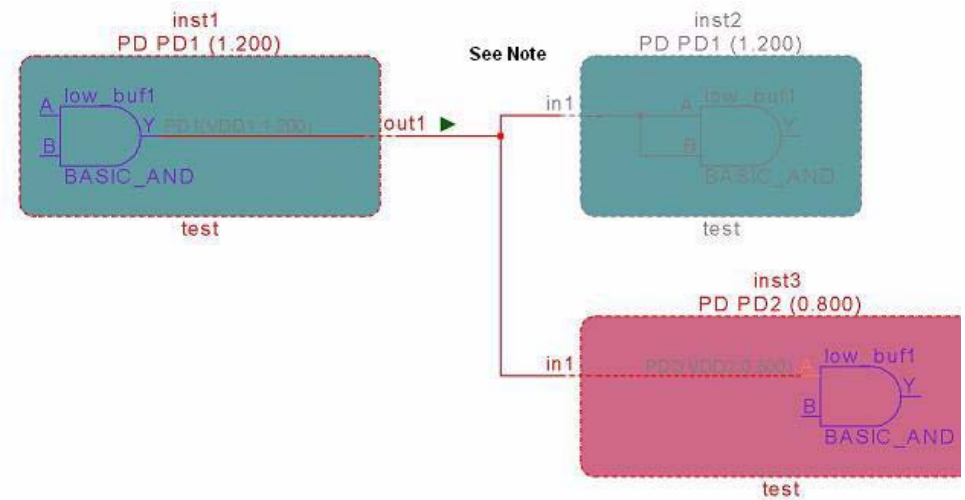


FIGURE 49. Incremental schematic

NOTE: *Only single fanout is shown in the actual schematic. Here, multiple fanouts are*

shown only to depict that the strategy is redundant for all paths.

Default Severity Label

Warning

Rule Group

Isolation Logic

Reports and Related Files

None

LPISO05B

Checks for cases where isolation is needed for at least one path but is not required for other paths

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPISO05B rule* reports a violation for a crossing where isolation is not needed but a valid isolation strategy is specified. An isolation strategy can be present at any power domain boundary port. This rule flags cases where isolation is needed for at least one path but is redundant for other paths.

Language

Verilog, VHDL

Parameter(s)

- *lp_flag_undriven_nets*: Default value is `0`. Set the value to 1 to consider outputs of power domains that are not driven.
- *lp_flag_unconnected_nets*: Default value is `0`. Set the value to 1 to consider unconnected outputs of power domains.
- *lp_skip_buf*: Default value is `1`. Set the value to 0 to consider buffers generated during design synthesis.
- *lp_skip_pwr_gnd*: Default value is `1`. Set the value to 0 to consider nets connected to power/ground supply.
- *lp_flag_pd_outputs*: Default value is `1`. Set the value to 0 to ignore the outputs of power domain that are directly connected to primary ports.

- *lp_skip_iso_check_on_ground*: Default value is `no`. Set this parameter to `yes` to not report missing isolation strategy and missing isolation cells on the ground nets.
- *lp_skip_aon_buf*: Default value is 1. Set the parameter to 0 to not skip always-on buffers to find an isolation crossing.
- *lp_skip_blackbox_checking*: Default value is `0`. Set the parameter to `1` to skip checking for black boxes.
- *lp_skip_lib_buf*: Default value is `0`. Set the parameter to `1` to skip the instances of library buffers while checking for crossings.

Constraint(s)

CPF Commands

- *create_power_domain* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_always_on_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *create_pst* (Optional)
- *add_pst_state* (Optional)
- *map_isolation_cell* (Optional)
- *set_repeater* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when an isolation strategy is defined for of the signal crossing from *<port-name1>* of *<dom-name1>* to *<port-name2>* of *<dom-name2>,* which does not require isolation:

[LPISO05B_1][WARNING] Crossing from source <port-name1> (domain '<dom-name1>') to destination <port-name2> (domain '<dom-name2>') does not need isolation, has isolation strategy '<iso-strategy-name>' specified for '<dom-boundary>'

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the *is_isolated* attribute is set at the destination side and the strategy specified is redundant:

[LPISO05B_2][WARNING] Crossing from source <port-name1> (domain '<dom-name1>') to destination <port-name2> (domain '<dom-name2>') does not need isolation, has 'is_isolated' attribute set at the <destination> side and has isolation strategy '<iso-strategy-name>' specified for '<dom-boundary>'

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when an isolation strategy *<strategy-name>* is specified for a crossing and the destination is an inout port/pin:

[LPISO05B_3][WARNING] Crossing between source'<source-hier-name>' (domain '<domain-and-supply-name>') and destination '<destination-hier-name>' (domain '<domain-and-supply-name>') doesn't need isolation, has isolation strategy '<strategy-name>' specified for '<dom-boundary>' which is ignored as <destination is> an inout port

For debugging information, click *How to Debug and Fix*.

Potential Issues

For Messages 1 and 2, the strategy mentioned on the power domain boundary is redundant and is reported by this rule.

For Message 3, the strategy cannot be specified on an inout port/pin.

Consequences of Not Fixing

An isolation strategy present at a crossing that does not need to be isolated is redundant and can lead to an additional isolation cell later on.

How to Debug and Fix

Double-click the message and click the **Incremental Schematic** button. The schematic shows the crossing that does not require isolation. This rule supports IS Abstraction between start and end points and abstracts all logic between these points, except level shifter cells. Refer to the *Abstraction between Start and End Points topic* in the *Atrenta Console Reference Guide*. Refer to the Example Code and/or Schematic section for an example.

The *LPISO05B* rule also generates a file named LPISO05B.csv. This file contains all the messages generated by the *LPISO05B* rule. To view the spreadsheet, in the Message Tree window, click the **Open Spreadsheet** option in the right-click menu of the rule.

To fix these violations, ensure that a correct isolation strategy is specified for all power domain boundary ports in your design.

Double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

Example Code and/or Schematic

Consider the following example:

A violation occurs when isolation is not needed for one fanout (inst1 to inst2) but is needed for the other fanout (inst1 to inst3). This rule reports this strategy as redundant.

```
Crossing from source TOP.inst1.low_buf1.Y (domain 'PD1(supply
VDD1:1.200)') to destination TOP.inst2.low_buf1.A (domain
'PD1(supply VDD1:1.200)') does not need isolation, has
isolation strategy 'ISO1'specified for 'TOP.inst1.out1'
```

The schematic is as follows:

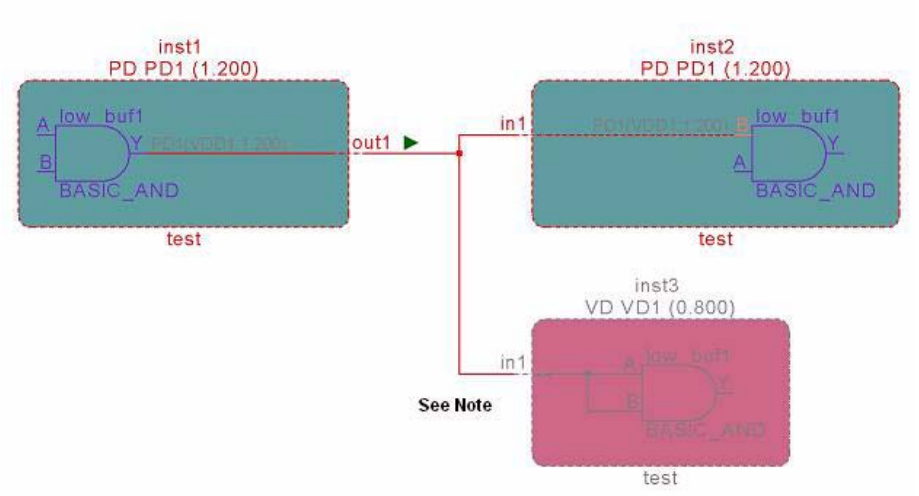


FIGURE 50. Incremental schematic

NOTE: *The crossing inst1 to inst3 is not shown in the schematic. Here, its shown only to depict that the strategy is valid for at least one crossing (inst1 to ins3).*

Default Severity Label

Warning

Rule Group

Isolation Logic

Reports and Related Files

None

LPISO06

Checks if enable pin of isolation cell is tied to isolating value or not

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPISO06* rule reports if isolation cell enable pin is tied to isolating value as specified in the UPF or not.

This rule has the following subrules:

- *LPISO06A*: Reports an error when the enable pin of an isolation cell is not tied to an isolating value.
- *LPISO06B*: Reports an informational message when the enable pin of an isolation cell is tied to an isolating value.

LPISO06A

Reports an error when the enable pin of an isolation cell is not tied to an isolating value

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Rule Description

The *LPISO06A* rule reports an error violation message if the enable pins of isolation cells do not have an isolating value. This rule checks isolation cells that have their enable pins:

- tied to a constant value, such as 0 or 1, or
- have a 0 or 1 logic value through simulation

Prerequisites

By default, the `LPISO06A` rule is not run while running the SpyGlass Power Verify solution. To enable this rule, specify the following command in the project file:

```
set_goal_option addrules LPISO06A
```

Parameter(s)

- *lp_skip_buf*: Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.

Constraint(s)

SGDC

- *set_case_analysis* (Optional)

CPF Commands

- *create_power_domain* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_always_on_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain*(Mandatory)
- *create_supply_port*(Mandatory)
- *set_pin_related_supply*(Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net*(Mandatory)
- *set_domain_supply_net* (Mandatory)
- *create_pst* (Optional)
- *add_pst_state* (Optional)
- *map_isolation_cell* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when the enable pin `<pin-name>` of the isolation or level-shifter cell `<cell-name>`, with an instance hierarchy name `<instance-hier-name>`, has a non-isolating value:

[LPISO06A_1][ERROR] `Enable pin '<pin-name>' of <isolation | level-shifter> cell '<instance-hier-name>' ('<cell-name>') is tied to non-isolating value`

Message 2

The following message appears when the pin `<pin-name>` (used in the `isolation_enable_condition` attribute) of instance `<instance-path-name>` is tied to a constant value such that the expression evaluates to `0`:

[LPISO06A_2][INFO] Pin '<pin-name>' of cell '<instance-name>' ('<cell-name>') used in isolation enable condition is tied to non isolating value

NOTE: *Violation is reported for each pin used in the isolation_enable_condition attribute.*

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

If this violation is not fixed, it means that the enable pin of isolation cell is not tied to an isolating value. Therefore, the data pin of the isolation cell will not be blocked and proper isolation will not take place.

How to Debug and Fix

The violation is reported at the instance of the isolation cell.

Double-click the message and click the **Incremental Schematic** button. The schematic shows the crossing that does not require isolation. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, ensure that the enable pin of isolation cells are appropriately tied to an isolating value.

Example Code and/or Schematic

Consider the following example.

A violation occurs when the enable pin `A` of the isolation cell `insttop_iso_and` has a non-isolating value:

[ERROR] Enable pin 'A' of isolation cell 'insttop_iso_and'('ISO_AND_EN') is tied to non-isolating value

The schematic is displayed as follows:

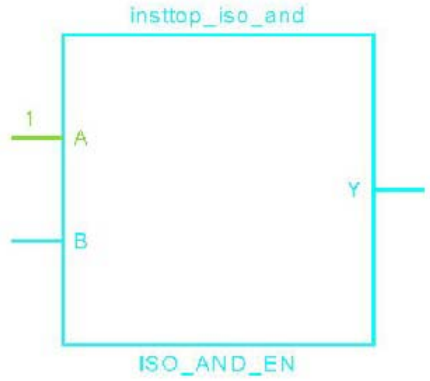


FIGURE 51. Incremental schematic

Default Severity Label

Error/Info

Rule Group

Isolation Logic

Reports and Related Files

None

LPISO06B

Reports an informational message when the enable pin of an isolation cell is tied to an isolating value

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPISO06B* rule reports an informational violation message if the enable pins of isolation cells have an isolating value. This rule checks isolation cells that have their enable pins:

- tied to a constant value, such as 0 or 1, or
- have a 0 or 1 logic value through simulation

Prerequisites

By default, the *LPISO06B* rule is not run while running the SpyGlass Power Verify solution. To enable this rule, specify the following command in the project file:

```
set_goal_option addrules LPISO06B
```

Parameter(s)

- *lp_max_viol_count*: Default is 1000. This indicates that the LPISO06B rule reports 1000 messages. Set the value to any positive integer number to report that number of messages.

Constraint(s)

SGDC

- *set_case_analysis* (Optional)

CPF Commands

- *create_power_domain* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_always_on_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain*(Mandatory)
- *create_supply_port*(Mandatory)
- *set_pin_related_supply*(Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net*(Mandatory)
- *set_domain_supply_net* (Mandatory)
- *create_pst* (Optional)
- *add_pst_state* (Optional)
- *map_isolation_cell* (Optional)

Messages and Suggested Fix

Message 1

An informational message appears when the violation count of this rule exceeds the limit set by the *lp_max_viol_count* parameter. Refer to *Message 5* for the message and, for debugging information, refer to *How to Debug and Fix*.

Message 2

The following message appears when the enable pin `<pin-name>` of the isolation or level-shifter cell `<cell-name>`, with an instance hierarchy name `<instance-hier-name>`, has an isolating value:

[LPISO06B_1][INFO] Enable pin '<pin-name>' of <isolation | level-shifter> cell '<instance-hier-name>' ('<cell-name>') is tied to isolating value

Message 3

The following message appears when the pin (used in the `isolation_enable_condition` attribute) of instance `<instance-path-name>` is tied to a constant value such that the expression evaluates to `1`:

```
[LPISO06B_3][ERROR] Pin '<pin-name>' of cell '<instance-name>' ('<cell-name>') used in isolation enable condition is tied to an isolating value
```

NOTE: *Violation is reported for each pin used in the isolation_enable_condition attribute.*

Potential Issues

This is an informational message that you can use for review purposes.

Consequences of Not Fixing

Not applicable

How to Debug and Fix

The message is reported at the instance of the isolation cell.

Double-click the message and click the **Incremental Schematic** button. The schematic shows the crossing that does not require isolation. Refer to the Example Code and/or Schematic section for an example.

Example Code and/or Schematic

Consider the following example.

The message appears when the enable pin `B` of the isolation cell `insttop_iso_or` has an isolating value:

```
[INFO] Enable pin 'B' of isolation cell 'insttop_iso_or'('ISO_OR_EN') is tied to isolating value.
```

The schematic is displayed as follows:

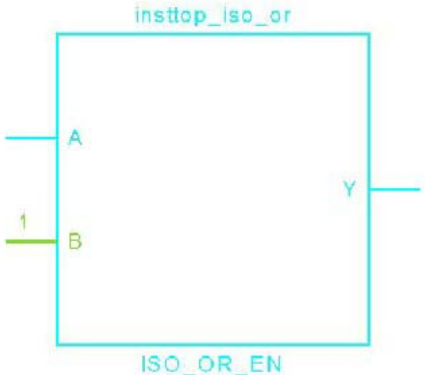


FIGURE 52. Incremental schematic

Default Severity Label

Error/Info

Rule Group

Isolation Logic

Reports and Related Files

None

LPISO07

Checks if the clamp value provided in isolation strategy is in sync with the constant value reaching at the element specified in the strategy

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPISO07* reports an error when a constant value reaching at an element does not match with the clamp value specified in the strategy for that element.

This rule:

- Considers port tied to a constant as well as constant value coming to port due to constant propagation.
- Checks only those isolation strategies where clamp value is either `0` or 1.
- Does not perform checking for an isolation strategy that has –`no_isolation` argument.

Consider the following scenario:

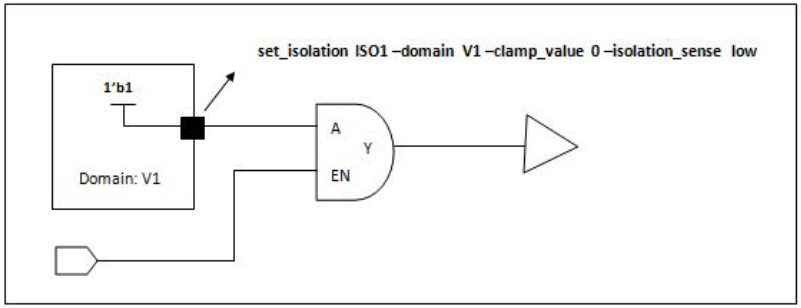


FIGURE 53. Isolation strategy applied on design port

In the above figure, isolation strategy ISO1 is applied on a design port that is connected to a constant value (1'b1). The clamp value specified in the strategy is 0. The tied inputs are often linked to generic IPs, with parameters. It is not recommended to have tie value on an ISO input different from the clamp value. Those configurations need to be analyzed carefully. This rule reports a violation if tie input on an isolation cell is different from the clamp value.

An input of isolation cell is considered tied high/low if:

- It is connected to supply0 (1'b0)/supply1 (1'b1)
- It is connected to TIE-HI/TIE-LO cell.

Some more design scenarios are given below.

The following two figures show an instrumented scenario where an isolation cell is placed:

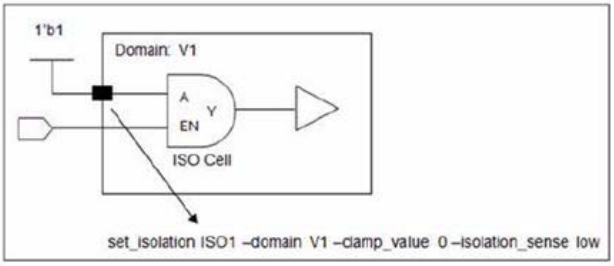


FIGURE 54. Isolation cell present

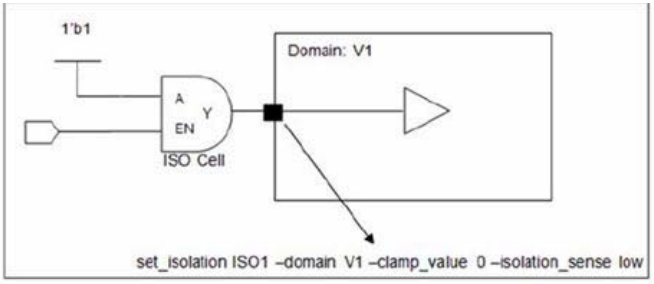


FIGURE 55. Isolation cell present

The following two figures show a non-instrumented scenario where isolation cell is not present:

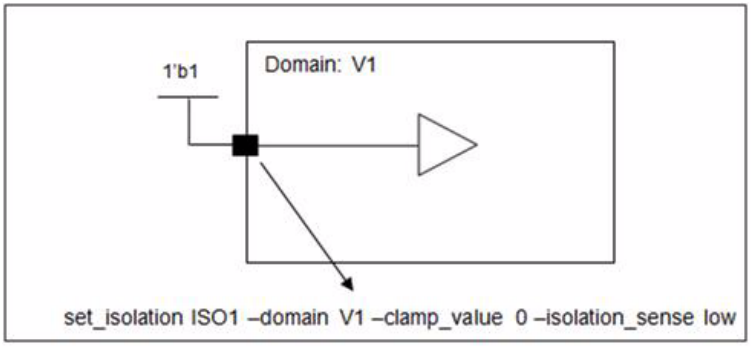


FIGURE 56. Isolation cell not present

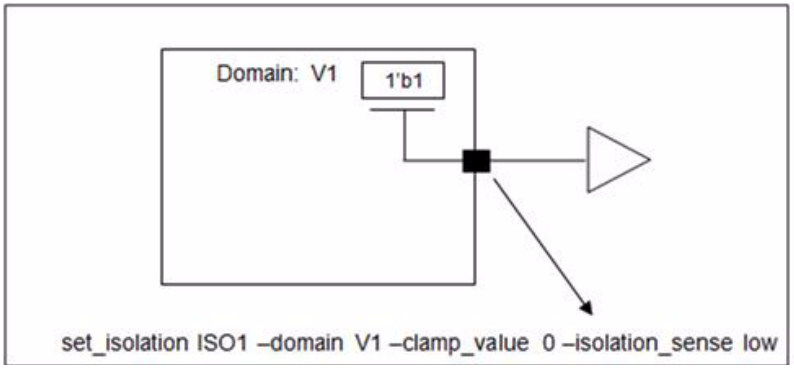


FIGURE 57. Isolation cell not present

Language

Verilog, VHDL

Parameter(s)

- *lp_skip_buf*: Default value is `1`. Set the value to `0` to consider buffers generated during design synthesis.

Constraint(s)

UPF Commands

- *create_power_domain*(Mandatory)
- *create_supply_port*(Mandatory)
- *add_port_state*(Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net*(Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_isolation* (Mandatory)
- *create_pst* (Optional)
- *add_pst_state* (Optional)
- *map_isolation_cell* (Optional)

Messages and Suggested Fix

The following message appears when a constant value reaching at an element <*object-name*> does not match with clamp value <*clamp-value*> specified in the strategy <*strategy-name*> for that element:

[LPISO07_1][WARNING] Tie value <tie-value> connected to element <object-name> is different from clamp value <clamp-value> specified in isolation strategy <strategy-name>

Potential Issues

The constant value reaching at each element should be in sync with the clamp value defined for that element in `set_isolation`. If the two values do not match, it is a design issue that must be fixed.

Consequences of Not Fixing

The output of isolation cell does not achieve proper value in case of power down condition.

How to Debug and Fix

Double-click the message and click the **Incremental Schematic** button. The schematic shows the crossing where a constant value reaching at an element does not match with clamp value specified in the strategy for that element. This rule supports IS Abstraction between the start and end

points and abstracts all logic between these points, except level shifter cells. Refer to the *Abstraction between Start and End Points* topic in the *Atrenta Console Reference Guide*.

Refer to the Example Code and/or Schematic section for an example.

Example Code and/or Schematic

Consider the following example.

Constant value reaching at element `Top.out` is `1` whereas in the strategy specified in UPF file for this element is `0`.

```
set_isolation ISO1 -domain TOP  -applies_to both -
isolation_power_net VDD -isolation_ground_net VSS -
clamp_value 0
```

As there is a mismatch between the constant value reaching `Top.out` and clamp value specified in `set_isolation_command`, the following violation message is reported for the element `Top.out`:

```
Tie value 1 connected to element TOP.out is different from
clamp value 0 specified in isolation strategy ISO1.
```

The schematic is displayed as follows:

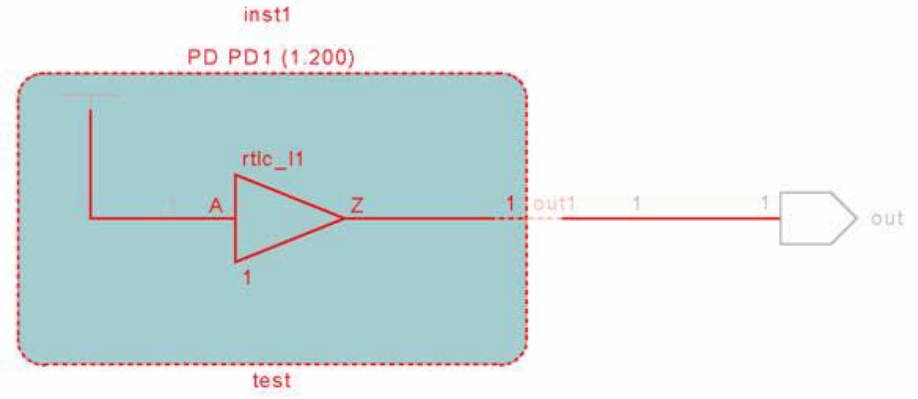


FIGURE 58. Incremental schematic

Default Severity Label

Warning

Rule Group

Isolation Logic

Reports and Related Files

None

LPSVM08

Checks for the presence of correct isolation cells in all power domain output crossings

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM08* rule checks for the presence of correct isolation cells in all the power domain output crossings. All power domain crossings should go through isolation cells. Here, checks are present to ensure if the correct isolation cell is present in a path from one power domain to other.

This rule has the following subrules:

- *LPSVM08A*: Checks for correct isolation logic of various power domains in the design
- *LPSVM08B*: Reports missing isolation cell at output terminals of power domain, not having any isolation strategy
- *LPSVM08C*: Reports non-existence of isolation cell at excluded output terminals of power domain

LPSVM08A

Checks for correct isolation logic of various power domains in the design

When to Use

To check for isolation in a multiple power design. This rule is applicable to all design phases after the insertion of isolation. Refer to:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The rule reports power domains with incorrect isolation logic.

The rule reports a violation if the output of a power domain is not connected to any isolation cell. A cell with the *is_isolation_cell* attribute set as `true` in the library, is recognized as an isolation cell. A level shifter with an enable pin, clamp level shifter cell, is also treated as an isolation cell.

The *LPSVM08A* rule reports power domains where:

- The output is connected to a gate other than an isolation cell or a two-stage isolation logic that is composed of a tristate buffer and a bus holder.

  An isolation cell is a cell where one of the isolation signals of the power domain is one of the inputs. These cells can be specified using the *isolation_cell* constraint in SGDC or the *is_isolation_cell* attribute in the library file. If both of these are used to specify isolation cells, all the cells are considered.

- The output of the power domain is connected to an isolation cell and one or more of the following conditions are true:

  - The isolation cell is not specified using the *isolation_cell* constraint. This case is checked only when you have specified one or more `isolation_cell` constraints.
  - The isolation cell is not a NAND/NOR/AND/OR gate provided the *lp_check_valid_iso* parameter is set.

- When an input pin of a library cell is tied to `1'b1` or `1'b0` and another supply is forced via `connect_supply_net/ set_related_supply_net` command on the same input pin such that isolation is required between the tied supply and forced supply.
- For UPF 2.0, the rule also reports if the isolation cell name does not match the `name_prefix/name_suffix` options specified in the `set_isolation` command

The *LPSVM08A* rule traverses to the fan-out of the isolation cell or two-stage isolation logic and checks whether the voltage value of the connected voltage domain is the same as the `ON` value of the power domain. A message is reported if the correct level shifter is not found between the power domain and the connected voltage domain.

When a power domain is shut off, the outputs of the block should be isolated through isolation logic.

Prerequisites

The information about power domains and their operating conditions are specified and the information about the isolation cell is available using either CPF, UPF or SGDC.

Rule Exceptions

The *LPSVM08A* rule does not check outputs of the power domain that do not have an isolation control signal, which is specified using the `-isosig` argument.

When the power format is CPF/SGDC, this rule does not check for those outputs that are coming from an always-on buffer because they would not need isolation.

When the power format is UPF, this rule does not check for the crossings that do not have any isolation rule specified for them. It also does not check for those outputs that have the default strategy and the same supply on both source and sink instances.

Language

Verilog, VHDL

Parameter(s)

- *lp_complex_iso_logic*: Default value is 0. Set the parameter 1 to have the rule assume that isolation cells are complex isolation cells.
- *lp_flag_undriven_nets*: Default value is 0. Set the parameter to 1 to have the *LPSVM08A* rule consider nets that are not driven at voltage crossings.
- *lp_flag_unconnected_nets*: Default value is 0. Set the parameter to 1 to have the *LPSVM08A* rule consider unconnected nets at voltage crossings.
- *lp_skip_pwr_gnd*: Default value is 1. Set the parameter to 0 to process nets connected to power/ground supply while checking for the correctness of the level shifters.
- *lp_skip_aon_buf*: Default value is 1. Set the parameter to 0 to not skip always-on buffers to find an isolation crossing.
- *lp_skip_buf*: Default value is 1 and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to 0 to consider SpyGlass-generated buffers during rule checking.
- *lp_skip_lib_buf*: Default value is 0. Set the parameter to 1 to skip the instances of library buffers while checking for crossings.
- *lp_flag_pd_outputs*: Default value is 1. Set the parameter to 0 to ignore the outputs of power domain that are directly connected to primary ports.
- *lp_isologic_in_pd*: Default value is 1. Set the parameter to 0 to have the consider the isolation cells outside the power domain only.
- *lp_check_valid_iso*: Default value is 0. Set the parameter to 1 to check whether the isolation cell is a NAND/NOR/AND/OR gate by functionality.
- *lp_disable_lib_attr_read*: Default value is 0. Set the parameter to 1 have the rule ignore library cells with *is_isolation_cell* attribute and consider only cells specified with the *isolation_cell* constraint as output-side isolation logic.
- *lp_match_location_by_domain*: Default value is yes and the isolation cell and level shifter cell locations are matched based on domain. To match these locations based on hierarchy, set the value of this parameter to no.

- *lp_report_domain_crossing_on_iso_signal*: Default value is `0`. Set the value of this parameter to `1` to enable the *LPSVM08A* rule to check isolation signals in the SGDC flow.
- *lp_allow_check_name_format*: Default value is `no` and this rule does not report a violation if isolation/level shifter instance name does not match with the name prefix/suffix specified in corresponding strategy. Set this parameter to `yes` to report violations for such cases.
- *lp_skip_blackbox_checking*: Default value is `0`. Set the parameter to `1` to skip checking for black boxes.
- *lp_skip_same_src_supply_buf*: Default value is `no`. Set this parameter to `yes` to enable the rule skip buffers and inverters between strategy node and isolation cell which are of same supply as source but are LESS ON w.r.t. destination.

Constraint(s)

SDGC

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage/ power domains in the design.
- *ignore_crossing* (Optional): Use this constraint to specify the power domain-to-voltage domain crossings and power domain-to-power domain crossings to be ignored with the `ignore_crossing` constraint.
- *power_state* (Optional): Use this constraint to specify the combinations of domain states which can exist at the same time during a design operation.
- *antenna_cell* (Optional): Use this constraint to specify the antennae protection cells (diode cells) which need to be ignored by the rules in the SpyGlass Power Verify solution.
- *always_on_buffer* (Optional): Use this constraint to specify the names of always-on buffers using the `always_on_buffer` constraint.
- *set_case_analysis* (Optional): Use this constraint to specify the case analysis conditions used by the *LPSVM08A* rule using the `set_case_analysis` constraint.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_isolation_rule* (Mandatory)
- *update_isolation_rules* (Optional)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_isolation_cell* (Optional)
- *define_always_on_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_isolation* (Mandatory)
- *set_isolation_control* (Mandatory)
- *map_isolation_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when the signal <*sig-name*> traverses from the power domain <*pd-name*> to domain <dom-name> does not have the correct isolation logic:

[LPSVM08A_1][WARNING] Signal '<sig-name>' going from source '<source-name>' (power domain '<pd-name>') to destination '<destination-name>' (domain '<dom-name>') must be connected to isolation cell corresponding to strategy '<isolation-strategy-

name>'

NOTE: *In CPF and SGDC flow, the domain information (*<pd-name>,<dom-name>*) shows the domain name and its voltage value. However, in UPF flow, the domain information shows the domain name along with the associated supply name and supply value.*

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the signal *<sig-name>* traverses from the power domain *<pd-name>* to domain <dom-name> is connected to an incorrect isolation cell *<iso-cell-name>*:

[LPSVM08A_2][WARNING] Signal '<sig-name>' going from source '<source-name>' (power domain '<pd-name>') to destination '<destination-name>' (domain '<dom-name>') is connected to incorrect isolation cell ‘<instance-name >(<iso-cell-name>)’ corresponding to strategy '<isolation-strategy-name>', Isolation cell is not present in map_isolation_cell command

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when the isolation cell <cellname>(<inst-name>) is present at an invalid location:

[LPSVM08A_6][WARNING] Isolation cell '<inst-name>(<cell-name>)' placed between source ‘<source-name>' (power domain ‘<domain-name>') and destination '<destination-name>' (domain '<domain-name>') is not present in valid location corresponding to strategy <isolation-strategy-name>

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appear in the CPF flow, when a location <loc> is specified with the -within_hierarchy option of the *update_isolation_rules* command for the isolation cell <cell-name>(<inst-name>), which is connected to the output <output-name> of power domain <pd-name>, present at an invalid location:

[LPSVM08A_8][WARNING] Isolation cell '<cell-name>(<inst-name>)' connected to the output '<output-name>' of power domain

'<pd-name>' is not present in valid location '<loc>'

For debugging information, click *How to Debug and Fix*.

Message 5

The following message appears when the signal *<sig-name>* traverses from the power domain *<pd-name>* to domain <dom-name> is connected to an isolation cell that is not a NAND/NOR/AND/OR cell by functionality:

[LPSVM08A_5][WARNING] Signal '<sig-name>' going from source '<source-name>' (power domain '<pd-name>') to destination '<destination-name>' (domain '<dom-name>') should be connected to NAND/NOR/AND/OR isolation logic

Message 6

The following message appears when tied signal *<sig-name>* going from source *<source>* (power domain *<tied-pd-name>*) to destination *<destination>* (domain *<forced-dom-name>*) does not have the correct isolation logic:

[LPSVM08A_4][WARNING] Signal '<sig-name>' going from source '<source>' (power domain '<tied-pd-name>') to destination '<destination>' (domain '<forced-dom-name>') must be connected to isolation cell

Message 7

The following message appears when for a signal *<sig-name>* going from source *<src>* to destination *<dest>*, the isolation instance *<iso-inst-name>* does not match the prefix/suffix convention specified in the corresponding strategy. In UPF 2.0, prefix/suffix convention can be applied using the –prefix/-suffix argument of the set_isolation command. In UPF 1.0, the prefix/suffix convention can be applied using the name_format command. Also, in UPF 1.0, the checking of the name_format command is done using the *lp_allow_check_name_format* parameter:

[LPSVM08A_9][WARNING] For signal '<sig-name>' going from source '<src>' (power domain '<pd-name>') to destination '<dest>' (domain '<domain-name>'), the isolation instance name '<iso-inst-name>(<iso-cell-name>)' does not match the name <prefix|suffix> specified in corresponding strategy '<strategy-

name>'

Potential Issues

There is a missing isolation cell on a signal traversing from a switched off power domain or an isolation cell is placed in the wrong location.

In some designs, standard AND, OR, and LATCH logic gates may be used as isolation logic. However, in other design styles, a specific library of isolation cells is supplied. In this case, cells from this library should only be used for isolation, and may not be used elsewhere.

Consequences of Not Fixing

The output value would float and can lead to design failure.

Message 7 only shows that the naming prefix/suffix convention has not been followed and does not cause the design to fail.

How to Debug and Fix

The message states the connection from source voltage domain to destination voltage domain has a missing or incorrect isolation cell. It also points out when the location of the isolation cell is different from the location specified by the user. In addition, it shows details of the source voltage domain, destination voltage domain, and their voltage value.

The violation is reported at the first place where the signal is used.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Turn on the Power View. The schematic highlights power domains with incorrect isolation logic.

The *LPSVM08A* rule also generates a file named LPSVM08A.csv. This file contains all the messages generated by the *LPSVM08A* rule. To view the spreadsheet, in the Message Tree window, click the Open Spreadsheet option in the right-click menu of the rule.

To resolve this violation, perform the following:

- Insert missing isolation cell at the output of power domain.
- Re-check isolation cell specification.

Refer to the *Example Code and/or Schematic* section to view a sample schematic and to understand this rule.

Refer to the *Reports and Related Files* section for information on the reports generated by this rule.

For **Message 7,** make sure isolation instance name corresponds to the

prefix/suffix convention specified in the corresponding strategy of the specified port, in the UPF.

For **Messages 1, 2, 6,** and **7**, double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

Example Code and/or Schematic

Example 1

Consider the following example for checking isolation logic at the RTL design stage. The *LPSVM08A* rule checks whether all the outputs of the power domain are isolated.

In the following example, there are two domains, VD0 and VD1. VD0 is a power domain and VD1 is a voltage domain. So, any signal going from VD0 to VD1 needs isolation.

UPF

```
create_power_domain VD0 -include_scope
create_power_domain VD1 -elements {inst1 }


add_port_state VDD1 -state {on 0.9}
add_port_state VDD0 -state {on 1.0} -state {OFF off}
add_port_state GND -state {on 0.0}


set_isolation iso1 -domain VD1 -applies_to inputs
clamp_value 0 -location parent -isolation_supply_set sp1
```

```
[WARNING] Signal 'top.in1' going from source 'top.in1' (power
domain 'VD0(supply VDD0:1.000)') to destination
'top.inst1.inst2.and1.B' (domain 'VD1(supply VDD1:0.900)') must
be connected to isolation cell corresponding to strategy 'iso1'
```

The incremental schematic is displayed as shown in figure below.

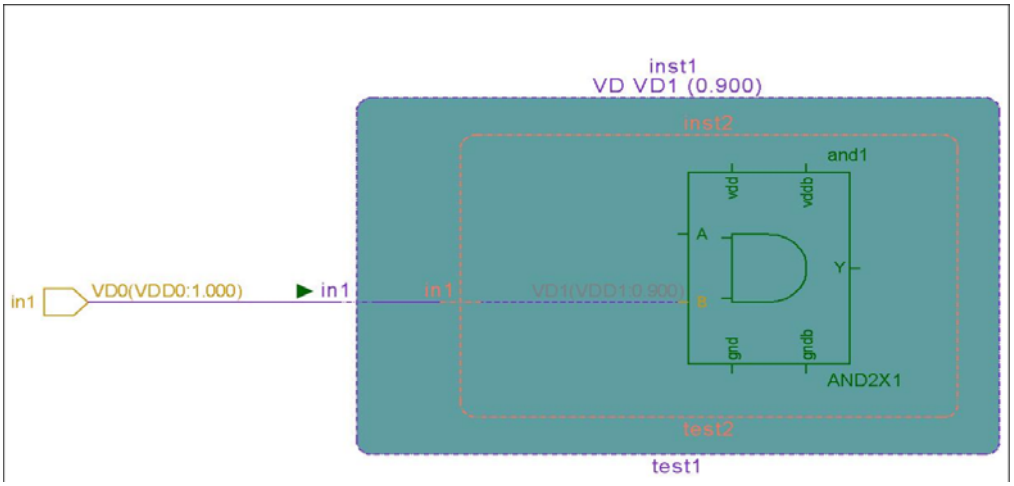


FIGURE 59. Incremental schematic

Example 2

Consider the following example.

```
[WARNING] Signal 'top.inst1.out1' going from source
'top.inst1.iso3.Y' (power domain 'PD1(1.200 0)') to destination
'top.buf3.A' (domain 'VD1(1.000)') is connected to incorrect
isolation cell top.inst1.iso3('ISO_AND_EN'), Isolation cell is
not present in map_isolation_cell command
```

The incremental schematic is displayed as shown in figure below.

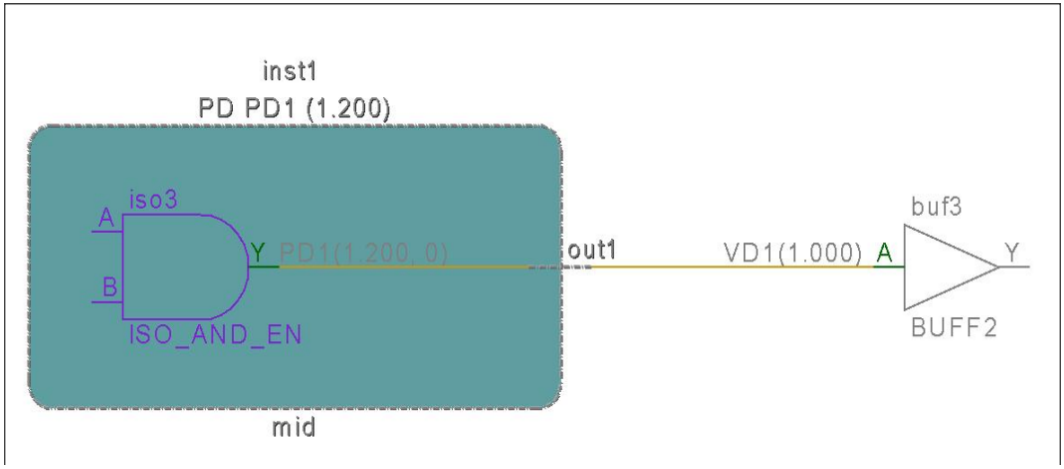


FIGURE 60. Incremental schematic

The schematic for the message shows the following:

- the connection between the two domains
- the source voltage domain and its value
- the destination voltage domain and its value
- (Optional) the (incorrect) isolation cell is highlighted in a different color.

Example 3

In the following example `set_related_supply_net` command is written on leaf level pin `inst1/and1/A` that overrides the related supply of the pin to `VDD2` and the same pin is tied to `1'b1` that connect pin A to supply VDD1 (primary power of the domain PD1). So an isolation cell is required between supply `VDD1` and `VDD2` as per the below UPF example.

```
create_power_domain TOP -include_scope
create_power_domain PD1 -elements {inst1}
create_power_domain PD2 -elements {inst2}

set_domain_supply_net TOP -primary_power_net VDD
primary_ground_net VSS
```

```
set_domain_supply_net PD1 -primary_power_net VDD1
primary_ground_net VSS
set_domain_supply_net PD2 -primary_power_net VDD2
primary_ground_net VSS

set_related_supply_net -object_list {inst1/and1/A} -power
VDD2 -ground VSS

add_port_state VDD -state {on1 1.0}
add_port_state VSS -state {def 0.00}
add_port_state VDD1 -state {on1 0.9} -state {off1 off}
add_port_state VDD2 -state {on1 0.8}

create_pst ps1 -supplies {VDD VDD1 VDD2 VSS}
add_pst_state s1 -pst ps1 -state {on1 on1 on1 def}
add_pst_state s2 -pst ps1 -state {on1 off1 on1 def}
```

For the above example, the rule reports the following violation message:

[WARNING] Signal 'VDD1' going from source 'VDD1' (power domain 'PD1(supply VDD1:0.900)') to destination 'top.inst1.and1.A' (domain 'PD1(supply VDD2:0.800)') must be connected to isolation cell

The incremental schematic is displayed as shown in figure below.

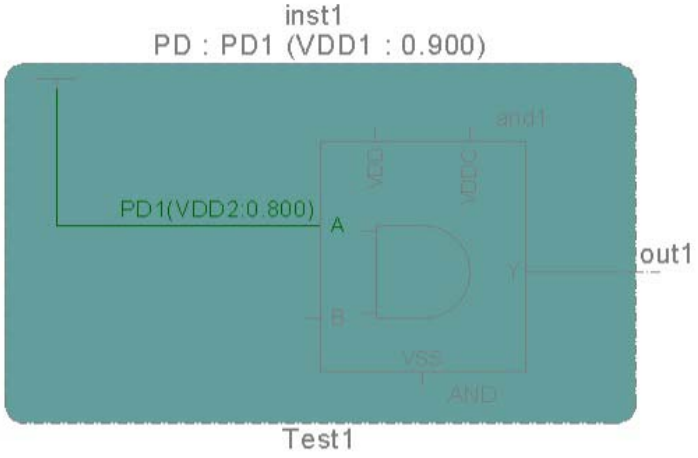


FIGURE 61. Incremental schematic

Example 4

Consider the following UPF snippet:

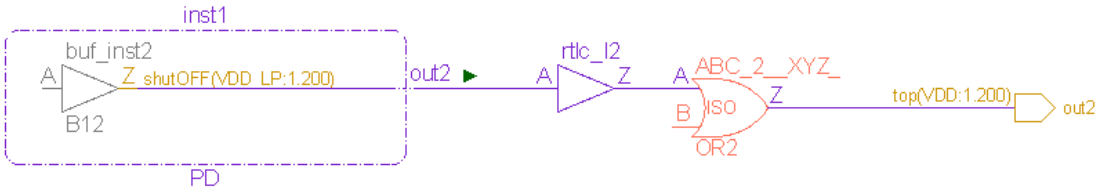
```
upf_version 2.0
# power domain definitions
create_power_domain top
create_power_domain shutOFF -elements { inst1 }

# supply nets
create_supply_port VDD
add_port_state VDD -state {default 1.2}
create_supply_net  VDD -domain top

create_supply_port VDD_LP
add_port_state VDD_LP -state {default 1.2} -state {off_state
off}
create_supply_net  VDD_LP -domain shutOFF

connect_supply_net VDD -ports VDD
connect_supply_net VDD_LP -ports VDD_LP
```

```
create_supply_port VSS
create_supply_net  VSS -domain top
connect_supply_net VSS -ports VSS
add_port_state VSS -state {default 0.0}

set_domain_supply_net top -primary_power_net VDD
primary_ground_net VSS
set_domain_supply_net shutOFF -primary_power_net VDD_LP
primary_ground_net VSS

# isolation rules
set_isolation iso_r1 -domain shutOFF -clamp_value 0
applies_to both -name_prefix ABC -name_suffix XYZ
set_isolation_control iso_r1 -domain shutOFF
isolation_signal iso -isolation_sense low -location parent
map_isolation_cell iso_r1 -domain shutOFF -lib_cells AN2

set_isolation iso_r2 -domain shutOFF -clamp_value 0 -elements
{ inst1/out1 }
set_isolation_control iso_r2 -domain shutOFF
isolation_signal iso -isolation_sense low -location parent
map_isolation_cell iso_r2 -domain shutOFF -lib_cells OR2
```

For the above example, the rule reports the following violation message because for the signal `top.inst1.out2` going from source `top.inst1.buf_inst2.Z` to destination `top.out2`, the isolation instance `top.ABC_2__XYZ_` does not match the suffix `suffix('XYZ')` specified in the corresponding strategy `iso_r1`:

```
[WARNING] For signal 'top.inst1.out2' going from source
'top.inst1.buf_inst2.Z' (power domain 'shutOFF(supply
VDD_LP:1.200)') to destination 'top.out2' (domain 'top(supply
VDD:1.200)'), the isolation instance name
'top.ABC_2__XYZ_'(OR2) does not match the name suffix('XYZ')
specified in corresponding strategy 'iso_r1'
```

The incremental schematic is displayed as shown in figure below.



Default Severity Label

Warning

Rule Group

Power_Domain_Rules

Reports and Related Files

- *lp_crossing_data*: Shows the valid isolation cell for the combination of source and destination voltage domains. The reports shows the following information:
  - list of valid isolation cell specified for the domain crossings in the `Allowed isolation cells` field.
  - valid location of the isolation cells in the `Isolation Location` field.

LPSVM08B

Reports missing isolation cell at output terminals of power domain, not having any isolation strategy

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM08B* rule reports for signal crossings, which have neither an isolation cell nor an isolation strategy.

Prerequisites

The *LPSVM08B* rule does not run while running the SpyGlass Power Verify policy. To enable this rule, select the *LPSVM08B* rule in the Atrenta Console GUI.

Language

Verilog, VHDL

Parameter(s)

- *lp_skip_feedthrough_buffer*: Default value is `0`. Set the value to `1` to skip feedthrough buffers.
- *lp_flag_undriven_nets*: By default, the `LPSVM08B` rule ignores outputs, which are not driven, of power domains. Set the *lp_flag_undriven_nets* rule parameter to have the `LPSVM08B` rule consider such outputs.
- *lp_flag_unconnected_nets*: By default, the `LPSVM08B` rule ignores unconnected outputs of power domains. Set the *lp_flag_unconnected_nets* rule parameter to have the `LPSVM08B` rule consider such outputs.
- *lp_flag_pd_outputs*: By default, the `LPSVM08B` rule flags the outputs of power domain that are directly connected to primary ports. Set the *lp_flag_pd_outputs* rule parameter to `0` to ignore such outputs.

- *lp_skip_buf*: By default, the *lp_skip_buf* parameter is set to 1 and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to 0 to consider SpyGlass-generated buffers during rule checking.
- *lp_skip_pwr_gnd*: By default, the LPSVM08B rule ignores nets connected to power/ground supply. Set the *lp_skip_pwr_gnd* rule parameter to 0 to process such nets also.
- *lp_skip_iso_check_on_ground*: Default value is no. Set this parameter to yes to not report missing isolation strategy and missing isolation cells on the ground nets.
- *lp_check_pwr_gnd_to_macro_without_prd*: Default value is yes. Set this parameter to no to ignore crossings between tie-high/low and macro cell (is_macro_cell: true)/pad cell (pad_cell: true), without domain boundary. Level shifter and isolation checking for these crossings at the RTL and Netlist levels will be ignored.
- *lp_skip_blackbox_checking*: Default value is 0. Set the parameter to 1 to skip checking for black boxes.

Constraint(s)

SDGC

- (Mandatory) *voltage_domain*: The LPSVM08B rule requires you to specify the power domains with the voltage_domain constraint and the name and value of the isolation signal using the -isosig argument and -isoval argument respectively in a SpyGlass Design Constraints file. When a power domain is shut-off, the outputs of the block should be isolated through isolation logic.
- (Optional) *ignore_crossing*: To specify the power domain-to-voltage domain crossings and power domain-to-power domain crossings to be ignored with the ignore_crossing constraint.
- (Optional) *power_state*: You can specify the power domain relationships and automatically find all the ignore_crossing constraints with the power_state constraint.
- (Optional) *assume_path*

- (Optional) *always_on_buffer*: You can specify the names of always-on buffers using the `always_on_buffer` constraint.
- (Optional) *set_case_analysis*: You can specify the case analysis conditions used by the `LPSVM08B` rule using `set_case_analysis` constraint.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_always_on_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *create_pst* (Optional)
- *add_pst_state* (Optional)
- *map_isolation_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

The following message appears when an isolation cell is not found at the output terminal <*term-name*> of the power domain <*domain-info*>:

[LPSVM08B_1][WARNING] Output signal '<signal-name>' going from source '<source-name>' (power domain '<pd-name>') to destination '<dest-name>' (domain 'dom-name') does not have any isolation cell placed in its path but need isolation

If the destination pin has been used in the `isolation_enable_condition` attribute as shown below, the violation message is changed to reflect that:

```
pin(d) {
  is_isolated : true;
  isolation_enable_condition : “en1 * en2”;
}
```

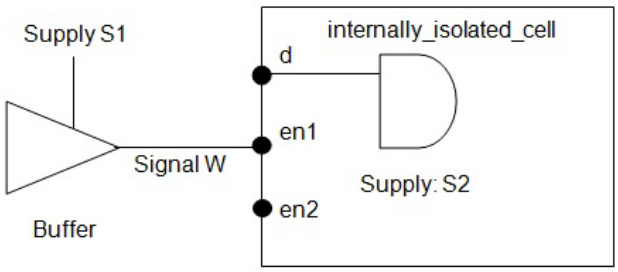


FIGURE 62. Output isolation required

In the above example, the pin en1 (supply S2) is driven by buffer that is powered by supply S1. S1 is less on than S2, therefore output isolation is required. This scenario is reported in the following violation message:

[LPSVM08B_2][WARNING] Output signal '<signal-name>' going from source '<source-name>' (power domain '<pd-name>') to destination '<dest-name>' (domain '<domain-name>') needs isolation. Destination pin is used in 'isolation_enable_condition' attribute

In CPF and SGDC flow, the domain information (<*domain-info*>) shows the domain name and its voltage value. However, in UPF flow, the domain information shows the domain name along with the associated supply name and supply value.

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not Fixing

If no isolation strategy or isolation cell is provided, the crossing will result in design failure.

How to Debug and Fix

The violation is reported at the first place where the output signal of power domain is used.

Double-click the message and click the **Incremental Schematic** button. The schematic shows the output terminal and the power domain. Refer to the Example Code and/or Schematic section for an example.

To fix the violation, ensure that a valid isolation strategy is present at the power domain boundary or an isolation cell is present at the output terminals of the power domain, which does not have any isolation strategy.

Double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

Example Code and/or Schematic

In the following example, a violation occurs when no isolation cell is present at the output terminal `top.pd_inst.out2` of the power domain *shutOFF* and no isolation strategy is specified for `top.pd_inst.out2`:

```
Output signal 'top.pd_inst.out2' going from source
'top.pd_inst.buf2.Y' (power domain 'shutOFF(supply
VDD_LP:1.20)') to destination 'top.out2' (domain 'top(supply
VDD:1.20)') does not have any isolation cell placed in its path
but need isolation
```

The schematic is as follows:

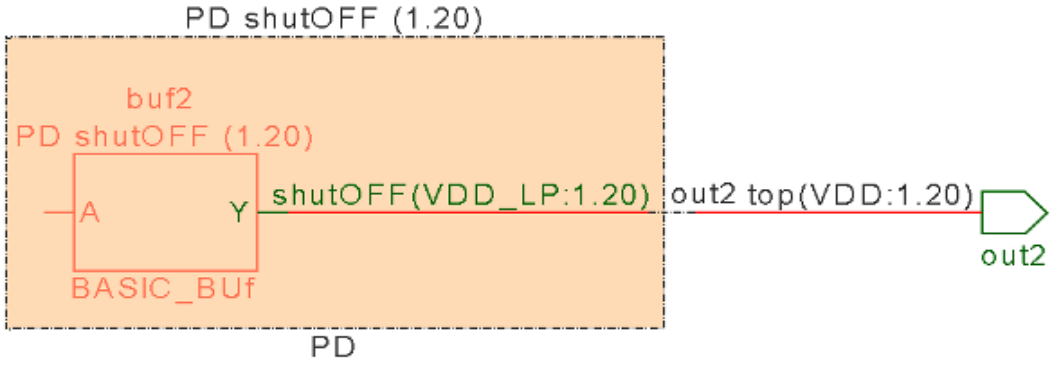


FIGURE 63. Incremental schematic

Default Severity Label

Warning

Rule Group

Isolation Logic

Reports and Related Files

None

LPSVM08C

Reports non-existence of isolation cell at excluded output terminals of power domain

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM08C* rule reports when a signal crossing from a power domain output to another domain is defined as excluded by specifying `no_isolation` at the power domain boundary, but needs isolation, and it does not have an isolation cell present.

Prerequisites

The *LPSVM08C* rule does not run while running the SpyGlass Power Verify policy. To enable this rule, select the *LPSVM08C* rule in the Atrenta Console GUI.

Language

Verilog, VHDL

Parameter(s)

- *lp_skip_feedthrough_buffer*: Default value is `0`. Set the value to `1` to skip feedthrough buffers.
- *lp_flag_undriven_nets*: By default, the `LPSVM08C` rule ignores outputs, which are not driven, of power domains. Set the *lp_flag_undriven_nets* rule parameter to have the `LPSVM08C` rule consider such outputs.
- *lp_flag_unconnected_nets*: By default, the `LPSVM08C` rule ignores unconnected outputs of power domains. Set the *lp_flag_unconnected_nets* rule parameter to have the `LPSVM08C` rule consider such outputs.

- *lp_flag_pd_outputs*: By default, the LPSVM08C rule flags the outputs of power domain that are directly connected to primary ports. Set the *lp_flag_pd_outputs* rule parameter to 0 to ignore such outputs.
- *lp_skip_buf*: By default, the *lp_skip_buf* parameter is set to 1 and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to 0 to consider SpyGlass-generated buffers during rule checking.
- *lp_skip_pwr_gnd*: By default, the LPSVM08C rule ignores nets connected to power/ground supply. Set the *lp_skip_pwr_gnd* rule parameter to 0 to process such nets also.
- *lp_skip_iso_check_on_ground*: Default value is no. Set this parameter to yes to not report missing isolation strategy and missing isolation cells on the ground nets.
- *lp_check_pwr_gnd_to_macro_without_prd*: Default value is yes. Set this parameter to no to ignore crossings between tie-high/low and macro cell (is_macro_cell: true)/pad cell (pad_cell: true), without domain boundary. Level shifter and isolation checking for these crossings at the RTL and Netlist levels will be ignored.
- *lp_skip_blackbox_checking*: Default value is 0. Set the parameter to 1 to skip checking for black boxes.

Constraint(s)

SDGC

- (Mandatory) *voltage_domain*: The LPSVM08C rule requires you to specify the power domains with the voltage_domain constraint and the name and value of the isolation signal using the -isosig argument and -isoval argument respectively in a SpyGlass Design Constraints file. When a power domain is shut-off, the outputs of the block should be isolated through isolation logic.
- (Optional) *ignore_crossing*: To specify the power domain-to-voltage domain crossings and power domain-to-power domain crossings to be ignored with the ignore_crossing constraint.

- (Optional) *power_state*: You can specify the power domain relationships and automatically find all the `ignore_crossing` constraints with the `power_state` constraint.
- (Optional) *assume_path*
- (Optional) *always_on_buffer*: You can specify the names of always-on buffers using the `always_on_buffer` constraint.
- (Optional) *set_case_analysis*: You can specify the case analysis conditions used by the `LPSVM08C` rule using `set_case_analysis` constraint.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_always_on_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *create_pst* (Optional)
- *add_pst_state* (Optional)
- *map_isolation_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

The following message appears when an isolation cell is not found at the excluded output terminal *<term-name>* of the power-domain *<domain-*

info>:

[LPSVM08C_1][INFO] Excluded output '<term-name>' going from source '<source-name>' (power domain '<pd-name>') to destination '<dest-name>' (domain '<dom-name>') does not have isolation cell placed in its path but need isolation

In CPF and SGDC flow, the domain information (<*domain-info*>) shows the domain name and its voltage value. However, in UPF flow, the domain information shows the domain name along with the associated supply name and supply value.

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not Fixing

The strategy provided at the power domain boundary has `no_isolation` specified, even though isolation is needed and an isolation cell should be placed. This results in design failure.

How to Debug and Fix

The violation is reported at the first place where the output signal of power domain is used.

Double-click the message and click the **Incremental Schematic** button. The schematic shows the output terminal and the power domain. Refer to the Example Code and/or Schematic section for an example.

To fix the violation, ensure that a valid isolation strategy without `no_isolation` is specified at the relevant power domain boundary or an isolation cell is present at the excluded output terminal of the power domain.

Double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

Example Code and/or Schematic

In the following example, a violation occurs when no isolation cell is present at the output terminal `top.pd_inst.out3` of the power domain *shutOFF* and `top.pd_inst.out3` is defined as excluded:

```
Excluded output 'top.pd_inst.out3' going from source
```

'top.pd_inst.buf3.Y' (power domain 'shutOFF(supply VDD_LP:1.20)') to destination 'top.out3' (domain 'top(supply VDD:1.20)') does not have isolation cell placed in its path but need isolation

The schematic is as follows:

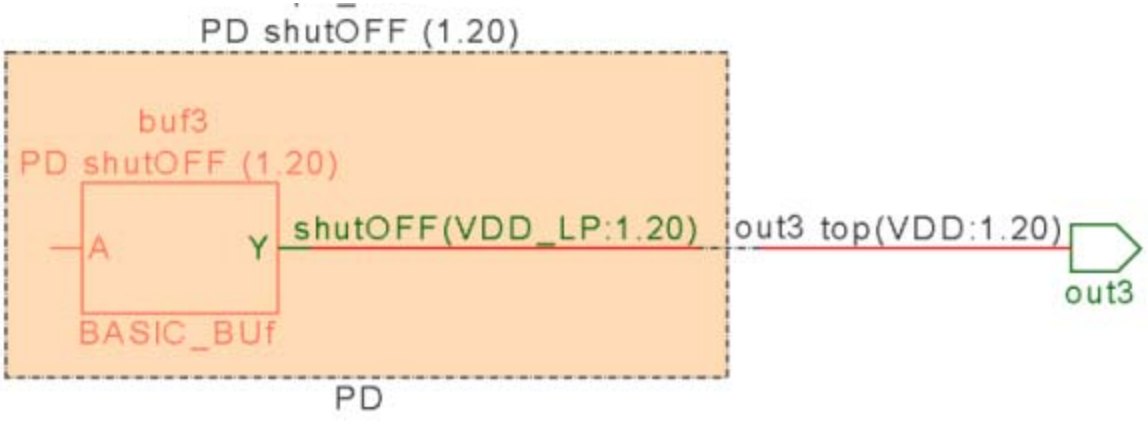


FIGURE 64. Incremental schematic

Default Severity Label

Info

Rule Group

Isolation Logic

Reports and Related Files

None

LPSVM09

Checks if power domain output attains the specified value in power down condition

When to Use

To check for isolation in a multiple power design. This rule is applicable to all design phases after the insertion of isolation. Refer to:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM09* rule checks if the specified power domain output net(s) attains the desired value under the power down condition. By default, the value should be 0. When a power domain is shut off, the outputs of the block should be isolated through isolation logic and the output of isolated signal should be either `0` or `1`.

If you do not specify the `-outputs` argument of the *voltage_domain* constraint, the rule checks for all the outputs for value `0`. You can specify the values using the *domain_outputs* constraint. Refer to the *Example Code and/or Schematic* section for information on how to specify values.

A cell with the *is_isolation_cell* attribute set as `true` in the library is recognized as an isolation cell. A level shifter with an enable pin, clamp level shifter cell, is also treated as isolation cell.

Prerequisites

The *LPSVM08A* rule should run with this rule.

Rule Exceptions

The *LPSVM09* rule does not check for the domain outputs that have missing isolation cells. The *LPSVM08A* rule reports such signals.

Rule Exceptions of the LPSVM08A rule also apply to the LPSVM09 rule.

Language

Verilog, VHDL

Parameter(s)

- *lp_flag_pd_outputs*: Default value is 1. Set the parameter to 0 to ignore the outputs of power domain that are directly connected to primary ports.
- *lp_complex_iso_logic*: Default value is 0. Set the parameter 1 to have the rule assume that isolation cells are complex isolation cells.
- *lp_flag_undriven_nets*: Default value is 0. Set the parameter to 1 to have the rule consider nets that are not driven at voltage crossings.
- *lp_flag_unconnected_nets*: Default value is 0. Set the parameter to 1 to have the rule consider unconnected nets at voltage crossings.
- *lp_skip_buf*: Default value is 1 and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to 0 to consider SpyGlass-generated buffers during rule checking.
- *lp_skip_aon_buf*: Default value is 1. Set the parameter to 0 to not skip always-on buffers to find an isolation crossing.
- *lp_skip_pwr_gnd*: Default value is 1. Set the parameter to 0 to process nets connected to power/ground supply while checking for the correctness of the level shifters.
- *lp_isologic_in_pd*: Default value is 1. Set the parameter to 0 to have the consider the isolation cells outside the power domain only.
- *lp_set_sim_val_x*: Default value is 1. Set this parameter to 0 to enable the rule to propagate the value for *set_case_analysis* and other control signals for the logic that lies in the power domain.

Constraint(s)

SDGC

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage/ power domains in the design.
- *domain_outputs* (Optional): Use this constraint to specify the values of various signals under the steady-state condition specified in the voltage_domain constraint.
- *levelshifter* (Mandatory): Use this constraint to specify the names of design units to be used as level shifters.

- *ignore_crossing* (Optional): Use this constraint to specify the power domain-to-voltage domain crossings and power domain-to-power domain crossings to be ignored with the `ignore_crossing` constraint.
- *power_state* (Optional): Use this constraint to specify the combinations of domain states which can exist at the same time during a design operation.
- *antenna_cell* (Optional): Use this constraint to specify the antennae protection cells (diode cells) which need to be ignored by the rules in the SpyGlass Power Verify solution.
- *always_on_buffer* (Optional): Use this constraint to specify the names of always-on buffers using the `always_on_buffer` constraint.
- *set_case_analysis* (Optional): Use this constraint to specify the case analysis conditions used by the `LPSVM08A` rule using `set_case_analysis` constraint.
- *always_on_cell* (Optional): Use this constraint to specify the always-on cells and their domain information.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_isolation_rule* (Mandatory)
- *update_isolation_rules* (Optional)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_isolation_cell* (Optional)
- *define_always_on_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)

- *set_domain_supply_net* (Mandatory)
- *set_isolation* (Mandatory)
- *set_isolation_control* (Mandatory)
- *map_isolation_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when the signal *<sig-name>* traverses from the domain *<dom-name1>* to <dom-name2> attains <value1> value instead of the specified value <value2>, when the corresponding isolation signal is asserted:

[LPSVM09_2][WARNING] Signal '<sig-name>' going from domain '<dom-name1>' to domain '<dom-name2>' has value '<value1>' instead of user-specified value '<value-2>'

Message 2

The following message appears when the signal *<sig-name>* connected to pin *<pin-name>* (used in the isolation_enable_condition attribute) of instance <instance-path-name> has a value X, instead of 0 or 1:

[LPSVM09_3][ERROR] Signal '<sig-name>' connected to pin '<pin-name>' of instance '<instance-path-name>' ('<domain-and-supply-name>') has value 'X' instead of known '0' or '1' value. Pin is used in isolation_enable_condition attribute

NOTE: *Violation is reported for each pin used in the isolation_enable_condition attribute.*

Potential Issues

This violation can arise because of the following:

- Incorrect or no isolation signal: Refer to the *LPSVM22* rule violation for the corresponding connection.
- Isolation signal traversing through an instance, which is off leading to X on path ahead.

- Isolation cell of incorrect functionality is used. For example, an `OR` gate is used in place of `AND` for forcing value `0`.
- Isolation cell is powered off with respect to the destination.
- Isolation cell output has multiple drivers.

NOTE: *In UPF flow, the violation message specified above explicitly states the reason for the violation to occur.*

Consequences of Not Fixing

The power domain output can have a floating or an incorrect value with respect to the specified value under power down condition.

How to Debug and Fix

The message states that the output of an off voltage domain have an unknown value under power down scenario. The violation is reported at the output of power domain.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Turn on the Power View. The schematic highlights the output of power domain that does not attain a known value, such as 0, 1, or retention, when the isolation signal is asserted.

To resolve this violation, trace back the enable pin of isolation cell to ensure that it is correctly connected to the isolation signal. Check if there are instances, which are turned off. Refer to the `Show Blackbox Schematic` and check that functionality of the isolation cell also.

Refer to the *Example Code and/or Schematic* section to view a sample schematic and to understand this rule.

Refer to the *Reports and Related Files* section for information on the reports generated by this rule.

Example Code and/or Schematic

Example 1

The *LPSVM09* rule checks that during power down, no unknown values propagate from a power domain past the isolation logic. For example, if an isolation cell exists on an output and has an incorrect isolation signal level, such as an AND gate where isolation signal is active high, then an unknown value is propagated even though an isolation gate is present. This rule checks for unknown (X) values at the isolation logic outputs. If AND-type isolation logic is used on these outputs, the outputs of these gates show as

X after the simulation, and these outputs are reported as violations by the rule.

In the following example, to declare a domain as a power domain, two voltage values are specified for a *voltage_domain* constraint.

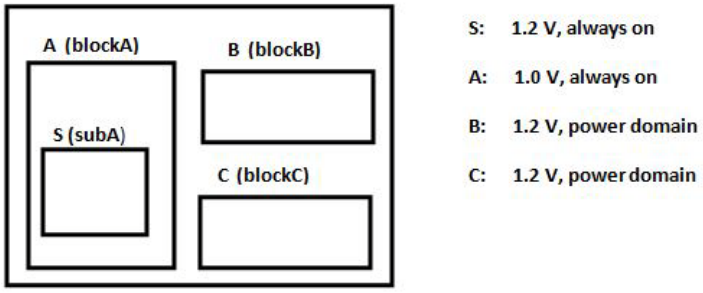


FIGURE 65. Power domain declaration

The related SpyGlass Design Constraints file is as follows:

```
current_design top

voltage_domain -name Vmain -value 1.2 -modname \
top -instname top.A.S

voltage_domain -name VA -value 1.0 -instname \
top.A -isosig enB -isoval 0

voltage_domain -name VB -value 1.2 0 -instname \
top.B -isosig enB -isoval 0

voltage_domain -name VC -value 1.2 0 -instname \
top.C -isosig enC -isoval 0
```

After running the analysis, the following message appears:

```
[WARNING] Signal 'top.Q' going from domain 'VB(1.20 0)' to
domain 'VA(1.0)' has value 'X' instead of user-specified value
'0'
```

Example 2

Consider the following example. The following message appears when the signal `'mid.t1.inst4.b[0]'` traverses from power domain `shutOFF` to

domain AO attains 'X' value instead of the specified value 0, when the corresponding isolation signal is asserted:

```
[WARNING] Signal 'mid.t1.inst4.b[0]' going from domain
'shutOFF(1.20 0)' to domain 'AO(1.20)' has value 'X' instead of
user-specified value '0'
```

The incremental schematic is displayed as shown in figure below.

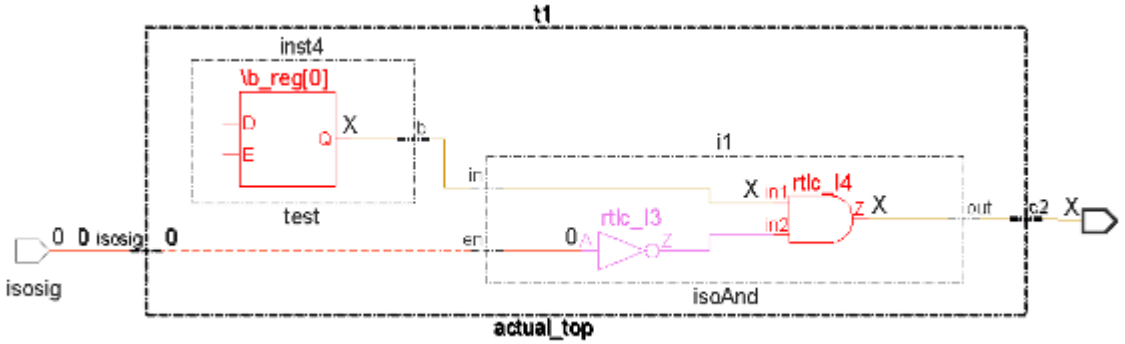


FIGURE 66. Incremental schematic

The schematic for the message shows the following:

- the connection between the two domains
- the source voltage domain and its value
- the destination voltage domain and its value
- (Optional) the (incorrect) isolation cell is highlighted in a different color
- the correct isolation signal (source net) and its isolation value (under power down scenario)

Example 3

The example shows how to specify the values using the *domain_outputs* constraint. For example:

```
voltage_domain -name V3 -value 1.2 0 -instname mc_top.u2 -
isosig susp_req_i -isoval 1 -outputs PD_V3_OUT

domain_outputs -name PD_V3_OUT -value obct_cs 1
```

In this example, the rule flags if mc_top.u2.obct_cs does not attain 1,

when the voltage domain V3 is switched off

Default Severity Label

Warning, Error

Rule Group

Power_Domain_Rules

Reports and Related Files

- *lp_crossing_data*: Shows the valid isolation cell for the combination of source and destination voltage domains. Refer to the following information in the report:
  - List of isolated output value in the Clamp Value field.
  - List of correct isolation signals that the user has specified for this crossing in the Output Isolation Info field.

LPSVM10

Displays values on the output of power domain under shutdown condition

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM10* rule reports and highlights steady-state values at the output of the power domains.

The *LPSVM10* rule highlights the following cases:

When the isolation signal and the isolation value are correctly specified and the correct value (`0` or `1`) is achieved at the output of the isolation signal.

Then, the instance in the power domain with the connecting net and the isolation cell is highlighted.

When the isolation signal is correctly specified but the isolation value is incorrectly specified, therefore an incorrect value (`X`) is achieved at the output of the isolation signal.

Then, the instance in the power domain with the connecting net and the isolation cell is highlighted.

When the isolation signal is incorrectly specified, therefore an incorrect value (`X`) is achieved at the output of the isolation signal.

Then, only the instance in the power domain with `X` value at the output of the power domain is highlighted. The `X` value is not propagated to the isolation cell and therefore the connecting net and the isolation cell are not highlighted.

In case, the output of the power domain is a bus signal, no value is annotated as its individual bits may have different values, which can be seen by descending in to the hierarchy.

Rule Exceptions

By default, the *LPSVM10* rule skips instances of level-shifters, specified with the *levelshifter* constraint, at the output of power domains. If the level-shifter has isolation capability (that is, the enableTerm argument of the *levelshifter* constraint is also specified), then the *LPSVM10* rule also considers such level-shifter instances.

When a power domain is shut-off, the outputs of the block should be isolated through isolation logic and the output of isolated signal should be 0, 1, or retention if correct, else X if incorrect.

If an always-on buffer is present at the power domain crossing, the *LPSVM10* rule ignores the power domain crossing and the steady state value at the crossing is not reported.

Language

Verilog, VHDL

Parameter(s)

- *lp_flag_pd_outputs*: Default value is `1`. Set to `0` to ignore outputs of power domain that are directly connected to primary ports
- *lp_complex_iso_logic*: Default value is `0`. Set to `1` to assume that isolation cells are complex cells.
- *lp_flag_undriven_nets*: Default values is `0`. Set to `1` to consider outputs that are not driven.
- *lp_flag_unconnected_nets*: Default values is `0`. Set to `1` to consider the unconnected outputs of power domains.
- *lp_skip_buf*: Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.
- *lp_skip_pwr_gnd*: Default value is `1`. Set to `0` to consider nets connected to power/ground supply.
- *lp_isologic_in_pd*: Default values is `1`. Set to `0` to consider the isolation cells outside the power domain only.
- *lp_set_sim_val_x*: Default values is `1`. Set to `0` to enable the propagation of the value for *set_case_analysis* and isolation control signals also for the logic that lies in power domain.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the power domains using the *voltage_domain* constraint in an SGDC file.
- *levelshifter* (Optional): Use to specify the names of design units to be used as level shifters.
- *ignore_crossing* (Optional): Use to specify the power domain-to-voltage domain crossings and power domain-to-power domain crossings to be ignored.
- *power_state* (Optional): Use to specify the power domain relationships and automatically find all the *ignore_crossing* constraints.
- *assume_path* (Optional): Use to specify the paths that exist between the input pins and the output pins of black boxes.
- *always_on_buffer* (Optional): Use to specify the names of always-on buffers.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_isolation_rule* (Mandatory)
- *update_isolation_rules* (Optional)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_isolation_cell* (Optional)
- *define_always_on_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)

- *set_isolation* (Mandatory)
- *set_isolation_control* (Mandatory)
- *map_isolation_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

The following message appears when highlight information is generated for a power domain *<pd-name>* having power values *<pd-value-list>*:

[LPSVM10_1][INFO] Steady state values for the outputs of power domain '<pd-name>(<pd-value-list>)' is reported

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

As this is an informational message, there is no implicit impact of not fixing this violation message.

How to Debug and Fix

The violation is reported at the first place where the output signal is used.

In the Incremental Schematic window, while debugging an isolation logic rule, you can view the value at the output of power domain along with the violation of other rules. For this, you can do the following:

First select the rule to be debugged and then double-click the violation message of the *LPSVM10* rule while holding down the *<Ctrl>* key.

Select the rule violation and open the Incremental Schematic window.

Further details about working with multiple messages can be found in the Atrenta Console User Guide.

Example Code and/or Schematic

The *LPSVM10* rule highlights steady-state values at the output of the power domains.

Consider the following example:

When highlight information is generated for a power domain V2, the

following message appears:

```
Steady state values for the outputs of power domain 'V2(1.20
0)' is reported
```

The schematic is as follows:

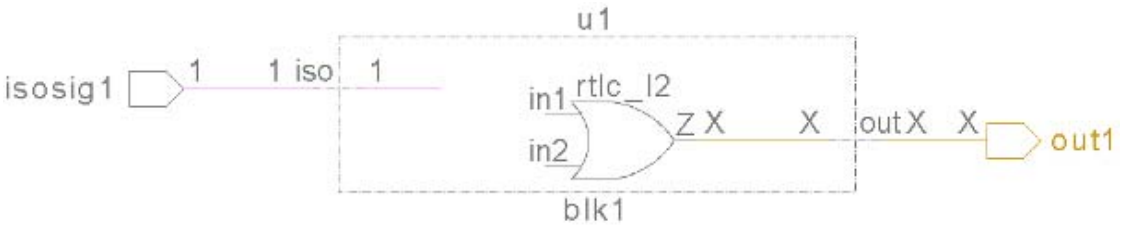


FIGURE 67. Incremental schematic

When a power domain is shut-off, the outputs of the block should be isolated through isolation logic and the output of isolated signal should be either 0 or 1 if correct, else X if incorrect. This rule reports the steady state values at the output of the power domain.

Default Severity Label

Info

Rule Group

Isolation Logic

Reports and Related Files

- *lp_vd_info*: Contains the steady state values at the output of the power domain is specified.

LPSVM12

Isolation signal should be a state signal and in always-on domain

Language

Verilog, VHDL

Rule Description

The LPSVM12 rule checks the isolation signal, which is the enable of an isolation cell. The rule checks both input side and output side isolation signals specified for power domains.

The LPSVM12 rule is divided into two sub-rules - *LPSVM12A*, and *LPSVM12B*. When you select the LPSVM12 rule, both these rules are run.

NOTE: *The LPSVM12 rule does not flag the enable of an isolation cell if it is coming from a primary input of an always-on domain.*

NOTE: *A cell with the is_isolation_cell attribute set as* `true` *in the library, is recognized as an isolation cell.*

NOTE: *A level shifter with an enable pin (clamp level shifter cell) that is tied to a constant value is not treated as an isolation cell.*

LPSVM12A

Checks that the isolation signal always comes from an always-on domain

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM12A* rule checks whether the isolation signal is coming from an always-on domain or a relatively-on domain. This rule checks both the input side and the output side isolation signals specified for power domains. If the *lp_allow_iso_sig_from_equivalent_domain* parameter is set to `yes`, the isolation signals from equivalent domains are also considered valid.

Language

Verilog, VHDL

Parameter(s)

- *lp_skip_buf*: Default value is 1 and the SpyGlass-generated buffers are skipped in a crossing. Set the parameter to 0 to consider SpyGlass-generated buffers.
- *lp_allow_iso_sig_from_equivalent_domain*: Default value is `no`. Set this parameter to `yes` to avoid violations for equivalent domains.
- *lp_set_design_stage*: Default value is `rtl`. Set this parameter to `netlist` or `pg_netlist` to specify the desired design stage.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/power domains and its information is used by *SP_01* rule.

- *set_case_analysis* (Optional): Use to specify the case analysis conditions.

CPF Commands

- *create_power_domain* (Mandatory): Use to create a power domain.
- *create_isolation_rule* (Mandatory): Use to create a rule for adding an isolation cell.
- *update_isolation_rules* (Mandatory): Use to update an isolation rule.
- *create_nominal_condition* (Mandatory): Use to infer voltage values for domains in a power state.
- *create_power_mode* (Mandatory): Use to create a power mode.

UPF Commands

- *create_power_domain* (Mandatory): Use to create power domain.
- *create_supply_port* (Mandatory): Use to create supply port.
- *add_port_state* (Mandatory): Use to state a port.
- *create_supply_net* (Mandatory): Use to create a supply net.
- *connect_supply_net* (Mandatory): Use to connect a supply net with a supply port.
- *set_domain_supply_net* (Mandatory): Use to specify a primary power and ground supply nets for a power domain.
- *set_isolation* (Mandatory): Use to create an isolation rule.
- *set_isolation_control* (Mandatory): Use to specify the control signal for an isolation rule.

Messages and Suggested Fix

Message 1

The following message appears, when the isolation signal <*sig-name*> is not coming from an `'always-on'` domain:

[LPSVM12A_1][RECOMMENDED] `Isolation signal '<sig-name>' is not coming from an always-on block`

Message 2

The following message appears when the isolation signal <*sig-name*>

has a receiver, which is placed in an equivalent/relatively-off domain:

[LPSVM12A_2][RECOMMENDED] Isolation signal '<sig-name>' defined for domain '<dom-name1>' has receiver '<pin-name>' placed in equivalent/relatively-off domain <dom-name2>

Message 3

The following message appears when the isolation signal is unconnected at all of it's fan-outs. Buffers and inverters (both RTLC and Library) are skipped while traversing on fan-out of an isolation signal.

[LPSVM12A_3][RECOMMENDED] Isolation signal '<sig-name>' is unconnected

NOTE: *If at least one of the fan-outs of isolation signal is connected, the LPSVM12A rule will not report the violation.*

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

Message 1: The isolation signal does not originate from an always-on domain. Therefore, this is a design flaw.

Message 2: The isolation signal has a receiver, which is placed in an equivalent/relatively-off domain. Therefore, this is a design flaw.

Message 3: There may be a scenario where an isolation signal is unconnected on its fanout and it can lead to design failure at later stage.

How to Debug and Fix

The violation is reported at the first place where the isolation signal is not a state signal.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix these violation messages, ensure that the enable of an isolation cell should come from an always-on domain and also ensure that it does not have a driver in an equivalent/relatively-off domain.

For **Message 3**, make sure that at least one fan-out of the isolation signal is connected.

Example Code and/or Schematic

Example 1

In the following example, a violation occurs when the isolation signal `'actual_top.isoinst.temp'` does not come from an '`alwayson`' domain:

```
Isolation signal 'actual_top.isoinst.temp' is not coming from
an always-on block
```

The schematic is as follows:

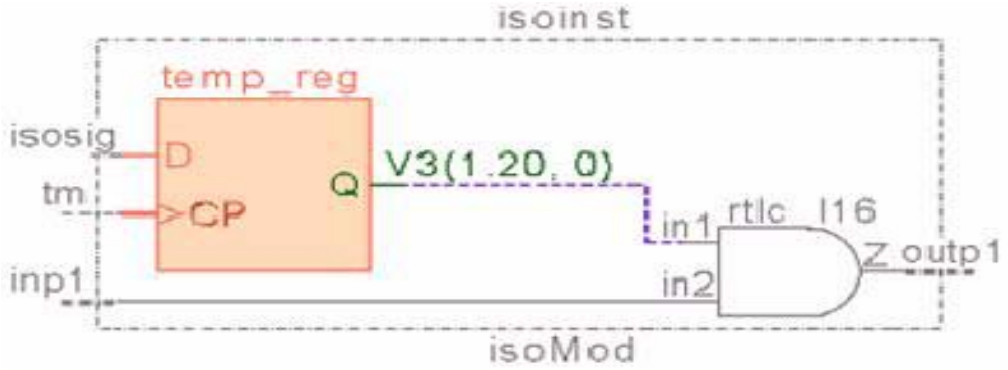


FIGURE 68. Incremental schematic

Here, the isolation signal 'actual_top.isoinst.temp' does not belong to an always-on domain. This isolation signal will serve no purpose when isolation will be required.

Example 2

Consider the following UPF snippet:

```
set_isolation iso1 -domain VD3 -isolation_power_net VDD -
applies_to inputs -location parent -clamp_value 0
set_isolation_control iso1 -domain VD3 -isolation_signal iso
-isolation_sense low -location parent
```

In the above UPF snippet, the following violation occurs because the `top.iso` signal is unconnected:

```
Isolation signal 'top.iso' is unconnected
```

The schematic for this example is as follows:

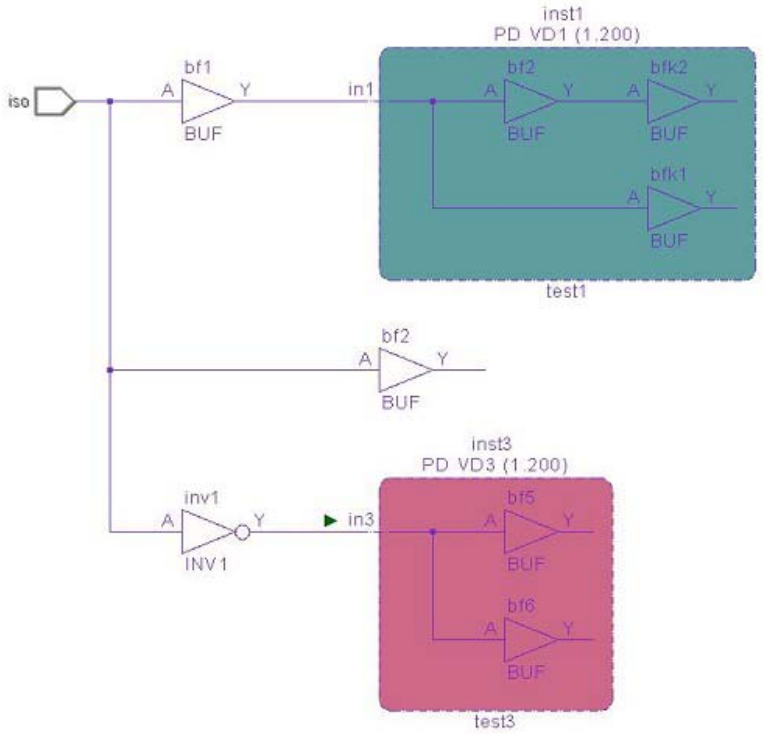


FIGURE 69. Incremental schematic

Default Severity Label

Recommended

Rule Group

Isolation Logic

Reports and Related Files

None

LPSVM12B

Ensures that the isolation signal is always a state signal

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM12B* rule checks whether the isolation signal is the output of a sequential element. The rule checks both the input side and the output side isolation signals specified for power domains.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/power domains and its information is used by *SP_01* rule.
- *set_case_analysis* (Optional): Use to specify the case analysis conditions.

CPF Commands

- *create_power_domain* (Mandatory): Use to create a power domain.
- *create_isolation_rule* (Mandatory): Use to create a rule for adding an isolation cell.
- *update_isolation_rules* (Mandatory): Use to update an isolation rule.
- *create_nominal_condition* (Mandatory): Use to infer voltage values for domains in a power state.

- *create_power_mode* (Mandatory): Use to create a power mode.

UPF Commands

- *create_power_domain* (Mandatory): Use to create power domain.
- *create_supply_port* (Mandatory): Use to create supply port.
- *add_port_state* (Mandatory): Use to state a port.
- *create_supply_net* (Mandatory): Use to create a supply net.
- *connect_supply_net* (Mandatory): Use to connect a supply net with a supply port.
- *set_domain_supply_net* (Mandatory): Use to specify a primary power and ground supply nets for a power domain.
- *set_isolation* (Mandatory): Use to create an isolation rule.
- *set_isolation_control* (Mandatory): Use to specify the control signal for an isolation rule.

Messages and Suggested Fix

The following message appears, when an isolation signal *<sig-name>* is not driven by a flip-flop:

[LPSVM12B_1][RECOMMENDED] Isolation signal '<sig-name>' is not a state signal

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

The isolation signal is not a state signal (that is, not the output of a flip-flop). This is a design flaw.

How to Debug and Fix

The violation is reported at the first place where the isolation signal is not a state signal.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix the violations ensure that the enable of isolation cell should be a state signal.

Example Code and/or Schematic

In this example, a violation occurs when the isolation signal `isosig` is not a state signal:

[RECOMMENDED] `Isolation signal 'isosig' is not a state signal`

The schematic is as follows:

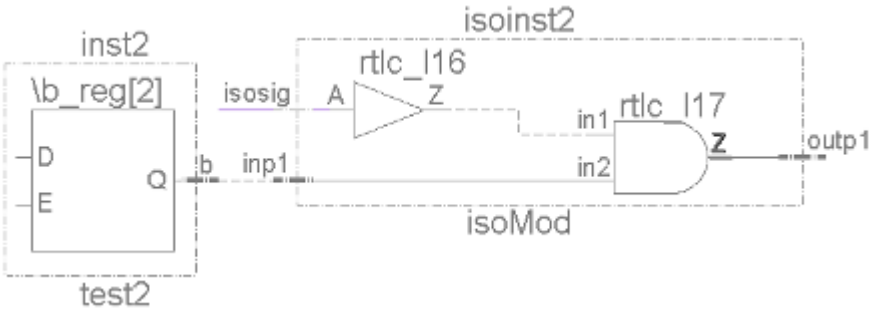


FIGURE 70. Incremental schematic

Here, the isolation signal `isosig` is not the output of a flip-flop. The *LPSVM12B* rule reports this issue.

Default Severity Label

Recommended

Rule Group

`Isolation Logic`

Reports and Related Files

None

LPSVM15

Reports power domains that have multiple enable signals

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM15* rule reports those power domains that have multiple isolation signals.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/power domains and its information is used by *SP_01* rule.
- *set_case_analysis* (Optional): Use to specify the case analysis conditions.

Messages and Suggested Fix

The following message appears when multiple enable signals have been specified and used for the Power Domain *<pd-name>*:

[LPSVM15_1][RECOMMENDED] More than one enable signal detected for Power Domain '<pd-name>'

Potential Issues

The *LPSVM15* rule reports power domains for which multiple isolation

signals have been specified using the `-isosig` argument of *voltage_domain* constraint.

Consequences of Not Fixing

A particular power domain should have only one isolation signal specified for it. Here, multiple isolation signals have been specified for the same power domain. This is a flaw in power intent.

How to Debug and Fix

The violation is reported at the place where isolation signal is set or used.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix the violation, ensure that only a single isolation signal is specified for all Power domains.

Example Code and/or Schematic

In the following example, a violation occurs when more than one enable signal have been specified and used for Power Domain `'V2'`:

```
[RECOMMENDED] More than one enable signal detected for Power
Domain 'V2'
```

The schematic is as follows:

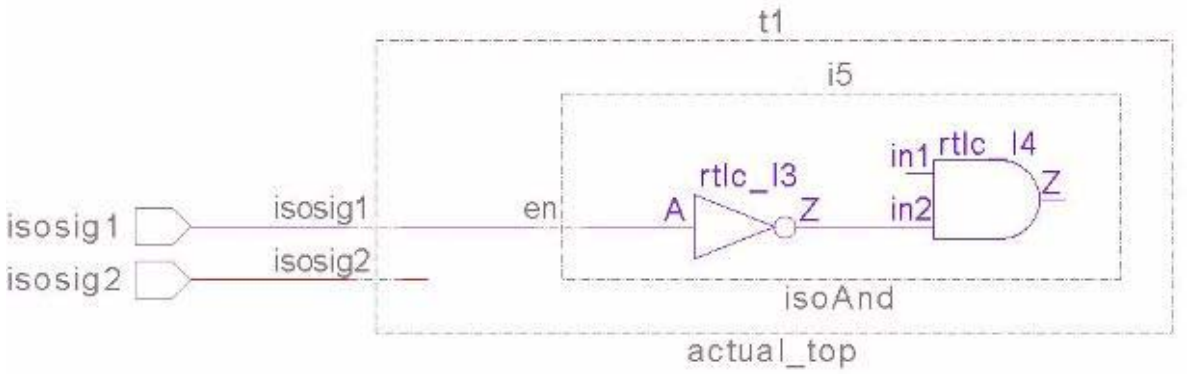


FIGURE 71. Incremental schematic

The influence of each isolation signal is highlighted using a different color.

Default Severity Label

Recommended

Rule Group

Isolation Logic

Reports and Related Files

None

LPSVM22

Reports inadvertently used isolation cells in the design

When to Use

To check for isolation in a multiple power design. This rule is applicable to all design phases after the insertion of isolation. Refer to:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM22* rule reports inadvertently used or improperly used isolation cells in the design. A cell with the *is_isolation_cell* attribute set as `true` in the library, is recognized as an isolation cell.

This rule reports instances of specified isolation cells in the following cases:

- Isolation signal is not reaching the enable pin of the isolation cell instance
- Out-of-place (inadvertent): When the input/output of the isolation cell instance is not connected to a power domain.
- Wrongly placed: When the input/output of the isolation cell instance is connected to a valid power domain, but the isolation signal of this domain is not connected to isolation cell.

Prerequisites

The *LPSVM22* rule requires you to specify the names of isolation cells for each power domain in any of the following ways:

- Using the *isolation_cell* or *input_isocell* constraint in a SpyGlass Design Constraints file.
- Using the *is_isolation_cell* attribute in the library file.

NOTE: *A level shifter with an enable pin (clamp level shifter cell) that is tied to a constant value is not treated as an isolation cell.*

Rule Exceptions

The *LPSVM22* rule ignores those cells that are used as a level shifter

without isolation. In addition, this rule ignores isolation cells with an undriven data pin or unconnected output pin.

Language

Verilog, VHDL

Parameter(s)

- *lp_disable_lib_attr_read*: Default value is `0`. Set this parameter to `1` to have the rule ignore library cells with *is_isolation_cell* attribute and consider only cells specified with *input_isocell*/*isolation_cell* constraint as input-side/output-side isolation logic.
- *lp_skip_buf*: Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.
- *lp_skip_pwr_gnd*: Default value is `1`. Set the parameter to `0` to process nets connected to power/ground supply while checking for the correctness of the isolation cells.
- *lp_check_pwr_gnd_to_macro_without_prd*: Default value is `yes`. Set this parameter to `no` to ignore crossings between tie-high/low and macro cell (`is_macro_cell: true`)/pad cell (`pad_cell: true`), without domain boundary. Level shifter and isolation checking for these crossings at the RTL and Netlist levels will be ignored.
- *lp_skip_blackbox_checking*: Default value is `0`. Set the parameter to `1` to skip checking for black boxes.
- *lp_skip_combo_cell_for_reference_toplevel_isolation_signal*: Default value is `0`. Set this parameter to `1` to skip combinational cells whose output pin is related to signal input pin to get correct source of control signal.
- *lp_skip_same_src_supply_buf*: Default value is `no`. Set this parameter to `yes` to enable the rule skip buffers and inverters between strategy node and isolation cell which are of same supply as source but are LESS ON w.r.t. destination.

Constraint(s)

SDGC

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage/power domains in the design.
- *isolation_cell* (Optional): Use this constraint to specify the isolation cells in power domains.
- *input_isocell* (Optional): Use this constraint to specify the isolation cells at inputs of a power domain.
- *ignore_crossing* (Optional): Use this constraint to specify the power domain-to-voltage domain crossings and power domain-to-power domain crossings to be ignored with the `ignore_crossing` constraint.
- *power_state* (Optional): Use this constraint to specify the combinations of domain states which can exist at the same time during a design operation.
- *reference_toplevel_isolation_signal* (Optional): Use this constraint to specify the reference top-level isolation signal at the SoC level.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_isolation_rule* (Mandatory)
- *update_isolation_rules* (Optional)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_isolation_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)

- *set_isolation* (Mandatory)
- *set_isolation_control* (Mandatory)
- *map_isolation_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when no isolation signal is reaching the enable pin <pin-name> for an instance *<inst-name>* of the isolation cell *<cell-name>*:

[LPSVM22_4][WARNING] No isolation signal is reaching the enable pin '<pin-name>' of isolation cell <cell-name>('<inst-name>') used in destination ‘<dest-name>’ (domain '<dom-name1>') driven by source ‘<source-name>’ in domain '<dom-name2>'

For debugging information, click *How to Debug and Fix*.

Message 2

The following violation when the isolation signal is reaching the data pin of the isolation cell:

[LPSVM22_8][WARNING] Isolation signal <iso-sig-name> is connected to data pin <pin-name> of isolation cell <cell-name>

For debugging information, click *How to Debug and Fix*.

Message 3

The following messages appear when an instance *<inst-name>* of the isolation cell *<cell-name>* has been used out-of-place:

[LPSVM22_2][WARNING] Isolation cell '<cell-name>' instantiated as '<inst-name>' has been used out of place: isolation cell is not required from source ‘<source-name>’ ('<dom-name1>' domain) to destination ‘<dest-name>’ ('<dom-name2>' domain)

For debugging information, click *How to Debug and Fix*.

Message 4

[LPSVM22_1][WARNING] Isolation cell '<cell-name>' instantiated

as '<inst-name>' has been used out of place: no isolation specified from source ‘<source-name>’ ('<dom-name1>' domain) to destination ‘<dest-name>’ ('<dom-name2>' domain)

For debugging information, click *How to Debug and Fix*.

Message 5

[LPSVM22_5][WARNING] Isolation cell '<cell-name>' instantiated as '<inst-name>' has been used out of place: isolation cell is not required from source ‘<source-name>’ ('<dom-name1>' domain) to destination ‘<dest-name>’ ('<dom-name2>' domain) according to Power state Table

For debugging information, click *How to Debug and Fix*.

NOTE: *This violation message appears only in the UPF flow.*

Message 6

[LPSVM22_6][WARNING] Isolation cell '<cell-name>' instantiated as '<inst-name>' has been used out of place: isolation strategy is not specified from source ‘<source-name>’ ('<dom-name1>' domain) to destination ‘<dest-name>’ ('<dom-name2>' domain)

For debugging information, click *How to Debug and Fix*.

NOTE: *This violation message appears only in the UPF flow.*

Message 7

[LPSVM22_7][WARNING] Isolation cell '<cell-name>' instantiated as '<isolation-cell-path>' has been used out of place: isolation cell is not required from '<source-name>' (‘<dom-name1>’ domain) to destination ‘<dest-name>’ ('<dom-name2>’ domain) as is_isolated attribute set on destination side '<destination-pin-path>'

For debugging information, click *How to Debug and Fix*.

Message 8

The following message appears when a correct isolation signal is not reaching the isolation cell *<cell-name>* (*'<inst-name>'*):

[LPSVM22_3][WARNING] Correct isolation signal is not reaching the isolation cell '<cell-name>' ('<inst-name>') used from source ‘<source-name>’ ('<dom-name1>' domain) to destination ‘<dest-name>’ ('<dom-name2>') domain. Correct isolation signal

is '<signal-name>'

For debugging information, click *How to Debug and Fix*.

Message 9

The following message is reported if the signal reaching the hard macro pins, with the `is_isolated` attribute set to true in the library, is equally-on or more-on with respect to the pin itself:

[LPSVM22_9][WARNING] Crossing from source <source-node>('<source-domain-name>') to destination <destination-node>('<destination-domain-name>') does not need isolation, has 'is_isolated' attribute set at the <source | destination> side

Where, the `is_isolated` attribute is provided on lib cell pins to specify that the pin is internally isolated and does not need isolation.

For debugging information, click *How to Debug and Fix*.

Message 10

The following message is reported if no reference top-level isolation signal is found while traversing isolation enable signal from an isolation cell:

[LPSVM22_10][ERROR] Signal ‘<signal-name>‘ reaching isolation enable pin ‘<pin-name>‘ of isolation cell ‘<cell-name>‘ is not defined as a reference top level isolation signal

For debugging information, click *How to Debug and Fix*.

Message 11

The following message is reported if a reference top-level isolation signal is found while traversing isolation enable signal from an isolation cell, and if the data pin of the isolation is driven by an instance that is powered by a supply other than the one mapped using the *reference_toplevel_isolation_signal* constraint:

[LPSVM22_11][ERROR] Supply ‘<supply-name-from-constraint>‘ specified for reference top level isolation signal ‘<signal-name>‘ does not match the source supply ‘<source-supply-name>‘ for isolation cell ‘<name-of-isolation-cell>’

For debugging information, click *How to Debug and Fix*.

Potential Issues

This violations can arise because of the following:

- an isolation cell is used inadvertently (out of place) as it is not needed from the source voltage domain to the destination voltage domain
- no isolation signal is reaching the isolation cell enable pin
- isolation signal reaching the isolation cell enable pin is incorrect

Message 6 appears because no strategy can be specified, as their is no port on the domain boundary in the crossing, and the isolation cell is also inserted as isolation is required for that crossing, but the destination side pin already has the *is_isolated* attribute set on it. Therefore, the isolation cell is redundant.

Consequences of Not Fixing

Messages 1-9: The isolation cell with incorrect/no isolation signal would have an incorrect or a floating value under power down mode and when used out-of-place the isolation cell may change the logic value.

Messages 10: If no isolation signal is reaching the isolation cell, the isolation cell fails to provide isolation. As a result undefined values are propagated at the output of the domain in power down conditions.

Messages 11: If incorrect isolation signal is reaching the isolation cell, the isolation cell fails to provide correct isolation. As a result undefined values can be propagated at the output of the domain in power down conditions.

How to Debug and Fix

The message states details of the source voltage domain, destination voltage domain, and their voltage value. The violation is reported at the first place where isolation cell is incorrectly used.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Turn on the Power View. The schematic highlights all incorrectly used isolation cell instances.

The *LPSVM22* rule also generates a file named LPSVM22.csv. This file contains all the messages generated by the *LPSVM22* rule. To view the spreadsheet, in the Message Tree window, click the Open Spreadsheet option in the right-click menu of the rule.

To resolve these violations, ensure that the correct isolation cell is used correctly at the output of power domains. In addition, ensure that no isolation cell is specified when the destination side pin has the *is_isolated* attribute set on it. Update the RTL file accordingly.

Refer to the *Example Code and/or Schematic* section to view a sample

schematic and to understand this rule.

Refer to the *Reports and Related Files* section for information on the reports generated by this rule.

For **Messages 1, 3, 4, 5, 6, 7, 8** and **9**, double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

For **Messages 10** and **11,** ensure that the correct isolation enable signal, which is reaching the enable pin of an isolation cell, is defined as a reference top level isolation signal using the *reference_toplevel_isolation_signal* SGDC constraint, at the SoC level. Also ensure that the supply mentioned in the *reference_toplevel_isolation_signal* constraint matches with the source supply that the isolation cell is isolating.

Example Code and/or Schematic

Example 1

Consider the following example. A violation occurs when an instance `TOP.iniso1` of the isolation cell `BASIC_LS_EN` has been used out-of-place. The following message appears:

```
Isolation cell 'BASIC_LS_EN' instantiated as 'TOP.iniso1' has
been used out of place: isolation cell is not required from
source 'TOP.pad1.P5' ('TOP(supply VDDH:1.200)' domain) to
destination 'TOP.pad2.P1' ('TOP(supply VDDL:0.800)' domain)
according to Power state Table
```

The incremental schematic is displayed as shown in figure below:

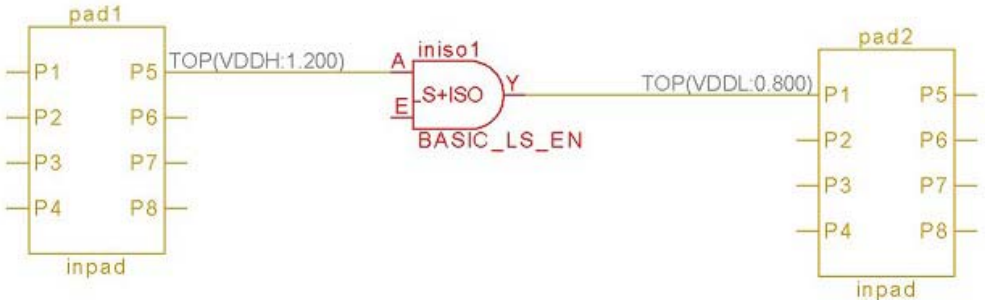


FIGURE 72. Incremental schematic

Example 2

Consider the following example. A violation occurs when an instance `TOP.i_dss_iso.u0_dsi_analogplldiv_iso.ISO1` of the isolation cell `ISO_AND_EN` has been used out-of-place. The following message appears:

```
Isolation cell 'ISO_AND_EN' instantiated as
'TOP.i_dss_iso.u0_dsi_analogplldiv_iso.ISO1' has been used out
of place: isolation cell is not required from source
‘TOP.i_dss_iso.u0_dsi_analogplldiv_iso.rtlc_I3.Z’
('DOMAIN_A(supply VDDL:0.80)' domain) to destination
‘TOP.i_dss_iso.u0_dsi_analogplldiv_iso.rtlc_I5.in2’ ('DOMAIN_A
(supply VDDL:0.80)' domain)
```

The incremental schematic is displayed as shown in figure below:

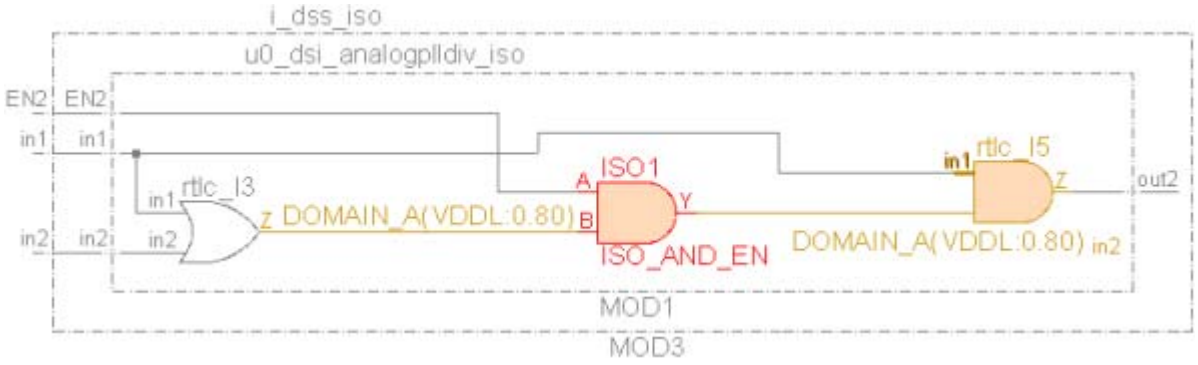


FIGURE 73. Incremental schematic

The schematic for the message shows the following:

- the connection between the two domains
- the source voltage domain and its value
- the destination voltage domain and its value
- the (incorrect) isolation cell is highlighted in a different color.

Example 3

Consider the following example.

SGDC:

```
current_design top
power_data -format upf -file const.upf

reference_toplevel_isolation_signal -name top.iso1 -supply
blk1/VDD1
```

UPF:

```
upf_version 2
set upf_dir .
set_design_top top

....
```

```
set_scope /blk1
set_isolation iso1 -domain PD1 -applies_to outputs  -
clamp_value { 0 }  -isolation_supply_set { ss }  -
isolation_signal { iso }  -isolation_sense { low }
....
```

For the above example, the following violation message is reported because no reference isolation signal is reaching `top.blk1.inst1.iso1` signal:

```
Signal 'top.iso' reaching isolation enable pin 'EN' of
isolation cell 'top.blk1.inst1.iso1' is not defined as a
reference top level isolation signal
```

The incremental schematic is displayed as shown in figure below:

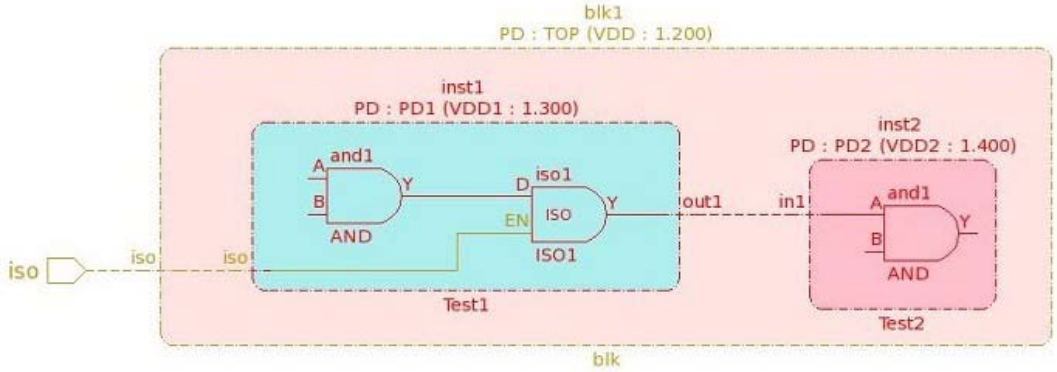


FIGURE 74. Incremental schematic

Example 4

Consider the following example.

SGDC:

```
current_design top
```

```
power_data -format upf -file const.upf

reference_toplevel_isolation_signal -name top.iso -supply
blk_top/blk1/VDD2
```

UPF:

```
upf_version 2
set upf_dir .
set_design_top top
....

set_scope /blk_top/blk1
set_isolation iso1 -domain PD1 -applies_to outputs  -
clamp_value { 0 }  -isolation_supply_set { ss }  -
isolation_signal { iso }  -isolation_sense { low }
....
```

For the above example, the following violation message is reported because supply provided in the SGDC constraint does not match with the supply associated with of the driver of the isolation:

```
Supply '/blk_top/blk1/VDD2' specified for reference top level
isolation signal 'top.iso' does not match the source supply '/
blk_top/blk1/VDD1' for isolation cell
'top.blk_top.blk1.inst1.iso1'
```

The incremental schematic is displayed as shown in figure below:

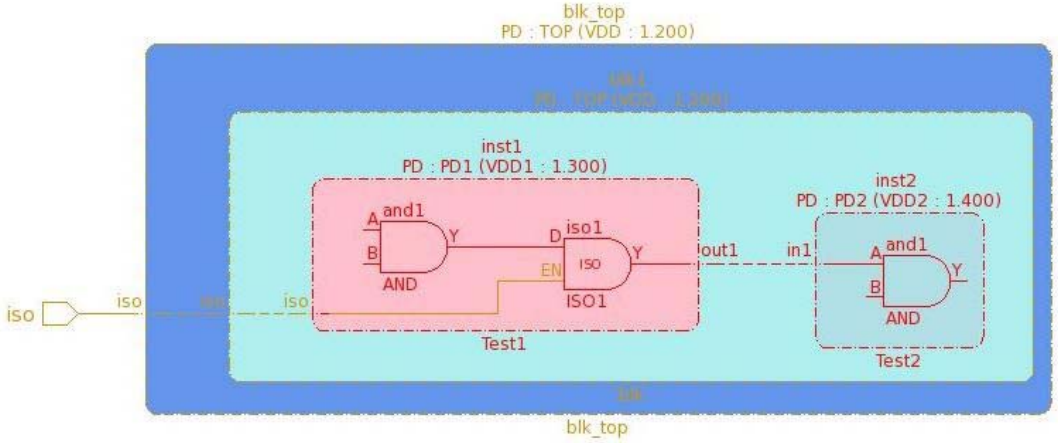


FIGURE 75. Incremental schematic

Default Severity Label

Warning/Error

Rule Group

Power_Domain_Rules

Reports and Related Files

- *lp_crossing_data*: Shows the valid isolation cell for the combination of source and destination voltage domains. The reports contains the following information:
  - the list of isolation cells specified for the domain crossings
  - the `Isolation Required` field will show `None`, in case isolation cell is not needed
  - the list of pins, which are specified as excluded pins

LPSVM26

Checks whether logic cells in the power domain in feedthrough between two voltage domain crossings exist

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM26* rule checks presence of logic cells, such as buffers and inverters, in the power domain in feedthrough between two voltage domains. It is required that there are no buffers/inverters in the power domain on the path of the signal originating and terminating in voltage domains.

The *LPSVM26* rule reports a violation only if the polarity does not change through this path. This rule reports a message if there are even number of inverters in the feedthrough path.

NOTE: *A level shifter with an enable pin (clamp level shifter cell) that is tied to a constant value is not treated as an isolation cell.*

Prerequisites

Specify the voltage domains and the power domains with the *voltage_domain* constraint.

Rule Exceptions

The *LPSVM26* rule does not report a violation if only always-on buffers are present in the feedthrough between two voltage domains.

Language

Verilog, VHDL

Parameter(s)

- *lp_skip_buf*: Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/power domains and its information is used by *SP_01* rule.
- *always_on_buffer* (Optional): Use to specify the always-on buffers.

CPF Commands

- *create_power_domain* (Mandatory): Use to create a power domain.
- *create_nominal_condition* (Mandatory): Use to infer voltage values for domains in a power state.
- *create_power_mode* (Mandatory): Use to create a power mode.
- *define_always_on_cell* (Optional): Use to identify the library cells in the .lib files that can be used as always-on cells.

UPF Commands

- *create_power_domain* (Mandatory): Use to create power domain.
- *create_supply_port* (Mandatory): Use to create supply port.
- *add_port_state* (Mandatory): Use to state a port.
- *create_supply_net* (Mandatory): Use to create a supply net.
- *connect_supply_net* (Mandatory): Use to connect a supply net with a supply port.
- *set_domain_supply_net* (Mandatory): Use to specify a primary power and ground supply nets for a power domain.

Messages and Suggested Fix

The following message appears when the feedthrough from the voltage domain *<vd1-name>* to the voltage domain *<vd2-name>* has logic cells in the power domain *<pd-name>*:

[LPSVM26_1][WARNING] Feedthrough from source '<source-name>' voltage domain '<vd1-name>' to destination '<destination-name>' voltage domain '<vd2-name>' has logic cells in the power domain '<pd-name>'

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

Power domains can be switched off. Therefore, the path between the two voltage domains will be in OFF state and this will result in design failure.

How to Debug and Fix

The violation is reported where logic cell in power domain is used.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. This rule supports IS Abstraction between start and end points and abstracts all logic between these points, except level shifter cells. Refer to the *Abstraction between Start and End Points* topic in the *Atrenta Console Reference Guide*. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, ensure that there are no logic cells in the power domain in feedthrough between two voltage domains.

Example Code and/or Schematic

The *LPSVM26* rule highlights logic cells in the power domain in feedthrough between two voltage domain crossings. In the following example, a violation occurs when feedthrough from voltage domain V2 to voltage domain V1 has logic cells in the power domain V3. The following message appears:

[WARNING] Feedthrough from voltage domain 'V2' to 'V1' has logic cells in the power domain 'V3'

The schematic generated is as follows:

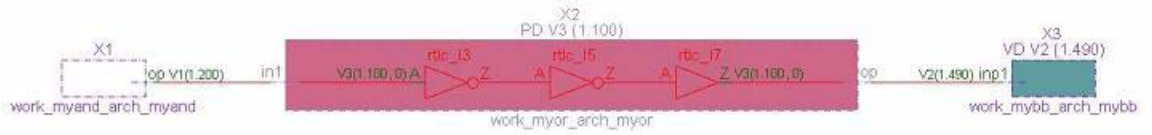


FIGURE 76. Incremental schematic

Here, the instance ‘X2’ in the feedthrough path between ‘X1’ and ‘X3’ belongs to a power domain ‘V3’ which can turn off and cause a design failure.

The schematic with IS abstraction is as follows:

FIGURE 77. Incremental schematic

Default Severity Label

Warning

Rule Group

`Isolation Logic`

Reports and Related Files

None

LPSVM28

Checks if the net attains the specified value in power down condition

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM28* rule checks whether the specified ports/nets attain the desired value under the power down condition. This rule checks whether the isolation signals have been specified properly so that the design reaches the expected state when the power domains are switched off.

The *LPSVM28* rule checks whether the pin/net specified, using the *power_down* constraint, attains the specified value when the corresponding power domain is switched off.

If you do not specify the `-domain` argument of the *power_down* constraint, the *LPSVM28* rule checks whether the specified pin(s)/net(s) attain the expected value when all power domains are switched off.

The LPSVM28 rule can be run in the CPF/UPF flow also. You need to specify the *power_down* constraint in the SGDC file, but the voltage domain information is picked up from the corresponding CPF/UPF file.

Language

Verilog, VHDL

Parameter(s)

- *lp_set_sim_val_x*: Default value is `1`. Set this parameter to `0` to enable the rule to propagate the value for *set_case_analysis* and other control signals for the logic that lies in the power domain.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the power domains.
- *power_down* (Mandatory): Use to specify pins/nets to be checked and their expected values under the power down condition.
- *set_case_analysis* (Optional): Use to specify the case analysis conditions used by the *LPSVM28* rule using *set_case_analysis* constraint.
- *always_on_cell* (Optional): Use to specify the always-on cells and their domain information.
- *always_on_buffer* (Optional): Specify the names of alwayson buffers using the *always_on_buffer* constraint.

Messages and Suggested Fix

Message 1

The following message appears when the signal <*sig-name*> (Specified using the `-signame` argument of a *power_down* constraint) does not attain the expected value <*value*> (specified using the `-value` argument of the *power_down* constraint) when the power domain <*pd-name*> (specified using the `-domain` argument of a *power_down* constraint) is switched off:

```
[LPSVM28_1][WARNING] Signal '<sig-name>' does not attain the specified value '<value>' in powerdown condition for power domain '<pd-name>'
```

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the signal <*sig-name*> (specified using the `-signame` argument of a *power_down* constraint) does not attain the expected value <*value*> (specified using the `-value` argument of the *power_down* constraint) during power down (when the `-domain` argument is not specified):

```
[LPSVM28_2][WARNING] Signal '<sig-name>' does not attain the specified value '<value>' in powerdown condition
```

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

If this is not fixed, it means that the specified signals have not reached their expected value when the power domain is shutoff and thus proper isolation may not happen.

How to Debug and Fix

The violation is reported at the place where the signal that does not attain the specified value is set or used.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix these violations, ensure that the:

- isolation signals is specified properly for all the power domains.
- all the nets/pins attain the specified/expected value in power down condition.

Example Code and/or Schematic

Example 1

Consider the following example. A violation occurs when the signal `'top.out3'` does not attain the expected value `'0'` when the power domain `'V2'` is switched off, as specified using the *power_down* constraint:

```
Signal 'top.out3' does not attain the specified value '0' in
powerdown condition for power domain 'V2'
```

The schematic generated is as follows:

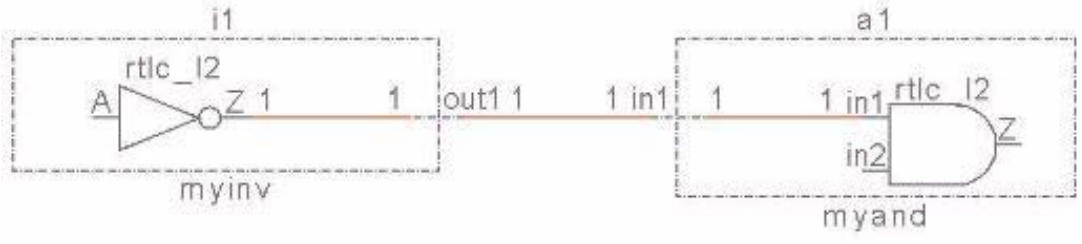


FIGURE 78. Incremental schematic

The schematic highlights the signal for which a violation has been reported and which has not attained the specified value in power down condition.

Example 2

For following SGDC snippet, the *LPSVM28* rule checks whether the pin/net `obct_cs` attains a value `0` when the power domain `V3` is switched off.

```
voltage_domain -name V3 -value 1.2 0

-instname mc_top.u2 -isosig susp_req_i -isoval 1

power_down -domain V3 -signame mc_top.obct_cs -value 0
```

If the signal `mc_top.obct_cs` does not attain a value 0 when domain V2 is shut down then the following violation will come:

```
Signal 'mc_top.obct_cs' does not attain the specified value '0'
in powerdown condition for power domain 'V3'
```

Default Severity Label

Warning

Rule Group

`Isolation Logic`

Reports and Related Files

None

LPSVM31

Reports isolation cells that do not receive a steady state value in the power down condition.

When to Use

NOTE: *This rule will be deprecated for the UPF flow, in a future release. It is recommended to use LPSVM09 rule instead, in the UPF flow.*

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM31* rule reports isolation cell instances that do not receive a steady state value during the power down condition. For the isolation cell to give a steady output in power down condition, the inputs of the isolation cell, except the one connected to power domain output, should get a steady value. These inputs are generally controlled by the isolation signals. Therefore the isolation signals should be able to control the inputs of the isolation cell to steady value(s).

In current scenario, a level shifter with an enable pin, clamp level shifter cell is also treated as an isolation cell.

Prerequisites

The *LPSVM31* rule requires the following:

- Specify power domains using the *voltage_domain* constraint with their isolation signals using the `-isosig` and `-isoval` arguments
- Specify isolation cells using the *isolation_cell* constraint in SGDC or using the *is_isolation_cell* attribute in the library file. If both of these are used to specify isolation cells, all the cells are considered.

Rule exceptions

The *LPSVM31* rule ignores level shifters that also work as isolation cells due to the presence of their enable terminal.

Language

Verilog, VHDL

Parameter(s)

- *lp_disable_lib_attr_read*: Default value is `0`. Set the value to `1` to disable the data to be auto-inferred from the library.
- *lp_set_sim_val_x*: Default value is `1`. Set the value to `0` to enable the *LPSVM31* rule to propagate the value for *set_case_analysis* and isolation control signals also for the logic that lies in power domain. The value for *set_case_analysis* and isolation control signals is not propagated if the combinational logic lies in the power domain.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/power domains in the design and its information is used by voltage and power domain rules.
- *isolation_cell* (Mandatory): Use to specify the isolation cells in power domains.
- *set_case_analysis* (Optional): Use to specify the case analysis conditions.
- *always_on_cell* (Optional): Use to specify the always-on cells and their domain information.
- *always_on_buffer* (Optional): Use to specify the names of always-on buffers.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_isolation_rule* (Mandatory)
- *update_isolation_rules* (Optional)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_isolation_cell* (Optional)
- *define_always_on_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_isolation* (Mandatory)
- *set_isolation_control* (Mandatory)
- *map_isolation_cell* (Optional)

Messages and Suggested Fix

The following message appears when the input *<sig-name>* of instance *<inst-name>* of isolation cell *<isocell-name>* does not receive a steady value under the power down condition:

[LPSVM31_1][WARNING] Input '<sig-name>' of the isolation cell '<instname>' (cell name : '<isocell-name>') will not reach a steady state in the powerdown condition

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

If this violation is not fixed, it means the input of isolation cell is not reaching a steady value in the power down condition, which might lead to incorrect isolation leading to errors in design.

How to Debug and Fix

The violation is reported at the place where the input of isolation cell which does not attain steady value is specified.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, ensure that all input signals of the isolation cells reach a steady state in the power down condition.

Example Code and/or Schematic

Consider the following example where a violation occurs when input signal `'pp'` of instance `'test.iso_mod'` of isolation cell `'iso_mod'` does not receive a steady value under powerdown condition:

```
[WARNING] Input 'pp' of the isolation cell 'test.iso_mod' (cell
name : 'iso_mod') will not reach a steady state in the
powerdown condition
```

The schematic is as follows:

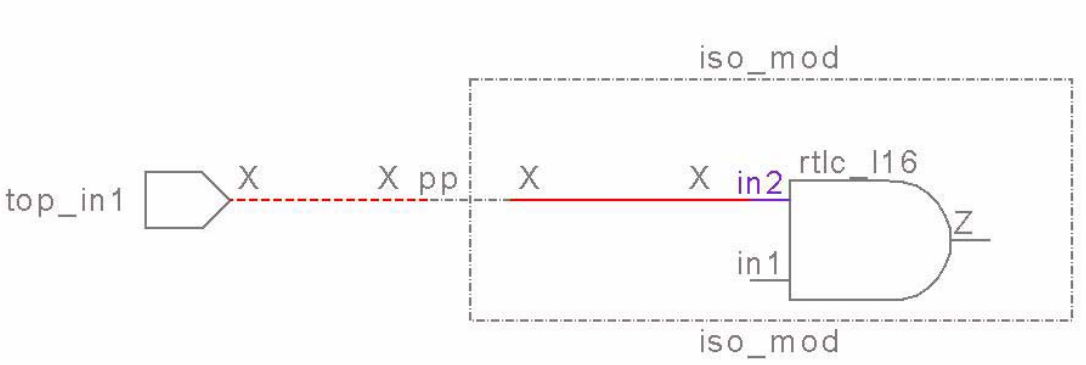


FIGURE 79. Incremental schematic

The schematic shows an isolation cell where the input signal, other than the one connected to power domain output, is not receiving a steady value.

Default Severity Label

Warning

Rule Group

Isolation Logic

Reports and Related Files

None

LPSVM47

Reports power domain inputs that do not attain the specified value in power down condition

When to Use

Use this rule to:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM47* rule checks if the output of an isolation cell connected to the specified power domain input attains the desired value under the power down condition. By default the value should be `0`. A level shifter with an enable pin, such as a clamp level shifter cell, is also treated as isolation cell.

If you do not specify the `-inputs` argument of the *voltage_domain* constraint, this rule checks for all the inputs for value `0`. Values can be specified using the *domain_inputs* constraint.

A cell with *is_isolation_cell* attribute set as `true` in the library is recognized as an isolation cell.

Rule Exceptions

The `LPSVM47` rule does not check:

- Inputs of the power domain if an isolation cell is not connected to the input.
- Crossings with no isolation present. The missing isolation violation is reported using the *LPSVM60* rule.

NOTE: *The power domain crossing and the steady state value at the crossing if an always-on buffer is present at the power domain input crossing.*

Parameter(s)

- *lp_skip_buf*: Default value is 1 and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to 0 to consider SpyGlass-generated buffers during rule checking.
- *lp_skip_aon_buf*: Default value is 1. Set the parameter to 0 to not skip always-on buffers to find an isolation crossing.
- *lp_skip_pwr_gnd*: Default value is 1. Set this parameter to 0 to consider nets connected to power/ground supply.
- *lp_flag_undriven_nets*: Default value is 0. Set this parameter to 1 to consider nets that are not driven.
- *lp_set_sim_val_x*: Default value is 1. Set this parameter to 0 to propagate the value for *set_case_analysis* and other control signals for the logic that is in the power domain.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Used to specify power domains.

NOTE: If the -inisosig and -inisoval arguments of the *voltage_domain* constraint are not specified, the *LPSVM47* rule considers the -isosig and -isoval arguments. However, it reports a warning message to provide the -inisosig and -inisoval arguments.

- *domain_inputs* (Optional): Use to specify the expected values of power domain inputs under power down condition.
- *ignore_crossing* (Optional): Use to specify the power domain to voltage domain crossings and power domain to power domain crossings that should be ignored.
- *power_state* (Optional): Use to specify the power domain relationships and automatically find all the *ignore_crossing* constraints.
- *always_on_buffer* (Optional): Use to specify the names of always-on buffers.
- *set_case_analysis* (Optional): Use to specify the case analysis conditions.
- *always_on_cell* (Optional): Use to specify the always-on cells and their domain information.

CPF Commands

- *create_power_domain* (Mandatory): Use to create a power domain.
- *create_isolation_rule* (Mandatory): Use to create a rule for adding an isolation cell.
- *update_isolation_rules* (Optional): Use to update an isolation rule.
- *create_nominal_condition* (Mandatory): Use to infer voltage values for domains in a power state.
- *create_power_mode* (Mandatory); Use to create a power mode.
- *define_isolation_cell* (Optional): Use to identify the library cells in the .lib files that can be used as isolation cells.
- *define_always_on_cell* (Optional): Use to identify the library cells in the .lib files that can be used as always-on cells.

UPF Commands

- *create_power_domain* (Mandatory): Use to create a power domain.
- *create_supply_port* (Mandatory): Use to create a supply port.
- *add_port_state* (Mandatory): Use to apply a state to a port.
- *create_supply_net* (Mandatory): Use to create a supply net.
- *connect_supply_net* (Mandatory) Use to connect a supply port with a supply net.
- *set_domain_supply_net* (Mandatory): Use to specify a primary power and ground supply nets for a power domain.
- *set_isolation* (Mandatory): Use to create an isolation rule.
- *set_isolation_control* (Mandatory): Use to specify the control signal for an isolation rule.
- *map_isolation_cell* (Optional): Use to map an isolation rule to a library cell or range of library cells.

Messages and Suggested Fix

The following message appears when the input <*name*> of power domain <*pd-name*> attains value <*actual-value*> under power down condition when it was expected to attain a value <*expected-value*>:

[LPSVM47_1][WARNING] Power domain (<pd-name>) input signal

```
'<sig-name>' has value '<actual-value>' instead of user-
specified value '<expected-value>'
```

Potential Issues

This violation can arise because of the following:

- Incorrect or no isolation signal: Refer to the *LPSVM22* rule violation for the corresponding connection.
- Isolation signal traversing through an instance, which is off leading to X on path ahead.
- Isolation cell of incorrect functionality is used. For example, an OR gate is used in place of AND for forcing value 0.
- Isolation cell is powered off with respect to the destination or the source.
- Isolation cell output has multiple drivers.

NOTE: *In the UPF flow, the violation message specified above explicitly states the reason for the violation to occur. If you have specified the set_required_input_isolation for a destination domain, the isolation cell should be ON when the source is ON. However, if the set_required_input_isolation for a destination domain is not specified, the isolation cell should be ON when the destination is ON.*

Consequences of Not Fixing

The power domain output can have a floating or an incorrect value with respect to the specified value under power down condition.

How to Debug and Fix

The violation is reported at the input of power domain. This message supports UPF cross-probing. Therefore, in addition to the design file, the UPF files are displayed in the **File** tab.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, review the potential issues and ensure that each issue is addressed.

Example Code and/or Schematic

Example 1

The following SGDC code snippet defines the V3 power domain and the

expected value (1) for its input obct_cs. The *LPSVM47* rule reports a violation if obct_cs does not attain a value of 1 under power down condition.

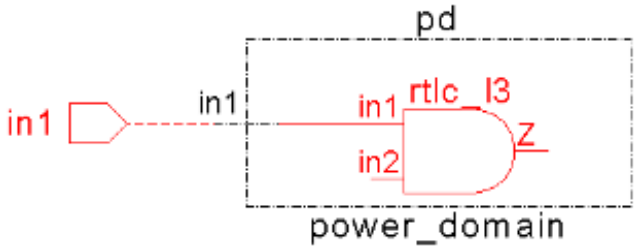
```
voltage_domain -name V3 -value 1.2 0
  -instname mc_top.u2 -isosig susp_req_i -inisoval 1
  -inputs PD_V3_IN
domain_inputs -name PD_V3_IN -value obct_cs 1
```

Example 2

Consider the following example. A violation occurs when the 'top.in1' input signal of the V2 power domain has a value other than the user-specified value:

```
[WARNING] Power domain (V2) input signal 'top.in1' has value
'X' instead of user-specified value '1'
```

The schematic is as follows:



The *LPSVM47* rule highlights the top.in1 input of the V2 power domain because it does not attain user-specified value when the isolation signal is asserted.

Default Severity Label

Warning

Rule Group

Isolation Logic

Reports and Related Files

None

LPSVM48

Reports cells other than isolation cells used at the input-side and output-side of the power domain

When to Use

NOTE: *This rule will be deprecated for the UPF flow. It is recommended to use the LPSVM08A and LPSVM60 rules instead, in the UPF flow.*

The *LPSVM48* rule is a post-synthesis gate-level check. Use this rule to:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM48* rule check ensures that the cell used at the:

- Input-side of the power domain is specified as an input-side isolation cell either by using the *input_isocell* constraint or library cells with the *is_isolation_cell* attribute set.
- Output-side of the power domain is specified as an isolation cell either by using the *isolation_cell* constraint or library cells with the *is_isolation_cell* attribute set.

The *LPSVM48* rule also reports the pins that are connected outside of the power domain without being specified using the `-pin` argument of the *input_isocell* constraint.

A cell with *is_isolation_cell* attribute set as `true` in the library is recognized as an isolation cell.

NOTE: *A level shifter with an enable pin, such as a clamp level shifter cell, is also treated as isolation cell.*

Rule Exceptions

The rule *LPSVM48 does not check:*

- The pins of the input isolation cell connected to outside of the power domain if the -pin argument is not specified.

- The power domain crossing (output side or input side) and the presence of isolation cell (or input isolation cell) if an always-on buffer is present at the power domain crossing.

Parameter(s)

- *lp_isologic_in_pd*: Default value is `1. By default, the` *LPSVM48* rule considers input-side isolation cells both inside and outside the power domain boundary. Set this rule parameter to `0` to consider the input-side isolation cells outside the power domain boundary only.
- *lp_disable_lib_attr_read:* Default value is `0`. By default, the *LPSVM48* rule considers the library cells specified with *is_isolation_cell* attribute as an isolation logic. Set this rule parameter to 1 to ignore library cells with the *is_isolation_cell* attribute and consider only the cell specified with *input_isocell* and *isolation_cell* constraints as input- and output-side isolation logic.

Constraint(s)

- *voltage_domain* (Mandatory): Use to specify power domains.
- *isolation_cell* (Mandatory): Use to specify output-side isolation cells.
- *input_isocell* (Mandatory): Use to specify input-side isolation cells and pins that are allowed to be connected outside of the power domain.
- *power_state* (Optional): Use to specify the power domain relationships and automatically find all the *ignore_crossing* constraints with the *power_state* constraint. The rule has support for *power_state* only for the crossings on the output side of power domain
- *always_on_buffer* (Optional): This constraint is used to specify the names of always-on buffers.

Messages and Suggested Fix

Message 1

The following message appears when the instance *<inst-name>* of cell *<cellname>* present at the output stage (outside the power domain) of power domain *<pdname>*, is not a valid isolation cell:

[LPSVM48_1][WARNING] `Instance <inst-name> of cell <cell-name> used at the output stage of power domain <pd-name> is not a`

```
valid isolation cell
```

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the instance <*inst-name*> of cell <*cellname*> present at the input stage (inside the power domain) of power domain <*pdname*> is not a valid input isolation cell:

```
[LPSVM48_2][WARNING] Instance <inst-name> of cell <cell-name>
used at the input stage of power domain <pd-name> is not a
valid input stage isolation cell
```

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when the instance <*inst-name*> of cell <*cellname*> is wrongly present at the input stage (inside the power domain) of power domain <*pdname*>:

```
[LPSVM48_3][WARNING] Instance <inst-name> of cell <cell-name>
cannot be used at the input stage of power domain <pd-name>
```

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears when the pin of the instance <*instname*> of cell <*cell-name*> is wrongly connected to outside of the power domain <*pd-name*>:

```
[LPSVM48_4][WARNING] Pin <pin-name> of cell <cell-name>(<inst-
name>) cannot be used at input stage of power domain <pd-name>
```

Potential Issues

These violations appear because of any of the following:

- The cell connected to an input of the power domain is not specified by using the *input_isocell* constraint.
- The cell connected to an output of the power domain is not specified by using the *isolation_cell* constraint.
- The pin of input isolation cell connected to outside of the power domain and is not specified by using the -pin argument of the *input_isocell* constraint.

- The cell connected to an input of the power domain is specified by using the *input_isocell* constraint with the -inhibit argument.

Consequences of Not Fixing

The consequences of not fixing the violations are as follows:

Message 1: The instance used at the output of the power domain is not a valid isolation cell. Therefore, it will not be able to serve its purpose of providing isolation and will lead to design failure.

Message 2: The instance used at the input stage (inside the power domain) is not a valid isolation cell. Therefore, it will not be able to serve its purpose of providing isolation and will lead to design failure.

Message 3: The instance is wrongly placed and cannot be used at the input stage of the power domain. Therefore, it will lead to design failure.

Message 4: The pin of the instance cannot be used at the input stage of the power domain. Therefore, it will not be able to serve its purpose of providing isolation.

How to Debug and Fix

These violations are reported at the instance of the isolation cell/pin used at the input/output of power domain that is not specified using the relevant constraint.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. The schematic highlights the potential issue. Refer to the Example Code and/or Schematic section for an example.

To fix these violations, ensure that the

- Cells connected to input/output of power domain should be properly specified.
- Pins of input isolation cell connected to input/output of power domain should be properly specified.

Example Code and/or Schematic

Example 1

In the following example, the power domain (PD), input-side isolation cell (`iso_or`), and output-side isolation cell (`iso_and`) have been defined. The *LPSVM48* rule reports the inputs of `PD` that do not have an instance of `iso_or` as their isolation cell and those outputs of `PD` that do not have an

instance of `iso_and` as their isolation cell.

```
voltage_domain -name PD -value 1.2 0.0
  -instname top.power_domain -isosig iso -isoval 1
input_isocell -names iso_or -belongsto PD
isolation_cell -names iso_and -belongsto PD
```

Similarly, for the following SGDC code snippet, the *LPSVM48* rule reports those inputs/outputs of power domain `PD` that do not have an instance of a cell having library attribute *is_isolation_cell* with value `true`.

```
voltage_domain -name PD -value 1.2 0.0
  -instname top.power_domain -isosig iso -isoval 1
isolation_cell -attribute is_isolation_cell -attributeval
true
input_isocell -attribute is_isolation_cell
     -attributeval true
```

Example 2

Consider the following examples. When the *lp_isologic_in_pd* parameter is set to `0`, the *LPSVM48* rule considers the isolation cells outside the power domain boundary and the input-side isolation cells outside the power domain boundary.

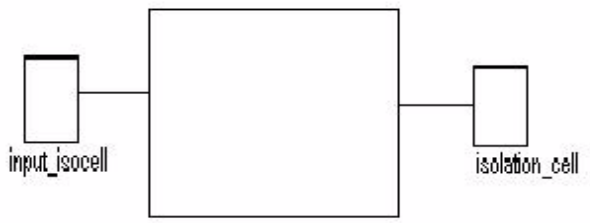


FIGURE 80. Isolation cells

When the *lp_isologic_in_pd* parameter is set to `1`, the *LPSVM48* rule considers the isolation cells both inside and outside the power domain boundary, as shown in the following

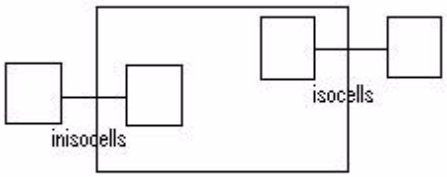


FIGURE 81. Isolation cells

Example 3

Consider the following example. A violation occurs when the instance `'top.mid_inst.Iso1'`of the cell `'HD65_LH_ISO0X10_VDDC'` is present at the input stage (inside the power domain) of power domain `'PD1'`:

```
Instance 'top.mid_inst.Iso1' of cell 'HD65_LH_ISO0X10_VDDC'
used at the input stage of Power domain 'PD1' is not a valid
input stage isolation cell
```

The schematic is as follows:

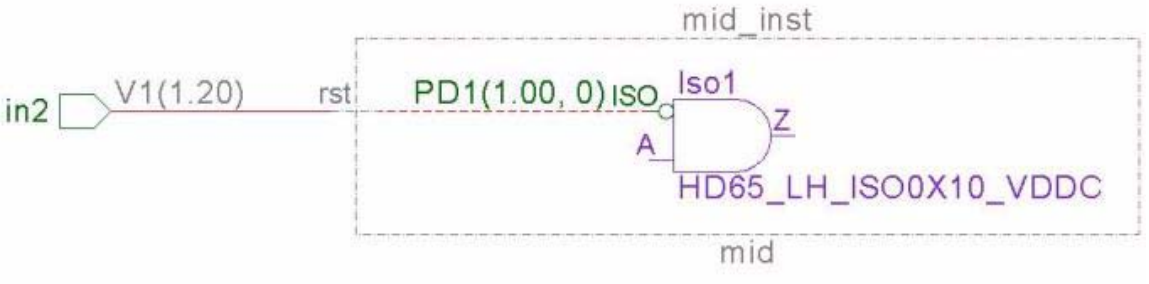


FIGURE 82. Incremental schematic

Here, the isolation cell placed in the input stage (inside the power domain) is not a valid isolation cell. Therefore, it will not be able to provide isolation when needed and might lead to design failure. The LPSVM48 rule reports a violation.

Default Severity Label

Warning

Rule Group

Isolation Logic

Reports and Related Files

None

LPSVM50

Reports incorrect instances at the input stage of a power domain

When to Use

Use this rule to identify the incorrect cell instances at the input stage of a power or voltage domain.

This rule is recommended for use with *Gate-level netlist files and their associated library files*, *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*, and *DEF files and their associated LEF files*.

Description

The *LPSVM50* rule reports the cell instances, specified by using the *non_pd_inputcells* constraint, at the input stage of a power or voltage domain. This rule traverses the entire instance list. In addition, this rule analyzes the instances that are inside any power domain or always-on domain with more than one input from a different domain. Such instances are located at the inputs of domains. If the master name of these instances matches any of the cell names specified in the *<name_list>* of the *non_pd_inputcells* constraint for the corresponding domain, a violation is reported.

This rule ensures that the specified cells should not be present at the input stage of a specified power domain.

A cell with the *is_isolation_cell* attribute, set as true in the library, is recognized as an isolation cell.

NOTE: *A level shifter with an enable pin (clamp level shifter cell) that is tied to a constant value is not treated as an isolation cell.*

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain* (Mandatory): Use this constraint to specify the power domains.
- *non_pd_inputcells* (Mandatory): Use this constraint to specify the cell names.

NOTE: *The LPSVM50 rule is also run in the CPF or UPF flow. Specify the non_pd_inputcells constraint in the SGDC file. However, the voltage domain information is retrieved only from the corresponding CPF or UPF file.*

Messages and Suggested Fix

The following message appears when the instance *<inst-name>* is at the input stage of the domain *<dom-name>*:

[LPSVM50_1][WARNING] Instance '<inst-name>'(cell '<cell-name>') should not be present at the input stage of domain '<dom-name>' with signal going to pin '<pin-name>'

Potential Issues

This violation appears if the instance of a cell is present at the input stage of the domain.

Consequences of Not Fixing

The cells specified in the *non_pd_inputcells* constraint are present at the input stage of the specified domain. Ideally, these cells should not be present at the input stage.

How to Debug and Fix

For a graphical view of the violation, double-click the message and click **Incremental Schematic**. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, ensure that the cells-pin pair specified by using the *non_pd_inputcells* constraint are not present at the input of the power or voltage domain.

Example Code and/or Schematic

Example 1

For the following code, the *LPSVM50* rule reports a violation if the cells

specified in the <*cell-name-list*> are present at the input of the <*pd-name*> domain.

```
non_pd_inputcells -names <cell-name-list> -pd <pd-name>
-pins <pin-list>
```

Example 2

The instance `top.mid_inst.fd1_inst` of the cell `FD1`, specified by using the *non_pd_inputcells* constraint, is present at the input stage of the domain `V2`. Therefore, the *LPSVM50* rule reports a violation.

```
[WARNING] Instance 'top.mid_inst.fd1_inst'(cell 'FD1') should
not be present at the input stage of domain 'V2' with signal going
to pin 'CP'
```

The schematic is as follows:

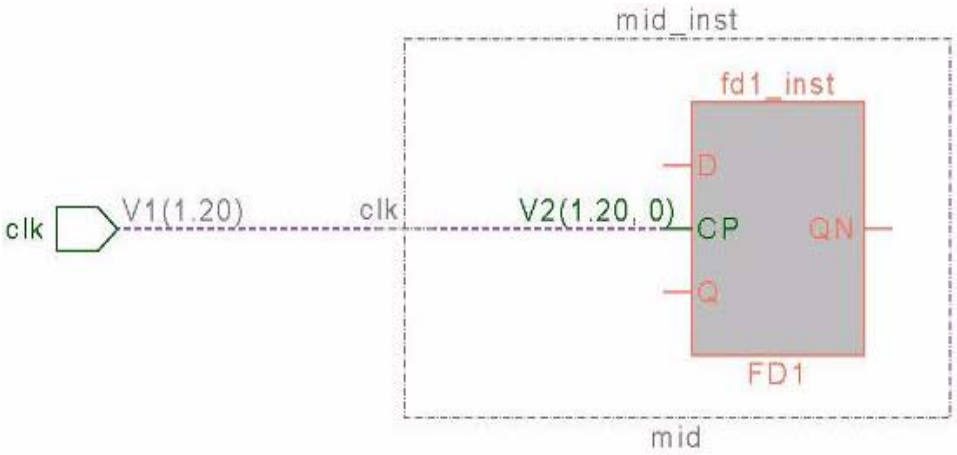


FIGURE 83. Incremental schematic

Default Severity Label

Warning

Rule Group

`Power_Domain_Rules`

Reports and Related Files

None

LPSVM51

Reports isolation cells that are not transparent in power up mode

When to Use

Use this rule to:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM51* rule reports isolation cell instances that are not transparent when the power domain is in power up mode. An isolation cell instance is transparent when the isolation cell input is propagated to the isolation cell output pin without inversion. A level shifter with an enable pin, such as a clamp level shifter cell, is also treated as isolation cell.

Language

Verilog, VHDL

Parameter(s)

- *lp_complex_iso_logic*: Default value is 0. Set the rule parameter to 1 to assume that isolation cells are complex cells and not simple cells comprising of a single gate.
- *lp_disable_lib_attr_read*: Default value is 0. This means that the rule considers library cells specified with *is_isolation_cell* attribute as input-side isolation logic. Set the value to 1 to make the rule ignore library cells with the *is_isolation_cell* attribute and consider only cells specified with the *input_isocell* constraint as output-side isolation logic.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify power domains.
- *isolation_cell* (Optional): Use to specify the output-side of isolation cells.
- *input_isocell* (Optional): Use to specify the input-side of isolation cells.

- *set_case_analysis* (Optional): Use to define the case analysis conditions.

CPF Commands

- *create_power_domain* (Mandatory): Use to create a power domain.
- *create_isolation_rule* (Mandatory): Use to create a rule for adding an isolation cell.
- *update_isolation_rules* (Optional): Use to update an isolation rule.
- *create_nominal_condition* (Mandatory): Use to infer voltage values for domains in a power state.
- *create_power_mode* (Mandatory); Use to create a power mode.
- *define_isolation_cell* (Optional): Use to identify the library cells in the .lib files that can be used as isolation cells.
- *define_always_on_cell* (Optional): Use to identify the library cells in the .lib files that can be used as always-on cells.

UPF Commands

- *create_power_domain* (Mandatory): Use to create a power domain.
- *create_supply_port* (Mandatory): Use to create a supply port.
- *add_port_state* (Mandatory): Use to apply a state to a port.
- *create_supply_net* (Mandatory): Use to create a supply net.
- *connect_supply_net* (Mandatory) Use to connect a supply port with a supply net.
- *set_domain_supply_net* (Mandatory): Use to specify a primary power and ground supply nets for a power domain.
- *set_isolation* (Mandatory): Use to create an isolation rule.
- *set_isolation_control* (Mandatory): Use to specify the control signal for an isolation rule.
- *map_isolation_cell* (Optional): Use to map an isolation rule to a library cell or range of library cells.

Messages and Suggested Fix

The following message appears at the location of isolation cell instance *<instname>* when the net *<in-net-name>* connected to the input pin is of a different polarity than the net *<out-net-name>* connected to the

output pin:

[LPSVM51_1][WARNING] Output '<out-net-name>' of isolation cell '<inst-name>' is of different polarity to input '<in-net-name>' in normal operation

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

In normal condition, the output of an isolation cell should be of the same polarity to the input of that isolation cell. Otherwise, the isolation cell will not remain transparent in the power-up mode. This will lead to design failure.

How to Debug and Fix

The violation is reported at the place where the isolation cell (which is not transparent) is instantiated.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, ensure that the

- Isolation cells on the inputs/outputs of power domains are transparent when the power domain is powered up and the isolation signal is not asserted.
- Isolation cell input is propagated to the isolation cell output pin without inversion in the power up mode.

Example Code and/or Schematic

Consider the following example. A violation occurs when the net `'top.w2'` of isolation cell instance `'INT_OR2'` is connected to the input pin is of a different polarity than the net `'top.out1'` connected to the output pin:

[WARNING] Output 'top.out1' of isolation cell 'INT_OR2' is of different polarity to input 'top.w2' in normal operation

The schematic is as follows:

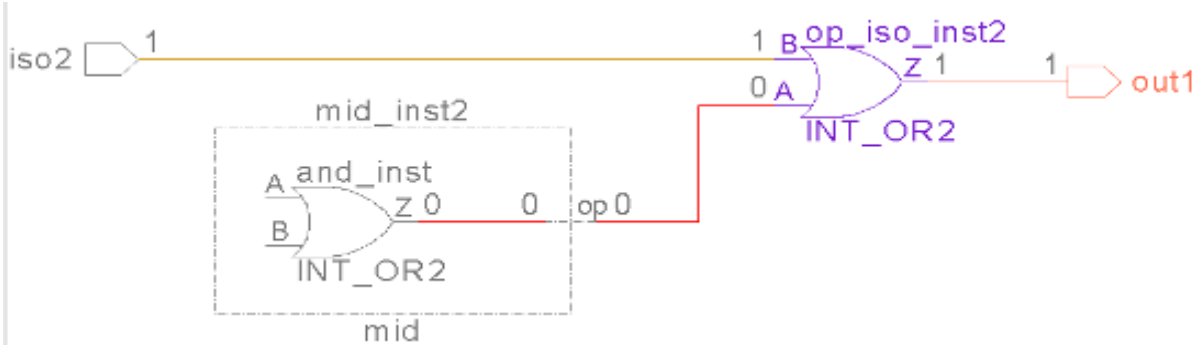


FIGURE 84. Incremental schematic

The schematic highlights the logical values on signals. The voltage domain information and the isolation cell information must be provided with the constraints.

Default Severity Label

Warning

Rule Group

`Isolation Logic`

Reports and Related Files

None

LPSVM52

Reports clock nets going into the power domain are feeding an instance of another domain

When to Use

Use this rule to:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM52* rule reports clock nets that are outputs of a power domain and are triggering flip-flops in another power domain.

Rule exceptions

The `LPSVM52` rule cannot be run in the UPF/CPF power format because information required for this rule is SGDC specific and is not available in power format.

Language

Verilog, VHDL

Parameter(s)

- *lp_check_feedthrough_path*: The default value is `1`. Use to specify whether the LPSVM52 rule reports clock nets that have only always-on buffers, specified using the *always_on_buffer* constraint, in their feed through path.
- *lp_disable_lib_attr_read*: Default value is `0`. This means that the rule considers library cells specified with *is_isolation_cell* attribute as input-side isolation logic. Set the value to 1 to make the rule ignore library cells with the *is_isolation_cell* attribute and consider only cells specified with the input_isocell constraint as output-side isolation logic.

Constraint(s)

- *voltage_domain* (Mandatory): Use to specify the voltage domains and power domains.
- *always_on_buffer* (Optional): Use to specify the output-side of Isolation cells.
- *isolation_cell* (Optional): Use to specify the isolation cells.
- *set_case_analysis* (Optional): Use to specify the case analysis conditions used by the *LPSVM52* rule using the *set_case_analysis* constraint.

Messages and Suggested Fix

The following message appears for clock net <*clk-name*> that is an output of power domain <*pd-name*> that is triggering a flip-flop in voltage/power domain <*vpd-name*>:

[LPSVM52_1][WARNING] Clock net '<clk-name>' is coming out of power domain '<pd-name>' and feeding an instance of another domain '<vpd-name>'

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

Signal from one domain is triggering flip-flops in another domain. The design will fail in case of different voltage states of the two domains.

How to Debug and Fix

The violation is reported at the place where the clock net signal is set or used.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, ensure that the clock nets going into the power domain are not feeding an instance of another domain.

Example Code and/or Schematic

Consider the following example. A violation occurs when clock net 'top.wir3'that is an output of power domain 'V2'triggers a flip-flop in

another voltage/power domain 'V1':

[WARNING] Clock net 'top.wir3' is coming out of power domain 'V2' and feeding an instance of another domain 'V1'

The schematic is as follows:

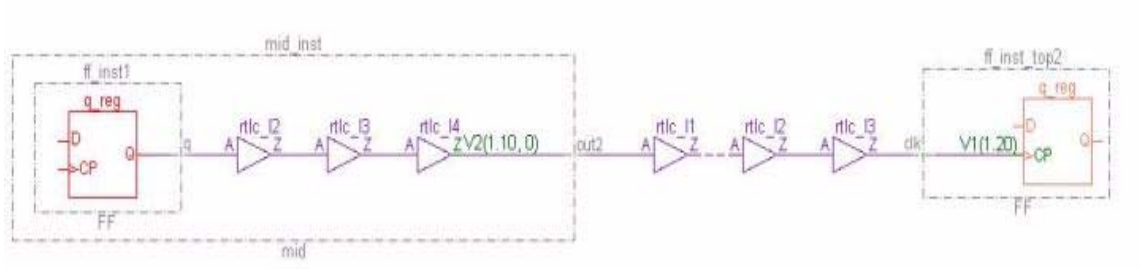


FIGURE 85. Incremental schematic

The schematic highlights the logical values on signals. The voltage domain information and the isolation cell information must be provided with the constraints.

Default Severity Label

Warning

Rule Group

Isolation Logic

Reports and Related Files

None

LPSVM55

Reports isolation cells not present at a specific location in the hierarchy

When to Use

Use this rule to:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The L*PSVM55* rule reports instances of isolation cells that are not at the specified instance hierarchy or any hierarchical levels under that.

The *LPSVM55* rule infers the following cells as isolation cells:

- Cells specified with the *isolation_cell* constraint
- Library cells with the *is_isolation_cell* attribute set
- Cells whose instances are directly driven (ignoring buffers and inverters) by isolation signals (specified with the `-isosig` argument of the *voltage_domain* constraint)
- Level shifters with an enable pin, such as a clamp level shifter cell.

The LPSVM55 rule can also be run in the CPF/UPF flow. In this flow, you need to specify the *switchoff_wrapper_instance* constraint in the SGDC file and then the voltage domain information is retrieved from the corresponding CPF/UPF file.

Language

Verilog, VHDL

Parameter(s)

- *lp_check_iso_hier*: The default value is 0. Set the value to 1 to report isolation cell instances when they are not present exactly at the specified instance hierarchy. This is illustrated in the following image.

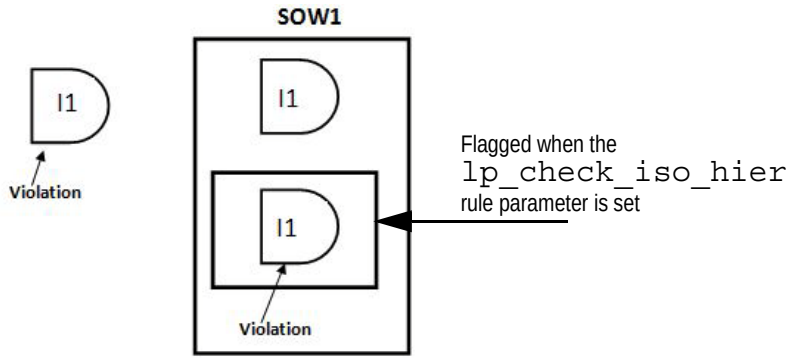


FIGURE 86. Isolation cell instances

- *lp_disable_lib_attr_read*: Default value is 0. This means that the rule considers library cells specified with *is_isolation_cell* attribute as input-side isolation logic. Set the value to 1 to make the rule ignore library cells with the *is_isolation_cell* attribute and consider only cells specified with the input_isocell constraint as output-side isolation logic.

Constraint(s)

SGDC

- *voltage_domain (*Mandatory): Use to specify always-on domains (voltage domains) in the design.
- *switchoff_wrapper_instance* (Mandatory): Use to specify the instance hierarchies for the isolation cells.
- *isolation_cell* (Optional): Use to specify the always-on isolation cells.

Messages and Suggested Fix

Message 1

The following message appears when instance *<inst-name>* of isolation cell *<cell-name>* is found outside the specified instance hierarchy:

[LPSVM55_1][WARNING] Isolation cell instance '<inst-name>'(<cell-name>) is found outside the specified switch off boundary

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when instance <*inst-name*> of isolation cell <*cell-name*> is found inside instance hierarchy <*hier*> but is not direct children of the specified hierarchy when the *lp_check_iso_hier* rule parameter is set:

[LPSVM55_3][WARNING] Isolation cell instance '<inst-name>'(<cell-name>) is not direct children of specified switch off boundary <hier>

Potential Issues

The violation messages explicitly state the potential issues.

For both messages, the cell name is reported for the library cell instances only; it is not reported for the isolation logic written using RTL statements.

Consequences of Not Fixing

The consequences of not fixing the violations are as follows:

Message 1: The isolation cell is found outside the specified instance hierarchy. This is either a design flaw or a fault in the power intent specification.

Message 2: The Isolation cell is found inside instance hierarchy, but is not direct children of the specified hierarchy. This is either a design flaw or a fault in the power intent specification.

How to Debug and Fix

The violations are reported at an instance of isolation cell which is outside the specified instance hierarchy.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, ensure that the isolation cells are present at the specified location in the hierarchy.

Example Code and/or Schematic

Consider the following example. A violation occurs when an instance `'top.PWROFF1.pd2.iso1'`of isolation cell `iso` is outside the specified instance hierarchy:

```
[WARNING] Isolation cell instance 'top.PWROFF1.pd2.iso1'(iso)
is not direct children of specified switch off boundary
top.PWROFF1
```

The schematic is as follows:

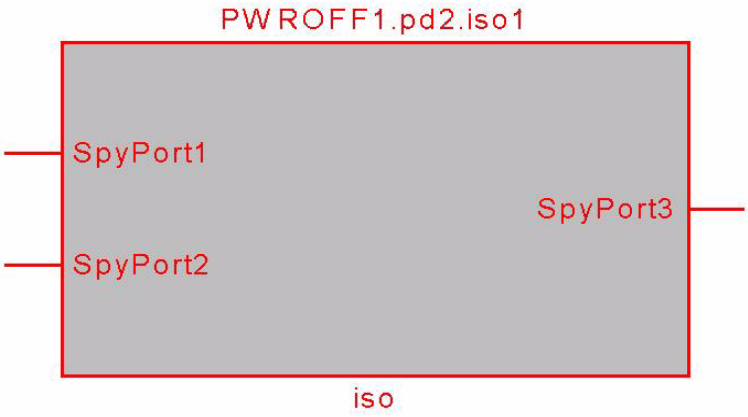


FIGURE 87. Incremental schematic

The schematic highlights instances of the isolation cells that are not in any of the instance hierarchies specified by using the *switchoff_wrapper_instance* constraint.

Default Severity Label

Warning

Rule Group

Isolation Logic

Reports and Related Files

None

LPSVM60

Reports power domain inputs that are not isolated

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM60* rule reports a violation when:

- An isolation cell is missing at the input of power domain.
- The input of the power domain is connected to an input isolation cell and one or more of the following conditions are true:
  - ❒ The input isolation cell is not specified using the *input_isocell* constraint. This case is checked only when you have specified one or more *input_isocell* constraints.
  - ❒ The input isolation cell is not a NAND/NOR/AND/OR gate when the *lp_check_valid_iso* rule parameter is set to `1`.
- For UPF 2.0, the rule also reports if the isolation cell name does not match the `name_prefix/name_suffix` options specified in the `set_isolation` command.

Input-side Isolation Cells

The *LPSVM60* rule infers the following cells as input-side isolation cells:

- Cells specified with the *input_isocell* constraint
- Library cells with the *is_isolation_cell* attribute set. These cells are considered only when the *lp_disable_lib_attr_read* rule parameter is set to 0.
- Cells present at the power domains inputs and whose instances are directly driven by input isolation signals, which are specified using the the `-inisosig` argument of the *voltage_domain* constraint.

- A level shifter with an enable pin, such as a clamp level shifter cell, is also treated as isolation cell.

The isolation signal value under isolation is specified through the `-inisoval` argument of the *voltage_domain* constraint.

Rule Exceptions

The *LPSVM60* rule does not check inputs of the power domain that do not have an input isolation control signal. This signal is specified using the -inisosig argument.

If an always-on buffer is present at the power domain crossing, the *LPSVM60* rule ignores the power domain crossing and the presence of an input isolation cell.

The LPSVM60 rule ignores buffers/inverters that lie between input of power domain and isolation logic, if the domain of buffer is always-on with respect to the power domain because it does not have any impact and the design is electrically correct.

Language

Verilog, VHDL

Parameter(s)

- *lp_disable_lib_attr_read*: Default value is `0`. This means that the rule considers library cells specified with the *is_isolation_cell* attribute as input-side isolation logic. Set the value to 1 to make the rule ignore library cells with the *is_isolation_cell* attribute and consider only cells specified with the *input_isocell* constraint as output-side isolation logic.
- *lp_flag_undriven_nets*: Default value is `0`. Set this parameter to 1 to consider inputs of power domains that are not driven.
- *lp_flag_unconnected_nets*: Default value is `0`. Set this parameter to 1 to consider unconnected inputs of power domains.
- *lp_skip_buf*: Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.
- *lp_skip_aon_buf*: Default value is 1. Set the parameter to 0 to not skip always-on buffers to find an isolation crossing.

- *lp_skip_pwr_gnd*: Default value is 1. Set this parameter to 0 to consider nets connected to power/ground supply.
- *lp_isologic_in_pd*: Default value is 1. By default, the *LPSVM60* rule considers the input isolation cells both inside and outside the power domain boundary. Set this rule parameter to 0 to have the LPSVM60 rule consider the input isolation cells outside the power domain only.
- *lp_check_valid_iso*: Default value is 0. By default, the *LPSVM60* rule does not check for valid isolation cells. Set this rule parameter to check whether the isolation cell is a NAND/NOR/AND/OR gate by functionality.
- *lp_match_location_by_domain*: Default value is `yes` and the isolation cell and level shifter cell locations are matched based on domain. To match these locations based on hierarchy, set the value of this parameter to `no`.
- *lp_allow_check_name_format*: Default value is `no` and this rule does not report a violation if isolation/level shifter instance name does not match with the name prefix/suffix specified in corresponding strategy. Set this parameter to `yes` to report violations for such cases.
- *lp_skip_iso_check_on_ground*: Default value is `no`. Set this parameter to `yes` to not report missing isolation strategy and missing isolation cells on the ground nets.
- *lp_skip_blackbox_checking*: Default value is `0`. Set the parameter to `1` to skip checking for black boxes.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the power domain and their sleep nets.
- *input_isocell* (Optional): Use to specify the isolation cells at inputs of a power domain.
- *power_state* (Optional): Use to specify the combinations of domain states that can exist at the same time during operation of the design
- *set_case_analysis* (Optional): Use to specify the case analysis conditions.
- *ignore_crossing* (Optional): Use to specify the power domain to voltage domain crossings and power domain to power domain crossings that should be ignored.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_isolation_rule* (Mandatory)
- *update_isolation_rules* (Optional)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_isolation_cell* (Optional)
- *define_always_on_cell* (Optional)
- *define_power_clamp_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_isolation* (Mandatory)
- *set_isolation_control* (Mandatory)
- *map_isolation_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when the signal *<sig-name>* going to the power domain *<pd-name>* from the domain <dom-name> is not connected to an input isolation cell:

[LPSVM60_1][WARNING] Signal '<sig-name>' going to destination '<dest-name>' (power domain '<pd-name>') from source '<source-name>' (domain '<dom-name>') must be connected to isolation cell corresponding to strategy <isolation-strategy-name>

In the CPF and SGDC flow, the domain information (<pd-name>,<dom-name>) shows the domain name and its voltage value. However, in the UPF flow, the domain information shows the domain name and the associated supply name and supply value.

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the signal *<sig-name>* going to the power domain *<pd-name>* from the domain <dom-name> is connected to an incorrect isolation cell <iso-name>:

[LPSVM60_3][WARNING] Signal '<sig-name>' going to destination '<dest-name>' (power domain '<pd-name>') from source '<source-name>' (domain '<dom-name>') is connected to incorrect isolation cell '<iso-instance-name>(<iso-cell-name>)' corresponding to strategy <isolation-strategy-name>, Isolation cell is not present in map_isolation_cell command

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when the signal *<sig-name>* going to the power domain *<pd-name>* from the domain <dom-name> is connected to an instance of the correct isolation cell but the cell is not a NAND/NOR/AND/OR cell by functionality provided the *lp_check_valid_iso* parameter is set:

[LPSVM60_5][WARNING] Signal '<sig-name>' going to destination '<dest-name>' (power domain '<pd-name>') from source '<source-name>' (domain '<dom-name>') should be connected to NAND/NOR/AND/OR isolation logic

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears when the isolation cell *<iso-name>* (*<inst-name>*) connected at the input *<input-name>* of the power domain *<pd-name>* is present at an invalid location:

[LPSVM60_6][WARNING] Isolation cell '<inst-name>(<iso-name>)' placed between destination '<destination-name>' (power domain '<domain-name>') and source '<source-name>' (domain '<domain-

name>') is not present in valid location corresponding to strategy '<isolation-strategy-name>'

For debugging information, click *How to Debug and Fix*.

Message 5

The following message appears in the CPF flow, when a location <loc> is specified with the -within_hierarchy option of the *update_isolation_rules* command for the isolation cell <iso-name>(<inst-name>), which is connected to the input <input-name> of power domain <pd-name>, present at an invalid location:

[LPSVM60_8][WARNING] Isolation cell '<iso-name>(<inst-name>)' connected to the input '<input-name>' of power domain '<pd-name>' is not present in valid location <loc>

Message 6

The following message appears when for a signal *<sig-name>* going from source *<src>* to destination *<dest>*, the isolation instance *<iso-inst-name>* does not match the prefix/suffix convention specified in the corresponding strategy. In UPF 2.0, prefix/suffix convention can be applied using the –prefix/-suffix argument of the set_isolation command. In UPF 1.0, the prefix/suffix convention can be applied using the name_format command. Also, in UPF 1.0, the checking of the name_format command is done using the *lp_allow_check_name_format* parameter:

[LPSVM60_9][WARNING] For signal '<sig-name>' going to destination '<dest>' (power domain '<pd-name>') from source '<src>' (domain '<domain-name>'), the isolation instance name ‘<iso-instance-name>(<iso-cell-name>)’ does not match the name <name> in corresponding strategy '<strategy-name>'

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

If this is not fixed, it means that either the isolation cell is missing at the input of a power domain, or an incorrect isolation cell is present for that location. This will lead to improper isolation and lead to design flaws.

Message 6 only shows that the naming prefix/suffix convention has not

been followed and does not cause the design to fail.

How to Debug and Fix

The violation is reported at the input signal not having proper isolation logic of power domain.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example. In addition, Messages 1, 2, and 4 support UPF cross-probing. Therefore, apart from the design file, the UPF files are displayed in the **File** tab.

The *LPSVM60* rule also generates a file named LPSVM60.csv. This file contains all the messages generated by the LPSVM60 rule. To view the spreadsheet, in the Message Tree window, click the Open Spreadsheet option in the right-click menu of the rule.

To fix these violations, insert the proper input-side isolation cell at the input of the power domain.

For **Message 6**, make sure isolation instance name corresponds to the prefix/suffix convention specified in the corresponding strategy of the specified port, in the UPF.

For **Messages 1, 2,** and **3**, double-click the violation message to view the **Supply Net Relationship** widget. This widget displays the relationship between the two selected supplies, and isolation and level shifter requirements.

Example Code and/or Schematic

Example 1

Consider the following example. A violation occurs when the signal 'top.inst2.in2' going from the domain mid to low is not connected to an isolation cell. The following message appears:

```
Signal 'top.inst2.in2' going to destination
‘top.inst2.and_cell.A’ (power domain 'low(1.20 0)') from source
‘top.inst1.and_cell.Z’ (domain 'mid(1.20 0)') must be connected
to isolation cell
```

The schematic is as follows:

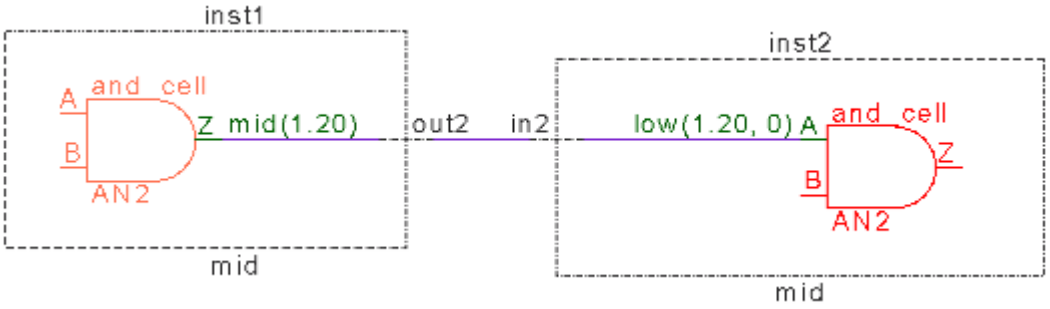


FIGURE 88. Incremental schematic

The schematic highlights the power domains with the missing input isolation logic.

Example 2

Consider the following UPF snippet:

```
upf_version 2.0
# power domain definitions
create_power_domain top


# supply nets
create_supply_port VDD
add_port_state VDD -state {default 1.2}
create_supply_net  VDD -domain top

create_supply_port VDD_LP
add_port_state VDD_LP -state {default 1.2} -state {off_state
off}
create_supply_net  VDD_LP -domain shutOFF

connect_supply_net VDD -ports VDD
connect_supply_net VDD_LP -ports VDD_LP

create_supply_port VSS
```

```
create_supply_net  VSS -domain top
connect_supply_net VSS -ports VSS
add_port_state VSS -state {default 0.0}

set_domain_supply_net top -primary_power_net VDD
primary_ground_net VSS
set_domain_supply_net shutOFF -primary_power_net VDD_LP
primary_ground_net VSS

# isolation rules
set_isolation iso_r1 -domain shutOFF -clamp_value 0
applies_to both
set_isolation_control iso_r1 -domain shutOFF
isolation_signal iso -isolation_sense low -location parent
map_isolation_cell iso_r1 -domain shutOFF -lib_cells AN2

set_isolation iso_r2 -domain shutOFF -clamp_value 0 -elements
{ inst1/in1 } -name_prefix ABC -name_suffix XYZ
set_isolation_control iso_r2 -domain shutOFF
isolation_signal iso -isolation_sense low -location parent
map_isolation_cell iso_r2 -domain shutOFF -lib_cells OR2
```

For the above example, the rule reports the following violation message because for the signal `top.inst1.in1` going to destination `top.inst1.buf_inst1.A` from source `top.in_wir1`, the isolation instance `top.inst1._ABC_iso_XYZ_(AN2)` does not match the prefix/suffix `prefix('ABC')/suffix('XYZ')` specified in the corresponding strategy `iso_r2`:

```
For signal 'top.inst1.in1' going to destination
'top.inst1.buf_inst1.A' (power domain 'shutOFF(supply
VDD_LP:1.200)') from source 'top.in_wir1' (domain 'top(supply
VDD:1.200)'), the isolation instance name
'top.inst1._ABC_iso_XYZ_(AN2)' does not match the name
prefix('ABC')/suffix('XYZ') in corresponding strategy 'iso_r2'
```

The incremental schematic is displayed as shown in figure below.

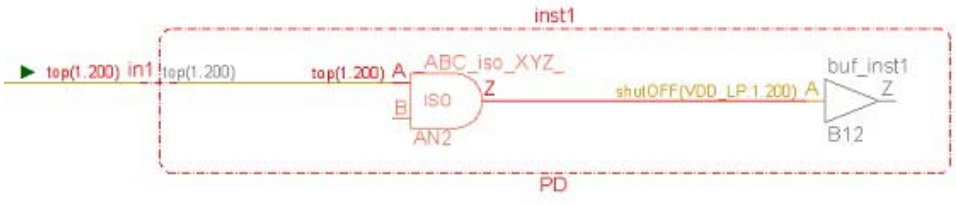


FIGURE 89. Incremental schematic

Default Severity Label

Warning

Rule Group

`Isolation Logic`

Reports and Related Files

- *The LP-report Report:* Generated in SpyGlass standard format (LP-report.rpt), when you specify when you specify the following command in the project file:
  set_option report LP-report

LPLIB_check01

Reports isolation cells that have ambiguous or undefined enable pins

When to Use

This rule is applicable to all design phases.

Description

The *LPLIB_check01* rule reports isolation cells present in the design that have ambiguous or undefined enable pins.

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

Message 1

The following message appears when an isolation cell *<iso-cell-name>* used in the design does not have an enable pin specified:

[LPLIB_check01_1][FATAL] Isolation cell <iso-cell-name> used in design doesn't have an enable pin specified

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when an isolation cell *<iso-cell-name>* used in the design does not have an enable pin specified by using the *isolation_cell_enable_pin* attribute:

[LPLIB_check01_2][FATAL] Isolation cell <iso-cell-name> used in design doesn't have an enable pin specified using 'isolation_cell_enable_pin' attribute in the library

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when an isolation cell *<iso-cell-name>* has a pin <pin-name> specified as enable, but does not have the *isolation_cell_enable_pin* attribute specified in the library:

[LPLIB_check01_3][WARNING] Isolation cell <iso-cell-name> used in design doesn't have an enable pin specified using 'isolation_cell_enable_pin' attribute in the library

Potential Issues

The violation messages explicitly state the potential issues.

For both messages, the cell name is reported for the library cell instances only; it is not reported for the isolation logic written using RTL statements.

Consequences of Not Fixing

The consequences of not fixing the violations are as follows:

Message 1: There is no enable pin specified for the isolation cell. This is a flaw in the power intent.

Message 2: A library cell has been used as an isolation cell in the design, but the *isolation_cell_enable_pin* attribute is missing in the library.

Message 3: A library cell has been used as an isolation cell in the design and the enable pin has been specified in the power intent, but the *isolation_cell_enable_pin* attribute is missing in the library.

How to Debug and Fix

The isolation cell stated in the violation messages is highlighted in the Atrenta Console GUI.

To fix these violations, ensure that the isolation cell has an enable pin that is specified using the *isolation_cell_enable_pin* attribute in the library.

Example Code and/or Schematic

Example 1

In this example, the library cell is used as an isolation cell in the design:

Design

```
ISO_AND IsoCell(input,Enable,output);
```

Library

```
cell (ISO_AND_EN) {
      is_isolation_cell : true;
      pin (A) { direction : input; }
      pin (B) { direction : input; }
      pin (Y) { direction : output; function : "A&B"; }
   }
```

In this case, the *isolation_cell_enable_pin* attribute is missing in the library and therefore the *LPLIB_check01* rule reports a violation. The correct library specification is:

```
cell (ISO_AND_EN) {
      is_isolation_cell : true;
      pin (A) { direction : input; }
      pin (B) { direction : input; isolation_cell_enable_pin
: true; }
      pin (Y) { direction : output; function : "A&B"; }
   }
```

Example 2

In this example, the module used as an isolation cell in the design (CPF/ SGDC power format)

Design

```
module isoAnd(in,en,out);
input in;
input en;
output out;
assign out = in & (~en);
endmodule
```

SGDC

```
isocell -names isoAnd -iso_enable_val 1
```

In this case, the enable pin has not been specified and therefore the

LPLIB_check01 rule reports a violation. The correct SGDC specification is:

```
isocell -names  isoAnd -iso_enable_val 1 -enable_pin en
```

Default Severity Label

Fatal/Warning

Rule Group

Isolation Logic

Reports and Related Files

None

LPLIB_check02

Checks for missing related_power_pin/related_ground_net attribute in library for signal pin

When to Use

This rule is to be used to identify design cells with signal pins that have the *related_power_pin*/*related_ground_pin* attribute specified in the library.

Description

The *LPLIB_check02* rule reports a violation message if the *related_power_pin* and *related_ground_pin* attribute is not specified in the library description of any signal pin.

This rule does not report violations for the following cells:

- If a cell contains only one power pin of type `primary_power`. In this case the power pin is used by default for all the signal pins as their related power pin.
- If a cell contains only one power pin of type `primary_ground`. In this case the ground pin is used by default for all the signal pins as their related ground pin.

If any of the `antenna_diode_related_power_pins`, `antenna_diode_related_ground_pins`, or `antenna_diode_type` attributes is defined for a signal pin, then this rule reports the missing `antenna_diode_related_power_pins`/`antenna_diode_related_ground_pins` attributes and does not report the `related_power_pin`/`related_ground_pin` attributes.

Rule Exception

This rule ignores library cells that are not instantiated in the design.

Parameter(s)

lp_skip_related_pgpin_checking_for_analog_pin: Default value is `yes`. Set this parameter as `no` to perform checking for analog pins.

Constraint(s)

None

Messages and Suggested Fix

The following message appears when the *related_power_pin* or *related_ground_pin* attribute is missing for the signal pin <*sig-pin*> or `antenna_diode_related_power_pins` or `antenna_diode_related_ground_pins` attribute is missing for a signal pin <*sig-pin*>:

[LPLIB_check02_1][ERROR] `Attribute '<related_power_pin| related_ground_pin | antenna_diode_related_power_pins | antenna_diode_related_ground_pins >' is missing for signal pin '<sig-pin>' of cell '<cell-name>'`

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

SpyGlass requires you to resolve this violation. If you do not resolve this violation, the result will not be as desired.

How to Debug and Fix

The signal pin and cell stated in the violation message are highlighted in the Atrenta Console GUI. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, specify the missing attribute for the signal pin as stated in the violation message.

Example Code and/or Schematic

Example 1

This example illustrates when the *LPLIB_check02* rule reports a violation message.

Design

```
MYPADPRIMARYMULTIPLE Pad1(.A(w1), .E(in2), .Y(w3));
```

Library

```
cell (MYPADPRIMARYMULTIPLE) {
  pg_pin (VDDI) { pg_type : primary_power; }
  pg_pin (VDDO) { pg_type : primary_power; }
```

```
    pg_pin (VSSI) { pg_type : primary_ground; }
    pg_pin (VSSO) { pg_type : primary_ground; }
    pin (A) { direction : input; }
    pin (E) { direction : input; }
    pin (Y) { direction : output; function : "A&E";}
 }
```

In this case the cell "MYPADPRIMARYMULTIPLE" is being used in the design. However, the signal pins A,E and Y do not have their *related_power_pin*/ *related_ground_pin* attributes specified in the library. Therefore, the *LPLIB_check02* rule reports a violation message.

The correct library specification is:

```
cell (MYPADPRIMARYMULTIPLE) {
  pg_pin (VDDI) { pg_type : primary_power; }
  pg_pin (VDDO) { pg_type : primary_power; }
  pg_pin (VSSI) { pg_type : primary_ground; }
  pg_pin (VSSO) { pg_type : primary_ground; }
  pin (A) { direction : input; related_power_pin : VDDI,
related_ground_pin : VSSI ;}
  pin (E) { direction : input; related_power_pin : VDDI,
related_ground_pin : VSSI ;}
  pin (Y) { direction : output; function : "A&E";
related_power_pin : VDDO, related_ground_pin : VSSO; }
}
```

Example 2

This example illustrates when the *LPLIB_check02* rule reports a violation message.

Design

```
SINGLE_INPUT_DIODE_CELL_1 ant1(.INP(in1));
```

Library

```
cell (SINGLE_INPUT_DIODE_CELL_1){
```

```
pg_pin (VDD) {
              voltage_name : "VDD";
              pg_type : "primary_power";
                }
pg_pin (VDD1) {
              voltage_name : "VDD1";
              pg_type : "primary_power";
              }
pg_pin (VSS) {
              voltage_name : "VSS";
              pg_type : "primary_ground";
              }
pin (INP) {
              antenna_diode_type : "power";
              atenna_diode_related_ground_pins : "VSS";
              direction : "input";
                                }
}
```

For the SINGLE_INPUT_DIODE_CELL_1 cell, SpyGlass reports the following violation massage because the pin `INP` is with antenna attributes and the `antenna_diode_related_power_pins` attribute is missing for the pin `INP`:

```
Attribute 'antenna_diode_related_power_pins' is missing for
signal pin 'INP' of cell 'SINGLE_INPUT_DIODE_CELL_1'
```

Default Severity Label

Error

Rule Group

`Isolation Logic`

Reports and Related Files

None

LPLIB_check03

Checks the isolation_enable_condition attribute in library for a signal pin

When to Use

This rule is to be used to report design cells with signal pins that have the `isolation_enable_condition` attribute specified in the library.

Description

The *LPLIB_check03* rule reports violation messages related to the `isolation_enable_condition` attribute in the following scenarios:

- When the `is_isolated` attribute is missing for a pin, for which the `isolation_enable_condition` attribute is present.
- When the `is_isolated` attribute value defined for a pin is not true on which the `isolation_enable_condition` attribute is present.
- When the boolean expression in the `isolation_enable_condition` attribute (defined on a pin group) includes the output pin.

NOTE: *This attribute is not supported for a bus group.*

Rule Exception

This rule ignores library cells that are not instantiated in the design.

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

Message 1

The following message appears when the `is_isolated` attribute is missing for pin *<pin-name>*, for which the `isolation_enable_condition` attribute is present:

[LPLIB_check03_1][FATAL] Attribute is_isolated missing for pin ‘<pin-name>’ (‘<cell-name>’) on which isolation_enable_condition attribute is defined

Message 2

The following message appears when the value of the `is_isolated` attribute is defined for pin *<pin-name>*, for which the `isolation_enable_condition` attribute is present but attribute value *<attribute-value>* is not true:

[LPLIB_check03_2][FATAL] Wrong value ‘<attribute-value>’ specified for is_isolated attribute for pin ‘<pin-name>’ (‘<cell-name>’) on which isolation_enable_condition attribute is defined

Message 3

The following message appears when the `isolation_enable_condition` attribute is present but the boolean expression also includes output pins *<output-pin-name>*:

[LPLIB_check03_3][FATAL] Boolean expression used in isolation_enable_condition attribute defined for pin ‘<pin-name>’ (‘<cell-name>’) includes output pin ‘<output-pin-name>’

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

SpyGlass requires you to resolve this violation.

How to Debug and Fix

The signal pin and cell stated in the violation message are highlighted in the Atrenta Console GUI. Refer to the Example Code and/or Schematic section for an example.

To fix these violations:

Message 1: Make sure that the `is_isolated` attribute is present for the signal pin mentioned in the violation message

Message 2: Make sure that the value of the `is_isolated` attribute is true.

Message 3: Make sure that the boolean expression does not contain any

output pins.

Example Code and/or Schematic

Example 1

Consider the following example:

```
cell (macro)
  {
    pin(d) {
      <is_isolated attribute missing>
      isolation_enable_condition: boolean_expression;
  }
}
```

In the above example, the *LPLIB_check03* rule reports the following violation message because the `is_isolated` attribute is missing for pin `d`, for which the `isolation_enable_condition` attribute is present.

```
Attribute is_isolated missing for pin ‘d’ (‘macro’) on which
isolation_enable_condition attribute is defined
```

Example 2

Consider the following example:

```
cell (macro)
  {
    pin(d) {
      is_isolated : false
      isolation_enable_condition: boolean_expression;
  }
}
```

In the above example, the *LPLIB_check03* rule reports the following violation message because the `is_isolated` attribute is defined for pin `d`, for which the `isolation_enable_condition` attribute is present but attribute value is not true.

```
Wrong value ‘false’ specified for is_isolated attribute for pin
‘d’ (‘macro’) on which isolation_enable_condition attribute is
defined
```

Example 3

Consider the following example:

```
cell(macro)
  {
    pin(out) {
      direction: output;
    }
    pin(en) {
     direction: input;
    }
   pin(d) {
     is_isolated : true;
     isolation_enable_condition: “en * out”;
   }
}
```

In the above example, the *LPLIB_check03* rule reports the following violation message because the `is_isolated` attribute is defined for pin `d` but the boolean expression also includes output pins.

```
Boolean expression used in isolation_enable_condition attribute
defined for pin ‘d’ (‘macro’) includes output pin ‘out’
```

Default Severity Label

Fatal

Rule Group

`Isolation Logic`

Reports and Related Files

None

Always-on Logic Rules

The Always-on Logic rules are as follows:

Rule	Reports
LPAON01	Constant value reaching the always-on buffer
LPAON02	Reports issues with always-on paths specified using the always_on_path constraint
LPPLIB11	Signals that are not connected or incorrectly connected to always-on buffers
LPSVM53	User-specified pins that are not connected to always-on domains
LPSVM54	Instances of user-specified always-on cells that are not in specified domains only
LPSVM40	Signals that are not connected or incorrectly connected to always-on buffers

LPAON01

Reports constant values reaching the input pin of always-on buffer

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPAON01* rule reports the instances of the always-on buffer, where a constant value is reaching the input of always-on buffer. The rule skips buffers and inverters in the input path of the always-on buffers, and highlights the source of constant supply.

Prerequisites

By default, the *LPAON01* rule will not run. To enable this rule, select the *LPAON01* rule in the Atrenta Console GUI.

Rule Exceptions

The *LPAON01* rule ignores always-on buffers that are driven by TIE cells.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/power domains.
- *always_on_pin* (Optional): Use to specify always-on cell pins.
- *always_on_buffer* (Mandatory): Use to specify the always-on buffers.
- *set_case_analysis* (Optional): Use to specify the case analysis conditions.

Messages and Suggested Fix

The following message appears when the always-on buffer *<aoncell-name>* instantiated as *<inst-name>* reaches the constant value of *<value>*:

[LPAON01_1][WARNING] Constant value '<value>' is reaching the always on buffer '<inst-name>' (<aon-cell-name>)

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

Since a constant value is reaching the AON buffer, you can directly connect the constant value to the destination buffer/cell. Therefore, the AON buffers used are redundant.

How to Debug and Fix

The violation is reported at the cell instance of such an AON buffer, which reaches a constant value.

For a graphical view, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for a sample.

To fix the violation, ensure that constant value does not reach the input pin of always-on buffers.

Example Code and/or Schematic

In the following example, a violation message appears when the instance `top.AON02` of always-on buffer `AON_BUF` attains `0` constant value:

```
Constant value '0' is reaching the always on buffer 'top.AON02' (AON_BUF)
```

The schematic is as follows:

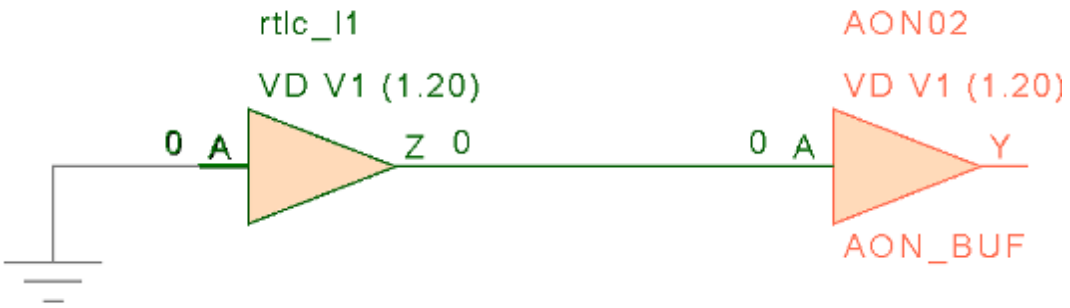


FIGURE 90. Incremental schematic

Default Severity Label

Warning

Rule Group

Always-on Logic

Reports and Related Files

None

LPAON02

Reports issues with always-on paths specified using the always_on_path constraint

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPAON02* rule reports a violation if a logic driven by switchable supply is found along a path specified using the *always_on_path* SGDC constraint.

Prerequisites

By default, the *LPAON02* rule will not run. To enable this rule, select the *LPAON02* rule in the Atrenta Console GUI.

Rule Exceptions

The *LPAON02* rule ignores single input cells, like buffers or inverters.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *always_on_path* (Mandatory): Use to specify an always-on path.

Messages and Suggested Fix

Message 1

The following message is reported if the path specified in the

always_on_path constraint does not exist in the design:

```
[LPAON02_1][WARNING] Path specified between -from and -to
fields in always_on_path constraint does not exist in design
```

Message 2

The following message is reported if the path specified in the `always_on_path` constraint contains elements that are not always-on:

```
[LPAON02_2][WARNING] Element '<element-hier-name>'('<domain-
name>(<supply-name>)') present in the path specified using
always_on_path constraint (file '<verilog-file-name>', line
'<line-number>') is not always on logic
```

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The always on path specified by the `always_on_path` SGDC constraint will not be always-on.

How to Debug and Fix

Message 1: The path specified in the `always_on_path` SGDC constraint should be corrected since such a path does not exist in the design.

Message 2: An always-on supply should be connected to the violating element.

Example Code and/or Schematic

Example 1

Consider the following SGDC code snippet:

```
always_on_path -from top.in1 -to top.inst1.out1
```

The above command leads to report the following violation message because no path exists from source `top.in1` to the destination `top.inst1.out1`:

```
Path specified between -from and -to fields in always_on_path
constraint does not exist in design
```

Example 2

In this example, the following violation message appears when the domain PD1 is a switching domain and element `top.inst1.and1.B` is present in an always-on path specified using the *always_on_path* constraint:

```
Element 'top.inst1.and1.B'('PD1(supply VDD1:1.100)') present in
the path specified using always_on_path constraint (file
'test.v', line '19') is not always on logic
```

The schematic is as follows:

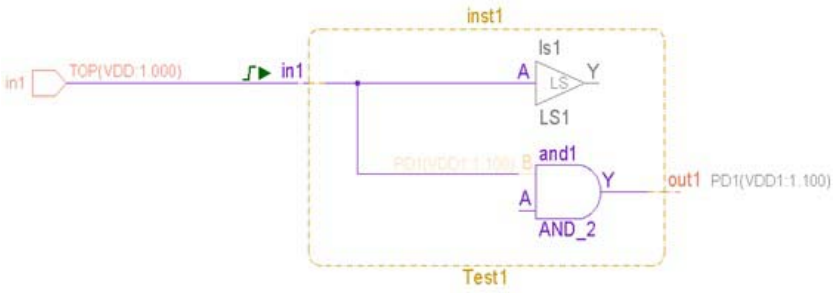


FIGURE 91. Incremental schematic

Default Severity Label

Warning

Rule Group

`Always-on Logic`

Reports and Related Files

None

LPPLIB11

Reports signals that are not connected or incorrectly connected to always-on buffers

When to Use

Use this rule for:

- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPPLIB11* rule reports signals that are not connected or incorrectly connected to always-on buffers.

The *LPPLIB11* rule considers only those buffers specified with the *always_on_buffer* constraint to be always-on buffers when the supply connected to their VDDC pin is an always-on supply rail.

The *LPPLIB11* rule reports violation messages in the following cases:

- Only the signal name is specified using the `-names` argument of the *aon_buffered_signals* constraint

  The *LPPLIB11* rule flags all instances in the fan-out of the signal.

- The signal name is specified using the `-names` argument and the cell-pin pair name list is specified using the `-terminatingcells` argument

  The *LPPLIB11* rule reports all instances in the fan-out of the signal till an instance of a user-specified terminating cell or a primary output port.

- The signal name is specified using the `-names` argument, the cell-pin pair name list is specified using the `-terminatingcells` argument, and cell name list of cells to be ignored using the `-ignorecells` argument

  The *LPPLIB11* rule reports all instances in the fan-out of the signal till an instance of a user-specified terminating cell or a primary output port skipping instances of cells specified using the `-ignorecells` argument.

- The signal name is specified using the `-names` argument and cell name list of cells to be ignored using the `-ignorecells` argument

  The *LPPLIB11* rule reports all instances in the fan-out of the signal skipping instances of cells specified using the `-ignorecells` argument.

NOTE: *For a library cell acting as buffer/inverter and specified with the always_on attribute set as* `true` *or the related_power_pin/related_ground_pin set as* `back_supply`*, is treated as an always_on_buffer.*

Parameter(s)

- *lp_sig_viol_count*: By default, the `LPPLIB11` rule flags only the first 50 rule messages for each signal.
  Use the `lp_sig_viol_count` rule parameter to set a different limit.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): This constraint is used to specify the name of the always-on domains in the design.
- *always_on_buffer* (Mandatory): This constraint is used to specify the name of the always-on buffers in the design.
- *aon_buffered_signals* (Mandatory): This constraint is used to specify the names of the signals that should be driven by an always-on buffer using the `-names` argument.
  Optionally, you can also specify the following arguments:
  - Terminating cell-pin pair name list using the `-terminatingcells` argument
  - Name list of cells to be ignored while traversing the fan-out of the specified signal using the `-ignorecells` argument.

NOTE: *The LPPLIB11 rule can be run in the CPF/UPF flow also. You need to specify the aon_buffered_signals constraint in the SGDC file but the voltage domain/supply information is picked up from the corresponding CPF/UPF file.*

- *supply* (Optional): This constraint is used to specify the name of the supply rail port.

- *pg_cell* (Optional): This constraint is used to specify the Power/Ground pins to provide details of Power/Ground pins for the cells that are not in the specified PLIB/LEF files.

Messages and Suggested Fix

Message 1

The following message appears at the location where signal <*sig-name*> specified with the aon_buffered_signals constraint is connected to instance <*inst-name*> that is not an always-on buffer:

[LPPLIB11_1][WARNING] Signal '<sig-name>' is connected to instance '<inst-name>' which is not an always-on buffer

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears at the location of instance <*inst-name*> of always-on buffer <*du-name*> when the instance is not connected to any of the signals specified with the aon_buffered_signals constraint:

[LPPLIB11_2][WARNING] Always-on buffer '<du-name>' instantiated as '<inst-name>' is not connected to any of the user specified signals

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears at the location where signal <*sig-name*> specified with the aon_buffered_signals constraint is connected to instance <*inst-name*> of terminating cell <*termcell-name*> at pin <*pin1-name*> instead of expected pin <*pin2-name*>:

[LPPLIB11_3][WARNING] Signal '<sig-name>' is connected to the terminate cell '<termcell-name>' instantiated as '<inst-name>' through the pin '<pin1-name>' and not '<pin2-name>'

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears at the location where signal <*sig-name*>

specified with the aon_buffered_signals constraint is directly connected to instance <*inst-name*> of terminating cell <*termcell-name*> at the expected pin:

[LPPLIB11_4][WARNING] Signal '<sig-name>' is directly connected to the terminate cell '<cell-name>' instantiated as '<inst-name>'

For debugging information, click *How to Debug and Fix*.

Message 5

The following message appears at the location where signal <*sig-name*> specified with the aon_buffered_signals constraint is directly connected to instance <*inst-name*> of terminating cell <*termcell-name*> at pin <*pin1-name*> instead of expected pin <*pin2-name*>:

[LPPLIB11_5][WARNING] Signal '<sig-name>' is directly connected to the terminate cell '<cell-name>' instantiated as '<inst-name>' through the pin '<pin1-name>' and not '<pin2-name>'

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The consequences of not fixing are:

- **Message 1**: A signal specified using *aon_buffered_signals* is not connected to any always-on cell. This makes the constraint of no use.
- **Message 2**: An always-on buffer is not connected to any signal specified using *aon_buffered_signals*. This is a design flaw.
- **Message 3**: A signal specified using *aon_buffered_signals* is not connected to the expected pin. This is a design flaw.
- **Message 4**: A signal specified using *aon_buffered_signals* is directly connected to the expected pin of the terminating cell. This is a design flaw.
- **Message 5**: A signal specified using *aon_buffered_signals* is directly connected to a pin of the terminating cell other than the expected pin. This is a design flaw.

How to Debug and Fix

The violations are reported at the cell instance to which an aon signal is incorrectly connected.

For a graphical view, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for a sample.

To fix the violation, ensure that all the signals specified with the *aon_buffered_signals* constraint are connected to always-on buffers only.

Example Code and/or Schematic

Consider the following example.

A violation appears when the signal `'top.in1'` gets connected to a cell instantiated as `'top.buf_inst1'`, which is not a always-on buffer:

```
Signal 'top.in1' is connected to instance 'top.buf_inst1' which is not an always-on buffer
```

The schematic is displayed as follows:

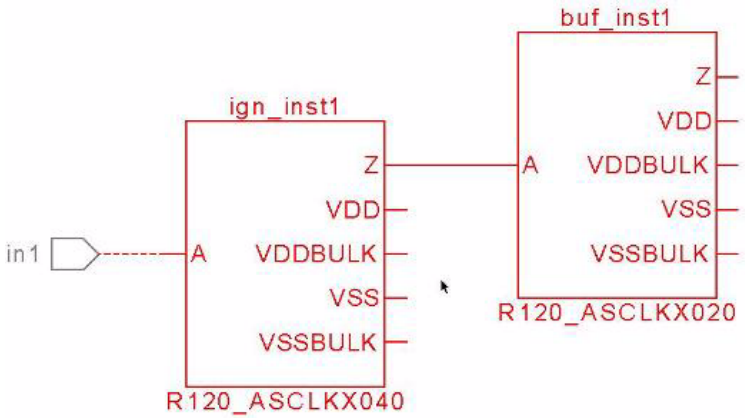


FIGURE 92. Incremental schematic

Default Severity Label

Warning

Rule Group

Always-on Logic

Reports and Related Files

None

LPSVM53

Reports the always-on cell pins that are not driven by an always-on domain

When to Use

Use this rule to check the connections of the control pins of the low power cells.

This rule is recommended for use with *Gate-level netlist files and their associated library files*, *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*, and *DEF files and their associated LEF files*.

Description

The *LPSVM53* rule reports the user-specified cell pins that are not driven by an always-on domain or a relatively-on domain. The control pins of the low power cells, such as retention cells, power switch cells, and input pins, of the always-on buffers should be driven by an always-on domain.

This rule reports a violation when the pins, specified by using the `-pin` argument of the *always_on_pin* constraint, in each instance of the cell, specified by using the `-cell` argument of the *always_on_pin* constraint, are not connected to an always-on domain. Such pins ignore the always-on buffers specified by using the *always_on_pin* constraint in the path.

For a library cell, if a signal pin is specified with the *always_on* attribute set as `true` or the *related_power_pin* or *related_ground_pin* attribute set as `back_supply`, the pin is treated as an always-on pin of the cell.

You can specify the cell pins connected to the always-on domains with the *always_on_pin* constraint.

Prerequisites

The *LPSVM53* rule requires you to:

- specify the always-on voltage domains with the *voltage_domain* constraint.

Rule Exception(s)

The *LPSVM53* rule ignores the always-on pins that are driven by the TIE cells.

Language

Verilog, VHDL

Parameter(s)

- *lp_skip_buf*: Default value is `1`. Set this parameter to `0` to ignore the buffers generated during the synthesis. Other possible values are `no` and `yes`.
- *lp_check_enable_pin*: Default value is `0`. Set this parameter to `1` to check the enable pin of the isolation cells or clamp level shifters.
- *lp_relative_aon_checking*: Default value is `0`. Set the value `1` to enable checking for relative always-on domain of source.
- *lp_check_relative_aon_buffers*: Default value is `0`. Set the value `1` to enable checking for relative always-on buffers.

Constraint(s)

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage or power domains in the design.
- *always_on_pin* (Optional): Use this constraint to specify the always-on cell pins.
- *always_on_buffer* (Optional): Use this constraint to specify the always-on buffers.
- *set_case_analysis* (Optional): Use this constraint to specify the case analysis conditions.

Messages and Suggested Fix

Message 1

The following message appears when the always-on pin *<pin-name>* of the instance *<inst-name>* of the cell *<cell-name>* is not connected to an always-on domain:

[LPSVM53_1][WARNING] Always-on pin ‘<pin-name>’ of instance ‘<inst-name>’ (<cell-name>) is not coming from always-on block

Message 2

The following message appears when the always-on pin *<pin-name>* of

the instance <*inst-name*> of the cell <*cell-name*> is not connected to an always-on domain and incorrect source is port:

[LPSVM53_2][WARNING] Always-on pin ‘<pin-name>’ of instance ‘<inst-name>’(<cell-name>) is not coming from always-on block, incorrect source <port-name>

Message 3

The following message appears when the always-on pin <*pin-name*> of the instance <*inst-name*> of the cell <*cell-name*> is not connected to an always-on domain and incorrect source is a terminal:

[LPSVM53_3][WARNING] Always-on pin ‘<pin-name>’ of instance ‘<inst-name>’(<cell-name>) is not coming from always-on block, incorrect source <terminal-name>

Message 4

The following message appears when the pin <*pin-name*> of the instance <*inst-name*> of the cell <*cell-name*> is not coming from equally-on/more-on block:

[LPSVM53_4][WARNING] Pin '<pin-name>' of instance '<inst-name>'(<cell-name>) is not coming from equally-on/more-on block

Message 5

The following message appears when the pin <*pin-name*> of the instance <*inst-name*> of the cell <*cell-name*> is not coming from equally-on/more-on block and has an incorrect source:

[LPSVM53_5][WARNING] Pin '<pin-name>' of instance '<inst-name>'(<cell-name>) is not coming from equally-on/more-on block, incorrect source '<source-name>'

Message 6

The following message appears when the pin <*pin-name*> of the instance <*inst-name*> of the cell <*cell-name*> is not coming from equally-on/more-on block and has an incorrect source:

[LPSVM53_6][WARNING] Pin '<pin-name>' of instance '<inst-name>'(<cell-name>) is not coming from equally-on/more-on block, incorrect source '<source-name>(<source-domain>)

Potential Issues

This violation appears if an always-on pin is not driven by an always-on domain.

Consequences of Not Fixing

If you do not fix these violations, the pin does not behave as always-on. As a result, the pin might shut down and you may not achieve the desired result.

How to Debug and Fix

For a graphical view of these violations, double-click the message and click **Incremental Schematic**. Refer to the Example Code and/or Schematic section for an example.

To fix these violations, ensure that all the always-on pins are connected to the always-on domains.

Example Code and/or Schematic

Consider the following example:

The pin `A` of library cell `GPGBUF` is made always on pin using liberty attribute `alwayson` with `input_voltage`:

```
pin (A) {
      sec_pin_type : data;
      direction : input;
      input_voltage : alwayson;
      related_ground_pin : VSSG;
      related_power_pin : VDDG;
    }
```

In the verilog file, always on pin `A` of instance `u_GPGB4` is connected to pin `Y` of instance `test_top.u_BUFEN4`.

```
module test_top (IN1);
BUF u_BUFEN4 (.A(en5), .Y(en6));
GPGBUF u_GPGB4 (.A(en6), .Y(post_iso_en));
endmodule
```

In the SGDC file, the `test_top` module lies in the voltage domain `TOP` that is not an always on domain.

```
voltage_domain -name TOP -value 1.0 0 -modname test_top -
isosig u_ALO.OUT -isoval 1
```

The pin A of instance test_top.u_GPGB4 is not driven by an always oN domain. It is driven by pin Y of instance test_top.u_BUFEN4 that does not lie in an always on domain. So, the following violation message is reported:

```
Always-on pin 'A' of instance 'test_top.u_GPGB4'(GPGBUF) is not
coming from always-on block,incorrect source
'test_top.u_BUFEN4.Y(BUF)
```

The schematic is shown in the following figure:

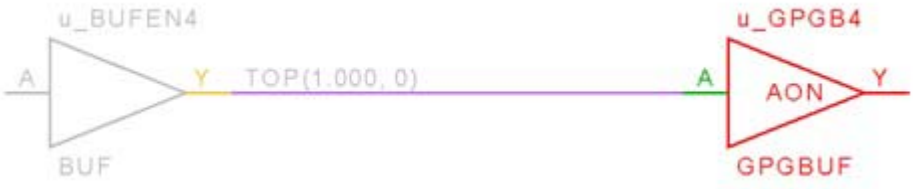


FIGURE 93. Incremental schematic

In the schematic, the LPSVM53 rule highlights the path for the pins that are not connected to the always-on domains.

Default Severity Label

Warning

Rule Group

Voltage_Domain_Rules

Reports and Related Files

None

LPSVM54

Reports always-on cells that are not present in specified domain

When to Use

Use this rule to:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

This *LPSVM54* rule checks that the always-on cells are located in the specified domain. The domain information is provided using `-locate` argument of the *always_on_cell* constraint.

Rule exceptions

The *LPSVM54* rule works on RTL when the *always_on_cell* is a .lib instance. However, the rule does not work on RTL instances.

This rule cannot be run in the UPF/CPF power format because information required for this rule is SGDC specific and is not available in power format.

Language

Verilog, VHDL

Parameter(s)

- *lp_ignore_isocell*: The default value is 0. Set the *lp_ignore_isocell* parameter 1 to make the *LPSVM54* rule ignore isolation cells.
- *lp_skip_buf*: Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.

Constraint(s)

- *voltage_domain* (Mandatory): Use to specify the always-on domains (voltage domains) in the design.

- *always_on_cell* (Optional): Use to specify the always-on cells and their domain information. You can specify `locate` argument as `AON` (for always-on domain), `OFF` (for power domain), or `BOTH` (for any domain).
- *isolation_cell* (Optional): Use to specify the always-on isolation cells.

Messages and Suggested Fix

Message 1

The following message appears for instance `<inst-name>` of always-on cell *<cell-name>* that is not in *<domain-type>* (power domain or always-on domain):

[LPSVM54_1][WARNING] `Always-on cell instance '<inst-name>'(<cell-name>) is found inside <domain-type>`

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears for instance *<inst-name>* of isolation cell *<cell-name>* that is not in an always-on domain:

[LPSVM54_2][WARNING] `Isolation cell instance 'inst-name'(cell-name) is found inside power domain`

Potential Issues

The violation messages occur because of any of the following:

- Instances of user-specified always-on-cells are not present in the specified domain.
- Instances of user-specified isolation cells, which are specified with the *isolation_cell* constraint, are not present in the always-on domain.

Consequences of Not Fixing

If these violations are not fixed, it means that the always-on-cells are present in a domain other than the domains specified explicitly in the SGDC file. This is not your intent.

Also, if an isolation cell is present in a switchable domain, the domain might go off, which leads to improper isolation.

How to Debug and Fix

The violation is reported at the instance of an always-on cell that is instantiated in domain other than the specified one.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix these violations, ensure that all the instances of the always-on cells are present in the specified domains.

Example Code and/or Schematic

Consider the following example defining the cell AN2 as always-on cell. The SGDC file is as follows:

```
current_design top

voltagedomain -name V0 -value 1.2 -modname top

voltagedomain -name V1 -value 1.2 0 -instname top.t1 -isosig
isosig_t -isoval 0 -portname isosig_t

always_on_cell -name AN2
```

After running the analysis, the following message appears:

```
[WARNING] Always-on cell instance 'top.t1.an5' (AN2) is found
inside power domain
```

The incremental schematic is as follows:

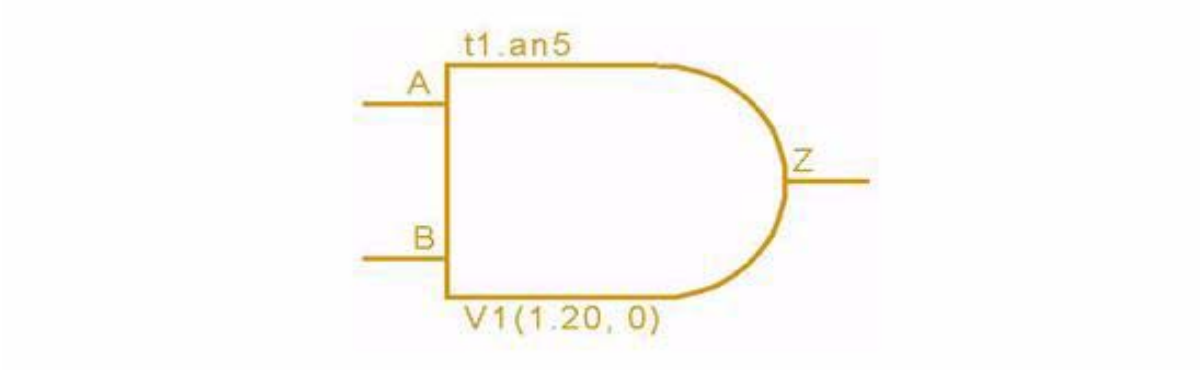


FIGURE 94. Incremental schematic

Default Severity Label

Warning

Rule Group

Always-on Logic

Reports and Related Files

None

LPSVM40

Reports the connectivity of special signals with always-on buffers

When to Use

When a domain is switched off, there may be signals in the domain that must remain on. For example, the control signals to retention registers must remain powered. In these cases, special buffers need to be used. These buffers have a special connection type for their power supply pin. Consequently, they are treated differently during power routing. Instead of being connected to the local supply, they are connected to the main supply. The local supply is switched while the main supply is not switched.

To check for presence of always-on buffer in the control signal path. These buffers should not be used in always-on domains Because of the special power connection. Only use them for certain signals in switched domains. This rule is applicable to:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/ LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM40* rule reports signals that are not connected or incorrectly connected to always-on buffers. This rule works on gate-level netlist designs only.

In the UPF Flow, this rule also checks the path of the isolation signal and reports a violation if anything other than always-on buffer is encountered. However, the check skips level shifters. When the traversal terminates at an isolation cell, this rule also check if the isolation signal is connected to the proper enable pin.

Prerequisites

Specify the information of voltage domains and their operating condition, special signals names, and always-on buffer.

Language

Verilog, VHDL

Parameter(s)

- *lp_sig_viol_count*: Default value is 50. Set this parameter to a positive integer number to specify the maximum number of violations to be reported by the rule.

Constraint(s)

- *voltage_domain* (Optional): Use this constraint to specify the name of the always-on domains in the design.
- *always_on_buffer* (Optional): Use this constraint to specify the name of the always-on buffers in the design. The *LPSVM40* rule considers buffers that:
  - ❒ are specified with this constraint,
  - ❒ are located in a voltage domain, or
  - ❒ have an always-on supply.
- *aon_buffered_signals* (Optional): Use this constraint to specify the names of the signals that should be driven by an always-on buffer. To run the *LPSVM40* rule in the CPF/UPF flow, specify this constraint in the SGDC file. The voltage domain/supply information is picked up from the corresponding CPF/UPF file.

Messages and Suggested Fix

Message 1

The following message appears at the location where the signal *<sig-name>* specified with the *aon_buffered_signals* constraint, or isolation signal for the UPF flow, is connected to the instance *<inst-name>* that is not an always-on buffer:

```
[LPSVM40_1][WARNING] Signal '<sig-name>' is connected to
instance '<inst-name>' which is not an always-on buffer
```

For debugging information, click *How to Debug and Fix*.

Message 2

In the SGDC flow, the following message appears at the location of the instance *<inst-name>* of the always-on buffer *<du-name>* when the instance is not connected to any of the signals specified with the *aon_buffered_signals* constraint:

[LPSVM40_2][WARNING] Always-on buffer '<du-name>' instantiated as '<inst-name>' is not connected to any of the user specified signals

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears at the location where the signal *<sig-name>* specified with the *aon_buffered_signals* constraint is connected to the instance *<inst-name>* of terminating cell *<termcell-name>* at pin *<pin1-name>* instead of expected pin *<pin2-name>*:

[LPSVM40_3][WARNING] Signal '<sig-name>' is connected to the terminate cell '<termcell-name>' instantiated as '<inst-name>' through the pin '<pin1-name>' and not '<pin2-name>'

For the UPF flow, this message appears at the location where the isolation signal *<sig-name>* is connected to the instance *<inst-name>* of isolation cell *<isocell-name>* at pin *<pin1-name>* instead of expected pin *<pin2-name>*.

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears at the location where the signal *<sig-name>* specified with the *aon_buffered_signals* constraint is directly connected to the instance *<inst-name>* of the terminating cell *<termcell-name>* at the expected pin:

[LPSVM40_4][WARNING] Signal '<sig-name>' is directly connected to the terminate cell '<cell-name>' instantiated as '<inst-name>'

For debugging information, click *How to Debug and Fix*.

Message 5

The following message appears at the location where the signal *<sig-name>* specified with the *aon_buffered_signals* constraint is directly connected to the instance *<inst-name>* of the terminating cell *<termcell-name>* at pin *<pin1-name>* instead of expected pin *<pin2-name>*:

[LPSVM40_5][WARNING] Signal '<sig-name>' is directly connected to the terminate cell '<cell-name>' instantiated as '<inst-name>' through the pin '<pin1-name>' and not '<pin2-name>'

Potential Issues

The potential issues are as follows:

- Only the signal name is specified using the `-names` argument of the *aon_buffered_signals* constraint.

  The *LPSVM40* rule reports all instances in the fan-out of the signal.

- The signal name is specified using the `-names` argument and the cell-pin pair name list is specified using the `-terminatingcells` argument

  The `LPSVM40` rule flags all instances in the fan-out of the signal till an instance of a user-specified terminating cell or a primary output port.

- The signal name is specified using the `-names` argument, the cell-pin pair name list is specified using the `-terminatingcells` argument, and cell name list of cells to be ignored using the `-ignorecells` argument

  The *LPSVM40* rule reports all instances in the fan-out of the signal till an instance of a user-specified terminating cell or a primary output port skipping instances of cells specified using the `-ignorecells` argument.

- The signal name is specified using the `-names` argument and cell name list of cells to be ignored using the `-ignorecells` argument.

  The *LPSVM40* rule reports all instances in the fan-out of the signal skipping instances of cells specified using the `-ignorecells` argument.

Consequences of Not Fixing

Due to the presence of a normal cell or buffer, instead of an always-on buffer, the control signal would float. Therefore, the correct control signal would not traverse to the special cells, such as isolation or retention cells.

How to Debug and Fix

The violation is reported at the instance to which aon-buffered signal, or isolation signal for the UPF flow, is incorrectly connected.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. This rule supports IS Abstraction between start and end points and abstracts all logic between these points, except level shifter cells. Refer to the *Abstraction between Start and End Points* topic in the *Atrenta Console Reference Guide*.

To resolve this violation, ensure all signals specified with the *aon_buffered_signals* constraint are connected to always-on buffers properly. For the UPF flow, ensure that the isolation signal is connected to the enable pin of the isolation cell through always-on buffers or level shifters only. Update the RTL files.

Example Code and/or Schematic

Consider the following example. A violation appears the signal `'vdd'` gets connected to a terminate cell `'FD1P'` instantiated as `'top.fd1p_inst'`:

```
[WARNING] Signal 'in1' is connected to the terminate cell
'FD1P' instantiated as 'top.fd1p_inst' through the pin 'D' and
not 'CP '
```

The following incremental schematic is displayed, as shown in following figure. This is without IS abstraction:

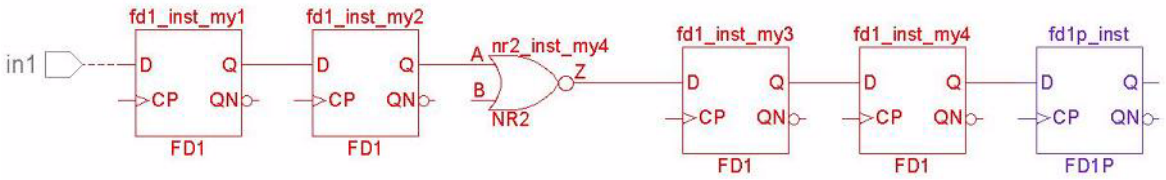


FIGURE 95. Incremental schematic without IS abstraction

The following incremental schematic is displayed, as shown in figure below.

This is with IS abstraction:

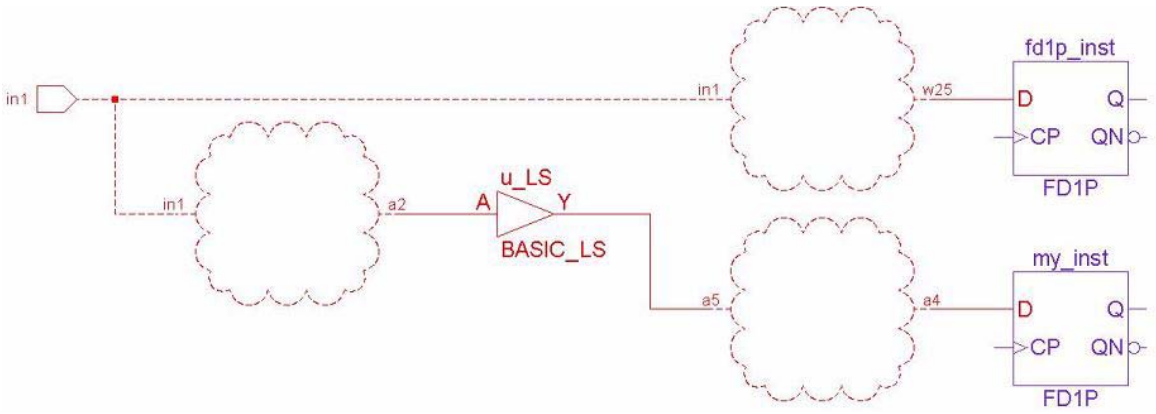


FIGURE 96. Incremental schematic with IS abstraction

Default Severity Label

Warning

Rule Group

Power_Domain_Rules

Reports and Related Files

None

State Retention Rules

The State Retention rules are as follows:

Rule	Reports
LPPLIB10	Reports incorrectly connected power pins of SRPG cells
LPSVM38	Incorrectly instantiated SRPG cells
LPSVM56	Check for the presence of always-on buffers in the path of sleep signal
LPSVM56A	Always-on buffer instances that are present in an always-on domain
LPSVM56B	Presence of always-on buffers in the path of sleep signal
LPSVM57	Incorrectly connected or unconnected control pins of retention cells
LPSVM58	Generate OVL assertion file for clock input to retention cells
LPSVM59	Ensure control pin(s) of retention cells attains the specified value
LPRET01	Checks if percentage of retention cells in sequential elements exceeds the set limit
LPRET02	Checks conflicting values of save restore signals in retention strategies
LPRET03	Reports partially or fully redundant strategies
LPRET03A	Reports if a retention strategy is applied on an always-on domain
LPRET03B	Reports if multiple retention strategies are applied on the same retention element
LPRET04	Reports power domains with partially or completely missing retention strategies
LPRET04A	Reports if a switching domain does not have a retention strategy
LPRET04B	Reports if retention strategy is specified for fewer sequential cells in a switching domain than the recommended percentage

LPPLIB10

Reports incorrectly connected power pins of SRPG cells

When to Use

NOTE: *This rule will be deprecated for the UPF flow, in a future release. It is recommended to use the LPPLIB06 rule instead, in the UPF flow.*

Use this rule for:

- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPPLIB10* rule reports instances of the specified SRPG cells where the Vdd/Vddc pins are not connected to the correct supply pins, specified with the `-supplyname` argument of the *voltage_domain* constraint and the *supply* constraint.

Prerequisites

The *LPPLIB10* rule requires you to specify the SRPG cell names along with their Vdd/Vddc pin names using the *retention_cell* constraint.

Rule Exceptions

This rule does not check for bias pins. The *LPPLIB18* rule checks for bias pins.

This rule ignores SRPG cells, where Vddc pin name of the cell is not specified using the *retention_cell* constraint.

Parameter(s)

- *lp_max_viol_count*: You can set the maximum number of rule messages to be reported by the rule with the `lp_max_viol_count` rule parameter. By default, the `lp_max_viol_count` rule parameter is set to `1000`, that is, a maximum of 1000 rule messages are reported.

Constraint(s)

SGDC

- *retention_cell* (Mandatory)
- *voltage_domain* (Mandatory)
- *supply* (Mandatory)

CPF Commands

- *create_power_nets* (Mandatory)
- *create_ground_nets* (Mandatory)
- *create_power_domain* (Mandatory)
- *update_power_domain* (Mandatory)
- *define_state_retention_cell* (Mandatory)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_retention* (Mandatory)
- *set_retention_control* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears at the location of instance <*inst-name*> of SRPG cell <*cell-name*> when the <*pin-name*> pin is not connected to the correct supply port <*supply-port-name*>:

[LPPLIB10_1][RECOMMENDED] Pin '<pin-name>' of the SRPG cell '<cell-name>' instantiated as '<inst-name>' is not correctly connected to the supply '<supply-port-name>'

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears at the location of instance <*inst-name*> of SRPG cell <*cell-name*> when no supply is defined for the switchable pin <*pin-name*>:

[LPPLIB10_2][RECOMMENDED] No supply defined for the switchable supply pin '<pin-name>' of SRPG cell '<cell-name>' instantiated as '<inst-name>'

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears at the location of instance <*inst-name*> of SRPG cell <*cell-name*> when no supply is defined for the always on supply pin <*pin-name*>:

[LPPLIB10_3][RECOMMENDED] No supply defined for the always on supply pin '<pin-name>' of SRPG cell '<cell-name>' instantiated as '<inst-name>'

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears at the location of instance <*inst-name*> of SRPG cell <*cell-name*> when <sup-type1> (Switchable, Always On or Ground) supply <sup-name> is connected to <sup-type2> (always on, switchable, or always on/switchable, respectively) supply pin <*pin-name*>:

[LPPLIB10_4][RECOMMENDED] <sup-type1> supply '<sup-name>' connected to the <sup-type2> supply pin '<pin-name>' of SRPG cell '<cell-name>' instantiated as '<inst-name>'

For debugging information, click *How to Debug and Fix*.

Message 5

An informational message appears when the violation count of this rule exceeds the limit set by the *lp_max_viol_count* parameter. Refer to *Message 5* for the message and, for debugging information, refer to *How to Debug and Fix*.

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The consequences of not fixing are:

- **Message 1**: The switchable/always-on pin of the SRPG cell is not connected to a switchable/always-on supply in the design. This is a design flaw.
- **Message 2**: The switchable pin of the SRPG cell does not have a switchable supply specified in its parent domain. This is a design flaw.
- **Message 3**: The always-on pin of the SRPG cell does not have an always-on supply specified in its parent domain. This is a design flaw.
- **Message 4**: The switchable/always-on pin of the SRPG cell is incorrectly connected to a always-on/switchable supply in the design. This is a design flaw.

How to Debug and Fix

The violation is reported at the place where SRPG cell with incorrectly connected power pins is instantiated.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Turn on the Power View. The schematic for the message shows the following:

- the instance (either flip-flop or retention cell)
- the voltage domain in which the reported instance lies

To resolve this violation, ensure that the always-on and switchable pins of the SRPG cell are routed to the correct supplies, respectively.

Example Code and/or Schematic

Example 1

Consider the following example. A violation occurs when power pin `VDD` of SRPG cell is not correctly connected to the supply `V1`. The following message appears:

```
[RECOMMENDED] Pin 'VDD' of the SRPG cell 'R120_ASLSIL120X160'
instantiated as 'top.mid_inst.bot_inst.inst1' is not correctly
connected to the supply 'V1 '
```

For a graphical view of the violation, double-click the message and click the Incremental Schematic button. The incremental schematic is displayed as shown in figure below.

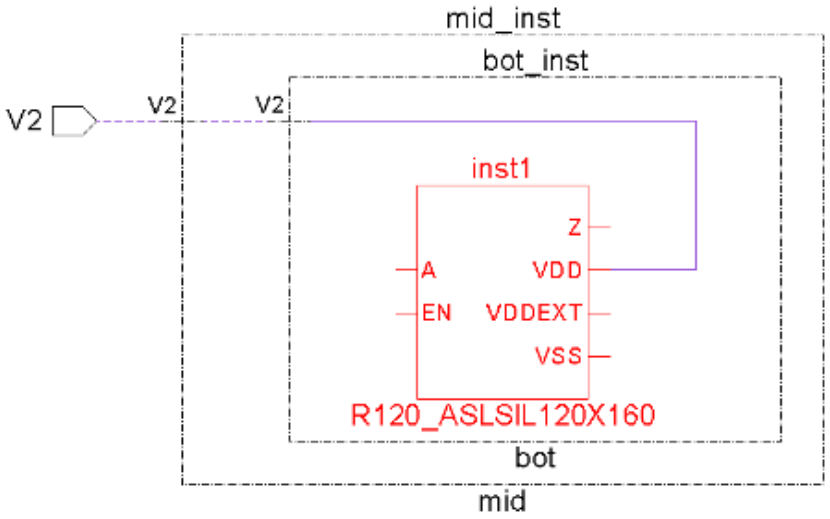


FIGURE 97. Incremental schematic

The schematic highlights the following:

- Pin that is not connected to the correct supply port.
- SRPG cell for which no supply is defined for vdd/vddc pin.

Example 2

Consider the following example showing power pin VDDC connected to supply pin VDD. The SpyGlass Design Constraints file is as follows:

```
current_design top
supply -name VDD -value 1.2 -alwayson 1
supply -name VDDA -net ua.VDDA -value 1.2 -alwayson 0 \
  -parent VDD -on enA1 enA2
supply -name VDDB -value 1.2 -alwayson 0 -parent VDD \
  -on enB
voltage_domain -name Vtop -value 1.2 -modname top \
  -supplyname VDD
```

```
voltage_domain -name VA -value 1.2 0 -instname top.ua \
  -supplyname VDD VDDA -noisosig
voltage_domain -name VB -value 1.2 0 -instname top.ub \
    -supplyname VDD VDDB -noisosig
retention_cell -name RETFF -domains VA VB -vddcpin VDDC
```

After running the analysis, the following message appears:

```
[RECOMMENDED] Pin 'VDDC' of the SRPG cell 'RETFF' instantiated
as 'top.ua.u4' is not correctly connected to the supply 'VDD '
```

To see details of the violation, click the message and then click the **Incremental Schematic** button.

Default Severity Label

Recommended

Rule Group

Power_Domain_Rules

Reports and Related Files

None

LPSVM38

Checks the presence of SRPG cell in the specified hierarchy

When to Use

Use this rule to check for the appropriate usage of retention registers. This rule is applicable to:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM38* rule reports inappropriate usage of retention registers.

When a domain is powered off, all of its flops or registers lose their stored values. In some cases, this may be the desired behavior. In modern designs it is desirable, if a controller is powered off and is intended to resume at the same state after power is returned, the values must be stored during power down.

To achieve this, special registers called retention registers are used. These registers have two power supply pins, two storage elements, and generally one or two extra control pins. The second "shadow" storage element is powered by the main supply, which is not switched off. Before power off, the extra control pins are used to copy the stored value into the always-on storage. After power on, the stored value is reloaded into the switched storage element.

The cells are considered as retention cells by SpyGlass in the following cases:

- If the *retention_cell* SGDC constraint is provided for that cell in the SGDC file.
- If the `retention_cell=yes` attribute is provided on the cell in the library file.
- If the `define_state_retention_cell` command is provided for that cell in the CPF file.

The rule reports a violation when:

- Retention register is either used in an always-on block or used outside the specified hierarchy of power domain (redundant cell)
- Flip-flop is used in the specified hierarchy of power domain instead of being mapped to a retention register (missing cell)
- The retention register mapped is different from the user specified cell (incorrect cell)

Prerequisites

The *LPSVM38* rule requires you to specify the retention rules on SRPG cells/instance hierarchies. Then, this rule reports the instances of the user-specified SRPG cells in a power domain other than the specified power domains or in a hierarchy other than the specified instance hierarchies.

Use this rule after synthesis is complete and flip-flop cells are mapped with the corresponding technology libraries.

Language

Verilog, VHDL

Parameter(s)

- *lp_max_viol_count*: Default value is `1000`. Set this parameter to a positive integer number to specify the maximum number of violations that should be reported by the rule.

Constraint(s)

SDGC

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage/power domains in the design.
- *retention_cell* (Mandatory): Use this constraint to specify retention cells.
- *retention_instance* (Optional): Use this constraint to specify the hierarchy information of retention cell instances.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_state_retention_rule* (Mandatory)
- *update_state_retention_rules* (Optional)
- *create_nominal_condition* (Mandatory)

- *create_power_mode* (Mandatory)
- *define_state_retention_cell* (Mandatory)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_retention* (Mandatory)
- *set_retention_control* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears at the location of instance <*inst-name*> of the SRPG cell <*cell-name*> when the instance is found outside the associated domain list <*pd-list*>:

[LPSVM38_1][RECOMMENDED] SRPG cell '<cell-name>' instantiated as '<inst-name>' does not lie in any of the domains '<pd-list>' specified

This message is reported when a particular cell is associated with a domain but it is found outside the domains specified. You can specify the cells for a particular domain using:

- domain of the map_retention_cell UPF command
- domains of the retention_cell SGDC constraint
- cells/–cell_type of the update_state_rentention_rules CPF command

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears at the location of instance <*inst-name*> of the SRPG cell <*cell-name*> when the instance does not belong to any of the corresponding instance hierarchies:

[LPSVM38_2][RECOMMENDED] SRPG cell '<cell-name>' instantiated as '<inst-name>' does not lie in any of the hierarchical names specified

You can specify the cells for a particular hierarchy using:

- elements of the map_retention_cell UPF Command
- name of the retention_instance SGDC constraint
- instances of the update_state_rentention_rules CPF command

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when the instance <*inst-name*> of flip-flop cell or latch cell <cell-name> is found to be in the retention instance hierarchy <*inst-hier*> but the flip-flop cell or latch cell does not have retention_cell attribute in the library or is mentioned in the -elements parameter in *set_retention* command of the UPF:

[LPSVM38_3][RECOMMENDED] Flop '<inst-name> (<cell-name>)' found in hierarchy '<inst-hier>' is not an SRPG cell

This happens when the flip-flops/latches inside the retention hierarchy have not been mapped to the retention cell correctly.

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears when a retention cell does not have a retention strategy specified:

[LPSVM38_4][RECOMMENDED] Retention cell (<ret-cell-name) instantiated as '<inst-name>' does not have retention strategy specified

This happens when:

- the flip-flops/latches outside the retention hierarchy have been incorrectly mapped to the retention cell, or
- the retention hierarchy was not provided correctly in the UPF file.

For debugging information, click *How to Debug and Fix*.

Message 5

The following message appears when the violation count of this rule

exceeds the limit set by the *lp_max_viol_count* parameter:

```
[LPSVM38_5][INFO] Violation count of this rule reached maximum
permissible limit (<lp_max_viol_count-value>) of
lp_max_viol_count parameter. Please increase the value of this
parameter if you want the rule to report more violations
```

Message 6

The following violation message is reported if the retention cell placed in the design is not available in the `lib_cells` or `lib_cell_type` argument of the corresponding `map_retention_cell` command:

```
[LPSVM38_6][WARNING] Incorrect retention cell '<inst-
name>(<cell-name>)' placed in design corresponding to retention
strategy '<ret-strategy-name>'. Retention cell is not present
in map_retention_cell command
```

Potential Issues

There are flops that have been used in a powered off domain. Alternatively, retention registers have been used in power-on domains.

Consequences of Not Fixing

The flops used in powered-off domain or specified hierarchy will not retain their values and their previous value would be lost.

Retention registers used in powered-on domain would consume more power than normal registers.

How to Debug and Fix

The message states the retention cell is not placed in the correct hierarchy or a flip-flop is located at a place inside the hierarchy, where a retention cell should have been used. The violation is reported at the place where the SRPG cell is instantiated in a power domain other than the specified power domain or in a hierarchy other than the specified instance hierarchies.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Turn on the Power View. The schematic for the message shows the following:

- the instance (either flip-flop or retention cell)
- the voltage domain in which the reported instance lies

To resolve these violations, ensure that the SRPG cell is located in the specified domain or hierarchies. The specified hierarchy should not have

any flip-flops other than SRPG cells. Update the RTL files.

To resolve *Message 5*, increase the value of the *lp_max_viol_count* parameter.

Refer to the *Example Code and/or Schematic* section to view a sample schematic and to understand this rule.

Refer to the *Reports and Related Files* section for information on the reports generated by this rule.

Example Code and/or Schematic

Example 1

Consider following case where retention strategy is defined in UPF as follows:

```
create_power_domain VON
create_power_domain PD1 -elements {mid/inst1 mid}
create_power_domain PD2 -elements mid/inst2
set_retention RET1 -domain PD1
map_retention_cell RET1 -domain PD1 -lib_cells FD1
set_retention RET2 -domain PD2
map_retention_cell RET2 -domain PD2 -lib_cells FD1
set_retention RET3 -domain PD1 -elements mid
```

The above retention strategy requires that instances in hierarchy `mid/inst1` and `mid/inst2` should be retention cells.

The module `mid` is defined as below:

```
module MID (in1,in2,in3,in4,clk,out1,out2,out3,out4);
    input in1,in2,in3,in4,clk;
    output out1,out2,out3,out4;
    PD inst1(in1,in2,in3,in4,clk,out1,out2,out3,out4);
    PD inst2(in1,in2,in3,in4,clk,out1,out2,out3,out4);
endmodule
module PD (in1,in2,in3,in4,clk,out1,out2,out3,out4);
    input in1,in2,in3,in4,clk;
    output out1,out2,out3,out4;
    FD1 fd1(.D(in1),.CP(clk),.Q(out1));
    FD2 fd2(.D(in2),.CP(clk),.Q(out2));
```

```
    FD1P fd3(.D(in3),.CP(clk),.Q(out3));
    FD2P fd4(.D(in4),.CP(clk),.Q(out4));
endmodule
```

In the above example, out of the flip-flops FD1, FD2, FD1P, and FD2P only FD1 is a retention cell. The LPSVM38 rule reports the following violation messages for the other three flip-flops:

```
Flop 'top.mid.inst1.fd2 (FD2)' found in hierarchy 'top.mid' is not an SRPG cell

Flop 'top.mid.inst2.fd2 (FD2)' found in hierarchy 'top.mid' is not an SRPG cell

Flop 'top.mid.inst1.fd3 (FD1P)' found in hierarchy 'top.mid' is not an SRPG cell

Flop 'top.mid.inst2.fd3 (FD1P)' found in hierarchy 'top.mid' is not an SRPG cell

Flop 'top.mid.inst1.fd4 (FD2P)' found in hierarchy 'top.mid' is not an SRPG cell

Flop 'top.mid.inst2.fd4 (FD2P)' found in hierarchy 'top.mid' is not an SRPG cell
```

The following is the incremental schematic:

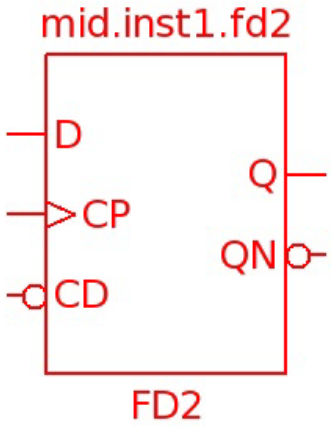


FIGURE 98. Incremental schematic

Example 2

Consider the following example. A violation occurs when the instance `'top.fd1'` of SRPG cell `'FD1'` does not belong to any of the corresponding instance hierarchies:

```
[RECOMMENDED] SRPG cell 'FD1' instantiated as 'top.fd1' does
not lie in any of the hierarchical names specified
```

The incremental schematic is displayed as shown in figure below:

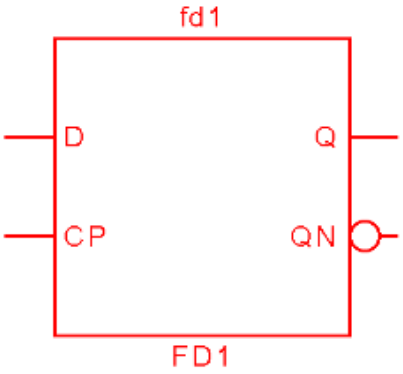


FIGURE 99. Incremental schematic

The *LPSVM38* rule highlights the following:

- SRPG cell that does not lie in any of the domains specified.
- SRPG cell does not lie in any of the hierarchical names specified.
- Flip-flop found in hierarchy is not an SRPG cell.

Example 3

Consider the following example where retention strategy R1 is applied on retention instance inst1/ret1(cell: RET1):

UPF:

```
set_retention R1 -domain PD1 -elements {inst1/ret1} -
retention_supply_set TOP.primary -save_signal {save low} -
restore_signal {res high}
```

```
map_retention_cell R1 -domain PD1 -lib_cells {RET2}
```

Verilog:

```
module top (input in1,SAV,RST,output out2);
Test1 inst1
(.in1(in1),.out1(out1),.save(SAV),.restore(RST));
endmodule

module Test1 (input in1,save,restore, output out1);
RET1 ret1 (.d(in1),.o(out1),.SAVE(save),.RESTORE(restore));
Endmodule
```

For the above example, the *LPSVM38* rule reports the following violation because the cell `RET1` is not mentioned in the `-lib_cells` list for the retention strategy `R1`:

```
Incorrect retention cell 'top.inst1.ret1(RET1)' placed in
design corresponding to retention strategy 'R1'. Retention cell
is not present in map_retention_cell command
```

Default Severity Label

Recommended

Rule Group

Power_Domain_Rules

Reports and Related Files

- *lp_constr_info*: Lists details about the retention cells. This section of the report shows the following information:
  - List of retention cells specified by the user or available in the .lib file
  - List of hierarchies in the design, where a retention cell should be used

LPSVM56

Check for the presence of always-on buffers in the path of sleep signal

Language

Verilog, VHDL

Rule Description

The LPSVM56 rule flags always-on buffer instance that are not present in a power domain in path of sleep signal of a power domain. It also flags non-always-on buffer instances that are present in the fan-out of the sleep signal of the power domain.

The LPSVM56 rule is divided into two sub-rules, *LPSVM56A* and *LPSVM56B*. When you select the LPSVM56 rule, both these rules are run.

NOTE: *A library cell with the* `retention_cell` *attribute specified, is recognized as a retention cell. For details of the library attributes, refer to Using Constraints in the SpyGlass Power Verify Solution section.*

LPSVM56A

Checks for always-on buffers in the always-on domain

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM56A* rule reports always-on buffer instances that are present in an always-on domain. Each always-on buffer instance must be present in a switched voltage domain.

Rule Exceptions

The *LPSVM56A* rule does not check for SpyGlass-generated buffer instances.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the power domain and their sleep nets by using the `-sleepnet` argument.
- *always_on_buffer* (Mandatory): Use to specify the always-on buffer cells.
- *retention_cell* (Optional): Use to specify the retention cells.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_state_retention_rule* (Mandatory)

- *update_state_retention_rules* (Optional)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_state_retention_cell* (Mandatory)

Messages and Suggested Fix

The following message appears at the location of always-on buffer instances, when instance *<inst-name>* of always-on buffer *<cell-name>* is present in an always-on domain *<domain-name>*:

[LPSVM56A_1][WARNING] Always-on buffer '<cell-name>' instantiated as '<inst-name>' is present in always-on domain <domain-name>

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

There is no special need of an always-on buffer in an always-on domain because the domain will never shut down. Therefore, a normal buffer is as good as an always-on buffer. Even if you do not fix this violation, it will not cause a design error.

How to Debug and Fix

The violation is reported at the always-on buffer instances that are present in always-on domain.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix the violation, ensure that the always-on buffer instances are present in a switched domain.

Example Code and/or Schematic

Consider the following example. A violation occurs when the always-on buffer instance `'top.iso1'` is present in always-on domain. The following message appears:

```
Always-on buffer 'iso' instantiated as 'top.iso1' is present in always-on domain 'V1'.
```

The schematic is as follows:

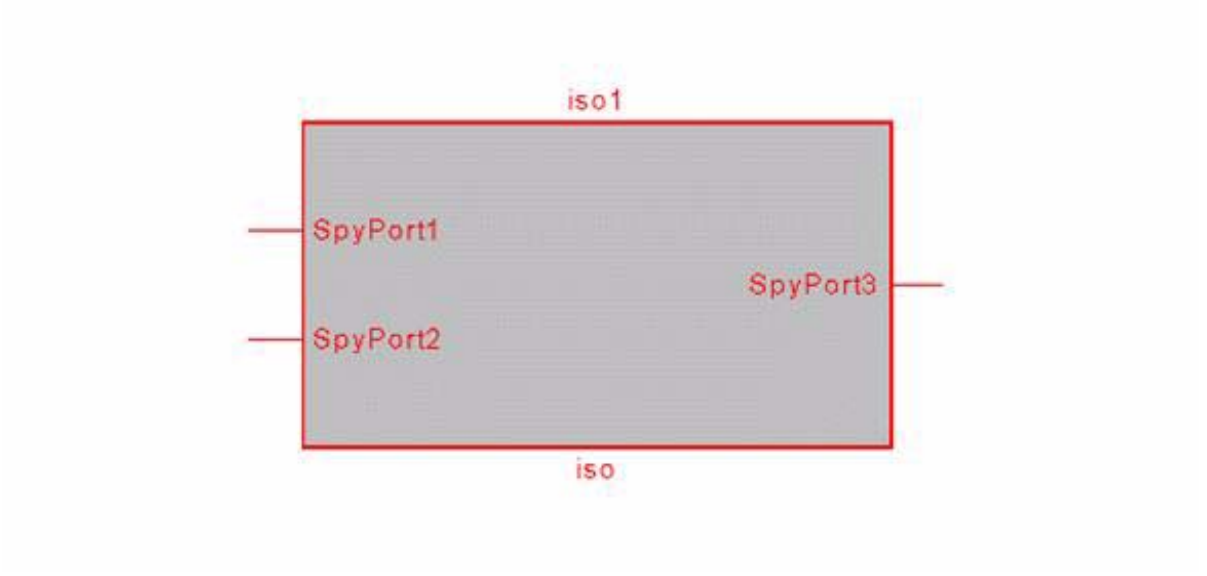


FIGURE 100. Incremental schematic

The schematic highlights the always-on buffer present in the always-on domain for which the violation is being reported.

Default Severity Label

Warning

Rule Group

`State Retention`

Reports and Related Files

- *lp_lib_data*: This report provides information about the always-on buffers.

- *lp_constr_info*: This report provides information about the always-on domains.

LPSVM56B

Check for always-on buffers in the path of retention control signals

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM56B* rule reports always-on buffer instances that are not present in the fan-out of the retention control signal of the power domain and relatively non-always-on buffer instances that are present in the fan-out of the retention control signal of the power domain.

Rule Exceptions

The *LPSVM56B* rule does not check for SpyGlass-generated buffer instances.

Language

Verilog, VHDL

Parameter(s)

- *lp_check_aob_on_control*: Default value is 0, and the rule does not report always-on buffer instances not present in the path of control (sleep/save/restore) signal. Set the value to 1 to report always-on buffer instances not present in the path of control (sleep/save/restore) signal.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the power domain and their sleep nets specified using the `-sleepnet` argument.
- *always_on_buffer* (Optional): Use to specify the always-on buffer cells.
- *retention_cell* (Optional): Use to specify the retention cells.

- *set_case_analysis* (Optional): Use to specify the case analysis conditions used by the *LPSVM56B* rule.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_state_retention_cell* (Mandatory)
- *define_always_on_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_retention* (Mandatory)
- *set_retention_control* (Mandatory)
- *set_retention_control* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the instance *<inst-name>* of the Always-on buffer *<cell-name>* is in a location other than the fan-out of the sleep signal of any power domain:

[LPSVM56B_1][INFO] Always-on buffer '<cell-name>' instantiated as '<inst-name>' is not present in the signal path of retention control signal

The above message is reported only when the *lp_check_aob_on_control* parameter is set to `yes or 1`.

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the instance *<inst-name>* of the Always-on buffer *<cell-name>* is in the fan-out of the <sig-type> (sleep/save/restore) signal <net-name> of the power domain *<domain2-name>* when the instance belongs to the voltage/power domain *<domain1-name>*:

[LPSVM56B_2][WARNING] <sig-type> signal '<net-name>' of power domain '<domain1-name>' present in incorrect domain '<domain2-name>' has an always-on buffer '<inst-name>'('<cell-name>') in its path

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when the instance *<inst-name>* of a non always-on buffer is in the fan-out of the <sig-type> (sleep/save/restore) signal *<net-name>* of the power domain *<pd-name>*:

[LPSVM56B_3][WARNING] Instance '<inst-name>' found in the path of the <sig-type> signal '<net-name>' of power domain '<pd-name>' is not an always-on buffer

Potential Issues

These violation messages occur because of any of the following:

- Instances of Always-on buffers that are not present in the fan-out of the control (sleep/save/restore) net of the corresponding power domain.
- Instances of Always-on buffers that are present in the fan-out of the control (sleep/save/restore) net of a power domain when the instance belongs to a different voltage/power domain.
- Instances of non-Always-on buffers that are present in the fan-out of the control (sleep/save/restore) net of the associated power domain.

Consequences of Not Fixing

The actual control signal may not reach the retention cell and it can change due to the presence of some cell or a non always-on buffer in its path. Thus the retention cell may not work properly.

How to Debug and Fix

The violations are reported at the always-on buffer instances that are not

present in the fan-out of the sleep signal of the power domain and non-always-on buffer instances that are present in the fan-out of the sleep signal of the power domain.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix these violations, ensure that the:

- Ensure that the always-on buffer instances are present in the fan-out of the control (sleep/save/restore) signal of the power domain.
- Ensure that the non-always-on buffer instances are not present in the fan-out of the control (sleep/save/restore) signal of the power domain.

Example Code and/or Schematic

Consider the following example. A violation occurs when an instance `top.pd1.isoInst2` of Always-on buffer `iso` is in the fan-out of the sleep signal `top.isosig2` of power domain `V3` when the instance belongs to voltage/power domain V2. The following message appears:

```
[WARNING] Sleep signal 'top.isosig2' of power domain 'V3'
present in incorrect domain 'V2' has an always-on buffer
'top.pd1.isoInst2'('iso') in its path
```

The schematic is as follows:

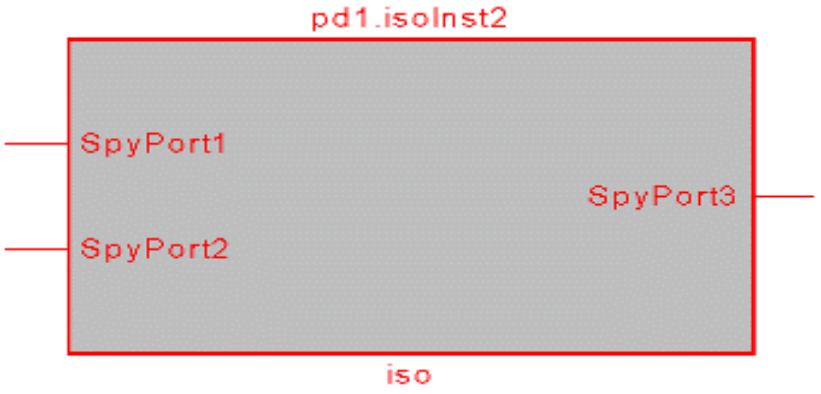


FIGURE 101. Incremental schematic

The schematic shown above highlights the instance of the always-on buffer which is present in path of a control signal of another domain. Depending on violation message, the schematic may also highlight the following:

- An instance of the always-on buffer that is not present in path of control signal, or
- An instance that is not an always-on buffer but is present in the path of the control signal.

Default Severity Label

Warning

Rule Group

State Retention

Reports and Related Files

None

LPSVM57

Reports incorrectly connected control pins of retention cells

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM57* rule reports those retention cell instances where the control pin is not properly connected to the net of the associated power domain.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *retention_cell* (Mandatory): Use to specify the retention cells and their control pins (sleep, save or restore) pins (using the `-sleep`, `-save` or `-restore` arguments).
- *voltage_domain* (Mandatory): Use to specify the power domain and their signal (sleep, save or restore) nets (using the `-sleepnet`, `savenet`, `-restorenet` arguments).
- *set_case_analysis* (Optional): Use to specify the case analysis conditions.
- *always_on_cell* (Optional): Use to specify the always-on cells and their domain information.
- *always_on_buffer* (Optional): Use to specify the names of always-on buffers.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_state_retention_cell* (Mandatory)
- *define_always_on_cell* (Optional)
- *create_state_retention_rule* (Optional)
- *update_state_retention_rules* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_retention* (Mandatory)
- *set_retention_control* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the `<pin-type>` (sleep, save or restore) pin `<pin-name>` of instance `<inst-name>` of retention cell `<cell-name>` is connected to a net other than a `<signal-type>` (sleep, save or restore) net `<net-name>` of the associated power domain `<pd-name>`:

```
[LPSVM57_1][WARNING] <pin-type> pin '<pin-name>' of instance
'<inst-name>' (cellname '<cell-name>') should be connected to
the <signal-type> net '<net-name>' of the power domain
'<pdname>'
```

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the <pin-type> (sleep, save or restore) pin *<pin-name>* of instance <inst-name> of retention cell <cell-name> is not connected to the net <net-name>:

[LPSVM57_2][WARNING] <pin-type> pin '<pin-name>' of instance '<inst-name>' (cellname '<cell-name>') should be connected to the specified <net-name> net <signal-type>

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when the <pin-type> (sleep, save or restore) pin *<pin-name>* of instance <inst-name> of retention cell <cell-name> is connected to a net <net-name>, but no corresponding control signal <signal-type> (sleep, save or restore) is specified for the associated power domain:

[LPSVM57_3][WARNING] <pin-type> pin '<pin-name>' of instance '<inst-name>' (cellname '<cell-name>') is wrongly connected to net <net-name>, which is not defined as <signal-type> net

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears when the <pin-type> (sleep, save or restore) pin <pin-name> of instance <inst-name> of retention cell <cell-name> is unconnected:

[LPSVM57_4][WARNING] <pin-type> pin '<pin-name>' of instance '<inst-name>' (cellname '<cell-name>') is unconnected

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

If this violation is not fixed, it means that control pins of the retention cell are not connected properly to their respective control signals. As a result, the retention cell may not work properly.

How to Debug and Fix

The violation is reported at the place where retention cell with incorrectly

connected or unconnected control pin(s) is instantiated.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the *Example Code and/or Schematic* section for an example.

To resolve these violations, ensure that the control pins of retention cells are properly connected to the net of the associated power domain.

Example Code and/or Schematic

Consider the following example. A violation occurs when the `<pin-type>` (sleep, save or restore) pin `'D'` of instance `'top.mid_inst.fd1_inst'` of retention cell `'FD1'` is connected to a net other than a *`<signaltype>`* (sleep, save or restore) net `'top.rst'` of the associated power domain `'V2'`:

```
[WARNING] 'Sleep' pin 'D' of instance 'top.mid_inst.fd1_inst'
(cellname 'FD1') should be connected to the 'sleep' net
'top.rst' of the power domain 'V2'
```

The schematic is as follows:

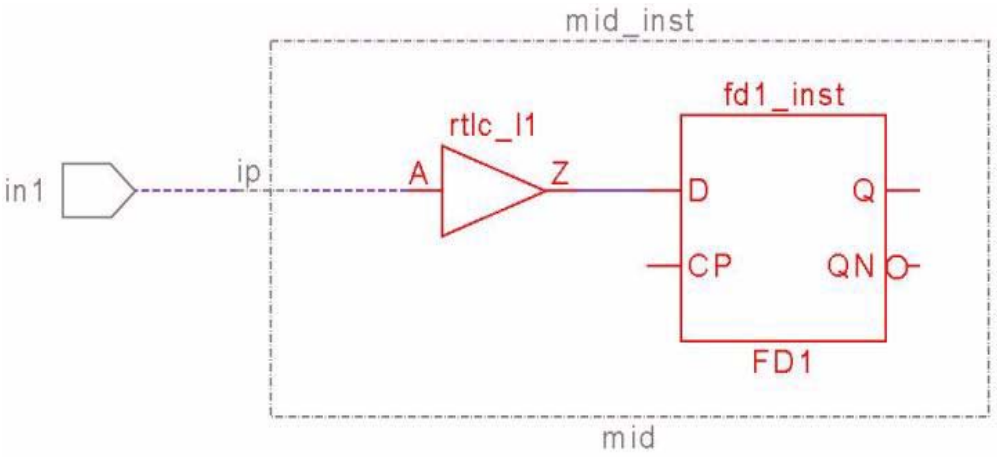


FIGURE 102. Incremental schematic

The schematic highlights the instance of the retention cell whose control pin (sleep, save, or restore) is not properly connected to the corresponding

control signal.

Default Severity Label

Warning

Rule Group

State Retention

Reports and Related Files

None

LPSVM58

Generate an OVL assertion file for clock input to retention cells

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM58* rule generates the *lp_assertion_info* report. This report contains Open Verification Library (OVL) assertions that can be used to formally check:

- Clock pin of retention cell does not toggle, when sleep and stopclock net are active
- Have a desired state at clock pin during power-down, when sleep is active

NOTE: *A library cell with the* `retention_cell` *attribute specified is recognized as a retention cell. For details of the library attributes, refer to the Using Constraints in the SpyGlass Power Verify Solution section.*

Prerequisites

Specify the `-sleepnet` argument of the *voltage_domain* constraint with the `-sleepval` argument. The *LPSVM58* rule does not generate assertions if the `sleepnet/-sleepval` pair is not defined for a power domain.

Rule Exceptions

The *LPSVM58* rule cannot be run in the UPF/CPF power format because information required for this rule is SGDC specific and is not available in power format.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *retention_instance* (Mandatory): Use to specify the hierarchy information of retention cell instances.
- *voltage_domain* (Mandatory): Use to specify the power domain along with their sleep nets and their active values by using the `-sleepnet/sleepval` arguments.
- *retention_cell* (Optional): Use to specify the retention cells along with their clock pins, using the `clk` argument, and the desired clock values during power down by using the `clkval` argument.

Messages and Suggested Fix

The following message appears when the *lp_assertion_info* report is generated:

[INFO] `Report lp_assertion_info containing OVL assertion statement is generated in $dir_name directory`

Potential Issues

Not applicable

Consequences of Not Fixing

Not applicable

How to Debug and Fix

To view the *lp_assertion_info* report, double-click the violation message.

Example Code and/or Schematic

Example 1

This is a gate-level example. In this case, retention cells are present in the design. The constraints are specified as shown:

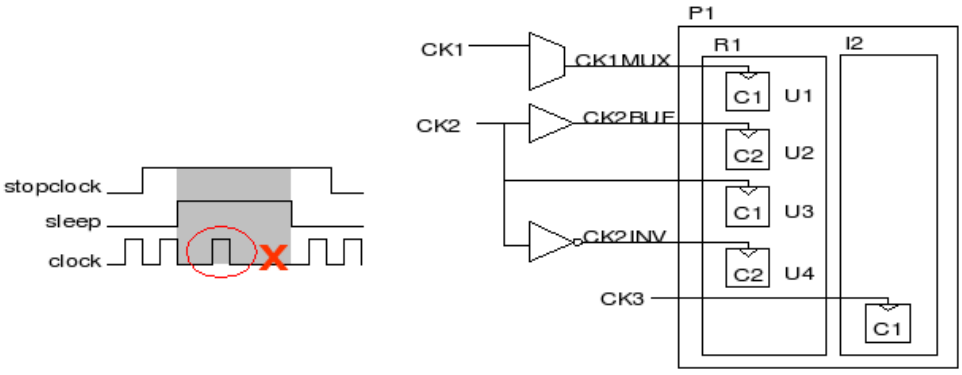


FIGURE 103. Retention cells in the design

The constraints specified are as follows:

```
voltage_domain -instname P1 -sleepnet SL1 -sleepval 1

retention_instance -name P1.R1

retention_cell -name C1 -clk CP -clkval 1
retention_cell -name C2 -clk CP -clkval 0
```

Here, `-clkval 0` or `1` defines the stop clock value.

The *LPSVM58* rule traces back the clock from the retention cells through buffers or inverters and generates the following assertions:

```
assert_proposition valid_clk_1(1'b1, !SL1 | CK1MUX);

assert_proposition valid_clk_2(1'b1, !SL1 | !CK2);

assert_proposition valid_clk_3(1'b1, !SL1 | CK2);
```

The `assert_proposition (reset_n, test_expr)` triggers when the `test_expr` is false.

Example 2

This is an RTL example. This scenario applies to RTL sequential processes and any sequential cell that is not specified by the *retention_cell* constraint. The *LPSVM58* rule generates assertions for each clock in the specified

instances.

The constraints specified are as follows:

```
voltage_domain -inst P1 -sleep SL1 -sleepval 1

retention_instance -name top.P1.R1
```

The *LPSVM58* rule generates the following assertions for each clock found inside the specified instances:

```
assert_never valid_clk_1r (clock1, SL1, SL1 && clock1);

assert_never valid_clk_1f (!clock1, SL1, SL1 && !clock1);
```

`assert_never (clk, reset_n, test_expr)` triggers on rising edge of clock if `reset_n` is not asserted (active low) and `test_expr` is set to true.

Default Severity Label

Info

Rule Group

State Retention

Reports and Related Files

- *lp_assertion_info*: This report contains OVL assertions.

LPSVM59

Checks whether control pins of retention cells attain the specified value

When to Use

Use this rule to check if the control pin of retention cell is properly connected to the net of the associated power domain. Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM59* rule reports retention cell instances where the control pin (sleep, save and restore) is not properly connected to the signal (sleep, save and restore) net of the associated power domain. For example, the *LPSVM59* rule reports a violation message when an active high sleep net is connected to a retention cell which has an active low sleep pin.

Language

Verilog, VHDL

Parameter(s)

- *lp_set_sim_val_x*: Default value is `1`. This means that the value for *set_case_analysis* and retention control signals is not propagated, if the combination logic lies in power domain. Set the value to `0` to enable the LPSVM59 rule to propagate the value for *set_case_analysis* and retention control signals also for the logic that lies in the power domain.

Constraint(s)

SGDC

- *retention_cell* (Mandatory): Use to specify the retention cells and their control pins (sleep, save or restore) pins (using the `-sleep`, `-save` or `-restore` arguments respectively) and active values of these control

pins (using the `sleepval`, `-saveval`, `-restoreval` arguments respectively).

- *voltage_domain* (Mandatory): Use to specify the power domain and their signals (sleep, save or restore) nets (using the `-sleepnet`, `savenet`, `-restorenet` arguments respectively) and active values of these control pins (using the `sleepval`, `-saveval`, `-restoreval` arguments respectively).
- *set_case_analysis* (Optional): Use to specify the case analysis conditions.
- *always_on_cell* (Optional): Use to specify the always-on cells and their domain information.
- *always_on_buffer* (Optional): Use to specify the names of always-on buffers.

CPF Commands

- *create_power_domain* (Mandatory)
- *create_nominal_condition* (Mandatory)
- *create_power_mode* (Mandatory)
- *define_state_retention_cell* (Mandatory)
- *define_always_on_cell* (Optional)
- *create_state_retention_rule* (Optional)
- *update_state_retention_rules* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_retention* (Mandatory)
- *set_retention_control* (Mandatory)
- *set_retention_control* (Mandatory)

Messages and Suggested Fix

The following message appears when the <pin-type> (sleep, save or restore) pin <pin-name> of instance <inst-name> of retention cell <cell-name> has value <actual-value> instead of the expected value <expected-value>:

[LPSVM59_1][WARNING] <pin-type> pin '<pin-name>' of retention cell '<inst-name>' (cellname '<cell-name>') has value '<actual-value>' instead of specified value '<expected-value>'

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

Since the control pins of the retention cell are improperly connected, the retention cell would not function properly.

How to Debug and Fix

The violation is reported at the instance of retention cell for which control pin (sleep, save or restore) does not attain the expected value.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix these violations, ensure that the control pins of retention cells are properly connected to the net of the associated power domain.

Example Code and/or Schematic

Consider the following example. A violation occurs when the pin 'D'of instance 'top.mid_inst.fd1_inst' of the retention cell 'FD1' attains a value of '0' instead of the expected value '1':

[WARNING] Restore pin 'D' of retention cell 'top.mid_inst.fd1_inst' (cellname 'FD1') has value '0' instead of specified value '1'

The schematic is as follows:

FIGURE 104. Incremental schematic

The schematic highlights the instance of the retention cell and the control pin that is attaining the expected value.

Default Severity Label

Warning

Rule Group

`State Retention`

Reports and Related Files

None

LPRET01

Checks if percentage of retention cells in sequential elements exceeds the set limit

When to Use

Run this rule to check if ratio of retention cells, with respect to normal flip-flops exceeds permissible limits. Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPRET01* rule checks if the ratio of retention cells with respect to the total number of sequential elements exceeds the limit set by `lp_max_retention_cell_percentage` parameter.

Language

Verilog, VHDL

Parameter(s)

- *lp_max_retention_cell_percentage*: Default value is `50`. Set the value of this parameter to any positive integer to report a violation if the ratio of retention cells with respect to sequential elements exceeds the set value.

Constraint(s)

UPF Commands

- *set_retention* (Mandatory)

Messages and Suggested Fix

Message 1

The following message is reported when the ratio of retention cells with

respect to the total number of sequential elements exceeds the limit set by the lp_max_retention_cell_percentage parameter:

[LPRET01_1][WARNING] Module <module-name> retention element ratio of <retention-flop-ratio>% (<number-of-retention-flops> retention flops of <number-of-flops> total flops) exceeds threshold limit of <ratio-limit>. Please refer to 'lp_retention_cell_list.rpt' file to see the list of retention cells found in the design

Message 2

The following message is reported when the ratio of retention cells with respect to the total number of sequential elements is within the limit set by the lp_max_retention_cell_percentage parameter:

[LPRET01_2][INFO] Retention element ratio for module <module-name> is <retention-flop-ratio>% (<number-of-retention-flops> retention flops of <number-of-flops> total flops). Please refer to 'lp_retention_cell_list.rpt' file to see the list of retention cells found in the design

Potential Issues

None.

Consequences of Not Fixing

None.

How to Debug and Fix

Refer to the lp_retention_cell_list.rpt report file for a list of retention cells.

Example Code and/or Schematic

Consider the following example, where the retention strategy is set in the UPF:

```
set_retention ret -domain VD1 -elements {inst1/ret1 inst2/ret2 ret3} -retention_power_net VDD -retention_ground_net VSS
```

In the above example, the following message is reported because the retention ratio of module top exceeds the threshold limit of 50.

```
[WARNING] Module 'top' retention element ratio of 75.000000% (3
retention flops of 4 total flops) exceeds threshold limit of
'50'. Please refer to ‘lp_retention_cell_list.rpt’ file to see
the list of retention cells found in the design
```

The following is the snippet from the report file:

```
################################################################
####################
# Purpose:
# This report contains count of retention cells found by
LPRET01
################################################################
####################

Total Number of Flops/Latches/Retention Cells : 4
Total Number of Retention Cells on which Retention Strategy
is Applied : 3
List of Retention Cells on which Retention Strategy is
Applied :
================================================================
=====
top.inst1.inst2.ret2
top.inst1.ret1
top.ret3
```

Default Severity Label

Warning/Info

Rule Group

State Retention

Reports and Related Files

The *lp_retention_cell_list* report lists the retention instances found in the design

LPRET02

Checks conflicting values of save / restore signals in retention strategies

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/ LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*
- This rule works only in the UPF format

Description

The *LPERT02* rule reports a violation if retention control signals are the same for multiple strategies while their polarities are different. There cannot be different polarities for the same save or restore signals in two different strategies.

This rule displays the conflicting values in a text viewer.

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *set_retention* (Mandatory)
- *set_retention_control* (Mandatory)

Messages and Suggested Fix

The following message appears when save or restore signals have conflicting values in different retention strategies:

[LPRET02_1][ERROR] Retention strategies have conflicting ‘<save

```
| restore>’ signal values for net '<signal-name>'
```

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

It is a design error when the same control signals of multiple retention strategies have different polarities.

How to Debug and Fix

To fix this issue, change the control signal’s value to be consistent across all retention strategies.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_domain VD1 -elements {inst1}
create_power_domain VD2 -elements {inst1/inst2}

set_retention ret -domain TOP -elements {reg1_reg}
retention_power_net VDD -retention_ground_net VSS
set_retention_control ret -domain TOP -save_signal {save
high} -restore_signal {res low}

set_retention ret1 -domain VD1 -retention_power_net VDD
retention_ground_net VSS
set_retention_control ret1 -domain VD1 -save_signal {save
low} -restore_signal {res low}
```

In the above example, the LPERT02 rule reports the following violation because there are two retention strategies and both strategies have the same save control signals while their polarities are different:

```
Retention strategies have conflicting 'save' signal values for
net 'top.save'
```

This conflicting values are displayed in a text viewer, as follows:

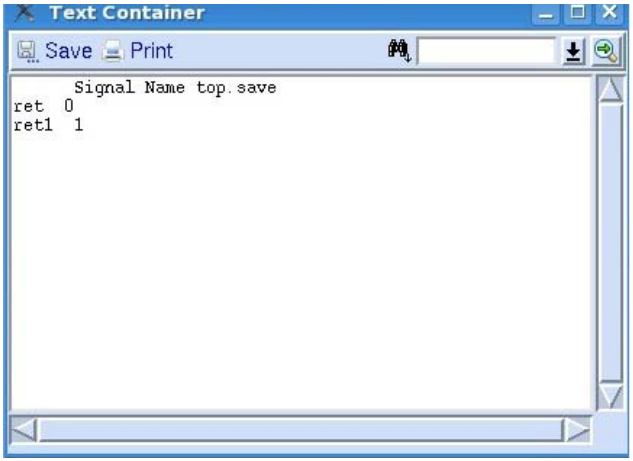


FIGURE 105. Text viewer

Default Severity Label

Error

Rule Group

State Retention

Reports and Related Files

None

LPRET03

Reports partially or fully redundant strategies

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*
- This rule works only in the UPF format

Description

The *LPRET03* rule checks for retention strategies that are redundant, partially or completely.

This rule has two sub-rules:

- *LPRET03A*: Reports if a retention strategy is applied on an always-on domain.
- *LPRET03B*: Reports if multiple strategies are applied on the same retention element.

LPRET03A

Reports if a retention strategy is applied on an always-on domain

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*
- This rule works only in the UPF format

Description

The *LPERT03A* rule reports a violation when a retention strategy is specified for an always-on domain. This is because an always-on domain does not need any retention strategy.

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *add_port_state* (Mandatory)
- *create_pst* (Mandatory)
- *add_pst_state* (Mandatory)
- *set_retention* (Mandatory)
- *set_retention_control* (Mandatory)

Messages and Suggested Fix

The following message appears when the retention strategy is specified to

an always-on domain:

[LPRET03A_1[ERROR] Retention strategy <Strategy-Name> is redundant for the AON domain <domain-name>

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

It is a design error when an always-on domain has a retention strategy.

How to Debug and Fix

To fix this issue, remove the retention strategy that was applied on the always-on domain.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_domain VD1 -elements {inst1}

set_retention ret -domain VD1 -elements {reg1_reg}
retention_power_net VDD -retention_ground_net VSS

set_retention_control ret -domain VD1 -save_signal {save
high} -restore_signal {res low}
```

In the above example, the LPERT03A rule reports the following violation because retention strategy `ret` is applied on the always-on domain `VD1`:

```
Retention strategy ret is redundant for the AON domain VD1
```

Default Severity Label

Error

Rule Group

State Retention

Reports and Related Files

None

LPRET03B

Reports if multiple retention strategies are applied on the same retention element

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*
- This rule works only in the UPF format

Description

The *LPERT03B* rule reports a violation when multiple retention strategies are specified for the same retention element. Even if there are multiple retention strategies specified on a retention element, only one strategy is applied and others strategies are ignored. The ignored retention strategies are reported in a text viewer.

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *set_retention* (Mandatory)
- *set_retention_control* (Mandatory)

Messages and Suggested Fix

The following message appears when multiple retention strategies are applied on the same retention element:

[LPRET03B_1[ERROR] Retention strategy on instance ‘<instance-

```
name>' will be ignored
```

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

It is a design error when multiple retention strategies are applied on the same element.

How to Debug and Fix

Double-click the message and check for redundant retention strategies, displayed on the text viewer. To fix this issue, remove the redundant strategies on the reported element.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_domain VD1 -elements {inst1}
create_power_domain VD2 -elements {inst1/a}

set_retention ret -domain VD1 -elements {inst1}
retention_power_net VDD -retention_ground_net VSS
set_retention_control ret -domain TOP -save_signal {save
high} -restore_signal {res low}

set_retention ret1 -domain VD2 -elements {inst1/reg1_reg}
retention_power_net VDD -retention_ground_net VSS
set_retention_control ret1 -domain VD2 -save_signal {save
high} -restore_signal {res low}
```

In the above example, the *LPERT03B* rule reports the following violation because multiple retention strategies are applied on retention element `reg1_reg`:

```
Retention strategy on instance 'reg1_reg' will be ignored
```

The redundant strategies are reported in the text viewer, as follows:

FIGURE 106. Text viewer

Default Severity Label

Error

Rule Group

State Retention

Reports and Related Files

None

LPRET04

Reports power domains with partially or completely missing retention strategies

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*
- This rule works only in the UPF format

Description

The *LPRET04* rule checks for power domains without retention strategies, since for switching domains a retention strategy should be specified.

This rule has two sub-rules:

- *LPRET04A*: Reports if a switching domain does not have a retention strategy.
- *LPRET04B*: Reports if a retention strategy is specified only for some elements in a switching domain.

LPRET04A

Reports if a switching domain does not have a retention strategy

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*
- This rule works only in the UPF format

Description

The *LPERT04A* rule reports a violation when a switching domain does not have a retention strategy. This is because if a domain is switching then it should have a retention strategy.

The rule checks only those domains that have retention cells.

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *add_port_state* (Mandatory)
- *create_pst* (Mandatory)
- *add_pst_state* (Mandatory)
- *set_retention* (Mandatory)
- *set_retention_control* (Mandatory)

Messages and Suggested Fix

The following message appears when retention strategy is not specified on a switching domain:

[LPRET04A_1[ERROR] Retention strategy is missing for the power domain <domain-name>

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

It is a design error if a switching domain does not have any retention strategy.

How to Debug and Fix

To fix this issue, specify a retention strategy for the switching domain.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_domain VD1 -elements {inst1}
```

In the above example, the *LPERT04A* rule reports the following violation because there are no retention strategies applied on switching domain VD1:

```
Retention strategy is missing for the power domain VD1
```

Default Severity Label

Error

Rule Group

State Retention

Reports and Related Files

None

LPRET04B

Reports if retention strategy is specified for fewer sequential cells in a switching domain than the recommended percentage

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*
- This rule works only in the UPF format

Description

The *LPRET04B* rule reports a violation if retention strategy is specified for fewer sequential cells in a switching domain than the recommended percentage. The percentage is recommended by the designer in line with the power budget. By default, the percentage is 50%, that is, half of the sequential elements are expected to be converted to retention cells and you should specify the retention elements.

NOTE: *This is a guideline rule and users should set the percentage as per their design methodology.*

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *set_retention* (Mandatory)
- *set_retention_control* (Mandatory)

Messages and Suggested Fix

The following message appears when retention strategy is specified only on

some elements of the switching domain:

[LPRET04B_1[ERROR] Retention strategy specified for switching power domain ‘<domain-name>’ is applied only on ‘<percentage-value>’ percentage of elements

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

It is a design error if all the elements in a switching domain does not have any retention strategy.

How to Debug and Fix

To fix this issue, specify a retention strategy for the all the elements in the switching domain.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_domain VD2 -elements {{inst1/reg2_reg inst1/
reg1_reg}
set_retention ret1 -domain VD2 -elements {inst1/reg1_reg} -
retention_power_net VDD -retention_ground_net VSS
```

In the above example, the *LPERT04B* rule reports the following violation because retention strategy specified for switching domain `VD2` is not applied on element `inst1/reg2_reg`, whereas it is applied on element `inst1/reg1_reg`:

```
Retention strategy specified for switching power domain 'VD2'
is applied only on '50.00'percentage of elements
```

Default Severity Label

Error

Rule Group

`State Retention`

Reports and Related Files

None

Connection Rules

The rules in this group are as follows:

Rule	Reports
LPCONN01	Output of Special cells that are hanging or tied to constant
LPCONN02	Checks domain anomalies
LPCONN03	Checks crossing paths where either source or destination or both are inout pin/port are homogeneous
LPCONN04	Checks control signals of retention cells, power switches, and isolation cells
LPCONN04A	If the source of the control signal is off when compared to the receiver
LPCONN04B	Checks the source of the control signal inputs and the destination of the acknowledge signal outputs
LPCONN04C	If the polarity of the control signal is not the same as its source
LPCONN04D	The continuity of the control signals across black boxes
LPCONN04E	Checks the connectivity of retention control signals
LPCONN05	Checks the supply relationship of source, destination and logic cells (between source and destination) on the clock/reset signal path
LPCONN05A	Checks if the primary clock/reset in the SGDC file is not connected to any clock/reset pin
LPCONN05B	Checks if the supply of primary source of clock/reset signal is relatively-more or equally-on than the supply of each of the logic cells in the path
LPCONN05C	Checks if the supply of a logic cell present in clock/reset path is relatively-more or equally-on than the supply of destination flip-flop
LPCONN06	Reports connections from the ON domain to input/inout pins/ ports of the OFF domain with the pass-gate attribute
LPCONN07	Checks for feedthrough paths

Rule	Reports
LPCONN07A	Checks if a path is a feedthrough path
LPCONN07B	Checks for set_related_supply_net on boundary ports in feedthrough paths
LPCONN08	Reports presence/absence of lockup latches between two flip-flops existing in different voltage domains but driven by synchronous clocks
LPCONN09	Report presence/absence of charged device model (CDM) cells in crossings
LPSVM49	Nets that are connected to pins having different supply names
LP_SPECIAL_PIN_CONNECTION	Generates report that shows special pin connections to ports

LPCONN01

Reports output or input of special cells that are hanging or tied to constant

When to Use

Use this rule for:

- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPCONN01* rule reports if the output or input pin of a Level shifter, Isolation, Retention, Power Switch and Always On Buffer cell in the design is hanging or tied to constant.

Rule Exceptions

This rule ignores SpyGlass generated buffers and inverters while checking output/input cone.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain (*Mandatory*:* Use this constraint to specify the Voltage/power domains in the design.
- *supply* (Mandatory): Use this constraint to specify the supply and ground port names for the LPPLIB rules.
- *pg_cell* (Optional): Use this constraint to specify the names of power/ground pins for cells present in the input netlist, which are missing from the respective PLIB/LIB/LEF libraries.
- *always_on_buffer* (Optional): Use this constraint to specify signals that should be driven by an always-on buffer.

- *power_switch* (Optional): Use this constraint to specify the power switches in a power domain.

CPF Commands

- *create_power_nets* (Mandatory)
- *create_ground_nets* (Mandatory)
- *create_power_domain* (Mandatory)
- *update_power_domain* (Mandatory)
- *define_always_on_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the input or output pin *<pin-name>* of the instance *<inst-name>* of *<sp-cell-type>* (Level shifter / Isolation / Retention /Always On Buffers / Power Switch) cell *<cell-name>* is hanging:

[LPCONN01_2][WARNING] <Input | Output> pin '<pin-name>' of <sp-cell-type> cell '<inst-name>(<cell-name>)' is unconnected

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the oinput or utput pin *<pin-name>* of the instance *<inst-name>* of *<sp-cell-type>* (Level Shifter / Isolation / Retention /Always On Buffers / Power Switch) cell *<cell-name>* is connected to constant:

[LPCONN01_1][WARNING] <Input | Output> pin '<pin-name>' of <sp-

```
cell-type> cell '<inst-name>(<cell-name>)' is connected to
constant
```

For debugging information, click *How to Debug and Fix*.

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

As the input/output pins are either unconnected or tied to a constant value, the cells containing them serve no purpose in the design.

How to Debug and Fix

The violation messages are reported at the cell instances with incorrectly connected power pins.

For a graphical view of the violation, double-click the message and then click **Incremental Schematic**. Refer to the Example Code and/or Schematic section for a sample.

To fix these violations, ensure that output or input pin of all instances of the special cell are properly connected. In addition, ensure the connection is not to a constant.

Example Code and/or Schematic

Example 1

Consider the following example. The following message appears when the output pin X of the retention cell RET_DEF is not connected:

```
Output pin 'X' of Retention cell
'top.mid_inst.bot_inst.fd1_inst(RET_DFF)' is unconnected
```

The schematic is as follows:

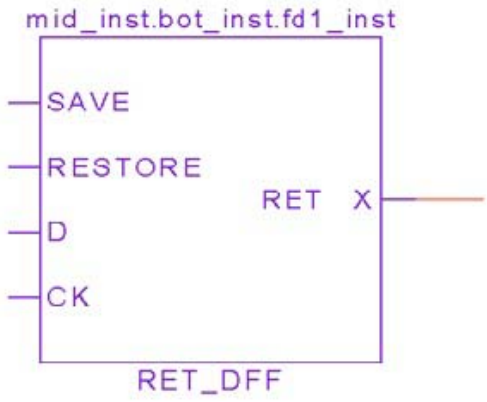


FIGURE 107. Incremental schematic

Example 2

Consider the following example. The following message appears when the output pin Z of the level shifter cell BASIC_LS is connected to constant:

```
Output pin 'Y' of Level Shifter cell
'top.INSTANCE_LS7(BASIC_LS)' is connected to constant
```

The schematic is as follows:

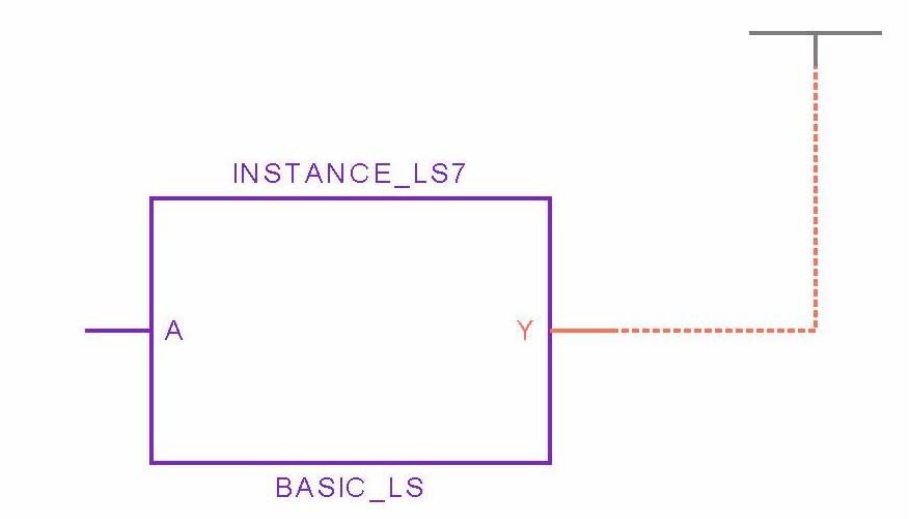


FIGURE 108. Incremental schematic

Example 3

Consider the following example. The following message appears when the input pin D of the isolation cell ISO_LATCH is not connected:

```
Input pin 'D' of Isolation cell
'top.mid_inst.bot_inst.iso1(ISO_LATCH_1)' is unconnected
```

The schematic is as follows:

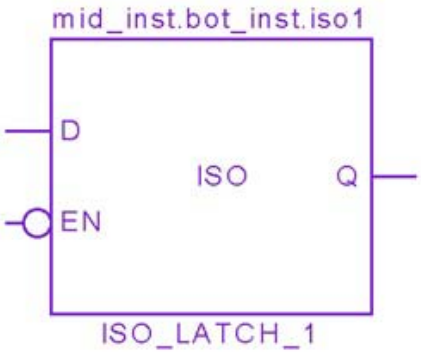


FIGURE 109. Incremental schematic

Example 4

Consider the following example. The following message appears when the input pin D of the isolation cell ISO_LATCH_1 is connected to constant:

```
Input pin 'D' of Isolation cell
'top.mid_inst.bot_inst.iso2(ISO_LATCH_1)' is connected to
constant
```

The schematic is as follows:

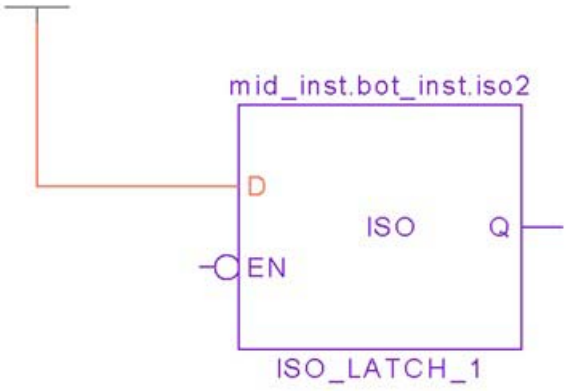


FIGURE 110. Incremental schematic

Default Severity Label

Warning

Rule Group

Connection

Reports and Related Files

None

LPCONN02

Reports domain anomalies

When to Use

This rule is applicable to all design phases.

This rule works only in the UPF format.

Description

The *LPCONN02* rule reports a message when there is a:

- **Domain feedthrough**: This is a direct path between input port & output port of the same domain.
- **Domain feedback**: This is a direct path between output port & input port of the same domain.
- **Disjoint domain**: This is a domain output from an instance in the domain that is feeding an input from another instance of the same domain.
- **Domain fork or tap**: This is a net that feeds through a domain with a branch that stops in the domain.

Rule Exceptions

The *LPCONN02* rule ignores buffers, AON buffers, SpyGlass generated buffers, and assign statements while checking the paths. In addition, this rule does not report a message when the design includes a macro with the pin powered by supplies other than the default supply for the domain.

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply*
- *create_supply_net* (Mandatory)

- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)

Messages and Suggested Fix

The following message appears:

[LPCONN02_1][INFO] <Type-of-path> path exists between port <port1> to port <port2> of domain <domain-name>.

Potential Issues

This message appears because your design has one of the following issues:

1. There is a feedthrough path between the domain input port to the domain output port.
2. There is a feedback path between the domain output port to the domain input port.
3. A domain output from an instance in the domain is feeding an input from another instance of the same domain.
4. There is a feedthrough of a domain with a tap or fork in that domain.

Consequences of Not Fixing

Unnecessary flow of data or control signals through the power domains may ask for additional power consideration to specified, which can be avoided. Such paths are specified in LPCONN02 rule. If an implementation tool inserts a buffer on the path going through less-on domain, it creates another crossing that needs to be taken care off by inserting the isolation cells.

How to Debug and Fix

You do not need to fix this message because it is an information message. However, you should review your design for any of the potential issues reported above. If the reported issue is not intentional, rectify the issue.

Example Code and/or Schematic

This section illustrates the instances when the *LPCONN02* rule reports a message.

Example 1

The following illustration displays a feedthrough from the domain input to the domain output port. Output from D1 flows through D2 and fed to D1.

D1 and D2 are siblings.

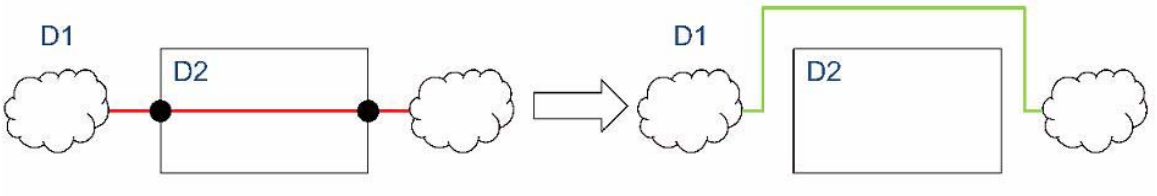


FIGURE 111. Feedthrough from the domain input to the domain output port

Example 2

The following illustration displays a feedback from the domain output port to the domain input port.

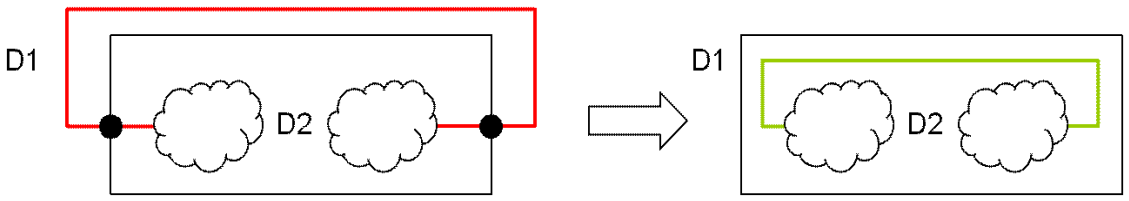


FIGURE 112. Feedback from the domain output port to the domain input port

Example 3

The following illustration displays a disjoined domain rule. D2 is the parent of D1 (two separate instances) where output of one D1 instance flows in D2 and fed to another instance of D1.

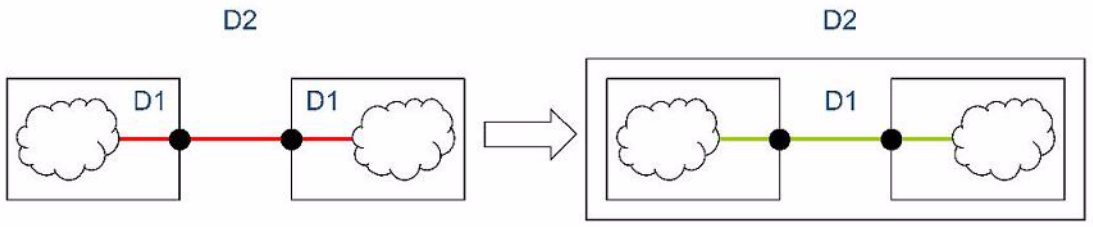


FIGURE 113. Disjoin domain

NOTE: *A new hierarchy needs to be created that is defined as D1 and all the D1 blocks are encapsulated in the new hierarchy.*

Example 4

The following illustration displays a domain fork.

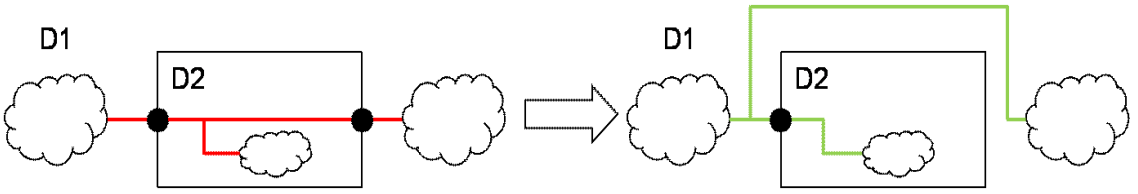


FIGURE 114. Domain fork

Default Severity Label

Info

Rule Group

Connection

Reports and Related Files

None

LPCONN03

Reports connections between inout pins/ports and other pins/ports that are not at the same voltage with compatible power states

When to Use

This rule is applicable to all design phases.

Description

The *LPCONN03* rule checks only those crossings where either source or destination or both are inout pin or ports. The purpose of this rule is to check homogeneity of these crossings. This means that source and destination supplies must have equivalent power states and the same voltage values. The supplies are considered to be equivalent when neither the output isolation nor the input isolation is needed between the source and destination.

Parameter(s)

- *lp_skip_buf*: Default value is `1`. Set this parameter to `0` to ignore buffers generated during synthesis.
- *lp_skip_pwr_gnd*: Default value is `1`. Set this value to `0` to consider nets connected to the power/ground supply.
- *lp_skip_aon_buf*: Default value is `1`. Set the parameter to `0` to not skip always-on buffers to find a crossing.
- *lp_ignore_input_isolation_for_inout*: Default value is `false`. Set this parameter to `true` to enable the rule ignore checking of input isolation (AON to gated crossing) for inout crossings.

Constraint(s)

SGDC

- *voltage_domain (*Mandatory*:* Use this constraint to specify the Voltage/power domains in the design.
- *supply* (Mandatory): Use this constraint to specify the supply and ground port names for the LPPLIB rules.

CPF Commands

- *create_power_nets* (Mandatory)
- *create_ground_nets* (Mandatory)
- *create_power_domain* (Mandatory)
- *update_power_domain* (Mandatory)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_port_attributes* (Optional)

  When set_port_attribute is used with `-driver_supply` or `-receiver_supply` or both, the LPCONN03 rule does report inout ports.

Messages and Suggested Fix

The following message appears when the crossing between the source <src> (domain <dom-name>) and destination <des-name> (domain <dom-name>) is not homogeneous:

[LPCONN03_1][ERROR] Crossing between source '<src>' (domain '<dom-name1>') and destination '<des-name>' (domain '<dom-name2>') is not homogeneous, <reason>

Where, *<reason>* can be:

- supplies do not have same voltage values and equivalent power states
- supplies do not have same voltage values
- supplies do not have equivalent power states

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

Since the crossing involves an inout pin that is an analog pin, an isolation cell or a level shifter cell cannot be inserted in the path. Hence, the end supplies have to be equivalent and must have the same voltage values.

How to Debug and Fix

The source and destination domains stated in the violation messages are highlighted in the Atrenta Console GUI.

For a graphical view of the violation, double-click the message and then click **Incremental Schematic**. Refer to the Example Code and/or Schematic section for a sample.

To fix these violations, review the supply values and power states. Ensure the supply values are the same and the power states are equivalent.

Example Code and/or Schematic

Consider the following example. `top.inst1.p1` is in power domain `PD1`. `top.inst2.p1` is in voltage domain `PD2`. The crossing is from `top.inst1.p1.P3` to `top.inst2.p1.P1` where both driver and receiver are of Inout type.

The following violation is reported for this case:

```
Crossing between source 'top.inst1.p1.P3' (domain 'PD1(supply
vddcore:0.950-1.100)') and destination 'top.inst2.p1.P1'
(domain 'PD2(supply vddcore1:1.300)') is not homogeneous,
supplies do not have same voltage values and equivalent power
states.
```

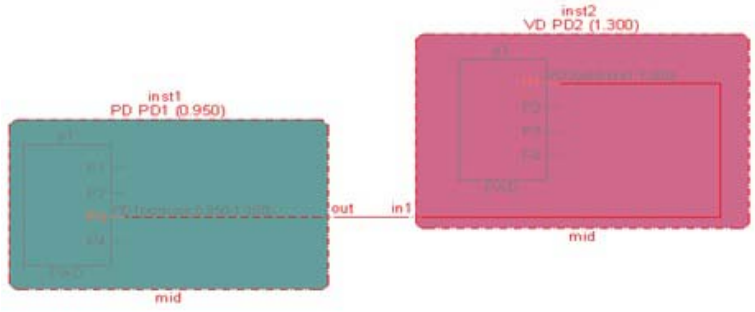


FIGURE 115. Incremental schematic

Default Severity Label

Info

Rule Group

Connection

Reports and Related Files

None

LPCONN04

Checks control signals of retention cells, power switches, and isolation cells

When to Use

Use this rule to check the control signals, such as retention, isolation and power switch enable, in accordance with the source.

This rule is applicable to all design phases.
This rule works only in the UPF format.

Description

The *LPCONN04* rule checks restore/save signals of retention cells and enable signals of power switches. In addition, it checks for isolation control signals of isolation cells. This rule has the following subrules:

- *LPCONN04A*: Checks if the source of the control signal is off when compared to the receiver
- *LPCONN04B*: Checks the source of the control signal inputs and the destination of the acknowledge signal outputs
- *LPCONN04C*: Checks if the polarity of the control signal is not the same as its source
- *LPCONN04D*: Checks the continuity of the control signals across black boxes.
- *LPCONN04E*: Checks the connectivity of retention control signals

LPCONN04A

Checks if the source of the control signal is off when compared to the receiver

When to Use

This rule is applicable to all design phases.

This rule works only in the UPF format.

Description

The *LPCONN04A* rule reports a violation when an output isolation is needed between the source and destination domain. This means that there is a state when the source is OFF and destination is ON.

Rule Exception

During traversal, this rule skips buffers and inverters, or combinational cells that are equivalent to buffers, for example an AND cell with one input tied to 1, and checks their state as per the destination.

Parameter(s)

- *lp_skip_buf:* Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.
- *lp_set_design_stage*: Default value is `rtl`. Set this parameter to `netlist` or `pg_netlist` to specify the desired design stage.

Constraint(s)

UPF Commands

- *create_power_switch* (Mandatory)
- *set_isolation* (Mandatory)
- *set_retention* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the control signal is driven by a less-

on source:

[LPCONN04A_1][ERROR] '<type>' signal '<sig-name>' of domain (domain '<domain-name>(<vdd_sink> :<vdd_sink_value>)') is driven by less-on source '<terminal/port-name>' (domain '<domain-name>(<vdd_sink> :<vdd_sink_value>)')

Where, <type> can be save, restore, power-switch enable or isolation.

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

It is a design error when the source is OFF when compared to the receiver domain.

How to Debug and Fix

For a graphical view of the violation, double-click the message and then click **Incremental Schematic**. Refer to the Example Code and/or Schematic section for a sample and an explanation of the schematic.

To fix this violation, review the UPF file and ensure that the control signal specified in the violation message is not driven by a less-on source.

Message 2

The following message appears when an isolation control signal is driven by a less-on source with respect to the actual destination of the isolation cell:

[LPCONN04A_2][ERROR] '<type>' signal '<sig-name>' of domain (domain '<domain-name>') is driven by less-on source '<source-name>' (domain '<source-domain-name>') with respect to the destination '<destination-name>' (domain '<destination-domain-name>')"

Where, <type> is isolation.

This violation is reported only for isolation signals at netlist and PG netlist design stage. This design stage is defined using the *lp_set_design_stage* parameter.

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

It is a design error when the source is OFF when compared to the receiver domain.

How to Debug and Fix

For a graphical view of the violation, double-click the message and then click **Incremental Schematic**. Refer to the Example Code and/or Schematic section for a sample and an explanation of the schematic.

To fix this violation, review the UPF file and ensure that the control signal specified in the violation message is not driven by a less-on source.

Message 3

The following message appears when a control signal is driven by multiple drivers:

[LPCONN04A_3][WARNING] <signal-type> signal '<signal-name>' of domain (domain '<domain-name>') has multiple drivers

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

It is a design issue when the control signal is driven by multiple drivers.

How to Debug and Fix

For a graphical view of the violation, double-click the message and then click **Incremental Schematic**. Refer to the Example Code and/or Schematic section for a sample and an explanation of the schematic.

To fix this violation, review the design file and ensure that the control signal specified in the violation message is not driven by multiple drivers.

Example Code and/or Schematic

Example 1

In this example, the `top.l4.d4` power-switch enable signal is driven by a less-on source, `top.in1`.

The following violation is reported for this case:

```
'Power-Switch enable' signal 'top.l4.d4' of domain (domain
'vd3(supply VDDB:1.100)') is driven by less-on source 'top.in1'
(domain 'vd1(supply VDD:1.100)')
```

The following is the schematic:

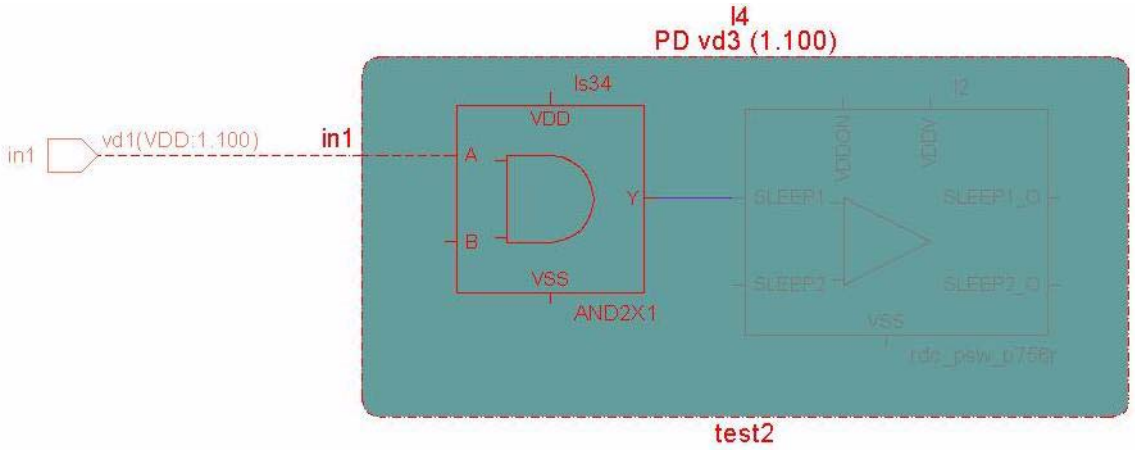


FIGURE 116. Incremental schematic

The schematic highlights the control signal that has caused this violation.

Example 2

In this example, the isolation enable of `TOP.EN2` signal is driven by a less-on source, `top.myflop.X`.

The following violation is reported for this case:

```
'Isolation' signal 'TOP.EN2' of domain (domain 'DOMAIN_B(supply
VDDH:1.200)') is driven by less-on source 'TOP.myflop.X'
(domain 'DOMAIN_C(supply VTOP:1.600)')
```

The following is the schematic:

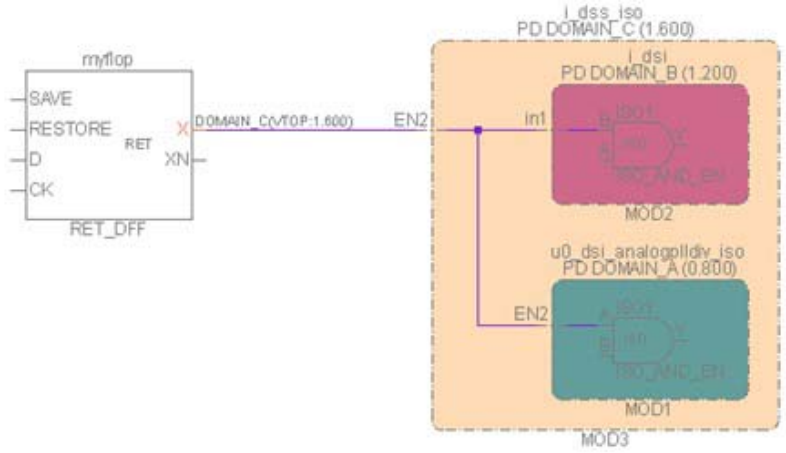


FIGURE 117. Incremental schematic

Example 3

In this example, the save enable of `top.save` signal is driven by a less-on source, `top.save`.

The following violation is reported for this case:

```
'Save' signal 'top.save' of domain (domain 'V2(supply
VDD_LP:1.200)') is driven by less-on source 'top.save' (domain
'V1(supply VDD:1.200)')
```

The following is the schematic:

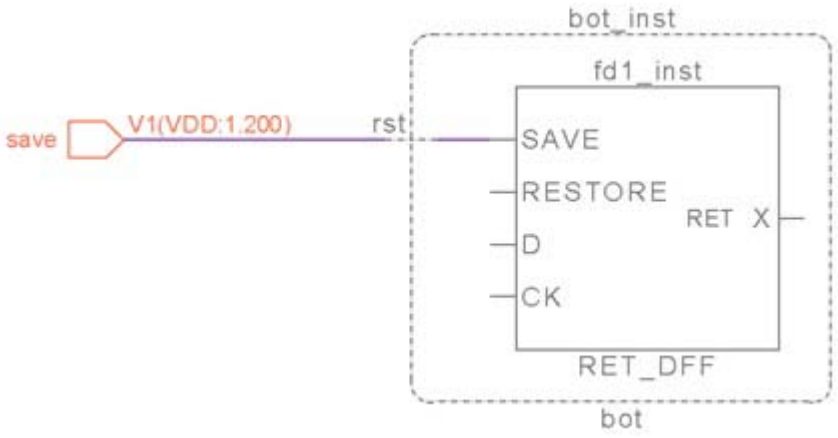


FIGURE 118. Incremental schematic

Example 4

Consider the following UPF commands:

```
set_isolation ISO1 -domain V2 -elements {mid_inst/op} -
clamp_value 0 -location parent \
                    -isolation_power_net VDD_LP1

set_isolation_control ISO1 -domain V2 -isolation_signal
{inst1/op} -isolation_sense high
```

In the above example, isolation cell `iso_inst` is driving a cell `an1`, which is in the domain `V1`. The isolation signal `top.inst1.op` is driven by the cell `top.inst1.op_reg_reg.Q`, which is in domain `V3`. Domain `V3` is less ON than the domain `V1`. So the following violation message is reported:

```
'Isolation' signal 'top.inst1.op' of domain (domain 'V2(supply
VDD_LP:1.200)') is driven by less-on source
'top.inst1.op_reg_reg.Q' (domain 'V3(supply VDD_LP1:1.200)')
with respect to the destination 'top.an1.A' (domain 'V1(supply
VDD:1.200)')
```

The following is the schematic:

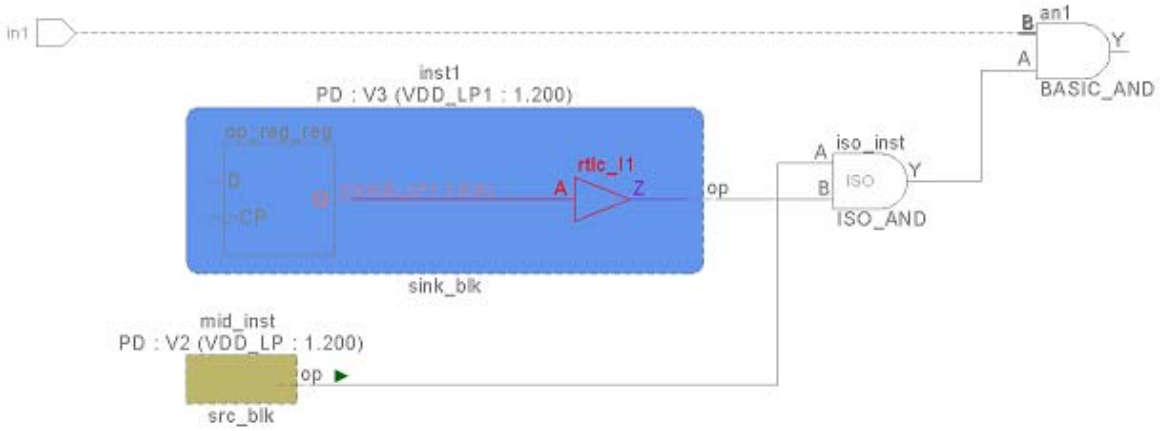


FIGURE 119. Incremental schematic

Example 5

Consider the following Verilog commands:

```
module top (in1,in2,en1,out1);
input in1,in2,en1;
output out1;

TEST1 inst1(.in1(in1),.out1(w1));
TEST1 inst2(.in1(in2),.out1(w1));

TEST2 inst3(.in1(w3),.en1(w1),.out1());

endmodule

module TEST1 (in1,out1);

input in1;
output out1;
AND and1(.A(in1),.Y(out1));
```

```
endmodule

module TEST2 (in1,en1,out1);

input in1,en1;
output out1;

ISO1 iso(.EN(en1),.D(in1),.Y(w1));
AND and1(.A(w1),.Y(out1));

endmodule
```

In this example, the isolation signal `top.w1` is driven by multiple drivers therefore the following violation is reported:

```
Isolation signal 'top.w1' of domain (domain 'PD3(supply
VDD3:1.000)') has multiple drivers
```

The following is the schematic:

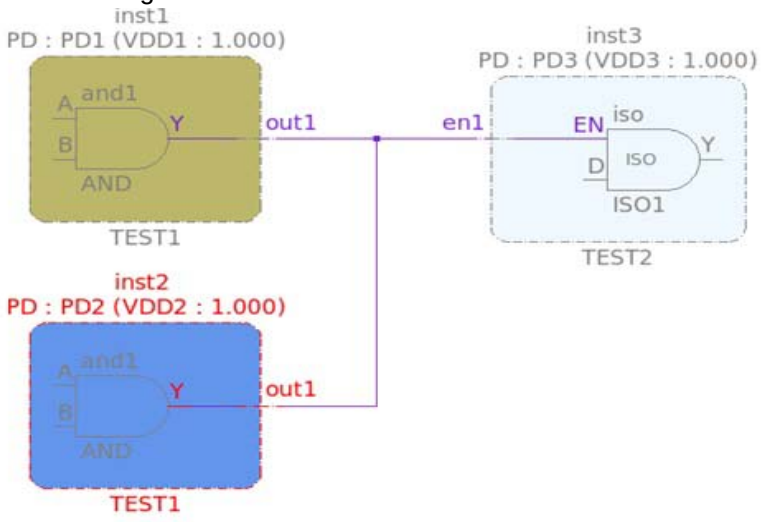


FIGURE 120. Incremental schematic

Default Severity Label

Error

Rule Group

Connection

Reports and Related Files

None

LPCONN04B

Checks the source of control signal and destination of power-switch acknowledge signal

When to Use

This rule is applicable to all design phases.

This rule works only in the UPF format.

Description

The *LPCONN04B* rule reports a violation message when source of control signal is not a steady state and is not driven by either flip-flop, top-level port, a latch, or a retention cell. This rule also reports a violation message when the receiver of power-switch acknowledge signal is not a flip-flop, top-level port, latch, retention cell, or control pin of power-switch.

Rule Exception

During traversal, this rule skips buffers and inverters, or combinational cells that are equivalent to buffers, for example an AND cell with one input tied to 1.

Parameter(s)

lp_set_design_stage: Default value is `rtl`. Set this parameter to `netlist` or `pg_netlist` to specify the desired design stage.

Constraint(s)

UPF Commands

- *create_power_switch* (Mandatory)
- *set_isolation* (Mandatory)
- *set_retention* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the control signal is not driven by a port, flip-flop output pin, latch, or retention cell:

[LPCONN04B_1][WARNING] '<type>' signal '<sig-name>' of domain (domain '<domain-name>(<vdd_sink> :<vdd_sink_value>)') is not driven by a top level port or flip-flop or latch or retention cell, it is driven by <pin-name> of Instance <instance-name> domain name <domain-name> (<vdd_sink> :<vdd_sink_value>)

Where, <type> can be save, restore, power-switch enable or isolation.

Message 2

The following message appears when the control signal is driven by a net:

[LPCONN04B_2][WARNING] '<type>' signal '<sig-name>' of domain (domain '<domain-name>') is not driven by a port or a flop output pin, it is driven by <power-net | ground-net | net> '<net-name>' domain name '<domain-name>'

Where, <type> can be save, restore, isolation, or power-switch enable.

Message 3

The following message appears when the acknowledge signal is not driving to a port, flip-flop input pin, latch, retention cell, or input control pin of power switch:

[LPCONN04B_3][WARNING] '<power-switch-acknowledge>' signal '<sig-name>' of domain (domain '<domain-name>(<vdd_sink> :<vdd_sink_value>)') is not driving a port or a flop or latch or power switch control pin, it is driving pin <pin-name> of instance <instance-name> (domain in <domain-name> (<vdd_sink> :<vdd_sink_value>)

Message 4

The following message appears in the RTL design when the acknowledge signal is not undriven:

[LPCONN04B_4][WARNING] '<power-switch-acknowledge>' signal '<sig-name>' of domain (domain '<domain-name>') is not undriven,it is driven by <pin/net/port> '<pin-name/net-name/port-name>' domain name '<domain-name>'

NOTE: *A design will be considered as an RTL design if the lp_set_design_stage parameter is set as rtl or netlist.*

Message 5

The following message appears in the PG netlist design when the

acknowledge signal is not driven by a power switch:

[LPCONN04B_5][WARNING] '<power-switch-acknowledge>' signal '<sig-name>' of domain (domain '<domain-name>') is not driven by power switch, it is driven by <pin/net/port> '<pin-name/net-name/port-name>' domain name '<domain-name>'

NOTE: *A design will be considered as PG netlist design if the lp_set_design_stage parameter is set as pg_netlist.*

Message 6

The following message appears in the PG netlist design when an odd number of inverters are found before reaching the power switch corresponding to the acknowledge signal. This is checked in the of backward (fan-in) traversal path of an acknowledge signal.

[LPCONN04B_6][WARNING] Path between power switch and power-switch acknowledge signal <sig-name> of domain (domain <domain-name>) is inverted due to inverter <inverter-name> (domain name <domain-name>

NOTE: *A design will be considered as PG netlist design if the lp_set_design_stage parameter is set as pg_netlist.*

Message 7

The following message appears in the PG netlist design when a buffer or an inverter with less-on supply is found before reaching the power switch corresponding to the acknowledge signal.
This is checked in the backward (fan-in) traversal path of an acknowledge signal.

[LPCONN04B_7][WARNING] Acknowledge signal '<sig-name>' of domain (domain '<domain-name>') is connected to a less on buffer or inverter '<inverter-name>' domain name '<domain-name>' before reaching to power switch

NOTE: *A design will be considered as PG netlist design if the lp_set_design_stage parameter is set as pg_netlist.*

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

It is a design error when the source of a control signal is not a port, a flip-flop output pin, transparent latch, or retention cell. Also, it is a design error

when the destination of an acknowledge signal is not a port, a flip-flop output pin, latch, retention cell, or input control pin of power switch.

How to Debug and Fix

For a graphical view of the violation, double-click the message and then click **Incremental Schematic**. Refer to the Example Code and/or Schematic section for a sample and an explanation of the schematic.

To fix this violation, review the UPF file and ensure that the control signal specified in the violation message is driven by a port or flip-flop output pin.

Message 8

The following message appears at netlist design stage, if the design driver of control signals (defined as logic net in UPF) does not match with the driver defined in UPF:

[LPCONN04B_8][ERROR] <signal-type> signal <signal-name> of domain (domain <domain-name>) is not driven by a port <port-name> defined as driver in UPF, it is driven by <design-driver>

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

If you do not fix this violation, the protection devices might not function correctly because the connectivity of the control signals are not correct.

How to Debug and Fix

For a graphical view of the violation, double-click the message and then click **Incremental Schematic**. Refer to the Example Code and/or Schematic section for a sample and an explanation of the schematic.

To fix this violation, review the UPF file and ensure that the control signal specified in the violation message is driven by the correct logic.

Message 9

The following message appears at netlist design stage, if the control signals (defined as logic net in UPF) is undriven in design:

[LPCONN04B_9][ERROR] <signal-type> signal <signal-name> of domain (domain <domain-name>) is not driven by a port <port-name> defined as driver in UPF, it is undriven.

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

If you do not fix this violation, the protection devices might not function correctly because the connectivity of the control signals are not correct.

How to Debug and Fix

For a graphical view of the violation, double-click the message and then click **Incremental Schematic**. Refer to the Example Code and/or Schematic section for a sample and an explanation of the schematic.

To fix this violation, review the UPF file and ensure that the control signal specified in the violation message is driven by the correct logic.

Example Code and/or Schematic

Example 1

In this example, the `top.l4.d4` power-switch enable signal is not driven by a top-level port or a flip-flop.

The following violation is reported for this case:

```
'Power-Switch enable' signal 'top.l4.d4' of domain (domain
'vd3(supply VDDB:1.100)') is not driven by a port or a flip-
flop output pin
```

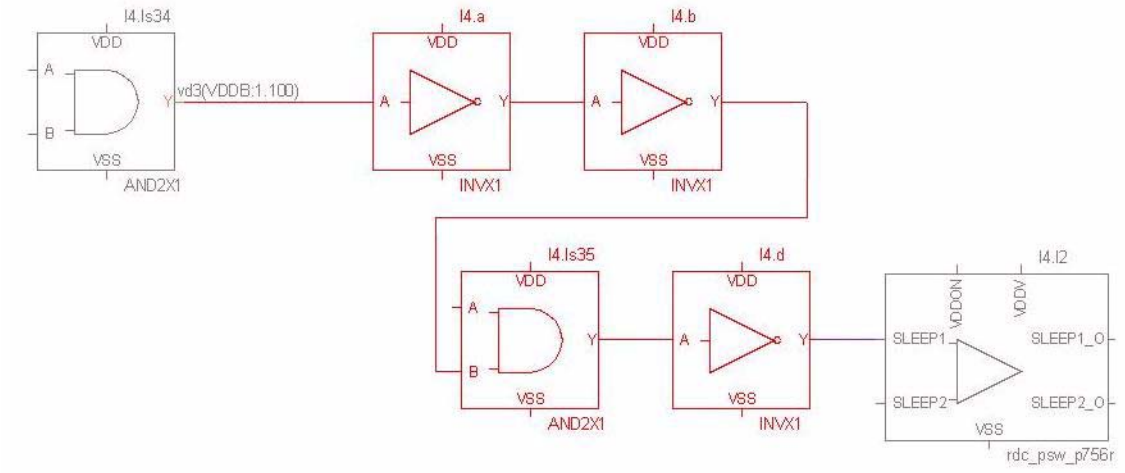


FIGURE 121. Incremental schematic

The schematic highlights the path to the source of the control signal. In this schematic, signal driven by an AND gate.

Example 2

In this example, the signal `top.mid_inst.iso` is not driven by a port, flip-flop output pin, latch, or retention cell.

```
'Isolation' signal 'top.mid_inst.iso' of domain (domain
'V2(supply VDD_LP:1.200)') is not driven by a port or a flop
output pin, it is driven by 'top.and2.Y' of Instance 'top.and2'
domain name 'V1(supply VDD:1.200)'
```

The following is the schematic:

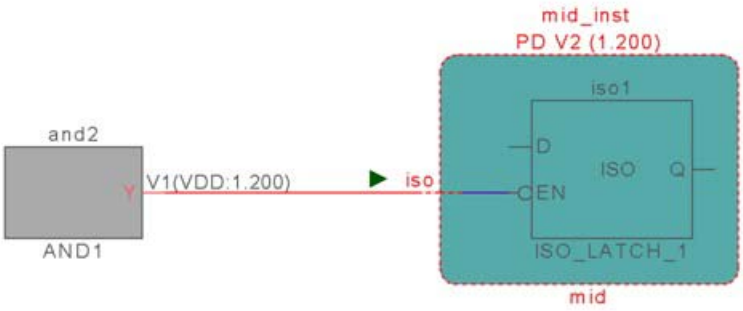


FIGURE 122. Incremental schematic

Example 3

In this example, the signal `top.bot_inst.save` is not driven by a port, or flip-flop output pin:

```
'Save' signal 'top.bot_inst.save' of domain (domain 'V2(supply
VDD_LP:1.200)') is not driven by a port or a flop output pin,
it is driven by 'top.bot_inst.save' domain name 'V1(supply
VDD:1.200)'
```

The following is the schematic:

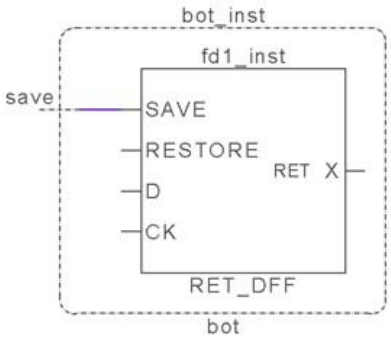


FIGURE 123. Incremental schematic

Example 4

In this example, the signal `top.a` is not driving to a port, flip-flop input pin, latch, retention cell, or input control pin of power switch:

```
'Power-Switch acknowledge 'signal 'top.a' of domain 'VD0(supply
VDD:1.000)' is not driving a port or flop or latch or power
switch control pin, it is driving pin 'top.inst1.and1.A' of
instance 'top.inst1.and1'(domain'VD1(supply VDDX:1.000)')
```

The following is the schematic:

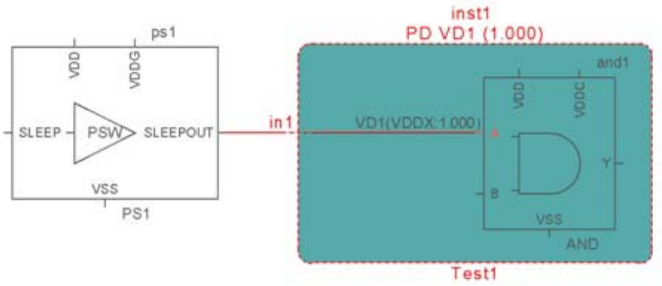


FIGURE 124. Incremental schematic

Example 5

In this example, the signal `top.a` is not undriven:

```
'Power-Switch acknowledge' signal 'top.a' of domain (domain
'VD0(supply VDD:1.000)') is not undriven, it is driven by pin
'top.bf1.Y' domain name 'VD0(supply VDD:1.000)'
```

The following is the schematic:

FIGURE 125. Incremental schematic

Example 6

In the diagram, below, an acknowledge signal is defined as `mid_inst/bot_inst/w1`. There is an inverter `'top.mid_inst.bot_inst.inv1'` between the power switch output and the defined acknowledge signal. This inverter is inverting the acknowledge signal, so the following violation is being reported in this case:

```
Path between power switch and 'Power-Switch acknowledge' signal
'top.mid_inst.bot_inst.w1' of domain (domain 'V2(supply
VDD_LP:1.200)') is inverted due to inverter
'top.mid_inst.bot_inst.inv1' (domain name 'V2(supply
VDD_LP:1.200)')
```

The following is the schematic:

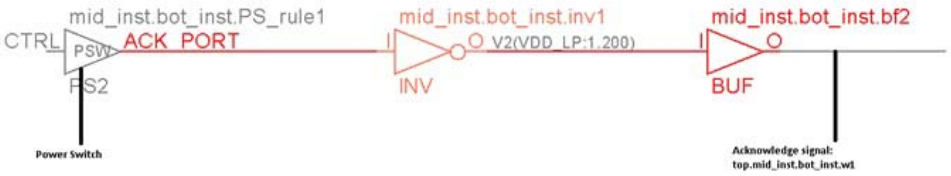


FIGURE 126. Incremental schematic

Example 7

In the diagram, below, an acknowledge signal for domain V2 with supply VDD_LP is defined as `mid_inst/bot_inst/w1`. In the path between power switch output and the defined acknowledge signal, there is a buffer whose supply (VDD) is less-on than the supply of power switch. In this case, the following violation is reported:

```
'Power-Switch acknowledge' signal 'top.mid_inst.bot_inst.w1' of
domain (domain 'V2(supply VDD_LP:1.200)') is connected to a
less on buffer/inverter 'top.mid_inst.bot_inst.bf2' domain name
'V2(supply VDD:1.200)'
```

The following is the schematic:

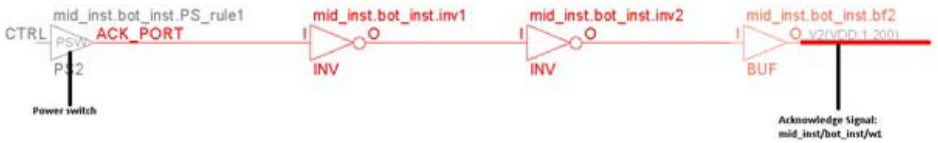


FIGURE 127. Incremental schematic

Default Severity Label

Warning

Rule Group

Connection

Reports and Related Files

None

LPCONN04C

Checks if the polarity of the control signal is not the same as its source

When to Use

This rule is applicable to all design phases.

This rule works only in the UPF format.

Description

The *LPCONN04C* rule reports a violation when the polarity of the signal changes between the signal, that is defined in the power intent file, and its source.

During traversal, this rule skips buffers and inverters, or combinational cells that are equivalent to buffers, for example an AND cell with one input tied to 1, and reports a violation message if there are an odd number of inverters present in the path.

Parameter(s)

lp_consider_inverted_output: Default value is `no`. Set this parameter to `yes` to enable the LPCONN04C rule consider inverted outputs NAND, NOR, XNOR cells as inverter plus logic.

Constraint(s)

UPF Commands

- *create_power_switch* (Mandatory)
- *set_isolation* (Mandatory)
- *set_retention* (Mandatory)

Messages and Suggested Fix

The following message appears when the polarity of the control signal is not the same as the source:

[LPCONN04C_1][WARNING] Polarity of <type> signal <signal-name> of domain <domain-name>[<vdd_sink> :<vdd_sink_value>] is not same as its source <terminal/port name> (domain-name [<vdd_source> :<vdd_source_value>])

Where, <type> can be save, restore, power-switch enable or isolation.

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

It is a design error when the polarity of the control signal is not the same as its source.

How to Debug and Fix

For a graphical view of the violation, double-click the message and then click **Incremental Schematic**. Refer to the Example Code and/or Schematic section for a sample and an explanation of the schematic.

To fix this violation, review the design file and ensure that the polarity of the control signal is the same as its source.

Example Code and/or Schematic

Example 1

In this example, the polarity of the control signal `top.l4.d4` is not the same as its source, `top.l4.ls34.Y`.

The following violation is reported for this case:

```
[WARNING] Polarity of 'Power-Switch enable' signal 'top.l4.d4'
of domain (domain 'vd3(supply VDDB:1.100)') is not same as its
source 'top.l4.ls34.Y' (domain 'vd3(supply VDDB:1.100)')
```

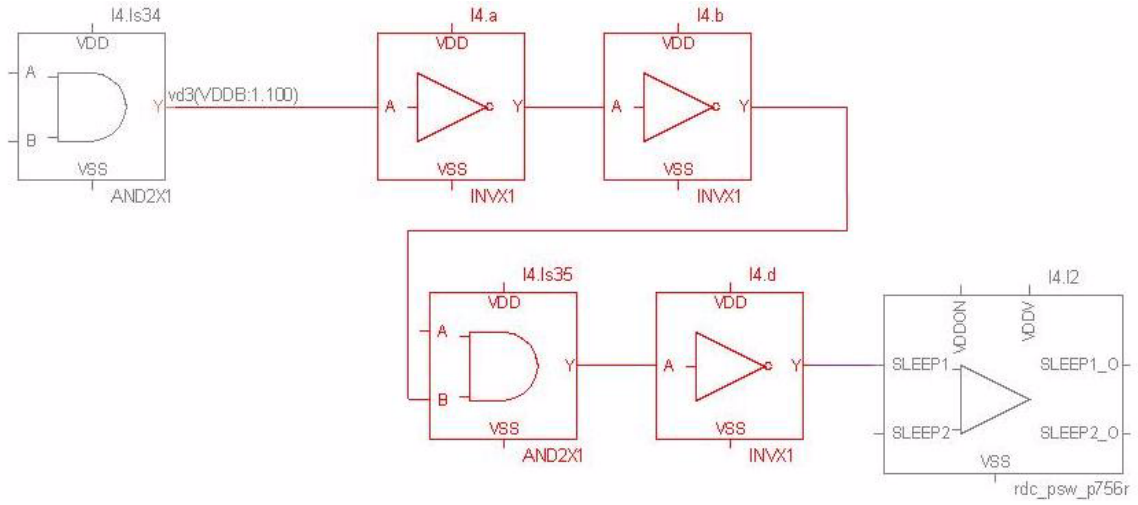


FIGURE 128. Incremental schematic

The schematic highlights the path to the source of the control signal. In this schematic, the path has an odd number (3) of inverters.

Example 2

In this example, the polarity of the isolation signal `top.a1` is not the same as its source, `top.l4.ls34.Y`

The following violation is reported for this case:

```
Polarity of 'Isolation' signal 'top.a1' of domain (domain
'VD1(supply VDDX:1.000)') is not same as its source
'top.and1.Y' (domain 'VD0(supply VDD:1.000)')
```

The following is the schematic:

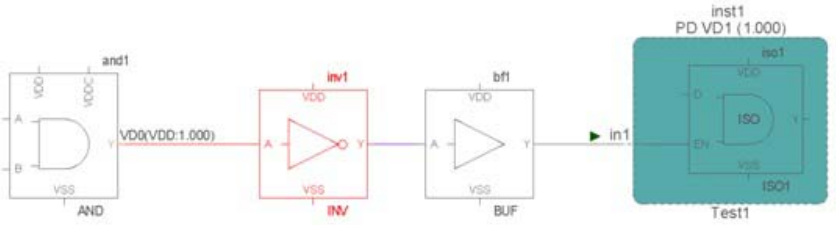


FIGURE 129. Incremental schematic

Example 3

In this example, the polarity of the save signal `top.a1` is not the same as its source, `top.and1.Y`.

The following violation is reported for this case:

```
Polarity of 'Save' signal 'top.a1' of domain (domain
'VD0(supply VDD:1.000)') is not same as its source 'top.and1.Y'
(domain 'VD0(supply VDD:1.000)')
```

The following is the schematic:

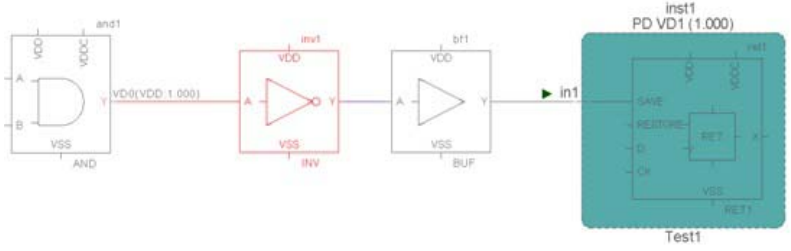


FIGURE 130. Incremental schematic

Default Severity Label

Warning

Rule Group

Connection

Reports and Related Files

None

LPCONN04D

Checks the continuity of the control signals across black boxes and buffers

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*
- This rule works only in the UPF format.

Description

The *LPCONN04D* rule checks for the continuity of the control signals across black boxes.

Rule Exceptions

This rule skips buffers, including combinational cells equivalent to buffers, and inverters. For example, the rule skips an AND cell with one input tied to 1.

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_power_switch* (Mandatory)
- *set_isolation* (Mandatory)
- *set_retention* (Mandatory)

SGDC

- *assume_path* (Optional): Use to specify the paths that exist between the input pins and the output pins of black boxes.

- *set_power_info* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the polarity of the control signal is not the same as the source:

[LPCONN04D_1][ERROR] Incorrect connection of <signal-type> signal <sig-name> of domain <domain-name> to blackbox <inst-name> (cell: <cell-name>): Input pin <pin-name> does not have attribute <attribute-name>

For debugging information, refer to *How to Debug and Fix*.

Message 2

The following message appears when an *assume_path* specification is missing:

[LPCONN04D_2][ERROR] Missing assume_path specification for input pin <pin-name> with attribute <attribute-name> of blackbox cell <cell-name>

For debugging information, refer to *How to Debug and Fix*.

Message 3

The following message appears when an related output pin does not have a required attribute:

[LPCONN04D_3][ERROR] Incorrect assume_path connection for input pin <pin-name> with attribute <attribute-name> of blackbox cell <cell-name>: Related output pin <pin-name> does not have attribute <attribute-name>

For debugging information, refer to *How to Debug and Fix*.

Message 4

The following message appears when an input pin is incorrectly connected:

[LPCONN04D_4][ERROR] Input pin <pin-name> with attribute <attribute-name> of blackbox <inst-name> (cell: <cell-name>) is incorrectly connected to net <net-name> which is not a control signal

For debugging information, refer to *How to Debug and Fix*.

Message 5

The following message appears when an input pin is incorrectly driven:

```
[LPCONN04D_5][ERROR] Input pin <pin-name> with attribute <attribute-name> of blackbox <inst-name> (cell: <cell-name>) is incorrectly driven by pin <pin-name> with attribute <attribute-name> of blackbox <inst-name> (cell: <cell-name>)
```

For debugging information, refer to *How to Debug and Fix*.

Message 6

The following message is reported if a less-on buffer is found in the path of a control signal. The less-on comparison is done with the supply of the destination node.

NOTE: *The less-on checking is only done for a buffer in the path where the control signal type matches with the destination node type.*

```
[LPCONN04D_6][ERROR] Instance <inst-name>(cell:<cell-name>) present in domain '<domain-name>' found in the path of control signal '<signal-name>' is off with respect to destination '<destination-name>'('<destination-domain-name>')
```

For debugging information, refer to *How to Debug and Fix*.

Message 7

The following message is reported if an isolation/retention/power-switch enable signal is blocked at the input pin of a black box instance and no *set_power_info* specification has been provided for that pin:

```
[LPCONN04D_7][ERROR] <type> <signal-name> blocked at input pin <pin-name> of black-box instance <instance-name>(cell:<cell-name>). Please provide set_power_info specification for this cell
```

Where, <type> can be save, restore, power-switch enable or isolation.

For debugging information, refer to *How to Debug and Fix*.

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

This is a design flaw and needs to be corrected.

How to Debug and Fix

The violations are reported at the place where the source signal of the crossing is used in the design file.

For a graphical view of the violation, double-click the message and then click **Incremental Schematic**. Refer to the Example Code and/or Schematic section for a sample and an explanation of the schematic.

To fix these violations, perform the following:

- **Message 1**: For the input pin, specify the attribute stated in the violation message.
- **Message 2**: Provide the *assume_path* specification for the input pin stated in the violation message.
- **Message 3**: Ensure that input and output pins specified in *assume_path* have corresponding attributes.
- **Message 4**: Ensure that the net connected to the pin is a control signal.
- **Message 5**: Ensure that an input pin is driven by an output pin of the corresponding attribute.
- **Message 6**: Ensure that buffer in the path of a control signal is more-on with respect to the domain for which the control signal is defined.
- **Message 7**: Ensure that *set_power_info* specifications are provided for the corresponding input pin.

Example Code and/or Schematic

This example illustrates when Message 1, 2, 3, 4, 5, 6, and 7 appear. The attributes specification is shown in the SGDC file. Review the SGDC file and the design file to understand the violating cases.

UPF Snippet

```
set_isolation iso1 -domain VD0 -isolation_power_net VDD
set_isolation_control iso1 -domain VD0 -isolation_signal
iso_en_in

create_power_switch psw1 -domain VD1 -input_supply_port
{VDDG VDD1} -output_supply_port {VDD VDDX} -control_port
```

```
{SLEEP pwr_en_in}

set_retention ret1 -domain VD0 -retention_power_net VDD
save_signal {ret_en_in high} -restore_signal {ret_en_in low}
```

The LPCONN04D rule reports the following violations and generates the corresponding schematic, as shown.

Message 1

SGDC Constraints

```
set_power_info -cell Bbox1 -pin iso_in -attribute
isolation_enable_in -value true
set_power_info -cell Bbox1 -pin ret_in -attribute
retention_enable_in -value true
set_power_info -cell Bbox1 -pin pwr_in -attribute
power_switch_enable_in -value true

set_power_info -cell Bbox1 -pin iso_out -attribute
isolation_enable_out -value true
set_power_info -cell Bbox1 -pin ret_out -attribute
retention_enable_out -value true
set_power_info -cell Bbox1 -pin pwr_out -attribute
power_switch_enable_out -value true
```

For the above example, the following message is reported:

```
Incorrect connection of retention enable signal 'top.ret_en_in'
of domain 'VD0' to blackbox 'top.bb1' (cell: 'Bbox1'): Input
pin 'ret_in' does not have attribute 'retention_enable_in'
```

The schematic generated is:

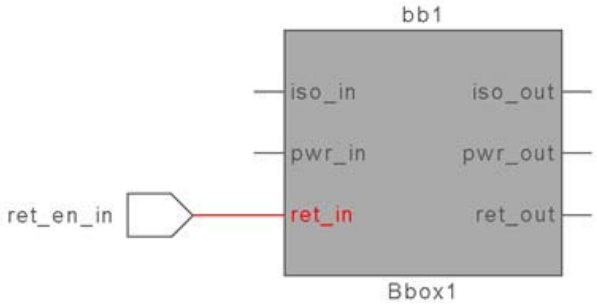


FIGURE 131. Incremental schematic

To resolve the violation, for the input pin `ret_in`, specify the `isolation_enable_pin` attribute stated in the violation message.

Message 2

SGDC Constraints

```
set_power_info -cell Bbox1 -pin iso_in -attribute
isolation_enable_in -value true
set_power_info -cell Bbox1 -pin ret_in -attribute
retention_enable_in -value true
set_power_info -cell Bbox1 -pin pwr_in -attribute
power_switch_enable_in -value true

set_power_info -cell Bbox1 -pin iso_out -attribute
isolation_enable_out -value true
set_power_info -cell Bbox1 -pin ret_out -attribute
retention_enable_out -value true
set_power_info -cell Bbox1 -pin pwr_out -attribute
power_switch_enable_out -value true
```

For the above example, the following message is reported:

```
[ERROR] Missing assume_path specification for input pin
'iso_in' with attribute 'isolation_enable_in' of blackbox cell
'Bbox1'
```

The schematic generated is:

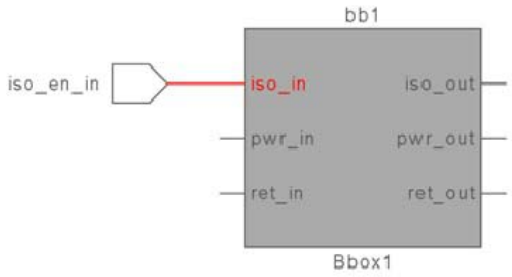


FIGURE 132. Incremental schematic

To resolve this violation, provide the *assume_path* specification for the `iso_in` input pin stated in the violation message.

Message 3

SGDC Constraints

```
set_power_info -cell Bbox2 -pin iso_in -attribute
isolation_enable_in -value true
set_power_info -cell Bbox2 -pin ret_in -attribute
retention_enable_in -value true
set_power_info -cell Bbox2 -pin pwr_in -attribute
isolation_enable_in -value true
set_power_info -cell Bbox2 -pin pwr_in -attribute
power_switch_enable_in -value true

assume_path -name Bbox2 -input iso_in -output iso_o
assume_path -name Bbox2 -input ret_in -output ret_o
assume_path -name Bbox2 -input pwr_in -output pwr_out
```

For the above example, the following message is reported:

```
Incorrect assume_path connection for input pin 'pwr_in' with
attribute 'isolation_enable_in'of blackbox cell 'Bbox2' :
Related output pin 'pwr_out' does not have attribute
'isolation_enable_out'
```

The following schematic is generated.

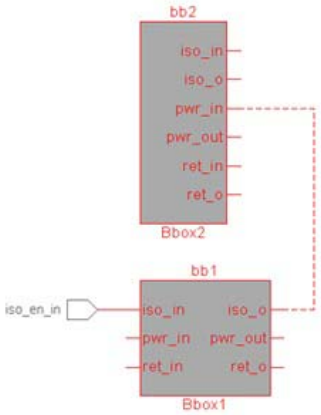


FIGURE 133. Incremental schematic

To resolve this violation, ensure that input `ret_in` and output `pwr_out` pins specified in *assume_path* have corresponding attributes.

Message 4

SGDC Constraints

```
set_power_info -cell Bbox1 -input_pin iso_in -output_pin
so_out -attribute isolation_enable
set_power_info -cell Bbox1 -input_pin pwr_in -output_pin
pwr_out -attribute power_switch_enable
set_power_info -cell Bbox1 -input_pin ret_in -output_pin
ret_out -attribute retention_enable
```

For the above example, the following message is reported:

```
Input pin 'iso_in' with attribute 'isolation_enable_in' of
blackbox 'top.bb1' (cell: 'Bbox1') is incorrectly connected to
net 'top.iso_en_in' which is not a control signal
```

The following schematic is generated.

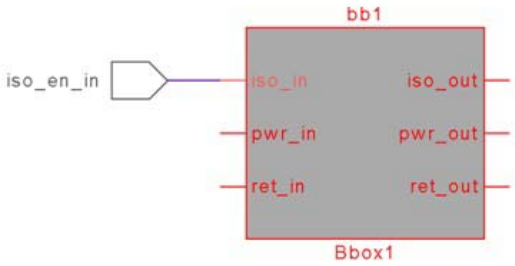


FIGURE 134. Incremental schematic

To resolve this violation, ensure that the net `top.iso_en_in` connected to the `iso_in` pin is a control signal.

Message 5

SGDC Constraints

```
set_power_info -cell Bbox1 -pin iso_out -attribute
isolation_enable_out -value true
set_power_info -cell Bbox1 -pin ret_out -attribute
retention_enable_out -value true
set_power_info -cell Bbox1 -pin pwr_out -attribute
power_switch_enable_out -value true

set_power_info -cell Bbox2 -input_pin iso_in -output_pin
iso_out -attribute isolation_enable
set_power_info -cell Bbox2 -input_pin pwr_in -output_pin
pwr_out -attribute power_switch_enable
set_power_info -cell Bbox2 -input_pin ret_in -output_pin
ret_out -attribute retention_enable
```

For the above example, the following message is reported:

```
Input pin 'pwr_in' with attribute 'power_switch_enable_in' of
blackbox 'top.bb2' (cell: 'Bbox2') is incorrectly driven by pin
'iso_out' with attribute 'isolation_enable_out' of blackbox
'top.bb1' (cell: 'Bbox1')
```

The following schematic is generated.

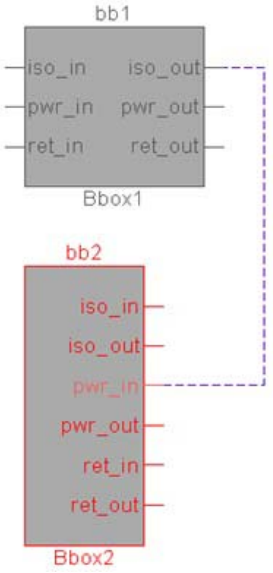


FIGURE 135. Incremental schematic

Message 6

Consider the following UPF snippet:

```
set_retention ret1 -domain TOP -retention_power_net VDD -
save_signal {ret_en_out high} -restore_signal {ret_en_in
low}
```

For the above example, the *LPCONN04D* rule reports the following violation message:

```
Instance 'top.buf_ret2'(cell:'BUF') present in domain
'TOP(supply VDD:0.700)' found in the path of control signal
'top.ret_en_out' is off with respect to destination
'top.inst3.ret1.SAVE'('PD3(supply VDD3:1.100)')
```

The following schematic is generated.

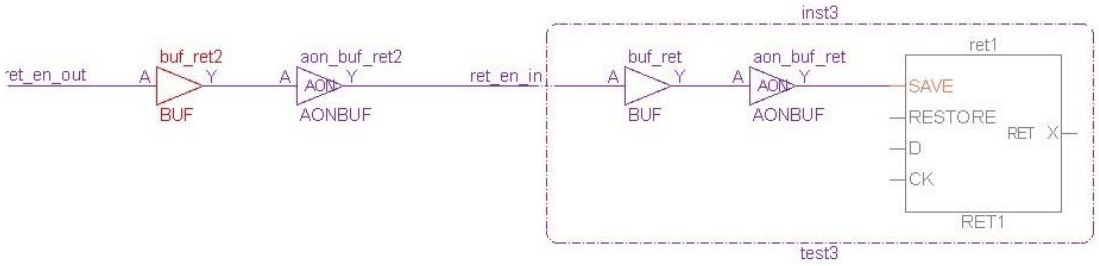


FIGURE 136. Incremental schematic

Message 7

SGDC Constraints

```
set_power_info -cell Bbox1 -input_pin iso_in -output_pin
iso_out -attribute isolation_enable
set_power_info -cell Bbox1 -input_pin pwr_in -output_pin
pwr_out -attribute power_switch_enable
set_power_info -cell Bbox1 -input_pin ret_in -output_pin
ret_out -attribute retention_enable
```

For the above example, the following message is reported:

```
Isolation signal 'top.iso_en_in' blocked at input pin 'iso_in'
of black-box instance 'top.bb2'(cell:'Bbox2').Please provide
set_power_info specification for this cell
```

The following schematic is generated.

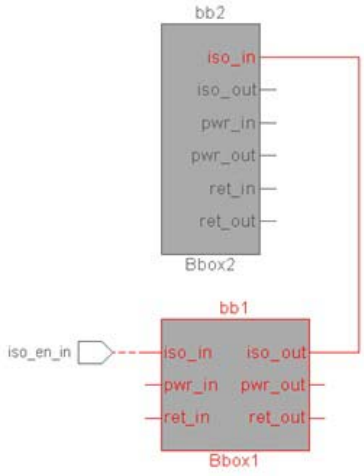


FIGURE 137. Incremental schematic

Default Severity Label

Error

Rule Group

Connection

Reports and Related Files

None

LPCONN04E

Checks the connectivity of retention control signals

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*
- This rule works only in the UPF format.

Description

The *LPCONN04E* rule reports a violation message when:

- Save or restore signals are hanging
- Save or restore signals are connected to wrong cells
- Save or restore signals are connected to ports
- Save or restore signals are connected to wrong pins
- Save or restore signals are connected to tied values

Parameter(s)

lp_skip_buf: Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.

Constraint(s)

UPF Commands

- *set_retention* (Mandatory)
- *set_retention_control* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when save or restore signals are hanging

instead of being connected to the retention cell:

[LPCONN04E_1][ERROR] ‘<save | restore>’ signal ‘<signal-name>’ defined for domain (‘<domain-name>’) in strategy (‘<strategy-name>’) is unconnected

Message 2

The following message appears when save or restore signals are connected to wrong cells:

[LPCONN04E_2][ERROR] ‘<save | restore>’ signal ‘<signal-name>’ defined for domain (‘<domain-name>’) in strategy (‘<strategy-name>’) is connected to incorrect instance (‘<instance-name>’) : cell (‘<cell-name>’) rather than a retention cell

Message 3

The following message appears when save or restore signals are connected to a port:

[LPCONN04E_3][ERROR] ‘<save | restore>’ signal ‘<signal-name>’ of domain (‘<domain-name>’) is connected to Port‘<port-name>’ rather than retention cell

Message 4

The Following message appears when save or restore signals are connected to the correct cell but wrong pins:

[LPCONN04E_4][ERROR] ‘<save | restore>’ signal ‘<signal-name>’ defined for domain (‘<domain-name>’) in strategy (‘<strategy-name>’) is connected to incorrect pin (‘<pin-name>’) of retention instance (‘<instance-name>’) : cell ‘<cell-name>’

Message 5

The following message appears when save or restore signals are connected to tied values (source side of signals):

[LPCONN04E_5][ERROR] ‘<save | restore>’ signal ‘<signal-name>’ defined for domain (‘<domain-name>’) in strategy (‘<strategy-name>’) is tied to constant value

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

It is a design error when control signals of retention strategy are connected

in wrong way.

How to Debug and Fix

For a graphical view of the violation, double-click the message and then click **Incremental Schematic**. Refer to the *Example Code and/or Schematic* section for a sample and an explanation of the schematic.

To fix this violation, review the UPF file and HDL file then ensure that the control signal specified in the violation message is connected correctly.

Example Code and/or Schematic

Example 1

Consider the following code:

UPF

```
create_power_domain V2 -elements {fd_inst}
set_retention RET1 -domain V2
set_retention_control RET1 -domain V2 -save_signal {in1 high}
-restore_signal {rst low }
map_retention_cell RET1 -domain V2 -lib_cells RET_DFF
```

Verilog

```
RET_DFF fd_inst(.D(),.SAVE(),.RESTORE(),.CK(vdd),.X(w3));
```

In the above example, the *LPCONN04E* rule reports the following message because the save signal `top.in1` is hanging:

```
'Save' signal 'top.in1' defined for domain ('V2') in strategy
'RET1' is unconnected
```

The following schematic is generated:

FIGURE 138. Incremental schematic

Example 2

Consider the following code:

UPF

```
create_power_domain V2 -elements {fd_inst}
et_retention RET1 -domain V2
set_retention_control RET1 -domain V2 -save_signal {in1 high}
-restore_signal {rst low }
map_retention_cell RET1 -domain V2 -lib_cells RET_DFF
```

Verilog

```
FD1P fd_inst(.D(in1),.CP(rst),.Q(o1),.QN());
RET_DFF
fd1_inst(.D(in1),.SAVE(),.RESTORE(rst),.CK(vdd),.X(w3));
```

In the above example, the *LPCONN04E* rule reports the following message because the save signal is connected to a wrong cell `fd_inst` instead of a retention cell:

```
'Save' signal 'top.in1' defined for domain ('V2') in strategy
'RET1' is connected to incorrect instance 'top.fd_inst'  : cell
'FD1P' rather than a retention cell
```

The following schematic is generated:

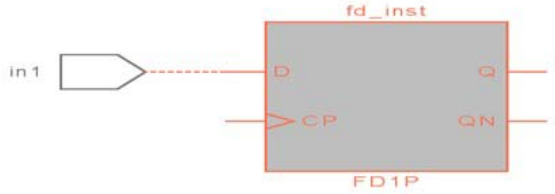


FIGURE 139. Incremental schematic

Example 3

Consider the following code:

UPF

```
create_power_domain V2 -elements {fd_inst}
```

```
set_retention RET1 -domain V2
set_retention_control RET1 -domain V2 -save_signal {w1 high}
-restore_signal {rst low }
map_retention_cell RET1 -domain V2 -lib_cells RET_DFF
```

Verilog

```
assign o1 = w1;
RET_DFF fd_inst(.D(),.SAVE(),.RESTORE(),.CK(),.X());
```

In the above example, the *LPCONN04E* rule reports the following message because the save signal is connected to the port `o1`:

```
'Save' signal 'top.w1' of domain ('V2') is connected to Port
'o1' rather than a retention cell
```

The following schematic is generated:

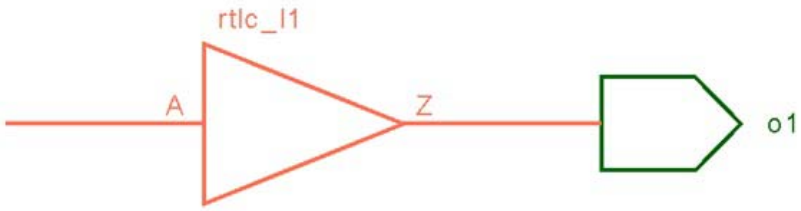


FIGURE 140. Incremental schematic

Example 4

Consider the following code:

UPF

```
create_power_domain V2 -elements {fd_inst}
set_retention RET1 -domain V2
set_retention_control RET1 -domain V2 -save_signal {in1 high}
-restore_signal {rst low }
map_retention_cell RET1 -domain V2 -lib_cells RET_DFF
```

Verilog

```
RET_DFF
fd_inst(.D(),.SAVE(rst),.RESTORE(in1),.CK(vdd),.X(w3));
```

In the above example, the *LPCONN04E* rule reports the following message

because the save signal `top.in1` is connected to the retention cell `fd_inst` but wrong pin `RESTORE`:

```
'Save' signal 'top.in1' defined for domain ('V2') in strategy
'RET1' is connected to incorrect pin 'RESTORE' of retention
instance 'top.fd_inst' : cell RET_DFF
```

The following schematic is generated:

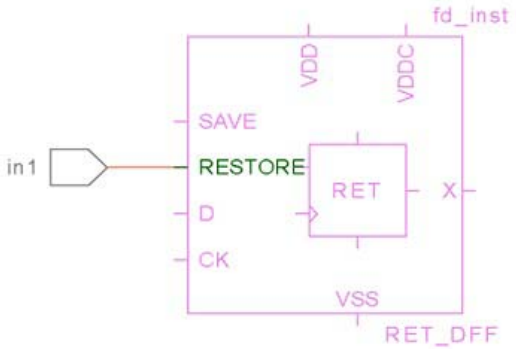


FIGURE 141. Incremental schematic

Example 5

Consider the following code:

UPF

```
create_power_domain V2 -elements {fd_inst}
set_retention RET1 -domain V2
set_retention_control RET1 -domain V2 -save_signal {w1 high}
-restore_signal {rst low }
map_retention_cell RET1 -domain V2 -lib_cells RET_DFF
```

Verilog

```
and and_gate(w1,in1,1'b0);
```

```
RET_DFF
fd_inst(.D(in1),.SAVE(w1),.RESTORE(rst),.CK(vdd),.X(w3));
```

In the above example, the *LPCONN04E* rule reports the following message because the save signal `top.w1` (source side) is connected to the constant value:

```
Save' signal 'top.w1' defined for domain ('V2') in strategy
'RET1' is tied to constant value
```

The following schematic is generated:

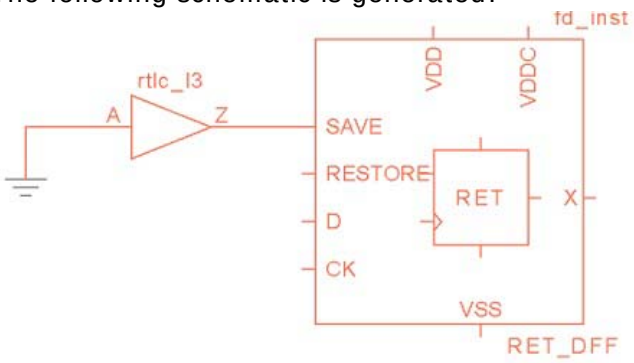


FIGURE 142. Incremental schematic

Default Severity Label

Error

Rule Group

Connection

Reports and Related Files

None

LPCONN05

Checks the supply relationship of source, destination and logic cells (between source and destination) on the clock/reset signal path

When to Use

Use this rule to check the supply is in accordance with the supply of logic cells or destination flip-flop.

This rule is applicable to all design phases and works only in the UPF format.

Description

The *LPCONN05* rule checks that the supply of primary clock source is relatively more- or equally-on than the supply of logic cells present along the clock path. This rule also checks that the supply of logic cells is relatively more- or equally-on than the clock destination. The same checks are also performed on the reset signal path.

This rule has three sub-rules:

- *LPCONN05A*: Checks if the primary clock/reset in the SGDC file is not connected to any clock/reset pin.
- *LPCONN05B*: Checks if the supply of primary source of clock/reset signal is relatively-more or equally-on than the supply of each of the logic cells in the path.
- *LPCONN05C*: Checks if the supply of a logic cell present in clock/reset path is relatively-more or equally-on than the supply of destination flip-flop.

LPCONN05A

Checks if the primary clock / reset in the SGDC file is not connected to any clock / reset pin

When to Use

This rule is applicable to all design phases.

This rule works only in the UPF format.

Description

The *LPCONN05A* rule reports a violation if the primary *clock*/*reset* source defined in the SGDC file is not connected to any clock/reset pin in design.

Rule Exceptions

To identify source and destination of a clock/reset signal, this rule skips/passes through following cells:

- RTLC (SpyGlass generated) buffers
- RTLC (SpyGlass generated) inverters
- Always on buffers/inverters
- Isolation cells
- Level shifter cells
- Other combinational cells

Parameter(s)

- None

Constraint(s)

SGDC

- *reset* (Mandatory)
- *clock* (Mandatory)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)

- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)

Messages and Suggested Fix

The following message appears:

[LPCONN05A_1][ERROR] <Clock | Reset> '<hierarchical-name>' is not connected to any <clock | reset> pin in design

Potential Issues

All the clocks/resets defined in SGDC file should be connected to at least one clock/reset pin in design.

Consequences of Not Fixing

If the clock source specified is not connected to any clock pin, *LPCONN05B*/*LPCONN05C* do not perform any supply relationship checks along the clock/reset path.

How to Debug and Fix

To fix this violation, ensure all clocks/resets defined in the SGDC file are connected to at least one clock/reset pin in design.

Example Code and/or Schematic

This example illustrates how to define primary clock/reset pins in the SGDC file by using the *clock*/*reset* constraints:

```
clock -name TOP.clk
reset -name TOP.reset


Clock 'TOP.clk' is not connected to any clock pin in design
Clock 'TOP.reset' is not connected to any clock pin in design
```

Default Severity Label

Error

Rule Group

Connection

Reports and Related Files

None

LPCONN05B

Checks if the supply of primary source of clock/reset signal is relatively-more or equally-on than the supply of each of the logic cells in the path

When to Use

This rule is applicable to all design phases.

This rule works only in the UPF format.

Description

The *LPCONN05B* rule reports a violation if the supply of primary source of the clock/reset signal is relatively more or equally-on than the supply of any of the logic cells in clock/reset path. The path is between primary clock source and destination.

Rule Exceptions

To identify source and destination of a clock/reset signal, this rule skips/passes through following cells:

- RTLC (SpyGlass generated) buffers
- RTLC (SpyGlass generated) inverters
- Always on buffers/inverters
- Isolation cells
- Level shifter cells
- Other combinational cells

Parameter(s)

- *lp_use_inferred_clocks*: Default is `no`. Set the value to `yes` to use the auto-generated clock information in addition to any user-defined clocks, which are specified using the *clock* constraint in the SGDC file.
- *lp_use_inferred_resets*: Default is `no`. Set the value to `yes` to use the auto-generated reset information in addition to any user-defined reset(s), which are specified using the *reset* constraint in the SGDC file.

Constraint(s)

SGDC

- *reset* (Optional)
- *clock* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)

Messages and Suggested Fix

The following message appears:

[LPCONN05B_1][ERROR] Cell '<cell-hierarchical-name>' (<domain-name>, <supply-name>) connected to '<destination-hierarchical-name>' (<domain-name>, <supply-name>) is relatively more on than the <clock | reset> source '<source-hierarchical-name>' (<domain-name>, <supply-name>)

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

This is a design error.

How to Debug and Fix

For a graphical view of the violation, double-click the message and then click **Incremental Schematic**. The schematic displays the complete path from the source to the destination. Refer to the *Example Code and/or Schematic* section for a sample and an explanation of the schematic.

To fix this violation, review the UPF file and ensure that the supply of primary source of clock/reset signal is relatively-more or equally-on than the supply of each of the logic cells in the path

Example Code and/or Schematic

This example illustrates when the *LPCONN05B* rule reports a violation. Suppose, you have defined a clock source in the SGDC file:

```
clock -name top.clk_generator.out_clk
```

In the actual design, `out_clk` is connected to the clock pin of a flip-flop through a series of buffers:

```
out_clk --> buf1 --> FLOP
```

The supply of `out_clk` should be relatively more or equally on than the supplies of `buf1`. Otherwise, this rule reports a the following violation:

```
Cell 'TOP.buf1' ('TOP(supply vddape:1.000)') connected to
destination 'TOP.u_VD.Q_reg.CP' ('VD_domain(supply
VDD3:1.000)') is relatively more on than the clock source
'TOP.clock_generator_inst.out_clk' ('TOP(supply VDD0:1.000)')
```

The following is the schematic for this example:

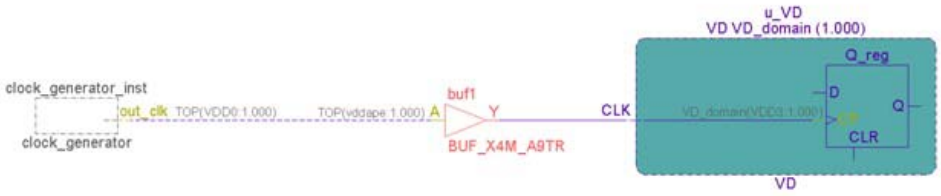


FIGURE 143. Incremental schematic

Default Severity Label

Error

Rule Group

`Connection`

Reports and Related Files

None

LPCONN05C

Checks if the supply of a logic cell present in clock/reset path is relatively-more or equally-on than the supply of destination flip-flop

When to Use

This rule is applicable to all design phases.

This rule works only in the UPF format.

Description

The *LPCONN05C* rule reports a violation if supply of logic cells present in clock/reset path, which is between the primary clock source and destination, is relatively less-on than the supply of the destination flip-flop.

Rule Exceptions

To identify source and destination of a clock/reset signal, this rule skips/passes through following cells:

- RTLC (SpyGlass generated) buffers
- RTLC (SpyGlass generated) inverters
- Always on buffers/inverters
- Isolation cells
- Level shifter cells
- Other combinational cells

Parameter(s)

- *lp_use_inferred_clocks*: Default is `no`. Set the value to `yes` to use the auto-generated clock information in addition to any user-defined clocks, which are specified using the *clock* constraint in the SGDC file.
- *lp_use_inferred_resets*: Default is `no`. Set the value to `yes` to use the auto-generated reset information in addition to any user-defined reset(s), which are specified using the *reset* constraint in the SGDC file.

Constraint(s)

SGDC

- *reset* (Optional)
- *clock* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)

Messages and Suggested Fix

The following message appears:

[LPCONN05C_1][ERROR] Cell '<cell-hierarchical-name>' (<domain-name>, <supply-name>) coming from <clock | reset> source '<source-hierarchical-name>' (<domain-name>, <supply-name>) is relatively less-on than the destination '<destination-hierarchical-name>' (<domain-name>, <supply-name>)

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

This is a design error.

How to Debug and Fix

For a graphical view of the violation, double-click the message and then click **Incremental Schematic**. The schematic displays the complete path from the source to the destination. Refer to the *Example Code and/or Schematic* section for a sample and an explanation of the schematic.

To fix this violation, review the UPF file and ensure the supply of a logic cell present in clock/reset path is relatively-more or equally-on than the supply of destination flip-flop.

Example Code and/or Schematic

This example illustrates when the *LPCONN05C* rule reports a violation. Suppose, you have defined a clock source in the SGDC file:

```
clock -name top.clk_generator.out_clk
```

In the actual design, `out_clk` is connected to the clock pin of a flip-flop through a buffer (or series of buffers):

```
out_clk --> buf1 --> FLOP
```

The supply of `buf1` should be relatively more or equally on than the supply of clock pin of `FLOP`. Otherwise, this rule reports the following violation:

```
Cell 'TOP.buf1' ('TOP(supply vddape:1.000)') coming from clock
source 'TOP.clock_generator_inst.out_clk' ('TOP(supply
VDD0:1.000)') is relatively less-on than the destination
'TOP.u_VD.Q_reg.CP' ('VD_domain(supply VDD3:1.000)')
```

The following is the schematic for this example:

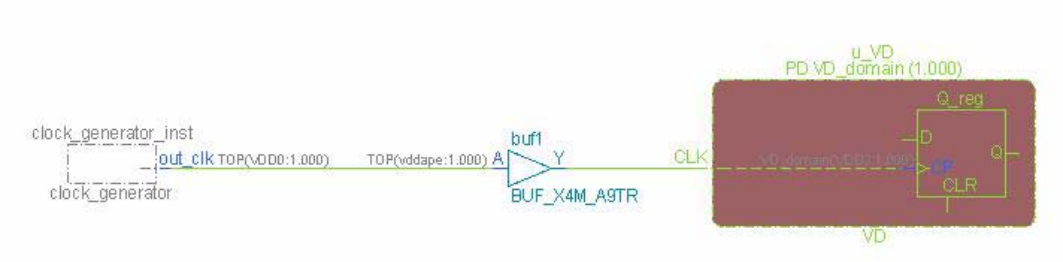


FIGURE 144. Incremental schematic

Default Severity Label

Error

Rule Group

`Connection`

Reports and Related Files

None

LPCONN06

Reports connections from the ON domain to input/inout pins/ports of the OFF domain with the pass-gate attribute

When to Use

- Use this rule to avoid the leakage path.
- This rule is applicable to all design phases.
- This rule works only in the UPF format.

Description

The *LPCONN06* rule checks those crossings where source logic is coming from an ON domain and destination is an input/inout pins/ports of OFF domain with `has_pass_gate` attribute. This rule checks each input/inout pins, which has the `has_pass_gate` attribute, of cell instances placed in the OFF domain and reports an error when its source is found in an ON domain.

Prerequisites

The *LPCONN06* rule supports the `has_pass_gate` user defined attribute. Set this attribute using the *set_power_info* SGDC command. For example:

```
set_power_info -cell macro -pin in1 -attribute has_pass_gate
-value true
```

Rule Exceptions

The *LPCONN06* rule does not report a violation if the source is an isolation cell or the source is in the same/relatively-off domain with reference to the destination domain.

Parameter(s)

- *lp_skip_buf:* Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.
- *lp_check_pass_gate_on_pwr_gnd:* Default value is `0`. Set this parameter to `1` to enable *LPCONN06* rule to check crossings involving supply nets (1'b1/1'b0 and UPF supply nets).

Constraint(s)

SGDC

- *set_power_info* (Mandatory)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)

Messages and Suggested Fix

The following message appears:

[LPCONN06_1][ERROR] Crossing between ON source '<source-name>' (domain '<source-domain>(supply <source-supply>)') and OFF destination '<destination-name>' (domain '<dest-domain>(supply <dest-supply>)') has attribute 'has_pass_gate' at destination pin '<pin-name>' (cell '<cell-name>')

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

This is a design error. The pass gate at the destination would lead to the leakage path between ON domain and OFF domain

How to Debug and Fix

For a graphical view of the violation, double-click the message and then click **Incremental Schematic**. The schematic displays the complete path from the source to the destination. Refer to the *Example Code and/or Schematic* section for a sample and an explanation of the schematic.

To fix this violation, add a buffer cell in the destination (OFF) domain or add an isolation cell to isolate the logic path.

Example Code and/or Schematic

Example 1

This example illustrates when the *LPCONN06* rule reports a violation. In the following schematic, the source domain VD0 is relatively ON with reference to the destination domain VD1 and the A pin of the BB1 cell is constrained with the has_pass_gate attribute.

```
Crossing between ON source 'top.rtlc_I2.Z' (domain 'VD0(supply
VDD:0.900)') and OFF destination 'top.inst1.bb.A' (domain
'VD1(supply VDD1:1.200)') has attribute 'has_pass_gate' at
destination pin 'A' (cell 'BB1')
```

The following is the schematic:

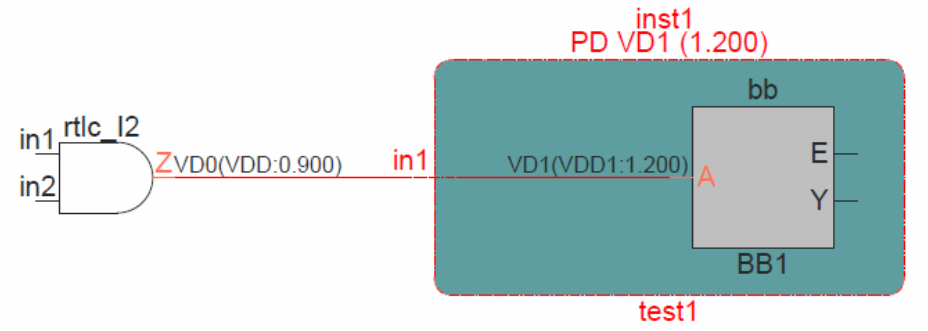


FIGURE 145. Incremental schematic

Example 2

This example illustrates a violation reported by the *LPCONN06* rule when the *lp_check_pass_gate_on_pwr_gnd* parameter is set. In the following schematic, the source domain VD0 is relatively ON with reference to the destination domain VD1 and the A pin of the BB1 cell is constrained with the has_pass_gate attribute. Here, the supply top.VDD is the source.

```
Crossing between ON source 'top.VDD' (domain 'VD0(supply
VDD:0.900)') and OFF destination 'top.inst1.bb.A' (domain
'VD1(supply VDD1:1.200)') has attribute 'has_pass_gate' at
destination pin 'A' (cell 'BB1')
```

The following is the schematic:

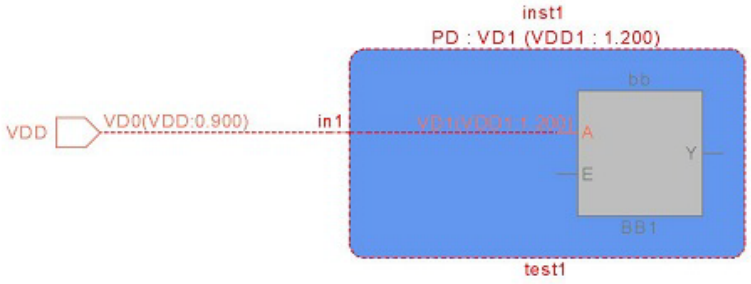


FIGURE 146. Incremental schematic

Default Severity Label

Error

Rule Group

Connection

Reports and Related Files

None

LPCONN07

Checks for feedthrough paths

When to Use

- Use this rule to report feedthrough paths in a design.
- This rule is applicable to all design phases.
- This rule works only in the UPF format

Description

The *LPCONN07* rule checks for feedthrough paths or such paths having set_related_supply_net specified.

This rule has two sub-rules:

- *LPCONN07A*: Checks if both source and destination are present in the same domain.
- *LPCONN07B*: Checks that the net associated with boundary ports on feedthrough paths, using *set_related_supply_net* or defined in parent domain, is more on than the source and destination of the feedthrough path.

LPCONN07A

Checks if a path is a feedthrough path

When to Use

Use this rule to report feedthrough paths in a design.

Description

The *LPCONN07A* rule checks if both source and destination are operating on the same supply. If yes, then the path is reported as a feedthrough path.

Prerequisites

None

Rule Exceptions

None

Parameter(s)

- *lp_check_domain_equivalence*: Default value is `0` and the rule checks if the domains are equal based on their primary supplies. Set this parameter to `1` to check buffers that are existing in the same domain while checking for feed through paths.
- *lp_check_ls_strategy_presence*: Default value is `0`. Set this parameter to `1` to check for the presence of level shifter strategy on any of the ports on the domain boundary from source to destination.

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_related_supply_net* (Optional)

Messages and Suggested Fix

The following message appears when SpyGlass finds a feedthrough path in the design:

[LPCONN07A_1][INFO] Crossing from '<source_name>' (domain '<source_domain>,<source-supply>') to '<destination_name>' (domain '<destination_domain>,<destination-supply>') is a feed through path

Potential Issues

Not applicable.

Consequences of Not Fixing

Not applicable.

How to Debug and Fix

Review both source and destination ports mentioned in the violation message.

Example Code and/or Schematic

Consider the following example:

```
module top(in1,in2,out1,out2);
input in1,in2;
output out1,out2;
wire w1,w2;
mid inst1(in1,in2,w1,w2);
mid inst4(w1,w2,out1,out2);
endmodule

module mid(in1,in2,out1,out2);
input in1,in2;
output out1,out2;
mid1 inst2(in1,out1);
mid1 inst3(in2,out2);
endmodule

module mid1(in1,out1);
input in1;
```

```
output out1;
assign out1=in1;
endmodule
```

The above example generates the following violation message:

```
[INFO] Crossing from 'top.in1' (domain 'TOP(supply VDD:1.100)')
to 'top.out1' (domain 'TOP(supply VDD:1.100)') is a feed
through path
```

The following is the schematic of this example:

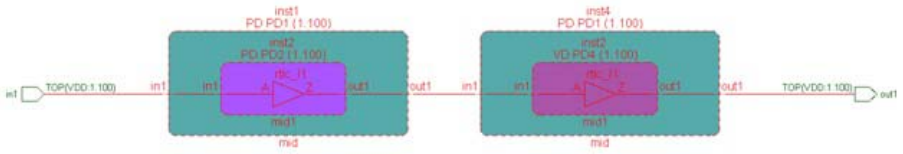


FIGURE 147. Incremental schematic

Default Severity Label

Info

Rule Group

Connection

Reports and Related Files

None

LPCONN07B

Checks for set_related_supply_net on boundary ports in feedthrough paths

When to Use

Use this rule to report feedthrough paths in a design.

Description

The *LPCONN07B* rule checks and reports:

- If the *set_related_supply_net* command is missing on boundary ports in a feedthrough path.
- If a less-on/equally-on supply is associated, using the *set_related_supply_net* command, with a boundary port in a feedthrough path.

Prerequisites

None

Rule Exceptions

None

Parameter(s)

- *lp_check_domain_equivalence*: Default value is `0` and the rule checks if the domains are equal based on their primary supplies. Set this parameter to `1` to check buffers that are existing in the same domain while checking for feed through paths.
- *lp_check_ls_strategy_presence*: Default value is `0`. Set this parameter to `1` to check for the presence of level shifter strategy on any of the ports on the domain boundary from source to destination.

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *create_supply_net* (Mandatory)

- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_related_supply_net* (Optional)

Messages and Suggested Fix

The following message appears if the *set_related_supply_net* command is missing on boundary ports in a feedthrough path and/or if a less-on/equally-on supply is associated, using the *set_related_supply_net* command, with a boundary port in a feedthrough:

[LPCONN07B_1][INFO] Feed through path between source '<source-name>' (domain '<source-domain>') and destination '<destination-name>' (domain '<destination-domain>') has no associated supply net specification for one or more boundary port(s)

Potential Issues

Not applicable.

Consequences of Not Fixing

Not applicable.

How to Debug and Fix

Review both source and destination ports mentioned in the violation message.

Example Code and/or Schematic

Consider the following example:

```
module top(in1,in2,out1,out2);
input in1,in2;
output out1,out2;
wire w1,w2;
mid inst1(in1,in2,w1,w2);
mid inst4(w1,w2,out1,out2);
endmodule

module mid(in1,in2,out1,out2);
input in1,in2;
```

```
output out1,out2;
mid1 inst2(in1,out1);
mid1 inst3(in2,out2);
endmodule

module mid1(in1,out1);
input in1;
output out1;
assign out1=in1;
endmodule
```

The above example generates the following violation message:

[INFO] Feed through path between source 'top.in1' (domain 'TOP(supply VDD:1.100)') and destination 'top.out1' (domain 'TOP(supply VDD:1.100)') has no associated supply net specification for one or more boundary port(s)

The following is the schematic of this example:

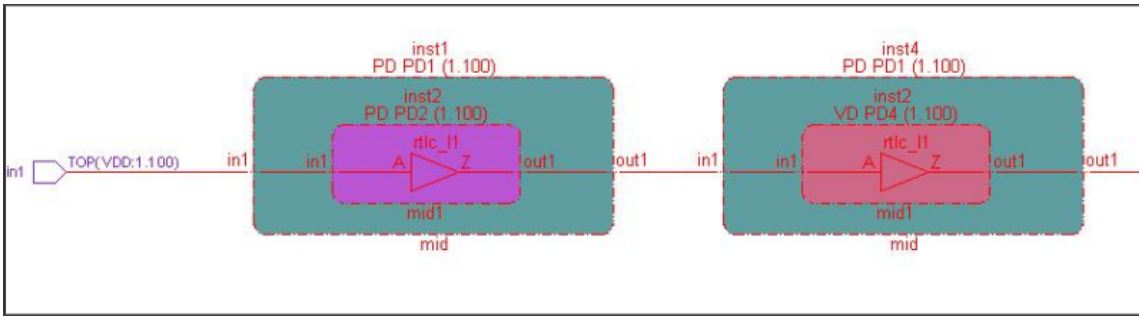


FIGURE 148. Incremental schematic

Default Severity Label

Info

Rule Group

Connection

Reports and Related Files

None

LPCONN08

Reports presence / absence of lockup latches between two flip-flops existing in different voltage domains but driven by synchronous clocks

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/ LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPCONN08* rule reports violations for both missing lockup latch and presence of a lockup latch between adjacent scan flip-flops lying in different voltage domains, if they are triggered by synchronous clocks.

A lockup latch is a latch where the d-pin of the latch has a single unblocked path to a scan flip-flop and the clock of that flip-flop is also connected to the lockup latch so that the latch is enabled (transparent) when the clock is at it's off state.

NOTE: *This rule is applicable to UPF only.*

Prerequisites

None

Rule Exceptions

None

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)

- *create_supply_port* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *add_port_state* (Mandatory)

SGDC

- *clock* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears to report a missing lockup latch for a voltage domain crossing in synchronous path between source flip-flop <flip-flop-full-hier-name> and destination flip-flop < flip-flop-full-hier-name>:

[LPCONN08_1][ERROR] Voltage domain crossing detected in synchronous path between flip-flops '<flip-flop-full-hier-name>' (CLK1: '<clock-name>', '<domain-and-supply-name>') and '<flip-flop-full-hier-name>' (CLK2: '<clock-name>', '<domain-and-supply-name>') without lockup latch.

Message 2

The following message appears to report the presence of a lockup latch in source or parent power domain for a voltage domain crossing in synchronous path between source flip-flop < flip-flop-full-hier-name> and destination flip-flop <flip-flop-full-hier-name>:

[LPCONN08_3][WARNING] Voltage domain crossing detected in synchronous path between flip-flops '<flip-flop-full-hier-name>' (CLK1: '<clock-name>', '<domain-and-supply-name>') and '<flip-flop-full-hier-name>' (CLK2: '<clock-name>', '<domain-and-supply-name>') with lockup latch

Message 3

The following message appears to report the presence of a lockup latch in the destination power domain for a voltage domain crossing in synchronous path between source flip-flop <flip-flop-full-hier-

name> and destination flip-flop <flip-flop-full-hier-name>:

[LPCONN08_2][ERROR] Voltage domain crossing detected in synchronous path between flip-flops ‘<flip-flop-full-hier-name>’ (CLK1: ‘<clock-name>’, ‘<domain-and-supply-name>’) and ‘<flip-flop-full-hier-name>’ (CLK2: ‘<clock-name>’, ‘<domain-and-supply-name>’) ‘<lockup-latch-name>’ (CLK: ‘<clock-name>’, ‘<domain-and-supply-name>’) with lockup latch in destination power domain.

Potential Issues

The violation messages appear because:

- **Message 1**: The source and destination flip-flops are present on two different domains while they are triggered by synchronous clocks, without a lockup latch.
- **Message 2:** The source and destination flip-flops are present on two different domains while they are triggered by synchronous clocks, with a lockup latch in source or parent power domain.
- **Message 3:** The source and destination flip-flops are present on two different domains while they are triggered by synchronous clocks, with a lockup latch in destination power domain.

Consequences of Not Fixing

This can lead to design failure.

How to Debug and Fix

Synchronous paths through voltage domain are not allowed. Due to this, the violations are reported. A violation is reported at the place where a voltage domain crossing is detected in synchronous path between flip-flops. These crossings can be with or without lockup latches, and the above violations are reported accordingly.

For a graphical view of the violation, double-click the message and click the Incremental Schematic button. It will show the violated crossings.

To resolve the violation of missing lockup latch, add a lockup latch between source and destination flip-flops.

Example Code and/or Schematic

Example 1

Consider the following example:

```
Clocks are defined in SGDC as follows

clock -name clk1 -domain clock_domain1
clock -name clk2 -domain clock_domain1
```

In the above example, the following violation occurs because no lockup latch is specified between the two synchronous flip-flops:

```
Voltage domain Crossing Detected in synchronous path between flip-flops 'top.inst2' (CLK1: 'top.clk1','PD1') and 'top.inst3' (CLK2: 'top.clk2','PD2') without lockup latch
```

The following is the schematic of this example:

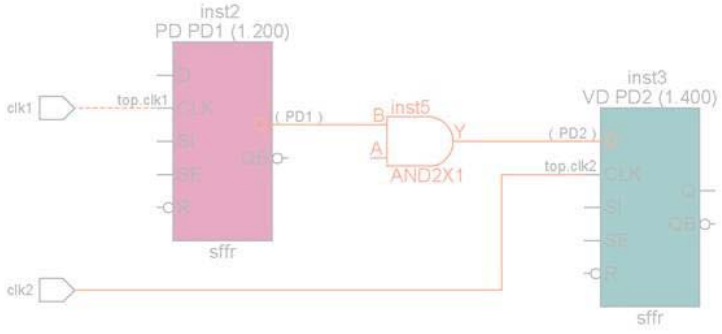


FIGURE 149. Incremental schematic

Example 2

Consider the following example:

```
Clocks are defined in SGDC as follows

clock -name clk1 -domain clock_domain1
clock -name clk2 -domain clock_domain1
```

In the above example, the following violation occurs because a lockup latch is specified in parent power domain between the two synchronous flip-flops:

```
Voltage domain Crossing Detected in synchronous path between
flip-flops 'top.inst1' (CLK1: 'top.clk1','PD1') and 'top.inst3'
(CLK2: 'top.clk2','PD2') with lockup latch
```

The following is the schematic of this example:

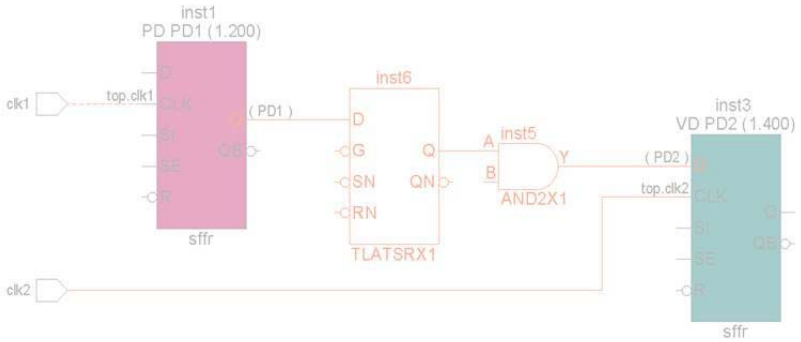


FIGURE 150. Incremental schematic

Example 3

Consider the following example:

```
Clocks are defined in SGDC as follows

clock -name clk1 -domain ck_d1
clock -name clk2 -domain ck_d1
```

In the above example, the following violation occurs because a lockup latch is specified in destination power domain between the two synchronous flip-flops:

```
Voltage domain Crossing Detected in synchronous path between
flip-flops 'top.inst1_0.ff2' (CLK1: 'top.clk1','PD1') and
'top.inst2_0.ff1' (CLK2: 'top.clk2','PD2') with lockup latch in
destination power domain
```

The following is the schematic of this example:

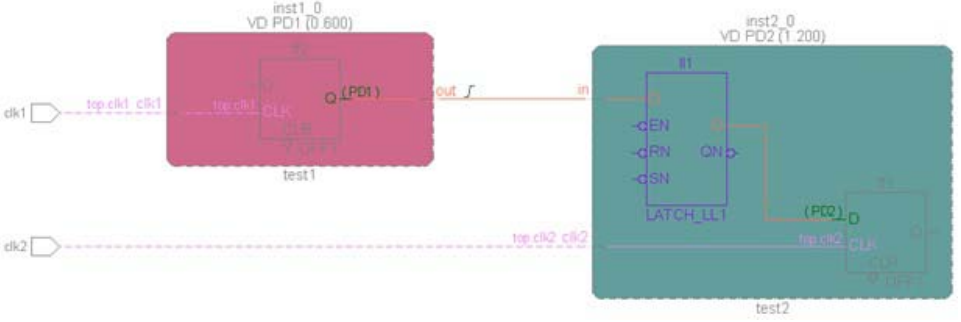


FIGURE 151. Incremental schematic

Default Severity Label

Error, Warning

Rule Group

Connection

Reports and Related Files

None

LPCONN09

Report presence / absence of charged device model (CDM) cells in crossings

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPCONN09* rule reports violations in the following situations:

Missing CDM Cell

The rule reports the missing CDM cell for crossings that have different ground supplies and CDM cell is missing from source and destination.

Present CDM Cell

The rule reports the presence of CDM cell if there is a CDM cell before a special cell (isolation, level shifter, or power switch) in a crossing and source and destinations have two different ground supplies.

Prerequisites

None

Rule Exceptions

None

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)

- *create_supply_port* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_related_supply_net* (Optional)
- *add_port_state* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when a CDM cell is missing for a crossing from source <*hier-name*> and destination <*hier-name*> while these source and destination have different ground supplies:

[LPCONN09_1][ERROR] Crossing Path detected from source ‘<hier-name>’ (domain ‘<source-domain-name>’ ground supply <ground-supply>) to destination ‘<hier-name>’ (domain ‘<destination-domain-name>’ ground supply <ground-supply>) without a CDM cell

Message 2

The following message appears for the presence of a CDM cell in a crossing before a special cell (isolation, level shifter, or power switch) from source <*hier-name*> and destination <*hier-name*> while these source and destination have different ground supplies:

[LPCONN09_2][WARNING] Crossing Path detected from source ‘<hier-name>’ (domain ‘<source-domain-name>’ ground supply <ground-supply>) to destination ‘<hier-name>’ (domain ‘<destination-domain-name>’ ground supply <ground-supply>) with a CDM cell

Potential Issues

The violation messages appear because:

- **Message 1**: The source and destination have two different ground supplies in a crossing while crossing does not include a CDM cell.
- **Message 2:** There is a CDM cell before a special cell (isolation, level shifter, or power switch) in a crossing while source and destination have different ground supplies.

Consequences of Not Fixing

This can lead to design failure.

How to Debug and Fix

For a graphical view of the violation, double-click the message and click the Incremental Schematic button. It will show the violated crossings.

- **Message 1**: Add a CDM cell between source and destination.
- **Message 2**: Remove the CDM cell from the crossing.

Example Code and/or Schematic

Example 1

The *LPCONN09* rule reports the following violation when no CDM cell is specified in crossing while source and destination have different ground supplies:

```
Crossing path detected from source 'top.in1' (domain : 'TOP'
ground supply 'VSSTOP') to destination 'top.inst2.D' (domain :
'PD2' ground supply 'VSS1') without a CDM cell.
```

The following is the schematic of this example:

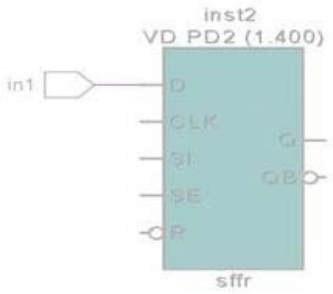


FIGURE 152. Incremental schematic

Example 2

The *LPCONN09* rule reports the following violation when a CDM cell is specified in a crossing before isolation cell:

```
Crossing path detected from source 'top.in1' (domain : 'TOP'
```

ground supply 'VSSTOP') to destination 'top.inst2.D' (domain : 'PD2' ground supply 'VSS1') with a CDM cell

The following is the schematic of this example:

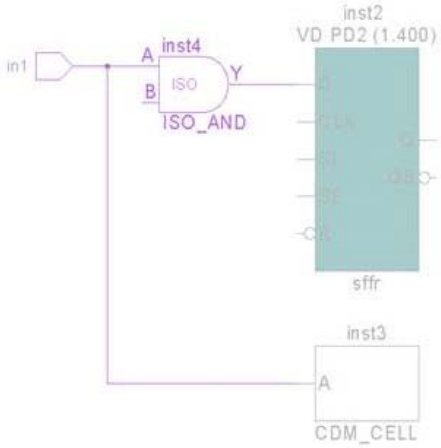


FIGURE 153. Incremental schematic

Default Severity Label

Error, Warning

Rule Group

Connection

Reports and Related Files

None

LPSVM49

Reports incorrect connections of power or ground signal pins of cells to their supplies

When to Use

Use this rule to check the electrical errors caused by multiple supply nets working as different voltages in the design.

This rule is recommended for use with *Gate-level netlist files and their associated library files*, *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*, and *DEF files and their associated LEF files*.

Description

The *LPSVM49* rule reports the nets that are connected to the pins with different supply names. For each net in the design, this rule checks whether the connected pins have the same supply name.

NOTE: *When you enable the LPSVM49 rule and set the lp_check_isocell parameter, the SGDC_lowpower87 rule reports a fatal violation if you do not specify the* `-isosig` *argument of the supply constraint for a switched supply.*

Prerequisites

The *LPSVM49* rule requires you to:

- specify the power or ground pin names in the PLIB or LEF files, by using the *pg_pins_naming* constraint, for single power supply cells.
- specify the power supply names for each pin of the multi-power supply cells, by using the *cell_pin_info* constraint.

Parameter(s)

- *lp_max_pins_viol*: Default value is `10`. This indicates that the *LPSVM49* rule reports up to `10` pins for a net. Set this parameter to a positive integer value to specify the maximum number of pins to be reported for a single net.
- *lp_ignore_same_voltage_error*: Default value is `0`. Set this parameter to `1` to ignore checking supplies with different names but the same value.
- *lp_check_isocell*: Default value is `1`. This indicates that the *LPSVM49* rule does not report the net crossings from the power domain gate to the

proper isolation gate. Set this parameter to 0 to report the net crossings.

Constraint(s)

- *supply* (Mandatory): Use this constraint to specify the supply and ground port names.
- *pg_pins_naming* (Optional): Use this constraint to specify the power or ground pin names, if not specified in the PLIB or LEF files.
- *cell_pin_info* (Optional): Use this constraint to specify the power supply cells or the power supply names for each pin of the multi-power supply cells.

NOTE: *The LPSVM49 rule is also run in the CPF or UPF flow. Specify the pg_pins_naming and cell_pin_info constraints in the SGDC file. However, the voltage domain information is retrieved only from the corresponding CPF or UPF file.*

Messages and Suggested Fix

Message 1

The following message appears when the net *<net-name>* is connected to the pins with different supply names:

[LPSVM49_1][WARNING] Net connected to multiple power supplies: '<net-name>' is connected to pin '<pin-supply-value-pair-list>'

Where,

<pin-supply-value-pair-list> is in the following format:

<pin1-name>(<supply1-name>), <pin2name>(<supply2name>),...

Potential Issues

This violation appears when a net is connected to the pins with different supply names.

Consequences of Not Fixing

A design with multiple supply nets working as different voltages may lead to a design error. In such a situation, the voltage value of the source terminal of a signal net is different from the value of the receiver terminal of that signal net.

How to Debug and Fix

To fix this violation, ensure that the nets are connected to the pins of the specified supply cells only.

Message 2

The following message appears for the multi-supply cell *<cell-name>* that is instantiated in the design but is not specified by using the `cell_pin_info` constraint:

```
[LPSVM49_2][WARNING] Multi power supply cell '<cell-name>' is
not specified through 'cell_pin_info' constraint
```

Potential Issues

This violation appears if a multi-supply cell is not specified by using the `cell_pin_info` constraint.

Consequences of Not Fixing

If you do not fix this violation, no supply is specified for the pins of the multi-supply cell.

How to Debug and Fix

To fix this violation, specify the multi-power supply cell, by using the `cell_pin_info` constraint, in the SGDC file.

Message 3

The following message appears for the cell *<cell-name>* that does not have the power or ground pins specified by using the `pg_pins_naming` constraint:

```
[LPSVM49_3][WARNING] Cell '<cell-name>' does not have power/
ground terminal as specified in 'pg_pins_naming' constraint
```

Potential Issues

You may encounter this violation message in the following cases:

- There is a typo error while specifying the power or ground pins in the SGDC file.
- The specified pins do not exist in the cell.

Consequences of Not Fixing

If you do not fix this violation, the pin of the cell is not treated as its power or ground pin. The rules using the power or ground pins of that cell may

produce incorrect results.

How to Debug and Fix

To fix this violation, ensure that the power or ground pins are correctly specified in the `pg_pins_naming` constraint.

Example Code and/or Schematic

Example 1

The names of the power pins of all single power supply cells are based on the regular expression *VDD*. Similarly, the names of the ground pins are based on the regular expression *GND* or *VSS*:

```
pg_pins_naming -power "*VDD*" -ground "*GND*" "*VSS*"
```

In this case, the *LPSVM49* rule infers the power supply names and ground supply names for the power pins and ground pins, respectively, of the single power supply cells.

Example 2

The following command specifies that the supply name for the pin VSS of the multi-power supply cells named INVX2* is VSS1 and so on:

```
cell_pin_info -cellname "INVX2*"
  -pin_name_supply VSS VSS1 VDD VDDC IN VDDC OUT VDDC
```

Example 3

The `top.VDDTOP` net is connected to the pins having different supply names. Therefore, the *LPSVM49* rule reports a violation.

For a graphical view of the violation, double-click the message and click **Incremental Schematic**. The schematic is shown in the following figure:

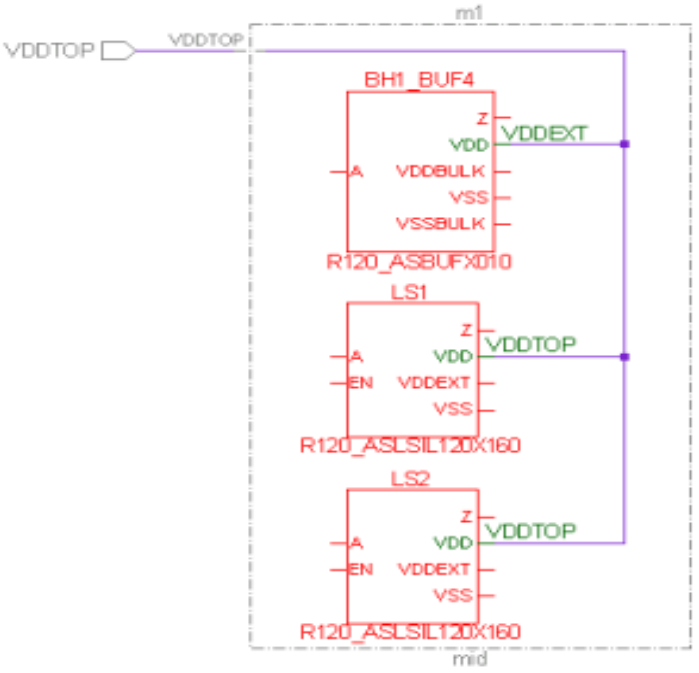


FIGURE 154. Incremental schematic

Default Severity Label

Warning

Rule Group

Connection

Reports and Related Files

None

LP_SPECIAL_PIN_CONNECTION

Generates report that shows special pin connections to ports

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LP_SPECIAL_PIN_CONNECTION* rule generates the *lp_special_pin_connection* report. This report contains information about special pin connections to ports as interpreted by SpyGlass.

Parameter(s)

- *lp_skip_buf*: Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.
- *lp_relative_aon_checking*: Default value is 0. Set the value to 1 to turn on relatively always-on checking.

Constraint(s)

- None

Messages and Suggested Fix

The following message after the *lp_special_pin_connection* report is generated:

```
[INFO] Special Pin connections to Ports is generated in the report 'lp_special_pin_connection'
```

Potential Issues

Not applicable

Consequences of Not Fixing

Not applicable

How to Debug and Fix

Review the *lp_special_pin_connection* report to understand how SpyGlass interprets special pin connections to ports.

Example Code and/or Schematic

Not applicable

Default Severity Label

Info

Rule Group

Connection

Reports and Related Files

lp_special_pin_connection

Supply Rules

The rules in this group are as follows:

Rule	Reports
LPPLIB06	Cells (other than level shifters) instantiated in a block whose power pins are not correctly connected to the block's power supply
LPPLIB13	Instances of user-specified special cells that are not connected to the correct supply rail
LPPLIB14	Reports signal pins that have improperly connected supply nets
LPPLIB15	Cells (other than level shifters) instantiated in a block whose ground pins are not correctly connected to the block's ground supply
LPPLIB16	Incorrect connection of tie low/high to voltage domain
LPPLIB17	Input/output power pins and enable pin of powerswitch instances that are not connected to the correct supply rails
LPPLIB18	Incorrect connections of bias power/ground pins to bias power/ground supply, respectively
LPPLIB18A	Ensures correct connections of bias power pin to bias power supply
LPPLIB18B	Ensures correct connections of bias ground pin to bias ground supply
LPPLIB19	Checks whether bias net is always-on with respect to power net
LPPLIB19A	Checks whether the bias_net is less always-on than the power net
LPPLIB20	Checks whether library cell instantiated in design is used out of place
LPSUP01	Standard cells that drive the supply nets
LPSUP03	Checks for consistency between supply net specified for leaf level signal pins and the related power and ground pin

LPPLIB06

Reports incorrect connections to power and voltage domains

When to Use

In power managed designs, each voltage domain and each power domain requires a separate supply net. Therefore, it is important to make sure that each gate in the post-route design is connected to the correct supply. Use this rule for supply net checking in the Post-layout design phase.

This rule is recommended for use with *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)* and *DEF files and their associated LEF files*.

Description

The *LPPLIB06* rule checks for supply connection for cells in the Post-layout design phase. The following list contains the criteria for the supply connection:

- Always-on buffers that appear in a switched power domain must be connected to the main (always-on) voltage.
- Retention registers appear in a switched power domain must be connected to both main voltage and the local voltage. The switched power pin should be connected to local voltage and the always-on power pin should be connected to main (always-on) voltage.
- Power switch cells must be connected to both local and switched voltage. The output power pin should be connected to switched supply net and the input supply pin should be connected main (external) voltage.

NOTE: *The rule infers pin type information for multi supply cells from the library files to determine whether a pin is a primary power pin or a backup power pin. For details of the library attributes, refer to Using Constraints in the SpyGlass Power Verify Solution section.*

Prerequisites

Specify the following information before running this rule:

- Supply and Ground net names, by using the *supply* constraint.

- (Optional) Power/Ground pins, by using the *pg_cell* constraint to provide details of Power/Ground pins for the cells that are not in the specified PLIB/LEF files.
- Supply and Ground port names for each voltage domain, by using the `-supplyname` argument of the *voltage_domain* constraint. For example, the following code specifies that the `VDD`, `VSS` supply is associated with the `V1` voltage domain and `VDDEXT`, `VSSEXT` supply is associated with the V2 voltage domain:

```
voltage_domain -name V1 -value 1.2 -instname mid.t1.inst1
-supplyname VDD VSS

voltage_domain -name V2 -value 1.6 -instname mid.t1.inst4
-supplyname VDDEXT VSSEXT

supply -name VDD -value 1.2

supply -name VDDEXT -value 1.6

supply -name VSS -value 0

supply -name VSSEXT -value 0'
```

Reported Information

The rule reports the cell instances in which the power pin is not correctly connected to the power supply of the parent voltage domain in the following cases:

- The power pin of the cell instance is hanging.
- The power pin is connected to a power supply rail other than the one specified for the parent voltage domain.

NOTE: *For the above three cases, the rule ignores the power pins specified with the* `-exclude` *argument of the power_switch constraint.*

- The power pin for an always-on buffer is not connected to an always-on supply, when the `VDDC` pin is not specified. When it is specified, then the rule checks whether the `VDDC` pin of an always-on buffer (specified using *always_on_buffer* constraint) is connected to an always-on supply and the other power pin is connected to a switchable supply.
- To not report violations pertaining to specific pins of a cell, set the *ignore_supply_pin* constraint.

- There is a mismatch between the connections mentioned in the `connect_supply_net` UPF command and the actual connection in PG netlist.

Language

Verilog, VHDL, DEF

Rule Exceptions

This rule does not check for bias pins. The *LPPLIB18* rule checks for bias pins.

Parameter(s)

- *lp_max_viol_count*: Default value is 1000. This indicates that a maximum of 1000 rule messages are reported. Set the value to any positive integer to report more or less rule messages.
- *lp_check_internal_pg_pin_connection*: Default value is `no`. Set this parameter to `yes` to check the PG connections of internal PG pin whose direction is output.
- *lp_check_equally_on_supplies_for_aon_pins*: Default value is `no`. Set this parameter to `yes` to consider equally-on supplies as more-on supplies.

Constraint(s)

SDGC

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage/power domains in the design.
- *supply* (Mandatory): Use this constraint to specify the supply and ground port names for the LPPLIB rules.
- *pg_cell* (Optional): Use this constraint to specify the names of power/ground pins for cells present in the input netlist, which are missing from the respective PLIB/LIB/LEF libraries.
- *always_on_buffer* (Optional): Use this constraint to specify signals that should be driven by an always-on buffer.
- *power_switch* (Optional): Use this constraint to specify the power switches in a power domain.
- *ignore_supply_pin* (Optional): Use this constraint to not report specific pins of a cell.

CPF Commands

- *create_power_nets* (Mandatory)
- *create_ground_nets* (Mandatory)
- *create_power_domain* (Mandatory)
- *update_power_domain* (Mandatory)
- *define_always_on_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the power pin <*pin-name*> of cell <*cell-name*> instantiated as instance <*inst-name*> is connected to the wrong supply rail <*supply-name*>, but should be connected to supply rail <*supply1-name*>:

[LPPLIB06_1][Recommended] Power pin '<pin-name>' for cell '<cell-name>' instantiated as '<inst-name>' is connected to wrong supply rail('<supply-name>')[correct supply rail is '<supply1-name>']

For information on debugging, click *How to Debug and Fix*.

Message 2

The following message appears when the cell <*cell-name*> (power pin <*pin-name*>) instantiated as instance <*inst-name*> is operating in the voltage domain <*vd-name*> is connected to the wrong supply rail <*supply-name*>, but should be connected to supply rail <*supply1-name*>:

[LPPLIB06_2][Recommended] Cell '<cell-name>', with power pin '<pin-name>', instantiated as '<inst-name>' operating in voltage domain '<vd-name>' is connected to wrong supply('<supply-name>')[correct supply rail is '<supply1-name>']

For information on debugging, click *How to Debug and Fix*.

Message 3

The following message appears when the power pin *<pin-name>* of cell *<cell-name>* instantiated as instance *<inst-name>* is not connected to any supply rail, but should be connected to supply rail *<supply1-name>*:

[LPPLIB06_3][Recommended] Power pin '<pin-name>' for cell '<cell-name>' instantiated as '<inst-name>' is not connected to supply rail[correct supply rail is '<supply1-name>']

For information on debugging, click *How to Debug and Fix*.

Message 4

The following message appears when the always-on power pin *<pin-name>* for the always-on buffer *<cell-name>* instantiated as instance *<inst-name>* is not connected to always-on supply:

[LPPLIB06_4][Recommended] Always-on power pin '<pin-name>' for cell '<cell-name>' instantiated as '<inst-name>' should be connected to always-on supply

Message 5

The following message appears when a net *<net-name>* connected to a power pin *<pin-name>* of instance *<inst-name>* (cell name *<cell-name>*) is not coming from a supply port or power switch output:

[LPPLIB06_5][Recommended] Net '<net-name>' connected to power pin '<pin-name>' of instance '<inst-name>' (cellname '<cell-name>') should come from supply port or power switch output

Message 6

The following message appears for power switch, level shifter, isolation and retention cells when there is a mismatch between connections mentioned in the `connect_supply_net` UPF command and the actual connection

in PG netlist:

[LPPLIB06_6][Recommended] Power pin '<pin-name>' for cell '<cell-name>' instantiated as '<instance-name>' is connected to the supply net '<net-name>' instead of the specified supply net '<net-name>'

Message 7

An informational message appears when the violation count of this rule exceeds the limit set by the *lp_max_viol_count* parameter. Refer to *Message 5* for the message and, for debugging information, refer to *How to Debug and Fix*.

Potential Issues

The order in which the rule checks of the supply of the power pin are made is a follows:

a. If the *connect_supply_net* constraint is used for this power pin, the power pin is expected to have the supply mentioned with this *connect_supply_net* command as its supply in the design.

b. If the *connect_supply_net* constraint is not used, but `isolation_power_net` in the *set_isolation* command is mentioned in the strategy present at the domain boundary directly connected to this cell, the power pin is expected to have this `isolation_power_net` as its supply in the design.

c. If neither `isolation_power_net` in the *set_isolation* command nor the *connect_supply_net* command is mentioned for this pin, the rule expects the power pin to be connected to the supply rail of the domain the cell is present in.

Message 1 appears for cases a and b listed above.

Message 2 appears for case c.

Messages 3 and 4 appear for the following reasons:

- The power pin in unconnected (hanging).
- The always-on power pin is connected to a not-always-on supply.

Consequences of Not Fixing

The instance will not function due to the power pin connected either to no voltage supply or to an incorrect voltage supply.

How to Debug and Fix

The message states that the power pin is either hanging or connected to the wrong supply. In addition, it also reports the details of the correct supplies to which the pin should be connected.

The violation message is reported at the cell instances which have incorrectly connected power pins.

For a graphical view of the violation, double-click the message and then click **Incremental Schematic**. Turn on Power View. The schematic shows the following:

- the power pin and its connection to the supply net
- the voltage domain in which this instance lies

To resolve this violation, ensure that the power pins of all the instances are properly connected to the power supply net associated with the power domain of the instance.

Example Code and/or Schematic

Example 1

The following example shows connection checking in voltage domains, where blocks can be at different voltages:

Consider a 1.2V design with one block running at 1.0 volts. Level shifters are placed at each output of the 1.0V block. The gates inside the 1.0V block must be connected to the 1.0V supply. The level shifters must have the input supply pin connected to the 1.0V supply, while the output supply pin must be connected to the 1.2V supply. In addition, the circuit checks whether a signal pin uses the correct supply and a ground connection is correct.

The violations of these checks (indicated by red lines) are shown in the following figure:

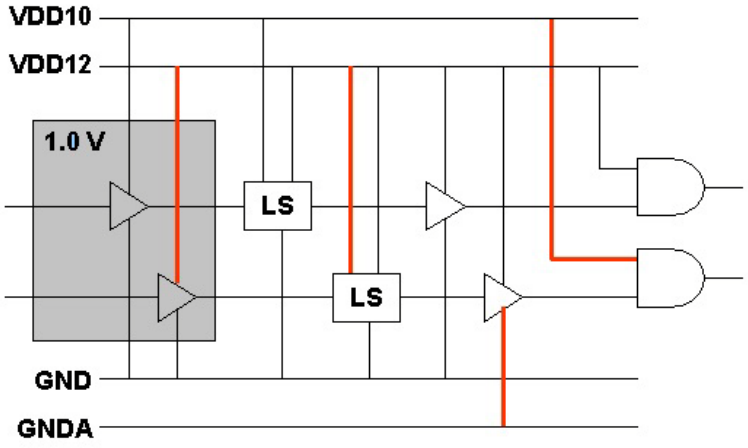


FIGURE 155. Connection checking in voltage domains

The SpyGlass Design Constraints file is as follows:

```
current_design top
supply -name VDD12 -value 1.2
supply -name VDD10 -value 1.0
supply -name GND -value 0.0
supply -name GNDA -value 0.0
voltage_domain -name Vtop -value 1.2 \
-modname top -supplyname VDD12 GND
voltage_domain -name VA -value 1.0 -instname top.ua \
-supplyname VDD10 GND -portname i1 i2
levelshifter -name LS10_12 -from VA -to Vtop \
    -inTerm I
    -outTerm O \
    -inSupplyTerm V10 -outSupplyTerm V12
```

After running the analysis, the following messages are displayed:

[Recommended] Cell 'BUFX1', with power pin 'VDD', instantiated

as 'top.ua.u2' operating in voltage domain 'VA' is connected to wrong supply('VDD12')

[Recommended] Cell 'BUFX1', with power pin 'VDD', instantiated as 'top.u6' operating in voltage domain 'Vtop' is connected to wrong supply('VDD10')

Example 2

The following message appears when the BUFX1 cell (VDD power pin) instantiated as the top.ua.u7 instance is connected to the supply VDDC. As the instance is operating in the VA domain, the power pin should be connected to supply VDD10 that is associated with the VA domain:

[Recommended] Cell 'BUFX1', with power pin 'VDD', instantiated as 'top.ua.u7' operating in voltage domain 'VA' is connected to wrong supply('VDDC')

The incremental schematic is displayed as shown below:

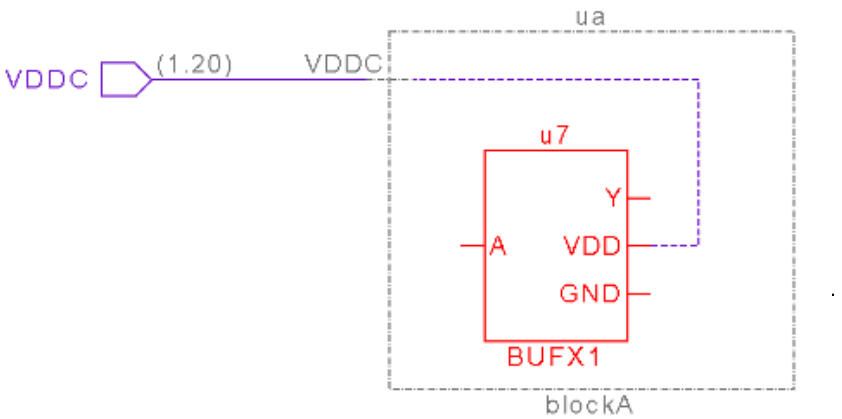


FIGURE 156. Incremental schematic

Example 3

This example illustrates when *Message 5* is reported. Consider the following Verilog snippet.

```
module top (VDDA,VTOP,VSS,VSS1,VDDB,iso);
```

```
T1 inst1(.int1(in1), .iso(iso), .aout1(aout1), .VDDA(),
.VSS1(VSS1), .VTOP(VTOP), .VSS(VSS));
endmodule


module T1 (int1,iso,aout1,aout2,VDDA,VSS1,VTOP,VSS);
ISO_AND
iso1(.A(int1),.B(iso),.Y(aout1),.VDD(VTOP),.VDDC(VDDA),.VSS(
VSS1));
ISO_AND
iso2(.A(int1),.B(iso),.Y(aout2),.VDD(VTOP),.VDDC(VDDA),.VSS(
VSS1));
endmodule
```

The following is the UPF file snippet:

```
create_power_domain top
create_power_domain UA -elements inst1


create_supply_port VDDA
create_supply_port VTOP


add_port_state VDDA -state {default_VDDA 1.2} -state
{off_state off}
add_port_state VTOP -state {default_VTOP 1.4} -state
{off_state off}


create_supply_net  VTOP -domain top
create_supply_net  VTOP -domain UA -reuse
create_supply_net  VDDA -domain top


connect_supply_net VDDA -ports VDDA
```

As shown in the UPF, VDDA has not been defined as a supply net for domain UA. Therefore, the *LPPLIB06* rule reports *Message 5*.

[Recommended] Net 'top.inst1.VDDA' connected to power pin 'VDDC' of instance 'top.inst1.iso1' (cellname 'ISO_AND') should come from supply port or power switch output

The schematic generated is as follows:

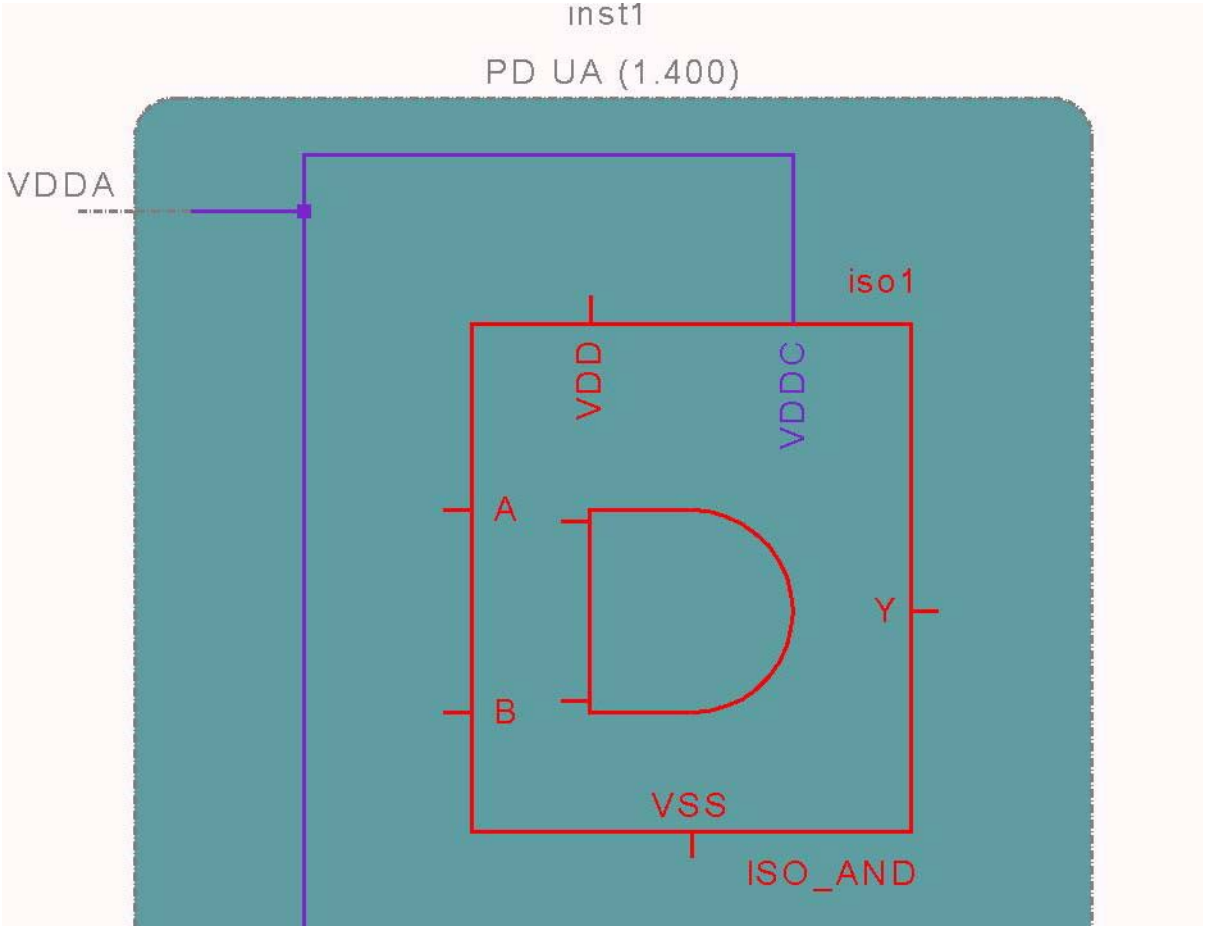


FIGURE 157. Incremental schematic

To resolve this violation, define VDDA as a supply net in domain UA.

Default Severity Label

Recommended

Rule Group

LayoutPowerConnectivityRules

Reports and Related Files

None

LPPLIB13

Reports supplies associated with special cells are inappropriately connected

When to Use

Use this rule for:

- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPPLIB13* rule reports instances of user-specified special cells that are unconnected or are not connected to the correct supply rail.

Language

Verilog, VHDL

Parameter(s)

- *lp_max_viol_count*: Default value is `1000`. Set this parameter to a positive integer number to specify the maximum number of violations that should be reported by the rule.

Constraint(s)

SGDC

- *special_cell* (Mandatory): Use to specify the special cells.

Messages and Suggested Fix

Message 1

The following message appears when the power pin *<pin-name>* of instance *<inst-name>* of special cell *<cell-name>* is not connected to any of the specified supply rails:

[LPPLIB13_1][WARNING] Power pin '<pin-name>' of instance '<inst-name>' (<cell-name>) is not connected to the correct supply

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the power pin <*pin-name*> of instance <*inst-name*> of special cell <*cell-name*> is not connected:

[LPPLIB13_2][WARNING] Power pin '<pin-name>' of instance '<inst-name>' (<cell-name>) is unconnected

For debugging information, click *How to Debug and Fix*.

Message 3

An informational message appears when the violation count of this rule exceeds the limit set by the *lp_max_viol_count* parameter. Refer to *Message 5* for the message and, for debugging information, refer to *How to Debug and Fix*.

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The consequences of not fixing the violations are as follows:

Message 1: The power pin of the instance specified by using the *special_cell* constraint is not connected to one of the supplies provided by using the '-supplies' argument. Connection with wrong supply would make the cell function incorrectly.

Message 2: The power pin of the instance specified by using the *special_cell* constraint are left unconnected. This is a design flaw.

How to Debug and Fix

The violation is reported at the place where the special cell with improper connections is instantiated.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. The schematic shows the following:

- Power pins of the instances, specified by using the *special_cell* constraint, which are not connected to one of the supplies specified by using the '-supplies' argument of the constraint.
- Power pins of the instances, specified by using the *special_cell* constraint, which are left unconnected in the design.

To fix these violation messages, ensure that the supplies associated with special cells are appropriately connected.

Example Code and/or Schematic

The *LPPLIB13* rule reports the instances of the special cells when they not connected to one of the specified supply rails.

Consider the following example. A violation occurs when the power pin `'VDD'` of instance `'top.m1.LS2'` of special cell `R120_ASLSIL120X160` is not connected:

```
[WARNING] Power pin 'VDD' of instance 'top.m1.LS2'
(R120_ASLSIL120X160) is unconnected
```

The schematic is as follows:

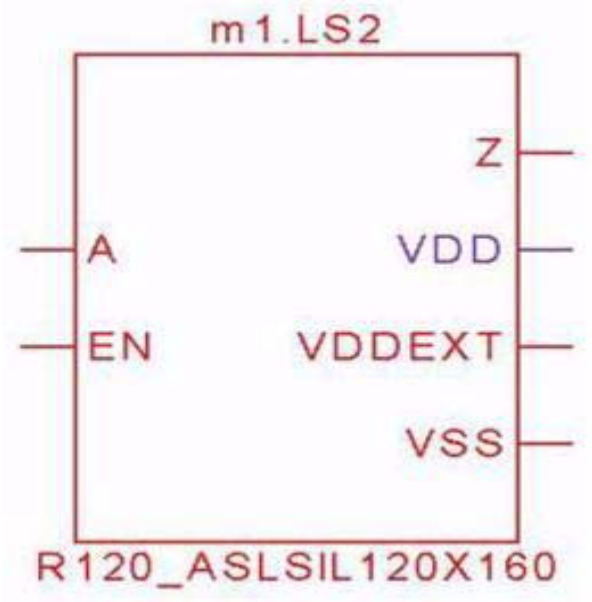


FIGURE 158. Incremental schematic

Default Severity Label

Warning

Rule Group

Supply Net Connection

Reports and Related Files

None

LPPLIB14

Reports signal pins that have improperly connected supply nets

When to Use

Use this rule for:

- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

This rule reports improper connections of supply nets to signal pins of instances as follows:

- Checks if the supply nets connected to the signal pins are the same as the ones connected to related power/ground supply pin.
- Reports the correct supply net in case of special cells and exception pins.
- Reports no-tie pins connected to supply nets.
- Reports exception pins that are not connected to a supply net.
- Checks when there is a mismatch between supply net connected to the input signal pin (Verilog connection) and related supply, through the `set_related_supply_net` or `connect_supply_net` commands.

In the SGDC flow, this rule works on two type of cells:

- **Normal Cells**
  - These are the cells that have one power and one ground pin. No constraint is required for these kind of cells. SpyGlass automatically picks the required information and checks the connections.
- **Special Cells**
  - These are the cells that have more than one power and ground pins. For these cells, you must mention the power/ground pin to be connected to signal `tie(1'b0,1'b1)` and the signal pins that should not be tied. The required information is given through the `cell_tie_class` constraint. For example:

```
cell_tie_class -cell <name> -tie1 <power_pin_name> -tie0
<ground_pin_name> -no_tie <pin1> -exception <pin1>
<pin1_tie1> <pin1_tie0>
```

Prerequisites

The LPPLIB14 rule requires you to specify the following:

- Power domains using the *voltage_domain* constraint
- Power and Ground port names using the *supply* constraint

Language

Verilog, VHDL

Parameter(s)

- *lp_max_viol_count*: Default value is 1000. Set the value to a positive integer to specify the maximum number of messages to be reported by the rule.
- *lp_check_same_srsn_supply*: Default value is no. Set the value to yes to report a violation for mismatch between supply net connected to the input signal pin (verilog connection) and related supply, through `set_related_supply_net` or `connect_supply_net` command.

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)

CPF Commands

- *create_power_nets* (Mandatory)
- *create_ground_nets* (Mandatory)
- *create_power_domain* (Mandatory)

SGDC

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage/power domains in the design.
- *supply* (Mandatory): Use this constraint to specify the supply and ground port names for the LPPLIB rules.

Messages and Suggested Fix

Message 1

The following message appears, when an input pin *<pin1-name>* of the cell *<cell-name>* instantiated as *<inst-name>* is connected to a supply net which is not the same as that of the net connected to the supply pin *<pin2-name>* of type *<pin-type>:*

[LPPLIB14_1][WARNING] Supply net connected to the input pin '<pin1-name>' of cell '<inst-name>(<cell-name>)' is not same as the net connected to the related <pin-type> pin '<pin2-name>'

Where, *<pin-type>* is ground or power.

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears, when an input pin *<pin-name>* of the special cell *<cell-name>* instantiated as *<inst-name>* is connected to a supply net *<net1-name>* instead of the supply net *<net2-name>* of type <net-type>:

[LPPLIB14_2][WARNING] Supply net '<net1-name>' connected to the input pin '<pin-name>' of special cell '<inst-name> (<cell-name>)' is not same as the specified <net-type> net '<net2-name>'

Where, *<net-type>* is *–tie0* or *–tie1*.

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears, when an exception input pin *<pin1-name>* of the special cell *<cell-name>* instantiated as *<inst-name>* is connected to a supply net which is not the same as that of the net connected to the supply pin *<pin2-name>* of type *<net-type>:*

[LPPLIB14_3][WARNING] Supply net connected to the exception pin '<pin1-name>' of special cell '<inst-name> (<cellname>)' is not same as the net connected to the <net-type> pin '<pin2-name>'

Where, *<pin-type>* is *–tie0* or *–tie1*.

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears, when an exception input pin *<pin1-name>* of the special cell *<cell-name>* instantiated as *<inst-name>* is connected to the supply net *<net1-name>* instead of the supply net *<net2-name>* of type *<net-type>*:

[LPPLIB14_4][WARNING] Supply net '<net1-name>' connected to the exception pin '<pin1-name>' of special cell '<inst-name> (<cell-name>)' is not same as the specified <net-type> net '<net2-name>'

Where, <pin-type> is –tie0 or –tie1.

For debugging information, click *How to Debug and Fix*.

Message 5

The following message appears, when a no-tie input pin *<pin-name>* of the special cell *<cell-name>* instantiated as *<inst-name>* is connected to the supply net *<net-name>*:

[LPPLIB14_5][WARNING] No tie pin '<pin-name>' of instance '<inst-name> (<cell-name>)' is connected to the supply '<net-name>'

For debugging information, click *How to Debug and Fix*.

Message 6

The following message appears, when the exception pin *<pin-name>* of the special cell *<cell-name>* instantiated as *<inst-name>* is not connected to any supply:

[LPPLIB14_6][WARNING] Exception pin '<pin-name>' of instance '<instname> (<cell-name>)' is not connected to any supply

For debugging information, click *How to Debug and Fix*.

Message 7

An informational message appears when the violation count of this rule exceeds the limit set by the *lp_max_viol_count* parameter. Refer to *Message 5* for the message and, for debugging information, refer to *How to Debug and Fix*.

Message 8

The following message appears when there is a mismatch between supply net connected to the input signal pin (Verilog connection) and related supply, through the `set_related_supply_net` command. This violation message is reported only when the `lp_check_same_srsn_supply` parameter is set to `yes`.

[LPPLIB14_8][WARNING] Supply <supply-name> associated with net <net-name> connected to the input pin <pin-name> of cell <inst-name>(<cell-name>) is not same as the supply <supply-name> mentioned as related supply in set_related_supply_net command

NOTE: *This violation message is reported only for macro cell (is_macro_cell = true) and pad cell (pad_cell = true).*

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The consequences of not fixing the violations are as follows:

Message 1: The supply net connected to the input power pin or ground pin of an instance of a cell does not match the net connected to the related power pin or ground pin of that instance respectively. Hence the warning.

Message 2: The supply net connected to the input pin of the instances, specified using the “special_cell” constraint does not match with the net which is connected to supply specified using the “-supplies” argument of the constraint.

Message 3: The supply net connected to the exception input pin of a special cell specified by “-exception” does not match with the net connected to the supply pin of its parent instance. Hence the warning.

Message 4: The exception input pin of a special cell specified by “-exception” is connected to a supply net which is not same with the specified power net or ground net connected to the supply pins.

Message 5: The no-tie input pin of an instance specified by ”-no_tie” is connected to the specified supply power net or ground net. So the pin

behaves like either tie0 or tie1. Hence the warning.

Message 6: The exception input pin of a special cell specified by "-exception" is not connected to any supply specified in its parent instance.

Message 8: There can be a conflict of connections if there is mismatch of tied supply given in Verilog file and related supply mentioned in the corresponding UPF command.

How to Debug and Fix

The violation is reported at the instance of cell (normal or special) for which signal pin is not properly connected.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

Power pins of the instances, specified using the 'special_cell' constraint which are not connected to one of the supplies specified using the '-supplies' argument of the constraint.

The exception input pin is either left unconnected in the design or connected to a net which is not same with the specified net connected with the supplies.

The no-tie input pin is connected with any of the specified supply nets.

To resolve these violation messages, ensure that the pins have properly connected supply nets.

Example Code and/or Schematic

Example 1

Consider the following verilog instantiation of cell `inpad` where `VB` is the related power pin of data pin `P1`.

Verilog:

```
module mid (VDDA, VTOP,VSS);
inpad pad1 (.P1(1'b1), .P2(1'b0),.VA(VDDA), .VC(VSS));
endmodule
```

UPF:

```
set_design_top mid

create_power_domain top

create_supply_port VTOP

add_port_state VDDA -state {default 1.0}
add_port_state VTOP -state {default 1.4}

create_supply_net  VDDA -domain top
create_supply_net  VTOP -domain top
connect_supply_net VDDA -ports VDDA
connect_supply_net VTOP -ports VTOP

set_domain_supply_net top -primary_power_net VTOP
primary_ground_net VSS
```

In the above example, the net connected to pin P1 is the primary supply of the domain in which it exists (VTOP in this case), however net VDDA is connected to its related power pin VA. To report this scenario, the LPPLIB14 rule reports the following violation:

```
Supply net connected to the input pin 'P1' of cell
'mid.pad1(inpad_01)' is not same as the net connected to the
related power pin 'VA'
```

The schematic generated is as follows:

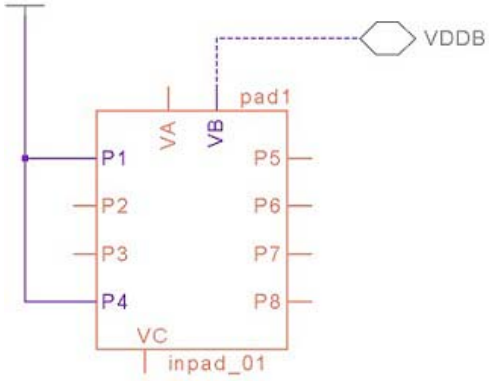


FIGURE 159. Incremental schematic

Example 2

Consider the following example. A violation occurs when a supply net connected to the input pin A of special cell is not same as the net connected to tie0 pin VSS. The following message appears:

```
[WARNING] Supply net connected to the input pin 'A' of special
cell 'top.LS2(R120_ASLSOL120X160)' is not same as the net
connected to the related tie0 pin 'VSS'
```

The schematic generated is as follows:

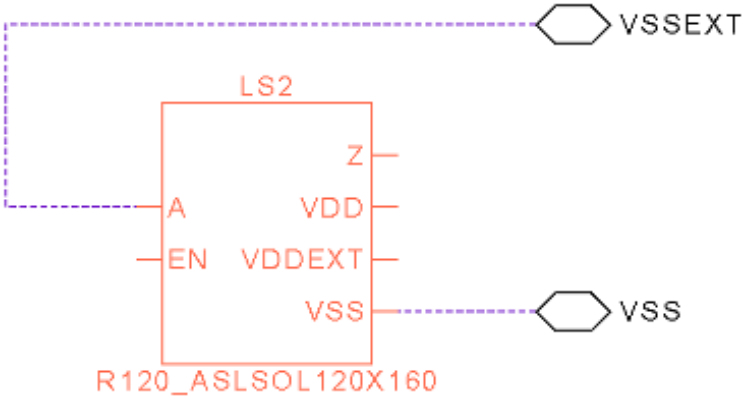


FIGURE 160. Incremental schematic

The *schematic* highlights the following:

- Input pins of special cell instances that are connected to a supply net which is not the same as that of the net connected to the supply pin.
- Input pins of special cell instances that are connected to a supply net which is different from the specified supply net.
- Exception input pins of special cell instances that are connected to a supply net which is not the same as that of the net connected to the supply pin.
- Exception input pins of the special cell instances are connected to a supply net which is different from the specified supply net.
- No-tie input pins of special cells/instances that are connected to a supply pin/net.

Example 3

Consider the following example. In the verilog file, the SEL pin is connected to the supply net `vdd1`:

```
PLL_PF_960MHZ_32K_TOP u_pll (
      .SEL(vdd1),
      .REFOUT(),
      .PHI2(),
```

```
.PHI1(),
.LOCKP(),
.INFOUT(),
.INFF(gnd),
.ENABLE(gnd),
.AGNDPLL1V8(gnd),
.AVDDPLL1V8(vdd1),
.DGNDPLL1V0(gnd),
.DVDDPLL1V0(vdd1));
```

In the UPF file, the `set_related_supply_net` command is written to connect the same SEL pin to vdd:

```
set_related_supply_net -object_list u_pll/SEL -power vdd -
ground gnd
```

In the above example, there is a mismatch in the supply mentioned in Verilog and UPF command for SEL pin, hence the following violation is reported:

```
Supply 'vdd1' associated with net 'ALL_CELLS.vdd1' connected to
the input pin 'SEL' of cell 'ALL_CELLS.u_pll
(PLL_PF_960MHZ_32K_TOP)' is not same as the supply 'vdd'
mentioned in set_related_supply_net command
```

Default Severity Label

Warning

Rule Group

Supply Net Connection

Reports and Related Files

None

LPPLIB15

Reports incorrect connections of the ground pin to the ground supply

When to Use

Use this rule for:

- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPPLIB15* rule reports cells, other than special cells such as level shifters and decoupling cells, which have been instantiated in a block whose ground pins are not correctly connected to the block's ground supply. This rule also checks supply connections of power switch cells.

Prerequisites

The *LPPLIB15* rule requires you to specify the following:

- Supply and Ground port names using the *supply* constraint
- Supply and Ground port names, specified using the supply constraint, for each voltage domain using the *–supplyname* argument of the *voltage_domain* constraint
- (Optional) Power/Ground pins using the *pg_cell* constraint to provide details of Power/Ground pins for the cells that are not in the specified PLIB/LEF files

Rule Exceptions

The *LPPLIB15* rule ignores the ground pins specified with the *-exclude* argument of the *power_switch* constraint.

Language

Verilog, VHDL

Parameter(s)

- *lp_max_viol_count*: Default value is 1000. Set the value to a positive integer to specify the maximum number of messages to be reported by the rule.
- *lp_check_internal_pg_pin_connection*: Default value is `no`. Set this parameter to `yes` to check the PG connections of internal PG pin whose direction is output.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/ power domains in the design.
- *supply* (Mandatory): Use to specify the supply and ground port names.
- *pg_cell* (Optional): Use to specify the names of Power/Ground pins for the cells present in the input netlist, and are missing in the specified PLIB/LIB/LEF libraries.
- *power_switch* (Optional): Use to specify the power switches in power domains.
- *always_on_buffer* (Optional): Use to specify the names of always-on buffers.

CPF Commands

- *create_power_nets* (Mandatory)
- *create_ground_nets* (Mandatory)
- *create_power_domain* (Mandatory)
- *update_power_domain* (Mandatory)
- *define_always_on_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)

- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears, when the ground pin <*pin-name*> of cell <*cell-name*> instantiated as instance <*inst-name*> is not connected to the specified supply rail, but should be connected to the supply rail <supply1-name>:

[LPPLIB15_1][RECOMMENDED] Ground pin '<pin-name>' for cell '<cell-name>' instantiated as '<inst-name>' is connected to wrong supply rail ('<supply-name>') [correct supply rail is '<supply1-name>']

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears, when the ground pin <*pin-name*> of cell <*cell-name*> instantiated as instance <*inst-name*> is not connected to any ground supply rail, but should be connected to the supply rail <supply1-name>:

[LPPLIB15_2][RECOMMENDED] Ground pin '<pin-name>' for cell '<cell-name>' instantiated as '<inst-name>' is not connected to supply rail [correct supply rail is '<supply1-name>']

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears, when the ground pin <*pin-name*>) of the cell <*cell-name*> instantiated as the instance <*inst-name*> operating in the specified voltage domain <*vd-name*> is connected to the wrong supply <*supply-name*>, but should be connected to the supply rail <supply1-name>:

[LPPLIB15_3][RECOMMENDED] Cell '<cell-name>', with ground pin '<pin-name>', instantiated as '<inst-name>' operating in voltage domain '<vd-name>' is connected to wrong supply rail('<supply-name>') [correct supply rail is '<supply1-name>']

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears, when the always-on ground pin *<pin-name>* for the cell *<cell-name>* instantiated as instance *<inst-name>* is not connected to an always-on supply:

[LPPLIB15_4][RECOMMENDED] Always-on ground pin '<pin-name>' for cell '<cellname>' instantiated as '<inst-name>' should be connected to always-on supply

For debugging information, click *How to Debug and Fix*.

Message 5

An informational message appears when the violation count of this rule exceeds the limit set by the *lp_max_viol_count* parameter. Refer to *Message 5* for the message and, for debugging information, refer to *How to Debug and Fix*.

Potential Issues

The violation messages appear because of cell instances where the ground pin is not correctly connected to the ground supply of the parent voltage domain in the following cases:

- The ground pin of the cell instance is hanging.
- The ground pin is not connected to a ground supply rail.
- The ground pin is connected to a ground supply rail other than the one specified for the parent voltage domain. The order in which the rule checks of the supply of the ground pin are made is a follows:
  a. If the *connect_supply_net* constraint is used for this ground pin, the ground pin is expected to have the supply mentioned with this *connect_supply_net* command as its supply in the design.
  b. If the *connect_supply_net* constraint is not used, but `isolation_ground_net` in the *set_isolation* command is mentioned in the strategy present at the domain boundary directly connected to this cell, the ground pin is expected to have this `isolation_ground_net` as its supply in the design.
  c. If neither `isolation_ground_net` in the *set_isolation* command nor the *connect_supply_net* command is mentioned for this pin, the rule expects the ground pin to be connected to the supply rail of the domain the cell is present in.

- The ground pin for an always-on buffer is connected to an always-on supply rail

Consequences of Not Fixing

The consequences of not fixing the violations are as follows:

Message 1: The ground pin of the instance is not connected to the correct supply rail. This is a design flaw.

Message 2: The ground pin of the instance is left unconnected. This is a design flaw.

Message 3: The ground pin of the instance is not connected to the ground supply of the voltage domain it is present in. This is a design flaw.

Message 4: The always-on ground pin of the instance is not connected to an always on supply. This is a design flaw.

How to Debug and Fix

The violation is reported at the place where cell (with incorrectly connected ground pins) is instantiated.

For a graphical view of the violation, double-click the message and click the Incremental Schematic button. Refer to the Example Code and/or Schematic section for an example.

To resolve these violation messages, ensure that the ground pins of all instances are properly connected.

Example Code and/or Schematic

Example 1

Consider the following example. A violation occurs when the ground pin gnd of cell IVHVT2 is not connected to the specified supply rail. The following message appears:

```
[RECOMMENDED] Ground pin 'gnd' for cell 'IVHVT2' instantiated as

'top.U16' is connected to wrong supply rail('vdd')
```

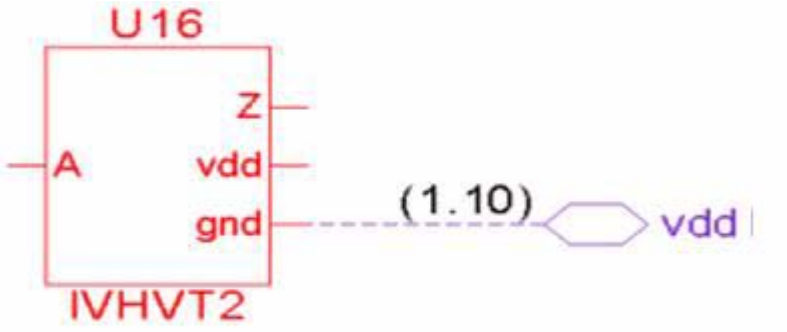


FIGURE 161. Incremental schematic

The schematic highlights the following:

- Ground pin that is not connected to the specified supply rail.
- Cell instantiated as instance is not operating in the specified voltage domain.
- Ground pin that is not connected to any ground supply rail.
- Ground pin of an always-on buffer is not connected to an always-on supply rail

Example 2

The SGDC file is as follows:

```
current_design top supply -name VDD12 -value 1.2
supply -name VDD10 -value 1.0
supply -name GND -value 0.0
supply -name GNDA -value 0.0
voltage_domain -name Vtop -value 1.2 -modname top \
-supplyname VDD12 GND
voltage_domain -name VA -value 1.0 -instname top.ua \
-supplyname VDD10 GND -portname i1 i2
```

```
levelshifter -name LS10_12 -from VA -to Vtop \
-inTerm I \
-outTerm O \
-inSupplyTerm V10 -outSupplyTerm V12
```

After running the analysis, the following message appears:

Cell 'BUFX1', with ground pin 'GND', instantiated as

'top.u5' operating in voltage domain 'Vtop' is connected to wrong supply rail('GNDA') [correct supply rail is 'GNDTOP']

As the instance ‘top.u5’ is instantiated in the voltage domain ‘Vtop’, then its ground pin ‘GND’ should have been connected to the ground supply ‘GNDTOP’ of the domain ‘Vtop’ instead of the supply rail (‘GNDA’)

Default Severity Label

Recommended

Rule Group

Supply Net Connection

Reports and Related Files

None

LPPLIB16

Ensures connection of tie low / high to correct voltage domain

When to Use

Use this rule for:

- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPPLIB16* rule reports cells instantiated in a block whose signal pins are not correctly connected to the block's supply.

Language

Verilog, VHDL

Parameter(s)

- *lp_max_viol_count*: Default value is 1000. Set the value to a positive integer to specify the maximum number of messages to be reported by the rule.

Constraint(s)

SGDC

- *supply* (Mandatory): Use to specify supply and ground port names.
- *voltage_domain* (Mandatory): Use to specify voltage domain using the `-supplyname` argument for which supply and ground port names are specified by using the *supply* constraint.
- *pg_cell* (Optional): Use to specify `Power/Ground` pins for the cells that are not in the specified `PLIB/LEF` files.

CPF Commands

- *create_power_nets* (Mandatory)
- *create_ground_nets* (Mandatory)
- *create_power_domain* (Mandatory)

- *update_power_domain* (Mandatory)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears, when the cell *<cell-name>* (signal pin *<pin-name>*) instantiated as instance *<inst-name>*, operating in voltage domain *<vd-name>* is not connected to the specified supply rail:

[LPPLIB16_1][RECOMMENDED] Cell '<cell-name>', with signal pin '<pin-name>', instantiated as '<inst-name>' operating in voltage domain '<vd-name>' is connected to wrong supply rail('<supply-name>')

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears, when the cell *<cell-name>* (signal pin *<pin-name>*) instantiated as instance *<inst-name>*, operating in voltage domain *<vd-name>* is connected to the net <supplyname>, which is not a supply rail:

[LPPLIB16_2][RECOMMENDED] Cell '<cell-name>', with signal pin '<pin-name>', instantiated as '<inst-name>' operating in voltage domain '<vd-name>' connected to net '<supply-name>' is not a supply rail

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears, when the signal pin *<pin-name>* of cell *<cell-name>* instantiated as instance *<inst-name>* is connected to

logic 1/logic 0:

```
[LPPLIB16_3][RECOMMENDED] Signal pin '<pin-name>' for cell
'<cell-name>' instantiated as '<inst-name>' is connected to
logic '<1/0>'
```

Message 4

An informational message appears when the violation count of this rule exceeds the limit set by the *lp_max_viol_count* parameter. Refer to the *LPSVM38* rule's *Message 5* for the message and, for debugging information, refer to *How to Debug and Fix*.

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The consequences of not fixing the violations are as follows:

Message 1: The signal pin of the cell instantiated in a power domain is connected to the wrong supply rail. This is a design flaw.

Message 2: The signal pin of the cell instantiated in a power domain is not connected to any supply rail. This is a design flaw.

Message 3: The signal pin of the cell is connected to logic 1 or 0. This is a design flaw.

How to Debug and Fix

The violation messages are reported at the place where the cell with improper connections is instantiated.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix these violation messages, ensure that signal pins of the cells instantiated in a block are correctly connected to the block's supply.

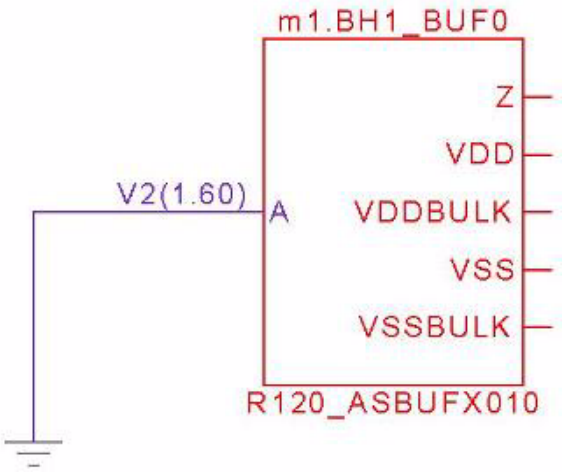
Example Code and/or Schematic

Example 1

Consider the following example. A violation occurs when the signal pin `'A'` of cell `'R120_ASBUFX010'` instantiated as `'top.m1.BH1_BUF0'` is

connected to logic '0':

```
[RECOMMENDED] Signal pin 'A' for cell 'R120_ASBUFX010'
instantiated as 'top.m1.BH1_BUF0' is connected to logic '0'
```



The schematic highlights the following:

- The signal pin is connected to logic 1/logic 0.
- The signal pin is connected to a supply rail other than the one specified for the parent voltage domain.

Example 2

Consider the following example. The SGDC file is as follows:

```
current_design top supply -name VDD12 -value 1.2
supply -name VDD10 -value 1.0
supply -name GND -value 0.0
supply -name GNDA -value 0.0
voltage_domain -name Vtop -value 1.2 -modname top -
supplyname VDD12 GND
voltage_domain -name VA -value 1.0 -instname top.ua \
-supplyname VDD10 GND -portname i1 i2
```

```
levelshifter -name LS10_12 -from VA -to Vtop -inTerm I
-outTerm O \
-inSupplyTerm V10 -outSupplyTerm V12
```

After running the analysis, the following message appears:

[RECOMMENDED] Cell 'AND2X1', with signal pin 'A', instantiated as 'top.u8' operating in voltage domain 'Vtop' is connected to wrong supply rail (‘supply VDD10:1.000’)

As the cell ‘AND2X1’ is instantiated in the voltage domain ‘Vtop’ it should have been connected to the supply VDD12:1.200, instead of the wrong supply VDD10:1.000

Default Severity Label

Recommended

Rule Group

Supply Net Connection

Reports and Related Files

None

LPPLIB17

Checks if all the power switches in the design are correctly connected

When to Use

In power managed designs, each voltage domain and each power domain requires a separate supply net. Therefore, it is important to make sure that each gate in the post-route design is connected to the correct supply. Use this rule for checking supply nets in the Post-layout design phase.

This rule is recommended for use with *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)* and *DEF files and their associated LEF files*.

Description

The *LPPLIB17* rule reports input/output power pins of a powerswitch instance which are not connected to the correct supply rails.

The rule reports the `enable` pins of a power switch instance that are not properly connected to the `-on` and `-on_2` signals. These signals are specified using the `-on` and `-on_2` argument of the *supply* constraint. For example, an active high `enable` connected to an active low `-on` signal is reported by this rule.

Prerequisites

Specify the following information before running this rule:

- Power switches, by using the *power_switch* constraint (which defines the power switch design unit, the `power-out` pin using the `-pwroutpin` argument, the `power-in` pin using the `-pwrinpin` argument, the `enable` pin using the `-en_inv_in` argument, the second `enable` pin using the `-en_inv_in_2` argument in case of dual `enable` power pins, and the `enable` value using the `-enableval` argument
- Power and ground supply, by using the `supply` constraint
- The parent-child relationship between the input and the output supply nets, by using the `-parent` argument of the *supply* constraint on

signals using the -on argument and their active values using the -onval argument

- PLIB files, by using the read_file -type plib <files> project file command or LEF files, by using the read_file -type lef <files> project file command

NOTE: *You can specify the same cell with different power_switch constraints.*

Reported Information

The *LPPLIB17* rule highlights the following:

- An input/output pin of a power switch instance which is not connected
- An input/output pin of a power switch instance which is not connected to the supply net
- An input pin of a power switch instance which is connected to the wrong supply net
- An output pin of a power switch instance which is connected to the always-on supply
- The input and output supply instances which do not have the same value
- An enable pin of a power switch instance which is either unconnected or not connected to the -on signal
- The second enable pin of a power switch instance which is either unconnected or not connected to the -on_2 signal

Language

Verilog, VHDL

Parameter(s)

- *lp_check_ps_loc*: Default value is 0. Set this parameter to 1 to check whether the power switches are located in their switching power domain. Other possible values are no and yes.
- *lp_set_sim_val_x*: Default value is 1. Set this parameter to 0 to propagate the value of the *set_case_analysis* constraint and the -on/-on_2 control signals for the logic lying in the power domain.

- *lp_check_dual_enable_psw*: Default value is `no`. Set this parameter to `yes` to check connection of control pins of dual enable power switch based on the control net, given in the `-control_port` argument of the `create_power_switch` command against the port name that is coming from library. This parameter removes the dependency of order and the `-control_port` argument is decided based on the port name coming from library.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage/ power domains in the design.
- *power_switch* (Mandatory): Use this constraint to specify the power switches.
- *supply* (Mandatory): Use this constraint to specify the supply and ground port names for the LPPLIB rules.
- *set_case_analysis* (Optional): Use this constraint to specify the case analysis conditions used by the rule.
- *always_on_cell* (Optional): Use this constraint to specify the always-on cells and their domain information.
- *always_on_buffer* (Optional): Use this constraint to specify signals that should be driven by an always-on buffer.

CPF Commands

- *create_power_nets* (Mandatory)
- *create_ground_nets* (Mandatory)
- *create_power_domain* (Mandatory)
- *update_power_domain* (Mandatory)
- *create_power_switch_rule* (Mandatory)
- *update_power_switch_rule* (Mandatory)
- *define_power_switch_cell* (Mandatory)
- *define_always_on_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *create_power_switch* (Mandatory)
- *map_power_switch* (Mandatory)

Messages and Suggested Fix

Message 1

The following messages appear when the <*pin-name*> input/output pin of the <*inst-name*> power switch instance is not connected:

[LPPLIB17_1][Recommended] Output power pin '<pin-name>' of powerswitch '<inst-name>(<cell-name>)' is unconnected.

[LPPLIB17_2][Recommended] Input power pin '<pin-name>' of powerswitch '<inst-name>(<cell-name>)' is unconnected.

For information on debugging, click *How to Debug and Fix*.

Message 2

The following messages appear when the <*pin-name*> input/output pin of the <*inst-name*> power switch instance is not connected to the supply net:

[LPPLIB17_3][Recommended] Output power pin '<pin-name>' of powerswitch '<inst-name>(<cell-name>)' is not connected to supply net.

[LPPLIB17_4][Recommended] Input power pin '<pin-name>' of powerswitch '<inst-name>(<cell-name>)' is not connected to supply net.

For information on debugging, click *How to Debug and Fix*.

Message 3

The following message appears when the <*pin-name*> input pin of the <*inst-name*> power switch instance is connected to the wrong supply net:

[LPPLIB17_5][Recommended] Input power pin '<pin-name>' of powerswitch '<inst-name>(<cell-name>)' is connected to supply net '<supply-name1>' in place of supply net '<supply-name2>'

How to Debug and Fix

For information on debugging, click *How to Debug and Fix*.

Message 4

The following message appears when the <*pin-name*> output pin of the <*inst-name*> power switch instance is connected to the always-on supply:

[LPPLIB17_6][Recommended] Output power pin '<pin-name>' of powerswitch '<inst-name>(<cell-name>)' should not be connected to always-on supply.

For information on debugging, click *How to Debug and Fix*.

Message 5

The following message appears when the <*supply-name1*> input supply instance and the <*supply-name2*> output supply instance do not have the same value:

[LPPLIB17_7][Recommended] Input supply '<supply-name1>(<supply-value1>)' and output supply '<supply-name2>(<supply-value2>)' should have the same value.

For information on debugging, click *How to Debug and Fix*.

Message 6

The following messages appear when the <*pin-name*> enable pin of the <*inst-name*> power switch instance is either unconnected or not connected to the specified enable condition signal:

[LPPLIB17_8][Recommended] Enable pin '<pin-name>' of powerswitch '<inst-name>(<cell-name>)' is unconnected.

[LPPLIB17_9][Recommended] Enable pin '<pin-name>' of powerswitch '<inst-name>(<cell-name>)' is connected to signal

'<sig1-name>' in place of the enable condition signal '<sig2-name>'.

For information on debugging, click *How to Debug and Fix*.

Message 7

The following message appears when the <inst-name> power switch instance is located in the <name> domain instead of its <domain-name-list> switching power domains:

[LPPLIB17_10][Recommended] Powerswitch '<inst-name>(<cell-name>)'is located in wrong domain '<name>' in place of powerdomain(s) '<domain-name-list>'

For information on debugging, click *How to Debug and Fix*.

Message 8

The following message appears when the <cellname> power switch instance <inst-name> is located incorrectly in the <name> domain when it is not switching any power domain:

[LPPLIB17_11][Recommended] Powerswitch '<inst-name>(<cell-name>)'is incorrectly located in domain '<name>'

For information on debugging, click *How to Debug and Fix*.

Message 9

The following message appears when the <pin-name> enable pin of the <inst-name> power switch instance has the <actual-value> value instead of the <expected-value> expected value:

[LPPLIB17_12][Recommended] Enable pin '<pin-name>' of powerswitch '<instname>(< cell-name>)' has value <actual-value> instead of the specified value <expected-value>

For information on debugging, click *How to Debug and Fix*.

Message 10

The following message appears when the <net-name> supply net connected to the <pin-name> output power pin of the <inst-name> power switch instance does not have the <field-name> expected field:

[LPPLIB17_13][ERROR] Supply '<net-name>' connected to output power pin '<pin-name>' of powerswitch '<instname>(<cell-name>)' should have '<field-name>' field

For information on debugging, click *How to Debug and Fix*.

Message 11

The following message appears when an associated power switch instance is not found in the design:

[LPPLIB17_14][ERROR] Power switch strategy '<pwr-strategy-name>' is not applied : No associated power switch instance found in the design

Potential Issues

When the power pin of a power switch is unconnected or connected to a signal net, instead of a supply net, the power switch will break the power rail connectivity.

If the input supply pin is connected to the local, or switched, supply and the output supply pin is connected to a main, always-on, supply rail, it is probably due to swapping of the supply rail connection.

If the enable pin is not connected or is connected to a wrong enable net, the power switch is probably not enabled at the correct time.

Consequences of Not Fixing

The design would not function because there would be a break in the supply rail connection.

How to Debug and Fix

The violation is reported at the place where a power switch is instantiated.

For a graphical view of the violation, double-click the message and then click **Incremental Schematic**. Turn on Power View. The schematic shows the following:

- The (incorrect) power switch instance
- The voltage domain, where this power switch instance lies
- (Optional) The output power pin and its connection with the supply net (with value)

- (Optional) The input power pin and its connection with the supply net (with value)
- (Optional) The path of the `enable` pin of the power switch
- (Optional) The valid power switch enable net with correct enable value

To view a sample of the schematic generated by this message, refer to the *Example Code and/or Schematic* section.

To resolve the violation, ensure that the enable of powerswitch is connected to the `-on` signal, the input power pin is connected to the `-parent` supply, and its output power pin is connected to the corresponding supply. Update the Post-layout netlist file accordingly.

For *Message 11*, ensure that an associated power switch instance is present in the design.

Review the information in the *lp_vd_info* and the *lp_constr_info* reports. For more information on these reports, refer to the *Reports and Related Files* section.

Example Code and/or Schematic

Example 1

Consider the following example where a power switch is defined using the *supply* constraint:

```
supply -name VDDA -value 1.2 -alwayson 1
supply -name VDD -value 1.2 -parent VDDA -on ENA
supply -name VSS -value 0
power_switch -name R120_ASLSIL120X160 -pwroutpin VDDO   -
pwrinpin VDDI -en_inv_in EN1 -enableval 0
  -en_inv_in_2 EN2 -enableval_2 0
```

The rule checks whether the `enable` pin is connected to the `-on` and `-on_2` signals correspondingly (`ENA`), the input power pin is connected to the `-parent` supply (`VDDA`), and the output power pin is connected to the corresponding supply (`VDD`).

Example 2

Consider another example showing the `VDDC` power pin connected to the `VDD` supply pin. The SpyGlass Design Constraints file is as follows:

```
current_design top
supply -name VDD -value 1.2 -alwayson 1
supply -name VDDA -net ua.VDDA -value 1.2 -alwayson 0   -
parent VDD -on enA1 enA2
supply -name VDDB -value 1.2 -alwayson 0 -parent VDD   -on
enB
voltage_domain -name Vtop -value 1.2 -modname top       -
supplyname VDD
voltage_domain -name VA -value 1.2 0 -instname top.ua
    -supplyname VDD VDDA -noisosig
voltage_domain -name VB -value 1.2 0 -instname top.ub \
    -supplyname VDD VDDB -noisosig
retention_cell -name RETFF -domains VA VB -vddcpin VDDC
always_on_buffer -name AONBUF -vddcpin VDDC
power_switch -name PSWITCH -en_inv_in EN -pwrinpin VI \
-pwroutpin VO     \
```

After running the analysis, the following message is displayed:

```
Enable pin 'EN' of powerswitch 'top.u1(PSWITCH)' is connected
to signal 'enA1' in place of the -on signal 'enB'
```

The output power pin of the power switch is connected to the `VDDB` supply, which has enable signal `enB`. The enable pin is connected to an incorrect signal `enA1` instead of `enB`.

Example 3

The following violation occurs when the `VSSIN` input power pin is connected to the `vdd` supply net instead of the `vss` supply net:

```
Input power pin 'VSSIN' of powerswitch 'top.p234(POWERSW)' is
connected to supply net 'vdd' in place of supply net 'vss'
```

The incremental schematic is displayed as shown below.

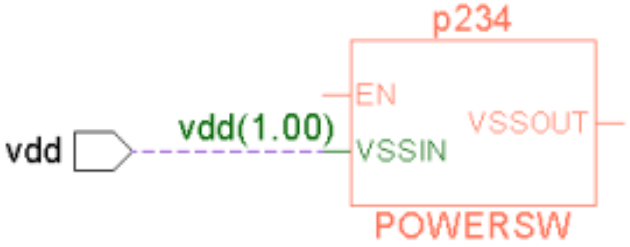


FIGURE 162. Incremental schematic

The cell shown above is a Footer power switch cell and should have its input supply pin `VSSIN` connected to the always-on ground supply net. It is incorrectly connected to the power supply net `vdd` instead.

Default Severity Label

Recommended

Rule Group

LayoutPowerConnectivityRules

Reports and Related Files

- *lp_vd_info*: Contains information about power switches on switched supply and their enables.
- *lp_constr_info*: Refer to the section that lists details about the supply rails. It presents the following information:
  - List of valid power switch enable signals for the corresponding supply net in the `on` field
  - Valid enable value of the power switch enable nets in `onVal` field

  In addition, refer to the section that lists details about the power switches. It presents the following information:
  - The power switch enable pin in the `enablepin` field
  - The valid value that should reach at the power switch enable in the `enableval` field

LPPLIB18

Ensure correct connections of bias power/ground pins to bias power/ground supply respectively.

Language

Verilog, VHDL

Rule Description

The LPPLIB18 rule checks if bias power/ground pins of all cells in the given block are connected to the correct bias power/ground supply associated with the block, respectively.

The LPPLIB18 rule flags a violation if bias power/ground pin of cell is connected to a different bias/ground supply rail than specified in the constraints.

The LPPLIB18 rule runs the rules *LPPLIB18A* and *LPPLIB18B*.

LPPLIB18A

Ensures correct connections of bias power pin to bias power supply

When to Use

Use this rule for:

- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPPLIB18A* rule checks if bias power pins (specified using the `-biaspower` argument of the *pg_pins_naming* constraint) of all cells in the given domain are connected to the correct bias power supply associated with the domain.

The *LPPLIB18A* rule reports a violation when the bias power pin of cell is connected to a different bias power supply rail than specified in the constraints.

Prerequisites

In SGDC, for every voltage domain, bias power and ground supply names associated with the domain must be specified using the `–biaspowernet` and `-biasgroundnet` arguments of the *voltage_domain* constraint, respectively. In UPF, bias power net and bias ground net can be specified by using the *create_supply_set* command using the -function option. For bias ground net, use -function pwell. For bias power net, use the - function nwell.

Rule Exceptions

The *LPPLIB18A* rule cannot be run in the CPF power format. This rule runs only in SGDC and UPF formats. Support for bias nets is available in UPF 2.0 only. Currently, UPF 1.0 is supported.

Language

Verilog, VHDL

Parameter(s)

- *lp_max_viol_count*: Default value is 1000. Set the value to a positive integer to specify the maximum number of messages to be reported by the rule.
- *lp_check_same_biasnet*: Default is 0. Set this parameter to 1 to check if the connection of the power pin and the related bias power pin are the same.

Constraint(s)

- *voltage_domain* (Mandatory): Use this constraint to specify the bias power and ground supply names.
- *supply* (Mandatory): Use this constraint to specify the name of the supply rail pin.
- *pg_pins_naming* (Mandatory): Use this constraint to specify the bias power pins.

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *create_supply_set* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears, when the bias power pin *<pinname>* of power switch instance *<inst-name>* is not connected to a bias net:

[LPPLIB18A_1][RECOMMENDED] Bias power pin '<pin-name>' for cell '<instname>(<cell-name>)' is not connected to bias net

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears, when a bias power pin *<pin-name>*

exists in a power domain <*domain-name*> of power switch instance <*inst-name*>, but a bias net is not defined:

[LPPLIB18A_2][RECOMMENDED] Cell '<inst-name> (<cell-name>' with bias power pin '<pin-name>' is present in domain '<domain-name>' without bias power net defined

For debugging information, click *How to Debug and Fix*.

Message 3

The following messages appear when the bias power pin <*pinname*> of power switch instance <*inst-name*> is connected to an incorrect bias net <*net1-name*> instead of <*net2-name*>:

[LPPLIB18A_3][RECOMMENDED] Bias power pin '<pin-name>' for cell '<instname> (<cell-name>' is connected to incorrect net '<net1-name>' instead of bias net '<net2-name>'

For debugging information, click *How to Debug and Fix*.

Message 4

The following messages appear when the bias power pin <*bias-pin-name*> and power pin <*power-pin-name*> of cell <*inst-name*>*(cell-name*> is not connected to the same supply:

[LPPLIB18A_4][RECOMMENDED] Bias power pin '<bias-pin-name>' and power pin '<power-pin-name>' for cell '<instname>(<cell-name>)' is not connected to same supply

For debugging information, click *How to Debug and Fix*.

Message 5

An informational message appears when the violation count of this rule exceeds the limit set by the *lp_max_viol_count* parameter. Refer to *Message 5* for the message and, for debugging information, refer to *How to Debug and Fix*.

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The consequences of not fixing the violations are as follows:

Message 1: As bias power pin is not connected to a bias net. This is a design flaw.

Message 2: As bias power net is not specified in UPF. This is a flaw in the power intent specified.

Message 3: As bias power pin is connected to an incorrect bias net. This is a design flaw.

Message 4: As the bias power pin and the power pin are not connected to the same supply, it is a design flaw.

How to Debug and Fix

The violation is reported at the place where the cell is instantiated.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To resolve these violation messages, ensure that the bias power pin of the cell is connected to the bias power supply associated with the domain.

Example Code and/or Schematic

Example 1

Consider the following example. A violation occurs when the bias power pin `VDDBIAS` is connected to a non-biasable net `gnd` instead of *<bias-net>*:

```
[RECOMMENDED] Bias power pin 'vddbias' for cell
'inst2(HS45_LS_FILLERSNPW5' is connected to incorrect net 'gnd'
instead of bias net '<bias-net>'
```

The schematic generated is as follows:

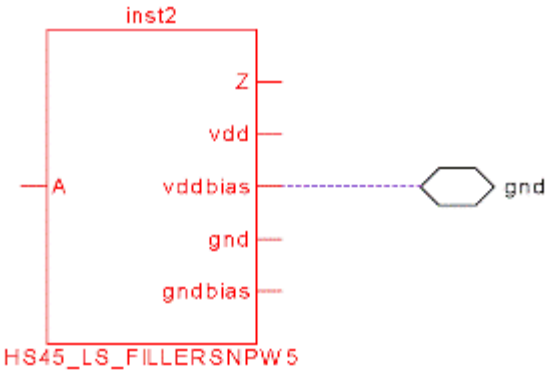


FIGURE 163. Incremental schematic

The schematic highlights the following:

- Bias power pin of a cell that is not connected to a bias net.
- Cell with bias power pin instantiated in a domain without bias power net defined.

Example 2

Consider the following example where power supply is defined using the *supply* constraint:

```
voltage_domain -name V1 -value 1.2 -instname
mid.t1.inst1 -biaspowernet VDDB -biasgroundnet VSSB
voltage_domain -name V2 -value 1.6 -instname
mid.t1.inst4 -biaspowernet VDDBEXT -biasgroundnet
VSSBEXT
supply -name VDDB -value 1.2
supply -name VDDBEXT -value 1.6
supply -name VSSB -value 0
supply -name VSSBEXT -value 0
```

In this example, the constraints specify that the bias power supply *VDDB*

and bias ground supply *VSSB* are associated with the voltage domain *V1* and bias power supply *VDDBEXT* and bias ground supply *VSSBEXT* are associated with the voltage domain *V2*.

Default Severity Label

Recommended

Rule Group

Supply Net Connection

Reports and Related Files

None

LPPLIB18B

Ensures correct connections of bias ground pin to bias ground supply

When to Use

Use this rule for:

- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPPLIB18B* rule checks if bias ground pins (specified using the `-biasground` argument of the *pg_pins_naming* constraint) of all cells in the given domain are connected to the correct bias ground supply associated with the domain.

The *LPPLIB18B* rule reports a violation, if bias ground pin of cell is connected to a different bias ground supply rail than specified in the constraints.

Prerequisites

In SGDC, for every voltage domain, bias power and ground supply names associated with the domain must be specified using the `–biaspowernet` and `-biasgroundnet` arguments of the *voltage_domain* constraint, respectively. In UPF, bias power net and bias ground net can be specified by the *create_supply_set* command using the -function option. For bias ground net, use -function pwell. For bias power net, use the - function nwell.

Rule Exceptions

The *LPPLIB18B* rule cannot be run in the CPF power format. This rule runs only in SGDC and UPF formats. Support for bias nets is available in UPF 2.0 only. Currently, UPF 1.0 is supported.

Language

Verilog, VHDL

Parameter(s)

- *lp_max_viol_count*: Default value is 1000. Set the value to a positive integer to specify the maximum number of messages to be reported by the rule.
- *lp_check_same_biasnet*: Default is `0`. Set this parameter to `1` check if the connection of the ground pin and the related bias ground pin are the same.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use this constraint to specify the bias power and ground supply names.
- *supply* (Mandatory): Use this constraint to specify the names of the supply pins.
- *pg_pins_naming* (Mandatory): Use this constraint to specify the bias ground pins.

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *create_supply_set* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears, when the bias ground pin <*pinname*> of powerswitch instance <*inst-name*> is not connected to a bias net:

[LPPLIB18B_1][RECOMMENDED] Bias ground pin '<pin-name>' for cell '<instname>(<cell-name>)' is not connected to bias net

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears, when a bias ground pin *<pin-name>* ground exists in a power domain *<domain-name>* of powerswitch instance *<inst-name>*, but a bias net is not defined:

[LPPLIB18B_1][RECOMMENDED] Cell '<inst-name>(<cell-name>' with bias ground pin '<pin-name>' is present in domain '<domain-name>' without bias ground net defined

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears, when the bias ground pin *<pinname>* of powerswitch instance *<inst-name>* is connected to an incorrect bias net *<net1-name>* instead of *<net2-name>*:

[LPPLIB18B_1][RECOMMENDED] Bias ground pin '<pin-name>' for cell '<instname> (<cell-name>' is connected to incorrect net '<net1-name>' instead of bias net '<net2-name>'

For debugging information, click *How to Debug and Fix*.

Message 4

The following messages appear when the bias ground pin *<bias-pin-name>* and ground pin *<gnd-pin-name>* of cell *<inst-name>(cell-name>* is not connected to the same supply:

[LPPLIB18B_1][RECOMMENDED] Bias ground pin '<bias-pin-name>' and ground pin '<gnd-pin-name>' for cell '<instname>(<cell-name>)' is not connected to same supply

For debugging information, click *How to Debug and Fix*.

Message 5

An informational message appears when the violation count of this rule exceeds the limit set by the *lp_max_viol_count* parameter. Refer to *Message 5* for the message and, for debugging information, refer to *How to Debug and Fix*.

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The consequences of not fixing the violations are as follows:

Message 1: As bias power pin is not connected to a bias net. This is a design flaw.

Message 2: As bias power net is not specified in UPF. This is a flaw in the power intent specified.

Message 3: As bias power pin is connected to an incorrect bias net. This is a design flaw.

Message 4: As the bias ground pin and the ground pin are not connected to the same supply, it is a design flaw.

How to Debug and Fix

The violation is reported at the place where the cell is instantiated.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix these violation messages, ensure that the bias ground pin of the cell is connected to the bias power supply associated with the domain.

Example Code and/or Schematic

Example 1

Consider the following example. A violation occurs when the input ground pin `gndbias` is connected to a non-biasable net `gnd_extra1` instead of *<bias-net>:*

```
[RECOMMENDED] Bias ground pin 'gndbias' for cell
'inst1(HS45_LS_FILLERSNPW5)' is connected to incorrect net
'gnd_extra1' instead of bias net '<bias-net>'
```

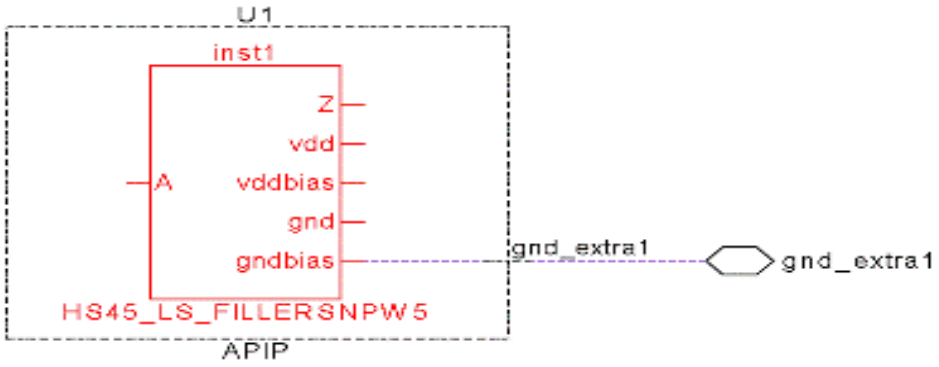


FIGURE 164. Incremental schematic

The *schematic* highlights the following:

- Bias ground pin of a cell that is not connected to a bias net.
- Cell with bias ground pin instantiated in a domain without bias ground net defined.

Example 2

Consider the following example where power supply is defined using the *supply* constraint:

```
voltage_domain -name V1 -value 1.2 -instname
mid.t1.inst1 -biaspowernet VDDB -biasgroundnet VSSB
voltage_domain -name V2 -value 1.6 -instname
mid.t1.inst4 -biaspowernet VDDBEXT -biasgroundnet
VSSBEXT
supply -name VDDB -value 1.2 supply -name VDDBEXT -value 1.6
supply -name VSSB -value 0
supply -name VSSBEXT -value 0
```

In this example, the constraints specify that the bias power supply VDDB and bias ground supply VSSB are associated with the voltage domain V1

and bias power supply VDDBEXT and bias ground supply VSSBEXT are associated with the voltage domain V2.

Default Severity Label

Recommended

Rule Group

Supply Net Connection

Reports and Related Files

None

LPPLIB19

Checks whether bias net is always-on with respect to power net

When to Use

Use this rule for:

- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPPLIB19* rule checks whether the bias net is always-on with respect to the power net.

This rule has the following subrule:

- *LPPLIB19A*: Checks whether the bias net is less always-on than the power net.

LPPLIB19A

Checks whether the bias_net is less always-on than the power net

When to Use

Use this rule for:

- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPPLIB19A* rule reports violations for instances when the `bias_net` connected to `bias_pin` of an instance is less always-on than the power net connected to `power_pin` (of *pg_type* primary_power | backup_power).

Prerequisites

In SGDC, for every voltage domain, bias power and ground supply names associated with the domain must be specified using the `biaspowernet` and `biasgroundnet` arguments of the *voltage_domain* constraint, respectively. In UPF, bias power net and bias ground net can be specified by the *create_supply_set* command using the `-function` option. For bias ground net, use `-function pwell`. For bias power net, use the `-function nwell`.

Rule Exceptions

The *LPPLIB18B* rule runs only in the UPF format.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_set* (Mandatory)
- *create_supply_port* (Mandatory)
- *create_supply_net* (Mandatory)
- *create_pst* (Mandatory)
- *set_domain_supply_net* (Mandatory)

Messages and Suggested Fix

The following message appears when the supply net connected to the bias pin is less always-on than the supply net connect to the power or ground pin:

[LPPLIB19A_1][ERROR] Supply net '<sup-name1>' connected to bias power pin '<pin-name1>' of instance <inst-name>(<mod-name>) is less always-on than the supply net '<sup-net2>' connected to power/ground pin '<pin-name2>'

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

This is a design error that can lead to silicon failure.

How to Debug and Fix

The violation is reported at the place where the cell is instantiated.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix these violation messages, ensure that the supply net connected to the bias pin is more always-on than the supply net connected to the power pin.

Example Code and/or Schematic

In this example, the *LPPLIB19A* rule reports a violation if VDDB is

connected to the bias pin and VDTOP is connected to the power pin of an instance since the PST has such a state in which VDDB can go OFF relative to VDTOP.

```
create_supply_port VDTOP
add_port_state VDTOP -state {VT 1.2}
create_supply_port VDDB
add_port_state VDDB -state {V2 1.4} -state {V2_off off}
create_pst PST -supplies { VDTOP VDDB }
add_pst_state PS2 -pst PST -state { VT V2_off }
```

Default Severity Label

Error

Rule Group

Supply Net Connection

Reports and Related Files

None

LPPLIB20

Checks whether library cell instantiated in design is used out of place

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

Library cells that are associated to specific domains through the *associate_lib* constraint should be used only within those domains. The LPPLIB20 rule reports a violation if the associated cells are used out of place with respect to the `associate_lib` constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *associate_lib*

Messages and Suggested Fix

The following message appears when a library cell instantiated in design has been used out of place with respect to the association of library cells with specific domains:

[LPPLIB20_1][ERROR] Cell '<instance- hierarchical-name>' (‘<cell-name’) should not be instantiated in domain ‘<domain-name>

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

This is a design error that can lead to silicon failure.

How to Debug and Fix

The violation is reported at the place where the cell is instantiated.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation message, ensure that the a valid library cell has been instantiated in a specific domain according to the *associate_lib* constraint.

Example Code and/or Schematic

Consider the following SGDC snippet.

```
associate_lib -domain V3 -cell "BUFF"
```

The following violation message is reported because library cell `BUFF1` instantiated as `top.u1_inst.u2_inst.u3_inst.buff` in domain `V3` has been used out of place, according to the contraint:

```
Cell 'top.u1_inst.u2_inst.u3_inst.buff'('BUFF1') should not be instantiated in domain 'V3'
```

The following is the schematic for this example:

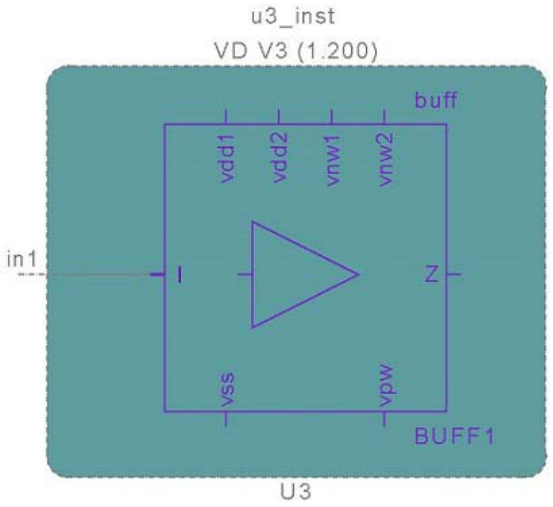


FIGURE 165. Incremental schematic

Default Severity Label

Error

Rule Group

Supply Rules

Reports and Related Files

None

LPSUP01

Reports standard cells that drive supply nets

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSUP01* rule reports instances, other than power switches, which have an output connected to a supply net.

Prerequisites

By default, the LPSUP01 rule is not run while running the SpyGlass Power Verify solution. To enable this rule, specify the following command in the project file: set_goal_option addrules LPSUP01

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *supply* (Mandatory): Use to specify the supply nets.

Messages and Suggested Fix

The following message appears when the pin <pin-name> of the instance <inst-name>, which is not a power switch, drives the supply net <net-name>:

[LPSUP01_1][WARNING] Pin '<pin-name>' of the instance '<cell-name>'

('<inst-name>') is driving the supply net '<supname>'

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

The supply net is being driven by an instance other than a power switch. This is a design flaw.

How to Debug and Fix

The violation is reported at the standard cell instance, which is driving a supply net.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the *Example Code and/or Schematic* section for an example.

To resolve this violation, ensure that the supply nets exist as a top supply port or as the output of a power switch.

Example Code and/or Schematic

Consider the following example. A violation appears when the pin `Y` of the instance `TOP.high_domain.iso2`, which is not a power switch, drives the supply net `TOP.high_domain.VDDH`:

```
[WARNING] 'Y' of the instance 'TOP.high_domain.iso2'
('ISO_AND_EN') is driving the supply net 'TOP.high_domain.VDDH'
```

The schematic is as follows:

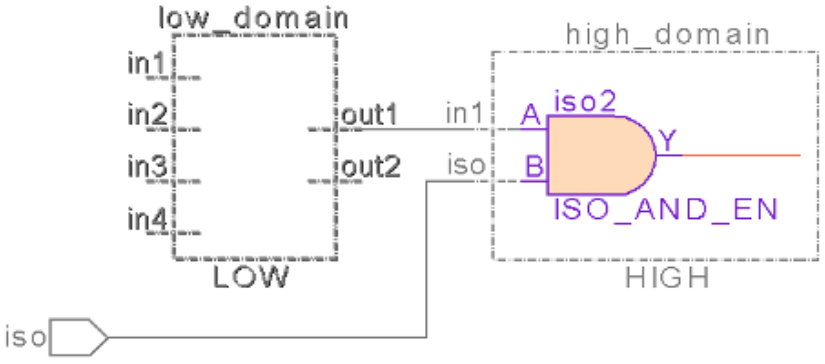


FIGURE 166. Incremental schematic

Default Severity Label

Warning

Rule Group

Supply Net Connection

Reports and Related Files

None

LPSUP03

Checks for consistency between supply net specified for leaf level signal pins and the related power and ground pin

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*

Description

The *LPSUP03* rule performs the following checks, depending on the supply net, after you constrain the subsystem ports with a power or ground supply net by using the *set_related_supply_net* UPF command:

- **Power Supply Nets**: This rule checks that the supply associated with the signal pin has the same operating voltages as that of the supply of the related power pin.
- **Power and Ground Supply Nets**: This rule checks that the supply associated with the signal pin is equivalent or more always-on to the supply associated with the related power or ground pins.

Prerequisites

By default, the *LPSUP03* rule is not run while running the SpyGlass Power Verify solution. To enable this rule, specify the following command in the project file:

```
set_goal_option addrules LPSUP03
```

Parameter(s)

lp_check_name_based_conflict: Default value is `no`. Set the value to `yes` to enable the *LPSUP03* rule report a violation if supplies specified in the `set_related_supply_net` command and the `connect_supply_net` commands have different names.

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *create_pst* (Mandatory)
- *add_pst_state* (Mandatory)
- *set_related_supply_net* (Mandatory)

Messages and Suggested Fix

The following message appears when the supply nets specified in the *set_related_supply_net* UPF command are inconsistent:

[LPSUP03_1][ERROR] Signal pin '<Pin1>' of instance '<inst1>' is constrained with '<power/ground>' supply net '<net1>' is incompatible with supply net '<net2>' associated to related '<power/ground>' pin '<Pin2>' (Reason : <reason>)

Where, <*reason*> can be:

- Operating voltages do not match
- Signal pin supply is less always-on
- Operating voltages do not match and also signal pin supply is less always-on

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

When the supply associated with the signal pin is not equivalent or is less always-on when compared to the supply associated with the related power/ground pins, the signal pin might go OFF when other pins are ON. You will then need to isolate the signal pin. This leads to a design flaw.

In addition, it is a design flaw to have a supply associated with the signal pin that has different operating voltages as that of the supply of the related power pin.

How to Debug and Fix

The signal pin, instance, and supply nets stated in the violation message are highlighted in the Atrenta Console GUI.

To fix the violation, review the reason states in the violation message and accordingly make changes in the UPF file.

Example Code and/or Schematic

In this example, the LPSUP03 rule checks VPD2 and VSS2 for equivalence with the related power and related ground pins of the u1_inst/LS1/A signal pin. If they are not equivalent, it reports a violation.

```
set_related_supply_net -ground {VSS2} -power {VDP2}  -
object_list {u1_inst/LS1/A}
```

Default Severity Label

Error

Rule Group

Supply Net Connection

Reports and Related Files

None

Automatic Fix Rules

The Automatic Fix rules are as follows:

Rule	Purpose
LPSVM23	Generates information about missing isolation logic
LPSVM30	Generates information about missing level shifters

LPSVM23

AutoFix missing isolation logic

When to Use

Use this rule for *RTL description files with or without library files*.

Description

This rule generates isolation logic modules for power domains, where isolation logic is missing.

Prerequisites

To run the *LPSVM23 rule, specify:*

- The power domains with missing isolation logic with additional information using the *voltage_domain* constraint.
- The steady-state condition for each power domain without isolation logic using the *domain_outputs* constraint.
- The corresponding isolation cell information using the *isolation_cell* constraint, if required.
- The power domain-to-voltage domain crossings and power domain-to-power domain crossings using the *ignore_crossing* constraint. You can specify the power domain relationships and automatically find all the *ignore_crossing* constraints using the power_state constraint.
- Set the *lp_is_gate_level* rule parameter if the source design is a gate-level netlist.
- (Optional) Set the *lp_iso_module_name* rule parameter to specify the file name prefix for the generated Verilog files containing the isolation cell instantiations.

Refer to the Example Code and/or Schematic section for an example that illustrates Steps 1, 2, and 3.

Language

Verilog

Parameter(s)

- *lp_is_gate_level:* Default value is `0`. Set the value to `1` to specify the design type as RTL or gate-level netlist. Other values can be `yes` or `no`.
- *lp_iso_module_name*: Default value is not set. It specifies the file name prefix for the Verilog files containing the isolation cell instantiations as generated by this rule.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory)*:* Use to specify the voltage/power domains in the design and its information is used by *SP_01* rule of the SpyGlass DFT DSM policy.
- *domain_inputs* (Mandatory): Use to specify the expected values of various inputs of power domain under the power down condition.
- *isolation_cell* (Optional): Use to specify the isolation cells in power domains.
- *ignore_crossing* (Optional): Use to specify the power domain-to-voltage domain crossings and power domain-to-power domain crossings that should be ignored that need to be ignored.

Messages and Suggested Fix

The following message appears:

[INFO] Isolation logic generated for power domain '<pd-name>'. See file 'isocell_island_module.v' and modified files in $CWD/<vdb name>_reports/lowpower directory and the lp_vd_info report.

Potential Issues

There are no potential issues.

Consequences of not Fixing

This is an informational message. It generates the missing isolation logic for power domains where isolation logic is missing. There are no consequences of not fixing as such.

How to Debug and Fix

Double-click the violation message. The *lp_vd_info* report appears. This report contains the missing isolation logic for power domains in the design.

Example Code and /or Schematic

In this example, Steps 1, 2, and 3 that are required run the LPSVM23 rule are explained in greater detail.

Step 1: Specify the power domains with missing isolation logic with additional information using the *voltage_domain* constraint.

Consider the following example that defines the power domain V3 that does not have isolation logic:

```
voltage_domain
-name V3 -value 1.2 0
-instname top.u6
-isosig isosig6
-isoval 1
-generate_iso_logic
-outputs PD_V3_OUT
```

Here, the -generate_iso_logic argument enables generation of missing isolation logic and the -outputs argument defines the steady state condition as PD_V3_OUT.

Step 2: Specify the steady-state condition for each power domain without isolation logic using the *domain_outputs* constraint.

Continuing with the above example, the corresponding *domain_outputs* constraint is as follows:

```
domain_outputs
-name PD_V3_OUT
-value top.d_o 0 top.ack_o 0 top.err_o 2 ...
-default 1
[-dest ]
```

Where, the -value argument specifies the values at specified signals for the power domain to reach steady-state. The steady-state value for all other signal is 1 as specified by the -default argument. If the -dest argument is

specified then isolation cells are placed in destination domain hierarchy, otherwise isolation cells are placed in top hierarchy.

Step 3: Specify the corresponding isolation cell information using the *isolation_cell* constraint, if required. By default, SpyGlass uses internal cells as isolation cells and assumes that the steady-state value is active high.

Continuing with the above example, the corresponding *isolation_cell* constraint is as follows:

```
isolation_cell
-iso_enable_val 1
-and_cell RX_ISO_AND
-or_cell RX_ISO_OR
-latch_cell RX_ISO_LATCH
-not_cell RX_NOT
-config 2
```

Here, a value 1 of the -iso_enable_val argument specifies that the steady-state value is active high. The cell names specified with the -and_cell, -or_cell and -latch_cell arguments are the AND cell, OR cell, and Latch Cell to be used as isolation cells respectively. The cell name RX_NOT specified with the -not_cell argument is the NOT cell to be used for changing the polarity of the isolation signal.

A value 2 of the -config argument specifies that the isolation cell instantiation configuration should be output-input-enable.

Default Severity Label

Info

Rule Group

Automatic Fix

Reports and Related Files

- *lp_vd_info*: Contains the missing isolation logic for power domains in the design.

LPSVM30

Generates new level shifters for Verilog RTL description

When to Use

Use this rule for *RTL description files with or without library files*

Description

The *LPSVM30* rule generates new Verilog RTL description by inserting missing level shifters in the original Verilog RTL description. All level shifters will be placed in "island" or sub-block. You can specify the location of level-shifter island using `-locate` field in *levelshifter* constraint. Level shifter insertion information is provides in the *lp_autofix_info* report. If level shifter is not inserted for any crossing then reason is also mentioned in the report.

Prerequisites

The *LPSVM30* rule requires you to specify the suggested level shifters for the voltage domain crossings with missing level shifters by using the *levelshifter* constraint in an SGDC file. The missing level shifters are reported by the *LPLSH05A* and *LPLSH05B* rules.

Language

Verilog

Parameter(s)

None

Constraint(s)

- *voltage_domain* (Mandatory): Use to specify the voltage/power domains in the design.
- *levelshifter* (Mandatory): Use to specify the names of design units to be used as level shifters.

Messages and Suggested Fix

Message 1

The following message appears when no voltage domain crossings are

found in the design:

```
[INFO] No voltage domain crossing found in design, and hence no
levelshifter was inserted
```

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when no *levelshifter* instance could be inserted in the design:

```
[INFO] No level shifter could be inserted. See the
lp_autofix_info report for details
```

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when all required level shifters were inserted in the design:

```
[INFO] Level shifters inserted for all voltage domain
crossings. See Island file and modified files in $dir_name/
LPSVM30 directory and the lp_autofix_info report
```

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears when some of the required level shifters could not be inserted in the design:

```
[INFO] Level shifter inserted for some crossings only. See
Island file and modified files in $dir_name/LPSVM30 directory
and the lp_autofix_info report
```

Potential issues

Not applicable

Consequences of not Fixing

Not applicable

How to Debug and Fix

Double-click the violation messages to review the *lp_autofix_info* report.

Example Code and/or Schematic

When a voltage domain crossing between voltage domains `A` and `B` does

not have the required *levelshifter* instance, the instance(s) of *levelshifter* design unit `LS_A_B` are inserted appropriately in source files.

After the *LPSVM30* rule has been run, you can review the *levelshifter* insertions and use the modified Verilog RTL files as required.

```
current_design top
voltage_domain -name A -value 1.0 -instname mod1
voltage_domain -name B -value 2.0 -instname mod2
levelshifter -name LS_A_B -from A -to B
-inTerm in -outTerm out -enableTerm en
-locate src
```

Default Severity Label

Info

Rule Group

`Automatic Fix`

Reports and Related Files

- *lp_autofix_info*: Contains information about auto insertion of level shifters.

Fine Grain Power Gate (MTCMOS) Rules

The MTCMOS rules are as follows:

:

Rule	Reports
LPSVM33	Non-High VT cell instances found in the fan-out cone of retention latch cell instances
LPSVM33A	Non-High VT cell instances of a power domain found in paths between two ALWAYS-ON voltage domains
LPSVM34	Non-High VT cell instances found in the fan-in cone of save, restore, and clock pins of the retention latch cell instances
LPSVM35	MTCMOS cell instances that have fine grain footer cells connected to them and whose outputs cross from a power domain to an always-on voltage domain without isolation cell instances
LPSVM36	Possible sneak leakage paths due to incorrect MTCMOS cells

LPSVM33

Reports non-High VT cell instances in the fan-out cone

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM33* rule reports all cells instances, other than the instances of specified High VT cells identified in the fan-out of the output pin of instances for the specified retention latch cells.

The LPSVM33 rule can also be run in the CPF/UPF flow. In this flow, you need to specify the *multivt_lib* constraint in the SGDC file and then the voltage domain/retention cell information is retrieved from the corresponding CPF/UPF file.

Prerequisites

The *LPSVM33* rule requires the following:

- Voltage domains and power domains using the *voltage_domain* constraint. This is required as the *LPSVM33* rule does not require isolation information; there is no need to specify the `-isosig` argument while defining a power domain.
- High VT cell libraries specified using the *multivt_lib* constraint as in the following example:

```
multivt_lib
  -type highvt
  -names CORE9GPHS1 CORE9GPHS2
```

- Retention latch cells and their output pin names by using the *retention_cell* constraint as in the following example:

```
retention_cell -name FD1S3AQV15 -qTerm Z
```

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/ power domains in the design and its information is used by voltage and power domain rules.
- *multivt_lib* (Mandatory): Use to specify the VT library groups and their member libraries.
- *retention_cell* (Mandatory): Use to specify the retention latch cells

Messages and Suggested Fix

The following message appears when a non-High VT cell instance <*inst-name*> of the power domain <*pd-name*> (voltage values <*pdvalues*>) is identified in the fan-out of a retention latch cell instance:

[LPSVM33_1][WARNING] Instance '<inst-name>' inside the power domain '<pdname>'(value '<pd-values>') has cells other than HighVt cells in its fan-out cone

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

None

How to Debug and Fix

The violation is reported at the place where a non-High VT cell is instantiated in the fan-out cone of a retention latch cell.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, ensure that the cells at the fan-out cone of a retention

cell are high VT cells.

Example Code and/or Schematic

Consider the following example:

A violation occurs when a non-High VT cell instance is identified in the fan-out of a retention latch cell instance 'CKT_TOP.U1.U1.Int_am_reg_3_V' of power domain 'V2' (voltage values '1.300000')

instance:`Instance 'CKT_TOP.U1.U1.Int_am_reg_3_V' inside the power domain 'V2'(value '1.300000') has cells other than HighVt cells in its fan-out cone`

The schematic highlights the instance of the retention cell, the output of which is connected to a non-high VT cell.

Default Severity Label

Warning

Rule Group

`MTCMOS`

Reports and Related Files

None

LPSVM33A

Checks for non-high VT cells in a power domain

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/ LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM33A* rule reports non-High VT cell instances of a power domain identified in paths between two always-on voltage domains.

The LPSVM33A rule can also be run in the CPF/UPF flow. In this flow, you need to specify the *multivt_lib* constraint in the SGDC file and then the voltage domain/retention cell information is retrieved from the corresponding CPF/UPF file.

Prerequisites

Refer to the prerequisites of the *LPSVM33* rule.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/ power domains in the design and its information is used by voltage and power domain rules.
- *multivt_lib* (Mandatory): Use to specify the VT library groups and their member libraries.
- *retention_cell* (Mandatory): Use to specify the retention latch cells.

Messages and Suggested Fix

The following message appears when a non-High VT cell instance of power domain <*pd-name*> (voltage values <*pdvalues*>) is found in the path from instance <*inst1*> of always-on voltage domain <*vd1-name*> (voltage value <*vd1value*>) to instance <*inst2*> of always-on voltage domain <*vd2-name*> (voltage value <*vd2value*>)

[LPSVM33A_1][WARNING] Path between instance '<inst1>' in always-on domain '<vd1-name>'(value '<vd1-value>') and instance '<inst2>' in always-on domain '<vd2name>'(value '<vd2-value>') has cells other than HighVt cells in the power domain '<pdname>'(value '<pd-values>')

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

The feed-through paths should have always-on buffers. These always-on instances should be low-Vt cells.

How to Debug and Fix

The violation is reported at the place where a non-High VT cell is instantiated in a power domain lying between two always-on domains.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, ensure that the non-High VT cells are not present in a power domain in the path that crosses between two always-on domains.

Example Code and/or Schematic

Consider the following example. A violation occurs when a non-High VT cell instance of power domain 'VPD1'(voltage values '1.20 0') is found in the path from instance 'SE.SE_FE0.SE_SORT0.U210' of always-on voltage domain 'VTOP' (voltage value '1.100000') to instance 'SE.SE_FE0.SE_FMUL00.U4015' of always-on voltage domain 'VTOP' (voltage value '1.100000'):

Path between instance 'SE.SE_FE0.SE_SORT0.U210' in always-on domain 'VTOP'(value '1.100000') and instance

```
'SE.SE_FE0.SE_FMUL00.U4015' in always-on domain 'VTOP'(value
'1.100000') has cells other than HighVt cells in the power
domain 'VPD1'(value '1.20 0')
```

The schematic is as follows:

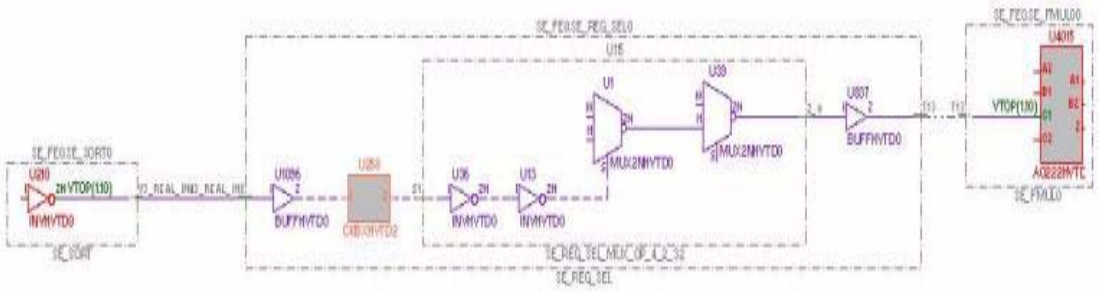


FIGURE 167. Incremental schematic

The schematic highlights the complete path between the two always-on voltage domains in one color and all the non-High VT cell instances present in the path in a different color.

Default Severity Label

Warning

Rule Group

MTCMOS

Reports and Related Files

None

LPSVM34

Reports non-high VT cells identified in the fan-in cone of retention latch cells

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM34* rule reports non-High VT cell instances identified in the fan-in cone of the specified pins for any combination of save, restore, and clock pins of retention latch cell instances.

The LPSVM34 rule can also be run in the CPF/UPF flow. In this flow, you need to specify the *multivt_lib* constraint in the SGDC file and then the voltage domain/retention cell information is retrieved from the corresponding CPF/UPF file.

Prerequisites

Refer to the prerequisites of the *LPSVM33* rule. In addition, for specify retention latch cells and their pins to be checked using the *retention_cell* constraint as in the following example:

```
retention_cell -name FD1S3AQV15 -save SAVE –restore  REST -clk CP
```

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/ power domains in the design and its information is used by voltage and power domain rules.
- *multivt_lib* (Mandatory): Use to specify the VT library groups and their member libraries.
- *retention_cell* (Mandatory): Use to specify the retention latch cells

Messages and Suggested Fix

The following message appears when a non-High VT cell instance <*inst-name*> of power domain <*pd-name*> (voltage values <*pdvalues*>) is found in the fan-in of <*pin-type*> pin of a retention latch cell instance:

[LPSVM34_1][WARNING] Instance '<inst-name>' inside the power domain '<pdname>'(value '<pd-values>') has cells other than HighVt cells in its fan-in cone of '<pin-name>' pin

Where, <*pin-type*> can be `save`, `restore`, or `clk`.

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

The save and restore pins are driven by the control signals. These signals have the always-on instances that should be high-Vt so that they have low leakage.

How to Debug and Fix

The violation is reported at the place where a non-High VT cell instance is instantiated in the fan-in cone of a retention latch cell instance.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, ensure that the cells at the fan-in cone of a retention cell are High VT cells.

Example Code and/or Schematic

Consider the following example. A violation occurs when a non-High VT cell instance `'CKT_TOP.U1.U1.R_STATE_reg_2_V'` of power domain `'V2'` (voltage values `'1.300000'`) is identified in the fan-in of the `'save'` pin of a retention latch cell instance:

```
[WARNING] Instance 'CKT_TOP.U1.U1.R_STATE_reg_2_V' inside the
power domain 'V2'(value '1.300000') has cells other than HighVt
cells in its fan-in cone of 'save' pin
```

The schematic is as follows:

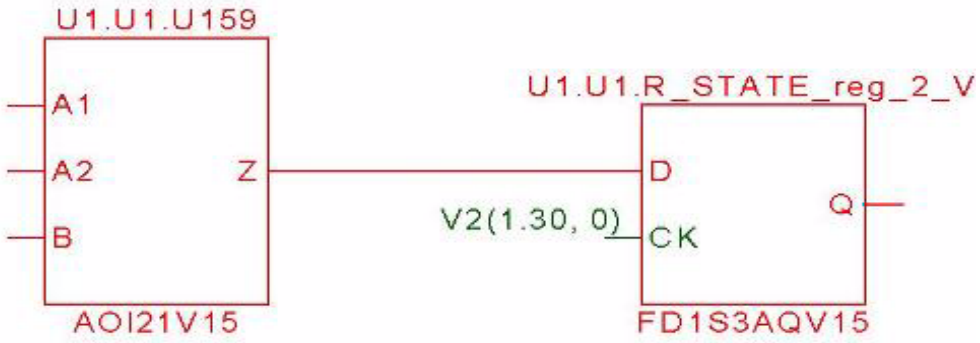


FIGURE 168. Incremental schematic

Default Severity Label

Warning

Rule Group

MTCMOS

Reports and Related Files

None

LPSVM35

Reports MTCMOS cell instances with outputs that are incorrectly isolated

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM35* rule checks for MTCMOS cell instances that have fine grain footer cells connected to them and whose outputs cross from a power domain to an always-on voltage domain without isolation cell instances, as shown in the diagram below:

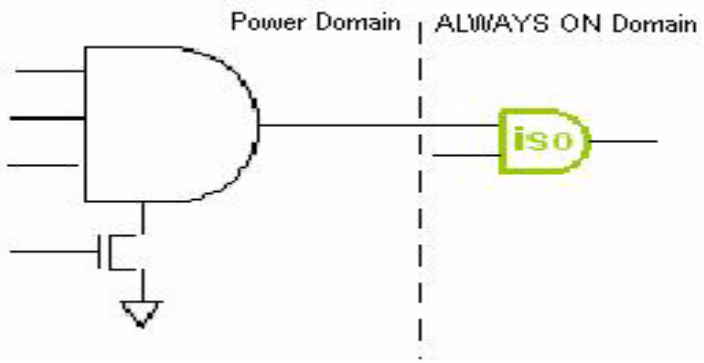


FIGURE 169. Isolation cell instances

The *LPSVM35* rule can also be run in the CPF/UPF flow. In this flow, you need to specify the *multivt_lib* constraint in the SGDC file and then the voltage domain/retention cell information is retrieved from the corresponding CPF/UPF file.

Prerequisites

The *LPSVM35* rule requires the following:

- Voltage domains and power domains using the *voltage_domain* constraint
- MTCMOS cell libraries specified using the *multivt_lib* constraint, such as:

  ```
  multivt_lib -type mtcmosH -names MTCMOS123
  ```

- Power domain isolation information either using the `-isosig/isoval` arguments of the *voltage_domain* constraint or using the *isolation_cell* constraint, as in the following example:

  ```
  isolation_cell -names "And_.*" "Or_.*" "Latch_.*"
  ```

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/ power domains in the design and its information is used by voltage and power domain rules.
- *multivt_lib* (Mandatory): Use to specify the VT library groups and their member libraries.
- *isolation_cell* (Optional): Use to specify the isolation cells in power domains.

Messages and Suggested Fix

The following message appears when the output *<sig-name>* of the power domain *<pd-name>* (voltage values *<pdvalues>*) is not correctly isolated:

```
[LPSVM35_1][WARNING] Output '<sig-name>' of power domain
'<pdname>(<pdvalues>)' must be connected to isolation cell
```

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

If this is not fixed, it means that proper isolation (for the always-on domain) will not take place when the MTCMOS cell placed in power domain goes off.

How to Debug and Fix

The violation is reported at the place where the MTCMOS cell whose outputs are not correctly isolated, is instantiated.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, ensure that outputs of the MTCMOS cell are correctly isolated.

Example Code and/or Schematic

Consider the following example where a violation occurs when the output `'CKT_TOP.DATA_OUT_2'` of power domain `'V2'` (voltage values `'1.30 0'`) is not correctly isolated:

```
[WARNING] Output 'CKT_TOP.DATA_OUT_2' of power domain 'V2(1.30
0)' must be connected to isolation cell
```

The schematic is as follows:

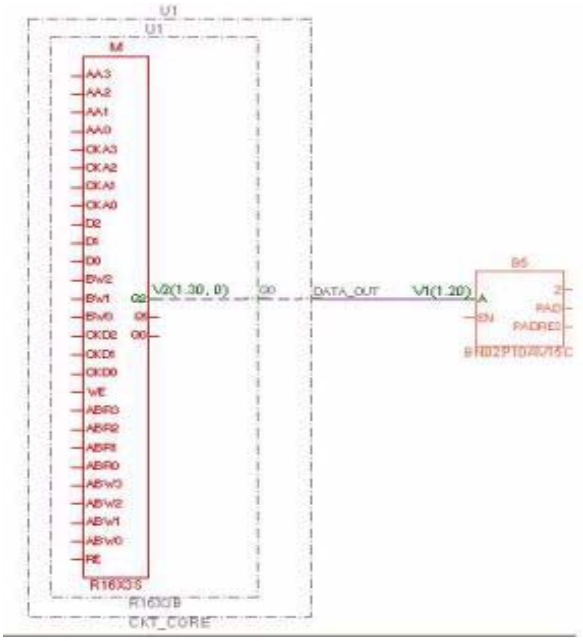


FIGURE 170. Incremental schematic

The schematic highlights the MTCMOS cell the output of which is going to an always-on domain but has no isolation cell placed in between.

Default Severity Label

Warning

Rule Group

MTCMOS

Reports and Related Files

None

LPSVM36

Reports potential sneak leakage paths due to incorrect MTCMOS cells

When to Use

Use this rule to check the possibility of sneak leakage paths due to incorrect MTCMOS cells. Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM36* rule reports potential sneak leakage paths due caused by incorrect MTCMOS cells. A sneak leakage path is a path that exists within or between two or more cells that allow leakage to occur.

The following illustration shows a sneak leakage path between incompatible MTCMOS cells. In this illustration, one cell has a header transistor and the other has a footer transistor.

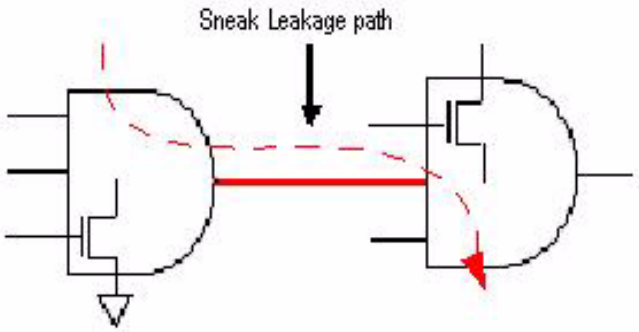


FIGURE 171. Leakage path

Prerequisites

Specify the MTCMOS cell libraries using the *multivt_lib* constraint, as in the following example:

multivt_lib -type mtcmosHF -names MTCMOS123

Language

Verilog, VHDL

Parameter(s)

- *lp_ignore_seqelem_retencell*: Default value is `0`. Set the value to `1` to ignore the sequential elements and retention cell instances from the analysis of sneak leakage paths.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/ power domains in the design and its information is used by voltage and power domain rules.
- *multivt_lib* (Mandatory): Use to specify the VT library groups and the member libraries.

Messages and Suggested Fix

Message 1

The following message appears when the instance *<inst-name>* of an MTCMOS cell instance is connected to a CMOS cell instance or a sequential cell/retention cell, but the MTCMOS cell has only supply-side or ground-side sleep devices:

[LPSVM36_1][WARNING] `Cell '<inst-name>' should have both header and footer to avoid sneak leakage`

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when instance *<inst1-name>* of an MTCMOS cell instance (type *<keyword1>*) is connected to instance *<inst2-name>* of another MTCMOS cell instance (type *<keyword2>*),

but both the cells do not have the exact set of sleep devices:

```
[LPSVM36_2][WARNING] Cell '<inst1-name>' (type <keyword1>)
should be of same type as cell '<inst2-name>' (type <keyword2>)
to avoid sneak leakage
```

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

Message 1: As both header and footer are not present on the MCTMOS cell, there will be sneak leakage. This is a design flaw.

Message 2: As two different kinds of MTCMOS cell instances with a different set of sleep devices are connected to each other, there will be sneak leakage. This is a design flaw.

How to Debug and Fix

The violations are reported at the place where the MTCMOS cell with a possibility of sneak leakage path is instantiated.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix these violations, ensure that:

- MTCMOS cell instances have the supply-side and the ground-side sleep cells when the instance shares an output with a CMOS cell instance, sequential cell, or retention cell, assuming the *lp_ignore_seqelem_retencell* rule parameter is not set.
- When two MTCMOS cell instances are connected, both the MTCMOS cells have exactly the same type of sleep devices.

Example Code and/or Schematic

Consider the following example. A violation occurs when the `'CKT_TOP.B0PU'` instance of an MTCMOS cell is connected to a CMOS cell instance or sequential cell/retention cell, but the MTCMOS cell has only supply-side or only ground-side sleep devices:

```
[WARNING] Cell 'CKT_TOP.B0PU' should have both header and
footer to avoid sneak leakage
```

The schematic is as follows:

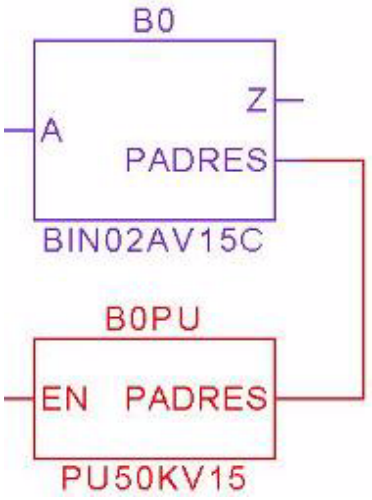


FIGURE 172. Incremental schematic

The schematic highlights two different kinds of MTCMOS cell instances with a different set of sleep devices connected to each other. Such a design scenario will lead to sneak leakage.

Default Severity Label

Warning

Rule Group

MTCMOS

Reports and Related Files

None

Best Design Practices

The Best Design Practices contains the following rules:

Rule	Reports
LPBUS01	This rule has been deprecated
LPBUS02	This rule has been deprecated
LPBUS03	This rule has been deprecated
LPFSM01	This rule has been deprecated
LPFSM02	This rule has been deprecated
LPFSM03	This rule has been deprecated
LPFSM04	This rule has been deprecated
LPFSM09	This rule has been deprecated
LPFSM20	This rule has been deprecated
LPGLT01	This rule has been deprecated
LPXFM01	This rule has been deprecated
LPXFM06	This rule has been deprecated

LPBUS01

This rule has been deprecated

LPBUS02

This rule has been deprecated

LPBUS03

This rule has been deprecated

LPFSM01

This rule has been deprecated

LPFSM02

This rule has been deprecated

LPFSM03

This rule has been deprecated

LPFSM04

This rule has been deprecated

LPFSM09

This rule has been deprecated

LPFSM20

This rule has been deprecated

LPGLT01

This rule has been deprecated

LPXFM01

This rule has been deprecated

LPXFM06

This rule has been deprecated

Special Purpose Rules

The Special Purpose rules are as follows:

Rule	Reports
LPPLIB12	Incorrectly connected RAM switches in RTL and gate-level netlist
LPPSW01	Incorrectly connected Power Switches
LPPSW02	Incorrectly connected Power Switch breaking a daisy chain
LPPSW03	Checks the presence of a delay buffer in the path of the input enable pin of a power switch
LPPSW04	Ensures that the enable signal of a power switch has only AON buffers or delay buffers in its path
LPSVM29	Generates design-level and hierarchical reports that list the cell composition of different VT type libraries
LPSVM37	User-defined cells that are instantiated outside the specified hierarchy
LPSVM41	Incorrect register connections of user-specified power down signals
LPSVM42	Lowpower signals that are being enabled while the design comes out of POR (Power On Reset)
LPSVM43	Incorrect power up and power down sequences of power domains
LPSVM44	User-specified signals that are incorrectly connected
LPSVM45	Incorrect enable logic for power switches in gate-level netlists
LPSVM46	Incorrect connections to RAM switches in RTL and gate-level netlists
LPTIE01	Incorrectly connected Tie cells
LPTIE02	Tie cells that have isolation cells in their path

Rule	Reports
LP_BLACKBOX_CHECK	Checks if percentage of black box elements exceeds the set limit
LP_MULTI_DOMAIN_CROSSING_CHECK	Ensures that signal paths do not have combinational cells in more than two different domains

LPPLIB12

Reports incorrectly connected RAM switches in RTL and gate-level netlist

When to Use

Use this rule for:

- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPPLIB12* rule checks incorrectly connected RAM switches for RTL and gate-level netlist and the correctness of RAM switch instance-to-RAM instance pairing with valid *ram_instance* constraints.

The *LPPLIB12* rule can also be run in the CPF/UPF flow. For this, you need to specify the *ram_instance* and *ram_switch* constraint in the SGDC file, but the voltage domain information is picked up from the corresponding CPF/UPF file.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/power domains and information is used by *SP_01* rule.
- *ram_switch* (Mandatory): Use to specify the RAM switches.
- *ram_instance* (Optional): Use to specify RAM instances.

Messages and Suggested Fix

Message 1

The following message appears when the signal driving the enable pin <*en-pin-name*> of instance <*swt-inst-name*> of RAM switch <*swt-name*> does not originate from an always-on domain:

[LPPLIB12_1][WARNING] Signal driving the enable pin '<en-pin-name>' of the RAM Switch '<swt-name>' instantiated as '<swtinstname>' does not originate from an always-on domain

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when no RAM instance is connected to the VSS-switched pin <*pin-name*> of instance <*swtinstname*> of RAM switch <*swt-name*>:

[LPPLIB12_2][WARNING] RAM Switch '<swt-name>' instantiated as '<swtinstname>' has no RAM connected to its VSS Switched pin '<pin-name>'

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when signal <*sig-name*> connected to the VSS-switched pin <*vss-swt-pin-name*> of instance <*swtinstname*> of RAM switch <*swt-name*> has more than one receiver:

[LPPLIB12_3][WARNING] Signal '<sig-name>' connected to the VSS Switched pin '<vss-swt-pin-name>' of RAM Switch '<swt-name>' instantiated as '<swtinstname>' has more than one receiver

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears when signal <*sig-name*> connected to the VSS-switched pin <*vss-swt-pin-name*> of instance <*swtinstname*> of RAM switch <*swt-name*> has more than one driver:

[LPPLIB12_4][WARNING] Signal '<sig-name>' connected to the VSS Switched pin '<vss-swt-pin-name>' of RAM Switch '<swt-name>'

instantiated as '<swtinstname>' has more than one driver

For debugging information, click *How to Debug and Fix*.

Message 5

The following message appears when signal *<sig-name>* connected to the VSS-switched pin *<vss-swt-pin-name>* of instance *<swtinstname>* of RAM switch *<swt-name>* is connected to instance *<raminstname>* of RAM *<ram-name>* at pin <pin-name> instead of the VSS pin *<ram-vss-pin-name>*:

[LPPLIB12_5][WARNING] Signal '<sig-name>' connected to the VSS Switched pin '<vss-swt-pin-name>' of RAM Switch '<swtname>' instantiated as '<swtinstname>' is going to the RAM '<ramname>' instantiated as '<raminstname>' through pin '<pin-name>' and not through the VSS pin '<ram-vss-pin-name>'

For debugging information, click *How to Debug and Fix*.

Message 6

The following message appears for a *ram_instance* constraint when object *<name>* specified with the `-switch_instance` argument does not exist as a switch instance in the design:

[LPPLIB12_6][WARNING] Object '<name>' specified with the 'switch_instance' option of the 'ram_instance' constraint does not exist as an instance in the design

For debugging information, click *How to Debug and Fix*.

Message 7

The following message appears for a *ram_instance* constraint when object *<name>* specified with the `-ram_instance` argument does not exist as a RAM instance in the design:

[LPPLIB12_7][WARNING] Object '<name>' specified with the '-ram_instance' option of the 'ram_instance' constraint does not exist as an instance in the design

For debugging information, click *How to Debug and Fix*.

Message 8

The following message appears for a *ram_instance* constraint when instance *<inst-name>* specified with the `switch_instance` argument is not a

RAM switch instance in the design:

[LPPLIB12_8][WARNING] Instance '<instname>' given with the 'switch_instance' option of the 'ram_instance' constraint is not a RAM Switch instance

For debugging information, click *How to Debug and Fix*.

Message 9

The following message appears for a *ram_instance* constraint when instance *<inst-name>* specified with the ram_instance argument is not a RAM instance in the design:

[LPPLIB12_9][WARNING] Instance '<instname>' given with the 'ram_instance' option of the 'ram_instance' constraint is not a RAM instance

For debugging information, click *How to Debug and Fix*.

Message 10

The following message appears for switch instance *<swinstname>* when it is not connected to the RAM instance *<ram-inst-name>* in the design as specified in the corresponding *ram_instance* constraint:

[LPPLIB12_10][WARNING] Switch instance '<swtinstname>' is not connected to RAM instance '<raminstname>' specified in the 'ram_instance' constraint

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of not Fixing

The consequences of not fixing the violations are as follows:

Message 1: The signal driving the enable pin of RAM switch doesn't originate from always-on domain, the RAM switch might not get enabled when required.

Message 2: The RAM Switch has no RAM connected to its VSS Switched pin. This makes the RAM Switch useless.

Message 3: The VSS-switched pin of RAM Switch should not have more than one receiver.

Message 4: The VSS-switched pin of RAM switch should not have more than one driver.

Message 5: The VSS-switched pin of RAM switch should be connected to the VSS pin of RAM instance, but is connected to another pin. This is a design flaw.

Message 6: The object specified with the '-switch_instance' argument does not exist as a switch instance in the design. This makes the *ram_instance* constraint useless.

Message 7: The object specified with the '-ram_instance' argument does not exist as a RAM instance in the design. This makes the *ram_instance* constraint useless.

Message 8: The object provided with the 'switch_instance' argument of the *ram_instance* constraint is not a RAM Switch instance. This is a design flaw.

Message 9: The object provided with the 'ram_instance' argument of the *ram_instance* constraint is not a RAM instance. This is a design flaw.

Message 10: The Switch instance is not connected to RAM instance specified using the *ram_instance* constraint. This is a design flaw.

How to Debug and Fix

The violation is reported at the instance of the RAM power switch that is incorrectly connected in RTL or gate-level netlist.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix these violation messages, ensure that the:

- signals driving the enable pin of a RAM switch originate from always-on logic.
- VSS-switched net of a RAM switch is connected to one RAM instance.
- RAM instances in a power domain are connected to a RAM switch of the correct type.

Example Code and /or Schematic

Example 1

In the following example, a violation occurs when signal 'top.mid_inst.w1' connected to the VSS-switched pin 'VSSON' of instance 'top.mid_inst.Pswitch1' of RAM switch 'LL_SLOGICSWITCH20' has more than one receiver:

[WARNING] Signal 'top.mid_inst.w1' connected to the VSS Switched pin 'VSSON' of RAM Switch 'LL_SLOGICSWITCH20' instantiated as 'top.mid_inst.Pswitch1' has more than one receiver

The schematic generated is as follows:

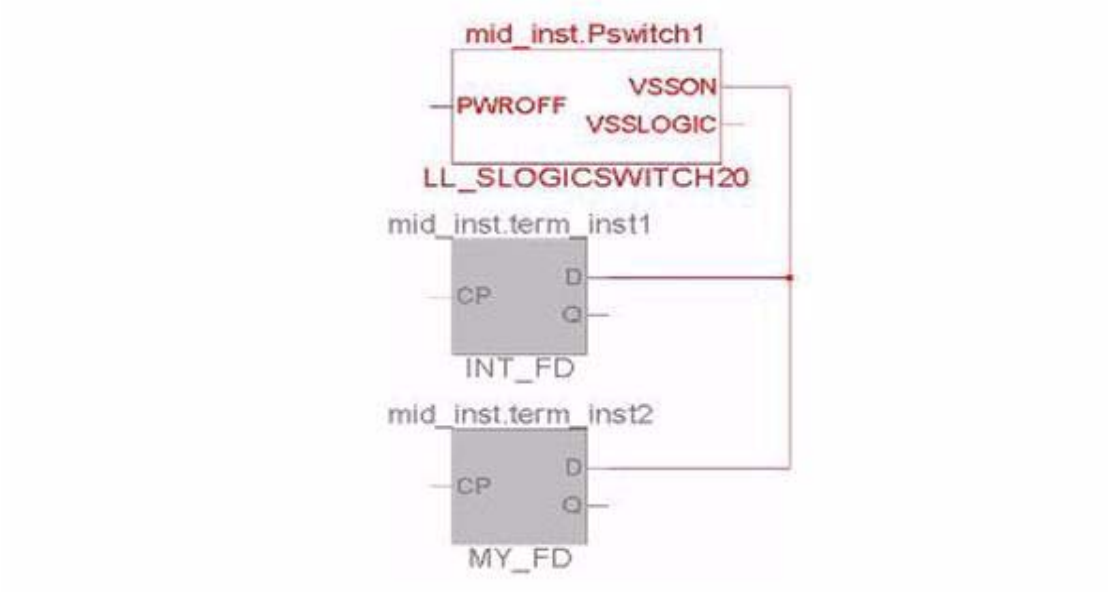


FIGURE 173. Incremental schematic

Schematic Highlight

The *schematic shows the following*:

- Signals driving the enable pin of a RAM switch and not originating from always-on logic.
- The VSS-switched net of a RAM switch, not connected to one RAM instance and,
- RAM instances in a power domain, not connected to a RAM switch of the correct type.

Example 2

Consider the following example code:

```
module top (VDD,VSS,rst,in1,in2,o);
input  VDD,VSS,rst,in1,in2;
output o;
 wire   w1,w2,w3,w4,w5,w6;
wire   vdd_buf;

//Pdomain
mid mid_inst(.PwrIN(VDD),
           .rst(rst),
              .ip(in1),
           .op(o)
             );
endmodule
module mid(PwrIN,rst,ip,op);
input  PwrIN,rst,ip;
output op;
wire   w1,w2,ip;

//assign w1 = ip;

INT_FD   term_inst1(.D(w1),.CP(PwrIN),.Q(op));
INT_FD1   term_inst2(.D(),.CP(),.Q());

//PSwitch
LL_SLOGICSWITCH20 Pswitch1(.PWROFF(),         //expecting
a
en not connected violation
                           .VSSLOGIC(rst),
                             .VSSON(w1)
                            );

LL_SLOGICSWITCH10 Pswitch2(.PWROFF(ip),          //
expecting
a en not connected violation
                            .VSSLOGIC(),
                            .VSSON()
                           );
endmodule
```

In the above example, the rule reports the following violation (**Message 1**) because the signal driving the enable pin PWROFF of instance LL_SLOGICSWITCH20 of RAM switch top.mid_inst.Pswitch1 does not originate from an always-on domain:

```
Signal driving the enable pin 'PWROFF' of the RAM Switch
'LL_SLOGICSWITCH20' instantiated as 'top.mid_inst.Pswitch1'
does not originate from an always-on domain
```

The incremental schematic is displayed as follows:

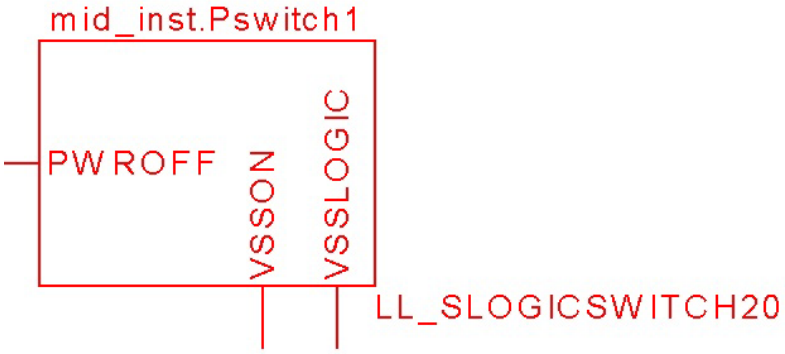


FIGURE 174. Incremental schematic

In the above example code, the rule LPLIB12 also reports the following violation (**Message 2),** because no RAM instance is connected to the VSS-switched pin VSSON of instance top.mid_inst.Pswitch2 of RAM switch LL_SLOGICSWITCH10:

```
RAM Switch 'LL_SLOGICSWITCH10' instantiated as
'top.mid_inst.Pswitch2' has no RAM connected to its VSS
Switched pin 'VSSON'
```

The incremental schematic is displayed as follows:

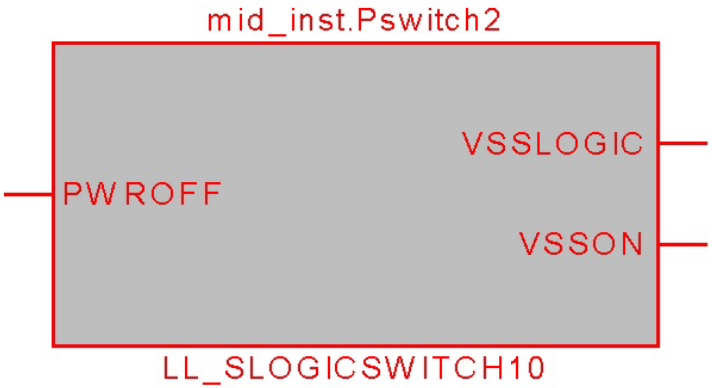


FIGURE 175. Incremental schematic

Default Severity Label

Warning

Rule Group

Special Purpose

Reports and Related Files

- *lp_vd_info*: Reports each RAM switch/RAM instance pair with RAM switch enable pin source cell name.
- *ram_info*: Reports the RAM Switch instance-to-RAM instance information.

LPPSW01

Checks the domain of the signal reaching the input enable of a power switch

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPPSW01* rule reports a violation when the signal reaching the input enable pin of a Power Switch cell is not from a relatively always-on domain.

A library cell specified with *switch_cell_type* attribute is treated as a Power Switch cell.

Prerequisites

By default, the *LPPSW01* rule is not run while running the SpyGlass Power Verify policy. To enable this rule, select the *LPISO01* rule in the Atrenta Console GUI.

Rule Exceptions

The *LPPSW01* rule can be run on the pre-layout netlist.

Language

Verilog, VHDL

Parameter(s)

- *lp_skip_buf*: Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.
- *lp_check_control_path_of_fine_grain_psw*: Default value is `yes`, and the *LPPSW01* rule reports violations for fine-grain power switches. Set this parameter to `no` to not report such violations.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/power domains in the design.
- *power_switch* (Mandatory): Use to specify the power switches in power domains.
- *always_on_buffer* (Optional): Use to specify always-on buffers.

CPF Commands

- *create_power_nets* (Mandatory)
- *create_ground_nets* (Mandatory)
- *create_power_domain* (Mandatory)
- *update_power_domain* (Mandatory)
- *create_power_switch_rule* (Mandatory)
- *update_power_switch_rule* (Mandatory)
- *define_power_switch_cell* (Mandatory)
- *define_always_on_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *create_power_switch* (Mandatory)
- *map_power_switch* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the signal reaching the input enable pin <*pin-name*> of the instance <inst-name> of the Power Switch cell

<cell-name> is coming from the port of the same/relatively-off domain *<dom-name>*:

[LPPSW01_1][WARNING] Signal reaching the input enable pin '<pin-name>' of the Power Switch '<inst-name>'('<cell-name>') belonging to domain <dom-name1>', is coming from same/ relatively-off domain '<dom-name2>'

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears, when the signal reaching the input enable pin *<pin-name>* of the instance *<inst-name1>* of the Power Switch cell *<cell-name1>* is coming from the instance *<inst-name2>* of the cell *<cell-name2>* of the same/relatively off domain *<dom-name2>:*

[LPPSW01_2][WARNING] Signal reaching the input enable pin '<pin-name>' of the Power Switch '<inst-name1>'('<cell-name1>') belonging to domain '<dom-name1>', is coming from an incorrect instance '<inst-name2>'('<cell-name2>') belonging to same/ relatively-off domain '<domname2>'

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The consequences of not fixing the violations are as follows:

Message 1: The signal reaching the input enable pin of the Power Switch cell is coming from the port of same/relatively-off domain. This is a design flaw.

Message 2: The signal reaching the input enable pin of the Power Switch cell is coming from the instance of the cell of a same/relatively-off domain. This is a design flaw.

How to Debug and Fix

The violation is reported at the Power Switch cell instance, where the signal reaching the input enable pin of the cell is coming from a relatively off domain.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. The schematic shows that the signal reaching the input enable pin or SLEEP pin of the power switch is coming

from a relatively OFF domain.

To fix this violation message, ensure that the signal reaching the input enable pin of a Power Switch cell comes from a relatively always on domain

Example Code and/or Schematic

Consider the following example. A violation appears when the signal reaching the input enable pin `SLEEP` of the instance `top.ua.U1` of the Power Switch cell `HEADBUF16_X1M_A9TR` is coming from the port of the relatively-off domain `Vtop`:

```
[WARNING] Signal reaching the input enable pin 'SLEEP' of the
Power Switch 'top.ua.U1'('HEADBUF16_X1M_A9TR') belonging to
domain 'VA', is coming from same/ relatively-off domain 'Vtop'
```

The schematic is as follows:

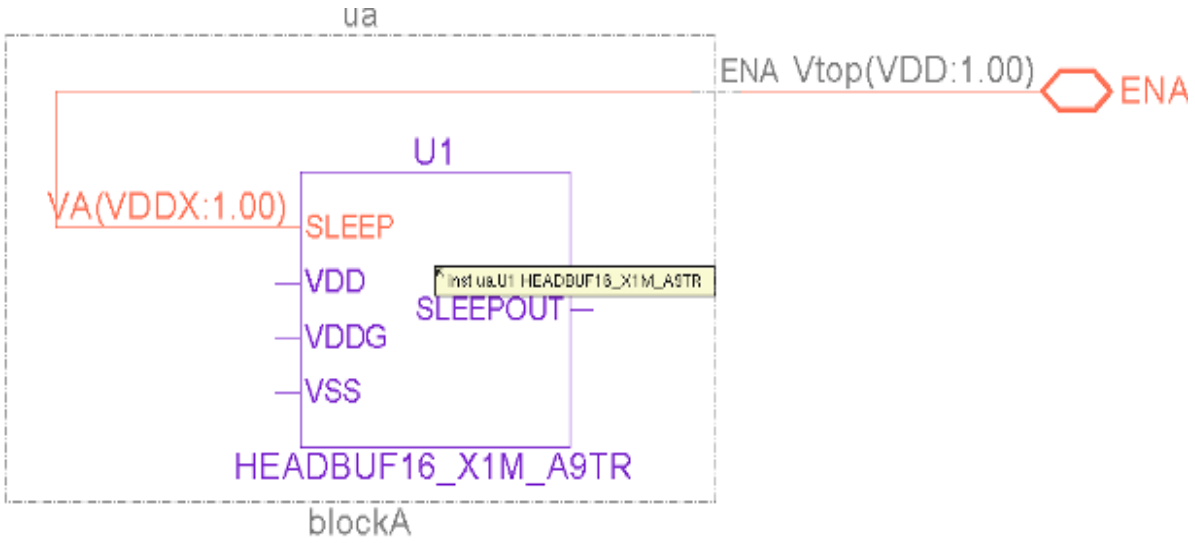


FIGURE 176. Incremental schematic

Default Severity Label

Warning

Rule Group

Special Purpose

Reports and Related Files

None

LPPSW02

Checks the connectivity of the signal reaching the input enable of the power switch

When to Use

Used this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

This rule can be also be run on the pre-layout netlist.

Description

The *LPPSW02* rule reports a violation when there is a break in the daisy chain of power switches in the pre-layout design. This rule also reports a violation when a correct enable signal, as specified in the provided power intent file, is not reaching the respective power switch.

Prerequisites

By default, the *LPPSW02* rule is not run while running the SpyGlass Power Verify solution. To enable this rule, specify the following command in the project file: set_goal_option addrules LPPSW02

Rule Exceptions

The *LPPSW02* rule does not work with the SGDC format.

Language

Verilog, VHDL

Parameter(s)

- *lp_skip_buf*: Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.
- *lp_check_all_ps_in_daisy_chain*: Default value is `0`, and most-on power switch is picked from the chain and all the buffers are validated against that power switch only. Set the value to 1 to make sure the buffers in

the path are validated to be more-on with respect to each power switch in the chain.

- *lp_check_control_path_of_fine_grain_psw*: Default value is `yes`, and the *LPPSW02* rule reports violations for fine-grain power switches. Set this parameter to `no` to not to report such violations.

Constraint(s)

CPF Commands

- *create_power_nets* (Mandatory)
- *create_ground_nets* (Mandatory)
- *create_power_domain* (Mandatory)
- *update_power_domain* (Mandatory)
- *create_power_switch_rule* (Mandatory)
- *update_power_switch_rule* (Mandatory)
- *define_power_switch_cell* (Mandatory)
- *define_always_on_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *create_power_switch* (Mandatory)
- *map_power_switch* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears, when the control pin *<pin-name>* of the instance *<inst-name>* of power switch *<cell-name>* is unconnected:

[LPPSW02_1][WARNING] Control pin '<pin-name>' of power switch '<instname>' ('<cell-name>') is unconnected

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears, when the control pin *<pin-name>* of the instance *<inst-name>* of power switch *<cell-name>* is undriven:

[LPPSW02_2][WARNING] Control pin '<pin-name>' of power switch '<instname>' ('<cell-name>') is undriven

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears, when correct enable signal is not of power switch *<cell-name>*:

[LPPSW02_3][WARNING] Enable signal is not reaching the control pin '<pinname>' of the power switch '<inst-name>' ('<cellname>')

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears, when constant value of *<val>* is reaching the control pin *<pin-name>* of the instance *<inst-name>* of power switch *<cell-name>*:

[LPPSW02_4][WARNING] Constant values '<val>' is reaching the control pin '<pin-name>' of power switch '<inst-name>' ('<cellname>')

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The consequences of not fixing the violations are as follows:

Message 1: The control pin of power switch is unconnected. This is a design flaw.

Message 2: The control pin of power switch is undriven. It is necessary for the pin to be driven for proper functioning. This is a design flaw.

Message 3: The enable signal needed is not reaching the control pin of

power switch. This is a design flaw.

Message 4: A constant value is reaching the control pin of power switch. The design will not work as desired. This is a design flaw.

How to Debug and Fix

The violation messages are reported at the power switch instance, which is breaking the daisy chain.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. This rule supports IS Abstraction between start and end points and abstracts all logic between these points, except level shifter cells. Refer to the *Abstraction between Start and End Points* topic in the *Atrenta Console Reference Guide*. Refer to the Example Code and/or Schematic section for an example.

To fix these violation messages, ensure proper connectivity of the signal reaching the input enable of the power switches in the design.

Example Code and/or Schematic

Example 1

Consider the following example. A violation appears when correct enable signal is not reaching the control pin `SLEEP` of the instance `top.ua.U1` of power switch `HEADBUF16_X1M_A9TR`:

```
[WARNING]Enable signal is not reaching the control pin 'SLEEP'
of the power switch 'top.ua.U1' ('HEADBUF16_X1M_A9TR')
```

The schematic generated is as follows:

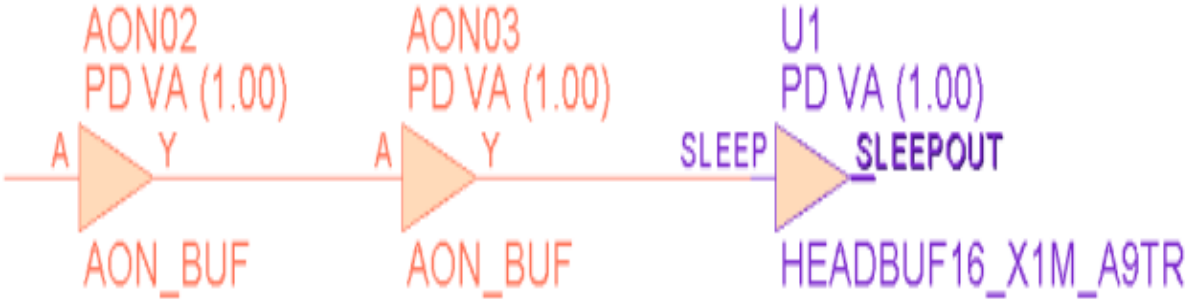


FIGURE 177. Incremental schematic

The schematic highlights the control pin of the power switch where the correct enable signal is not reaching.

Example 2

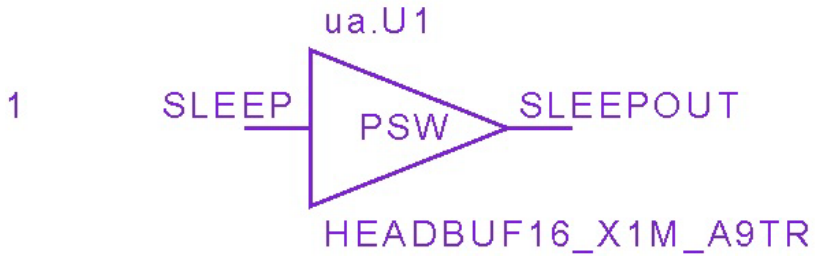
Consider the following code:

```
module top(in1,in2,in3,VDD,ENA,out1,out2,out3);
    input in1,in2,in3;
    inout VDD;
    inout [0:2] ENA;
    output out1,out2,out3;
    blockA ua(in1,in2,in3,VDD,ENA[1],ENA[2],out1,out2,out3);
endmodule

module blockA(in1,in2,in3,VDD,ENA,ENA_1,out1,out2,out3);
    input in1,in2,in3;
    inout VDD,ENA,ENA_1;
    output out1,out2,out3;
    supply1 VDDX;
    wire w1;
    wire w2;
    wire w1_1;
    TIEHI_X1M_A9TH(.Y(w2));
    HEADBUF16_X1M_A9TR U1 (.SLEEP(w2), .SLEEPOUT(w1));
    HEADBUF16 U2 (.SLEEP(ENA_1));
endmodule
```

The following violation message is reported because the`LPSVM41` constant value `1` is reaching the control pin `SLEEP` of the instance `HEADBUF16_X1M_A9TR` of power switch `top.ua.U1`:

```
Constant value '1' is reaching the control pin 'SLEEP' of power switch 'top.ua.U1' ('HEADBUF16_X1M_A9TR')
```

The schematic generated is as follows:



Default Severity Label

Warning

Rule Group

Special Purpose

Reports and Related Files

None

LPPSW03

Checks the presence of a delay buffer in the path of the input enable pin of a power switch

When to Use

The *LPPSW03* rule is supported in the SGDC, UPF and CPF formats. You can run this rule on the pre-layout netlist.

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The LPPSW03 rule reports a violation message when a delay buffer is not found in the path leading to the input enable pin of a power switch. If there is a chain of power switches, this rule checks for buffers between the input enable pin and the output pin of the power switches in the chain. All level shifters and combinational elements are skipped during the check.

Prerequisites

By default, the LPPSW03 rule is not run while running the SpyGlass Power Verify solution. To enable this rule, specify the following command in the project file:

```
set_goal_option addrules LPPSW03
```

Parameter(s)

- *lp_max_viol_count*: Default is 1000. This indicates that the LPPSW03 rule reports 1000 messages. Set the value to any positive integer number to report that number of messages.
- *lp_flag_missing_timing_arc_on_pin*: Default value is no. Set this parameter to yes to enable the LPPSW03 rule report missing timing arc for pins found in enable signal path of power switches.

Constraint(s)

SGDC

- *delay_buffer* (Mandatory)

CPF Commands

- *create_power_nets* (Mandatory)
- *create_ground_nets* (Mandatory)
- *create_power_domain* (Mandatory)
- *update_power_domain* (Mandatory)
- *create_power_switch_rule* (Mandatory)
- *update_power_switch_rule* (Mandatory)
- *define_power_switch_cell* (Mandatory)
- *define_always_on_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *create_power_switch* (Mandatory)
- *map_power_switch* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when a delay buffer is not present at the input enable <pin-name> of power switch <cell-name>:

[LPPSW03_1][ERROR] Delay buffer is not present at the input '<pin-name>' of power switch cell '<cell-name>'

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the cell <bufcell-name> specified in delay_buffer is not a buffer:

[LPPSW03_2][ERROR] Cell '<bufcell-name>' specified in delay_buffer is not a buffer

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when a terminal with multiple fan-outs, black box, or gray box <inst-name> based on timing arc is encountered in the path of the enable signal <cell-name> because checking is stopped at this point:

[LPPSW03_3][WARNING] Instance '<inst-name>' found in the path of enable signal '<cell-name>'. Rule checking terminated here for this path

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears when delay_buffer is not specified:

[LPPSW03_4][WARNING] Rule checking is not done as mandatory constraint '<constraint-name>' is not specified

For debugging information, click *How to Debug and Fix*.

Message 5

An informational message appears when the violation count of this rule exceeds the limit set by the *lp_max_viol_count* parameter. Refer to *Message 5* for the message and, for debugging information, refer to *How to Debug and Fix*.

Message 6

The following message appears if during the forward traversal of power switch control signal, a pin is encountered whose timing arc is not given in the library:

LPPSW03_6[Warning] Missing timing arc information on pin <pin-name> of instance <instance-name> found in the path of enable signal <sig-name>. Rule checking terminated here for this path

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

If these violation messages are not fixed, it means that there is no delay buffer present in the path of input enable pin of power switches. Ideally, in a daisy chain of power switches, there should be at least one delay buffer present in the path of the input enable pin of the power switch to introduce a delay when the power switches go OFF/ON in that daisy chain. Otherwise, the power switches might not work properly.

How to Debug and Fix

These violation messages are reported at the following locations:

- **Message 1**: Reported at the instance of the power switch that does not have a `delay_buffer` in the input enable pin path.
- **Message 2**: Reported at the instance specified in the `delay_buffer`.
- **Message 3**: Reported at the instance where rule checking is terminated.
- **Message 4**: Points to the SGDC file.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. This rule supports IS Abstraction between start and end points and abstracts all logic between these points, except level shifter cells. Refer to the *Abstraction between Start and End Points* topic in the *Atrenta Console Reference Guide*. Refer to the Example Code and/or Schematic section for an example.

To fix these violation messages, ensure that there is at least one delay buffer present in the path of the input enable pin of the power switch.

Example Code and/or Schematic

Consider the following example.

A violation appears a delay buffer is not present at the input enable `<pin-name>` of power switch `<cell-name>`:

```
[ERROR] Delay buffer is not present at the input 'SLEEP' of
power switch cell 'top.ua.u3'
```

The schematic is displayed as follows:

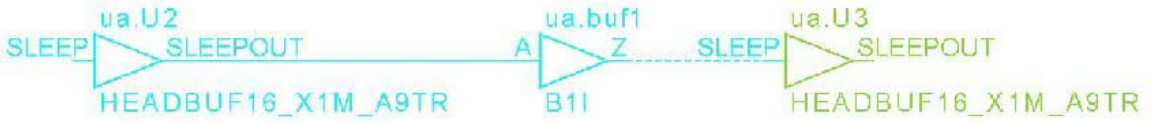


FIGURE 178. Incremental schematic

Default Severity Label

Error/Warning

Rule Group

Special Purpose

Reports and Related Files

None

LPPSW04

Ensures that the enable signal of a power switch has only AON buffers or delay buffers in its path

When to Use

The *LPPSW04* rule is supported in the SGDC, UPF and CPF formats. You can run this rule on the pre-layout netlist.

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPPSW04* rule reports a violation message if an object other than an AON or delay buffer is found in the path of the enable signal of a power switch. In addition, this rule detects power switches, which have a fan-out count exceeding the permitted value, in the path of the enable signal. This rule also reports a violation if the power switch control signal gets inverted before reaching a power switch instance.

Prerequisites

By default, the *LPPSW04* rule is not run while running the SpyGlass Power Verify solution. To enable this rule, specify the following command in the project file:

```
set_goal_option addrules LPPSW04
```

Rule Exceptions

All level shifters and combinational elements are skipped during the check.

Parameter(s)

- *lp_max_viol_count*: Default is 1000. This indicates that the `LPPSW04` rule reports 1000 messages. Set the value to any positive integer number to report that number of messages.
- *lp_max_psw_fanout_count*: Default is 1. This indicates that the *LPPSW04* rule reports any power switch, which has a fan-out count more than 1,

in the path of the enable signal. Set the value to any positive integer permit more fan-outs.

- *lp_enable_buf_check*: Default is `0`. Set the value to 1 to check for signals that are not buffered by a relatively-AON power.
- *lp_flag_missing_timing_arc_on_pin*: Default value is `no`. Set this parameter to `yes` to enable the `LPPSW04` rule report missing timing arc for pins found in enable signal path of power switches.

Constraint(s)

SGDC

- *always_on_buffer* (Optional)
- *delay_buffer* (Optional)

CPF Commands

- *create_power_nets* (Mandatory)
- *create_ground_nets* (Mandatory)
- *create_power_domain* (Mandatory)
- *update_power_domain* (Mandatory)
- *create_power_switch_rule* (Mandatory)
- *update_power_switch_rule* (Mandatory)
- *define_power_switch_cell* (Mandatory)
- *define_always_on_cell* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *create_power_switch* (Mandatory)
- *map_power_switch* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the enable signal <sig-name> of a power switch is connected to an instance <inst-name> that is neither an AON buffer nor a delay buffer:

[LPPSW04_1][ERROR] Signal '<sig-name>' is connected to instance '<inst-name>' which is not an always-on buffer or a delay-buffer

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the cell <bufcell-name> specified in the delay_buffer is not a buffer:

[LPPSW04_2][ERROR] Cell '<bufcell-name>' specified in delay_buffer is not a buffer

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when a terminal with multiple fan-outs, black box, or grey box <inst-name> based on timing arc is encountered in the path of the enable signal <cell-name> because checking is stopped at this point:

[LPPSW04_3][WARNING] Instance '<inst-name>' found in the path of enable signal '<cell-name>'. Rule checking terminated here for this path

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears when a power switch <pwr-switch>, which is in the path of the enable signal <sig-name>, has fan-outs that exceed the limit:

[LPPSW04_4][ERROR] Power switch '<pwr-switch>' found in the path of enable signal '<sig-name>' has more fanouts than maximum permitted fanouts : '<fan-out-limit>'

For debugging information, click *How to Debug and Fix*.

Message 5

An informational message appears when the violation count of this rule exceeds the limit set by the *lp_max_viol_count* parameter. Refer to *Message 5* for the message and, for debugging information, refer to *How to Debug and Fix*.

Message 6

The following message appears when there are odd number of inverters in the path of control signal before reaching any power switch:

[LPPSW04_6][ERROR] Control signal <signal-name> is inverted before reaching power switch <power-switch-name>

For debugging information, click *How to Debug and Fix*.

Message 7

The following message appears if during the forward traversal of power switch control signal, a pin is encountered whose timing arc is not given in the library:

LPPSW04_7[Warning] Missing timing arc information on pin <pin-name> of instance <instance-name> found in the path of enable signal <sig-name>. Rule checking terminated here for this path

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

Messages 1, 2, and 3: If these violation messages are not fixed, it means that something other than delay buffer or AON buffer is present in the path of the power switch enable signal and the power switch input enable pin. This may modify the power switch enable signal and the actual signal might not reach the input enable pin of the power switch. Therefore, the power switch may not work properly.

Message 4: If not fixed, this is a design error.

Message 6: The expected control signal will not reach to the power switch.

How to Debug and Fix

These violation messages are reported at the following locations:

- **Message 1**: Reported at the instance found in the path of the power switch enable signal.
- **Message 2**: Reported at the instance specified in the `delay_buffer`.
- **Message 3**: Reported at the instance where rule checking is terminated.
- **Message 4**: Reported at the power switch stated in the violation message.
- **Message 6**: Reported at the power switch stated in the violation message.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. This rule supports IS Abstraction between start and end points and abstracts all logic between these points, except level shifter cells. Refer to the *Abstraction between Start and End Points* topic in the *Atrenta Console Reference Guide*. Refer to the Example Code and/or Schematic section for an example.

To fix these violation messages, ensure that only AON or delay buffers are present in the path of the enable signal of the power switch.

To fix Message 4, increase the permitted fan-out count by changing the *lp_max_psw_fanout_count* parameter value. Alternatively, change the design to decrease the fan-out count of the power switch.

Example Code and/or Schematic

Example 1

In this example, Message 1 appears when the enable signal `<sig-name>` of a power switch is connected to an instance `<inst-name>` that is neither an AON buffer nor a delay buffer:

```
[ERROR] Signal 'top.ena' is connected to instance 'top.and1'
which is not an always-on buffer or a delay-buffer
```

The schematic is displayed as follows:

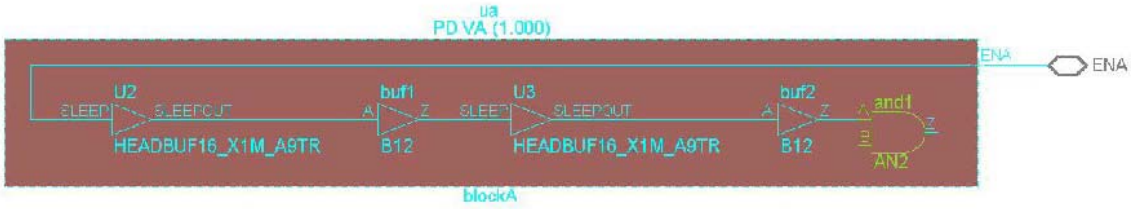


FIGURE 179. Incremental schematic

Example 2

Suppose the *lp_max_psw_fanout_count* parameter is set to 2. In this example, Message 4 appears because the power switch `top.ua.U2` is in the enable signal `top.ENA[1]` path and the switch has a fan-out count greater than 2.

```
Power switch 'top.ua.U2' found in the path of enable signal
'top.ENA[1]' has more fanouts than maximum permitted fanouts :
'2'
```

The schematic is displayed as follows:

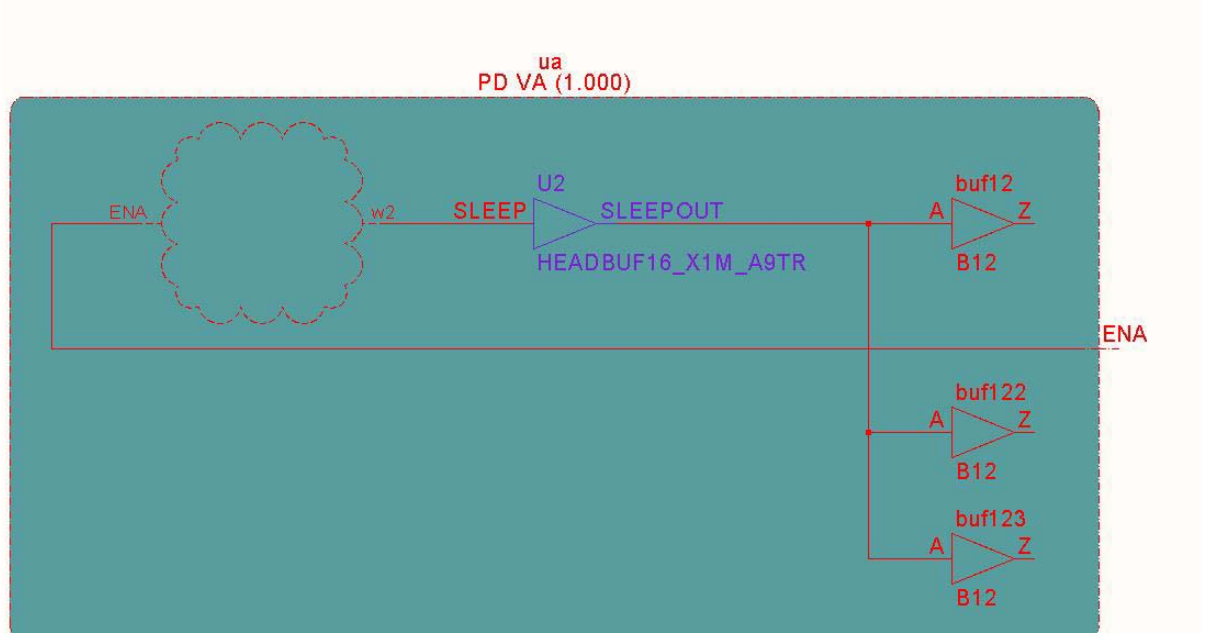


FIGURE 180. Incremental schematic

As shown in the schematic, the power switch `top.ua.U2` has three fan-outs. To resolve the violation error, increase the permitted fan-out count by changing the *lp_max_psw_fanout_count* parameter value. Alternatively, change the design to decrease the fan-out count of the power switch.

Example 3

Consider this Verilog snippet:

```
INVX1 inv1(nen1,a);
INVX1 inv3(a,ein1);
ps_custom
ps1(.a(ein1),.b(ein1),.aout(aoutps1),.bout(boutps1));

ps_custom
ps2(.a(aoutps1),.b(boutps1),.aout(aoutps2),.bout(boutps2));
INVX1 inv4(aoutps2,wire1);
INVX1 inv5(wire1,wire2);
```

```
INVX1 inv6(wire2,wire3);

ps_custom
ps3(.a(wire2),.b(wire3),.aout(aoutps3),.bout(boutps3));
```

The corresponding UPF command:

```
create_power_switch PS_rule1 -domain Vtop -input_supply_port
{VSSIN vss} -control_port  {EN nen1} -output_supply_port
{VSSOUT vss1}
```

In the above example, the following violation message is reported because there are five inverters between control signal `nen1` and control pin `a` of the power switch `top.ps3`:

```
Control signal 'top.nen1' is inverted before reaching power
switch 'top.ps3'
```

The schematic is displayed as follows:

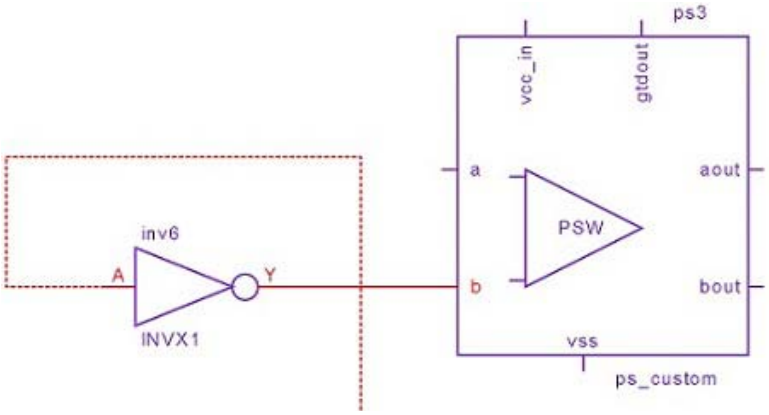


FIGURE 181. Incremental schematic

Default Severity Label

Error/Warning

Rule Group

Special Purpose

Reports and Related Files

None

LPSVM29

Generates design-level and hierarchical reports

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM29* rule generates design-level and hierarchical reports that list the cell composition of different VT type libraries. This rule generates two types of reports (design level & hierarchical) stating the cells composition of different vt type libraries.

This rule can be run in the CPF/UPF flow also. You need to specify the *multivt_lib* constraint in the SGDC file, but the voltage domain/supply information is picked up from the corresponding CPF/UPF file.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *multivt_lib* (Mandatory): Use to specify VT type libraries.

Messages and Suggested Fix

Message 1

The following message appears for directory <*dir-name*>, where each report is generated:

[INFO] Report 'lp_multivtreport' containing information about

multivt library composition in the design is generated in <dir-name> directory

For debugging information, click *How to debug and fix*.

Message 2

The following message appears for the second report:

[INFO] Report 'lp_multivt_perblock' containing hierarchical information about multivt library composition is generated in <dir-name> directory

Potential Issues

Not applicable

Consequences of Not Fixing

Not applicable

How to debug and fix

Double-click the violation messages to view the *lp_multivtreport* and *lp_multivt_perblock* reports in the Atrenta Console GUI.

Example Code and/or Schematic

Since in above SGDC file, the *multivt_lib* constraint is specified for cells CORE9GPHS1, CORE9GPHS2, CORE9GPLL1, and CORE9GPLL2. Therefore, the following reports are generated for these cells:

- lp_multivtreport.rpt (see report snapshot at *lp_multivtreport*)
- lp_multivt_perblock.rpt (see report snapshot at *lp_multivt_perblock*)

```
multivt_lib -type highvt -names CORE9GPHS1 CORE9GPHS2
multivt_lib -type lowvt -names  CORE9GPLL1 CORE9GPLL2
```

Default Severity Label

Info

Rule Group

Special Purpose

Reports and Related Files

- *lp_multivtreport*: Contains instantiation information about VT type library cells in the complete design.
- *lp_multivt_perblock:* Contains instantiation information about VT type library cells in each hierarchy.

LPSVM37

Checks for special cells in specified regions

When to Use

Use this rule to check the existence of cells specified by using the 'specialcell' constraint in the design region specified by using the argument '-regions' of the constraint. Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The LPSVM37 rule reports user-defined cells that are instantiated outside the specified hierarchy.

The *LPSVM37* rule can also be run in the CPF/UPF flow. In this flow, you need to specify the *special_cell* constraint in the SGDC file and then the voltage domain information is retrieved from the corresponding CPF/UPF file.

Prerequisites

Specify the *special_cell* constraint in the SGDC file.

Language

Verilog, VHDL

Parameter(s)

- *lp_check_hier*: Default value is 1. Set the *lp_check_hier* rule parameter to 0 to limit the checking to just the specified region without traversing the hierarchy.

Constraint(s)

SGDC

- *special_cell* (Mandatory): Use to specify the names of the special cells as checked by the LPSVM37 rule.

Messages and suggested Fix

Message 1

The following message appears when the instance *<inst-name>* of cell *<cell-name>* specified using the *special_cell* constraint is outside the permitted hierarchical region *<region-list>*:

[LPSVM37_1][RECOMMENDED] Special cell '<cell-name>' instantiated as '<instname>' does not match with the specified regions '<region-list>

For debugging information, click *How to debug and fix*.

Message 2

The following message appears when no instance of cell *<cell-name>* specified using the *special_cell* constraint is in the permitted hierarchical region *<region-list>*:

[LPSVM37_2][RECOMMENDED] No special cell '<cell-name>' is found in the specified regions '<region-list>'

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not fixing

No instance of the cell mentioned using the *special_cell* constraint is present in the specified hierarchy mentioned using the '-regions' argument of the constraint. This makes the constraint redundant.

How to debug and fix

These violations are reported at the instantiation of a special cell that is in a region other than the permissible hierarchical regions.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix these violations, ensure that:

- Special cells, specified using the *special_cell* constraint, are instantiated in the specified region
- Special cells are located in any of the specified regions only.

Example Code and/or Schematic

Consider the following example. A violation occurs when instance `'mid.t1.inst4.m2'` of cell `'myand'`, which is specified by using the *special_cell* constraint, is outside the permitted hierarchical region `'mid.t1.inst3'`:

[RECOMMENDED] `Special cell 'myand' instantiated as 'mid.t1.inst4.m2' does not match with the specified regions 'mid.t1.inst3'`

The schematic as follows:

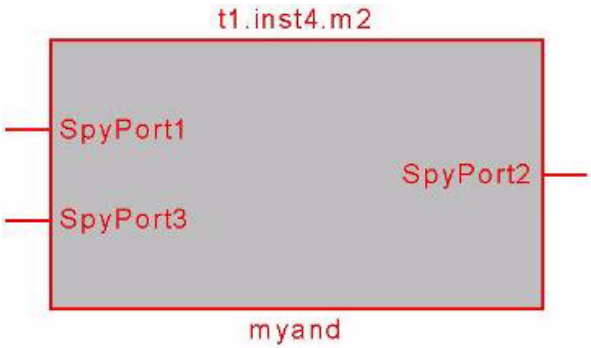


FIGURE 182. Incremental schematic

The schematic shows the instance which is of the cell type provided using the *special_cell* constraint but is not actually present in the design in the region provided by the argument ‘-regions’ of the constraint.

Default Severity Label

Recommended

Rule Group

Special Purpose

Reports and Related Files

None

LPSVM41

Reports registers that are not in their power down chain

When to Use

Use this rule to:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM41* rule reports incorrect register connections of user-specified power down signals.

The *LPSVM41* rule can also be run in the CPF/UPF flow. In this flow, you need to specify the *power_down_sequence* constraint in the SGDC file and then the voltage domain information is retrieved from the corresponding CPF/UPF file.

Prerequisites

Specify the *power_down_sequence* constraint in the SGDC file.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *power_down_sequence* (Mandatory): Use to specify the registers that should be connected to the specified power down signals. Optionally, specify the following:
  - Register instance name list using the `-instnames` argument.
  - Name list of cells to be skipped while traversing the fan-out of the specified power down signal using the `-ignorecells` argument.

Messages and Suggested Fix

Message 1

The following message appears at the location of instance *<instname>* specified using the `instnames` argument when the power down signal *<sig-name>* is not connected to the instance:

```
[LPSVM41_1][WARNING] Special register instance '<inst-name>'
not connected to power down signal '<sig-name>'
```

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears at the location of instance *<instname>* to which the power down signal *<sig-name>* is connected but the instance is not specified with the `instnames` argument:

```
[LPSVM41_2][WARNING] Instance '<inst-name>' connected to power
down signal '<sig-name>' but not specified in its instance list
```

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

To control the rush current the domain is switched on in a sequence such that a block of instances are powered up first followed by the another block and so on. In addition, there may be initialization dependency as well. These blocks will have different power down signals that have different delay from the same derivative signals. Each group of the instances should be connected to its own power down signal. A mismatch can cause failure is design functioning.

How to Debug and Fix

The violations are reported at the place where the power down signal that is not connected to the specified register instance is set or used.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix these violations, ensure the following:

- Power down signals are properly connected to the specified register instances using the *power_down_sequence* constraint.

- Power down signals are properly connected to the specified register instances specified using the `instnames` argument.

Example Code and/or Schematic

Consider the following example. A violation occurs when power down signal `'vdd'` is connected to an instance `'top.fd1_inst'` that is not specified with the `instnames` argument:

```
[WARNING] Instance 'top.fd1_inst' connected to power down
signal 'vdd' but not specified in its instance list
```

The schematic is as follows:

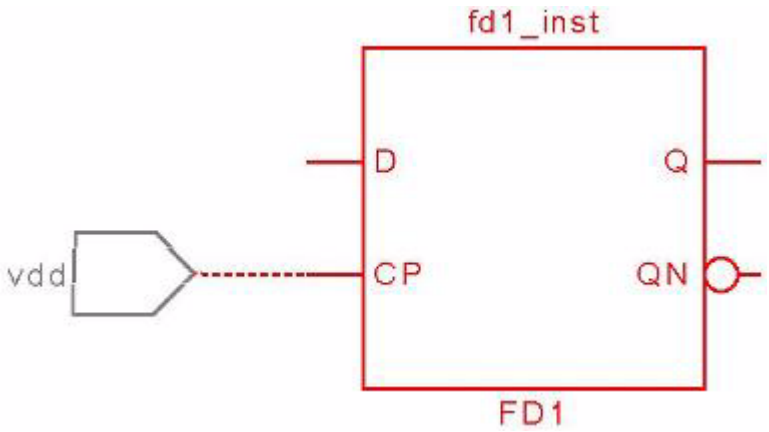


FIGURE 183. Incremental schematic

Schematic Highlight

The *LPSVM41* rule reports the following:

- Only the power down signal name is specified using the `signalname` argument
- The *LPSVM41* rule reports all register instances to which the power down signal is directly connected.

- The power down signal name is specified using the `-signalname` argument and the instance name list is specified using the `instnames` argument.

  The *LPSVM41* rule reports the following:

  - All those register instances other than the specified instances to which the power down signal is directly connected.
  - All user-specified register instances to which the power down signal is not directly connected.
- The power down signal name is specified using the `-signalname` argument, the register instance name list is specified using the `instnames` argument, and cell name list of cells to be skipped using the `-ignorecells` argument
- The *LPSVM41* rule skips the instances of cells specified with using the `-ignorecells` argument while traversing the fan-out of the specified power down signal and reports the following:
  - All those register instances other than the specified instances to which the power down signal is directly connected.
  - All user-specified register instances to which the power down signal is not directly connected.
- The signal name is specified using the `-signalname` argument and cell name list of cells to be ignored using the `-ignorecells` argument.

The *LPSVM41* rule reports all register instances to which the power down signal is directly connected skipping instances of cells.

Default Severity Label

Warning

Rule Group

`Special Purpose`

Reports and Related Files

None

LPSVM42

Checks to ensure low power signals coming up correctly out of power on reset

When to Use

Use this rule to:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM42* rule reports low power signals that are being enabled while the design comes out of Power on Reset (POR).

The LPSVM42 rule can also be run in the CPF/UPF flow. In this flow, you need to specify the *assertion_signal* constraint in the SGDC file. This retrieves the voltage domain information from the corresponding CPF/UPF file.

Prerequisites

It is mandatory to specify the clock period (in nanoseconds) by using the `-period` argument of the *clock* constraint, as shown in the following example:

```
clock -name clk -period 20
```

For these clocks, the toggle activity of the highest period clock is assumed to be 2 and the toggle activity value of all other clocks is a suitable divisor of 2.

Language

Verilog, VHDL

Parameter(s)

- *lp_genvcdfile*: Default value is `none`. Specify a single file name or a space-separated list of file names of the input VCD testbench files. Enclose the file names in double quotes.
- *lp_vcdtopname*: Default value is `none`. Specify the instantiated name of the top module in the VCD testbench file.
- *lp_islc*: Default value is `0`. Set value to `1` to convert the signal names including the module names to lowercase during translation.
- *lp_vcdstarttime* and *lp_vcdendtime* (Optional): Default value of both *lp_vcdstarttime* and *lp_vcdendtime* is `-1`. Specify the start time and end time for VCD analysis. When set on default, the start time and end time specified in the VCD file are used.
- *lp_vcdminclk* (Optional): Default value is `-1`. Specify the minimum clock period. When set on default, the clock period of the VCD clock with minimum clock period is used for reference.
- *lp_write_sgdc* (Optional): Default value is `1`. SpyGlass generates an SGDC file, called auto_activity.sgdc, which contains activity constraints based on the specified VCD file. Therefore, you can use the generated SGDC file in place of the VCD file in the subsequent runs. Set the value to 0 to not generate the SGDC file.

Constraint(s)

SGDC

- *assertion_signal* (Mandatory): Use to specify the Power On Reset (POR) signal with its values and low power signals with their active high/active low values using the *assertion_signal* constraint.
- *clock* (Optional): Use to specify the names of the clocks in the design by using the *clock* constraint.

Messages and Suggested Fix

The following message appears when the low power signal *<sig-name>* is enabled with the value *<value>* at simulation time *<time>* while the POR signal *<por-sig-name>* is active:

[LPSVM42_1][WARNING] `At simulation time '<time>' when POR`

`Signal '<por-sig-name>' is active, lowpower Signal '<sig-name>' is enabled with value '<value>'`

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

The power on reset signal is like an interrupt signal that supersedes all low power control signals. When it is active, no other low power signal should be enabled.

How to Debug and Fix

The violation is reported at the place where the low power signal which is coming out of POR is set or used. It mentions the line number of the VCD file pointing the value change for the low power signal.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, ensure that the low power signals are not enabled while the design comes out of POR (Power On Reset).

Example Code and/or Schematic

Consider the following example. A violation occurs when low power signal `'mc_top.u3.rd_fifo_clr'` is enabled with value `'1'` at simulation time `'160'` while the POR signal `'mc_top.u3.rst'` is active:

`At simulation time '160' when POR Signal 'mc_top.u3.rst' is active, lowpower Signal 'mc_top.u3.rd_fifo_clr' is enabled with value '1'`

The schematic is as follows:

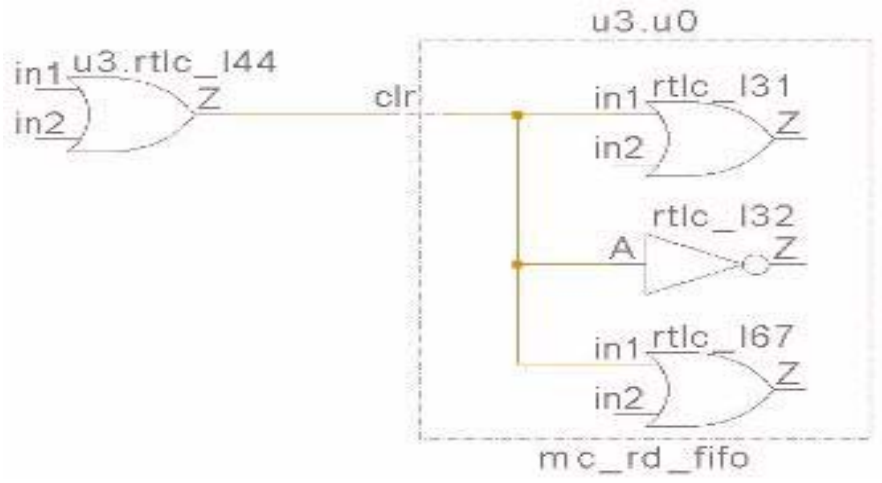


FIGURE 184. Incremental schematic

The schematic highlights the low power signals that are being enabled as active high or active low while the design is in the POR state determined by the value of the POR signal. The schematic shows the POR signals and violating low power signal.

Default Severity Label

Warning

Rule Group

Special Purpose

Reports and Related Files

None

LPSVM43

Reports incorrect power up and power down sequences of power domains

When to Use

Use this rule to:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM43* rule checks the following:

- All clocks associated with the power up/power down signals are de-asserted to `0` prior to power up/power down.
- The actual sequence of the specified signals match with the specified sequence during power up/power down.

The LPSVM43 rule can also be run in the CPF/UPF flow. In this flow, you need to specify the *domain_signal* constraint in the SGDC file. This retrieves the voltage domain information from the corresponding CPF/UPF file.

Specifying Activity Information

To perform Activity analysis, specify the activity information using the rule parameters mentioned in the `Parameters` section of the rule.

SpyGlass calculates and propagates the activity and probability data for the nets that are not specified in the VCD file. For all the other nets, the probability value is assumed to be 0.5, that is, 0 and 1 being equi-probable and the toggle activity value is assumed to be 0.1.

Prerequisites

It is mandatory to specify the clock period (in nanoseconds) by using the `-period` argument of the *clock* constraint, as shown in the following example:

```
clock -name clk -period 20
```

For these clocks, the toggle activity of the highest period clock is assumed to be 2 and the toggle activity value of all other clocks is a suitable divisor of 2.

Language

Verilog, VHDL

Parameter(s)

- *lp_genvcdfile*: Default value is `none`. Specify a single file name or a space-separated list of file names of the input VCD testbench files. Enclose the file names in double quotes.
- *lp_vcdtopname*: Default value is `none`. Specify the instantiated name of the top module in the VCD testbench file.
- *lp_islc*: Default value is `0`. Set value to `1` to convert the signal names including the module names to lowercase during translation.
- *lp_vcdstarttime* and *lp_vcdendtime* (Optional): Default value of both *lp_vcdstarttime* and *lp_vcdendtime* is `-1`. Specify the start time and end time for VCD analysis. When set on default, the start time and end time specified in the VCD file are used.
- *lp_vcdminclk* (Optional): Default value is `-1`. Specify the minimum clock period. When set on default, the clock period of the VCD clock with minimum clock period is used for reference.
- *lp_write_sgdc* (Optional): Default value is `1`. SpyGlass generates an SGDC file, called auto_activity.sgdc, which contains activity constraints based on the specified VCD file. Therefore, you can use the generated SGDC file in place of the VCD file in the subsequent runs. Set the value to 0 to not generate the SGDC file.

Constraint(s)

SGDC

- *domain_signal* (Mandatory) : Use to specify the following:
  - Power up/power down signals using the `-name` argument.
  - Signals and their expected sequences during power up and power down using the `-seqsignals` and `-seqvalue` arguments, respectively.

- Associated clock sources using the `-clocks` argument.

- *clock* (Optional): Use to specify the names of the clocks in the design using the *clock* constraint.

Messages and Suggested Fix

Message 1

The following message appears when clock *<clk-name>* is not de-asserted to value `0` prior to power up/power down while the power down signal *<pd-sig-name>* is enabled:

```
[LPSVM43_1][WARNING] At simulation time '<time>', Clock '<clk-name>' is not de-asserted when power down signal '<pd-sig-name>' is enabled
```

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when signals *<sig-name-list>* are found to have sequence *<pattern1>* against the expected sequence *<pattern2>* at simulation time *<time>* while the power down signal *<pd-sig-name>* is enabled:

```
[LPSVM43_2][WARNING] At simulation time '<time>', Mismatch found for signals '<sig-name-list>' when power down signal '<pd-sig-name>' is enabled, Expected Sequence: '<pattern1>' Found Sequence: '<pattern2>'
```

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The power sequence controls provide the various control signals of the design. It is mandatory that various control signals are asserted and de-asserted in accordance to the power states of the design. For example, when a power down signal is asserted (i.e. power domain would be off) the clock signal of that power domain can be de-asserted but clock should be asserted again before power down signal is de-asserted power would be on.

How to Debug and Fix

The violation is reported at the first place where the clock signal or *domain_signal* is set or used.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix the violation, ensure that:

- All the clocks going inside the power domain are not asserted prior to power down, and power up.
- The down sequence for a given power domain are correct.

Example Code and/or Schematic

Consider the following example. A violation occurs when signals `'mc_top.wb_cyc_i, mc_top.u3.rd_fifo_clr'` have sequence `'01'` against the expected sequence `'10'` at simulation time `'117'` while the power down signal `'mc_top.u3.rst'` is enabled: :

```
[WARNING] At simulation time '117', Mismatch found for signals
'mc_top.wb_cyc_i, mc_top.u3.rd_fifo_clr' when power down signal
'mc_top.u3.rst' is enabled, Expected Sequence: '10' Found
Sequence: '01'
```

The schematic is as follows:

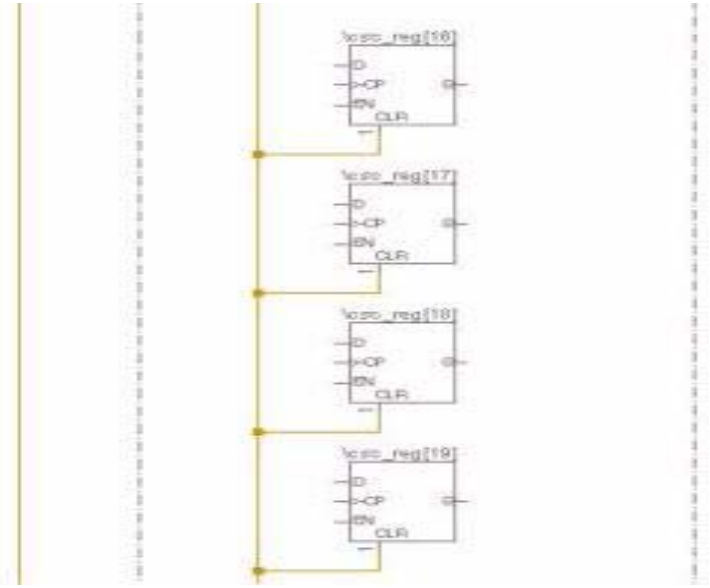


FIGURE 185. Incremental schematic

The schematic highlights the following:

- The power down signal
- Violating signals

Default Rule Severity

Warning

Rule Group

`Special Purpose`

Reports and Related Files

None

LPSVM44

Reports user-specified signals that are incorrectly connected

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM44* rule checks to ensure that a specified signal connects only to specified pins of a list of cells. For example, a specified isolation signal should be connected to a specified pin of an isolation cell.

The LPSVM44 rule can also be run in the CPF/UPF flow. In this flow, you need to specify the *cell_hookup* constraint in the SGDC file and then the voltage domain cell information is retrieved from the corresponding CPF/UPF file.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *cell_hookup* (Mandatory): Use to specify the signal to be checked using the *cell_hookup* constraint. Optionally, also specify the following:
  - Cell-pin pair name list using the `-names` argument
  - Name list of cells to be skipped while traversing the fan-out of the specified signal using the `-ignorecells` argument

Messages and Suggested Fix

The following message appears when the user-specified signal *<sig-name>* is incorrectly connected to the pin *<pin-name>* of the cell *<cellname>*:

[LPSVM44_1][WARNING] Signal '<sig-name>' connected to incorrect pin '<pin-name>' of cell '<cell-name>'

Potential Issues

The violation messages explicitly states the potential issues.

Consequences of Not Fixing

If this is not fixed, it means that the signal is connected to a pin of cell other than what is specified by *cell_hookup* constraint in SGDC.This is clearly against the user intent when he specified cell_hookup explicitly.

How to Debug and Fix

The violation is reported at the cell instance where the specified signal is connected to a pin other than the specified list.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix the violation, ensure that specified signal connects only to a list of specified pins of cells.

Example Code and/or Schematic

Consider the following example. A violation occurs when the signal 'CKT_TOP.CLK' is incorrectly connected to pin 'A' of cell 'SBNSV15':

[WARNING] Signal 'CKT_TOP.CLK' connected to incorrect pin 'A' of cell 'SBNSV15'

The schematic is as follows:

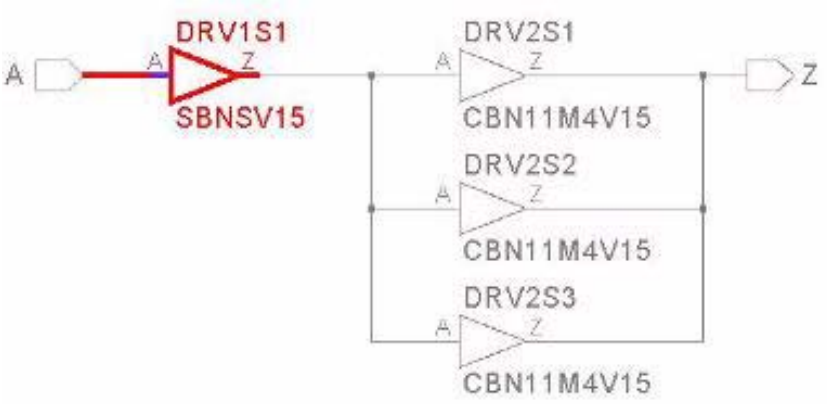


FIGURE 186. Incremental schematic

Schematic Highlight

The *LPSVM44* rule reports as follows:

- Only the signal name is specified using the `-signame` argument

  The *LPSVM44* rule reports all instances to which the signal is directly connected.

- The signal name is specified using the `-signame` argument and the cell-pin pair name list is specified using the `-names` argument.

  The *LPSVM44* rule reports the following:

  - ❒ All those instances, other than the instances of the specified cells to which the signal is directly connected.
  - ❒ All instances of the specified cells where the signal is directly connected but not to the specified pin.

- The signal name is specified using the `-signame` argument, the cell-pin pair name list is specified using the `-names` argument, and cell name list of cells to be ignored using the `-ignorecells` argument

  The *LPSVM44* rule skips the instances of cells specified with using the `-ignorecells` argument while traversing the fan-out of the specified signal and reports the following:

- ❒ All those instances other than the instances of specified cells to which the signal is directly connected.
- ❒ All instances of the specified cells where the signal is directly connected but not to the specified pin.

- The signal name is specified using the `-signame` argument and cell name list of cells to be ignored using the `-ignorecells` argument.

The *LPSVM44* rule reports all instances to which the signal is directly connected while skipping the instances of cells specified with using the `-ignorecells` argument.

Default Rule Severity

Warning

Rule Group

`Special Purpose`

Reports and Related Files

None

LPSVM45

Checks for the correct enable logic in a power switch for the gate level and PG netlist

When to use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LPSVM45* rule reports incorrect enable logic for power switches in gate-level netlists.

Consider the following constraint specifications specifying power domain `PD1` and power switch `pwrsw1`:

```
voltage_domain -name PD1 -value 1.2 0
  -instname "TOP.lower1" -isosig top.iso_sig
  -isoval 1 -enableports 'EN1 1 EN2 "-1" EN3[1] 100'
power_switch -name pwrsw1 -en_inv_in en
```

The `-enableports` argument of the *voltage_domain* constraint specifies the ports of power domain through which enable signals of power switches must be connected. The enable port of the power domain and the maximum number of enables of power switches allowed to be connected to it, are specified as a space-separated pair. For example, `EN1 1` indicates that only one power switch enable can be connected to port `EN1`.

The *LPSVM45* rule reports the following:

- Enable pin of a power switch that is not directly connected to one of the enable ports specified for its power domain.
- Enable pin of a power switch that does not originate directly from an always-on domain with no intervening combinational logic (buffers, gates, etc.) in any power domain that can be powered down.
- Enable port of a power domain that is not connected to any power switches.

- Enable port of a power domain that has more than specified number of power switch enables connected.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify power domains by using the *voltage_domain* constraint and their enable ports by using the `-enableports` argument.
- *power_switch* (Mandatory): Use to specify power switches.

CPF Commands

- *create_power_domain* (Mandatory)
- *update_power_domain* (Mandatory)
- *define_power_switch_cell* (Mandatory)
- *create_power_switch_rule* (Mandatory)
- *update_power_switch_rule* (Mandatory)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *create_power_switch* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the source driving the enable port *<en-port-name>* (connected enable net *<en-net-name>*) of the power domain *<pd-name>* does not exist in an always-on domain:

[LPSVM45_1][WARNING] Source driving the enable port '<en-port-name>' (enable net '<en-net-name>') defined for power domain '<pd-name>' does not exist in always-on domain

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the enable pin *<enpinname>* of powerswitch instance *<ps-inst-name>* is not connected to any of the enable ports *<en-port-list>* of the power domain *<pd-name>*:

[LPSVM45_2][WARNING] Enable pin '<en-pin-name>' of powerswitch instance '<ps-inst-name>' is not connected to any of the enable ports '<en-port-list>' specified with the power domain '<pd-name>'

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when the enable port *<enportname>* of power domain *<pd-name>* does not connect to any power switch enable:

[LPSVM45_3][WARNING] Enable port '<en-port-name>' defined for power domain '<pd-name>' does not connect to at least one power switch

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears when the number of powerswitch enables *<count>* connected to enable port *<enportname>* of power domain *<pd-name>* exceeds the limit *<limit>* specified for the enable port:

[LPSVM45_4][WARNING] Total number of power switches '<count>' connected to the enable port '<en-port-name>' defined for power domain '<pd-name>' exceeds the maximum count '<limit>' as given in the constraint file

For debugging information, click *How to Debug and Fix*.

Message 5

The following message appears when an instance *<ps-inst-name>* of dual-port power switch is found in the daisy chain with incorrect connections:

[LPSVM45_5][WARNING] Incorrect Dual Port PowerSwitch '<ps-inst-name>' found in Daisy chain

Potential Issues

The violation messages occur because of any of the following issues:

- Enable pin of a power switch that is not directly connected to one of the enable ports specified for its power domain.
- Enable pin of a power switch that does not originate directly from an always-on domain with no intervening combinational logic (buffers, gates, etc.) in any power domain that can be powered down.
- Enable port of a power domain that is not connected to any power switches.
- Enable port of a power domain that has more than specified number of power switch enables connected.

Consequences of Not Fixing

The consequences of not fixing the violations are as follows:

Message 1: The source driving enable port defined for a power domain must exist in always-on domain. Hence the warning.

Message 2: The enable pin of an instance of power switch is not connected to any of the enable ports specified in the power domain.

Message 3: The enable port defined for a power domain is not connected to any power switch enable. So the enable port is of no use. Hence the warning.

Message 4: Total number of power switches connected to a particular enable port defined in the power domain should not exceed the maximum number as specified by a space separated pair in “-enableports” constraint.

Message 5: A dual-port power switch is found in the daisy chain with incorrect connections. Hence the warning.

How to Debug and Fix

The violation is reported at the enable pin of a power switch which is not connected to the specified enable port of the power domain.

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix the violation, ensure that:

- The pin of power switches should be connected to the specified enable ports for its power domain.
- Ports of power domains should be connected to some power switches.
- Ports of power domains should note have more than the permissible number of power switch enables connected.

Example Code and/or Schematic

Consider the following example. A violation occurs when the source driving enable port `'PwrIN1'` (connected enable net `'top.vdd_buf'`) of power domain `'top.vdd_buf'`does not exist in an always-on domain:

```
[WARNING] Source driving the enable port 'PwrIN1' (enable net
'top.vdd_buf') defined for power domain 'top.vdd_buf' does not
exist in always-on domain
```

The schematic is as follows:

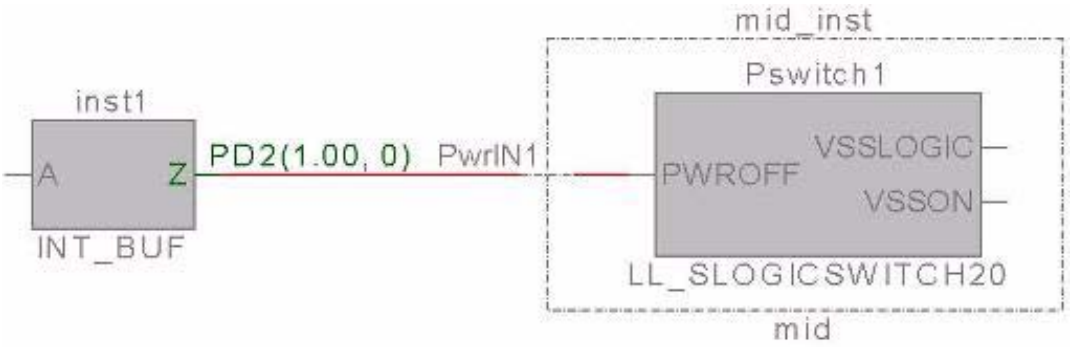


FIGURE 187. Incremental schematic

Schematic Highlight

The schematic highlights the enable pin of a power switch that does not originate directly from an always-on domain with no intervening combinational logic (buffers, gates, etc.) in a power domain that can be powered down.

Default Severity Label

Warning

Rule Group

`Special Purpose`

Reports and Related Files

- *lp_vd_info*: Contains the total number of power switches (by type) that are connected to each enable port of each power domain.

LPSVM46

Reports incorrect connections in a RAM switch

When to Use

In the SpyGlass Power Verify solution, a RAM switch is used for gating the memories. Use this rule to check the connections of a RAM switch with enable signals and memories.

This rule is recommended for use with *Gate-level netlist files and their associated library files*, *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*, and *DEF files and their associated LEF files*.

Description

The *LPSVM46* rule reports incorrect connections to the RAM switches in RTL and gate-level netlist. This rule performs the following checks:

- All signals reaching the enable pin of a RAM switch must originate from the always-on logic.
- The total number of RAM switches must match the total number of RAM instances in the power down mode.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain* (Mandatory): Use this constraint to specify the power domains.
- *ram_switch* (Mandatory): Use this constraint to specify the RAM switches that you want to check. You can run the *LPSVM46* rule in the CPF or UPF flow as well. For that purpose, specify this constraint in the SGDC file. The voltage domain information is picked from the corresponding CPF or UPF file.

CPF Commands

- *create_power_domain* (Mandatory)
- *update_power_domain* (Mandatory)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the signal driving enable pin *<en-pin-name>* of the RAM switch *<swt-name>*, which is instantiated as *<swt-inst-name>*, does not originate from an always-on domain:

[LPSVM46_1][WARNING] Signal driving the enable pin '<en-pin-name>' of the RAM Switch '<swt-name>' instantiated as '<swtinstname>' does not originate from an always-on domain

Potential Issues

This violation appears when a signal driving enable pin of a RAM switch originates from a shut-off domain instead of an always-on domain. This is a design error.

Consequences of Not Fixing

If you do not fix this violation, the correct logic value does not reach an enable pin in the power down mode and memory may not function correctly.

How to Debug and Fix

For a graphical view of the violation, double-click the message, and then click **Incremental Schematic**. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, ensure that the RAM switches are instantiated with

proper connections, such that the signals driving enable pin of a RAM switch comes from an always-on domain.

Message 2

The following message appears when the total number of the RAM switch instances <*count1*> , specified by using the `switch_name` argument, is not equal to the total number of the RAM instances <*count2*>, specified by using the `memory_cellnames` argument:

[LPSVM46_2][WARNING] `Number of RAM Switches '<count1>' is not equal to the number of RAM instances '<count2>' in the design`

Potential Issues

This violation appears when the RAM switch exceeds the specified load factor and is not able to drive all memories.

Consequences of Not Fixing

If you do not fix this violation, the memories connected to the RAM switch do not function properly.

How to Debug and Fix

For a graphical view of the violation, double-click the message, and then click **Incremental Schematic**. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, ensure that the total number of the RAM switch instances, specified by using the `switch_name` argument, is equal to the total number of the RAM instances, specified by using the `memory_cellnames` argument.

Example Code and/or Schematic

The signal driving enable pin PWROFF of the RAM switch `LL_SLOGICSWITCH20`, instantiated as `top.mid_inst.Pswitch1`, does not originate from an always-on domain. Therefore, the *LPSVM46* rule reports the following violation.

[WARNING] `Signal driving the enable pin 'PWROFF' of the RAM Switch 'LL_SLOGICSWITCH20' instantiated as 'top.mid_inst.Pswitch1' does not originate from an always-on domain`

The schematic is as follows:

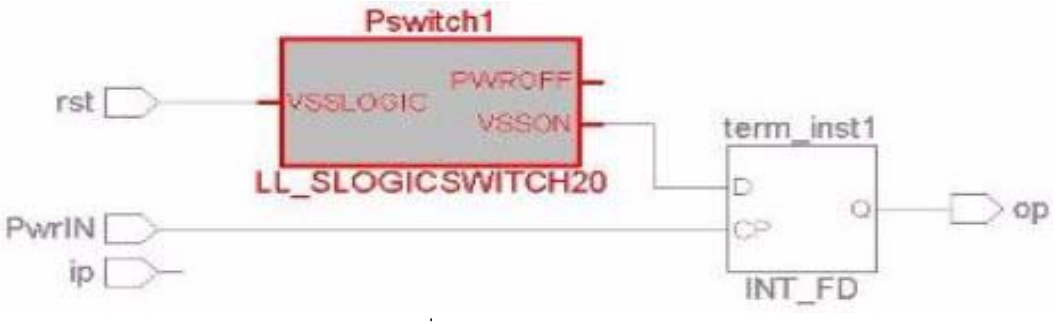


FIGURE 188. Incremental schematic

Default Severity Label

Warning

Rule Group

Power_Domain_Rules

Reports and Related Files

None

LPTIE01

Checks the connections of the always-on normal Tie cells in a design

When to Use

Use this rule to check the connection of the always-on Tie cells. This rule is applicable to all design phases.

This rule is recommended for use with *Gate-level netlist files and their associated library files*, *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*, and *DEF files and their associated LEF files*.

Description

The *LPTIE01* rule checks the output connections of the Tie cells and detects the following wrong connection cases:

- A normal Tie cell driving an always-on pin
- An always-on Tie cell driving a 'not always-on pin' of any cell

Tie cells are the special cells from the library with a constant function in their output pin description, written in the pin scope.

For example, *function*: "0"

For a library cell acting as a buffer or inverter and specified with the *always_on* attribute set as `true` or the *related_power_pin* or *related_ground_pin* attribute set as `back_supply` is treated as an *always_on_buffer*.

Prerequisites

By default, the *LPTIE01* rule is not enabled. To enable this rule, specify the following command in the project file:

```
set_goal_option addrules LPTIE01
```

Language

Verilog, VHDL

Parameter(s)

- *lp_skip_buf*: Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.

Constraint(s)

SGDC Commands

- *voltage_domain* (Mandatory): Use this constraint to specify the power domain with their sleep nets by using the `-sleepnet` argument.
- *always_on_pin* (Mandatory): Use this constraint to specify the `always-on` cell pins.
- *always_on_buffer* (Optional): Use this constraint to specify the `always-on` buffer cells.

CPF Commands

- *create_power_domain* (Mandatory)
- *update_power_domain* (Mandatory)
- *define_always_on_cell* (Mandatory)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)

Messages and Suggested Fix

The following message appears when the instance of the normal or always-on Tie cell drives the always-on or not always-on pin of the cell instance:

[LPTIE01_1][WARNING] <cell-type> tie cell '<inst-name1>'(<cell-name1>) is driving the <pin-type> pin '<pin-name>' of the cell '<inst-name2>'(<cell-name2>)

Potential Issues

This violation appears due to the following reasons:

- A normal Tie cell is connected to an `always-on` pin, which is equivalent to leaving the always-on input pin floating in a particular design state.
- An `always-on` Tie cell is connected to a normal pin, which may cause the normal pin to get powered up accidentally.

Consequences of Not Fixing

If you do not fix this violation, indeterminate input values and accidental power up of input pins can lead to functional errors in the design.

How to Debug and Fix

For a graphical view of the violation, double-click the message and click **Incremental Schematic**. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, ensure that:

- The normal Tie cells in the design drive the not always-on pins.
- The always-on Tie cells in the design drive the `always-on` pins.

Example Code and/or Schematic

The instance `top.ua.Uwe2` of the normal Tie cell `TIELO_X1M_A9TH` is driving the always-on pin of the instance `top.ua.AON` of the cell `Always_On`. Therefore, the *LPTIE01* rule reports a violation.

The schematic is as follows:

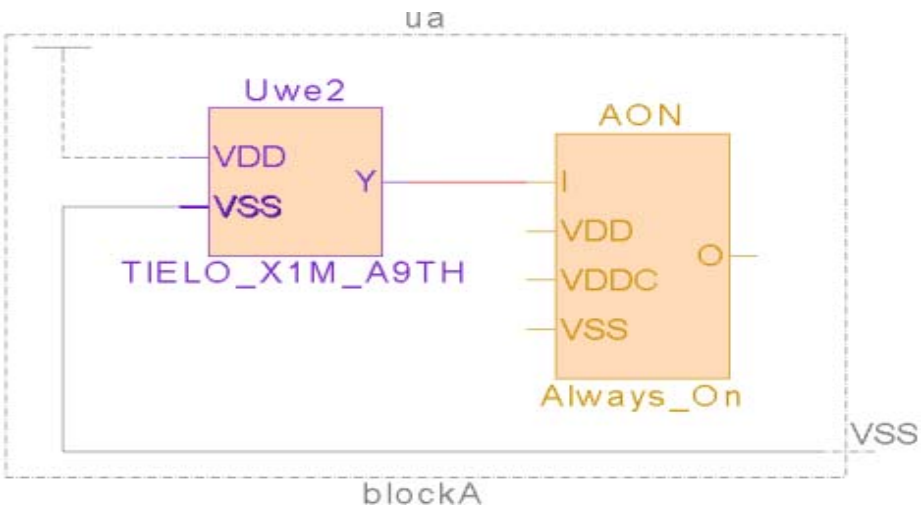


FIGURE 189. Incremental schematic

Default Severity Label

Warning

Rule Group

`Voltage_Domain_Rules`

Reports and Related Files

- *lp_lib_data*: This report provides information about the `always-on` normal Tie cells used in the design.
- *lp_constr_info*: This report provides information about the `always-on` pins.

LPTIE02

Reports the TIE cells that have isolation cells in their path

When to Use

Use this rule when the TIE and isolation cells are present in the design.

This rule is recommended for use with *Gate-level netlist files and their associated library files*, *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*, and *DEF files and their associated LEF files*.

Description

The *LPTIE02* rule reports a violation when an isolation cell is present in the path of the TIE cell. This rule checks if the TIE cells are connected to the isolation cells, when it is not required.

TIE cells are the special cells from the library with a constant function in their output pin description, written in the pin scope.

For example, *function*: "0"

Prerequisites

By default, the LPTIE02 rule is not run while running the SpyGlass Power Verify solution. To enable this rule, specify the following command in the project file:

```
set_goal_option addrules LPTIE02
```

Language

Verilog, VHDL

Parameter(s)

- *lp_skip_buf*: Default value is `1` and the SpyGlass-generated buffers are skipped during rule checking. Set the parameter to `0` to consider SpyGlass-generated buffers during rule checking.

Constraint(s)

SGDC Commands

- *voltage_domain* (Mandatory): Use this constraint to specify the power domains along with their sleep nets by using the `sleepnet` argument.
- *always_on_pin* (Mandatory): Use this constraint to specify the always-on cell pins.
- *always_on_buffer* (Optional): Use this constraint to specify the always-on buffer cells.
- *isolation_cell* (Optional): Use this constraint to specify the isolation cells in the power domains.

CPF Commands

- *create_power_domain* (Mandatory)
- *update_power_domain* (Mandatory)
- *define_always_on_cell* (Mandatory)
- *define_isolation_cell* (Optional)
- *create_isolation_rule* (Optional)
- *update_isolation_rules* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_isolation* (Optional)
- *set_isolation_control* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when the instance *<inst-name>* of the normal TIE cell *<cell-name>* in the *<dom-type>* (`switched` or `always on`) domain is connected to the instance *<iso-inst-name>* of the isolation cell *<iso-cell-name>*:

[LPTIE02_1][WARNING] '<Normal>' tie cell '<inst-name>'(<cell-name>) in the <dom-type> domain '<dom-name>' is connected to the isolation cell '<iso-inst-name>'(<iso-cell-name>)

Potential Issues

This violation appears when the output of the normal TIE cell is connected to the isolation cell and not in the domain with the constant value inputs.

Consequences of Not Fixing

If you do not fix the violation, it may cause functional errors in the design.

How to Debug and Fix

Check the design and the power intent.

To fix this violation, locate the TIE cell in the destination domain.

Message 2

The following message appears when the instance *<inst-name>* of the always-on TIE cell *<cell-name>* in the *<dom-type>* (switched or always on) domain is connected to the instance *<iso-inst-name>* of the isolation cell *<iso-cell-name>*:

[LPTIE02_2][ERROR] 'Always on' tie cell '<inst-name>'(<cell-name>) in the <dom-type> domain '<dom-name>' is connected to the isolation cell '<iso-inst-name>'(<iso-cell-name>)

Potential Issues

This violation appears if the always-on TIE cell is connected to the isolation cell. The error message appears because the always-on TIE cells do not require any isolation.

Consequences of Not Fixing

Presence of the redundant isolation cells may cause functional errors in the design. In addition, there may be placement, routing, and power overheads.

How to Debug and Fix

For a graphical view of the violation, double-click the message and click **Incremental Schematic**.

Check the design and the power intent. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, do not connect the isolation cell to the always-on TIE

cells.

Example Code and/or Schematic

Consider the following example. The violation message appears when the always-on TIE cell `TIELO_X1M_A9TH` is connected to the isolation cell `ISO_AND_EN`:

```
[WARNING] 'Always on' tie cell
'TOP.i_dss_iso.u0_dsi_analogplldiv_iso.INST_TIE'(TIELO_X1M_A9T
H) in the switched domain 'DOMAIN_A' is connected to the
isolation cell 'TOP.i_dss_iso.ISO1'(ISO_AND_EN)
```

The schematic is shown in the following figure:

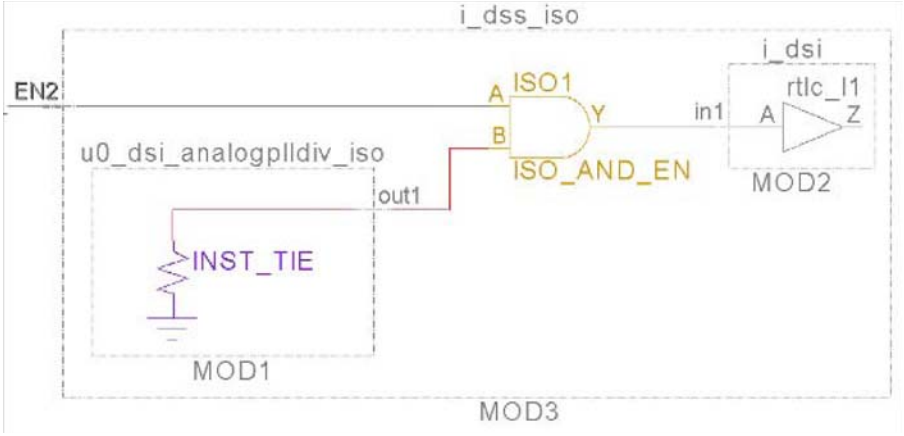


FIGURE 190. Incremental schematic

Default Severity Label

Warning/Error

Rule Group

`Voltage_Domain_Rules`

Reports and Related Files

- *lp_lib_data*: This report provides information about the always-on normal TIE cells used in the design.
- *lp_constr_info*: This report provides information about the always-on pins.

LP_BLACKBOX_CHECK

Checks if percentage of black box elements exceeds the set limit

When to Use

Run this rule to check if ratio of black box instances, with respect to all elements in design exceeds the permissible limits.

This rule is recommended for use with:

- *Gate-level netlist files and their associated library files*,
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*.

Description

The *LP_BLACKBOX_CHECK* rule checks if the ratio of black box instances with respect to the total number of elements exceeds the limit set by `lp_max_blackbox_percentage` parameter.

Language

Verilog, VHDL

Parameter(s)

- *lp_max_blackbox_percentage*: Default value is 100. Set the value of this parameter to any positive integer to report a violation if the ratio of black box elements with respect to all elements in the design exceeds this value.

Constraint(s)

None

Messages and Suggested Fix

The following message is reported when the ratio of black box elements with respect to the total number of elements exceeds the limit set by the `lp_max_blackbox_percentage` parameter:

[LP_BLACKBOX_CHECK_1][FATAL] Black box instance ratio of '<percentage>' exceeds the threshold limit of '<value>'

Potential Issues

None

Consequences of Not Fixing

None

How to Debug and Fix

Providing the correct liberty files will resolve this fatal error. This can also be fixed by updating the value of the *lp_max_blackbox_percentage* parameter.

Example Code and/or Schematic

Suppose, in a given case, the value of the `lp_max_blackbox_percentage` parameter is set to 40. But actual design has 50% of black boxes. Therefore, it reports the following violation message:

```
Black box instance ratio of '50.000' exceeds the threshold limit of '40
```

To avoid this violation, specify correct libraries to decrease the percentage of black boxes. You can also change the value of the `lp_max_blackbox_percentage` parameter to be more than 50.

Default Severity Label

Fatal

Rule Group

Special Purpose Rules

Reports and Related Files

None

LP_MULTI_DOMAIN_CROSSING_CHECK

Ensures that signal paths do not have combinational cells in more than two different domains

When to Use

Use this rule for:

RTL description files with or without library files

Description

This rule generates a spreadsheet in CSV format for signals going from source domain to destination domain, which is crossing an intermediate domain different than the source and destination domain. An intermediate domain is called as a different domain with respect to source/destination domain if there is an isolation/level shifter between intermediate domain and source/destination domain.

Source/destination of the crossing is non-combinational cell/ports. To find the destination, all the combinational cells in the path are skipped.

The spreadsheet viewer is associated with this rule in SpyGlass GUI. Also, the `lp_multi_domain_crossing_check.csv` report is generated in the `spyglass_reports/lowpower` directory.

Language

Verilog, VHDL

Parameter(s)

- *lp_skip_pwr_gnd*: Default value is `1`. Set the parameter to `0` to process nets connected to power/ground supply while checking for the correctness of the level shifters.
- *lp_flag_undriven_nets*: Default value is `0`. Set this value to `1` to check input pins that are not driven.
- *lp_flag_unconnected_nets*: Default value is `0`. Set this value to `1` to check output pins that are not connected.
- *lp_flag_iso_ls_crossing*: Default value is `yes`. Set this value to `no` to report violation for intermediate domain irrespective of isolation/level shifter requirement.

- *lp_flag_rtlc_cell*: Default value is no. Set the value of this parameter to yes to report violation for SpyGlass generated buffers/inverters.

Constraint(s)

None

Messages and Suggested Fix

This rule reports the following message when at least one signal crossing the combinational cells exists in more than two domains and the CSV report is generated:

[INFO] CSV Report 'lp_multi_domain_crossing_check_csv.rpt' containing information of signal which are crossing more than two domains has been generated

Potential Issues

It may lead to a design flaw.

Consequences of Not Fixing

If the signal is crossing more than two domains, there may be a need of more than one isolation/level shifter cells in the path. In that case, the functionality of the second isolation cell may not be correct. So you may break this path by inserting flip-flops between the domain crossings.

How to Debug and Fix

Go to the lp_multi_domain_crossing_check.csv report and review all the rows of this report. Each row contains information of source destination and intermediate domains. Break the crossing between source and intermediate domain or intermediate and destination domain by inserting flip-flops.

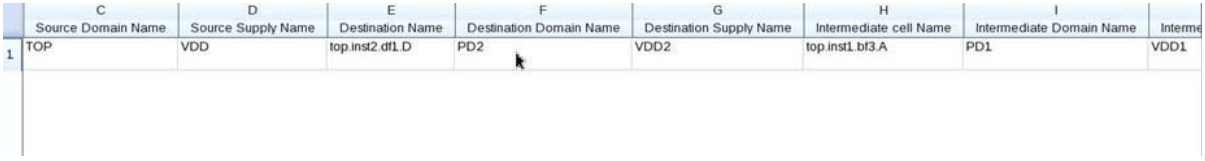
Example Code and/or Schematic

Consider the following Verilog snippet:

```
module top (input iso,isoa,isob,in1,in2,clk,output
out1,out2,iso_o );
assign c=isob;
BUF bf4 (.A(c),.Y(c1));
Test1 inst1
(.in1(c1),.in2(a1),.in3(b2),.out1(w1),.out2(w2),.out3(w3),.o
```

```
ut4(w4));
Test2 inst2 (.in1(w1),.in2(w2),.in3(w3),.in4(w4),.out1(w1));
endmodule

module Test1 (input in1,in2,in3,in4, output
out1,out2,out3,out4);
assign x1=in1;
BUF bf3 (.A(x1),.Y(out1));
endmodule

module Test2 (input in1,in2,in3,in4, output
out1,out2,out3,out4);
DFF df1 (.D(in1),.Q(out1));
endmodule
```

In the above example, three different domains `TOP, PD1, and PD2,` are applied, through UPF. TOP domain is applied on module `top`, `PD1` on `top.inst1`, and `PD2` on `top.inst2`. So the signal `top.isob` is going from TOP domain to PD2 domain through PD1 domain. In this case, the generated *lp_multi_domain_crossing_check* report contains an entry for this signal.



	C	D	E	F	G	H	I	
	Source Domain Name	Source Supply Name	Destination Name	Destination Domain Name	Destination Supply Name	Intermediate cell Name	Intermediate Domain Name	Interme
1	TOP	VDD	top.inst2.df1.D	PD2	VDD2	top.inst1.bf3.A	PD1	VDD1

FIGURE 191. Sample CSV report

Default Severity Label

Info

Rule Group

Special Purpose Rules

Reports and Related Files

lp_multi_domain_crossing_check

LP_INTERMEDIATE_DOMAIN_CROSSING_CHECK

Checks if a crossing between source and destination is going through a feedthrough less-on domain

When to Use

Use this rule to:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The rule identifies all end-to-end crossing skipping buffers (library and SpyGlass generated), inverters, and domain boundary ports. Then, the rule reports a violation if crossings complies any of the conditions shown in the *Figure 192* and *Figure 193*:

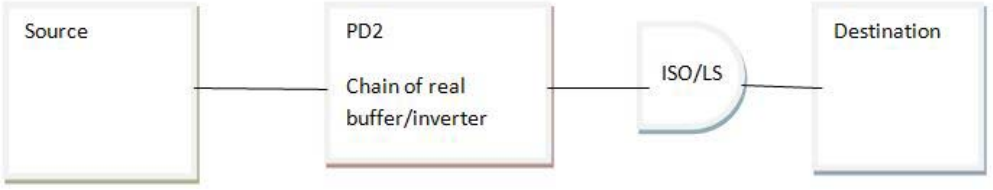


FIGURE 192. Isolation/level shifter strategies at destination side only

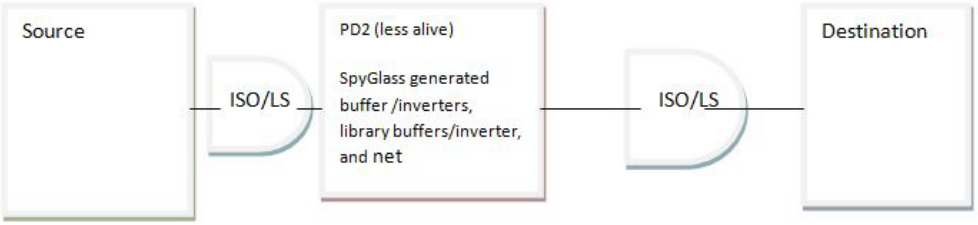


FIGURE 193. Isolation/level shifter strategies on both sides

At the RTL level, the rule checks for the presence of level shifter and isolation strategies in the violating crossing. At the Netlist level, the rule checks for the presence of level shifter and isolation cells in the violating crossing.

Language

Verilog, VHDL

Parameter(s)

- *lp_flag_undriven_nets*: Default value is `0`. Set the parameter to `1` to have this rule consider nets that are not driven at voltage crossings.
- *lp_flag_unconnected_nets*: Default value is `0`. Set the parameter to `1` to have this rule consider unconnected nets at voltage crossings.
- *lp_skip_pwr_gnd*: Default value is `1`. Set the parameter to `0` to process nets connected to power/ground supply while checking for the correctness of the level shifters.
- *lp_set_design_stage*: Default value is `rtl`. Set the value of this parameter to `netlist` or `pg_netlist` to set the desired design stage as Netlist.

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *set_pin_related_supply* (Mandatory)

- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_retention* (Mandatory)
- *set_retention_control* (Mandatory)

Messages and Suggested Fix

The following message appears when a crossing between source and destination is going through a feedthrough less-on domain:

[LP_INTERMEDIATE_DOMAIN_CROSSING_CHECK_1][WARNING] The path between source '<source-name>'('<source-domain-name>') and destination '<destination-name>'('destination-domain-name') is going through '<intermediate-node>'('<intermediate-domain>') which can be OFF while source and destination are still ON

Potential Issues

Due to the presence of a less-on cell between actual source and destination, the crossing between actual source and destination will be split into different crossings with different isolation need. So signal from actual source may not reach actual destination correctly.

Consequences of Not Fixing

This leads to a design flaw.

How to Debug and Fix

Double click on the violation message in the GUI. It will open the schematic of violation message. In the schematic, the intermediate node and domain will be highlighted. You can cross-refer to the RTL file from the intermediate node and rectify the design.

Example Code and/or Schematic

Consider the following example:

```
module top (input in1,in2, iso, output out1);

TEST1 inst1 (.in1(in1),.in2(in2),.out1(w1));
TEST2 inst2 (.in1(w1), .in2(iso), .out1(out1));
endmodule
```

```
module TEST1 (input in1,in2, output out1);
MYAND and1(.A(in1),.B(in2),.Y(w1));
ISO1 iso1(.D(w1),.EN(iso),.Y(out1));
endmodule

module TEST2 (input in1,in2, output out1);
MYAND and1(.A(w1),.B(in2),.Y(w1));
ISO1 iso1(.D(in1),.EN(in2),.Y(w1));
endmodule
```

In the above example, `inst1` is present in the `PD1` domain and `inst2` is present in the `PD2` domain, while `top` is present in `PD3` domain. According to the PST states defined in the UPF, at one state, the `PD3` domain is getting off while `PD1` and `PD2` are on. The wire `top.w1` that is connecting source `inst1` (domain `PD1`) to destination `inst2` (domain `PD2`) is present in the `PD3` domain. So in this case, the rule reports the following violation message:

```
The path between source 'top.inst1.and1.rtlc_I2.Z'('PD1(supply
VDD1:1.000)') and destination
'top.inst2.and1.rtlc_I2.in1'('PD2(supply VDD2:1.000)') is going
through 'top.w1'('TOP(supply VDD:1.000)')  which can be OFF
while source and destination are still ON
```

The following is the schematic for the above example:

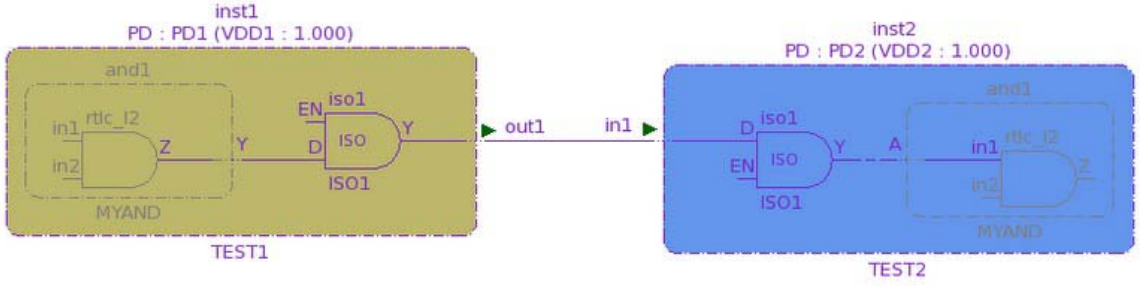


FIGURE 194. Incremental schematic

Default Severity Label

Warning

Rule Group

Special Purpose Rules

Reports and Related Files

None

Powerdata Checking Rules

The powerdata checking rules are as follows:

Rule	Reports
LP_POWERDATA_CHECK	More than one power format are specified
LP_POWERDATA_INFO	Information about data being read successfully from a power format file.
LP_POWERDATA_READ	Reports successful read of power format files

LP_POWERDATA_CHECK

Reports multiple power formats

When to Use

This rule is applicable to all design phases.

Description

The *LP_POWERDATA_CHECK* rule checks that only one of the CPF or UPF format file is given as an input. The rule reports a violation if both CPF and UPF power data format files are provided in the same run of SpyGlass.

In addition, the *LP_POWERDATA_CHECK* rule also reports a violation, if there is an error while reading power format file.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

Message 1

The following message appears when more than one power format files are specified in this rule:

[LP_POWERDATA_CHECK_1][FATAL] Both '<power-format1>' and '<power-format2>' power formats cannot be specified together

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears:

[LP_POWERDATA_CHECK_2][FATAL] Error while reading power format file

Potential Issues

The violation messages appear if your design has files of more than one power formats specified as input for the same run of SpyGlass or when there is an error in reading the power format file.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

Specify only one of the CPF or UPF format file as an input for the same run of SpyGlass.

Example Code and/or Schematic

Not applicable

Default Severity Label

Fatal

Rule Group

`Power Data Checking`

Reports and Related Files

None

LP_POWERDATA_INFO

Reports successful read of power format files

When to Use

This rule is applicable to all design phases.

Description

The *LP_POWERDATA_INFO* rule reports an informational message when power format (CPF/UPF) file has been read successfully. The commands parsed by SpyGlass are reported in the *lp_power_data* report.

It also reports an INFO message stating rules, which do not run in that format.

Language

Verilog, VHDL

Parameter(s)

- *lp_find_objects_match_count*: Default value is 1000. If the SpyGlass hangs during the execution of this rule, reduce the value of the parameter so that they wildcards are not expanded in-line.

Constraint(s)

None

Messages and Suggested Fix

Message 1

The following message appears after a power format (CPF/UPF) file *<format-file-name>* is read successfully:

[LP_POWERDATA_INFO_1][INFO] Power format file '<format-file-name>' has been read

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when we try to run a rule which is not supported for that format (UPF/CPF)

[LP_POWERDATA_INFO_2][INFO] Rule(s) '<rule-list>' will not run in '<pwr-frmt>' format

Where, *<rule-list>* is the list of rules which we tried to run, but were not supported for that particular format.

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

Message 1: There are no consequences if this violation message is not fixed.

Message 2: There are no consequences if this violation message is not fixed. However, you should be aware that the SpyGlass Power Verify solution will not use the rules stated in the violation message for its analysis.

How to Debug and Fix

Review the *lp_power_data* report for the commands parsed by SpyGlass.

Example Code and/or Schematic

Refer to the *lp_power_data* report for the commands parsed by SpyGlass.

Default Severity Label

Info

Rule Group

Power Data Checking

Reports and Related Files

- *lp_power_data*: This report contains information about user-specified CPF/UPF commands parsed by SpyGlass.

LP_POWERDATA_READ

Reports successful read of power format files

When to Use

When user wants to read power format files before synthesis of the design.

Description

The *LP_POWERDATA_READ* rule reports an informational message when the power format (UPF) file has been read successfully. This is an alternative rule for the *LP_POWERDATA_INFO* rule, however it runs at the setup stage. The commands parsed by SpyGlass are reported in the *lp_power_data* report. It also reports an INFO message stating rules that do not run in the UPF format. This rule reports a FATAL message if the `find_objects` UPF command is used, as this command cannot be parsed at the setup stage. The *lp_read_powerdata_at_setup* parameter should be set to `1` to enable this rule.

This rule currently supports UPF only.

Language

Verilog, VHDL

Parameter(s)

- *lp_find_objects_match_count*: Default value is `1000`. If the SpyGlass hangs during the execution of this rule, reduce the value of the parameter so that they wildcards are not expanded in-line.
- *lp_read_powerdata_at_setup*: Default value is `0`. Set this parameter to `1` to enable UPF command parsing at the setup stage.

Constraint(s)

None

Messages and Suggested Fix

Message 1

The following message appears after a power format (UPF) file *<format-file-name>* is read successfully:

[LP_POWERDATA_READ_1][INFO] Power format file '<format-file-name>' has been read

Message 2

The following message appears when we try to run a rule that is not supported for the UPF format:

[LP_POWERDATA_READ_2][INFO] Rule(s) '<rule-list>' will not run in '<pwr-frmt>' format:

Where, *<rule-list>* is the list of rules that are not supported in the UPF format.

Message 3

The following message appears when `find_objects` UPF command is found in any UPF file:

[LP_POWERDATA_READ_3][FATAL] Command 'find_objects' is used in '<file-name>' file. Kindly re-run with spyParameter 'lp_read_powerdata_at_setup' set to no

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

Message 1: There are no consequences if this violation message is not fixed.

Message 2: There are no consequences if this violation message is not fixed. However, you should be aware that the SpyGlass Power Verify solution will not use the rules stated in the violation message for its analysis.

Message 3: SpyGlass cannot be run without fixing this as this is a FATAL violation.

How to Debug and Fix

Review the *lp_power_data* report for the commands parsed by SpyGlass.

Example Code and/or Schematic

Refer to the *lp_power_data* report for the commands parsed by SpyGlass.

Default Severity Label

Info, Fatal

Rule Group

Power Data Checking

Reports and Related Files

lp_power_data: This report contains information about user-specified UPF commands parsed by SpyGlass.

Detailed Reporting Rules

The Detailed Reporting rules are as follows:

Rule	Reports
LP_DECOMPILE_CONSTR	Information about user-specified LowPower constraints as interpreted by SpyGlass
LP_CROSSING_DATA	Information about crossing within SGDC/CPF/ UPF formats
LP_LIB_DATA	Information about power related attributes available in library
LpWildCardMatchReport	Matches found for SGDC constraints with wildcards
PairWiseVDCrossing	Voltage domain crossings
vdPDInfo	Voltage domain and power domain related information
LP_ISO_REPORT	Generates spreadsheet for supported rules related to isolation strategy and isolation cells
LP_LSH_REPORT	Generates spreadsheet for supported rules related to level shifter strategy and level shifter cells

LP_DECOMPILE_CONSTR

Reports details of user-specified SpyGlass Power Verify constraints as interpreted by SpyGlass

When to Use

To report the power information provided in the power format file. This rule applicable to all design phases. This rule is recommended for use with:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *LP_DECOMPILE_CONSTR* rule generates the *lp_constr_info* report For details on reports, refer to the *SpyGlass Power Verify Reports* section.

This report contains details of the power domains and their operating voltages, supplies and their association with power domains, isolation cells, level shifter cells, retention cells, and power switch cells.

Rule Exceptions

The information about the constraints containing wildcards in not provided by the rule. The *LpWildCardMatchReport* rule generates wildcard report.

Language

Verilog, VHDL

Parameter(s)

lp_dump_scope_supply_in_domain_info: Default is `no`. Set this parameter to `yes` to include information on the scope supplies in the `lp_domain_info.rpt` report.

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage/ power domains in the design.
- *levelshifter* (Mandatory): Use this constraint to specify the names of design units to be used as level shifters.
- *isolation_cell* (Mandatory): Use this constraint to specify the isolation cells in power domains.
- *input_isocell* (Mandatory): Use this constraint to specify the isolation cells at inputs of a power domain.
- *ignore_crossing* (Mandatory): Use this constraint to specify the power domain-to-voltage domain crossings and power domain-to-power domain crossings that should be ignored.

CPF Commands

- *create_power_domain*
- *create_level_shifter_rule*
- *create_nominal_condition*
- *create_power_mode*
- *update_level_shifter_rules*
- *define_level_shifter_cell*
- *define_power_clamp_cell*
- *create_state_retention_rule*
- *update_state_retention_rules*
- *create_isolation_rule*
- *update_isolation_rules*
- *define_isolation_cell*
- *define_always_on_cell*

UPF Commands

- *create_power_domain*
- *create_supply_port*

- *set_pin_related_supply*
- *create_supply_net*
- *connect_supply_net*
- *set_domain_supply_net*
- *set_level_shifter*
- *map_level_shifter_cell*
- *create_pst*
- *add_pst_state*
- *set_retention*
- *set_retention_control*
- *set_isolation*
- *set_isolation_control*
- *map_isolation_cell*

Messages and Suggested Fix

Message 1

When the *lp_constr_info* report is generated, the following message appears:

[LP_DECOMPILE_CONSTR_1][Info] LowPower constraints interpretation information is generated in report 'lp_constr_info'

Message 2

When the *lp_domain_info* report is generated, the following message appears:

[LP_DECOMPILE_CONSTR_2][Info] LowPower Domain related information is generated in report 'lp_domain_info'

Default Severity Label

Info

Rule Group

VoltageDomainInformation

Reports and Related Files

- *lp_constr_info*

LP_CROSSING_DATA

Populates crossing information from the power intent specified in SGDC / CPF / UPF file

When to Use

This rule is applicable to all design phases.

Description

The *LP_CROSSING_DATA* rule populates the domain crossing information from the power intent specified in the SGDC/UPF/CPF files. The domain crossing data is in the *lp_crossing_data* report, while the power state information is in the *lp_power_state* report. View these reports in the `spyglass_reports/lowpower` directory or in the `consolidated_reports` directory.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The *LP_CROSSING_DATA* rule generates the following message:

[INFO] `Domain crossings data is generated in the report 'lp_crossing_data'`

Potential Issues

Not applicable

Consequences of Not Fixing

Not applicable

How to Debug and Fix

Double click the violation message to view the *lp_crossing_data.csv* report in

the spreadsheet viewer. You can also review the domain crossings data reported in the *lp_crossing_data.rpt* report located in the lowpower directory.

Example Code and/or Schematic

This example illustrates the *lp_crossing_data* report generated for a sample UPF file. The UPF power intent file is as follows:

```
set_design_top TOP
create_power_domain TOP  -include_scope
create_power_domain LOW  -elements /low_domain
create_power_domain HIGH -elements /high_domain

create_supply_port VDDT
add_port_state VDDT -state {active 1.4}
create_supply_net  VDDT -domain TOP
connect_supply_net VDDT -ports VDDT

create_supply_port VDDH
add_port_state VDDH -state {active 1.2}
create_supply_net  VDDH -domain HIGH
connect_supply_net VDDH -ports VDDH

create_supply_port VDDL
add_port_state VDDL -state {active 0.8} -state {off off}
create_supply_net  VDDL -domain LOW
connect_supply_net VDDL -ports VDDL

create_supply_port VSS
add_port_state VSS -state {active 0.0}
create_supply_net  VSS -domain TOP
create_supply_net  VSS -domain LOW  -reuse
create_supply_net  VSS -domain HIGH –reuse
connect_supply_net VSS -ports VSS

set_domain_supply_net HIGH -primary_power_net VDDH -
primary_ground_net VSS
set_domain_supply_net LOW  -primary_power_net VDDL -
primary_ground_net VSS
set_domain_supply_net TOP -primary_power_net VDDT -
primary_ground_net VSS
```

```
set_isolation ISO1 -domain LOW -applies_to outputs

set_isolation_control ISO1 -domain LOW  -isolation_signal iso
-isolation_sense low -location parent

map_isolation_cell ISO1 -domain LOW -lib_cells { ISO_AND_EN }

create_pst PST -supplies { VDDT VDDL VDDH VSS }
add_pst_state PS1 -pst PST -state { active active active
active }   add_pst_state PS2 -pst PST -state { active off
active active }
```

For the above example, the following CSV report is generated in the spreadsheet viewer:

A	B	C	D	E	F	G	H	I	J
Input Domain	/oltage Range	/oltage Range	Output Domair	Voltage Range	Voltage Range	hifter(Type Re	hifter Cells av	Isolation(Required)	Cross Probe
TOP	1.4	1.4	HIGH	1.2	1.2	High to Low	None	Not Required	Show Relationship
TOP	1.4	1.4	LOW	0.8	0.8	High to Low	None	Input Isolation	Show Relationship
LOW	0.8	0.8	TOP	1.4	1.4	Low to High	None	Output Isolation	Show Relationship
LOW	0.8	0.8	HIGH	1.2	1.2	Low to High	None	Output Isolation	Show Relationship
HIGH	1.2	1.2	TOP	1.4	1.4	Low to High	None	Not Required	Show Relationship
HIGH	1.2	1.2	LOW	0.8	0.8	High to Low	None	Input Isolation	Show Relationship

FIGURE 195. Sample CSV report

Default Severity Label

Info

Rule Group

Detailed Reporting

Reports and Related Files

- *lp_crossing_data*: This contains domain crossing data information as inferred by SpyGlass Power Verify using the SGDC/CPF/UPF files and library files.This report is generated in two variants.
  - *lp_crossing_data.csv*

- *lp_crossing_data.rpt*

- *lp_power_state*: This shows the power state information inferred by SpyGlass Power Verify from the power intent SGDC/CPF/UPF files.

LP_LIB_DATA

Reports information of power related attributes available in library

When to Use

This rule is applicable to all design phases.

Description

The *LP_LIB_DATA* rule reports information of the power related attributes, available in the library. The rule generates the *lp_lib_data* report located in the lowpower directory.

For the CPF flow, if the information present in the library is overridden by any CPF command, the latter gets precedence and the information provided in the CPF file is used.

For the UPF flow, if the information present in the library is overridden by `set_pin_related_supply` command, the latter gets precedence and the information provided in the UPF file is used.

Language

Verilog, VHDL

Parameter(s)

- *lp_disable_lib_attr_read*: Default value is `0`. Set this value to `1` to disable the data to be auto-inferred from the library.

Constraint(s)

None

Messages and Suggested Fix

The *LP_LIB_DATA* rule generates the following message:

[INFO] `Information of power related attributes available in library is generated in report 'lp_lib_data'`

Potential Issues

Not applicable

Consequences of Not Fixing

Not applicable

How to Debug and Fix

Review the information of power related attributes available in the library as reported in the *lp_lib_data* report located in the lowpower directory.

Example Code and/or Schematic

The *lp_lib_data* report shows comprehensive library information with respect to Level Shifter, Isolation, Always-on, Retention, Multiple Supply, Power Switch, and Tie cells. Refer to *lp_lib_data for more details.*

Default Severity Label

Info

Rule Group

Detailed Reporting

Reports and Related Files

- *lp_lib_data*: This report contains information from the library files as interpreted by SpyGlass Power Verify.

LpWildCardMatchReport

Report the matches found for SGDC constraints with wildcards

When to Use

This is a setup rule and always runs by default.

Description

The *LpWildCardMatchReport* rule reports the matches found for SGDC constraints using wildcards (* or ?) in the *lp_wild_card* report.

Language

Verilog, VHDL

Constraint(s)

None

Parameter(s)

- *lp_no_report*: Default value is `no`. Set the parameter to `yes` to disable report generation.
- *lp_report_all_wildcards*: Default value is `0`, and the report does not contain matches found for `instname`, `portname`, and `netname` arguments. Set the parameter to `1` to enable the report generation of matches found for these arguments with wildcards of the *voltage_domain* constraint.
- *lp_wildcard_report_count*: Specifies the maximum number of desired matches to be used for `instname` and `netname` arguments. Default value is `30`.

Messages and Suggested Fix

The following message appears:

[INFO] `Wildcard fields are expanded in lp_wild_card report`

Potential Issues

Not applicable

Consequences of Not Fixing

Not applicable

How to Debug and Fix

Review the *lp_wild_card* report through the Reports menu in the Atrenta Console GUI.

Example Code and/or Schematic

The *lp_wild_card* report shows the matches found for SGDC constraints using wildcards (* or ?). Refer to *lp_wild_card for more details.*

Default Severity Label

Info

Rule Group

Detailed Reporting

Reports and Related Files

- *lp_wild_card*: Contains the matches found for the cell names specified in the SGDC constraints with wildcards (* or ?).

PairWiseVDCrossing

Reports voltage domain crossings that do not have level shifters

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/ LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *PairWiseVDCrossing* rule checks the voltage domain information from the *voltage_domain* constraints in a SpyGlass Design Constraints file and reports voltage domain crossings among these voltage domains found in the design.

Prerequisites

By default, the *PairWiseVDCrossing* rule is not enabled. To enable this rule, specify the following command in the project file:
set_goal_option addrules PairWiseVDCrossing

Rule Exceptions

The *PairWiseVDCrossing* rule does not report voltage domain crossings with level shifters.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain* (Mandatory): Use to specify voltage/ power domains in the design.

Messages and Suggested Fix

Message 1

The following message appears when voltage domain crossings are found in the design unit <*du-name*>:

```
[INFO] Voltage Domain Crossings for design (<du-name>) exists
for the following pair(s) [from to] : [<vd1name> <vd2-name>]
...
```

Where, <*vd1-name*>, <*vd2-name*>, ... are the names of voltage domains in the design and there exists a voltage domain crossing between these voltage domains.

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when no voltage domain crossings are found in the design unit <*du-name*>:

```
[INFO] No voltage Domain Crossings for design (<du-name>)
```

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

As this is an informational message, there is no implicit impact of not fixing this violation message.

How to Debug and Fix

The constraint reported in the violation message is highlighted in the Atrenta Console GUI.

If required, update the constraint to include level shifters at the voltage domain crossings.

Example Code and/or Schematic

There is no schematic. This rule only provides information of all the voltage domain crossings existing in the design.

Default Severity Label

Info

Rule Group

Detailed Reporting

Reports and Related Files

None

vdPDInfo

Reports voltage domain and power domain information for the design

When to Use

Use this rule to detect:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

The *vdPDInfo* rule reads the user-supplied *voltage_domain* constraints information in the SpyGlass Design Constraints file and reports the following:

- Names of the instances specified for the voltage and power domains in the design.
- Level shifters from low-to-high voltage domains, if the *LPSVM04A* rule is also run.
- Level shifters from high-to-low Voltage Domains, if the *LPSVM04B* rule is also run.
- Steady-state values at the outputs of isolation cells of each power domain, if the *LPSVM10* rule is also run.

The *vdPDInfo* rule prints the voltage domain and power domain information for each design unit in a the *lp_vd_info* report. You can view this report from the Report menu in Atrenta Console.

Prerequisite

By default, the *vdPDInfo* rule is not enabled. To enable this rule, specify the following command in the project file:

```
set_goal_option addrules vdPDInfo
```

Rule Exceptions

The *vdPDInfo* rule is not supported for CPF and UPF mode. Therefore, no

data is provided in the *lp_vd_info* report for CPF and UPF modes.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message is generated for each design unit <*du-name*>:

[INFO] Voltage Domain and Power Domain information for design '<du-name>' reported in report 'lp_vd_info'

Potential Issues

Not applicable

Consequences of Not Fixing

None

How to Debug and Fix

The lp_vd_info report is generated. The report contains the voltage domain and power domain information for each design unit. You can view this report from the Report Menu in Atrenta Console.

Example Code and/or Schematic

The *vdPDInfo* rule generates the *lp_vd_info* report. Click *lp_vd_info* to view a sample report.

Default Severity Label

Info

Rule Group

Detailed Reporting

Reports and Related Files

- *lp_vd_info*: The report contains the following details:
  - Voltage and power domain count
  - Gate count information for voltage domains
  - Lists all instances belonging to each voltage and power domain

LP_ISO_REPORT

Generates spreadsheet for supported rules related to isolation strategy and isolation cells

When to Use

This is a setup rule and always runs by default.

Description

This rule generates a spreadsheet in CSV format for the supported rules related to isolation strategy and isolation cells.

This spreadsheet contains detailed information of all the violations reported by the *LPISO04A*, *LPISO04B*, *LPISO04C*, *LPISO05A*, *LPISO05B*, *LPSVM08B*, *LPSVM08C*, *LPSVM22*, *LPSVM60*, and *LPSVM08A* rules.

NOTE: *Some isolation rules are not supported since their violations are not crossing specific. These rules are LPISO01, LPISO02, LPISO04D, LPISO06A, LPISO06B, LPISO07, LPSVM09, LPSVM10, LPSVM12A, LPSVM12B, LPSVM15, LPSVM26, LPSVM28, LPSVM31, LPSVM47, LPSVM48, LPSVM50, LPSVM51, LPSVM52, and LPSVM55.*

The spreadsheet viewer is associated with this rule in SpyGlass GUI. Also, the `lp_isolation_spreadsheet.csv` report is generated in the `spyglass_reports/lowpower` directory.

Rule Exceptions

This rule is supported only in UPF format, and not in SGDC and CPF formats.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

Message 1

The following information message is reported when any violation is reported by the supported *Isolation Logic Rules*:

```
[INFO] Isolation related spreadsheet
'lp_isolation_spreadsheet.csv' has been generated
```

Message 2

The following information message is reported when isolation spreadsheet has not been generated because none of the supported isolation rules is specified:

```
[INFO] Isolation spreadsheet has not been generated as none of
the isolation rules are enabled
```

Message 3

The following information message is reported when isolation spreadsheet has not been generated because there are no violations reported for the specified isolation rules:

```
[INFO] Isolation spreadsheet has not been generated as no
isolation violations are reported
```

Potential Issues

Not applicable

Consequences of Not Fixing

None

How to Debug and Fix

None

Example Code and/or Schematic

The following is a sample snapshot of the `lp_isolation_spreadsheet.csv` report.

NOTE: *For the full details of the spreadsheet report, see the table given after this snapshot.*

ISO Needed	O Strategy Nan	Strategy Check	ISO Cell Name	Cell Check	e prefix/Name s	Reason	Rule Name
Yes	NA	NA	NA	NA	NA	No isolation ...	LPISO03A
Yes	NA	NA	NA	NA	NA	No isolation ...	LPISO03A
Yes	Not Present	Missing	NA	NA	NA	Strategy not...	LPISO04A
Yes	Not Present	Missing	NA	NA	NA	Strategy not...	LPISO04A
Yes	Not Present	Missing	NA	NA	NA	Strategy not...	LPISO04A
Yes	Not Present	Missing	NA	NA	NA	Strategy not...	LPISO04A
Yes	Not Present	Missing	NA	NA	NA	Strategy not...	LPISO04A
Yes	NA	NA	NA	NA	NA	No isolation ...	LPISO03A
Yes	NA	NA	NA	NA	NA	No isolation ...	LPISO03A
Yes	Not Present	Missing	NA	NA	NA	Strategy not...	LPISO04A
Yes	NA	NA	NA	NA	NA	No isolation ...	LPISO03A
Yes	NA	NA	NA	NA	NA	No isolation ...	LPISO03A

FIGURE 196. Sample CSV report

The following table contains the details of the information provided in this spreadsheet report:

Column Name	Possible Values	Information Provided
Src File(s) (RTL/ UPF)	Name of the RTL/UPF file reported in violation message	All supported isolation rules.
		If more than one rule is merged onto one row, then file name is unique and separated by ";".
Line No	Line number of the RTL/UPF file reported in violation message	All supported isolation rules.
		If more than one rule is merged onto one row, then the line number is unique and separated by ";".

Column Name	Possible Values	Information Provided
Source Name, Source Domain, Source Supply Name, Source Supply Value, Destination Name, Destination Domain, Destination Supply Name, Destination Supply Value	Any	All isolation rules. If more than one rule is merged onto one row, then value of these columns is the same as each row is crossing specific.
ISO Needed	Y/N	LP_CROSSING_DATA
ISO Strategy Name	Not Present	LPISO04A
	<Strategy-Name>	LPISO04B/LPISO04C (if strategy is wrong), LPISO05 (if strategy is redundant), LPSVM08, LPSVM60 (in all other cases)
	NA	LPSVM08*, LPSVM22
Strategy Check	Incorrect	LPISO04B/LPISO04C/LPSVM08B
	Missing	LPISO04A/LPSVM08
	Redundant	LPISO05*
	Correct	LPSVM08/LPSVM60
	NA	In all other cases
ISO Cell Name	NA	LPISO04*, LPSVM08*, LPISO05*
	Cell_Inst_Name:Lib_Cell_Name	LPSVM08A, LPSVM60, LPSVM22 (if violation mentions an ISO cell)
	Not Present	LPSVM08A, LPSVM60, LPSVM22 (if violation does not mention an ISO cell)
Cell Check	Incorrect	LPSVM08A/LPSVM60
	Missing	LPSVM60/LPSVM08*
	Redundant	LPSVM22
	Correct	LPSVM22
	NA	In all other cases

Column Name	Possible Values	Information Provided
Name prefix/ suffix	Prefix and/or Suffix mismatch	LPSVM08A, LPSVM60
	NA	LPISO*, LPSVM22
Reason	Any	All isolation rules. The reason is an elaborated versions of the issues.
		If more than one rule is merged onto one row, then the reasons are separated with “\n”.
Rule Name	Name of the rule from which this rule row has been populated.	All isolation rules.
		If more than one rule is merged onto one row, then the rule names are separated with “;”.

Default Severity Label

Info

Rule Group

Detailed Reporting

Reports and Related Files

Refer to the *The LP-report Report* section.

LP_LSH_REPORT

Generates spreadsheet for supported rules related to level shifter strategy and level shifter cells

When to Use

This is a setup rule and always runs by default.

Description

This rule generates a spreadsheet in CSV format for the supported the rules related to level shifter strategy and level shifter cells.

The spreadsheet contains detailed information of all the violations reported by the *LPLSH03*, *LPLSH04*, *LPLSH05A*, *LPLSH05B*, *LPLSH07*, *LPSVM04*, *LPSVM04A*, *LPSVM04B*, *LPSVM04C*, *LPSVM04D*, *LPSVM04E*, and *UPF_lowpower21* rules.

NOTE: *Some level shifter rules are not supported since their violations are not crossing specific. These rules are LPPLIB04, LPPLIB05, LPPLIB07, LPPLIB08, LPLSH01, LPLSH02, LPLSH06, LPSVM17, and LPSVM24.*

The spreadsheet viewer is associated with this rule in SpyGlass GUI. Also, the `lp_levelshifter_spreadsheet.csv` report is generated in the `spyglass_reports/lowpower` directory.

Rule Exceptions

This rule is supported only in UPF format, and not in SGDC and CPF formats.

The LPLSH07 does not have a crossing but still it gets populated in the spreadsheet report. In this case, the source and destination values are `NA`.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

Message 1

The following information message is reported when any violation is reported by the supported *Level Shifter Rules*:

```
[INFO] Level Shifter spreadsheet
'lp_levelshifter_spreadsheet.csv' has been generated
```

Message 2

The following information message is reported when level shifter spreadsheet has not been generated because none of the supported level shifted rules is specified:

```
[INFO] Level shifter spreadsheet has not been generated as none
of the level shifter rules are enabled
```

Message 3

The following information message is reported when level shifter spreadsheet has not been generated because there are no violations reported for the specified level shifter rules:

```
[INFO] Level shifter spreadsheet has not been generated as no
level shifter violations are reported
```

Potential Issues

Not applicable

Consequences of Not Fixing

None

How to Debug and Fix

None

Example Code and/or Schematic

The following is a sample snapshot of the `lp_levelshifter_spreadsheet.csv` report.

NOTE: *For the full details of the spreadsheet report, see the table given after this snapshot.*

LS Type	S Strategy Nam	Strategy Check	LS Cell Name	Cell Check	e prefix/Name s	Location Check	Reason	Rule Name
HL	NA	NA	NA	Missing	NA	NA	Missing Lev...	LPSVM04B
NA	NA	NA	NA	NA	NA	NA	no valid lev...	LPLSH07
LH;HL	NA	Missing	NA	NA	NA	NA	LH strategy ...	LPLSH05A;LPL...
HL	NA	Missing	NA	NA	NA	NA	HL strategy ...	LPLSH05B
LH	NA	Missing	NA	NA	NA	NA	LH strategy ...	LPLSH05A
HL	NA	Missing	NA	NA	NA	NA	HL strategy ...	LPLSH05B
LH	NA	NA	NA	Missing	NA	NA	Missing Lev...	LPSVM04A
NA	NA	NA	NA	NA	NA	NA	no valid lev...	LPLSH07
NA	NA	NA	NA	NA	NA	NA	no valid lev...	LPLSH07
HL	NA	NA	NA	Missing	NA	NA	Missing Lev...	LPSVM04B
LH	NA	Missing	NA	NA	NA	NA	LH strategy ...	LPLSH05A
LH;HL	NA	Missing	NA	NA	NA	NA	LH strategy ...	LPLSH05A;LPL...
LH	NA	NA	NA	Missing	NA	NA	Missing Lev...	LPSVM04A
HL_LH	NA	NA	NA	Missing	NA	NA	Missing Lev...	LPSVM04A
HL_LH	NA	NA	NA	Missing	NA	NA	Missing Lev...	LPSVM04A

FIGURE 197. Sample CSV report

The following table contains the details of the information provided in this spreadsheet report:

Column Name	Possible Values	Information Provided
Src File(s) (RTL/UPF)	Name of the RTL/UPF file reported in the violation message	All supported level shifter rules.
		If more than one violations belong to the same source and destination (crossing), then the file name unique and separated by “;”.

Column Name	Possible Values	Information Provided
Line No	Line number of the RTL/ UPF file reported in violation message	All supported level shifter rules.
		If more than one rule is merged onto one row, then the line number is unique and separated by ";".
Source Name, Source Domain, Source Supply, Destination Name, Destination Domain, Destination Supply	Any	All level shifter rules
		For LPLSH07, source and destination names are NA
		If more than one rule is merged onto one row, then value of these columns is the same as each row is crossing specific.
LS Needed	Y/N	LP_CROSSING_DATA
LS Type	LH, HL, LH_HL	LP_CROSSING_DATA
LS Strategy Name	NA	In all other rules
	<Strategy-Name>	LPLSH5A, LPLSH05B (if strategy is wrong)
		LPLSH04 (if strategy is redundant)
		LPLSH03 (if strategy is correct but level shifter is not present)
		UPF_lowpower21 (if strategy is correct but source/destination supplies are incorrect)

Column Name	Possible Values	Information Provided
Strategy Check	Incorrect	LPLSH05A, LPLSH05B (if strategy is wrong)
	Missing	LPLSH05A, LPLSH05B (if strategy is missing)
	Redundant	LPLSH04 (if strategy is redundant)
	Correct	UPF_lowpower21 (if strategy is correct but source/destination supplies are incorrect)
		LPLSH03 (if strategy is correct but level shifter is not present)
	NA	In all other cases.
LS Cell Name	NA	LPSVM04A/LPSVM04B/ LPSVM04D (for missing messages)
	Cell_Inst_Name:Lib_Cell_ Name	LPSVM04A/LPSVM04B/ LPSVM04D (if cell name does not match with map_level_shifter_cell)
		LPSVM04C (if cell is redundant)
		LPSVM04E (if cell is incorrectly placed)
Cell Check	Incorrect	LPSVM04A/LPSVM04B, LPSVM04D (if cell name does not match with map_level_shifter_cell).
	Missing	
	NA	In all other cases.
Name prefix/name suffix	Prefix and/or suffix mismatch	LPSVM04A, LPSVM04B
	NA	In all other cases
Location Check	Incorrect	LPSVM04E (Location) incorrect.
	NA	

Column Name	Possible Values	Information Provided
Reason	Any	All level shifter rules. The reason is an elaborated version of the issue. (for example, for LPLSH07 “no valid level shifter in library”).
		If more than one rule is merged onto one row, then the reason is separated with new line.
Rule name	Name of the rule from which this rule row has been populated.	Related level shifter rule name.
		If more than one rule is merged onto one row, then the rule names are separated with “;”.

Default Severity Label

Info

Rule Group

Detailed Reporting

Reports and Related Files

Refer to the *The LP-report Report* section.

Constraints Checking Rules

The Constraints Checking Rules are as follows:

Rule	Reports
LP_CHECK_CONSTR	Validates information specified for level shifter, isolation and retention cells with respect to information specified in library using attributes
LP_SGDC_CHECKS	Checks the existence of cells specified with cell_hookup and aon_buffered_signals
LpParamSanityCheck	Checks the user-specified rule parameter values.
SGDC_voltagedomain01	Non-existent instances specified with the `-instname` argument of the `voltage_domain` constraint
SGDC_voltagedomain02	Non-existent design units specified with the `-modname` argument of the `voltage_domain` constraint
SGDC_voltagedomain03	Non-existent ports specified with the `-portname` argument of the `voltage_domain` constraint
SGDC_voltagedomain04	Non-existent/incorrect signals specified with the `-isosig` argument of the `voltage_domain` constraint
SGDC_lowpower05	`voltage_domain` constraints without the required arguments specified
SGDC_lowpower06	`voltage_domain` constraints where instance names are incorrectly specified with the `-instname` argument
SGDC_lowpower07	`voltage_domain` constraints with incorrect or incomplete isolation information
SGDC_voltagedomain05	Incorrect values specified with the `-isoval` argument of the `voltage_domain` constraints
SGDC_lowpower09	Instance names specified with more than one `voltage_domain` constraint

Rule	Reports
SGDC_voltagedomain06	Incorrect values specified with the `-value` argument of the `voltage_domain` constraints
SGDC_lowpower12	Redundant arguments specified with the `voltage_domain` constraints while defining a voltage domain
SGDC_voltagedomain07	Incorrect OFF value specified with the `voltage_domain` constraints while defining a power domain
SGDC_lowpower15	`voltage_domain` constraints that are defining a power domain but the `-isosig` argument is not specified
SGDC_lowpower17	`voltage_domain` constraints defining power domains where the `-generate_iso_logic` argument is specified but the `-outputs` argument is not specified
SGDC_lowpower18	`voltage_domain` constraints defining power domains where the steady-state condition name specified with the `-outputs` argument has not been defined with the `-name` argument of a `domain_outputs` constraint under the same environment
SGDC_lowpower19	`voltage_domain` constraints defining power domains where the supply signal name specified with the `-supplyname` argument has not been defined with the `-name` argument of a `supply` constraint under the same environment
SGDC_voltagedomain08	Non-existent signals specified with the `-clkdomain` argument of the `power_switch` constraint
SGDC_lowpower23	Incorrect signal values specified with the `-value` argument of the `domain_outputs` constraints

Rule	Reports
SGDC_lowpower24	Incorrect voltage domains specified with the `-to/-from` arguments of the `levelshifter` constraints
SGDC_supply01	Missing or incorrect value specified with the `-value` argument of the `supply` constraint
SGDC_lowpower30	Signal names that are repeated in the signal list of the `-value` argument of a `domain_outputs` constraint
SGDC_lowpower31	Incorrect top specifications
SGDC_lowpower32	Existence check of the power-out and power-in terminal of power switch
SGDC_lowpower40	Performs sanity check for the existence of the domains argument of retention_cell
SGDC_lowpower47	Non-existent pins specified with the `-vddpin/-vddcpin` argument of the *retention_cell* constraint
SGDC_lowpower48	Incorrect values specified with the `-seqvalue` argument of the `domain_signal` constraint
SGDC_lowpower52	`voltage_domain` constraints defining power domains when both `-isosig` and `-noisosig` arguments are specified
SGDC_lowpower59	Non-existent terminals specified with the `-enableports` argument of the `voltage_domain` constraint
SGDC_lowpower60	Non-existent objects specified with the `ram_switch` constraint
SGDC_lowpower61	Incorrect `input_isocell` constraints specifications
SGDC_lowpower62	Incorrect `cell_pin_info` constraint specifications
SGDC_lowpower65	Incorrect supply values

Rule	Reports
SGDC_lowpower66	power_switch constraints with non-existent pins specified with the -en_inv_in, -en_inv_out, and -en_buf_in arguments
SGDC_lowpower67	levelshifter constraints with non-existent pins specified with the -inSupplyTerm and -outSupplyTerm arguments
SGDC_lowpower68	special_cell constraints without the rule-specific arguments specified
SGDC_lowpower69	Constraints specification with a different current_design
SGDC_lowpower71	levelshifter constraints with invalid -inTerm/-outTerm arguments
SGDC_lowpower72	pin_voltage constraints with incorrect arguments specified
SGDC_lowpower75	Non-existent pins specified with the -names argument of the pin_voltage constraint
SGDC_lowpower77	Cells that are specified more than once with the -cell argument of the always_on_pin constraint
SGDC_lowpower78	Incorrect specifications of the arguments in the cell_tie_class constraint
SGDC_lowpower82	Incorrect or missing sleep signals for power domains
SGDC_lowpower85	Incorrect value specified with the -stopclockval argument of the voltage_domain constraint
SGDC_lowpower86	Incorrect value specified with the -clkval argument of the *retention_cell* constraint
SGDC_lowpower87	Incorrect supply names specified with the -name argument of the supply constraint

Rule	Reports
SGDC_lowpower89	Incorrect voltage domains specified with the `-to/-from` arguments of the `ignore_crossing` constraint.
SGDC_lowpower90	Sanity check to ensure that constraints are given properly while running LPSVM58
SGDC_lowpower91	If steady-state condition names is not correctly specified
SGDC_lowpower92	Non-existent power down condition names, specified using the `domain_inputs` constraint and with `-inputs` argument of the `voltage_domain` constraint, while running the LPSVM47 rule.
SGDC_lowpower93	Incorrectly specified `levelshifter` constraint.
SGDC_lowpower94	Incorrectly specified `power_state` constraint.
SGDC_lowpower95	Incorrectly specified `-tie1` and `-tie0` arguments using the `cell_tie_class` constraint.
SGDC_lowpower96	Non-existent module names that are specified using the `-name` argument of the `levelshifter` constraint.
SGDC_lowpower97	Sanity check to ensure that constraints are given properly while running LPSVM59.
SGDC_lowpower98	Existence check of pins specified with '-pin' argument of `input_isocell` constraint.
SGDC_lowpower99	Flags if domain name specified by '-belongsto' argument of 'isolation_cell' constraint is not a valid powerdomain.
SGDC_lowpower100	Flags if domain name specified by '-domain' argument of 'power_down' constraint is not a valid powerdomain.
SGDC_lowpower101	Flags if 'vddcpin' argument not specified with *retention_cell* constraint when LPPLIB10 is enabled

Rule	Reports
SGDC_lowpower103	Check for missing '-value' and '-default' field in 'domain_outputs' constraint.
SGDC_lowpower104	Check for '-vddcpin' argument of 'always_on_buffer' constraint.
SGDC_lowpower105	Sanity check to ensure that 'isolation_cell' constraint is given properly
SGDC_lowpower107	Existence check of pins specified with '-inTerm', '-outTerm' and '-enableTerm' arguments of the `levelshifter` constraint
SGDC_lowpower108	Existence check of pins specified with '-input_pin', '-output_pin' and '-enable_pin' arguments of 'isolation_cell' constraint.
SGDC_lowpower109	Check to ensure same number of arguments specified by '-signame' and '-value' fields of 'power_down' constraint.
SGDC_lowpower110	Check to ensure retention_instance lies in the hierarchy of power domain.
SGDC_lowpower112	Checks the -vddcpin and -vddpin arguments of retention_cell
SGDC_lowpower113	Missing inisosig argument of voltage_domain
SGDC_lowpower114	Power or ground bias nets of voltage_domain not specified as the power or ground supply
SGDC_lowpower115	If pins specified by using the arguments of power_switch are of improper types
SGDC_lowpower116	Check for '-name' and '-value' options should be specified in SGDC command 'voltage_domain'.
SGDC_lowpower118	Performs the sanity check in the associate_lib constraint
SGDC_lowpower119	Performs the sanity checking for the disallow_upf_command SGDC constraint
SGDC_lowpower_RuleReq	Reports those SpyGlass Design Constraints missing from the user-supplied SpyGlass Design Constraints file that are required for the selected rules to run

Rule	Reports
SGDC_powerdomainoutputs01	Reports non-existent signal names specified through domain_outputs
SGDC_powerdomainoutputs02	Reports incorrect value specification in domain_outputs

LP_CHECK_CONSTR

Validates information specified for level shifter, isolation and retention cells with respect to information specified in library using attributes

When to Use

This is a setup rule and always runs by default.

Description

The *LP_CHECK_CONSTR* rule ensures that the information specified for level shifter, isolation, and retention cells do not contradict with information available in the library file. The level shifter, isolation, and retention cells are specified using the *levelshifter*, *isolation_cell*, and *retention_cell* constraints.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *levelshifter* (Optional): Use to specify the names of design units to be used as level shifters.
- *isolation_cell* (Optional): Use to specify the isolation cells in power domains.
- *retention_cell* (Optional): Use to specify the retention latch cells.

Messages and Suggested Fix

The following message appears when the specified <*pin-name*> using constraint <*constr*> contradicts with <*pin-name-list*> in the specified library files:

[FATAL] Pin '<pin-name>' specified using '<arg-name>' field of constraint '<constr>' contradicts with pin(s) '<pin-name-list>'

```
specified in library files/RTL design
```

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The SGDC constraint reported is highlighted in the Atrenta Console GUI.

To resolve the violation message, update the argument for the constraint stated in the violation message.

Example Code and/or Schematic

Example 1

This example illustrates the violation report for the *levelshifter* constraint. The library file is as follows:

```
cell (BASIC_LS) {
      is_level_shifter : true;
      pin (A) { direction : input; }
      pin (Y) { direction : output; function : "A"; }
   }
```

The SGDC file is as follows:

```
levelshifter -name BASIC_LS -from V2 -to V1 -inTerm Y -
outTerm A
```

Therefore, the *LP_CHECK_CONSTR* rule reports the following violation:

```
Pin 'A' specified using '-outTerm' field of constraint
'levelshifter' contradicts with pin(s) 'Y' specified in library
files/RTL design
```

Example 2

This example illustrates the violation report for the *isolation_cell* constraint. The library file is as follows:

```
cell (ISO_LATCH_0) {
      is_isolation_cell : true;
      pin (D) { direction : input; }
```

```
      pin (EN) { direction : input; }
      latch (IQ,IQN) { enable : "EN"; data_in : "D"; }
      pin (Q) { direction : output; function : "IQ"; }
   }
```

The SGDC file is as follows:

```
isolation_cell -names  ISO_LATCH_0 -input_pin EN -output_pin
D -enable_pin EN1  -iso_enable_val 1
```

Therefore, the *LP_CHECK_CONSTR* rule reports the following violation:

```
Pin 'D' specified using '-output_pin' field of constraint
'isocell' contradicts with pin(s) 'Q' specified in library
files/RTL design
```

Example 3

This example illustrates the violation report for the *retention_cell* constraint. The library file is as follows:

```
cell (RET_DFF_QN) {
      retention_cell : "true";
      pin (SAVE) { direction : input; }
      pin (RESTORE) { direction : input; }
      pin (D) { direction : input; }
      pin (CK) { direction : input; }
      ff (IQ,IQN) { clocked_on : "CK"; next_state : "D"; }
      pin (X) { direction : output; function : "IQ"; }
      pin (XN) { direction : output; function : "IQN"; }
   }
```

The SGDC file is as follows:

```
retention_cell -name RET_DFF_QN -domains V3 -clk CK -qTerm  D
```

Therefore, the *LP_CHECK_CONSTR* rule reports the following violation:

```
Pin 'D' specified using '-qTerm' field of constraint
'retention_cell' contradicts with pin(s) '{XN,X}' specified in
library files/RTL design
```

Default Severity Label

Fatal

Rule Group

Constraints Checking

Reports and Related Files

- *lp_lib_data*
- *lp_constr_info*

LP_SGDC_CHECKS

Checks the existence of cells specified with cell_hookup and aon_buffered_signals

When to Use

This rule applicable to all design phases.

Description

The *LP_SGDC_CHECKS* rule reports the following:

- Non-existent pins specified with the `-names` argument of the *cell_hookup* constraint
- Non-existent pins specified with the `–terminatingcells` argument of the *aon_buffered_signals* constraint

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *cell_hookup* (Mandatory): Use to specify special cell hookups. The pins are specified using the `–names` argument.
- *aon_buffered_signals* (Mandatory): Use to specify signals that should be driven by an always-on buffer. Use the `–terminatingcells` argument to specify the cell-pin pair names list of terminating cells and their terminating pins.

Messages and Suggested Fix

Message 1

The following message appears when pin *<pin-name>* is not found on design unit(s) *<du-name-pattern>* when both are specified as a pair with *<argument>* argument of constraint *<constr>*:

[FATAL] Pin '<pin-name>' not found in at least one of the

`modules with name matching '<du-name-pattern>' specified with field '<argument>' of constraint '<constr>' in the design`

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when no pin is specified with design unit(s) *<du-name-pattern>* for *<argument>* argument of constraint *<constr>*:

[FATAL] `No pin associated with module '<du-name-pattern>' specified with field '<argument>' of constraint '<constr>' in the design`

Where, *<constr>* and *<argument>* can be:

- *aon_buffered_signals* and `terminatingcells`
- *cell_hookup* and `names`

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The consequences of not fixing the violations are as follows:

Message 1: The pin mentioned in the constraint is not present on the modules specified in the design. This makes the constraint redundant.

Message 2: No pin is specified in the constraint. This makes the constraint redundant and unusable.

How to Debug and Fix

The SGDC commands reported in the violation messages are highlighted in the Atrenta Console GUI.

For the *aon_buffered_signals* commands, ensure the cell-pin pair specified through the `terminatingcells` argument exists.

For the *cell_hookup* commands, ensure the pins specified through the `name` argument exists.

Example Code and/or Schematic

Example 1

In the following code, the pin `pin1` is not present in the module `AN2`, which is specified in the *aon_buffered_signals* constraint. This makes the constraint redundant.

```
aon_buffered_signals -names out -terminatingcells AN2 pin1
```

Example 2

In the following code, no pin is specified in the *cell_hookup* constraint. This makes the constraint redundant and unusable.

```
cell_hookup -signame out -names "AN2"
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

LpParamSanityCheck

Performs syntactic checks on parameters

When to Use

This is a setup rule and always runs by default.

Description

The *LpParamSanityCheck* rule checks validity of parameter values that you have specified.

Languages

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

Message 1

The following message appears when the parameter <*param*> is specified without any value:

[FATAL] `Null value is passed to the parameter '<param>'`

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the file <file-name> specified with the parameter <*param*> is not found:

[FATAL] `File '<file-name>' specified through parameter '<param>' not found`

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when the file <*file-name*> specified

with the parameter <*param*> could not be opened:

[FATAL] File '<file-name>' specified through parameter '<param>' cannot be opened for reading

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The parameter reported is highlighted in the Atrenta Console GUI.

To resolve these violation messages, check the value of the parameter. In addition, ensure that the file passed through the parameter has appropriate access permissions.

Example and /or Schematic

Suppose you have set the lp_check_enable pin as shown. The *LpParamSanityCheck* rule reports a violation because the parameter is not set with any value.

```
set_parameter lp_check_enable_pin
```

To resolve the violation, set the value to a valid parameter value. In this case, by default, you can set it to 0 or 1, as shown:

```
set_parameter lp_check_enable_pin 1
```

Default Severity Label

Fatal

Rule Group

Constraints Checking

Reports and Related Files

None

SGDC_lowpower05

Reports the voltage_domain constraints specified without the required arguments

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower05 rule* reports the *voltage_domain* constraints specified without the required arguments.

You can use various arguments in this rule:

- Use the `-modname` argument only when specifying the voltage domain of the top-level design unit.
- Use the `-instname` argument while specifying the voltage domains of the hierarchical instances in the top-level design unit.
- Use the `-portname` argument with the `-external` argument for specifying a port's voltage domain that is external to the design.
- Use the `-external` argument only for specifying the external voltage domains.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

voltage_domain (Mandatory): Use this constraint to specify the voltage or power domains in the design.

Messages and Suggested Fix

The following message appears for the *voltage_domain* constraint that is specified without the required arguments:

[FATAL] Either 'modname', 'instname' or 'external' argument should be specified for voltage_domain

Potential Issues

This violation appears when:

- the *voltage_domain* constraint is specified without the -modname or the -instname arguments.
- the *voltage_domain* constraint is specified with the -portname argument without the -external field.

Consequences of Not Fixing

If you do not fix this violation, the *voltage_domain* constraint is ignored.

How to Debug and Fix

To fix this violation, ensure that:

- at least one design element is specified by using either the -modname, -instname, or –portname argument.
- the -external argument is specified with the –portname argument.

Example Code and/or Schematic

Example 1

For the following code, the *SGDC_lowpower05* rule reports a violation because the -modname, -instname, and -portname arguments are missing.

```
voltage_domain -name V1 -value 1.0
```

Example 2

For the following code, the *SGDC_lowpower05* rule reports a violation because the –portname argument is specified without the -external argument.

```
voltage_domain -name V1 -value 1.0 –portname In1
```

Default Severity Label

Fatal

Rule Group

VoltageConstraintCheck

Reports and Related Files

None

SGDC_lowpower06

Reports instances that are not separated by space

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower06* rule reports the following cases of multiple instance names specified with the `-instname` argument:

- Instance names have not been specified with at least one space character between instance names.
- At least one hierarchical instance name contains special or escaped characters but has not been specified enclosed in double quotes.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain*: Use to specify the voltage/power domains in the design.

Messages and Suggested Fix

The following message appears for a *voltage_domain* constraint where instance names are incorrectly specified with the `instname` argument:

[FATAL] `Instance names should be space-separated`

Potential Issues

This violation appears explicitly states the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix this violation, review the `instname` argument and ensure that the instance names are separated by white spaces. In addition, ensure that all instance names that have escaped characters are enclosed in double quotes.

Example Code and/or Schematic

For the following snippet, the SGDC_lowpower06 rule reports a violation because the instance name specified contains an escaped character.

```
current_design top
...
voltage_domain -name PD1 -value 1.2 0
-instname top.\lower2
-isosig top.iso_sig -isoval 1
```

To resolve the violation, make sure the instance name is enclosed in double quotes, as shown in the following snippet.

```
current_design top
...
voltage_domain -name PD1 -value 1.2 0
-instname "top.\lower2 "
-isosig top.iso_sig -isoval 1
```

Default Severity Label

Fatal

Rule Group

Constraint Checking Rules

Reports and Related Files

None

SGDC_lowpower07

Reports inconsistent specification of isoval or inisoval in voltage_domain

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower07* rule reports the *voltage_domain* constraints with incorrect or incomplete isolation information.

The *SGDC_lowpower07* rule requires the following:

- If the `-isosig` argument is specified, the `-isoval` argument must also be specified. In addition, the number of values specified in the `-isoval` argument must match the number of signals specified in the `-isosig` argument.
- If the `-inisosig` argument is specified, the `-inisoval` argument must also be specified. In addition, the number of values specified in the `-inisoval` argument must match the number of input isolation signals specified in the `-inisosig` argument.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

voltage_domain (Mandatory): Use this constraint to specify the voltage or power domains in the design.

Messages and Suggested Fix

Message 1

The following message appears for the *voltage_domain* constraint when the

-isosig argument is specified, but the -isoval argument is not specified:

[FATAL] Field 'isoval' not defined even though 'isosig' defined

Potential Issues

This violation appears if the -isosig argument is specified in the *voltage_domain* constraint, but the -isoval argument is not specified.

Consequences of Not Fixing

If you do not fix this violation, the *voltage_domain* constraint is ignored.

How to Debug and Fix

To fix this violation, review the *voltage_domain* constraint and ensure that the isoval and isosig arguments are specified.

Message 2

The following message appears for the *voltage_domain* constraint where the -inisosig argument is specified, but the -inisoval argument is not specified:

[FATAL] Field 'inisoval' not defined even though 'inisosig' defined

Potential Issues

This violation appears if the -inisosig argument is specified in the *voltage_domain* constraint, but the -inisoval argument is not specified.

Consequences of Not Fixing

If you do not fix this violation, the *voltage_domain* constraint is ignored.

How to Debug and Fix

To fix this violation, review the *voltage_domain* constraint and ensure that the inisoval and inisosig arguments are specified.

Message 3

The following message appears for the *voltage_domain* constraint where the number of signals specified in the -isosig or -inisosig argument is not equal to the number of values specified in the corresponding -isoval or -inisoval argument:

[FATAL] Mismatch in the number of values provided for <*sig-*

type> signals

Where,

<sig-type> is isolation or input isolation.

Potential Issues

This violation appears if the number of values specified in the –isoval or -inisoval argument does not match the number of input isolation signals specified in the -isosig or -inisosig argument of the *voltage_domain* constraint.

Consequences of Not Fixing

If you do not fix this violation, the *voltage_domain* constraint is ignored.

How to Debug and Fix

To fix this violation, ensure that there is no mismatch in the number of values in the–isoval or -inisoval argument and the number of signals in the -isosig or –inisosig argument.

Example Code and/or Schematic

Example 1

For the following code, the *SGDC_lowpower07* rule reports a violation because the -isoval argument is not provided with the -isosig argument.

```
voltage_domain -name V3  -value 1.8 0 -instname inst1 -isosig
sig1
```

Example 2

For the following code, the *SGDC_lowpower07* rule reports a violation because the -inisoval argument is not provided with the -inisosig argument.

```
voltage_domain -name V3  -value 1.8 0 -instname inst1 -
inisosig sig1
```

Example 3

For the following code, the *SGDC_lowpower07* rule reports a violation because the number of isolation signals in the –isosig argument (only one signal, sig1) is not equal to the number of values in the –isoval

argument (three values, 0 1 1).

```
voltage_domain -name V3  -value 1.8 0 -instname inst1 -isosig
sig1 -isoval 0 1 1
```

Example 4

For the following code, the *SGDC_lowpower07* rule reports a violation because the number of the input isolation signals in the `-inisosig` argument (only one signal, `sig1`) is not equal to the number of values in the `-inisoval` argument (three values, `0 1 1`).

```
voltage_domain -name V3  -value 1.8 0 -instname inst1 -
inisosig sig1 -inisoval 0 1 1
```

Default Severity Label

Fatal

Rule Group

Constraints Checking

Reports and Related Files

None

SGDC_lowpower09

Reports when same instance is used in more than one voltage_domain constraint

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower09* rule reports a violation when the instance name specified in the `instname` argument is not unique for all the *voltage_domain* constraints defined under the same environment, that is, the design unit specified as the `current_design`.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

voltage_domain (Mandatory): Use this constraint to specify the voltage or power domains in the design.

Messages and Suggested Fix

The following message appears when the instance <*instname*>, specified in the `-instname` argument, is already specified in another *voltage_domain* constraint defined under the same environment:

[FATAL] `Instance '<inst-name>' has been specified more than once through the voltage_domain constraint(s) in the same top design`

Potential Issues

This violation appears if the same instance is specified in more than one *voltage_domain* constraint.

Consequences of Not Fixing

If you do not fix this violation, the *voltage_domain* constraint is ignored.

How to Debug and Fix

To fix this violation, ensure that the instance name specified in the *voltage_domain* constraint of a particular design unit is not repeated in another *voltage_domain* constraint of the same design unit.

Example Code and/or Schematic

For the following code, the *SGDC_lowpower09* rule reports a violation because the single design element `inst1` is specified in two different voltage domains `V1` and `V2`.

```
voltagedomain -name V1 -value 1.9 -instname inst1
voltagedomain -name V2 -value 1.1 -instname inst1
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_lowpower12

Reports redundant arguments in the voltage_domain constraints

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower12* rule reports a violation when the following arguments are specified in the *voltage_domain* constraint while defining a voltage domain, as these arguments are specific only to the power domains:

`-isosig`

`-isoval`

`-inisosig`

`-inisoval`

`-generate_iso_logic`

`-outputs`

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

voltage_domain (Mandatory): Use this constraint to specify the voltage or power domains in the design.

Messages and Suggested Fix

The following message appears when redundant arguments are specified in the *voltage_domain* constraint:

[FATAL] Redundant field(s) 'generate_iso_logic/isosig/isoval/inisosig/inisoval/outputs' defined for the Voltage Domain'<pin-type>'

Potential Issues

This violation appears when the `-isosig`, `-isoval`, `-inisosig`, `-inisoval`, `-generate_iso_logic`, and `-outputs` arguments are specified in the *voltage_domain* constraint.

Consequences of Not Fixing

If you do not fix this violation, the *voltage_domain* constraint is ignored.

How to Debug and Fix

To fix this violation, ensure that the `-isosig`, `-isoval`, `-inisosig`, `-inisoval`, `-generate_iso_logic`, and `-outputs` arguments are specified only for power domains and not for voltage domains.

Example Code and/or Schematic

Example 1

For the following code, the *SGDC_lowpower12* rule reports a violation because the `-isosig` and `-isoval` arguments are specified for the always-on domain `V1`.

```
voltage_domain -name V1 -value 1.2 -instname U1.T1.inst2 -isosig U1.T1.isosig1 -isoval 1 -generate_iso_logic -outputs U1.T1.out1
```

Example 2

For the following code, the *SGDC_lowpower12* rule reports a violation because the `-inisosig`, `-inisoval`, `-generate_iso_logic`, and `-outputs` arguments are specified for the always-on domain `V1`.

```
voltage_domain -name V1 -value 1.2 -instname U1.T1.inst2 -inisosig U1.T1.isosig -inisoval 1 -generate_iso_logic -outputs U1.T1.out1
```

Default Severity Label

Fatal

Rule Group

Constraints Checking

Reports and Related Files

None

SGDC_lowpower15

Reports power domains with at least one of the three arguments, isosig, inisosig, and noisosig, missing

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower15* rule reports a violation when the *voltage_domain* constraints are not specified with at least one of the `isosig, inisosig, and noisosig` arguments while defining a power domain.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

voltage_domain (Mandatory): Use this constraint to specify the voltage or power domains in the design.

Messages and Suggested Fix

The following message appears for the *voltage_domain* constraint defining a power domain when at least one of the `–isosig, -inisosig, and -noisosig` arguments is not specified:

[FATAL] `For a Power Domain, '-isosig', '-inisosig', or 'noisosig' field must be defined`

Potential Issues

This violation appears when at least one of the `–isosig, -inisosig, and -noisosig` arguments is not specified for a power domain

Consequences of Not Fixing

If you do not fix this violation, the *voltage_domain* constraint is ignored. The

isolation signals are used to control the power-down and power-up conditions of a power domain. Therefore, it is mandatory to specify an isolation signal for a power domain.

How to Debug and Fix

To fix this violation, ensure that at least one of the `–isosig, -inisosig, -noisosig` arguments is specified for a power domain.

Example Code and/or Schematic

For the following code, the *SGDC_lowpower15* rule reports a violation because the isolation signals are not specified for the power domain V1.

```
voltage_domain -name V1 -value 1.2 0 -instname inst1 inst2 inst3
```

The above *voltage_domain* constraint is actually specifying a power domain, and not a voltage domain. The presence of at least one of the three arguments is required.

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_lowpower17

Reports if the outputs argument is not defined with the generate_iso_logic argument for power domains

When to Use

This is a setup rule and runs by default.

Description

The *SGDC_lowpower17* rule reports a violation when the `generate_iso_logic` argument is specified, but the `outputs` argument is not specified for *voltage_domain* constraints that define power domains.

Placeholder steady-state condition names must be specified with the `outputs` argument to define the condition of output values when using the `generate_iso_logic` argument to automatically generate isolation logic.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage/ power domains in the design.

Messages and Suggested Fix

The following message appears for a *voltage_domain* constraint defining a power domain where the `generate_iso_logic` argument is specified, but the `outputs` argument is not specified:

```
[FATAL] '-outputs' field is not defined even when
generate_iso_logic is defined for the Power Domain
```

Potential Issues

The violation message appears if `output` argument is not defined when

`generate_iso_logic` is defined for the power domain.

Consequences of Not Fixing

If you do not fix this violation, the behavior becomes unclear and the constraint becomes unusable because the specification of the `outputs` argument along with the `generate_iso_logic` argument is mandatory for the *voltage_domain* constraint.

How to Debug and Fix

Ensure that the `outputs` argument is also specified if the `generate_iso_logic` argument is specified.

Example Code and/or Schematic

For the following snippet, the *SGDC_lowpower17* rule reports a violation because the output argument has not been specified.

```
[FATAL] Voltage_domain -name V1  -value 1.2 0 -instname
T1.inst4  -isosig T1.isosig -isoval 1 -generate_iso_logic
```

Default Severity Label

Fatal

Rule Group

VoltageConstraintCheck

Reports and Related Files

None

SGDC_lowpower18

Reports when the outputs argument is defined with a value that does not match any of the defined domain_outputs names

When to Use

This is a setup rule and runs by default.

Description

The *SGDC_lowpower18* rule reports when the *voltage_domain* constraints defines power domains and the placeholder steady-state condition name specified with the `outputs` argument is not defined with the `name` argument of a *domain_outputs* constraint under the same environment. The same environment implies that the design unit specified as the `current_design` when using the `generate_iso_logic` argument automatically generates isolation logic for the power domain.

Placeholder steady-state condition names specified with the `outputs` argument must be created using the *domain_outputs* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage/ power domains in the design.
- *domain_outputs* (Mandatory): Use this constraint to specify the values of various signals under the steady-state condition.

Messages and Suggested Fix

The following message appears when the `outputs` arguments is not defined with the `name` argument of a *domain_outputs* constraint in the design unit specified as the `current_design`:

[FATAL] '-outputs' field is defined with a value that does not match any of the defined domain_outputs constraint

Potential Issues

The violation message appears if the `output` argument is defined with a value that is not defined in the *domain_outputs* constraint.

Consequences of Not Fixing

If you do not fix this violation, the constraint becomes unusable as the intended behavior is unclear.

How to Debug and Fix

To resolve this violation, ensure that the `outputs` argument is defined with a value that is specified by the *domain_outputs* constraint.

Example Code and/or Schematic

Example 1

For the following snippet, the opname in the `outputs` argument is not specified by the *domain_outputs* constraint. Therefore, the *SGDC_lowpower18* rule reports a violation message.

```
voltage_domain -name V2  -value 1.2 0 -instname mid.t1.inst4
-isosig mid.sig1 -isoval 1 -outputs opname
```

Here,

Example 2

For the following snippet, the opname in the outputs argument is specified by the *domain_outputs* constraint. Therefore, the *SGDC_lowpower18* rule does not report a violation message.

```
voltage_domain -name V3  -value 1.6 0 -instname mid.t1  -
isosig mid.sig1 -isoval 1 -outputs opname domain_outputs -
name opname -value mid.c1 1  mid.c2 0
```

Default Severity Label

Fatal

Rule Group

VoltageConstraintCheck

Reports and Related Files

None

SGDC_lowpower19

Reports supply names not defined through the supply constraint

When to Use

This is a setup rule and runs by default.

Description

The *SGDC_lowpower19* rule reports where the supply signal name specified with the `supplyname` argument is not defined with the `name` argument of a *supply* constraint in power domains defined by *voltage_domain* constraints, under the design unit specified as the `current_design`.

The voltage and ground rails specified with the `supplyname` argument of the *voltage_domain* constraint should have values defined using the *supply* constraint. If the value of a supply rail is not specified using the *supply* constraint, its value is assumed to be the same as the voltage value of its voltage domain.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify the voltage/power domains in the design.
- *supply* (Mandatory): Use to specify the supply and ground port names.

Messages and Suggested Fix

The following message appears when the supply name <supply-name> specified with the `supplyname` argument is not defined with the `name` argument of the supply constraint under the design unit specified as the `current_design`:

[FATAL] Supply name, <supply-name>, is not defined through any supply constraint

Potential Issues

The violation message appears if you do not define the supply name specified in the `supplyname` argument of the *voltage_domain* constraint by using the *supply* constraint.

Consequences of Not Fixing

If you do not fix the violation, the *supply* constraint assumes the value of the voltage domain for which `supplyname` was used.

How to Debug and Fix

To resolve this violation, specify a valid name in the `supplyname` argument of the *voltage_domain* constraint that is defined as a supply using the `names` argument of the *supply* constraint in the SGDC file.

Example Code and/or Schematic

Consider the following SGDC file:

```
current_design top1

voltagedomain -name V -value 7.3   -modname top1

voltagedomain -name V1 -value 1.20   -instname top1.O12  -
supplyname VSS1 VSS2

voltagedomain -name V2 -value 1.49  -instname top1.A21 -
supplyname VSS2 VSS3

voltagedomain -name V3 -value 1.10 -instname top1.A11 -
supplyname VSS3 VSS4

voltagedomain -name V4 -value 1.12 0 -instname top1.A1111  -
isosig top1.in1[3:0] top1.in2 -isoval 0  1 -supplyname VSS
VSS2

supply -name VSS -value 1

supply -name VSS3 -value 2

supply -name VDD -value 5
```

Here, VSS1, VSS2 and VSS4 are not defined with the *supply* constraint. Therefore, the following violation messages appear:

```
[FATAL] Supply name, VSS1, is not defined through any supply
constraint

[FATAL] Supply name, VSS2, is not defined through any supply
constraint

[FATAL] Supply name, VSS4, is not defined through any supply
constraint
```

Default Severity Label

Fatal

Rule Group

VoltageConstraintCheck

Reports and Related Files

None

SGDC_lowpower23

Reports incomplete -value argument specification in domain_outputs

When to Use

This is a setup rule and runs by default.

Description

The *SGDC_lowpower23* rule reports incorrect signal values specified with the `value` argument of the *domain_outputs* constraint.

The *SGDC_lowpower23* rule requires that each signal name should be followed by a value in the `value` argument of the *domain_outputs* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *domain_outputs* (Mandatory): Use to specify the values of various signals under the steady-state condition.

Messages and Suggested Fix

The following message appears for a *domain_outputs* constraint where a value does not follow signal <signal-name> in the `value` argument:

[FATAL] `signal '<signal-name>' used in 'domain_outputs' constraint is defined without a value`

Potential Issues

The violation message appears if you do not specify a value for signal in the `value` argument of the *domain_outputs* constraint.

Consequences of Not Fixing

If you do not fix this violation, the arguments of the *domain_outputs* constraint should be in the form of <signal,value> tuple.

If you do not specify the value for any signal, the value of that signal in steady-state condition is not known and therefore the *LPSVM23* rule may produce wrong results.

How to Debug and Fix

To fix this violation, specify a value for every signal mentioned in the *domain_outputs* constraint in the SGDC file.

Example Code and/or Schematic

Consider the following SGDC file:

```
current_design mid

voltagedomain  -name V1 -value 1.2 -modname mid

voltagedomain -name V2  -value 1.2 0 -instname mid.t1.inst4
-isosig mid.t1.isosig -isoval 1 -outputs opname

powerdomainoutputs -name opname -value mid.c2[0] 1  mid.c2[1]
1 mid.c2[2] 1  mid.c2[3]
```

The *SGDC_lowpower23* rule reports the following violation message:

```
[FATAL] signal 'mid.c2[3]' used in 'domain_outputs' constraint
is defined without a value
```

Default Severity Label

Fatal

Rule Group

VoltageConstraintCheck

Reports and Related Files

None

SGDC_lowpower24

Reports incorrect voltage domains specified with to/from arguments of the levelshifter constraint

When to Use

This is a setup rule and runs by default.

Description

The *SGDC_lowpower24* rule requires that each voltage domain name specified with the `to` argument or `from` argument of a *levelshifter* constraint must be defined with the *voltage_domain* constraint.

Both `from` and `to` arguments of the *levelshifter* constraint must be defined with a valid voltage domain or power domain. For a level shifter that shifts multiple domains, each pair of `from` domain and `to` domain with the level shifter name must be specified as a separate *levelshifter* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *levelshifter* (Mandatory): Use this constraint to specify the names of design units to be used as level shifters.
- *voltage_domain* (Mandatory): Use to specify the voltage/power domains in the design.

Messages and Suggested Fix

Message 1

The following message appears for a *levelshifter* constraint where a voltage domain name <vd-name> specified with the `from` argument has not been defined with a *voltage_domain* constraint:

[FATAL] From domain '<vd-name>' used for the levelshifter is not defined as a voltage domain in the SGDC file

Potential Issues

The violation message appears if the voltage domain for the *levelshifter* constraint specified with the `from` argument is not defined with a *voltage_domain* constraint.

Consequences of Not Fixing

If you do not fix this violation, it implies that we have defined the level shifter whose source domain actually does not exist. This is wrong. The rules using this information may produce wrong results.

How to Debug and Fix

To fix this violation, ensure that the domain specified in the `from` argument of the *levelshifter* constraint is a valid domain defined by the *voltage_domain* constraint in the SGDC file.

Message 2

The following message appears for a *levelshifter* constraint where a voltage domain name <vd-name> specified with the `to` argument has not been defined with a *voltage_domain* constraint:

[FATAL] To domain '<vd-name>' used for the levelshifter is not defined as a voltage domain in the SGDC file

Potential Issues

The violation message appears if a domain used for the *levelshifter* constraint specified with `to` argument is not defined as a voltage domain in the SGDC file.

Consequences of Not Fixing

If you do not fix this violation, it implies that you have defined the level shifter that has a non-existent destination domain. This is wrong. The rules using this information may produce wrong results.

How to Debug and Fix

To fix this violation, ensure that the domain specified in `to` argument of the *levelshifter* constraint is a valid domain defined by the *voltage_domain* constraint in the SGDC file.

Example Code and/or Schematic

Consider the following SGDC file:

```
current_design top1

voltagedomain -name V -value 7.3 0  -modname top1    -isosig in1  -isoval 0

voltagedomain -name V1 -value 1.20   -instname top1.O12

voltagedomain -name V2 -value 1.49  -instname top1.A21

voltagedomain -name V3 -value 1.10 -instname top1.A11

voltagedomain -name V4 -value 1.12 0 -instname top1.A1111 -isosig top1.in1[3:0] top1.in2 -isoval 0  1  -supplyname VSS VSS2

levelshifter -name myor -from V1 -to V5  -inTerm in1 -enableTerm in2

levelshifter -name myand -from V3 -to V4 -inTerm in1 -enableTerm in2

levelshifter -name myor -from V6 -to V -inTerm in1 -enableTerm in2

levelshifter -name myor -from V4 -to V -inTerm in1 -enableTerm in2

supply -name VSS -value 1

supply -name VSS2 -value 3
```

The following violation messages appear:

[FATAL] To domain 'V5' used for the levelshifter is not defined as a voltage domain in the SGDC file

[FATAL] From domain 'V6' used for the levelshifter is not defined as a voltage domain in the SGDC file

Default Severity Label

Fatal

Rule Group

VoltageConstraintCheck

Reports and Related Files

None

SGDC_lowpower30

Reports signal names repeated in the signal list of value argument of domain_outputs

When to Use

This is a setup rule and runs by default.

Description

The *SGDC_lowpower30* reports when two signals are defined with the same name in the *domain_outputs* constraint. The *SGDC_lowpower30* rule reports signal names that are repeated in the signal list of the `value` argument of a *domain_outputs* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *domain_outputs* (Mandatory): Use this constraint to specify the values of various signals under the steady-state condition.

Messages and Suggested Fix

The following message appears for *domain_outputs* constraint when the signal name <sig-name> is repeated in the signal list of the `value` argument:

[FATAL] `signal, <sig-name>, used more than once in constraint 'domain_outputs`

Potential Issues

The violation message appears if a signal is repeated in the signal list of the `value` argument.

Consequences of Not Fixing

If you do not fix this is violation, the value retrieved for the signal remains undecided.

How to Debug and Fix

To fix this violation, ensure that each signal is specified only once in the *domain_outputs* constraint in the SGDC file.

Example Code and/or Schematic

Consider the following SGDC file:

```
current_design mid

voltagedomain  -name V1 -value 1.2 -modname mid

voltagedomain -name V2  -value 1.2 0 -instname mid.t1.inst4
-isosig mid.t1.isosig -isoval 1 -outputs opname

powerdomainoutputs -name opname -value mid.c2[0] 1  mid.c2[1]
1 mid.c2[3] 1  mid.c2[3] 0
```

It is a setup error as the signal `mid.c2[3]` is used twice in the constraints with different values. A signal should be defined only once.

Default Severity Label

Fatal

Rule Group

VoltageConstraintCheck

Reports and Related Files

None

SGDC_lowpower31

Reports incorrectly defined tops

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower31* rule reports incorrect top specifications.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain*: Use to specify the voltage/power domains in the design.

Messages and Suggested Fix

Message 1

The following message appears when more than one design unit names are specified with the `-modname` argument of a *voltage_domain* constraint or two `-modname` arguments are specified:

```
[FATAL] Only one top module must be specified with -modname
field in 'voltage_domain' constraint
```

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the top-level design unit *<topdu-name>* specified with the `-modname` argument of a *voltage_domain* constraint is not the same as the top-level design unit specified as the environment:

```
[FATAL] Top module '<top-du-name>' specified with -modname
field in voltage_domain constraint does not match the top
module specified as the current design in the SGDC file
```

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when the top-level design unit <*topdu-name*> that is specified as the environment but has not been specified with the -modname argument of any *voltage_domain* constraint:

```
[FATAL] Voltage domain for the top module '<top-du-name>' must
be specified by using the -modname field of the voltage_domain
constraint
```

Potential Issues

This violation appears explicitly states the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix these violations, ensure the following:

- Only on design unit name is specified with the -modname argument of the *voltage_domain* constraint.
- Only one -modname argument is specified for the *voltage_domain* constraint.
- The top-level design unit specified in the -modname argument of a *voltage_domain* constraint is the same as that specified as the environment.
- The top-level design unit specified as the environment is not specified in any *voltage_domain* constraint.

Example Code and/or Schematic

The *SGDC_lowpower31* rule requires that the top-level design unit must be specified as a voltage domain or a power domain. It should be directly specified only once with the modname argument of a *voltage_domain* constraint, as shown in the following snippet:

```
current_design top
  voltage_domain
```

```
-name VD1 -modname top
...
```

Default Severity Label

Fatal

Rule Group

Constraint Checking Rules

Reports and Related Files

None

SGDC_lowpower32

Performs an existence check of the power-out and power-in terminal of power switch

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower32* rule reports non-existent terminals specified with the `-pwroutpin/-pwrinpin` argument of *power_switch* constraints.

This rule requires that the terminal specified with the `-pwroutpin/-pwrinpin` argument of a *power_switch* constraint must exist in the design unit specified with the `-name` argument.

Parameter(s)

None

Constraint(s)

- *power_switch*: (Optional): Use this constraint to specify the power switches in a power domain.

Message Details

Message 1

The following message appears for a *power_switch* constraint when the terminal *<term-name>* specified with the `-pwroutpin` argument does not exist in the design unit *<du-name>* specified with the `-name` argument:

[FATAL] `The power-out terminal '<term-name>' of module '<du-name>' does not exist`

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears for a *power_switch* constraint when the terminal *<term-name>* specified with the `-pwrinpin` argument does

not exist in the design unit <*du-name*> specified with the `-name` argument:

[FATAL] `The power-in terminal '<term-name>' of module '<du-name>' does not exist`

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when the `poweroutpin` argument is not defined for a design unit <*du-name*>:

[FATAL] `-poweroutpin constraints is not defined for power_switch '<du-name>'`

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears when the `pwrinpin` argument is not defined for a design unit <*du-name*>.

[FATAL] `-pwrinpin constraints is not defined for power_switch '<du-name>'`

Potential Issues

This violation appears explicitly states the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix these violations, ensure the following:

- Both `pwroutpin` and `pwrinpin` arguments are provided in the constraint.
- The pins provided with the `pwroutpin` and `pwrinpin` arguments exist in the design.

Example Code and/or Schematic

For the following snippet, the *SGDC_lowpower32* rule reports a violation

message because the AAA pin, which is specified with the `pwroutpin` argument, does not exist in the design.

```
powerswitch -name R120_ASLSOL120X160 -pwroutpin AAA
```

Default Severity Label

Fatal

Rule Group

Constraint Checking Rules

Reports and Related Files

None

SGDC_lowpower40

Performs sanity check for the existence of the domains argument of retention_cell

When to Use

This is a setup rule and runs by default.

Description

The *SGDC_lowpower40* rule flags incorrect *retention_cell* and *retention_instance* constraint specifications when the *LPSVM38* rule is selected to run.

The *SGDC_lowpower40* rule flags the following cases:

- The `domains` argument of *retention_cell* constraint is not specified as a valid power domain (as specified with the `name` argument of the *voltage_domain* constraint).
- Neither the *retention_instance* constraint nor the `domains` argument for any *retention_cell* constraint is specified.

Rule Exception(s)

This rule reports a violation only when *LPSVM38* rule runs.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage/ power domains in the design.
- *retention_cell* (Mandatory): Use this constraint to specify the retention latch cells.
- *retention_instance* (Mandatory): Use this constraint to specify the hierarchy information of retention cell instances.

Messages and Suggested Fix

The following message appears when the power domain *<domain-name>* specified with the domains argument of *retention_cell* constraint of cell *<cell-name>* is not a valid power domain:

[FATAL] Domain '<domain-name>' specified for retention_cell '<cell-name>' should be valid powerdomain.

Since *retention_instance* constraint is not specified, the domains argument of *retention_cell* should be specified.

Potential Issues

The violation message appears if the domain specified in the domains argument of *retention_cell* constraint is not a valid domain.

Consequences of Not Fixing

If you do not fix this violation, it implies that you have defined the retention cell for a non-existent power domain. This value is used by rule *LPSVM38* and may produce wrong results.

How to Debug and Fix

To fix this violation, ensure that the domain specified in the domains argument of the *retention_cell* constraint is a properly defined power domain using the *voltage_domain* constraint.

Example Code and/or Schematic

Consider the following SGDC file:

```
current_design CKT_TOP

voltagedomain -name V1 -value 1.2 -modname CKT_TOP

voltagedomain -name V2 -value 1.3 0 -instname CKT_TOP.U1  -isosig BFC_B -isoval 1

retencell -name FD1S3AQV15 -domains V3
```

The following violation message appears:

[FATAL] Domain 'V3' specified for retention_cell 'FD1S3AQV15' should be valid powerdomain

Default Severity Label

Fatal

Rule Group

VoltageConstraintCheck

Reports and Related Files

None

SGDC_lowpower47

Reports non-existent pins specified with vddpin/vddcpin arguments of retention cell

When to Use

This is a setup rule and runs by default.

Description

The *SGDC_lowpower47* rule reports non-existent pins specified with `vddpin/vddcpin` arguments of the *retention_cell* constraint.

The pins specified with `vddpin/vddcpin` arguments of the *retention_cell* constraint must exist as an input/inout pin of the cells specified with the `name` argument.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *retention_cell* (Mandatory): Use this constraint to specify the retention latch cells.

Messages and Suggested Fix

The following message appears when the pin *<pin-name>* specified with the `vddpin/vddcpin` arguments does not exist as an input/inout pin of the cell *<cell-name>* specified with the `name` argument:

[FATAL] `Pin '<pin-name>' of cell '<cell-name>' specified with '<arg>' argument of 'retention_cell' constraint should be input/inout terminal`

Potential Issues

The violation message appears when a pin specified with the `vddpin/`

vddcpin arguments does not exist as an input/inout pin of the cell specified with the name argument.

Consequences of Not Fixing

If you do not fix this violation, the pins specified do not exist for that cell. These pins are used by *LPPLIB10* and may cause the rule to produce incorrect results.

How to Debug and Fix

To fix this violation, ensure that the pin specified with the vddcpin/ vddpin arguments of the *retention_cell* constraint is a valid pin that exists for the cell and has direction as input/inout.

Example Code and/or Schematic

Consider the following SGDC file:

```
current_design top

voltagedomain -name V1 -value 1.2 -modname top

voltagedomain -name V2 -value 1.3 0 -instname top.mid_inst  -isosig rst -isoval 0

retencell -name FD1 -vddpin CPPI -vddcpin DK
```

The following violation messages appear:

```
Pin 'CPPI' of cell 'FD1' specified with '-vddpin' argument of 'retention_cell' constraint should be in  put/inout terminal

Pin 'DK' of cell 'FD1' specified with '-vddcpin' argument of 'retention_cell' constraint should be inp ut/inout terminal
```

Default Severity Label

Fatal

Rule Group

VoltageConstraintCheck

Reports and Related Files

None

SGDC_lowpower48

Reports incorrect values specified in the seqvalue argument of domain_signal

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower48* rule reports a violation when:

- The width of the sequence specified with the `seqvalue` argument of the *domain_signal* constraint is not equal to the number of signals specified in the `seqsignals` argument.
- The sequence specified in the `seqvalue` argument of the *domain_signal* constraint contains values other than `0` and `1`.

Language

Verilog, VHDL

Constraint(s)

SGDC

- *domain_signal* (Mandatory): Use this constraint to specify the power-up or power-down signals.

Messages and Suggested Fix

Message 1

The following message appears for the *domain_signal* constraint when the width of the sequence specified in the `-seqvalue` argument does not match the number of signals specified in the `-seqsignals` argument:

```
[FATAL] No. of values in patterns provided with field
'seqvalue' does not match with no. of signals provided with
field '-seqsignal'
```

Potential Issues

This violation appears when the number of values in patterns provided in the `-seqvalue` argument for the *domain_signal* constraint does not match

the number of signals specified in the –seqsignals argument.

Consequences of Not Fixing

If you do not fix this violation, the *domain_signal* constraint is ignored.

How to Debug and Fix

To fix this violation, ensure that the width of the sequence specified in the -seqvalue argument is equal to the number of signals specified in the -seqsignals argument.

Message 2

The following message appears for the *domain_signal* constraint when the pattern *<pattern>* of the sequences specified in the -seqvalue argument contains values other than 0 and 1:

```
[FATAL] Pattern '<pattern>' provided with field '-seqvalue' should have only sequence of '0' or '1'
```

Potential Issues

This violation appears for the *domain_signal* constraint when the pattern of the sequences specified in the -seqvalue argument contains values other than 0 and 1.

Consequences of Not Fixing

If you do not fix this violation, the *domain_signal* constraint is ignored.

How to Debug and Fix

To fix this violation, ensure that the sequences specified in the -seqvalue argument contain only 0 and 1.

Example Code and/or Schematic

Example 1

For the following code, the *SGDC_lowpower48* rule reports a violation because the width of the sequence specified in the -seqvalue argument is not equal to the number of signals specified in the -seqsignals argument.

```
domain_signal -name sig1 -value 1 -clocks clk1 -seqsignals s1 s2 -seqvalue 000 10
```

In this case, the width of the sequence, 000, specified in the `-seqvalue` argument, is 3. The signals specified in the `-seqsignals` argument are s1 and s2.

Example 2

For the following code, the *SGDC_lowpower48* rule reports a violation because the pattern, 84, specified in the `-seqvalue` argument, contains values other than 0 and 1.

```
domain_signal -name sig2 -value 0 -clocks clk1 -seqsignals s1 s2 -seqvalue 84
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_lowpower52

Flags if -noisosig have been specified with -isosig or -inisosig field for the power domain

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower52* rule reports the *voltage_domain* constraints defining power domains if both `-isosig` and `-noisosig` arguments are specified for the same power domain. In addition, this rule reports the *voltage_domain* constraints if both `-inisosig` and `-noisosig` arguments are specified for the same power domain.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

Message 1

The following message appears for the *voltage_domain* constraint when both `-isosig` and `-noisosig` arguments are specified:

```
[FATAL] Both '-noisosig' and '-isosig' should not be specified for the same Power Domain
```

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

Both `-isosig` and `-noisosig` arguments have been specified together. This makes the *voltage_domain* constraint useless.

How to Debug and Fix

Specify either `-isosig` or `-noisosig` argument in the *voltage_domain* constraint.

Message 2

The following message appears for the *voltage_domain* constraint when both `-inisosig` and `-noisosig` arguments are specified:

```
[FATAL] Both '-noisosig' and '-inisosig' should not be
specified for the same Power Domain
```

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

Both `-inisosig` and `-noisosig` arguments have been specified together. This makes the *voltage_domain* constraint useless.

How to Debug and Fix

Specify either `-inisosig` or `-noisosig` argument in the *voltage_domain* constraint.

Example Code and/or Schematic

Example 1

Consider the following code:

```
voltagedomain -name mid1 -instname inst1 -value 1.2 0 -
inisosig iso -inisoval 1 –noisosig
```

Here, both `-inisosig` and `-noisosig` arguments are specified together. Only one should be specified.

Example 2

Consider the following code:

```
voltagedomain -name mid2 -instname inst2 -value 1.2 0 -isosig
iso -inisoval 1 –noisosig
```

Here, both `-isosig` and `-noisosig` arguments are specified together. Only one should be specified.

Default Severity Label

Fatal

Rule Group

Constraints Checking

Reports and Related Files

None

SGDC_lowpower59

Reports non-existent terminals specified with enableports of voltage_domain

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower59* rule reports the following:

- Non-existent ports are specified with the `enableports` argument.
- The number of items specified with the `enableports` argument is not even and hence the last port does not have a limit.
- An even-numbered item is not an integer or "`-1`" (indicates unlimited fan-out).
- Wildcard based value specified for the `instname` argument when the *LPSVM45* rule is selected to run.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain*: Use to specify the voltage/power domains in the design.

Messages and Suggested Fix

Message 1

The following message appears for a *voltage_domain* constraint when the port <*port-name*> specified with the `enableports` argument is not a port in the design:

[FATAL] Port '<port-name>' given with the '-enableports' options of the 'voltage_domain' constraint does not exist for instance '<inst-name>' in the design

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears for a *voltage_domain* constraint when the number of items specified with the `enableports` argument is not an even number or an even-numbered entry is not an integer or "`-1`":

```
[FATAL] Each '-enableports' entry must have a corresponding max fan-out integer value specified
```

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears for a *voltage_domain* constraint when wildcard-based value has been specified with the `-instname` argument and the *LPSVM45* rule was selected to run:

```
[FATAL] Wildcard is not supported for '-instname' field, when rule LPSVM45 is enabled
```

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix these violations, ensure the following:

- The ports specified in the `enableports` argument have been defined.
- The number of items specified with the `enableports` argument is even.
- An even-numbered item is an integer and not "`-1`".
- There is no wildcard based value specified for the `instname` argument when the *LPSVM45* rule is selected to run.

Example Code and/or Schematic

For the following snippet, the *SGDC_lowpower59* rule reports a violation because the number of power switch enables that can be connected to the

enable port of the power domain, PD1, is set to "-1". This indicates an unlimited fan-out.

```
voltage_domain -name PD1 -value 1.2 0
-instname "TOP.lower1" -isosig top.iso_sig
-isoval 1 -enableports 'EN12 “-1”'
```

To fix the violation, specify a positive integer, as shown in the following snippet.

```
voltage_domain -name PD1 -value 1.2 0
-instname "TOP.lower1" -isosig top.iso_sig
-isoval 1 -enableports 'EN12 100'
```

Default Severity Label

Fatal

Rule Group

Constraint Checking Rules

Reports and Related Files

None

SGDC_lowpower60

Reports non-existent objects specified with ram_switch

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower60* rule reports the following:

- Non-existent design units specified with the `-switch_name` argument and the `-memory_cellnames` argument
- Enable pins specified with the `switch_enable_pin` argument that do not exist on the design unit specified with the `switch_name` argument
- VSS output pins specified with the `switch_vss_outpin` argument that do not exist on the design unit specified with the `switch_name` argument
- VSS pins specified with the `memory_vss_pin` argument that do not exist on the design unit specified with the `memory_cellnames` argument
- The *ram_switch* constraints without the `switch_vss_outpin` argument and/or the `memory_vss_pin` argument when the *LPPLIB12* rule is selected to run

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *ram_switch*: Use to specify RAM switches.

Messages and Suggested Fix

Message 1

The following message appears for a *ram_switch* constraint when no instance of design unit <*du-name*> specified with the `switch_name` argument is found in the design:

[FATAL] `No instance found for module '<du-name>' specified with field '-switch_name' of constraint 'ram_switch' in the design`

For debugging information, click *How to Debug and Fix* .

Message 2

The following message appears for a *ram_switch* constraint when the Enable pin <*pin-name*> specified with the `switch_enable_pin` argument does not exist on the design unit <*du-name*> specified with the `switch_name` argument:

[FATAL] `Enable pin '<pin-name>' not found for module '<duname>' specified with field '-switch_name' of constraint 'ram_switch' in the design`

For debugging information, click *How to Debug and Fix* .

Message 3

The following message appears for a *ram_switch* constraint when the VSS pin <*pin-name*> specified with the `switch_vss_outpin` argument does not exist on the design unit <*du-name*> specified with the `switch_name` argument:

[FATAL] `Vss pin '<pin-name>' not found for module '<duname>' specified with field '-switch_name' of constraint 'ram_switch' in the design`

For debugging information, click *How to Debug and Fix* .

Message 4

The following message appears for a *ram_switch* constraint when no instance of design unit <*du-name*> specified with the `memory_cellnames` argument is found in the design:

[FATAL] `No instance found for module '<du-name>' specified with field '-memory_cellnames' of constraint 'ram_switch' in the design`

For debugging information, click *How to Debug and Fix* .

Message 5

The following message appears for a *ram_switch* constraint when the VSS pin <*pin-name*> specified with the `memory_vss_pin` argument does not exist on the design unit <*du-name*> specified with the `memory_cellnames` argument:

[FATAL] `Vss pin '<pin-name>' not found for module '<duname>' specified with field '-memory_cellnames' of constraint 'ram_switch' in the design`

For debugging information, click *How to Debug and Fix* .

Message 6

The following message appears for a *ram_switch* constraint without the `switch_vss_outpin` argument when the *LPPLIB12* rule has been selected to run:

[FATAL] `Vss pin for RAM switch should be specified with field '-switch_vss_outpin' of constraint 'ram_switch', when rule LPPLIB12 is run`

For debugging information, click *How to Debug and Fix* .

Message 7

The following message appears for a *ram_switch* constraint without the `memory_vss_pin` argument when the *LPPLIB12* rule has been selected to run:

[FATAL] `Vss pin for memory cells should be specified with field '-memory_vss_pin' of constraint 'ram_switch', when rule LPPLIB12 is run`

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix this violation, review the *ram_switch* constraint specification and ensure the following:

- The design units specified with the `switch_name` and the `memory_cellnames` arguments exist
- The enable pins specified with the `switch_enable_pin` argument exist on the design unit specified with the `switch_name` argument
- The VSS output pins specified with the `switch_vss_outpin` argument exist on the design unit specified with the `switch_name` argument
- The VSS pins specified with the `memory_vss_pin` argument that exist on the design unit specified with the `memory_cellnames` argument
- The *ram_switch* constraints have the `switch_vss_outpin` and/or the `memory_vss_pin` arguments when the *LPPLIB12* rule is selected to run

Example Code and/or Schematic

The *SGDC_lowpower60* rule reports a violation for the following code, as the *ram_switch* constraint is used without the `switch_vss_outpin` argument and/or the `memory_vss_pin` argument when the *LPPLIB12* rule is selected to run:

```
current_design top

voltagedomain -name V1 -value 1.2 -modname top

ramswitch -switch_name mid -switch_enable_pin i -
memory_cellnames lshfter_V1_V2
```

Default Severity Label

Fatal

Rule Group

`Constraint Checking Rules`

Reports and Related Files

None

SGDC_lowpower61

Reports incorrectly defined input_isocell constraint

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower61* rule reports the following cases:

The power domain specified with the `-belongsto` argument has not been specified using the *voltage_domain* constraint.

Some *input_isocell* constraints for a power domain use the `inhibit` argument while some other *input_isocell* constraints for the same power domain do not.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain*: Use to specify the voltage/power domains in the design.
- *input_isocell*: Use to specify the isolation cells at inputs of a power domain.

Messages and Suggested Fix

Message 1

The following message appears for an *input_isocell* constraint specification when the power domain `<dom-name>` specified with the `belongsto` argument has not been specified by using the *voltage_domain* constraint.

[FATAL] `Domain '<dom-name>' given as '-belongsto' in constraint input_isocell is not defined as powerdomain using a voltage_domain constraint in the SGDC file`

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when some *input_isocell* constraints for a power domain use the `inhibit` argument while some other *input_isocell* constraints for the same power domain do not:

[FATAL] `All input_isocell should either be with -inhibit or all should be without -inhibit`

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix these violation, ensure the following:

- The power domain specified with the `-belongsto` argument has been specified by using the *voltage_domain* constraint.
- If the *input_isocell* constraint is specified with the `inhibit` argument for a power domain, all *input_isocell* constraints for that power domain must use the `inhibit` argument.

Example Code and/or Schematic

The *SGDC_lowpower61* rule reports a violation for the following code, as V2, given as '-belongsto' in the *input_isocell* constraint, is not defined as power domain by using the *voltage_domain* constraint in the SGDC file:

```
current_design top

voltagedomain -name V1 -value 1.2 -modname top

voltagedomain -name PD1 -value 1 0 -instname top.mid_inst -isosig iso -isoval 0 -inputs PD_IN

inisocell -names HD65_LH_ISO0X10_VDDC -belongsto V2
```

Default Severity Label

Fatal

Rule Group

Constraint Checking Rules

Reports and Related Files

None

SGDC_lowpower62

Reports incorrectly defined cell_pin_info specification

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower62* rule reports the following cases:

- Pins of multi-supply cells that are not specified with the `pin_name_supply` argument.
- Supply name specified by using the `-pin_name_supply` argument is not defined with the *supply* constraint.
- Pins specified with the `pin_name_supply` argument that do not exist on the cell specified with the `cellname` argument.
- Incomplete pairs specified with the `pin_name_supply` argument.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *cell_pin_info*: Use to specify the pin-to-supply name pairs for multi-supply cells.
- *supply*: Use to specify the supply and ground port names.

Messages and Suggested Fix

Message 1

The following message appears for the *cell_pin_info* constraint specification when the pin *<pin-name>* of the cell *<cellname>* has not been specified with the `pin_name_supply` argument:

[FATAL] `Power supply name for pin '<pin-name>' of cell '<cell-`

name>' is not specified through 'cell_pin_info' constraint

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears for the *cell_pin_info* constraint specification when the pin *<pin-name>* of the cell *<cellname>* has not been specified with the pin_name_supply argument:

[FATAL] Supply '<sup-name>' given with pin '<pin-name>' of cell '<cell-name>' as specified through 'cell_pin_info' constraint is not defined with supply constraint

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears for the *cell_pin_info* constraint specification when the pin *<pin-name>* specified with the pin_name_supply argument is not found on cell *<cellname>*:

[FATAL] Pin '<pin-name>' of cell '<cell-name>' as specified through 'cell_pin_info' constraint not found in the design

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears for the *cell_pin_info* constraint specification when an odd number of names are specified with the pin_name_supply argument for the cell *<cellname>:*

[FATAL] The number of pin names is not equal to the number of supply names for cell '<cell-name>' in 'cell_pin_info' constraint

For debugging information, click *How to Debug and Fix*.

Message 5

The following message appears for the *cell_pin_info* constraint specification when the cell *<cell-name>* specified with the cellname argument:

[FATAL] The cellname '<cell-name>' has been specified more than once through 'cell_pin_info' constraint

Potential Issues

The violation messages explicitly states the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix these violations, ensure the following:

- Specify all pins of multi-supply cells with the `pin_name_supply` argument.
- Specify the supply name, which is defined with the *supply* constraint, using the `pin_name_supply` argument.
- Specify pins with the `pin_name_supply` argument that exist on the cell specified with the `cellname` argument.
- Specify complete pairs with the `pin_name_supply` argument.

Example Code and/or Schematic

Supply `V0` given with the pin `VSS` of the cell `R120_ASLSOL120X160` as specified through the *cell_pin_info* constraint is not defined with the *supply* constraint.

```
current_design top

voltagedomain -name VDD -value 1.2 -modname top  -supplyname
VDD VSS

voltagedomain -name VDDEXT -value 1.5 -instname top.to1
top.to2 -supplyname VDDEXT VSSEXT

levelshifter -name R120_ASLSOL120X160 -from VDD -to VDDEXT -
inTerm A -enableTerm EN

supply -name VDD -value 1.2

supply -name VDDEXT -value 1.5

supply -name VSS -value 0

supply -name VSSEXT -value 0

pgpins_naming  -power "VD*" -ground "VS*"

set_cell_pin_info -cellname R120_ASLSOL120X160
```

```
-pin_name_supply VDD VSS V0 VDDEXT
```

Default Severity Label

Fatal

Rule Group

Constraint Checking Rules

Reports and Related Files

None

SGDC_lowpower65

Reports incorrect supply values

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower65* rule reports when a supply specified with the `supplyname` argument of the *voltage_domain* constraint is specified in the *supply* constraint with a value different from the voltage domain voltage value or in another *voltage_domain* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain*: Use to specify the voltage/power domains in the design.
- *supply:* Use to specify the supply and ground port names.

Messages and Suggested Fix

The following message appears for a *voltage_domain* constraint when the supply *<supply-name>* specified with the `supplyname` argument is specified with a different value *<value>* in a *supply* constraint or another *voltage_domain* constraint:

[WARNING] Supply '<supply-name>' is defined with different value '<value>' in the SGDC file

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

Though the SpyGlass Power Verify solution will continue with its analysis, the results may not be as expected.

How to Debug and Fix

The constraint reported in the violation message is highlighted in the Atrenta Console GUI.

To fix this violation, review the *supply* and *voltage_domain* constraints. Ensure the supply name reported in the violation message has a consistent value.

Example Code and/or Schematic

In the following case, the warning will be received:

```
voltagedomain -name PD -value 0.9 0 -instname
test.power_domain -isosig test.iso test.lshift_pd -isoval 0 1
-supplyname in

supply -name in -value 0.8
```

Default Severity Label

Warning

Rule Group

Constraint Checking Rules

Reports and Related Files

None

SGDC_lowpower66

Reports existence of the en_inv_in, en_inv_out and en_buf_in terminals of power switch

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower66* rule reports the *power_switch* constraints with non-existent pins specified with the `-en_inv_in`, `-en_inv_out`, and `en_buf_in` arguments.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *power_switch*: Use to specify power switches in power domains.

Messages and Suggested Fix

Message 1

The following message appears for a *power_switch* constraint when the pin *<pin-name>* specified with the `en_inv_in` argument does not exist as an input/inout pin of the design unit *<du-name>* specified with the `name` argument:

[FATAL] `The 'en_inv_in' terminal '<pin-name>' of module '<du-name>' should exist as input terminal`

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears for a *power_switch* constraint when the pin *<pin-name>* specified with the `en_inv_out` argument does not exist as an output/inout pin of the design unit *<du-name>* specified with the `name`

argument:

[FATAL] The en_inv_out terminal '<pin-name>' of module '<du-name>' should exist as output terminal

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears for a *power_switch* constraint when the pin *<pin-name>* specified with the en_buf_in argument does not exist as an input/inout pin of the design unit *<du-name>* specified with the name argument:

[FATAL] The en_buf_in terminal '<pin-name>' of module '<du-name>' should exist as input terminal

Potential Issues

The violation messages explicitly states the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix these violations, ensure the following:

- The pin specified with the en_inv_in and en_buf_in arguments exist as an input/inout pin of the powerswitch design unit.
- The pin specified with the en_inv_out argument exist as an output/inout pin of the power switch design unit.

Example Code and/or Schematic

The en_inv_in terminal VSSLOGI of the module LL_SLOGICSWITCH20 should exist as an input terminal.

```
current_design top

voltagedomain -name V1 -value 1.2 -modname top

voltagedomain -name PD1 -value 1 0 -instname top.mid_inst -
isosig top.rst -isoval 0 -enableports PwrIN1 0  PwrIN2 1
PwrIN3 2
```

```
powerswitch -name LL_SLOGICSWITCH20 -en_inv_in VSSLOGI -
pwroutpin PWROFF
```

Default Severity Label

Fatal

Rule Group

Constraint Checking Rules

Reports and Related Files

None

SGDC_lowpower67

Performs sanity checks for the terminals of a level shifter

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower67* rule reports a violation if a level shifter module has the terminal name as specified in the constraint. The name of level shifter module can be specified using the `-name` argument of the *levelshifter* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *levelshifter*: Use to specify the names of design units to be used as level shifters.

Messages and Suggested Fix

Message 1

The following message appears for a *levelshifter* constraint when the pin *<pin-name>* specified with the `inSupplyTerm` argument or the `outSupplyTerm` argument does not exist as an input/inout pin of the design unit *<du-name>* specified with the `name` argument:

[FATAL] `The '<type>' terminal '<pin-name>' of module '<du-name>' should exist as input terminal`

Where, *<type>* can be `inSupplyTerm` or `outSupplyTerm`.

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears for a *levelshifter* constraint when an

argument <argument> has not been specified:

[FATAL] '<argument>' argument is not defined for 'levelshifter' constraint.

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears for a *levelshifter* constraint when the terminal <pin-name> specified with the argument <argument> is not of the correct type:

[FATAL] The terminal '<pin-name>' specified with argument '<argument>' of 'levelshifter' constraint should be of type 'signal'.

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears for a *levelshifter* constraint when the terminal <pin-name> specified with the argument <argument> is not of type power:

[FATAL] The terminal '<pin-name>' specified with argument '<argument>' of 'levelshifter' constraint should be of type 'power'.

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix these violations, ensure the following:

- The in-supply terminal names exist as input or inout terminals of the level shifter defined by using the *levelshifter* constraint
- The out-supply terminal name exist as input/inout terminals of the level shifter
- The in-supply or out-supply terminals is of the type `power`

- The input, output, or enable terminals is of the type `signal`

Example Code and/or Schematic

The terminal `VDD` specified with the argument `-outSupplyTerm` of the *levelshifter* constraint should be of the type `power`.

```
current_design top
voltagedomain -name V2 -value 1.2 -modname top -supplyname VDDTOP VSSTOP
voltagedomain -name V1 -value 1.6 -instname top.m1.BH1_BUF1 -supplyname VDDEXT VSSEXT
levelshifter -name R120_ASLSIL120X160 -from V1 -to V2 -inTerm A -enableTerm EN -outSupplyTerm VDD
supply -name VDD -value 1.6
```

Default Severity Label

Fatal

Rule Group

`Constraint Checking Rules`

Reports and Related Files

None

SGDC_lowpower68

Reports incorrectly defined special_cell

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower68* rule reports the *special_cell* constraints that do not have the mandatory rule-specific arguments.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *special_cell*: Use to specify special cells.
- *supply*: Use to specify the supply and ground port names.

Messages and Suggested Fix

Message 1

The following message appears for a *special_cell* constraint without the `-regions` argument when the *LPSVM37* rule is selected:

```
[FATAL] Field '-regions' should be specified for 'special_cell' constraint, when rule LPSVM37 is run
```

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears for a *special_cell* constraint without the `-supplies` argument when the *LPPLIB13* rule is selected:

```
[FATAL] Field '-supplies' should be specified for 'special_cell' constraint, when rule LPPLIB13 is run
```

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears for a supply name specified by using the `-supplies` argument of the *special_cell* constraint when it is not defined with the `-name` argument of the *supply* constraint

[FATAL] `Supply name given with '-supplies' field of 'special_cell' constraint should be defined with '-name' argument of 'supply' constraint.`

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix these violations, ensure the following:

- Specify the `regions` argument, if the *LPSVM37* rule is selected.
- Specify the `supplies` argument, if the *LPPLIB13* rule is selected.
- Specify the supply name, by using the `supplies` argument, as is defined with the `name` argument of the *supply* constraint.

Example Code and/or Schematic

Message 1 is received in case –regions field is not as in specified in the following snippet:

```
specialcell -name myand -regions mid.t1.inst3 mid.t1.inst4
```

Message 2 is received in case –supplies argument is not as in specified in the following snippet:

```
specialcell -name R120_ASLSIL120X160 -supplies VDDTOP VDDEXT
```

Message 3 is received when the following snippet is defined:

```
specialcell -name R120_ASLSIL120X160 -supplies VDDTOP VDDEXT
```

But at the same time this is not defined:

```
supply –name VDDTOP VDDEXT
```

Default Severity Label

Fatal

Rule Group

Constraint Checking Rules

Reports and Related Files

None

SGDC_lowpower69

Reports constraints specifications with multiple design units defined as current designs

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower69* rule checks if there are multiple current designs specified in constraints file for more than one top design unit.

The SpyGlass Power Verify policy works with only one top. Therefore, specify only one and the same top-level design unit as the `current_design` in all your constraints specification.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears for the first constraints specification with a different `current_design` from the constraints specifications found earlier:

[FATAL] `Only one top design unit should be defined as current design`

Potential Issues

This violation message appears because multiple current designs are specified in the constraints file for more than one top design unit.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix these violations, ensure the following:

- For a design with multiple tops, specify one top.
- Remove the `current_design` from the SGDC file other than the working top design unit.

Example Code and/or Schematic

For the following snippet, the *SGDC_lowpower69* rule reports a violation because there are multiple current designs specified in the constraints. You should specify only one current design unit.

```
current_design sr_flop_gate
current_design counter_gate
```

Default Severity Label

Fatal

Rule Group

`Constraint Checking Rules`

Reports and Related Files

None

SGDC_lowpower71

Reports missing inTerm and outTerm for level shifter

When to Use

This is a setup rule and it runs by default when the *LPSVM30* rule is enabled.

Description

The *SGDC_lowpower71* rule reports the *levelshifter* constraints in the following cases:

- The level shifter design unit has more than one input terminal but the `-inTerm` argument is not specified. In this case, specify the input terminal to be considered for processing using the `-inTerm` argument of the *levelshifter* constraint.
- The level shifter design unit has more than one output terminal but the `-outTerm` argument is not specified. In this case, specify the output terminal to be considered for processing using the `outTerm` argument of the *levelshifter* constraint.

Prerequisites

Enable the *LPSVM30* rule.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *levelshifter* (Optional): Use to specify the names of design units to be used as level shifters.

Messages and Suggested Fix

Message 1

The following message appears for a *levelshifter* constraint without the `inTerm` argument when the level shifter design unit <*du-name*> has

more than one input terminal:

```
[FATAL] Please specify inTerm of Levelshifter '<du-name>' as it could not be resolved from design
```

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears for a *levelshifter* constraint without the `outTerm` argument when the level shifter design unit <*du-name*> has more than one output terminal:

```
[FATAL] Please specify outTerm of Levelshifter '<du-name>' as it could not be resolved from design
```

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix these violations, ensure the following:

- **Message 1:** If there is more than one input pin, all except one pin should be specified as enableTerm otherwise this rule will not be able to resolve the inTerm pin.
- **Message 2:** If the level shifter has multiple input or output pins, specify the input pin though inTerm and the output pin through outTerm.

Example Code and/or Schematic

For the following snippet, the *SGDC_lowpower71* rule reports a violation because the `outTerm` of the level shifter is not specified.

```
levelshifter -name LS -from PD -to VD -inTerm in
```

To resolve the violation, specify the `outTerm` as shown in the following:

```
levelshifter -name LS -from PD -to VD -inTerm in -outTerm out
```

Default Severity Label

Fatal

Rule Group

Constraint Checking Rules

Reports and Related Files

None

SGDC_lowpower72

Reports incorrect arguments specified for pin_voltage

When to Use

This is a setup rule and it runs by default.

Description

The `SGDC_lowpower72` rule reports *pin_voltage* constraints in the following cases:

- Neither the `-module` nor the `-instance` argument is specified.
- Both the `-module` and the `-instance` arguments are specified.
- Neither the `-names` nor the `-default` argument is specified.
- Both the `-names` and the `-default` arguments are specified.
- The voltage domain specified with the `-voltage` argument has not been defined using the *voltage_domain* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *pin_voltage* (Mandatory): Use to specify the voltage/power domain for primary ports, pins of design units or instances in the design.
- *voltage_domain* (Mandatory): Use to specify the voltage/power domains of design units and instances.

Messages and Suggested Fix

Message 1

The following message appears for a *pin_voltage* constraint where either both or none of the `-module` and the `-instance` arguments are specified:

```
[FATAL] Either '-module' or '-instance' argument should be
specified for 'pin_voltage' constraint
```

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears for a *pin_voltage* constraint where either both or none of the `-names` and the `-default` arguments are specified:

```
[FATAL] Either '-names' or '-default' argument should be specified for 'pin_voltage' constraint
```

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears for a *pin_voltage* constraint where the voltage domain specified with the `-voltage` argument has not been defined by using the *voltage_domain* constraint:

```
[FATAL] Voltage domain specified with '-voltage' argument of 'pin_voltage' constraint is not defined using 'voltage_domain' constraint
```

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix these violations, ensure the following:

- **Message 1**: Check the constraints and specify one of module/instname
- **Message 2**: Check the constraints and specify one of names/default arguments
- **Message 3**: Specify the voltage domain with the *voltage_domain* constraint

Example Code and/or Schematic

For the following snippet, the SGDC_lowpower72 rule reports violations because:

- Either -module or -instance argument should be specified for the *pin_voltage* constraint

- Either -names or -default argument should be specified for *pin_voltage* constraint

```
pinvoltage -voltage V1 -instance top.t1 -default -module AN3
-names out3
```

Default Severity Label

Fatal

Rule Group

Constraint Checking Rules

Reports and Related Files

None

SGDC_lowpower75

Reports non-existent pins specified with '-names' argument of pin_voltage constraint

When to Use

Use this rule to check if the pins specified with the *pin_voltage* constraint are actually present in the design.

Description

The *SGDC_lowpower75* rule reports violation messages when a pin specified with the `names` argument of the *pin_voltage* constraint is not a pin of a design unit specified with the `module` argument or an instance specified with the `instance` argument in the design unit that is the `current_design`.

The *SGDC_lowpower75* rule also checks if the specified pin along with the pins of the module/instance matches with the pattern specified using the `module/instance` argument. By default, this rule reports a single message for the wildcarded instance/module name pattern.

Language

Verilog, VHDL

Parameter(s)

- *lp_report_all_wildcards*: Default value is `no`. Set the value to `1` to enable violation messages for each instance/module matched with the specified pattern.

Constraint(s)

SGDC

- *pin_voltage* (Mandatory): Use to specify the voltage/power domain for primary ports or pins of design units or instances in the design.

Messages and Suggested Fix

Message 1

The following message appears when the pin *<pin-name>* specified with

the names argument does not exist as a pin of the design unit *<du-name>* specified with the module argument:

[FATAL] Pin '<pin-name>' of module '<du-name>' does not exist in the description of RTL module/library cell

Potential Issues

The violation message appears if the pin does not exist in the definition of the module.

Consequences of Not Fixing

If you do not fix this violation, properties of the pin are not explored in the definition of the cell.

How to Debug and Fix

Check the module definition/library cell definition and provide the pin with that module/library cell. Alternatively, remove the pin from the design.

Message 2

The following message appears for *pin_voltage* constraint when the pin <pin-name> specified with the names argument does not exist as a pin of the instance <inst-name> specified with the -instance argument:

[FATAL] Pin '<pin-name>' does not exist for instance '<inst-name>' in the design

Potential Issues

The violation message appears if there is a pin is not defined for the instance in the design.

Consequences of Not Fixing

If you do not fix this violation, the pin of the instance is not inferred on the constraint because the pin does not exist for that instance.

How to Debug and Fix

Check the parent of the instance and provide the pin definition. Alternatively, remove the pin from the design.

Message 3

The following message appears for a *pin_voltage* constraint when the pin <pin-name> specified with the names argument does not exist as a pin of the modules matched with the pattern <mod-name-pattern> specified

with the module argument using wildcards (For example: "mid*" will be expanded for all modules initiating with mid):

[FATAL] Pin '<pin-name>' does not exist in the description of RTL module/library cell of one or more modules expanded for '<mod-name-pattern>'

Potential Issues

The violation message appears if the module definitions inferred by the wildcard of module name do not have the pin used in the names arguments of the *pin_voltage* constraint.

Consequences of Not Fixing

If you do not fix this violation, the pin of the module/cell is not inferred on the constraint because the pin does not exist for that module/cell.

How to Debug and Fix

Check the module definitions and provide the pin definitions there. Alternatively, remove the pin from the design.

Message 4

The following message appears when the pin <pin-name> specified with the names argument does not exist as a pin of the instances that matches with the pattern <inst-name-pattern> specified with the instance argument using wildcards:

[FATAL] Pin '<pin-name>' does not exist in one or more instances expanded for '<inst-name-pattern>' in the design

Potential Issues

This violation message appears if the pin specified in the names argument does not exist in the instance names inferred from instance name pattern.

The above violation message appears for all the matched instances when the *lp_report_all_wildcards* parameter is set to 0. Otherwise, individual messages appear for all the matched instances.

Consequences of Not Fixing

If you do not fix the violation, the pin of the instance on which the constraint should have been applied will not be inferred because the pin does not exist for that instance.

How to Debug and Fix

Check the instance definitions and provide the pin definitions. Alternatively, remove the pin from the design.

Example Code and/or Schematic

For the following snippet, the SGDC_lowpower75 rule reports Message 3.

```
pinvoltage -voltage B -module "mid*" -names junk
```

The following violation message appears:

```
[FATAL] Pin 'junk' does not exist in the description of RTL
module/library cell of one or more modules expanded for 'mid*'
```

Default Severity Label

Fatal

Rule Group

VoltageConstraintCheck

Reports and Related Files

None

SGDC_lowpower77

Reports repetition of cells specified in the always_on_pin constraint

When to Use

Use this rule to specify all the always-on pins of a particular cell within a single constraint. This rule runs by default.

Description

The *SGDC_lowpower77* rule reports cells specified multiple times with the `cell` argument of the *always_on_pin* constraint.

The SGDC_lowpower77 rule requires that the cell name specified with the `cell` argument of an *always_on_pin* constraint must be unique for all *always_on_pin* constraints defined under the same environment.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *always_on_pin* (Mandatory): Use to specify always-on cell pins.

Messages and Suggested Fix

The following message appears for the *always_on_pin* constraint when the cell name *<cell-name>* specified with the `cell` argument is already specified with another *always_on_pin* constraint defined under the same environment:

[FATAL] `cell '<cell-name>' has been specified more than once for constraint 'always_on_pin' in the same top design`

Potential Issues

The violation message appears if you specify the same cell name in two different constraints.

Consequences of Not Fixing

If you do not fix this violation, only the last constraint is picked up and some pins missed in the other constraint are not treated as always-on.

How to Debug and Fix

Ensure to specify all always-on pins for same cell in only one *always_on_pin* constraint.

Example Code and/or Schematic

Consider the following example code:

```
current_design top

voltagedomain -name V0 -value 1.2 -modname top

voltagedomain -name V1 -value 1.2 0 -instname top.t1  -isosig
isosig_t -isoval 0 -portname isosig_t

always_on_pin -cell AN2 -pin B

always_on_pin -cell AN3 -pin A B C

always_on_pin -cell "AN*" -pin A B C
```

Since, AN* covers AN2 and AN3 cells and these cells are also specified in another constraint, the following violation appears:

```
[FATAL] Cell 'AN*' has been specified more than once for
constraint 'always_on_pin' in the same top design.
```

Default Severity Label

Fatal

Rule Group

VoltageConstraintCheck

Reports and Related Files

None

SGDC_lowpower78

Reports incorrectly specified cell_tie_class

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower78* rule reports a violation in the following cases:

- The instance specified in the `-instance` argument of the *cell_tie_class* constraint is not a leaf-level instance in the design unit specified as the `current_design`.
- Neither the `-cell` argument nor the `-instance` argument of the *cell_tie_class* constraint is specified, or both the `-cell` and `-instance` arguments are specified.
- The cell or instance pin specified in the `-tie1`, `-tie0`, `-no_tie`, or `-exception` argument of the *cell_tie_class* constraint does not exist in the design unit specified as the `current_design`.
- The net specified in the `-tie1`, `-tie0`, or `-exception` argument of the *cell_tie_class* constraint does not exist in the design unit specified as the `current_design`.
- The `-exception` argument of the *cell_tie_class* constraint does not contain a valid number of values.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *cell_tie_class* (Mandatory): Use this constraint to specify the tie conditions for the multi-power and multi-ground cells.

Messages and Suggested Fix

Message 1

The following message appears when the instance *<instance-name>* specified in the *-instance* argument is not a leaf-level instance in the design unit specified as the current_design:

[FATAL] Instance '<instance-name>' specified with 'instance' argument is not a leaf-level instance in the design

Potential Issues

The cell_tie_class constraint defines logic when applied to a leaf-level cell. Therefore, this violation appears when the -instance argument is not a leaf-level instance in the design unit.

Consequences of Not Fixing

If you do not fix this violation, an intermediate hierarchy is created as an instance name. As a result, the information is not applied to the cell instance, and you may not get the expected results.

How to Debug and Fix

To fix this violation, ensure that the instance specified in the –instance argument of the cell_tie_class constraint in the SGDC file is a leaf-level instance.

Message 2

The following message appears when neither the -cell argument nor the -instance argument is specified, or both the -cell and -instance arguments are specified:

[FATAL] Either '-cell' or '-instance' argument should be specified for 'cell_tie_class' constraint

Potential Issues

The cell_tie_class constraint defines logic with either the –cell or –instance argument so that the tie conditions are specified for a cell or an instance. Therefore, this violation appears if neither the -cell argument nor the -instance argument is specified, or both these arguments are specified.

Consequences of Not Fixing

If you do not fix this violation, the `cell_tie_class` constraint is not applied to any cell or instance, and you may not get the expected results.

How to Debug and Fix

To fix this violation, specify either the `–cell` or `–instance` argument in the `cell_tie_class` constraint.

Message 3

The following message appears when the pin *<pin-name>* of the cell *<cell-name>*, specified in the argument `-tie1`, `-tie0`, `-no_tie`, or `-exception` (*<argument-name>*), does not exist in the design unit specified as the `current_design`:

[FATAL] `Pin '<pin-name>' of cell '<cell-name>' specified with '<argument-name>' argument does not exist in the design`

Potential Issues

This violation appears if the pin name does not exist in the design. This means that the `cell_tie_class` constraint is applied on a non-existent design object.

Consequences of Not Fixing

If you do not fix this violation, the `cell_tie_class` constraint is not applied, and you may not get the expected results.

How to Debug and Fix

To fix this violation, ensure that the pin name specified in the `cell_tie_class` constraint for a particular cell exists in the design unit.

Message 4

The following message appears when the pin *<pin-name>* of the instance *<instancename>*, specified in the argument `-tie1`, `-tie0`, `-no_tie`, or `-exception` (*<argumentname>*), does not exist in the design unit specified as the `current_design`:

[FATAL] `Pin '<pin-name>' for instance '<instance-name>' specified with '<argument-name>' argument does not exist in the design`

Potential Issues

This violation appears if the pin name does not exist in design.

Consequences of Not Fixing

If you do not fix this violation, the `cell_tie_class` constraint is not applied, and you may not get the expected results.

How to Debug and Fix

To fix this violation, ensure that the pin name specified in the `cell_tie_class` constraint for a particular instance does exist in the design unit.

Message 5

The following message appears when the net *<net-name>* specified in the argument `-tie1`, `tie0`, or `-exception` (*<argument-name>*) does not exist in the design unit specified as the `current_design`:

[FATAL] `Net '<net-name>' specified with '<argument-name>' argument does not exist in the design`

Potential Issues

This violation appears if the net name does not exist in the design unit.

Consequences of Not Fixing

If you do not fix this violation, the `cell_tie_class` constraint is not applied if the net does not exist in the design, and you may not get the expected results.

How to Debug and Fix

To fix this violation, ensure that the net name specified in the `cell_tie_class` constraint exists in the design.

Message 6

The following message appears when the `-exception` argument does not contain a valid number of values:

[FATAL] `Argument '-exception' does not contain valid number of values for 'cell_tie_class' constraint`

Potential Issues

The `–exception` argument has a space-separated list of the following

tuples for each exception pin:

<pin-name> <pwr-net-name> <gnd-net-name>

This violation appears if the total number of pins is not divisible by 3 or any of these tuples are missing.

Consequences of Not Fixing

If you do not fix this violation, the `cell_tie_class` constraint is not applied correctly.

How to Debug and Fix

To fix this violation, specify the pin names with the proper tuples in the `–exception` argument of the `cell_tie_class` constraint.

Example Code and/or Schematic

For the following SGDC file, the *SGDC_lowpower78* rule reports a violation because the `top.m.LS2.VDD` net and the `JUNK` pin do not exist in the design.

```
current_design top

 voltagedomain -name V1 -value 1.2 -modname top -supplyname
VDDTOP VSSTOP

 voltagedomain -name V2 -value 1.6 -instname top.m1.BH1_BUF1
top.m1.BH1_BUF3 top.m1.BH1_BUF0 -supplyname VDDEXT VSSEXT

set_cell_tie_class -cell "R120_ASBUF*" -tie1 VDD -tie0 VSS -
no_tie A -exception Z VDD VSS

set_cell_tie_class -cell "R120_ASLS*" -tie1 VDD -tie0 VSS -
no_tie A Z

set_cell_tie_class -instance "top.m1.BH1_BUF2" -tie1
top.VDDTOP -tie0 top.VSSTOP -no_tie A -exception VDDBULK
top.VDDTOP top.VSSTOP

set_cell_tie_class -instance "top.m1.LS2.LS" -tie1
top.VDDTOP -tie0 top.VSSTOP -no_tie A -exception JUNK
top.m.LS2.VDD top.m1.LS2.VSS

supply -name VDDTOP -value 1.2

supply -name VSSTOP -value 0
```

```
supply -name VDDEXT -value 1.6
supply -name VSSEXT -value 0
```

The following violations appear:

```
Net 'top.m.LS2.VDD' specified with '-exception' argument does
not exist in the design.
Pin 'JUNK' for instance 'top.m1.LS2.LS' specified with '-
exception' argument does not exist in the design
```

Default Severity Label

Fatal

Rule Group

VoltageConstraintCheck

Reports and Related Files

None

SGDC_lowpower82

Reports incorrect sleep, save, and restore signal specifications

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower82* rule reports violations in the following cases:

- The `sleep`, `save` and `restore` signals are specified for a voltage domain. These signals should be specified only for the power domains.
- The same `sleep`, `save` and `restore` signals are specified for more than one power domain. Only one of the `sleep`, `save` and `restore` signals must be specified for each power domain and this sleep signal must be unique for the power domain.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain* (Mandatory): Use this argument to specify the voltage or power domains of design units and instances.

Messages and Suggested Fix

Message 1

The following message appears for the `voltage_domain` constraint that defines a voltage domain when either of the `-sleepnet`, `-savenet` and `-restorenet` arguments, (*<fld-type>*), is also specified:

[FATAL] `Field <fld-type> not allowed for a voltage domain`

Potential Issues

This violation appears if the `-sleepnet`, `-savenet` and `-restorenet` arguments are defined for a voltage domain. As a voltage domain never

shuts down, there is no need to define these arguments for a voltage domain.

Consequences of Not Fixing

If you do not fix the violation, the `voltage_domain` constraint is ignored.

How to Debug and Fix

To fix this violation, ensure not to specify the `–sleepnet`, `-savenet` and `– restorenet` arguments for a voltage domain.

Message 2

The following message appears for the `voltage_domain` constraint specifying a power domain when the *<signal-type>* `sleep`, `save` or `restore` signal *<sig-name>* is specified for at least one other power domain:

[FATAL] `Same <signal-type> signal '<sig-name>' used for more than one power domain`

Potential Issues

This violation appears if the signal *<sig-name>* is specified for more than one power domain.

Consequences of Not Fixing

If you do not fix this violation, the tool may restore the register values for the other power domain when you want to restore register values for just one power domain.

How to Debug and Fix

To fix this violation, ensure that the `sleep`, `save` or `restore` signals should not be same for any two power domains.

Example Code and/or Schematic

Example 1

For the following SGDC file, the *SGDC_lowpower82* rule reports a violation because `V1` is a voltage domain.

```
current_design top
voltagedomain -name V1 -value 1.2 -modname top -sleepnet
```

```
top.in1 -sleepval 1

voltagedomain -name V2 -value 1.3 0 -instname top.mid_inst -
isosig rst -isoval 0

voltagedomain -name V3 -value 1.3 0 -instname
top.mid_inst.bot_inst -isosig rst -isoval 0

retencell -name FD1 -sleep D
```

The following violations appear:

```
Field '-sleepnet' not allowed for a voltage domain

Field '-sleepval' not allowed for a voltage domain
```

Example 2

For the following SGDC file, the *SGDC_lowpower82* rule reports a violation because the same sleep signal `top.rst` is specified in both the `V2` and `V3` domains.

```
current_design top

voltagedomain -name V1 -value 1.2 -modname top

voltagedomain -name V2 -value 1.3 0 -instname top.mid_inst -
isosig rst -isoval 0 -sleepnet top.rst -sleepval 0

voltagedomain -name V3 -value 1.3 0 -instname
top.mid_inst.bot_inst -isosig rst -isoval 0 -sleepnet top.rst
-sleepval 1

retencell -name FD1 -sleep D
```

The following violation appear:

```
Same sleep signal 'top.rst' used for more than one power domain
```

Default Severity Label

Fatal

Rule Group

`VoltageConstraintCheck`

Reports and Related Files

None

SGDC_lowpower85

Reports incorrectly specified stopclockval of voltage_domain

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower85* rule reports a violation, if you specify values other than 0 and 1 in the `-stopclockval` argument of the *voltage_domain* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage or power domains of design units and instances.

Messages and Suggested Fix

The following message appears when the `-stopclockval` argument of the `voltage_domain` constraint is specified with a value other than 0 and 1:

[FATAL] `Field '-stopclockval' of 'voltage_domain' constraint can have value 0 or 1`

Potential Issues

This violation appears when the `-stopclockval` argument holds a value other than 0 and 1.

Consequences of Not Fixing

If you do not fix this violation, the rules that are using this clock value may not work properly.

How to Debug and Fix

To fix this violation, specify 0 or 1 in the `-stopclockval` argument of the `voltage_domain` constraint.

Example Code and/or Schematic

For the following SGDC file, the *SGDC_lowpower85* rule reports a violation because the `-stopclockval` argument is specified as 2.

```
current_design CKT_TOP

voltagedomain -name V1 -value 1.2 -modname CKT_TOP

voltagedomain -name V2 -value 1.3 0 -instname CKT_TOP.U1
-isosig BFC_B -isoval 1 -sleepnet CKT_TOP.BFC_B.junk -
sleepval 1 -stopclock CKT_TOP.BFC_B -stopclockval 2

retencell -name FD1S3AQV15 -domains V2 -clk CK -clkval 1
```

In the given example, the following violation message appears:

```
Field ‘–stopclockval’ argument of ‘voltage_domain’ constraint
can have value 0 or 1
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_lowpower86

Reports incorrectly specified clkval argument of retention_cell

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower86* rule reports a violation, if you specify values other than 0 or 1 in the `clkval` argument of the *retention_cell* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *retention_cell* (Mandatory): Use this constraint to specify the retention latch cells.

Messages and Suggested Fix

The following message appears if the `-clkval` argument of the *retention_cell* constraint has a value other than 0 and 1:

[FATAL] `Field '-clkval' of 'retention_cell' constraint can have value 0 or 1`

Potential Issues

This violation appears when the `-clkval` argument has a value other than 0 and 1.

Consequences of Not Fixing

If you do not fix this violation, the rules that are using this clock value may not work properly.

How to Debug and Fix

To fix this violation, ensure that the `-clkval` argument is has either 0 or 1.

Example Code and/or Schematic

For the following SGDC file, the *SGDC_lowpower86* rule reports a violation because the `-clkval` argument is specified as 3.

```
current_design CKT_TOP

voltagedomain -name V1 -value 1.2 -modname CKT_TOP

voltagedomain -name V2 -value 1.3 0 -instname CKT_TOP.U1 -isosig BFC_B -isoval 1 -sleepnet CKT_TOP.BFC_B -sleepval 1 -stopclock CKT_TOP.BFC_B -stopclockval l 0

retencell -name FD1S3AQV15 -domains V2 -clk CK -clkval 3
```

In the above example, the following violation message appears:

```
Field '-clkval' of 'retention_cell' constraint can have value 0 or 1
```

Default Severity Label

Fatal

Rule Group

`VoltageConstraintCheck`

Reports and Related Files

None

SGDC_lowpower87

Reports incorrect arguments of the supply constraint

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower87* rule reports a violation in the following scenarios:

- The `-name` argument of the *supply* constraint does not exist in the design.
- If the `-on` argument is specified and the `-alwayson` argument is specified as `1`.
- The number of arguments specified in the `-on` argument is not equal to the number of arguments specified in the `-onval` argument.
- The `-parent` argument of the *supply* constraint is not defined as supply.
- The `-parent` argument specified for the switchable supply is not an always-on supply.
- The `-parent` argument for the switchable supply is not specified when the *LPPLIB17* rule is enabled.
- The `-isosig` argument is specified for an always-on supply.
- The `-isosig` argument is not specified for a switched supply, when the *LPSVM49* rule is enabled and the *lp_check_isocell* parameter is set.
- If the `-on` or `-on_2` arguments are specified and the `–alwayson` argument is also set as `1`.
- The number of values specified in the `-onval_2` argument is different from the number of values specified in the `-on_2` argument.

Language

Verilog, VHDL

Parameter(s)

- *lp_check_isocell*: Default value is 1. Set this parameter to 0 to enable the *LPSVM49* violations for the net crossings from the power domain gate to proper isolation gate.

Constraint(s)

- *supply* (Mandatory): Use this constraint to specify the supply and ground port names.

Messages and Suggested Fix

Message 1

The following message appears when the specified supply name does not exist in the design:

[FATAL] Supply '<supply-name>' does not exist in the design

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the -parent argument *<arg>* of the supply constraint is not defined as supply:

[FATAL] The '-parent' argument '<arg>' should be defined through '-name' argument of supply constraint.

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when the -on argument *<arg>* is specified and the -alwayson argument is specified as 1:

[FATAL] The '-alwayson' argument should not be given '1' if '-on' argument is specified.

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears when the number of arguments specified in the -on argument is not equal to the number of arguments specified in the -onval argument:

[FATAL] The number of values specified through '-on' argument

and '-onval' argument should be same

For debugging information, click *How to Debug and Fix*.

Message 5

The following message appears when the `-parent` argument of the supply <*supply-name*> is not an always-on supply:

[FATAL] The '-parent' argument of supply '<supply-name>' should be always-on supply

For debugging information, click *How to Debug and Fix*.

Message 6

The following message appears when the `-parent` argument for a switchable supply is not specified, when the *LPPLIB17* rule is enabled:

[FATAL] The '-parent' argument for switchable supply '<supply-name>' should be specified, when rule LPPLIB17 is enabled

For debugging information, click *How to Debug and Fix*.

Message 7

The following message appears when the `-isosig` argument is specified for the always-on supply <*supply-name*>:

[FATAL] The '-isosig' argument should not be given for alwayson supply '<supply-name>'

For debugging information, click *How to Debug and Fix*.

Message 8

The following message appears when the `-isosig` argument is specified for the switchable supply <*supply-name*>, when the *LPSVM49* rule is enabled and the `lp_check_isocell` parameter is set:

[FATAL] The isolation signal(s) should be specified by '-isosig' argument for switched supply '<supply-name>', when rule LPSVM49 is enabled

For debugging information, click *How to Debug and Fix*.

Message 9

The following message appears when the `-on` or `-on_2` argument <*arg*> is specified and the `-alwayson` argument is specified as `1`:

[FATAL] The '-alwayson' argument should not be given '1' if '-on/-on_2' argument is specified.

For debugging information, click *How to Debug and Fix*.

Message 10

The following message appears when the number of arguments specified in the -on_2 argument is not equal to the number of arguments specified in the -onval_2 argument:

[FATAL] The number of values specified through '-on_2' argument and '-onval_2' argument should be same

Potential Issues

These violations appear when you specify incorrect argument values in the supply constraint.

Consequences of Not Fixing

The incorrect argument values in the *supply* constraint affect the rules that are using the *supply* constraint. It may produce incorrect results.

How to Debug and Fix

To fix this violation, specify valid values in the arguments of the supply constraint.

Example Code and/or Schematic

For the following SGDC file, the *SGDC_lowpower87* rule reports a violation because the –alwayson and –parent arguments are specified with incorrect values.

```
current_design top

voltagedomain -name Vtop -value 1.0 -modname top -supplyname VDDA

voltagedomain -name VA -value 1.0 0 -instname ua -supplyname VDD VDDA -noisosig

powerswitch -name PWRSW -pwroutpin VDDO -pwrinpin VDDI -en_inv_in ON

supply -name VDD -value 1.0 -parent VDDX -on !ENA&VDD ENA !ENA -alwayson 1
```

```
supply -name VDDA -value 1.0 -alwayson 1
```

Default Severity Label

Fatal

Rule Group

Constraints Checking

Reports and Related Files

None

SGDC_lowpower89

Reports incorrect voltage domain specified in the -to or -from argument of ignore_crossing

When to Use

Use this rule to detect errors in the voltage domain specification.

Description

The *SGDC_lowpower89* rule reports incorrect voltage domains specified in the `-to` or `-from` arguments of the *ignore_crossing* constraints.

The *SGDC_lowpower89* rule requires that each voltage domain name specified in the `-to` or `-from` argument of the *ignore_crossing* constraint must be defined in the *voltage_domain* constraint.

Define both the `-to` and `-from` arguments of the *ignore_crossing* constraint with a valid voltage domain or a power domain.

The `-to` and `-from` arguments of the *ignore_crossing* constraint should have different values.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *ignore_crossing* (Mandatory): Use this constraint to specify the power domain-to-voltage domain and power domain to power domain crossings.

Messages and Suggested Fix

Message 1

The following message appears when the voltage domain name <*vd-name*> specified in the `-to` argument of the `ignore_crossing` constraint is not defined in the `voltage_domain` constraint:

[FATAL] To domain '<vd-name>' given with the 'ignore_crossing' is not defined as a voltage domain in the SGDC file

Potential Issues

This violation appears if the `-to` or `-from` argument of the `ignore_crossing` constraint is applied on the voltage domain that is not defined in the `voltage_domain` constraint in the SGDC file.

Consequences of Not Fixing

If you do not fix this violation, the `ignore_crossing` constraint is not applied on the specified domain.

How to Debug and Fix

To fix this violation, define the voltage domain specified in the `–to` or `-from` argument of the `ignore_crossing` constraint, in the `voltage_domain` constraint.

Message 2

The following message appears when the `-to` and `-from` arguments are same for a voltage or power domain:

[FATAL]'-to' and '-from' field of 'ignore_crossing' should be different.

Potential Issues

This violation appears when the `-to` and `-from` arguments of the `ignore_crossing` constraint have the same values.

Consequences of Not Fixing

If you do not fix this violation, the `ignore_crossing` constraint may not be applied correctly.

How to Debug and Fix

To fix this violation, specify different valid voltage domains in the `–to` and `–from` arguments of the `ignore_crossing` constraint.

Message 3

The following message appears when the power domain name *<pd-name>* specified in the `-from` argument is not a power domain:

[FATAL] Domain '<pd-name>' specified with field '-from' of

'ignore_crossing' constraint is not a 'powerdomain'

Potential Issues

This violation appears if the domain name specified in the `-from` argument is not a power domain. Since a voltage domain is an always-on domain, the `-from` argument of the `ignore_crossing` constraint should logically be a power domain only.

Consequences of Not Fixing

If you do not fix this violation, there may be an error in the design and the `ignore_crossing` constraint may not be applied correctly.

How to Debug and Fix

To fix this violation, ensure that the domain specified in the `-from` argument of the `ignore_crossing` constraint is a valid power domain.

Example Code and/or Schematic

For the following SGDC file, the *SGDC_lowpower89* rule reports a violation because the voltage domain `V7`, specified in the `-to` argument of the `ignore_crossing` constraint, is not defined in the `voltage_domain` constraint.

```
current_design top1

voltage_domain -name V -value 7.3 -modname top1 #-isosig in1 in2 -isoval 0 1

voltage_domain -name V1 -value 1.20 -instname top1.O12

voltage_domain -name V2 -value 1.49 -instname top1.A21

voltage_domain -name V3 -value 1.10 -instname top1.A11

voltage_domain -name V4 -value 1.12 0 -instname top1.A1111  \

-     -isosig top1.in1[3:0] top1.in2 \

|     -isoval 0  1 \

ignore_crossing -from V1 -to V7

ignore_crossing -from V4 -to V3

ignore_crossing -from V5 -to V3

ignore_crossing -from V9 -to V3
```

Similarly, the rule reports a violation because the voltage domains V1, V5, and V9, specified in the `-from` argument of the `ignore_crossing` constraint, are not power domains.

The following violation messages appear:

```
To domain 'V7' given with the 'ignore_crossing' is not defined as a voltage domain in the SGDC file

Domain 'V1' specified with field '-from' of 'ignore_crossing' constraint is not a 'powerdomain'

Domain 'V5' specified with field '-from' of 'ignore_crossing' constraint is not a 'powerdomain'

Domain 'V9' specified with field '-from' of 'ignore_crossing' constraint is not a 'powerdomain'
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_lowpower90

Checks the usage of voltage_domain and retention_cell while running the LPSVM58 rule

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower90* rule reports a violation when the following conditions are not met:

- For the *voltage_domain* constraint, the `-sleepnet` argument should be specified with the `-sleepval` argument.
- For the *voltage_domain* constraint, the `-sleepval` argument should be specified with the `-sleepnet` argument.
- For the *retention_cell* constraint, the `-clk` argument should be specified with the `-clkval` argument.
- For the *retention_cell* constraint, the `-clkval` argument should be specified with the `-clk` argument.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage power domains in the design.
- *retention_cell* (Mandatory): Use this constraint to specify the retention latch cell.

Messages and Suggested Fix

Message 1

The following message appears when the `–sleepnet` argument of a

power domain is not specified with the –sleepval argument:

[FATAL]'-sleepnet' argument for powerdomain should be specified with '-sleepval' argument, while running rule LPSVM58.

Potential Issues

This violation appears when the –sleepnet argument of a power domain is not specified with the –sleepval argument.

Consequences of Not Fixing

If you do not fix this violation, the voltage_domain constraint is ignored by the *LPSVM58* rule.

How to Debug and Fix

To fix this violation, specify the –sleepnet argument with the -sleepval argument in the voltage_domain constraint, while running the *LPSVM58* rule.

Message 2

The following message appears when the –sleepval argument of a power domain is not specified with the –sleepnet argument:

[FATAL]'-sleepval' argument for powerdomain should be specified with '-sleepnet' argument, while running rule LPSVM58.

Potential Issues

This violation appears when the -sleepval argument is not specified with the -sleepnet argument.

Consequences of Not Fixing

If you do not fix this violation, the voltage_domain constraint is ignored by the *LPSVM58* rule.

How to Debug and Fix

To fix this violation, specify the –sleepval argument with the -sleepnet argument in the voltage_domain constraint, while running the *LPSVM58* rule.

Message 3

The following message appears when the –clk argument of the *retention_cell* constraint is not specified with the –clkval argument:

[FATAL]'-clk' argument for 'retention_cell' constraint should be specified, if '-clkval' argument is specified.

Potential Issues

This violation appears if the -clk argument is not specified with the -clkval argument.

Consequences of Not Fixing

If you do not fix this violation, the *retention_cell* constraint is ignored by the *LPSVM58* rule.

How to Debug and Fix

To fix this violation, specify the –clk argument with the -clkval argument in the *retention_cell* constraint, while running the *LPSVM58* rule.

Message 4

The following message appears when the -clkval argument of the *retention_cell* constraint is not specified with the –clk argument:

[FATAL]'-clkval' argument for 'retention_cell' constraint should be specified, if '-clk' argument is specified.

Potential Issues

This violation appears if the -clkval argument is not specified with the -clk argument.

Consequences of Not Fixing

If you do not fix this violation, the *retention_cell* constraint is ignored by the *LPSVM58* rule.

How to Debug and Fix

To fix this violation, specify the –clkval argument with the -clk argument in the *retention_cell* constraint, while running the *LPSVM58* rule.

Example Code and/or Schematic

Example 1

For the following code, the *SGDC_lowpower90* rule reports a violation because the –sleepnet argument is not specified with the –sleepval argument.

```
voltage_domain -name V2 -value 1.3 0 -instname CKT_TOP.U1 -
isosig BFC_B -isoval 1 -sleepval 1
```

Example 2

For the following code, the *SGDC_lowpower90* rule reports a violation because the `–sleepval` argument is not specified with the `–sleepnet` argument.

```
voltage_domain -name V2 -value 1.3 0 -instname CKT_TOP.U1 -
isosig BFC_B -isoval 1 -sleepnet CKT_TOP.BFC_B
```

Example 3

For the following code, the *SGDC_lowpower90* rule reports a violation because the `–clk` argument is not specified with the `–clkval` argument.

```
retention_cell -name FD1S3AQV15 -clkval 1
```

Example 4

For the following code, the *SGDC_lowpower90* rule reports a violation because the `–clkval` argument is not specified with the `–clk` argument.

```
retention_cell -name FD1S3AQV15 -clk CK
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_lowpower91

Reports if steady-state condition names is not correctly specified

When to Use

This is a setup rule and runs by default.

Description

The *SGDC_lowpower91* rule reports if steady-state condition names specified with the `name` argument of the *domain_outputs* constraint is not specified with the `output` argument of the *voltage_domain* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Mandatory): Use to specify voltage/power domains in the design.
- *domain_outputs* (Mandatory): Use to specify the values of various signals under the steady-state condition specified in the *voltage_domain* constraint.

Messages and Suggested Fix

The following message appears when steady-state condition names <ss-condition-name> specified using the `name` argument of the *domain_outputs* constraint is not specified with the `outputs` argument of the *voltage_domain* constraint:

[FATAL] The steady-state condition name '<ss-condition-name>' specified with '-name' argument of 'domain_outputs' constraint should also be specified with '-outputs' argument of 'voltage_domain' constraint"

Potential Issues

The violation message appears if you do not associate the steady-state conditions with any *voltage_domain* declaration.

Consequences of Not Fixing

If you do not fix this violation, the steady-state conditions are not applicable anywhere in the design.

How to Debug and Fix

Specify the steady-state conditions with the `outputs` argument of *voltage_domain* constraint.

Example Code and/or Schematic

Consider the following SGDC file fragment:

```
current_design mid

voltagedomain -name V1 -value 1.2 -modname mid

voltagedomain -name V2  -value 1.2 0 -instname mid.t1.inst4 -isosig mid.isosig -isoval 0 -generate_iso_logic -outputs OUT -portname "op1"

powerdomainoutputs -name OUT2 -value mid.t1.c2_int[1] 1
```

The *SGDC_lowpower91* rule reports the following violation message:

```
[FATAL] The steady-state condition name 'OUT2' specified with '-name' argument of 'domain_outputs' constraint should also be specified with '-outputs' argument of 'voltage_domain' constraint
```

Default Severity Label

Fatal

Rule Group

VoltageConstraintCheck

Reports and Related Files

No related files or reports.

SGDC_lowpower92

Performs an existence check of power down condition name

When to Use

This is a setup rule and it runs by default. This rule is run only when the LPSVM47 rule is selected.

Description

The *SGDC_lowpower92* rule reports non-existent power down condition names specified using the *domain_inputs* constraint and with `-inputs` argument of the *voltage_domain* constraint, while running the LPSVM47 rule.

Prerequisites

Enable the LPSVM47 rule.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *domain_inputs* (Mandatory): Use to specify the expected values of various inputs of power domain under the power down condition.
- *voltage_domain* (Mandatory): Use to specify the voltage/ power domains in the design.

Messages and Suggested Fix

Message 1

The following message appears when *LPSVM47* rule is run and the `inputs` argument of the *voltage_domain* constraint is not present:

```
[WARNING] For powerdomain field '-inputs' is not present when rule 'LPSVM47' is run.
```

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when power down condition name <*pd-condition-name*> is not specified with the `-inputs` argument of the *voltage_domain* constraint:

```
[WARNING] The powerdown condition name '<pd_condition-name>'
specified with '-name' field in 'domain_inputs' constraints
should also be present with '-inputs' argument
of'voltage_domain' constraint.
```

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

Though the SpyGlass Power Verify solution will continue with its analysis, the results may not be as expected.

How to Debug and Fix

The constraint reported in the violation message is highlighted in the Atrenta Console GUI.

To fix this violation, ensure the following:

- The `name` argument for the *domain_inputs* constraint has also been specified with the `inputs` argument.
- For power domain argument, the `inputs` argument must be specified by using the *voltage_domain* constraint.

Example Code and/or Schematic

Consider the following code:

```
voltagedomain  -name V1 -value 1.2 -modname top

voltagedomain -name V2  -value 1.2 0 -instname top.pd -isosig
iso -isoval 0

powerdomaininputs -name PD_IN_V2 -value in1 1 in2 1 in3 1
```

For this snippet, the *SGDC_lowpower92* rule reports the following violations:

```
For powerdomain field '-inputs' is not present when rule
'LPSVM47' is run.
```

```
The powerdown condition name 'PD_IN_V2' specified with '-name'
field in 'domain_inputs' constraints should also be present
with '-inputs' argument of 'voltage_domain' constraint.
```

To resolve the violations, specify the `-inputs` argument, as shown in the following snippet:

```
voltagedomain  -name V1 -value 1.2 -modname top

voltagedomain -name V2  -value 1.2 0 -instname top.pd -isosig
iso -isoval 0 -inputs PD_IN_V2

powerdomaininputs -name PD_IN_V2 -value in1 1 in2 1 in3 1
```

Default Severity Label

Fatal

Rule Group

Constraint Checking Rules

Reports and Related Files

None

SGDC_lowpower93

Checks incorrectly defined level shifter

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower93* rule reports the following cases:

- If domain name specified for both `from` and `to` arguments is same.
- If any pin is specified with more than one of the following arguments: `inTerm`, `outTerm`, `outSupplyTerm`, or `enableTerm`.
- If the domain specified using the `locate` argument of the *levelshifter* constraint does not match with the source/destination domain specified using `from`/`to` argument respectively.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *levelshifter* (Mandatory): Use to specify the names of design units to be used as level shifters.

Messages and Suggested Fix

Message 1

The following message appears when arguments <*arg1*> and <*arg2*> are the same:

[FATAL]'<arg1>' and '<arg2>' argument of 'levelshifter' constraint should not be same.

Where, <*arg1*> and <*arg2*> can be `inTerm`, `-outTerm`, `-outSupplyTerm`, or `-enableTerm`.

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the *levelshifter* constraint is specified incorrectly:

[FATAL] '<name1>' field of 'levelshifter' constraint should not be same as '<name2>'.

Where, <*name1*> and <*name2*> can be domain or pin names.

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when the domain name <dom-name> specified by the locate argument of the *levelshifter* constraint does not match with the source/destination domain name specified using from/to argument respectively:

[FATAL] Domain '<dom-name>' specified by '-locate' field does not match with '<src-dom-name>' specified as '-from' domain or '<dest-dom-name>' specified as '-to' domain

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix these violations, ensure the following:

- The domain name specified for from and to arguments is not the same.
- No pin is specified with more than one of the following arguments: inTerm, outTerm, outSupplyTerm, or enableTerm.
- The same domain is specified in the locate argument and the from/to argument.

Example Code and/or Schematic

Example 1

Consider the following code:

```
levelshifter -name LS1 -from V2 -to V1 -inTerm A -outTerm A -
enableTerm A
```

The violations reported are:

```
-inTerm' and '-enableTerm' argument of 'levelshifter'
constraint should not be same

-inTerm' and '-outTerm' argument of 'levelshifter' constraint
should not be same

-outTerm' and '-enableTerm' argument of 'levelshifter'
constraint should not be same
```

To resolve the violation, do not use same arguments.

Example 2

Consider the following code:

```
levelshifter -name LS1 -from V1 -to V1 -inTerm A.
```

The following violation is reported:

```
-to' and '-from' argument of 'levelshifter' constraint should
not be same.
```

To resolve the violation:

```
levelshifter -name LS1 -from V1 -to V2 -inTerm A.
```

Above case is for same domain names, a similar case for same pin names also.

Example 3

Consider the following code:

```
levelshifter –name LS1 -from V2 -to V1 -inTerm A -enableTerm
EN  -locate srk
```

The following violation is reported:

```
Domain 'srk' specified by '-locate' field does not match with
'V2' specified as '-from' domain or 'V1' specified as '-to'
domain
```

To resolve the violation, use any of the following four possible ways:

- levelshifter –name LS1 -from V2 -to V1 -inTerm A -enableTerm EN -locate src
- levelshifter –name LS1 -from V2 -to V1 -inTerm A -enableTerm EN -locate dest
- levelshifter –name LS1 -from V2 -to V1 -inTerm A -enableTerm EN -locate V2
- levelshifter –name LS1 -from V2 -to V1 -inTerm A -enableTerm EN -locate V1

Default Severity Label

Fatal

Rule Group

Constraint Checking Rules

Reports and Related Files

None

SGDC_lowpower94

Reports incorrectly specified power_state

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower94* rule reports the following cases:

- The domain name specified using the `domains` argument does not match with any of the names defined using the *voltage_domain* constraint.
- The domain value specified using the `domains` argument is not a positive valid float value.
- The voltage value of domain is missing in the `domains` argument.
- The domain name specified using the `domains` argument is given more than once for the same power state.
- The valid active voltage value (greater than zero) is not specified for a voltage domain in any of the defined power state.
- The voltage value for a power domain is not specified in any of the defined power state.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *power_state* (Mandatory): Use to specify combinations of domain states that can exist at the same time during operation of the design.
- *voltage_domain* (Mandatory): Use to specify voltage/ power domains in the design.

Messages and Suggested Fix

Message 1

The following message appears when the specified domain name <*domain-name*> does not match with domain name defined using the *voltage_domain* constraint:

[FATAL] The domain name '<domain-name>' specified through '-domains' argument should be defined with '-name' argument of 'voltage_domain' constraint

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the specified voltage value <*value*> of the domain <*domain-name*> is not a valid float value:

[FATAL] The voltage value '<value> of domain '<domain-name>' specified through '-domains' argument should be positive float type

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when voltage value is not specified for the domain <*domain-name*> in the power state <*pwr-st-name*>*:*

[FATAL] The voltage value should be specified for domain '<domain-name>' in '-domains' argument for power state '<pwr-st-name>'

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears when you specify the domain <*domain-name*> more than once using `-domains` argument for the power state <*pwr-st-name*>*:*

[FATAL] In '-domains' field domain name '<domain-name>' has been specified more than once for '<pwr-st-name>' power state

For debugging information, click *How to Debug and Fix*.

Message 5

The following message appears when the voltage value <*vol-value*>

specified for the domain <*domain-name*> in the power state <*pwr-st-name*> is invalid:

[FATAL] The voltage value for domain '<domain-name>' should not be specified as '<vol-value>' for power state '<pwr-st-name>'

For debugging information, click *How to Debug and Fix*.

Message 6

The following message appears when the power domain <*domain-name*> has not been specified with value greater than zero for any of the defined power states:

[FATAL] The powerdomain '<domain-name>' should have active value specified in at least one power state

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

If you do not fix the violation, the state will be ignored by the Lowpower rules.

How to Debug and Fix

The constraint reported in the violation message is highlighted in the Atrenta Console GUI.

To fix this violation, ensure the following:

- **Message 1:** Ensure that any identifier used as domain in `-domains` field of *power_state* constraint is predefined by using the *voltage_domain* constraint.
- **Message 2:** Specify only positive float voltage values in the *power_state* constraint
- **Message 3:** Specify voltage values for each domain in the *power_state* constraint.
- **Message 4:** Ensure to specify the domain only once in the same *power_state* constraint.
- **Message 5:** Specify a valid voltage value for the domain.
- **Message 6:** Specify the domain with its active value at least once in *power_state* constraints.

Example Code and/or Schematic

Example 1

For the following snippet, the SGDC_lowpower94 rule reports a violation because vt is not defined as a domain in the *voltage_domain* constraint.

```
Voltage_domain -name Vtop -value 1.2 -modname top

Voltage_domain -name VA -value 1.2 -instname top.inst1

Voltage_domain -name VB -value 1.2 -instname top.inst2 inst3

power_state -name LL -domains VA@0.8 VB@0.8 Vt@1.2
```

Example 2

For the following snippet, the SGDC_lowpower94 rule reports a violation because aa is not a float value and there is a negative value specified.

```
power_state -name LH -domains VA@0.8 VB@1.2 Vtop@aa

power_state -name LL -domains VA@-0.8 VB@0.8 Vt@ Vtop@aa
```

Example 3

For the following snippet, the SGDC_lowpower94 rule reports a violation because the voltage value is not specified after VA@.

```
power_state -name HL -domains VA@ VB@0.8 Vtop@0.8
```

Example 4

For the following snippet, the SGDC_lowpower94 rule reports a violation because the V2 domain has been specified more than once in the same *power_state* constraint.

```
power_state -name PowerON -domains V2@1.2 V1@1.2 V2@1.0
```

Example 5

For the following snippet, the SGDC_lowpower94 rule reports a violation because the V4 domain with its active state has not been mentioned in any of the *power_state* constraints.

```
Voltage_domain -name V0 -value 1.2 -modname top

Voltage_domain -name V1 -value 1.2 0 -instname top.inst1
top.inst2 -isosig isosig_t -isoval 0

Voltage_domain -name V2 -value 1.2 0 -instname top.inst3
```

```
top.inst4 -isosig isosig_n -isoval 1

Voltage_domain -name V3 -value 1.5 0 -instname top.f1  -
isosig isosig -isoval 0

voltage_domain -name V4 -value 1.2 0 -instname top.inst5
top.inst6  -isosig isosig_n -isoval 0

power_state -name PowerON -domains V1@1.2 V2@1.2

power_state -name PowerON_1 -domains V1@1.2 V2@1.2 V3@1.0

power_state -name PowerON_2 -domains V1@1.2 V2@1.2 V3@1

power_state -name PowerOFF -domains V1@11 V2@01 V3@12.5
```

Default Severity Label

Fatal

Rule Group

Constraint Checking Rules

Reports and Related Files

None

SGDC_lowpower95

Checks the validity of the –tie1 and –tie0 arguments of cell_tie_class

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower95* rule reports a violation if the following conditions are not satisfied:

- The `-tie1` and `-tie0` arguments, specified by using the *cell_tie_class* constraint, should not have the same values.
- The `-tie1` and `-tie0` arguments, specified by using the *cell_tie_class* constraint, should be of the `power` and `ground` types, respectively.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *cell_tie_class* (Mandatory): Use this constraint to specify the tie conditions for multi-power and multi-ground cells.

Messages and Suggested Fix

Message 1

The following message appears when the specified `-tie1` and `-tie0` arguments are same:

[FATAL]`'-tie1' and '-tie0' argument for 'cell_tie_class' constraint should not be same'.`

Potential Issues

This violation appears if you specify the `–tie1` and `-tie0` arguments that have the same values for the `cell_tie_class` constraint.

Consequences of Not Fixing

If you do not fix this violation, the `cell_tie_class` constraint is ignored by the rules of the SpyGlass Power Verify solution in the SGDC file.

How to Debug and Fix

To fix this violation, specify different `-tie1` and `-tie0` arguments for the `cell_tie_class` constraint.

Message 2

The following message appears when the specified pin *<pin-name>* is not of the type *<pin-type>* `power` or `ground`:

[FATAL] <pin-name> pin specified with argument '<arg>' of 'cell_tie_class' constraint is not of type '<pin-type>'

Potential Issues

This violation appears if a mismatch occurs between the pin specified in the `-tie1` or `-tie0` argument and its type. The following two issues are possible:

- The pin of the `power` type is specified in the `-tie0` argument.
- The pin of the `ground` type is specified in the `-tie1` argument.

Consequences of Not Fixing

If you do not fix this violation, the `cell_tie_class` constraint is ignored by the rules of the SpyGlass Power Verify solution in the SGDC file.

How to Debug and Fix

To fix this violation, specify the pins in the `-tie0` and `–tie1` arguments as the `ground` and `power` types, respectively.

Example Code and/or Schematic

Example 1

For the following code, the *SGDC_lowpower95* rule reports a violation because same argument, VSS, is present in both the `–tie1` and `–tie0` arguments.

```
set_cell_tie_class -tie1 VSS -tie0 VSS -no_tie A
```

Example 2

For the following code, the *SGDC_lowpower95* rule reports a violation because the `VDDEXT` pin is specified as the `power` type.

```
set_cell_tie_class -cell R120_ASBUFX010 -tie1 VDDBULK -tie0
VDDEXT -no_tie A
```

Where, the VDDEXT pin is specified as the power pin in the .lib file, as shown below:

```
pin(VDDEXT) {
        direction : inout;
        pin_type : power;
          }
```

Default Severity Label

Fatal

Rule Group

`VoltageConstraintCheck`

Reports and Related Files

None

SGDC_lowpower96

Reports non-existent modules specified in level shifter

When to Use

This is a setup rule and it runs by default.

Description

The SGDC_lowpower96 rule reports non-existent module names that are specified using the -name argument of the *levelshifter* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *levelshifter* (Mandatory): Use to specify the names of design units to be used as level shifters.

Messages and Suggested Fix

The following message appears when the specified module <*module-name*> does not exist or is not instantiated in the design <*top-du-name*>:

[WARNING] '<du-name>' [<module-name>] is never instantiated within environment '<top-du-name>

Potential Issues

The module specified as a level shifter with the *levelshifter* constraint has not been used in the design.

Consequences of Not Fixing

There is no point specifying a module as level shifter if it is not used in the design.

How to Debug and Fix

The SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix this violation, instantiate and use the module in the design or remove it from the *levelshifter* constraint

Example Code and/or Schematic

For the following snippet, the *SGDC_lowpower96* rule reports a violation because the `is2` module has not been used in the design.

```
levelshifter -name ls2  -from Va -to Vtop -enableTerm E
```

To resolve this violation, use a pre-defined module.

Default Severity Label

Warning

Rule Group

`Constraint Checking Rules`

Reports and Related Files

None

SGDC_lowpower97

Reports incorrectly defined constraints when the LPSVM59 rule is run

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower97* rule checks the following cases for the *voltage_domain* constraint when the `LPSVM59` rule is enabled:

- `-sleepval` argument should be specified with `-sleepnet` argument.
- `-saveval` argument should be specified with `-savenet` argument.
- `-restoreval` argument should be specified with `restorenet` argument

The *SGDC_lowpower97* rule checks the following cases for the *retention_cell* constraint when rule `LPSVM59` is enabled:

- `-sleepval` argument should be specified with `-sleep` argument.
- `-saveval` argument should be specified with `-save` argument.
- `-restoreval` argument should be specified with `-restore` argument

Prerequisites

Enable the LPSVM59 rule.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain* (Mandatory): Use to specify voltage or power domains in the design.
- *retention_cell* (Mandatory): Use to specify the retention latch cells.

Messages and Suggested Fix

The following message appears when an argument <agr1-name> is not specified along with the required argument <agr2-name> for the constraint <constr-name>:

[WARNING] '<arg1-name>' argument for '<constr-name>' constraint should be specified with '<arg2-name>' argument, while running rule LPSVM59.

Where,

- <agr1-name> is sleepval, saveval, or restoreval.
- <agr2-name> is sleepnet, savenet, restorenet, sleep, save, or restore.
- <constr-name> is *voltage_domain* or *retention_cell*.

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

Though the SpyGlass Power Verify solution will continue with its analysis, the results may not be as expected.

How to Debug and Fix

The constraint reported in the violation message is highlighted in the Atrenta Console GUI.

To fix this violation, ensure that you specify the correct pair of arguments as stated in the violation message.

Example Code and/or Schematic

For the following snippet, the SGDC_lowpower97 rule reports a violation because the sleepval argument is not specified.

```
current_design top
...
voltage_domain -name VD2 -value 1.5
voltage_domain -instname P1 -sleepnet SL1
```

To fix the violation, specify the sleepval argument as shown in the

following snippet:

```
current_design top
...
voltage_domain -name VD2 -value 1.5
voltage_domain -instname P1 -sleepnet SL1 -sleepval 1
```

Default Severity Label

Warning

Rule Group

Constraint Checking Rules

Reports and Related Files

None

SGDC_lowpower98

Reports incorrectly specified pin in the –pin argument of input_isocell

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower98* rule reports a violation when the pin specified in the `-pin` argument does not match the pins of the cell specified in the `-names` argument of the *input_isocell* constraint. In addition, this rule reports if the pin specified in the `-pin` argument of the *input_isocell* constraint is not an input or inout port of the cell.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *input_isocell* (Mandatory): Use this constraint to specify the isolation cells at the inputs of a power domain.

Messages and Suggested Fix

Message 1

The following message appears when the pin *<pin-name>* specified by using the `-names` argument of the `input_isocell` constraint does not exist in the description of the *<cell-name>* cell:

[FATAL] Pin '<pin-name>' given with '-names' argument of input_isocell constraint does not exist in the description of RTL module/ library cell '<cell-name>'.

Potential Issues

This violation appears if an invalid or a non-existent pin is specified in the –

pin argument.

Consequences of Not Fixing

If you do not fix this violation, the input_isocell constraint is ignored by the rules of the SpyGlass Power Verify solution.

How to Debug and Fix

To fix this violation, ensure that the pin specified in the –pin argument exists for the cells specified in the –names argument.

Message 2

The following message appears when the pin *<pin-name>* specified by using the -pin argument of the input_isocell constraint is not an input or inout port of the cell *<cellname>*:

[FATAL] Pin '<pin-name>'of cell '<cell-name>' specified using -pin argument of input_isocell constraint should be input/inout terminal.

Potential Issues

This violation appears if only the input and inout pins are allowed in the –pin argument.

Consequences of Not Fixing

If you do not fix this violation, the input_isocell constraint is ignored by the rules of the SpyGlass Power Verify solution.

How to Debug and Fix

To fix this violation, ensure that only input or inout pins are specified in the –pin argument.

Example Code and/or Schematic

Example

Suppose the cell HD65_LH_ISO0X has two input pins, A, EN, and one output pin Z. Since the EN1 pin does not exist for the cell, the *SGDC_lowpower98* rule reports a violation message (Message 1). Furthermore, the pin Z is an output pin and is not allowed in the –pin argument. Therefore, the rule again reports a violation message (Message 2).

Default Severity Label

Fatal

Rule Group

Constraints Checking

Reports and Related Files

None

SGDC_lowpower99

Reports invalid power domain specified by the -belongsto argument of isolation_cell

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower99* rule reports a fatal violation if the domain name specified in the -belongsto argument of the *isolation_cell* constraint is not a valid power domain.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *isolation_cell* (Mandatory): Use this constraint to specify the isolation cells in a power domain.

Messages and Suggested Fix

The following message appears when the domain name *<dom-name>* specified by using the -belongsto argument of the isolation_cell constraint is not a valid power domain:

[FATAL] Domain name '<dom-name>' given in -belongsto field of 'isolation_cell' constraint must be powerdomain

Potential Issues

This violation appears if you provide an invalid domain name in the –belongsto argument of the isolation_cell constraint.

Consequences of Not Fixing

If you do not fix the violation, the rules in the SpyGlass Power Verify

solution ignore the *isolation_cell* constraint.

How to Debug and Fix

To fix this violation, specify the valid power domain names by using the `-belongsto` argument of the `isolation_cell` constraint.

Example Code and/or Schematic

For the following code, the *SGDC_lowpower99* rule reports a violation because the voltage domain `V1` is not allowed in the `-belongsto` argument of the `isolation_cell` constraint.

```
voltage_domain -name V1 -value 1.2 -modname mid
isolation_cell -names "iso*" -belongsto V1
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_lowpower100

Reports incorrectly specified domain name in the power_down constraint

When to Use

This is an always ON rule.

Description

The rule *SGDC_lowpower100* reports violation, when the domain name stated in the `-domain` argument of a *power_down* constraint, is not a valid power domain.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *power_down* (Mandatory): Use this constraint to specify the powerdown conditions, as used by the LPSVM28 rule.

Messages and Suggested Fix

The following message appears when the domain name `<dom-name>` specified with `domain` argument of the *power_down* constraint is not a valid powerdomain:

[FATAL] Domain name '<dom-name>' given in '-domain' field of 'power_down' constraint must be powerdomain

Potential Issues

The SGDC command *power_down* is applicable to an off domain only.

Consequences of Not Fixing

The SGDC command would be ignored by rule LPSVM28.

How to Debug and Fix

Ensure that the domain name use is an off domain only.

Example Code and/or Schematic

In example below the line in BOLD is redundant and would be ignored as VD1 is an always-on domain. The SGDC constraint *power_down* is applicable on off domains only.

```
voltagedomain –name VD1 –value 1.0 –modname top . . .
voltagedomain –name VD2 –value 1.2 0 –instname top.U1 –isosig top.iso . . .
power_down –domain VD1 . . .
power_down –domain VD2 . . .
```

Default Severity Label

Fatal

Rule Group

VoltageConstraintCheck

Reports and Related Files

None

SGDC_lowpower101

Reports if the -vddcpin argument is not specified in retention_cell

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower101* rule reports a violation, if you do not specify the `-vddcpin` argument of the *retention_cell* constraint while running the *LPPLIB10* rule. It is mandatory to specify the `-vddcpin` argument of the *retention_cell* constraint when you run the *LPPLIB10* rule.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *retention_cell* (Mandatory): Use this constraint to specify the retention latch cells used by the *LPPLIB10* rule.

Messages and Suggested Fix

The following message appears when the `-vddcpin` argument of the *retention_cell* constraint is not specified when you run the *LPPLIB10* rule:

```
[WARNING] Field '-vddcpin' should be specified for
'retention_cell' constraint, when rule LPPLIB10 is run
```

Potential Issues

This violation appears if you do not specify the `-vddcpin` argument in the *retention_cell* constraint, when you run the *LPPLIB10* rule.

Consequences of Not Fixing

If you do not fix this violation, the *LPPLIB10* rule may work properly.

How to Debug and Fix

To fix this violation, specify the `-vddcpin` argument in the *retention_cell* constraint, when you run the *LPPLIB10* rule.

Example Code and/or Schematic

For the following constraint, the *SGDC_lowpower101* rule reports a violation because the `-vddcpin` argument is not provided.

```
retention_cell -name RET_DFF -domains V1
```

Default Severity Label

Warning

Rule Group

`VoltageConstraintCheck`

Reports and Related Files

None

SGDC_lowpower103

Checks for missing arguments in domain_outputs constraint

When to Use

This is a setup rule and runs by default.

Description

The *SGDC_lowpower103* rule reports a violation message when both arguments, `value` and `default`, are not specified for the *domain_outputs* constraint.

While using *domain_outputs* constraint, you must specify either the `value` or `default` argument.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *domain_outputs* (Mandatory): Use to specify the values of various signals under the steady-state condition.

Messages and Suggested Fix

The following message appears when you neither specify the `value` nor `default` arguments for the *domain_outputs* constraint:

```
[FATAL] For domain_outputs constraint either '-value' or'-default' must be specified
```

Potential Issues

The violation message appears if you do not specify both `value` and `default` arguments of the *domain_outputs* constraint.

Consequences of Not Fixing

If you do not fix the violation, the constraint is unusable.

How to Debug and Fix

To resolve this violation, ensure that either the `value` or `default` argument is specified.

Example Code and/or Schematic

Since both the value and default arguments are missing in the following snippet, the *SGDC_lowpower103* rule reports a violation.

```
domain_outputs -name opname
```

Default Severity Label

Fatal

Rule Group

VoltageConstraintCheck

Reports and Related Files

None

SGDC_lowpower104

Reports incorrectly specified VDDC pin in always_on_buffer

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower104* rule reports a violation if the VDDC pin name specified in the `-vddcpin` argument of the *always_on_buffer* constraint is not of the power type.

While using the *always_on_buffer* constraint, the pin name, specified by using the `-vddcpin` argument, should be of the power type. This rule reports a violation if you specify the ground or signal type pins in the constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *always_on_buffer* (Mandatory): Use this constraint to specify the always-on buffers.

Messages and Suggested Fix

The following message appears when the pin name *<pin-name>* specified by using the `-vddcpin` argument of the `always_on_buffer` constraint is not of the power type:

[FATAL] `Pin '<pin-name>' of cell '<cell-name>' given with '-vddcpin' argument in 'always_on_buffer' constraint should be of type power`

Potential Issues

This violation appears if the VDDC pin specified in the `–vddcpin`

argument of the `always_on_buffer` constraint is not of the power type.

Consequences of Not Fixing

The VDDC pins specified in the `–vddcpin` argument of the `always_on_buffer` constraint are used by the *LPPLIB06* and *LPPLIB11* rules. These rules expect the pins to be of the power type. If you do not fix this violation, these rules may produce incorrect results.

How to Debug and Fix

To fix this violation, ensure that the pin used in the `-vddcpin` argument should be of the power type.

Example Code and/or Schematic

For the following SGDC file, the *SGDC_lowpower104* rule reports violations because the pins A and Y are not of the `power` type.

```
current_design top

voltagedomain -name Vtop -value 1.0 -modname top  -supplyname VDDA

voltagedomain -name VA -value 1.0 0 -instname ua  -supplyname VDD VDDA -noisosig

powerswitch -name PWRSW  -pwroutpin VDDO -pwrinpin VDDI -en_inv_in ON

supply -name VDD -value 1.0 -parent VDDA -on ENA

supply -name VDDA -value 1.0 -alwayson 1

supply -name VDDX -value 1.0 -net ua.VDDX

aonbuffer -name BUFX2 -vddcpin A

aonbuffer -name AND2X2 -vddcpin Y
```

Default Severity Label

Fatal

Rule Group

`VoltageConstraintCheck`

Reports and Related Files

None

SGDC_lowpower105

Reports if the –names argument is not specified in isolation_cell

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower105* rule reports a violation if the `-names` argument is not specified in the *isolation_cell* constraint.

The `–names` argument in the *isolation_cell* constraint is mandatory for all the rules in the SpyGlass Power Verify solution, except the *LPSVM23* rule.

Rule Exception(s)

The *SGDC_lowpower105* rule does not report a violation if the *LPSVM23* rule is enabled.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *isolation_cell* (Mandatory): Use this constraint to specify isolation cells in a power domain.

Messages and Suggested Fix

The following message appears when you do not specify the `-names` argument in the `isolation_cell` constraint:

```
[FATAL] Field '-names' should be specified while using the isolation_cell constraint
```

Potential Issues

This violation appears when you do not specify the `–names` argument in the `isolation_cell` constraint.

Consequences of Not Fixing

If you do not fix this violation, no cell is specified as the isolation cell.

In addition, the other arguments of the `isolation_cell` constraint, such as `-iso_enable_val`, are not applicable to any cell.

How to Debug and Fix

To fix this violation, specify the `–names` argument in the `isolation_cell` constraint, if the *LPSVM23* rule is not to be run.

Example Code and/or Schematic

For the following SGDC file, the *SGDC_lowpower105* rule reports a violation because the `–names` argument is not specified in the `isolation_cell` constraint.

```
current_design mid

voltagedomain -name V1 -value 1.2 -modname mid

voltagedomain -name V2  -value 1.2 0 -instname mid.t1.inst4 -
isosig isosig -isoval 0

isocell -belongsto V2
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_lowpower107

Checks the existence of pins specified in the -inTerm, -outTerm, and -enableTerm arguments of level shifter

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower107* rule checks the existence of the pins specified in the `-inTerm`, `-outTerm`, and `-enableTerm` arguments of the *levelshifter* constraint. This rule reports a fatal violation in the following scenarios:

- The pin specified in the `-inTerm`, `-outTerm`, or `-enableTerm` argument is not a pin in the cell specified in the corresponding `-name` argument of the *levelshifter* constraint.
- The direction of the pin specified in the `-inTerm` or `-enableTerm` argument is not an input or inout port.
- The direction of the pin specified in the `-outTerm` argument is not an output or inout port.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *levelshifter* (Mandatory): Use this constraint to specify the names of the design units that are used as level shifters.

Messages and Suggested Fix

Message 1

The following message appears when the pin *<pin-name>*, specified by using the *<arg-1>* argument of the `levelshifter` constraint, is not a pin in the cell *<cell-name>*, specified in the `-name` argument of the

levelshifter constraint:

[FATAL] Pin <pin-name> given with '<arg-1>' argument of 'levelshifter' constraint does not exist in the description of RTL module/library cell '<cell-name>'

Where, *<arg-1>* is the -inTerm, -outTerm, or -enableTerm argument.

Potential Issues

This violation appears when the pin specified in the -interm, -outTerm, or -enableTerm argument of the levelshifter constraint is not a pin in the cell specified by using the –name argument of this constraint.

Consequences of Not Fixing

If you do not fix this violation, the rules using these pins of the level shifter cells may produce incorrect results.

How to Debug and Fix

To fix this violation, ensure that the pins specified in the –inTerm, -outTerm, and -enableTerm arguments of the levelshifter constraint are valid pins that exist for the cell mentioned in the –names argument of the same constraint.

Message 2

The following message appears when the direction of the pin *<pin-name>*, specified by using the *<arg-1>* argument of the levelshifter constraint, is not the *<type>* or inout port, respectively:

[FATAL] Pin '<pin-name>' of cell '<cell-name>' specified with '<arg-1>' argument of 'levelshifter' constraint should be <type>/inout terminal

Where,

<arg-1> is the -inTerm, -outTerm, or -enableTerm argument.

<type> is an input or output terminal.

Potential Issues

This violation appears in the following cases:

- The pin in the `–inTerm` or `-enableTerm` argument of the `levelshifter` constraint does not have an input or inout direction.
- The pin mentioned in the `–outTerm` argument of the `levelshifter` constraint does not have an output or inout direction.

Consequences of Not Fixing

If you do not fix this violation, the rules using this information may produce incorrect results.

How to Debug and Fix

To fix this violation, ensure that the pin specified in the `–inTerm` or `-enableTerm` argument is an input or inout pin only and the pin specified in the `–outTerm` argument is an output or inout pin only.

Example Code and/or Schematic

For the following SGDC file, the *SGDC_lowpower107* rule reports a violation because the `junk3` pin, specified in the `-inTerm` argument of the `levelshifter` constraint, does not exist in the description of the RTL module or the library cell `ls`.

```
current_design top

voltagedomain -name V1 -value 1.2 -modname top

voltagedomain -name V3 -value 1.8 -instname top.inst1

levelshifter -name ls -from V1 -to V3 -enableTerm en -inTerm
junk3 -outTerm junk4
```

Similarly, the rule reports a violation because the `junk4` pin, specified in the `-outTerm` argument of the `levelshifter` constraint, does not exist in the description of the RTL module or the library cell `ls`.

Default Severity Label

Fatal

Rule Group

VoltageConstraintCheck

Reports and Related Files

None

SGDC_lowpower108

Checks if pins specified with the arguments of isolation_cell exist in the design

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower108* rule checks the pins specified in the `input_pin`, `-output_pin`, and `-enable_pin` arguments of the *isolation_cell* constraint. This rule reports a fatal violation in the following scenarios:

- The pin specified in the `-input_pin`, `-output_pin`, or `-enable_pin` argument is not a pin in the cell specified in the corresponding `-name` argument of the *isolation_cell* constraint.
- The direction of the pin specified in the `-input_pin` or `-enable_pin` argument is not an input or inout port.
- The direction of the pin specified in the `-output_pin` argument is not an output or inout port.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *isolation_cell* (Mandatory): Use this constraint to specify the isolation cells in power domains.

Messages and Suggested Fix

Message 1

The following message appears when the pin *<pin-name>*, specified by using the *<arg-1>* argument of the `isolation_cell` constraint, is not a pin in the cell *<cell-name>*, specified in the `-name` argument of the

isolation_cell constraint:

[FATAL] Pin <pin-name> given with '<arg-1>' argument of 'isolation_cell' constraint does not exist in the description of RTL module/library cell '<cell-name>'

Where,

<arg-1> is the -input_pin, -output_pin, or -enable_pin argument.

Potential Issues

This violation appears when a pin that does not exist in the cell is defined in the –input_pin, -output_pin, or -enable_pin argument of the isolation_cell constraint.

Consequences of Not Fixing

If you do not fix this violation, the rules using these pins of isolation cells may produce incorrect results.

How to Debug and Fix

To fix this violation, ensure that the pins specified in the arguments of the isolation_cell constraint are valid pins that exist in the cell defined in the –names argument of the same constraint.

Message 2

The following message appears when the direction of the pin *<pin-name>*, specified by using *<arg-1>* argument of the isolation_cell constraint, is not the *<type>* or inout port, respectively:

[FATAL] Pin '<pin-name>' of cell '<cell-name>' specified with '<arg-1>' argument of 'isolation_cell' constraint should be <type>/inout terminal

Where,

<arg-1> is the -input_pin, -output_pin, or -enable_pin argument.

<type> is an input or output terminal.

Potential Issues

This violation appears in the following cases:

- The pin mentioned in the –input_pin or -enable_pin argument of the isolation_cell constraint does not have an input or inout direction.
- The pin mentioned in the –output_pin argument of the isolation_cell constraint does not have an output or inout direction.

Consequences of Not Fixing

If you do not fix this violation, the rules using this information may produce incorrect results.

How to Debug and Fix

To fix this violation, ensure that the pin specified in the –input_pin or -enable_pin argument is an input or inout pin only and the pin specified in the –output_pin argument is an output or inout pin only.

Example Code and/or Schematic

For the following SGDC file, the *SGDC_lowpower108* rule reports a violation because the junk1 and junk2 pins, specified in the -input_pin and –output_pin arguments, respectively, of the isolation_cell constraint, do not exist in the description of the RTL module or the library cell iso1.

```
current_design top

voltagedomain -name V1 -value 1.2 -modname top

voltagedomain -name V3 -value 1.8 -instname top.inst1

isocell -names iso1 -enable_pin out -input_pin junk1 -
output_pin junk2

isocell -names BASIC_LS_EN -enable_pin abc -input_pin Y -
output_pin A
```

The pin A of the BASIC_LS_EN cell, specified in the -output_pin argument of the *isolation_cell* constraint, should be an output or inout terminal.

Default Severity Label

Fatal

Rule Group

Constraints Checking

Reports and Related Files

None

SGDC_lowpower109

Reports when the number of values specified in the -signame and -value arguments of power_down are not same

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower109* rule reports a violation if the number of values specified in the `–signame` argument is not equal to the number of values specified in the `–value` argument of the *power_down* constraint. The `-signame` argument specifies the space-separated list of pin or net names that need to be checked. The `–value` argument specifies the space-separated list of expected values.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *power_down* (Mandatory): Use this constraint to specify the power down conditions.

Messages and Suggested Fix

The following message appears when the number of values specified in the `-signame` and `-value` arguments of the `power_down` constraint do not match:

[FATAL] `The number of arguments specified by '-signame' and '-value' field of power_down constraint does not match`

Potential Issues

This violation appears when the number of values in the `–signame` argument is not equal to the number of values in the `–value` argument of the `power_down` constraint.

Consequences of Not Fixing

If you do not fix this violation, the *LPSVM28* rule may produce incorrect results.

If the number of values in the `-signame` argument is more than the number of values in the `-value` argument, the expected value of that signal in the power down condition is not known.

If the number of values in the `-signame` argument is less than the number of values in the `-value` argument, the value taken as an expected value of that signal is not clear in the power down condition.

How to Debug and Fix

To fix this violation, ensure in the SGDC file that the number of values specified in the `-signame` and `-value` arguments of the `power_down` constraint is same.

Example Code and/or Schematic

For the following SGDC file, the *SGDC_lowpower109* rule reports a violation because the number of values in the `-signame` and `-value` arguments are not same.

```
current_design top

voltagedomain -name V1 -value 1.0 -modname top

voltagedomain -name V2 -value 1.2 0.0 -instname top.pd1 -
isosig top.iso -isoval 1

voltagedomain -name V3 -value 1.3 0.0 -instname top.iso1 -
noisosig

power_down -domain V2 -signame "top.w*[3:0]" "top.w1[2]" -
value 0
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_lowpower110

Checks to ensure retention_instance lies in the hierarchy of a power domain

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower110* rule reports a violation when the instance specified by using the `-name` argument of the *retention_instance* constraint is lying in the hierarchy of an always-on domain.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *retention_instance* (Mandatory): Use this constraint to specify the hierarchy information of the retention cell instances.

Messages and Suggested Fix

The following message appears when the retention instance <*rtn-inst*>, specified by using the `retention_instance` constraint, is lying in the hierarchy of an always-on domain <*dom-name*>:

[FATAL] `Retain instance '<rtn-inst>' cannot be defined on the hierarchy of an always on domain '<dom-name>'`

Potential Issues

This violation appears when you specify an instance, which lies in the hierarchy of an always-on domain, in the `retention_instance` constraint.

Consequences of Not Fixing

If you do not fix this violation, the *LPSVM38* and *LPSVM58* rules, which use

the instance mentioned in the `retention_instance` constraint, may produce incorrect results.

How to Debug and Fix

To fix this violation, ensure that the `–name` argument of this constraint has only those instances that lie in the hierarchy of a power domain.

Example Code and/or Schematic

For the following SGDC file, the *SGDC_lowpower110* rule reports a violation because the retain instance `CKT_TOP.U1.U1` is defined in the hierarchy of the always-on domain V2:

```
current_design CKT_TOP

voltagedomain -name V1 -value 1.2 0 -modname CKT_TOP -
noisosig

voltagedomain -name V2 -value 1.3 -instname CKT_TOP.U1

retencell -name FD1S3AQV15 -domains V1

retain_instance -name CKT_TOP.U1.U1
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_lowpower112

Checks the -vddcpin and -vddpin arguments of retention_cell

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower112* rule reports a violation if the pin names, specified by using the `-vddcpin` and/or `-vddpin` arguments of the *retention_cell* constraint, are not of the power type. This rule reports a violation for the ground or signal type pins.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *retention_cell* (Mandatory): Use this constraint to specify the retention latch cells or SRPG cells.

Messages and Suggested Fix

The following message appears when the pin *<pin-name>* specified in the `–vddcpin` and/or `-vddpin` argument of the *retention_cell* constraint is not of the power type:

[FATAL] `Pin '<pin-name>' of cell '<cell-name>' given with '<arg>' argument in 'retention_cell' constraint should be of type power`

Potential Issues

This violation appears if a pin that is not of the power type is specified in the `–vddpin` and/or `–vddcpin` argument of the *retention_cell* constraint.

Consequences of Not Fixing

If you do not fix this violation, the *LPPLIB10* rule may produce incorrect results.

How to Debug and Fix

To fix this violation, ensure that the pins specified in the `–vddcpin` and/or `-vddpin` arguments of the *retention_cell* constraint are power pins.

Example Code and/or Schematic

For the following SGDC file, the *SGDC_lowpower112* rule reports a violation because the pin `A` of the cell `AND2X2`, specified in the `-vddcpin` argument of the *retention_cell* constraint, is not of the power type.

```
current_design top

voltagedomain -name Vtop -value 1.0 -modname top -supplyname VDDA

voltagedomain -name VA -value 1.0 0 -instname ua -supplyname VDD VDDA -noisosig

powerswitch -name PWRSW -pwroutpin VDDO -pwrinpin VDDI -en_inv_in ON

supply -name VDD -value 1.0 -parent VDDA -on ENA

supply -name VDDA -value 1.0 -alwayson 1

supply -name VDDX -value 1.0 -net ua.VDDX

retencell –name AND2X2 -vddcpin A -vddpin A
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_lowpower113

Reports missing inisosig argument of voltage_domain

When to Use

This rule is run by default when the *LPSVM47* rule is used.

Description

The *SGDC_lowpower113* rule checks whether the input isolation signal is specified with the `-inisosig` argument of the *voltage_domain* constraint when the *LPSVM47* rule is used.

Prerequisites

Run the *LPSVM47* rule.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain*: Use to specify the voltage/power domains in the design.

Messages and Suggested Fix

Message 1

The following message appears when the *LPSVM47* rule is used, but the `inisosig` argument of the *voltage_domain* constraint is not specified:

```
[WARNING] Input isolation signal should be specified with
'inisosig' argument for powerdomain '<pwr-dmn-name>' when rule
LPSVM47 is run
```

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the *LPSVM47* rule is used and the `inisosig` argument of the *voltage_domain* constraint is not specified, but

the isosig argument is specified. In this case, the -isosig argument is used:

```
[WARNING] Input isolation signal is not specified for
powerdomain '<pwr-dmn-name>', isolation signal specified with
'isosig' argument is used in rule LPSVM47 run
```

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

Though the SpyGlass Power Verify solution will continue with its analysis, the results may not be as expected.

How to Debug and Fix

The constraint reported in the violation message is highlighted in the Atrenta Console GUI.

To fix these violations, specify the inisosig argument of the *voltage_domain* constraint. If there are multiple input signals for the power domain, ensure that the signal names are space-separated.

Example Code and/or Schematic

For the following snippet, the *SGDC_lowpower113* rule reports a violation because the inisosig argument is not specified.

```
voltage_domain -name PD1 -value 1.2 0
-instname "TOP.lower1"
-isoval 1 -enableports 'EN1 1 EN2 100 EN3[2] 100'
```

To resolve the violation, specify the inisosig argument, as shown in the following snippet:

```
voltage_domain -name PD1 -value 1.2 0
-instname "TOP.lower1" -inisosig top.iso_sig
-isoval 1 -enableports 'EN1 1 EN2 100 EN3[2] 100'
```

Default Severity Label

Warning

Rule Group

Constraint Checking Rules

Reports and Related Files

None

SGDC_lowpower114

Reports power or ground bias nets of voltage_domain not specified as the power or ground supply

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower114* rule reports a violation if the power or ground bias nets, specified by using the `-biaspowernet` or `-biasgroundnet` arguments of the *voltage_domain* constraint, are not specified in the *supply* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage or power domains in the design.
- *supply* (Mandatory): Use this constraint to specify the supply and ground port names for all the LPPLIB rules.

Messages and Suggested Fix

Message 1

The following message appears when the power bias net *<net-name>* for the voltage domain *<domain-name>* is not specified as power supply in the `supply` constraint:

[FATAL] Bias power net '<net-name>' for voltagedomain '<domain-name>' is not specified as power supply through 'supply' constraint

Potential Issues

This violation appears when you do not specify the power net of a voltage domain as power supply by using the `supply` constraint.

Consequences of Not Fixing

If you do not fix the violation, the bias power net specified in the `voltage_domain` constraint serves no purpose.

How to Debug and Fix

To fix this violation, specify the bias power net of a voltage domain as power supply by using the `supply` constraint.

Message 2

The following message appears when the ground bias net *<net-name>* for the voltage domain *<domain-name>* is not specified as ground supply in the `supply` constraint:

[FATAL] `Bias power <SHOULD BE: ground instead of power> net '<net-name>' for voltagedomain '<domain-name>' is not specified as ground supply through 'supply' constraint`

Potential Issues

This violation appears when you do not specify the ground net of a voltage domain as ground supply by using the `supply` constraint.

Consequences of Not Fixing

If you do not fix the violation, the bias ground net specified in the `voltage_domain` constraint serves no purpose.

How to Debug and Fix

To fix this violation, specify the bias ground net of a voltage domain as ground supply by using the `supply` constraint.

Example Code and/or Schematic

For the following constraint, the *SGDC_lowpower114* rule reports a violation because the bias power net `VDDB` is not specified as power supply and the bias ground net `VSSB` is not specified as ground supply:

```
voltage_domain -name V2 -value 1.2 0 -instname top.mid_inst -isosig rst -isoval 0 -biaspowernet VDDB -biasgroundnet VSSB
```

To fix the violation, specify the `supply` constraint as follows:

```
supply -name VDDB -value 1.2
supply -name VSSB -value 0
```

Default Severity Label

Fatal

Rule Group

Constraints Checking

Reports and Related Files

None

SGDC_lowpower115

Reports when pins specified by using the arguments of *power_switch* are of improper types

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower115* rule reports a violation when:

- Pins specified in the `-pwrinpin` or `-pwroutpin` arguments are not of the power or ground type.
- Pins specified in the `-en_inv_in` or `-en_inv_in_2` arguments are not of the signal type.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *power_switch* (Mandatory): Use this constraint to specify power switches in the power domains.

Messages and Suggested Fix

Message 1

The following message appears when the pin *<pin-name>*, specified by using the *<arg-name>* argument of the *power_switch* constraint, is not of the proper *<pin-type>* power or ground type:

[FATAL] Pin '<pin-name>' of cell '<cell-name>' given with '<arg-name>' argument in power_switch constraint should be of type '<pin-type>'

Where, *<arg-name>* is the `–pwrinpin` or `-pwroutpin` argument

Potential Issues

This violation appears when the pin specified in the `-pwrinpin` or `-pwroutpin` arguments of the *power_switch* constraint is neither power nor ground type.

Consequences of Not Fixing

If you do not fix this violation, the *power_switch* constraint becomes unusable because it is mandatory to specify only power or ground pin in the `-pwrinpin or -pwroutpin` argument.

How to Debug and Fix

To fix this violation, ensure that the pin specified in the `-pwrinpin` or `-pwroutpin` argument of the *power_switch* constraint is either power or ground type.

Message 2

The following message appears when the pin *<pin-name>*, specified by using the *<arg-name>* argument of the *power_switch* constraint, is not of the signal type:

[FATAL] `Pin '<pin-name>' of cell '<cell-name>' given with '<arg-list>' argument in power_switch constraint should be of type 'signal'`

Where,

<arg-name> is the `-en_inv_in` or `-en_inv_in_2` argument.

Potential Issues

This violation appears when the pin specified in the `-en_inv_in or -en_inv_in_2` argument of the *power_switch* constraint is not of the signal type.

Consequences of Not Fixing

If you do not fix this violation, the *power_switch* constraint becomes unusable because it is mandatory to specify only signal pin in the `-en_inv_in` or `-en_inv_in_2` argument.

How to Debug and Fix

To fix this violation, ensure that the pin specified in the `-en_inv_in or -en_inv_in_2` argument of the *power_switch* constraint is of the signal

type.

Example Code and/or Schematic

For the following constraint, the *SGDC_lowpower115* rule reports a violation because the `Opin1` and `Ipin1` pins are not the power or ground pins. In addition, the `Ipin2` pin is not of the signal type.

```
power_switch -name PWRSW -pwroutpin Opin1 -pwrinpin Ipin1 -
en_inv_in Ipin2 -enableval 1
```

To fix this violation, specify the *power_switch* constraint as follows:

```
power_switch -name PWRSW -pwroutpin PWR -pwrinpin GND -
en_inv_in ON -enableval 1
```

In this case, the `PWR` and `GND` pins are power and ground pins of the power switch `PWRSW`. In addition, the `ON` pin is of the signal type.

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_lowpower116

Reports the missing -name and -value arguments of the voltage_domain constraint

When to Use

This is a setup rule and it runs by default.

Description

The *SGDC_lowpower116* rule reports a violation if either the `-name` or `-value` argument is missing in the *voltage_domain* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain* (Mandatory): Use this constraint to specify the voltage or power domains in the design. This information is used by the voltage and power domain rules of the SpyGlass Power Verify solution.

Messages and Suggested Fix

The following message appears when the *<arg-name>* argument is missing in the `voltage_domain` constraint:

[Fatal] The '<arg-name>' option should be specified in SGDC command 'voltage_domain'

Where,

<arg-name> is `-name` or `–value` argument.

Potential Issues

This violation appears if you specify the `voltage_domain` constraint without the `–name` and `-value` arguments.

Consequences of Not Fixing

If you do not fix this violation, the `voltage_domain` constraint becomes unusable because it is mandatory to specify the `-name` and `-value` arguments in the `voltage_domain` constraint.

How to Debug and Fix

To fix this violation, specify both the `-name` and `-value` arguments in the `voltage_domain` constraint.

Example Code and/or Schematic

For the following constraint, the *SGDC_lowpower116* rule reports a violation because the `–value` and `-name` arguments are missing in the `voltage_domain` constraint:

```
voltage_domain -name V3
voltage_domain -value 1.8
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_lowpower118

Performs the sanity check in the associate_lib constraint

When to Use

This is a setup rule and always runs by default.

Description

The SGDC_lowpower118 rule performs a sanity checking for following scenarios in the *associate_lib* constraint:

- If the domain name does not exist (not defined in the UPF)
- If the constraint is not specified for top-level domain
- If library name is not found in loaded physical libraries

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *associate_lib*

Messages and Suggested Fix

Message 1

The following message appears when domain specified in the *associate_lib* constraint is not defined in the UPF:

[FATAL] `Domain '<domain-name>' specified through -domain field of associate_lib constraint does not exist in UPF`

Message 2

The following message appears when library name specified in *associate_lib* constraint is not present in any `.lib` file:

[FATAL] `Library '<library_name>' specified through -lib field of associate_lib constraint is not valid`

Message 3

The following message appears when the *associate_lib* constraint has not been defined for top-level domain:

[FATAL] associate_lib constraint is missing for top domain <top-domain>

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

- Message 1: Provide a valid domain name that has been defined in the UPF.
- Message 2: Provide a valid library name that has been defined in the given `.lib` files.
- Message 3: Make sure the *associate_lib* constraint is always defined for the top-level domain.

Example Code and/or Schematic

Consider the following SGDC code.

```
current_design mid
power_data -format upf -file const.upf

associate_lib -domain UB  -lib "example" -cell BASIC_LS_1
associate_lib -domain UA  -lib "exampleY" -cell
BASIC_LS_1
associate_lib -domain UX  -lib "example"
```

Corresponding UPF file snippet is:

```
upf_version 2.0
set_design_top mid

create_power_domain top
```

```
create_power_domain UA -elements inst1
create_power_domain UB -elements inst2
```

For the above example, the following violation messages are reported:

```
Library 'exampleY' specified through -lib field of
associate_lib constraint is not valid


Domain 'UX' specified through -domain field of associate_lib
constraint does not exist in UPF


associate_lib constraint is missing for top domain 'top'
```

Default Severity Label

Fatal

Rule Group

Constraints Checking

Reports and Related Files

None

SGDC_lowpower119

Performs the sanity checking for the disallow_upf_command and make_mandatory_upf_commands_options SGDC constraints

When to Use

This is a setup rule and always runs by default.

Description

This rule performs the sanity checking for the *disallow_upf_command* and *make_mandatory_upf_commands_options* SGDC constraints. For example:

```
disallow_upf_command -name <command-name> -options
<argument-list>
```

```
make_mandatory_upf_commands_options -name <command-name> -
options <argument-list>
```

A violation is reported:

- If the command name specified is not a valid UPF 1.0/2.0 command.
- If arguments specified in `-options` argument is not a valid argument of the command given in the constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *disallow_upf_command*
- *make_mandatory_upf_commands_options*

Messages and Suggested Fix

Message 1

The following message appears when command specified is not a valid UPF 1.0/2.0 command:

[FATAL] Command '<command-name>' specified through -name field of <disallow_upf_command | make_mandatory_upf_commands_options> constraint is not a valid UPF command

Message 2

The following message appears when an argument specified in -options is not a valid argument of the command given in the constraint:

[FATAL] Option '<argument-name>' specified for command '<command-name>' through -options field of <disallow_upf_command | make_mandatory_upf_commands_options> constraint is not valid

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

- Message 1: Provide a valid UPF command.
- Message 2: Provide a valid argument in -options argument of the command given in the constraint.

Example Code and/or Schematic

Consider the following SGDC code.

```
disallow_upf_command -name create_pst1
disallow_upf_command -name set_isolation -options location
isolation_power_net_name
```

For the above example, the following violation messages are reported:

```
Command 'create_pst1' specified through -name field of
disallow_upf_command constraint is not a valid UPF command
```

```
Option 'isolation_power_net_name' specified for command
'set_isolation' through -options field of disallow_upf_command
constraint is not valid
```

Default Severity Label

Fatal

Rule Group

Constraints Checking

Reports and Related Files

None

SGDC_lowpower_RuleReq

Reports missing constraints required by the specified rules

When to Use

This is a setup rule and always runs by default.

Description

The *SGDC_lowpower_RuleReq* rule reports SpyGlass Design Constraints that are missing from the SpyGlass Design Constraints file. These constraints are required for the selected rules to run.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears after the *lp_cons_req* report is generated:

```
[WARNING] Some constraints are missing for module that are
required for running the specified rules. Please refer to
lp_cons_req report for details
```

Potential Issues

The violation messages explicitly states the potential issues.

Consequences of Not Fixing

The rules for which some constraints are missing are either not run or may not function as published.

How to Debug and Fix

Review the *lp_cons_req* report. You can access this report from the Reports menu in the Atrenta Console GUI.

Example Code and/or Schematic

Review the *lp_cons_req* report. You can access this report from the Reports menu in the Atrenta Console GUI.

Default Severity Label

Warning

Rule Group

`Constraints Checking`

Reports and Related Files

- *lp_cons_req:* This is the list of mandatory/optional constraints missed by rule.

SGDC_powerdomainoutputs01

Reports non-existent signal names specified through domain_outputs

When to Use

This is a setup rule and always runs by default.

Description

This rule reports the following cases:

- The object(s) specified with the `-value` argument does not exist as a top-level port, internal net, or instance terminal in the design unit specified as `current_design` of the *domain_outputs* constraint.
- The object name specified is not a hierarchical name, with respect to the top.
- Multiple object names are not space-separated and are specified with the `value` argument.
- The specified hierarchical name is not specifically enclosed in double quotes and contains special (escaped) characters.

Language

Verilog, VHDL

Parameters

None

Constraints

- *domain_outputs*: Use to specify the values of various signals under the steady-state condition specified in the voltage_domain constraint.

Messages and Suggested Fix

The following message appears for a *domain_outputs* constraint when the object <*name*> specified with the `value` argument does not exist in the environment as a top-level port, internal net, or instance terminal:

[FATAL] `'name'[TopPort + Net + HierTerminal] not found on/ within module 'mod'`

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

The signal is not present in design. The constraint will have no meaning if not fixed.

How to Debug and Fix

The constraint reported in the violation message is highlighted in the Atrenta Console GUI.

Update the `value` argument of the *domain_outputs* constraint such that the object exists in the environment as a top-level port, internal net, or instance terminal.

Example Code and/or Schematic

For the following snippet, the *SGDC_powerdomainoutputs01* rule reports a violation because the 'mid.t1.c3_int[1]' does not exist in design.

```
voltage_domain -name V3  -value 1.3 0 -instname mid.t1.inst5
-isosig mid.t1.isosig1 -isoval 1 -generate_iso_logic -
outputs OUT1

domain_outputs -name OUT -value mid.t1.c3_int[1] 1
```

To resolve the violation, specify `mid.t1.c2_int[1]` which exists in design.

Default and Severity Label

Fatal

Rule group

Constraints Checking

Reports and Related Files

None

SGDC_powerdomainoutputs02

Reports incorrect value specification in domain_outputs

When to Use

This is a setup rule and always runs by default.

Description

The *SGDC_powerdomainoutputs02* rule reports a violation when the `-value` argument of the *domain_outputs* constraint is set to a value other than `0`, `1`, or `2`. Signals specified with the `-value` argument of the *domain_outputs* constraint with valid signal values are as follows:

- `0`: To isolate to steady state value of `0`
- `1`: To isolate to steady state value of `1`
- `2`: To retain value before isolation

Language

Verilog, VHDL

Parameters

None

Constraints

- *domain_outputs*: Use to specify the values of various signals under the steady-state condition specified in the voltage_domain constraint.

Messages and Suggested Fix

The following message appears for a *domain_outputs* constraint where the signal *<signal-name>* specified with the `value` argument has an associated value other than `0`, `1`, and `2`:

[WARNING] signal '<signal-name>' used in 'domain_outputs' constraint is defined with a value other than 0/1/2

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

Only values 0, 1, and 2 are acceptable. Any other value will make the constraint redundant.

How to Debug and Fix

The constraint reported in the violation message is highlighted in the Atrenta Console GUI.

Update the `value` argument of the *domain_outputs* constraint so that the value is either 0, 1, or 2.

Example Code and/or Schematic

For the following snippet, the *SGDC_powerdomainoutputs02* rule reports a violation because the `-value` argument has a value other than 0, 1, or 2.

```
voltage_domain -name V3 -value 1.2 0 -instname mc_top.u2 -
isosig susp_req_i -isoval 1 -outputs PD_V3_OUT

domain_outputs -name PD_V3_OUT -value obct_cs 11
```

To resolve the violation, specify either 0, 1, or 2 in the `-value` argument, as shown in the following snippet:

```
voltage_domain -name V3 -value 1.2 0 -instname mc_top.u2 -
isosig susp_req_i -isoval 1 -outputs PD_V3_OUT

domain_outputs -name PD_V3_OUT -value obct_cs 1
```

Default and Severity Label

Warning

Rule group

`Constraints Checking`

Reports and Related Files

None

SGDC_supply01

Reports supply constraint defined with a negative value

When to Use

This is a setup rule and always runs by default.

Description

The *SGDC_supply01* rule reports missing or incorrect value specified with the `value` argument of the *supply* constraint.

Prerequisite

The *SGDC_supply01* rule requires that the value specified with the `value` argument of the *supply* constraint is a positive floating point value or zero.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *supply* (Mandatory): Use to specify the value of the power and ground rails. The `-value` argument of the *supply* constraint should have a positive floating-point value for power rails and a zero for ground rails.

Messages and Suggested Fix

The following message appears for the *supply* constraint when the value specified with the `-value` argument is not a positive floating point value or zero:

[ERROR] `Supply Constraint should not be defined with a negative value`

Potential Issues

This violation message explicitly states the potential issues.

Consequences of Not Fixing

A supply value cannot be negative. Therefore, the constraint becomes redundant.

How to Debug and Fix

The constraint reported in the violation message is highlighted in the Atrenta Console GUI.

To fix this violation, ensure that the `value` argument of the *supply* constraint has either a positive floating point value or zero.

Example Code and/or Schematic

For the following snippet, the *SGDC_supply01* rule reports a violation because the value argument has a negative value.

```
supply –name SUP1 –value -2
```

To resolve the violation, set the value argument to a positive floating point value or zero, as shown in the following snippet:

```
supply –name SUP1 –value 2
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_voltagedomain01

Reports non-existent instances

When to Use

This is a setup rule and always runs by default.

Description

The *SGDC_voltagedomain01* rule reports instances that do not exist, but have been specified in the `instname` argument of the *voltage_domain* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Ma*ndatory)*

Messages and Suggested Fix

The following message appears for a *voltage_domain* constraint when the instance *<inst-name>* specified with the `instname` argument does not exist (indicated by *[Instance]*) in the environment *<top-du-name>*:

[FATAL] <inst-name>'[Instance] not found on/within module '<top-du-name>

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

As the instance provided does not exist in the design, the constraint has no use.

How to Debug and Fix

The *voltage_domain* SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix this violation, ensure:

- The instance name specified in the `-instname` argument exists in the design unit specified in `current_design` of the *voltage_domain* constraint.
- The instance name specified in the `instname` argument is a hierarchical instance name, with respect to the top.
- Multiple instance names that have been specified with the `instname` argument have instance names that are separated by white spaces.
- The hierarchical instance name does not contain special, such as escaped, characters. If it does, make sure the characters are enclosed in double quotes.

Example Code and/or Schematic

For the following hierarchical instance name, the *SGDC_voltagedomain01* rule reports a violation because it contains escaped characters, such as / and a white space.

```
\DOM_BUF_RING_INP_is_duc_PD_TOP_d1[13
```

To fix the violation, specify the hierarchical instance name in double quotes, as shown in the following.

```
“\DOM_BUF_RING_INP_is_duc_PD_TOP_d1[13”
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_voltagedomain02

Reports non-existent modname

When to Use

This is a setup rule and always runs by default.

Description

The *SGDC_voltagedomain02* rule reports design units that do not exists, but have been specified with the *modname* argument of the *voltage_domain* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain (Mandatory)*

Messages and Suggested Fix

The following message appears for a *voltage_domain* constraint when the design unit *<du-name>* specified with the modname argument has not been instantiated (indicated by *[SubModule]*) in the environment *<top-du-name>*:

[FATAL] <du-name>'[SubModule] is never instantiated within environment '<top-du-name>

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

As the sub-module does not exist in the design, this constraint is irrelevant.

How to Debug and Fix

The *voltage_domain* SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix this violation, ensure:

- The design unit specified with the `modname` argument is instantiated in the design unit specified as `current_design` of the *voltage_domain* constraint.
- The design unit name does not contain special, such as escaped, characters. If it does, make sure the characters are enclosed in double quotes.

Example Code and/or Schematic

For the following design unit name, the *SGDC_voltagedomain02* rule reports a violation because it contains escaped characters, such as / and a white space.

```
\DO_d1[13
```

To fix the violation, specify the design unit name in double quotes, as shown in the following.

```
“\DO _d1[13 ”
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

No related reports or files.

SGDC_voltagedomain03

Reports non-existent port name

When to Use

This is a setup rule and always runs by default.

Description

The *SGDC_voltagedomain03* rule reports ports that do not exist, but have been specified with the `portname` argument of the *voltage_domain* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain* (Mandatory)

Messages and Suggested Fix

The following message appears for a *voltage_domain* constraint when the port *<port-name>* specified with the `portname` argument does not exist (indicated by *[TopPort]*) in the environment *<top-du-name>*:

[FATAL] <port-name>'[TopPort] not found on/within module '<top-du-name>

Potential Issues

This violation appears because the ports specified with the `portname` argument must exist in the design unit specified as `current_design` of the *voltage_domain* constraint.

Consequences of Not Fixing

The port specified is non-existent in the design, therefore this constraint has no use.

How to Debug and Fix

The *voltage_domain* SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix this violation, ensure that the port specified with the `portname` argument exists in the design unit.

Example Code and/or Schematic

For the following port name, the *SGDC_voltagedomain03* rule reports a violation because the specified port doesn't exist in the design.

```
Top.P1
```

To fix the violation, specify the port name which exists in the design.

```
Top.In1
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_voltagedomain04

Reports non-existent signals specified in the isosig field

When to Use

This is a setup rule and always runs by default.

Description

The *SGDC_voltagedomain04* rule reports signals that do not exist or are incorrect, but have been specified in the `isosig` argument of the *voltage_domain* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain* (Mandatory)

Messages and Suggested Fix

The following message appears for a *voltage_domain* constraint when the object <*name*> specified with the `isosig` argument does not exist in the environment <*top-du-name*> as a top-level port, internal net, or instance terminal (indicated by`[TopPort + Net + HierTerminal]`):

[FATAL]<name>'[TopPort + Net + HierTerminal] not found on/ within module '<top-du-name>'

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

As the signal does not exist in the design, it is irrelevant to specify it. It serves no purpose in the constraint.

How to Debug and Fix

The *voltage_domain* SGDC constraint reported is highlighted in the Atrenta

Console GUI.

To fix this violation, ensure:

- The object(s) specified with the `-isosig` argument exist as a top-level port, internal net, or instance terminal in the design unit specified in `current_design` of the *voltage_domain* constraint.
- The object name specified is not a hierarchical name, with respect to the top.
- Multiple object names that have been specified with the `isosig` argument have object names that are space separated.
- The hierarchical instance name does not contain special, such as escaped, characters. If it does, make sure the characters are enclosed in double quotes.

Example Code and/or Schematic

For the following hierarchical instance name, the *SGDC_voltagedomain04* rule reports a violation because it contains escaped characters, such as / and a white space.

```
\DOM_BUF_RING_INP_is_duc_PD_TOP_d1[13
```

To fix the violation, specify the hierarchical instance name in double quotes, as shown in the following.

```
“\DOM_BUF_RING_INP_is_duc_PD_TOP_d1[13”
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_voltagedomain05

Reports incorrect isoval specification

When to Use

This is a setup rule and always runs by default.

Description

The *SGDC_voltagedomain05* rule reports a violation when unsupported values are specified in the `isoval` argument of the *voltage_domain* constraints.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Mandatory)

Messages and Suggested Fix

The following message appears for a *voltage_domain* constraint where a value other than 0 or 1 has been specified with the `isoval` argument:

[FATAL] `IsosigVal should be 0 or 1`

Potential Issues

The violation appears because the `isoval` argument contains unsupported values.

Consequences of Not Fixing

As only values 0 or 1 should be specified, specifying any other value makes the constraint redundant.

How to Debug and Fix

The *voltage_domain* SGDC constraint reported is highlighted in the Atrenta

Console GUI. Review the values of the `isoval` argument and ensure that it contains either 0 or 1.

Example Code and/or Schematic

For the following snippet, the *SGDC_voltagedomain05* rule reports a violation because the isoval argument contains an unsupported value, 123.

```
current_design top
...
voltage_domain -name PD1 -value 1.2 0
-instname "top.\lower2 "
-isosig top.iso_sig -isoval 123
```

To resolve the violation, specify a supported value, 0 or 1, for the isoval argument, as shown in the following.

```
 current_design top
...
voltage_domain -name PD1 -value 1.2 0
-instname "top.\lower2 "
-isosig top.iso_sig -isoval 1
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_voltagedomain06

Reports incorrect value specification

When to Use

This is a setup rule and always runs by default.

Description

The *SGDC_voltagedomain06* rule reports a violation message when unsupported values are specified in the `value` argument of the *voltage_domain* constraints.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *voltage_domain* (Mandatory)

Messages and Suggested Fix

The following message appears for a *voltage_domain* constraint defining a voltage domain where the value specified with the `value` argument is not a positive non-zero floating-point value:

[FATAL] `Voltage Domain should be defined with a non-zero positive value`

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

As the voltage of a voltage domain provided is a negative value, the constraint is not used.

How to Debug and Fix

The *voltage_domain* SGDC constraint reported is highlighted in the Atrenta Console GUI. Review the values of the `value` argument and ensure that it contains a positive non-zero floating-point value.

Example Code and/or Schematic

For the following snippet, the *SGDC_voltagedomain06* rule reports a violation because the value argument contains an unsupported value, 0, for a voltage domain.

```
current_design top
...
voltage_domain -name VD1 -value 0
-instname "top.\lower2 "
-isosig top.iso_sig -isoval 1
```

To resolve the violation, specify a positive non-zero floating-point value for the value argument, as shown in the following.

```
current_design top
...
voltage_domain -name VD1 -value 1.2
-instname "top.\lower2 "
-isosig top.iso_sig -isoval 1
```

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

SGDC_voltagedomain07

Reports incorrect value specification for power domain

When to Use

This is a setup rule and always runs by default.

Description

The *SGDC_voltagedomain07* rule reports a violation when an unsupported OFF value is specified with the *voltage_domain* constraints while defining a power domain.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears for a *voltage_domain* constraint that defines a power domain when the second value (OFF value) is not zero:

[FATAL] `OFF voltage for Power Domain definition should be 0`

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears for a *voltage_domain* constraint that defines a power domain when more than two values have been specified:

[FATAL] `Multiple OFF values specified for the Power Domain`

Potential Issues

Message 1 appears because the second value, OFF value, specified with the `-value` argument of a *voltage_domain* constraint that is defining a power

domain in a non-zero value.

Message 2 appears because more than two values have been specified in the value argument.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The *voltage_domain* SGDC constraint reported is highlighted in the Atrenta Console GUI. Review the values of the `value` argument and ensure that it contains only two values. The second value represents the OFF value and this must be set to zero.

Example Code and/or Schematic

For the following snippet, the *SGDC_voltagedomain07* rule reports a violation because the value argument contains an unsupported OFF value, 1, for a power domain.

```
current_design top
...
voltage_domain -name PD1 -value 1.2 1
-instname "top.\lower2 "
-isosig top.iso_sig -isoval 1
```

To resolve the violation, specify zero OFF value for the value argument, as shown in the following.

```
current_design top
...
voltage_domain -name PD1 -value 1.2 0
-instname "top.\lower2 "
-isosig top.iso_sig -isoval 1
```

Default Severity Label

Fatal

Rule Group

Constraints Checking

Reports and Related Files

None

SGDC_voltagedomain08

Performs existence check of the clock signal specified by the clkdomain argument of voltage_domain

When to Use

This is a setup rule and always runs by default.

Description

The *SGDC_voltagedomain08* rule reports non-existent objects specified in the `clkdomain` argument of the *voltage_domain* constraint.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

- *voltage_domain* (Mandatory)

Messages and Suggested Fix

The following message appears for a *voltage_domain* constraint when the object *<name>* specified with the `clkdomain` argument does not exist in the environment *<top-du-name>* as a top-level port, internal net, or instance terminal (indicated by *[TopPort + Net + HierTerminal]*):

[FATAL]'<name>'[TopPort + Net + HierTerminal] not found on/ within module '<top-du-name>'

Potential Issues

This violation appears explicitly states the potential issues.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The SGDC constraint reported is highlighted in the Atrenta Console GUI.

To fix this violation, review the clkdomain argument and ensure:

- The object(s) specified with the `clkdomain` argument exist as a top-level port, internal net, or instance terminal in the design unit specified as `current_design` of the *voltage_domain* constraint.
- The object name specified is a hierarchical name, with respect to the top.
- When multiple object names have been specified with the `clkdomain` argument, the object names are space separated.
- The hierarchical instance name does not contain special, such as escaped, characters. If it does, make sure the characters are enclosed in double quotes.

Example Code and/or Schematic

For the following snippet, the *SGDC_voltagedomain08* rule reports a violation because the clock `clk1` is inexistent in the design.

```
voltage_domain -name PD1 -value 1 0 -instname top.inst -
clkdomain clk1
```

To resolve this issue, make sure a valid clock `clk1` exists in the design specified with `-clock` constraint in the SGDC file.

Default Severity Label

Fatal

Rule Group

`Constraints Checking`

Reports and Related Files

None

Electrical Checks Rules

The `LP_ERC_Checks` rules are as follows:

Rule	Reports
LPERC01	Checks incompatibility between supply nets of driver and receiver
LPERC01A	Checks if the supply net of the driver is working at the same voltage as the supply net of the receiver
LPERC01B	Checks if supply net of driver is less-on than the supply net of receiver
LPERC01C	Checks if supply net of driver is working at different voltage than supply net of receiver and is less on than receiver supply net
LPERC02	Checks incompatibility between input/output supply for power switch and level shifters
LPERC02A	Checks if the input supply net of the power switch is operating at a different voltage than the output supply net
LPERC02B	Checks if input supply net of level-shifter or power switch is less-on than the output supply net
LPERC03	Checks incompatibility between backup power and primary power of a library cell
LPERC03A	Checks relationship between supply nets connected to backup_power pin and primary_power pin
LPERC04	Checks for connectivity in supply and signal pins of library cells in design
LPERC04A	Checks power/ground supply pin connection with supply nets
LPERC04B	Checks if driver is an unconnected net, hanging net or is a supply net undefined in power intent
LPERC05	Reports power/ground nets that are connected to multiple inout/output power/ground pins
LPERC06	Reports Analog signals which are driving or getting driven by digital logic

LPERC01

Checks incompatibility between supply nets of driver and receiver

When to Use

- Check for incompatibility between the supply nets of driver pins/ports and supply nets of receiver pins/ports of library cells present in the design.
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*.
- *DEF files and their associated LEF files*.

Description

The *LPERC01* rule checks the incompatibility between supply nets of a driver and receiver.

This rule has the following subrules:

- *LPERC01A*: Checks if supply net of the driver is operating at a different voltage than the supply net of the receiver.
- *LPERC01B*: Checks if supply net of the driver is less-on than the supply net of the receiver.
- *LPERC01C*: Checks if supply net of driver is working at different voltage than supply net of receiver and is less on than receiver supply net.

LPERC01A

Checks if supply net of the driver is operating at a different voltage than the supply net of the receiver

When to Use

Check for incompatibility between the supply nets of driver pins/ports and supply nets of receiver pins/ports of library cells present in the design.

Description

The *LPERC01A* rule reports a violation message for a pin of a library cell or a TOP-LEVEL port when the supply net of the driver does not have the same voltage value as that of the supply net of the receiver.

Parameter(s)

- *lp_check_src_driver_supply_name*: Default is `0` and this rule does not check if the supply nets of driver and receiver are the same. Set the value to `1` or `yes` to report a violation when the supply nets of driver and receiver are different.
- *lp_flag_violation_on_antenna_cell*: Default is value `no`. Set the value of this parameter to `yes` to perform rule checking on antenna cells.

Constraint(s)

SGDC

- *lp_ignore_cells_for_erc* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *create_pst* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when there is a voltage difference between the supply net of a driver and the supply net of a receiver:

[LPERC01A_1][ERROR] Incompatible Supply Nets: Driver supply net at different voltage compared to receiver supply net - <power|ground> supply net <source_supply_net> [<vdd_source_value>] of driver <driver_name> is operating at different voltage w.r.t. to <power|ground> supply net <vdd_sink>[<vdd_sink_value>] of receiver <receiver_name> of instance <instance_name>

Where, *driver_name* and *receiver_name* are:

- **Port**: port <hierarchal_name_of_port>
- **Pin**: pin '<pin_name>' of instance 'hierarchal_ instance_name' (Cell: <cell_name>)

Message 2

The following message appears when the supply nets of driver and receiver are different:

[LPERC01A_2][ERROR] Incompatible Supply Nets : Driver supply net is different compared to receiver supply net - <power|ground> supply net <source_supply_net> of driver <driver_name> is different compared to <power|ground> supply net <sink_supply_net> of receiver <receiver_name> of instance <instance_name>

Where, *driver_name* and *receiver_name* are:

- **Port**: port <hierarchal_name_of_port>
- **Pin**: pin '<pin_name>' of instance 'hierarchal_ instance_name' (Cell: <cell_name>)

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not fixing

Message 1: Voltage difference between driver supply net and receiver supply net can lead to a design failure through level shifter requirement.

Message 2: If the driver supply and receiver supply are different, it may lead to a design failure.

How to debug and fix

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix the violation, update the UPF file and ensure that the supply net of the driver and the receiver do not have voltage difference.

Example Code and/or Schematic

Example 1

For the following UPF file snippet, the *LPERC01A* rule reports violation Message 1, because the VDD and VDD1 are working at different voltages.

```
create_supply_port VDD
add_port_state VDD -state {VT 1.2}
create_supply_net VDD -domain TOP
connect_supply_net VDD -ports VDD

create_supply_port VDD1
add_port_state VDD1 -state {VT 1.0}
create_supply_net VDD1 -domain TOP

connect_supply_net VDD1 -ports VDD1
```

The following violation message is reported.

```
Incompatible Supply Nets : Driver supply net at different
voltage compared to receiver supply net - power supply net
VDD['1.200'] of driver 'Y' of instance 'top.buf1'(Cell:BUF) is
operating at different voltage w.r.t. to power supply net
VDD1['1.000'] of receiver 'A' of instance
'top.inst1.and1'(Cell:AND)
```

The following schematic is generated.

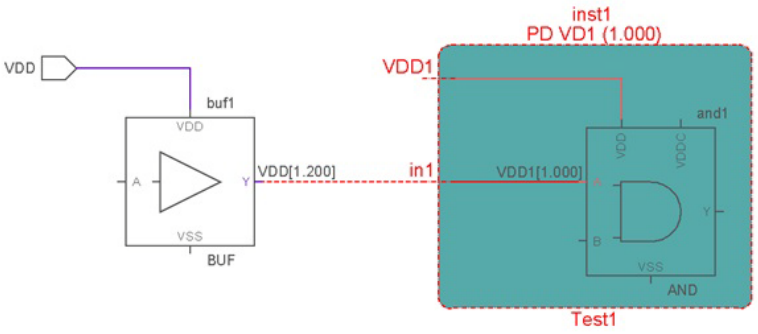


FIGURE 198. Incremental schematic

To resolve the violation, update the UPF file to ensure that VDD and VDD1 are working at the same voltage.

Example 2

For the following UPF file snippet, the *LPERC01A* rule reports the violation Message 2, because the VDD and VDD1 are different.

```
create_supply_port VDD
add_port_state VDD -state {VT 1.2}
create_supply_net VDD -domain TOP
connect_supply_net VDD -ports VDD

create_supply_port VDD1
add_port_state VDD1 -state {VT 1.0}
create_supply_net VDD1 -domain TOP

connect_supply_net VDD1 -ports VDD1
```

The following violation message is reported.

```
Incompatible Supply Nets : Driver supply net is different
compared to receiver supply net - power supply net VDD['1.200']
of driver 'Y' of instance 'top.buf1'(Cell:BUF) is different
compared to power supply net VDD1['1.000'] of receiver 'A' of
instance 'top.inst1.and1'(Cell:AND)
```

The following schematic is generated.

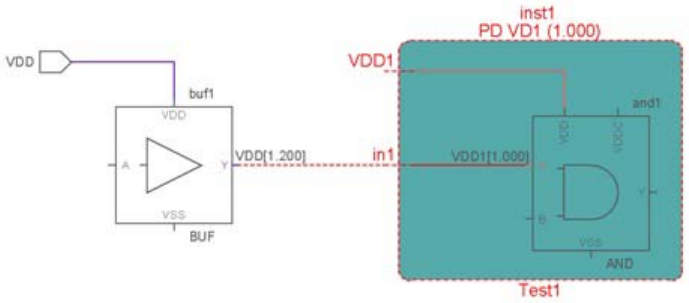


FIGURE 199. Incremental schematic

To resolve the violation, update the UPF file to ensure that `VDD` and `VDD1` are the same and the `lp_check_src_driver_supply_name` parameter should be set to `1`.

Default and Severity label

Error

Rule Group

`LP_ERC_Checks`

Reports and Related files

None

LPERC01B

Checks if supply net of the driver is less-on than the supply net of the receiver

When to Use

Check for incompatibility between the supply nets of driver pins/ports and supply nets of receiver pins/ports of library cells present in the design.

Description

The *LPERC01B* rule reports a violation message for an input/output pin of a library cell or a TOP-LEVEL input/output port when the supply net of the driver is less-on than the supply net of the receiver.

Rule Exceptions

This rule does not report violation messages for the:

- data pin of an Isocell.

Parameter(s)

- *lp_flag_violation_on_antenna_cell*: Default value is `no`. Set the value of this parameter to `yes` to perform rule checking on antenna cells.
- *lp_check_data_pin_of_els*:Default value is `no`. Set this parameter to `yes` to check data pins of the enable level shifter cells.

Constraint(s)

SGDC

- *lp_ignore_cells_for_erc* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)h
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)

- *create_pst* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the supply net of the driver is less-on than the supply net of the receiver:

[LPERC01B_1][ERROR] Incompatible Supply Nets: Driver supply net less-on than receiver supply net - <power|ground> supply net <source-supply-net> [<vdd_source_value>] of driver <driver-name> is less-on w.r.t. to <power|ground> supply net <vdd-sink>[<vdd-sink-value>] of receiver <receiver-name> of instance <instance-name>

Where, *driver_name* and *receiver_name* are:

- **Port**: port <hierarchal_name_of_port>
- **Pin**: pin '<pin_name>' of instance 'hierarchal_ instance_name' (Cell: <cell_name>)

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not fixing

SpyGlass requires you to resolve this violation. If you do not resolve this violation, SpyGlass will exit.

How to debug and fix

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, update the UPF file and ensure that the driver supply net is not less always-on than the receiver supply net.

Message 2

The following message appears when the supply of a level shifter or enabled level shifter (ELS) cell is off while both source and destination domains are on:

[LPERC01B_2][ERROR] Incompatible Supply Nets : Input supply of LS (or ELS) is OFF when both source and destination are ON -

```
Level Shifter instance <level-shifter-name> (Cell: '<level-
shifter-cell>') has input <power|ground> supply '<level-
shifter-supply>' is less-on w.r.t. to <power|ground> supply
'<source-supply-net> [<source-supply-net>]' of driver '<driver-
name>' and <power|ground> supply '<vdd-sink>[<vdd-sink-value>]'
of receiver '<receiver-name>'
```

Where, *driver_name* and *receiver_name* are:

- **Port**: port <hierarchal_name_of_port>
- **Pin**: pin '<pin_name>' of instance 'hierarchal_ instance_name' (Cell: <cell_name>)

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not fixing

If the supply of level shifter or enabled level shifter cell is off when source and destination are on, the cell does not serve its purpose. Level shifting does not take place and design does not function properly.

How to debug and fix

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button.

To fix this violation, check supply of source domain, destination domain and level shifter cell. Also, check the power states of the supply of level shifter cell. Update the UPF such that level shifter/enabled level shifter cell is on when both source and destination are on.

Example Code and/or Schematic

For the following UPF file snippet, the *LPERC01B* rule reports a violation message because VDD is less always-on than VDD1.

```
create_supply_port VDD
add_port_state VDD -state {VT 1.2} -state {VT_OFF off}
create_supply_net VDD -domain TOP
connect_supply_net VDD -ports VDD

create_supply_port VDD1
add_port_state VDD1 -state {VT 1.2}
```

```
create_supply_net VDD1 -domain TOP
connect_supply_net VDD1 -ports VDD1

create_pst ps1 -supplies {VDD VDD1}
add_pst_state s2 -pst ps1 -state {VT_OFF VT}
```

The following violation message is reported.

```
Incompatible Supply Nets : Driver supply net less-on than
receiver supply net - power supply net VDD['1.200'] of driver
'Y' of instance 'top.buf1'(Cell:BUF) is less-on w.r.t. to power
supply net VDD1['1.200'] of receiver 'A' of instance
'top.inst1.and1'(Cell:AND)
```

The following schematic is generated.

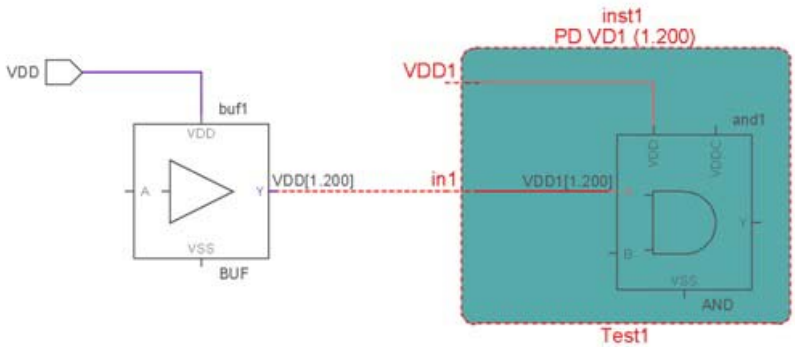


FIGURE 200. Incremental schematic

The schematic highlights the following information:

1. Source pin/port with supply information annotated.
2. Sink pin/port with supply information annotated.
3. Connection between point (1) and (2)
4. Supply net associated with source pin/port
5. Sink net associated with source pin/port

To resolve the violation, update the UPF file to ensure that VDD is more

always-on than VDD1.

Default and Severity label

Error

Rule Group

LP_ERC_Checks

Reports and Related files

None

LPERC01C

Checks if supply net of driver is different or working at different voltage than supply net of receiver and is less on than receiver supply net

When to Use

Check for a crossing where supply net of driver is less-on and is working at different voltage than supply net of receiver.

Description

The *LPERC01C* rule reports a violation message for an input signal pin of a library cell or a TOP-LEVEL output port when the supply net of the driver is less-on than the supply net of the receiver and is working at different voltage than supply net of receiver.

Parameter(s)

- *lp_check_src_driver_supply_name*: Default value is `0` and this rule does not check if the supply nets of driver and receiver are the same. Set the value to `1` or `yes` to report a violation when the supply nets of driver and receiver are different.
- *lp_flag_violation_on_antenna_cell*: Default value is `no`. Set the value of this parameter to `yes` to perform rule checking on antenna cells.

Constraint(s)

SGDC

- *lp_ignore_cells_for_erc* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)

- *create_pst* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the supply net of the driver is less-on than the supply net of the receiver and supply net of the driver and supply net of the receiver are operating at different voltages:

[LPERC01C_1][ERROR] Incompatible Supply Nets : Driver supply net is less-on and at different voltage compared to receiver supply net -<power|ground> supply net<vdd-source> of driver <driver-Name> is operating at different voltage w.r.t. to <power\ground> supply net <vdd-sink> of receiver <receiver-Name>

Where, *driver_name* and *receiver_name* are:

- **Port**: port <hierarchal_name_of_port>
- **Pin**: pin '<pin_name>' of instance 'hierarchal_ instance_name' (Cell: <cell_name>)

Message 2

The following message appears when the supply net of the driver is less-on and is different than the supply net of the receiver:

[LPERC01C_2][ERROR] Incompatible Supply nets : Driver supply net is less-on and different compared to Receiver supply net --<power|ground> supply net <vdd-source> of driver <driver-Name> is different compared to -<power|ground> supply net<vdd-sink> of receiver <receiver-Name>

Where, *driver_name* and *receiver_name* are:

- **Port**: port <hierarchal_name_of_port>
- **Pin**: pin '<pin_name>' of instance 'hierarchal_ instance_name' (Cell: <cell_name>)

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not fixing

SpyGlass requires you to resolve this violation. If you do not resolve this

violation, SpyGlass will exit.

How to debug and fix

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation:

- Update the UPF file and ensure that the driver supply net is not less-on than the receiver supply net.
- Make sure if the *lp_check_src_driver_supply_name* parameter is set to `1`, then driver supply net and receiver supply net should be the same.
- Make sure if the *lp_check_src_driver_supply_name* parameter is set to `0` (default), then driver supply net voltage and receiver supply net voltages should be the same.

Example Code and/or Schematic

Example 1

For the following UPF file snippet, the LPERC01C rule reports a violation message because driver supply net VDTOP is less on than receiver supply net VDPD that is always on. Also these two supply nets have different voltages 1.2V and 1.4V.

```
create_supply_port VDTOP
add_port_state VDTOP -state {VT 1.2} -state {VT_off off}
create_supply_net  VDTOP -domain TOP
connect_supply_net VDTOP -ports VDTOP

create_supply_port VDPD
add_port_state VDPD -state {V2 1.4}
create_supply_net  VDPD -domain TOP
connect_supply_net VDPD -ports {VDPD}
```

The following violation message is reported.

```
Incompatible Supply Nets : Driver supply net is less-on and at
different voltage compared to receiver supply net - power
supply net VDTOP['1.200'] of driver 'top.in1' is operating at
different voltage w.r.t. to power supply net VDPD['1.400'] of
```

```
receiver 'P1' of instance 'top.u2_inst.A1'(Cell:BUS_PAD)
```

The following schematic is generated.

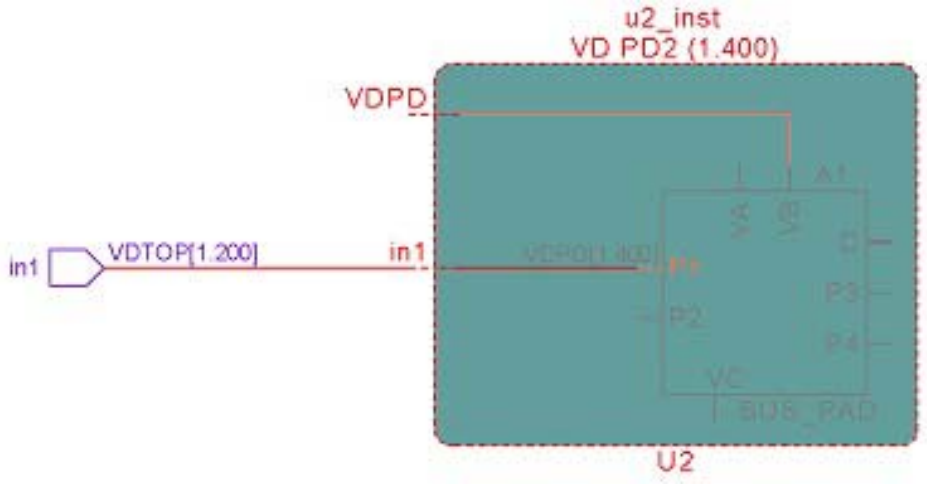


FIGURE 201. Incremental schematic

To resolve the violation, update the UPF file to ensure that VDTOP is not less-on than VDPD and the driver supply net voltage and receiver supply net voltages are the same.

Example 2

For the following UPF file snippet, the LPERC01C rule reports a violation message because the supply net VDTOP is less on than receiver supply net VDPD that is always on. Although, the supply nets have the same voltage 1.2 V, the *lp_check_src_driver_supply_name* parameter is set to 1. Therefore, the driver supply net (VDTOP) and receiver supply net (VDPD) are being checked. Since the following two conditions are getting true, the Message 2 gets reported:

- ❒ VDTOP is less on than VDPD
- ❒ The supply nets are different

```
create_supply_port VDTOP
add_port_state VDTOP -state {VT 1.2} -state {VT_off off}
create_supply_net  VDTOP -domain TOP
```

```
connect_supply_net VDTOP -ports VDTOP

create_supply_port VDPD
add_port_state VDPD -state {V2 1.2}
create_supply_net  VDPD -domain TOP
connect_supply_net VDPD -ports {VDPD}
```

The following violation message is reported.

```
Incompatible Supply Nets : Driver supply net is less-on and
different compared to receiver supply net - power supply net
VDTOP['1.200'] of driver 'top.in1' is different compared to
power supply net VDPD['1.200'] of receiver 'P1' of instance
'top.u2_inst.A1'(Cell:BUS_PAD)
```

The following schematic is generated.

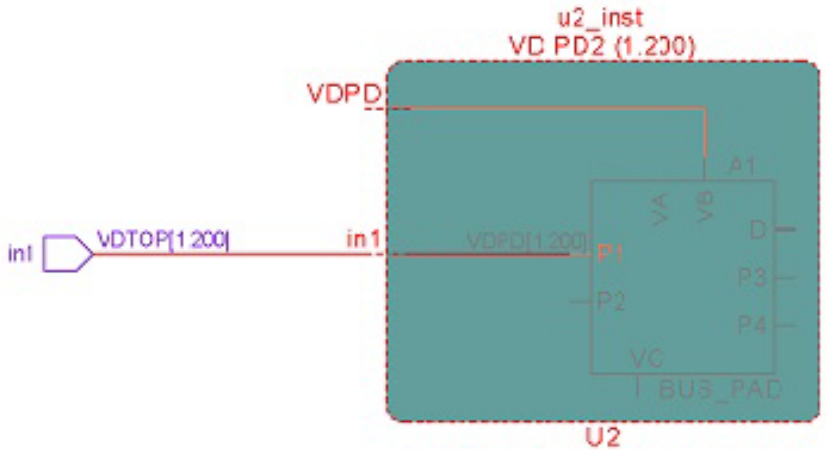


FIGURE 202. Incremental schematic

To resolve the violation, update the UPF file to ensure that VDTOP is not less-on than VDPD. Also, since the *lp_check_src_driver_supply_name* parameter is set to 1, the driver supply net and receiver supply net should be the same.

Default and Severity label

Error

Rule Group

LP_ERC_Checks

Reports and Related files

None

LPERC02

Checks incompatibility between input/output supply for power switch and level shifters

When to Use

Check for incompatibility between the supply nets of driver pins/ports and supply nets of receiver pins/ports of library cells present in the design.

Description

The *LPERC02* rule checks incompatibility between input/output supply for power switches and level shifters.

This rule has the following subrules:

- *LPERC02A*: Checks if the input supply net of the power switch is operating at a different voltage than the output supply net.
- *LPERC02B*: Checks if input supply net of level-shifter or power switch is lesson than the output supply net.

LPERC02A

Checks if the input supply net of the power switch is operating at a different voltage than the output supply net

When to Use

Check for incompatibility between the supply nets of input power pins and the supply nets of output power pins of the power switch cells present in the design.

Description

The *LPERC02A* rule reports a violation for a power switch when there is a mismatch in the voltage value of the input and output supply nets.

Parameter(s)

- *lp_flag_violation_on_antenna_cell*: Default value is `no`. Set the value of this parameter to `yes` to perform rule checking on antenna cells.

Constraint(s)

SGDC

- *lp_ignore_cells_for_erc* (Optional)

Messages and Suggested Fix

The following message appears when the input supply net is at a different voltage compared to the output supply net for the power switch:

[LPERC02A_1][ERROR] Incompatible Supply Nets: Input supply net at different voltage compared to output supply net - Power Switch instance <instance_name> (Cell: <cell_name>) has input supply net <vdd_input> [<vdd_source_value>] connected to supply pin <pin_name> that is operating at different voltage w.r.t. to output supply net <vdd_output>[<vdd_output_value>] connected to supply pin <pin-name-2>

Potential Issues

The power switches present in the design should have the input supply net operating at the same voltage as that of the output supply net.

Consequences of not fixing

If input supply net is has different voltage compared to the output supply net of a power switch, the functioning of the power switch will not be correct.

How to debug and fix

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, update the UPF file and make sure there is no voltage difference between the input and output supply nets.

Example Code and/or Schematic

This example illustrates when the violation message is reported. The following message appears when the input supply net is at a different voltage compared to the output supply net.

```
Incompatible Supply Nets: Input supply net at different voltage compared to output supply net - Power Switch instance 'top.l6.l3'(Cell: PWRSW) has input supply net VDD['1.100'] connected to supply pin 'VDDI' that is operating at different voltage w.r.t. to output supply net VDDA['1.600'] connected to supply pin 'VDDO'
```

The following schematic is generated.

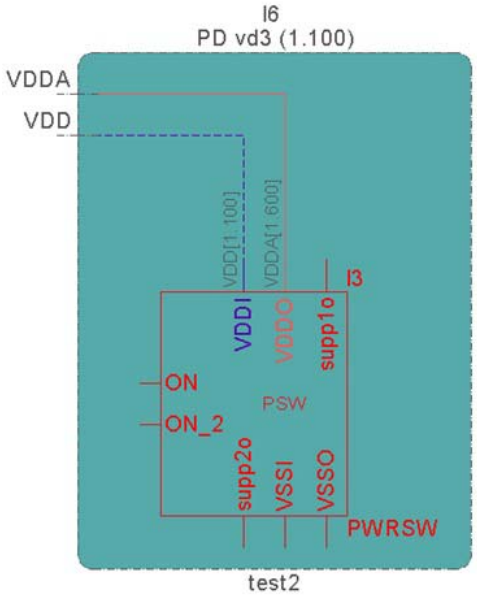


FIGURE 203. Incremental schematic

The schematic illustrates that the input supply net VDD connected to supply pin VDDI is operating at a different voltage compared to the output supply VDDA connected to VDDO supply pin.

To resolve the violation, make sure that VDD and VDDA have the same voltage.

Default and Severity label

Error

Rule Group

LP_ERC_Checks

Reports and Related files

None

LPERC02B

Checks if input supply net of level-shifter or power switch is less-on than the output supply net

When to Use

Check for incompatibility between the supply nets of input power pins and the supply nets of output power pins of the power switch and level shifter cells present in the design.

Description

The *LPERC02B* rule reports a violation for a power switch or a level shifter if the input supply net is less-on as compared to the output supply net.

Parameter(s)

- *lp_flag_violation_on_antenna_cell*: Default value is `no`. Set the value of this parameter to `yes` to perform rule checking on antenna cells.
- *lp_check_data_pin_of_els*: Default value is `0`. Set this parameter to `yes` to check data pins of the enable level shifter cells.

Constraint(s)

SGDC

- *lp_ignore_cells_for_erc* (Optional)

Messages and Suggested Fix

The following message appears when the input supply net is less-on than the output supply net:

[LPERC02B_1][ERROR] Incompatible Supply Nets: Input supply net less-on than output supply net - <Level Shifter | Power Switch> instance<instance_name> (Cell: <cell_name>) has input supply net <vdd_input> [<vdd_source_value>] connected to supply pin <pin_name> that is less-on w.r.t. to output supply net <vdd_output>[<vdd_output_value>] connected to supply pin <pin-name-2>

Potential Issues

The power switches and level shifters present in the design should not have

the input supply net as less-on as compared to the output supply net.

Consequences of not fixing

If input supply net is less on with-respect-to the output supply net of a power switch or a level shifter, the functioning of the power switch or level shifter will not be correct.

How to debug and fix

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, update the UPF file and make sure that the input supply net is not less-on than the output supply.

Example Code and/or Schematic

Example 1

The following message appears when the input supply net level shifter is less-on than the output supply net.

```
Incompatible Supply Nets: Input supply net less-on than output
supply net - Level Shifter instance 'mid.LS'(Cell: MYLS) has
input supply net VDDA['0.400'] connected to supply pin 'VDDI'
that is less-on w.r.t. to output supply net VDDB['1.800']
connected to supply pin 'VDDO'
```

The following schematic is generated.

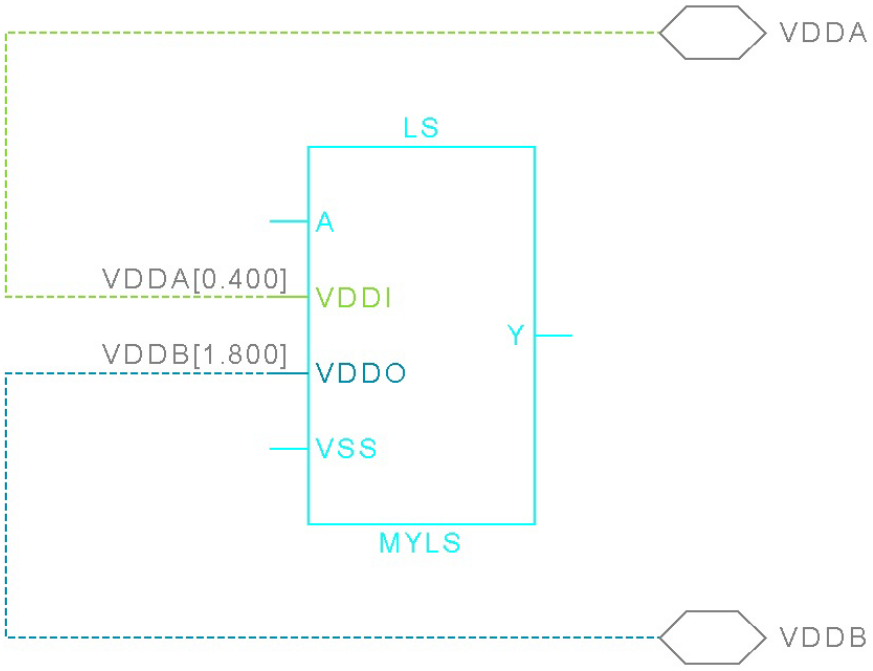


FIGURE 204. Incremental schematic

The schematic illustrates that the input supply net VDDA connected to supply pin VDDI is less-on than the output supply VDDB connected to VDDO supply pin.

To resolve the violation, VDDA should not be less-on than VDDB.

Example 2

The following message appears when the input supply net power switch is less-on than the output supply net.

```
Incompatible Supply nets: Input supply net less-on than Output
supply net - Power Switch instance 'top.l6.l3'(Cell: PWRSW) has
input supply net VDD['1.100'] connected to supply pin 'VDDI'
that is less-on w.r.t. to output supply net VDDA['1.100']
connected to supply pin 'VDDO'
```

The following schematic is generated.

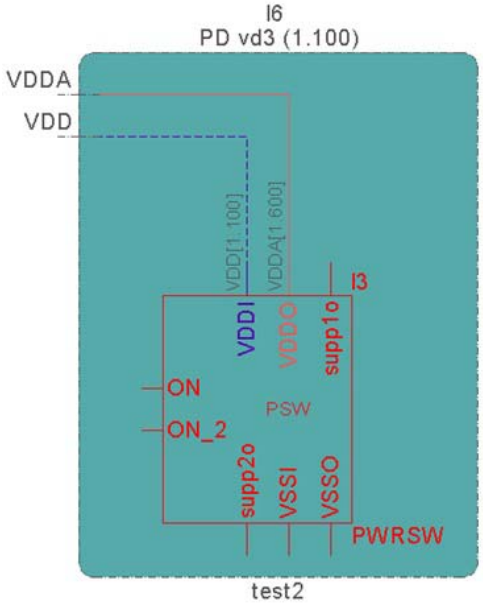


FIGURE 205. Incremental schematic

The schematic illustrates that the input supply net VDDA connected to supply pin VDDI is less-on than the output supply VDDB connected to VDDO supply pin.

To resolve the violation, VDDA should not be less-on than VDDB.

Default and Severity label

Error

Rule Group

LP_ERC_Checks

Reports and Related files

None

LPERC03

Checks incompatibility between backup power and primary power of a library cell

When to Use

Check for incompatibility between the supply nets of driver pins/ports and supply nets of receiver pins/ports of library cells present in the design.

Description

The *LPERC03* rule checks incompatibility between backup power and primary power of a library cell.

This rule has the following subrule:

- *LPERC03A*: Checks relationship between supply nets connected to backup_power pin and primary_power pin.

LPERC03A

Checks relationship between supply nets connected to backup_power pin and primary_power pin

When to Use

Check for incompatibility between the supply nets of backup power pins and the supply nets of primary power pins of library cells present in the design.

Description

The *LPERC03A* rule reports a violation message if the supply net connected to the `backup_power` pin is less-on than the supply net connected to the `primary_power` pin.

Rule Exceptions

This rule does not report a violation for the following cases:

- If instance is an AON Buffer and the *lp_check_aon_buffer* parameter is set to false.
- If instance is an Isolation cell or Level Shifter.

Parameter(s)

- *lp_check_aon_buffer*: Default is yes. Set the value to no or 0 to not check AON buffers.
- *lp_flag_violation_on_antenna_cell*: Default value is `no`. Set the value of this parameter to `yes` to perform rule checking on antenna cells.

Constraint(s)

SGDC

- *lp_ignore_cells_for_erc* (Optional)

Messages and Suggested Fix

The following message appears when the backup supply net is less-on than the primary supply net:

[LPERC03A_1][ERROR] Incompatible Supply Nets: Backup supply net less-on than Primary supply net - Instance <instance_name>

```
(Cell: <cell_name>) has supply net <vdd_input>
[<vdd_source_value>] connected to backup_power pin <pin_name>
that is less-on w.r.t. to supply net
<vdd_output>[<vdd_output_value>] connected to primary_power pin
<pin_name>
```

Potential Issues

The library instances used in the designs should not have the backup supply net less-on than the primary supply net.

Consequences of not fixing

If backup supply of a library cell is less on than the primary supply, then the functioning of the cell will not be correct.

How to debug and fix

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation, the backup supply should not be less-on than the primary supply.

Example Code and/or Schematic

This example illustrates when the violation message is reported. The following message appears when the backup supply net is less-on than the primary supply net.

```
Incompatible Supply Nets: Backup supply net less-on than
Primary supply net - Instance 'mid.Pad1'(Cell: MYPADMIXED1) has
supply net VDDA['0.400'] connected to backup_power pin 'VDDA'
that is less-on w.r.t. to supply net VDDB['1.800'] connected to
primary_power pin 'VDDB'
```

The following schematic is generated.

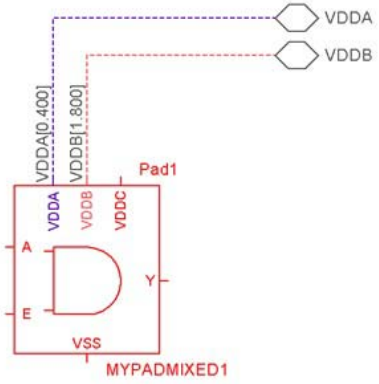


FIGURE 206. Incremental schematic

Through the schematic, you can see that the backup supply net VDDA connected to backup supply pin VDDA is less-on than the primary supply VDDB connected to primary supply pin VDDB.

To resolve the violation, the backup supply VDDA should not be less-on than primary supply VDDB.

Default and Severity label

Error

Rule Group

LP_ERC_Checks

Reports and Related files

None

LPERC04

Checks for connectivity in supply and signal pins of library cells in design

When to Use

Check for incompatibility between the supply nets of driver pins/ports and supply nets of receiver pins/ports of library cells present in the design.

Description

The *LPERC04* rule checks for connectivity in supply and signal pins of a library cells in the design. It captures unconnected, undriven, and incorrect connection to supply/signal pins.

This rule has the following subrules:

- *LPERC04A*: Checks power/ground supply pin connection with supply nets.
- *LPERC04B*: Checks if driver is an unconnected net, hanging net or is a supply net undefined in power intent.

LPERC04A

Checks power/ground supply pin connection with supply nets

When to Use

Check for incorrectly connected or unconnected supply pins.

Description

The LPERC04A rule reports a violation message for:

- unconnected supply pins.
- supply pins not connected to a supply net.
- power pins connected to a ground net or vice versa.

Parameter(s)

- *lp_flag_violation_on_antenna_cell*: Default value is `no`. Set the value of this parameter to `yes` to perform rule checking on antenna cells.

Constraint(s)

SGDC

- *lp_ignore_cells_for_erc* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the supply pin is unconnected:

```
[LPERC04A_1][ERROR] Incompatible Supply pin connection: Supply pin unconnected - Instance '<inst-name>' (Cell: <cell_name>) has <power|ground> pin '<pin-name>' unconnected
```

For debugging information, click *How to debug and fix*.

Message 2

The following message appears when the power pin is connected to the ground net or vice versa:

```
[LPERC04A_2][ERROR] Incompatible Supply pin connection: Power pin connected to ground net or vice versa - Instance '<inst-name>' (Cell: <cell_name>) has <power|ground> pin '<pin-name>' connected to a <power|ground> supply net <vdd_output>[<vdd_output_value>]
```

For debugging information, click *How to debug and fix*.

Message 3

The following message appears when the supply pin is connected to the signal net or an undefined supply net:

```
[LPERC04A_3][ERROR] Incompatible Supply pin connection: Supply pin connected to signal net or undefined supply net - Instance '<inst-name> (Cell: <cell_name>) has <power|ground> pin '<pin-name>' connected to signal net or undefined supply net
```

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not fixing

- **Message 1:** If supply pin of a library cell is unconnected, the functioning of the cell will not be correct.
- **Message 2:** If power pin is connected to ground net or vice versa, the functioning of the cell will not be correct.
- **Message 3:** If supply pin is connected to signal net or undefined supply net, the functioning of the cell will not be correct.

How to debug and fix

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix these violation messages, perform the following in the design file:

- **Message 1**: Connect the power pin to a power supply net.

- **Message 2**: Connect the power pin to a power supply net or connect the ground pin to a ground supply net.
- **Message 3**: Connect the power pin to a power supply net or connect to a supply net defined in the UPF file.

Example Code and/or Schematic

Example 1

This example illustrates how and why violation Message 1 appears.

For the following design file snippet, the *LPERC04A* rule reports a violation message because the power pin VDD has been left unconnected.

```
AONBUF aonbuf1(.I(in2),.O(w2));
```

The following violation message is reported.

```
Incompatible Supply pin connection : Supply pin unconnected -
Instance 'top.aonbuf1'(Cell: AONBUF) has power pin 'VDD'
unconnected
```

The following schematic is generated.

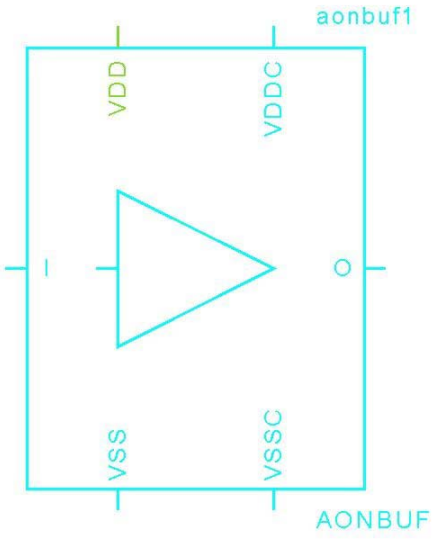


FIGURE 207. Incremental schematic

The schematic highlights the following information:

- Instance
- Supply net and supply pin

To resolve this violation, update the design file to ensure that the power pin VDD is connected to a supply net.

Example 2

This example illustrates how and why violation Message 2 appears.

For the following design file snippet, the *LPERC04A* rule reports a violation message because the power pin VDD is connected to the ground supply net VSS2.

```
AONINV aoninv1(.I(w3),.O(out3),.VDD(VSS2),.VSSC(VDD2));
```

The following violation message is reported.

```
Incompatible Supply pin connection : Power pin connected to
ground net or vice versa - Instance
'top.inst1.inst2.aoninv1'(Cell: AONINV) has power pin 'VDD'
connected to a ground supply net VSS2['0.000']
```

The following schematic is generated.

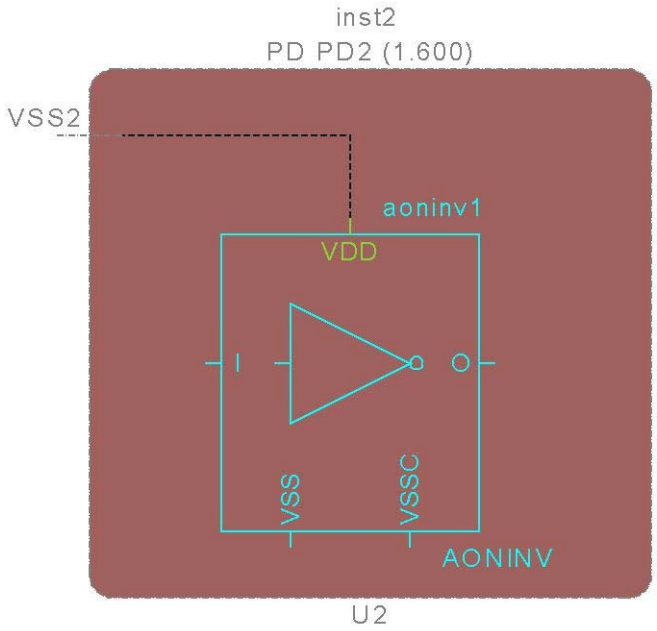


FIGURE 208. Incremental schematic

The schematic highlights the following information:

- Instance
- Supply net and supply pin

To resolve this violation, update the design file to ensure that the power pin VDD is connected to a power supply net, and not to a ground supply net.

Example 3

This example illustrates how and why violation Message 3 appears.

For the following design file snippet, the *LPERC04A* rule reports a violation message because the power pin VDD is connected to a supply net that is not defined in the UPF file.

```
BUF buf1(.I(w1),.O(out1),.VDD(1'b0),.VSS(1'b1));
```

The following violation message is reported.

```
Incompatible Supply pin connection : Supply pin connected to
signal net or undefined supply net - Instance
'top.inst1.inst2.buf1'(Cell: BUF) has power pin 'VDD' connected
to signal net or undefined supply net
```

The following schematic is generated.

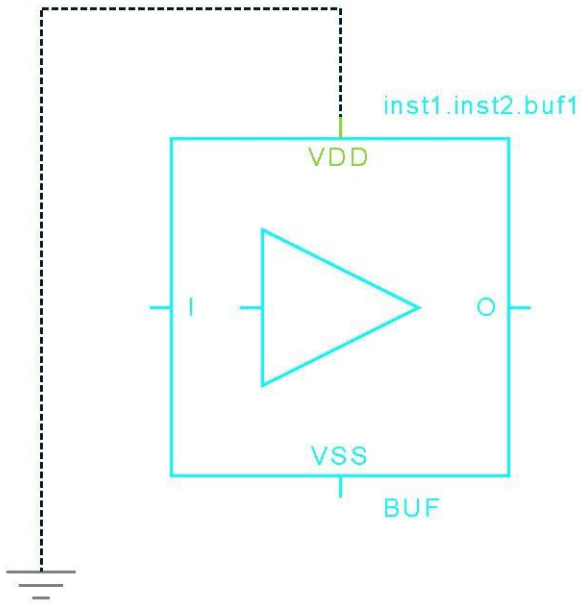


FIGURE 209. Incremental schematic

The schematic highlights the following information:

- Instance
- Supply net and supply pin

To resolve this violation, update the design file to ensure that the power pin VDD is connected to a power supply net that has been defined in the UPF file.

Default and Severity label

Error

Rule Group

LP_ERC_Checks

Reports and Related files

None

LPERC04B

Checks if driver is an unconnected net, hanging net or is a supply net undefined in power intent

When to Use

Check for incorrectly connected or unconnected signal pins.

Description

The LPERC04B rule reports a violation message for and:

1. unconnected input signal pin.
2. input signal pin connected to an undriven signal net.
3. input signal pin connected to a supply net that is not a supply net defined in the UPF file.

Parameter(s)

lp_flag_violation_on_antenna_cell: Default value is `no`. Set the value of this parameter to `yes` to perform rule checking on antenna cells.

Constraint(s)

SGDC

- *lp_ignore_cells_for_erc* (Optional)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)

Messages and Suggested Fix

Message 1

The following messages appear for a pin/port that is not driven by a primary input port or cell output pin:

[LPERC04B_1][ERROR] Incorrectly driven pin/port: Pin '<pin_name>' of instance 'hierarchal_ instance_name' (Cell: <cell_name>) is not driven by primary input port or cell output pin

For debugging information, click *How to debug and fix*.

Message 2

The following message appears for a pin/port that should be driven by a supply net defined in the power intent:

[LPERC04B_2][ERROR] Incorrectly driven pin/port: Pin '<pin_name>' of instance 'hierarchal_ instance_name' (Cell: <cell_name>) is driven by <supply0|supply1> instead of a supply net defined in power intent

For debugging information, click *How to debug and fix*.

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not fixing

- **Message 1:** If the driver of a pin of a cell is hanging, the functioning of the cell will not be correct.
- **Message 2:** If the driver of a pin of a cell is supply 0/1 instead of supply defined in power intent, the functioning of the cell will not be correct.

How to debug and fix

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix these violation messages, perform the following in the design file:

- **Message 1**: Connect the undriven pin to a driver.
- **Message 2**: Connect the signal pin to a supply net defined in the UPF file, instead of a supply0/supply1.

Example Code and/or Schematic

Example 1

This example illustrates how and why violation Message 1 appears.

For the following design file snippet, the *LPERC04B* rule reports a violation

message because the signal pin E is undriven.

```
MY_LS_EN ls1(.A(1'b1),.Y(w1),.E(),.VDDA(VDD2),.VSS(VSS2));
```

The following violation message is reported.

```
Incorrectly driven pin/port: pin 'E' of instance
'top.inst1.inst2.ls1'(Cell:MY_LS_EN) is not driven by primary
input port or cell output pin
```

The following schematic is generated.

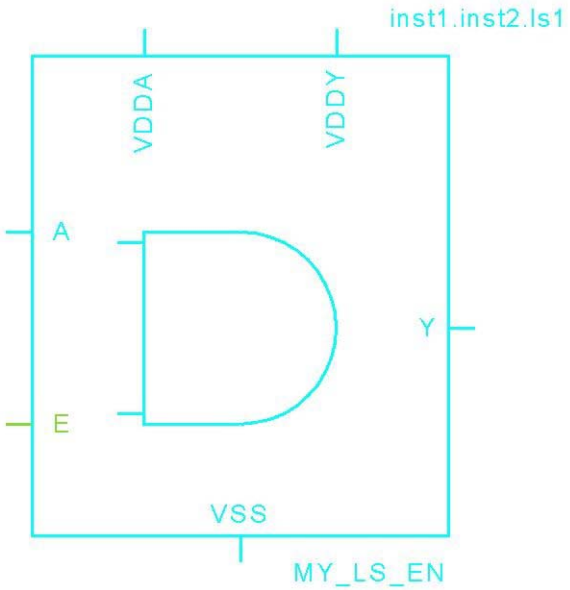


FIGURE 210. Incremental schematic

The schematic highlights the:

- Source net
- Sink pin/port with supply information annotated.
- Instance

To resolve this violation, update the design file to ensure that the undriven pin is connected to a driver.

Example 2

This example illustrates how and why violation Message 2 appears.

For the following design file snippet, the *LPERC04B* rule reports a violation message because the signal pin A is driven by supply1, instead of a supply net defined in the UPF file.

```
MY_LS_EN ls1(.A(1'b1),.Y(w1),.E(),.VDDA(VDD2),.VSS(VSS2));
```

The following violation message is reported.

```
Incorrectly driven pin/port : pin 'A' of instance
'top.inst1.inst2.ls1'(Cell:MY_LS_EN) is driven by supply1
instead of a supply net defined in power intent
```

The following schematic is generated.

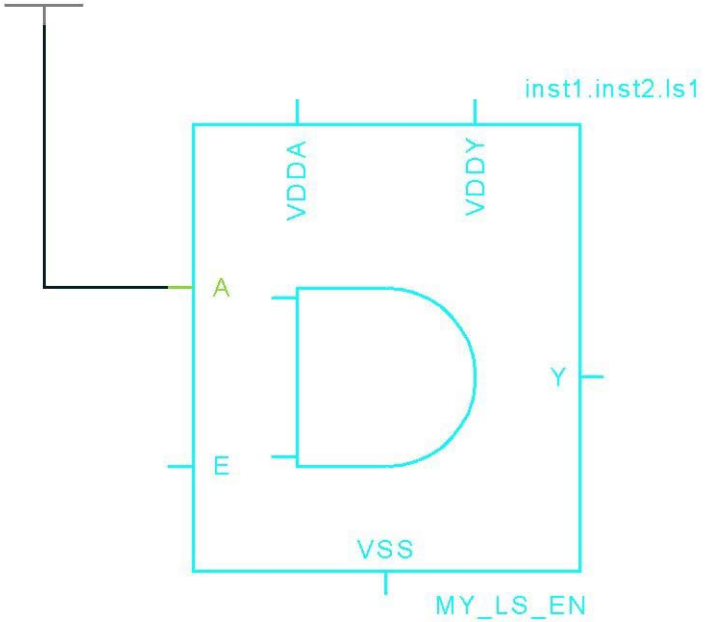


FIGURE 211. Incremental schematic

The schematic highlights the:

- Source net

- Sink pin/port with supply information annotated.
- Instance

To resolve this violation, update the design file to ensure that the signal pin is connected to a supply net defined in the UPF file, instead of a supply0/ supply1.

Default and Severity label

Error

Rule Group

LP_ERC_Checks

Reports and Related files

None

LPERC05

Reports Analog signals which are driving or getting driven by digital logic

When to Use

Use this rule to find the analog signals which are driving or getting driven by digital logic.

Description

The *LPERC05* rule reports a violation if any library cell pin, which has `is_analog` attribute as `true`, is driving or getting driven by digital logic. Everything except top level ports, tie-offs to power/ground and pins with `is_analog` set to `true` are considered as digital logic. At the RTL level, this rule also ensures that the net connected to `is_analog` port is a pure net without any logic including level shifter/isolation strategies.

Prerequisites

By default, the *LPERC05* rule does not run. To enable this rule, select the *LPERC05* rule in the Atrenta Console GUI.

Rule Exceptions

SpyGlass generated buffers (assign a=b) and domain boundary ports are skipped.

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)

Messages and Suggested Fix

Message 1

The following message is reported if a pin with the `is_analog` attribute, set to true, is driving or getting driven by a digital logic:

[LPERC05_1][ERROR] Analog pin '<pin-name>' of instance '<instance-hier-name> (<library-cell-name>)' is <direction - driving | getting driven by> a digital pin '<pin-name>' of instance '<instance-hier-name> (<library-cell-name>)'

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not fixing

There are analog pins that are driving or getting driven by the digital logic in the design, which may lead to design failure.

How to debug and fix

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix this violation message, make sure analog pin mentioned in the violation massage is not driving or getting driven by a digital logic.

Message 2

The following message is reported if isolation/level shifter strategy is present in a signal path, which starts or ends with an analog pin:

[LPERC05_2][ERROR] Analog pin '<pin-name>' of instance '<instance-hier-name> (<library-cell-name>)' has <strategy-type – Isolation | Level Shifter> strategy '<strategy-name>' specified at '<prd/port-name>' in its path

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not fixing

Isolation/level shifter strategy is present in analog logic path that may lead to design failure.

How to debug and fix

For a graphical view of the violation, double-click the message and click the **Incremental Schematic** button. Refer to the Example Code and/or Schematic section for an example.

To fix these violation messages, make sure no isolation/level shifter strategy is present in the signal path mentioned in the violation massage, since that signal is starting or ending with an analog logic.

Example Code and/or Schematic

Example 1

Consider the following library definitions of AND and AND_1:

```
cell (AND) {
      pg_pin (VDD) { pg_type : primary_power; }
      pg_pin (VDDC) { pg_type : backup_power; }
      pg_pin (VSS) { pg_type : primary_ground; }
      pin (A) { direction : input;related_power_pin: VDD;
related_ground_pin: VSS; }
      pin (B) { direction : input;related_power_pin: VDD;
related_ground_pin: VSS; }
      pin (Y) { direction : output; function :
"A&B";related_power_pin: VDD;related_ground_pin: VSS;  }
   }
cell (AND_1) {
      pg_pin (VDD) { pg_type : primary_power; }
      pg_pin (VDDC) { pg_type : backup_power; }
      pg_pin (VSS) { pg_type : primary_ground; }
      pg_pin (VSSC) { pg_type : primary_ground; }
      pin (A) { direction : input;related_power_pin:
VDD;related_ground_pin: VSS; is_analog : true  }
      pin (B) { direction : input;related_power_pin:
VDD;related_ground_pin: VSS;  }
      pin (Y) { direction : output; function :
"A&B";related_power_pin: VDD;related_ground_pin: VSS;  }
   }
```

Verilog:

```
module top (input in1,in2,in3,en1,iso,output
out1,out2,out3);
```

```
wire w1,w2;
AND and1 (.A(in2),.B(in2),.Y(w1));
Test1 inst1 (.in1(w1),.in2(in2),.iso(iso),.out1(out1));
endmodule
module Test1 (input in1,in2,iso, output out1);
AND_1 and2 (.A(in1),.B(in2),.Y(out2));
Endmodule
```

The following violation message is reported by the *LPERC05* rule because analog pin A of AND_1 is getting driven by digital pin Y of AND:

```
Analog pin 'A' of instance 'top.inst1.and2 (AND_1)' is getting
driven by a digital pin 'Y' of instance 'top.and1 (AND)'
```

The following schematic is generated.

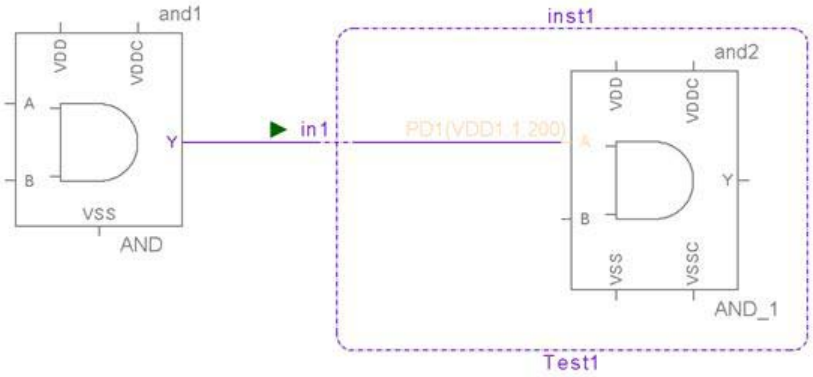


FIGURE 212. Incremental schematic

Example 2

Consider the following UPF snippet:

```
create_power_domain TOP -include_scope
create_power_domain PD1 -elements {inst1}
create_supply_port VDD -domain TOP
create_supply_net  VDD -domain TOP
connect_supply_net VDD -ports {VDD}
set_isolation ISO1  -domain PD1 -applies_to inputs  -elements
inst1/in1 -clamp_value 0 -location automatic  -
```

```
isolation_power_net VDD -isolation_signal {iso} -
isolation_sense high
```

Verilog:

```
module top (input in1,in2,in3,en1,iso,output
out1,out2,out3);
wire w1,w2;
AND and1 (.A(in2),.B(in2),.Y(w1));
Test1 inst1 (.in1(w1),.in2(in2),.iso(iso),.out1(out1));
endmodule
module Test1 (input in1,in2,iso, output out1);
AND_1 and2 (.A(in1),.B(in2),.Y(out2));
Endmodule
```

For this example, the *LPERC05* rule reports the following message because there is an isolation strategy present in the signal path ending with an analog pin:

```
Analog pin 'A' of instance 'top.inst1.and2 (AND_1)' has
Isolation strategy 'ISO1' specified at 'top.inst1.in1' in its
path
```

Default and Severity label

Error

Rule Group

LP_ERC_Checks

Reports and Related files

None

LPERC06

Reports power/ground nets that are connected to multiple inout/output power/ground pins

When to Use

Use this rule to find PG nets that are connected to multiple inout/output PG pins.

Description

The *LPERC06* rule reports violation when a power/ground net is connected to more than one inout or output power/ground pins of lib cells, as the power/ground net gets conflicting values.

Prerequisites

By default, the LPERC06 rule will not run. To enable this rule, select the *LPERC06* rule in SpyGlass Console.

Rule Exceptions

Input PG pins that are connected to each other are not be reported by this rule.

Language

Verilog, VHDL.

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)

Messages and Suggested Fix

The following message is reported if a power/ground net *<net-name>* is connected to more than one inout or output power/ground pins:

[LPERC06_1][ERROR] Signal '<net-name>' has multiple simultaneous drivers

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not fixing

Since the reported power/ground net is connected to more than one output or inout pg-pins, it will lead to conflicting values on the net, which can result in design failure.

How to debug and fix

Back-reference is available for the reported power/ground net. To fix this violation message, review and update the design.

Example Code and/or Schematic

Consider the following example code:

```
module top(in1, in2, out1, out2, out3, out4, VDD1, VDD2,
VDD3, VSS);
input in1, in2, VDD1, VDD3, VDD2, VSS;
output out1, out2, out3, out4;
wire w1, w2, w3, w4, w5, w6;
AND_3 instOfAnd3(.A(in1), .B(w2), .Y(w2), .VDD(VDD1),
.VSS(VSS));
DOM1 pqr(.inp1(w2), .inp2(in2), .outp1(w4), .outp2(w6),
.VDD(VDD1), .VSS(VSS));
BUFA instOfBufA(.VDD(VDD1), .VSS(VSS), .A(w6), .Y(out3));
AND_2 instOfAnd2(.VDD(VDD1), .VSS(VSS), .A(w6), .B(w4),
.Y(out4));
endmodule

module DOM1(inp1, inp2, outp1, outp2, VDD, VSS);
input inp1, inp2, VDD, VSS;
output outp1, outp2;
AND_4 instOfAnd4(.VDD(VDD), .VSS(VSS), .A(inp1), .B(inp2),
```

```
.Y(outp1));
Endmodule
```

The following violation message is reported by the *LPERC06* rule because VDD1 is connected to VDD pin(inout) of cell BUFA,VDD pin(inout) of cell AND_2, and VDD pin(inout) of cell AND_4:

```
Signal 'top.VDD1' has multiple simultaneous drivers
```

The following schematic is generated.

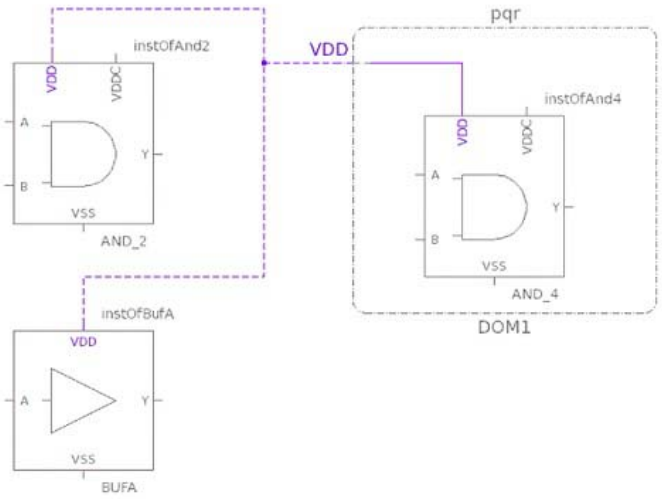


FIGURE 213. Incremental schematic

Default and Severity label

Error

Rule Group

Electrical Check Rules

Reports and Related files

None

CPF Check Rules

The CPF Check rules are as follows:

Rule	Reports
checkCPF_existence	If design objects (instances/nets/pins) specified with CPF commands do not exist in design.
CPF_lowpower01	If a valid voltage value is not inferred for a power domain
CPF_lowpower03	If complex expressions are not supported in CPF command
CPF_lowpower04	If switchable pin type information is not available in the `define_power_switch_cell` command
CPF_lowpower05	If the top-level design unit specified with the `set_design` command is not same as the top-level design unit specified as `current_design` in the SGDC file
CPF_lowpower07	If none of the option, `-from` or `-to`, is specified for any isolation rule with the `create_isolation_rule` command
CPF_lowpower09	If 'supply' is not inferred properly from CPF commands
CPF_lowpower10	If the power switch mentioned in the `update_power_switch_rule` command is not found
CPFSEM_2	Checks the presence of the power domain in scope
CPFSEM_3	Checks if the default power domain is already present in scope
CPFSEM_4	Checks if default power domain is not present in scope
CPFSEM_5	Checks the name conflict in the same scope
CPFSEM_6	Checks the presence of the nominal condition
CPFSEM_7	Checks the presence operating corner
CPFSEM_8	Checks if library set is defined
CPFSEM_10	Checks if a ground net is defined
CPFSEM_11	Checks the presence of power net
CPFSEM_12	Checks the presence of bias net
CPFSEM_13	Checks the presence of isolation rule
CPFSEM_14	Checks the presence of level shifter rule

Rule	Reports
CPFSEM_15	Checks the presence of power mode
CPFSEM_17	Checks the presence power switch rule
CPFSEM_19	Checks if set_design is preceded by a set_instance command
CPFSEM_21	Checks if set_design is specified for the current instance
CPFSTX_1	Checks if mutually exclusive options are specified
CPFSTX_2	Checks if at least one mandatory argument is specified
CPFSTX_4	Checks the presence of an invalid domain condition
CPFSTX_5	Checks the presence of a wrong array naming style
CPFSTX_6	Checks the presence of a wrong hierarchy delimiter
CPFSTX_7	Checks the presence of an invalid value
CPFSTX_8	Checks the presence of an invalid register naming style
CPFSTX_9	Checks if the arguments are specified together
CPFSTX_10	Checks if wrong number of arguments are specified
CPFSTX_11	Checks if the specified value is out of range
CPFSTX_13	Checks if an invalid option is specified for command
CPFSTX_14	Checks if no value is specified for argument
CPFSTX_15	Checks if an invalid value type is specified
CPFSTX_16	Checks if an invalid value is specified for argument
CPFSTX_17	Checks if a mandatory argument is missing
CPFSTX_19	Checks the presence of a wildcard reference
CPFSTX_20	Checks the presence of hierarchical separator
CPFSTX_22	Checks the presence of top design
CPFSTX_23	Checks the presence of control group
CPFSTX_24	Checks the presence of invalid commands in macro model
CPFSTX_25	Checks the presence of macro model
CPFSTX_26	Checks the presence of the specified file
CPFSTX_27	Checks the presence of the specified design
CPFSTX_28	Checks if an invalid command is specified in control group
CPFSTX_29	Checks if an invalid object name is specified

Rule	Reports
CPFSTX_30	Checks if an invalid group view is specified
CPFSTX_31	Checks the presence of an analysis view
CPFSTX_32	Checks if an incorrect argument format is specified in some switch of command
CPFSTX_33	Checks if an incorrect design name is specified in end_design
CPFSTX_34	Checks if an incorrect model name is specified in end_macro_model
CPFSTX_35	Checks if set_macro_model is specified for the current block
CPFSTX_36	Checks if an unknown command is specified
CPFSTX_38	Checks if power domain is visible inside control group
CPFSTX_39	Checks if control group is visible inside control group
CPFSTX_40	Checks if set_instance without instance name contains parameters
CPFSTX_41	Checks the presence of a command outside set_macro_model and end_macro_model block
CPFSTX_42	Checks if power net is specified as power supply net to an always on power domain

checkCPF_existence

Reports design objects, specified with CPF commands, which do not exist

When to Use

This is a setup rule and always runs by default.

Description

The *checkCPF_existence* rule reports design objects, such as instances, nets, pins, or cells, which do not exist. These design objects have been specified with CPF commands.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

Message 1

The following message appears when the design object <*obj-name*> <*type*> (instances/nets/pins/cells) specified with CPF commands is not found in the design <*top-du-name*>:

[checkCPF_existence_1][FATAL] '<obj-name>'[<type>] not found on/within module '<top-du-name>'

[checkCPF_existence_2][WARNING] '<obj-name>'[<type>] not found on/within module '<top-du-name>'

Message 2

The following message appears when the design object <*obj-name*> <*type*> (instances/nets/pins/cells) specified with CPF commands is neither found in the design nor in the library:

[checkCPF_existence_3][FATAL] '<obj-name>'[<type>] not found in design as well as library

Potential Issues

The violation message explicitly states the potential issues.

Consequences of not fixing

The power intent specified for the design objects, which do not exist, will not be applied.

How to debug and fix

Review the CPF file and update the file to reflect the name of the design object. Ensure there are no spelling mistakes.

The name should be the same as that stated in the design file.

Example Code and/or Schematic

Suppose `top.i1` exists in the design but `top.i2` does not exist in the design. For the following CPF command, the *checkCPF_existence* rule reports a violation because `top.i2` does not exist in the design.

```
create_power_domain –instances top.i1 top.i2
```

Default and Severity label

Warning/Fatal

Rule Group

`CPF Check`

Reports and Related files

None

CPF_lowpower01

Reports if a valid voltage value is not inferred for a power domain

When to Use

This is a setup rule always runs by default.

Description

The *CPF_lowpower01* rule reports when an invalid voltage value is specified for a power domain. This condition is possible in the following cases:

- For the main power net of a power domain, the `-primary_power_net` argument is not specified in the *update_power_domain* command.
- For a power domain, the nominal operating condition, that is, voltage greater than zero, is not specified.
- For the main power net of a power domain, the `-voltage` argument of the *create_power_nets* command is not specified.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

Message 1

The following message appears when you do not specify the main power net or the nominal operating condition for the domain <*domain-name*>:

```
[FATAL] The main power net or nominal operating condition
(voltage greater than zero) should be specified for domain
'<domain-name>'
```

Potential Issues

This violation appears when there is a setup error in the power intent.

Consequences of Not Fixing

A valid voltage value is mandatory for the downstream rules to function. If you do not fix this violation, a valid voltage value may not be inferred for the domain. As a result, the downstream rules may not function properly.

How to Debug and Fix

Check the specifications of the reported power domain.

To fix this violation, specify the nominal operating condition for the power domain. In addition, provide the power net for the domain by using the `-primary_power_net` argument in the `update_power_domain` command.

Message 2

The following message appears when the `-voltage` argument of the `create_power_nets` command is not specified for the main power net of the domain *<domain-name>*:

[FATAL] The '-voltage' field should be given with '-primary_power_net '<power-net-name>' specified with command 'update_power_domain' for domain '<domain-name>'

Potential Issues

This violation appears when there is a setup error in the power intent.

Consequences of Not Fixing

A valid voltage value is mandatory for the downstream rules to function. If you do not fix this violation, a valid voltage value may not be inferred for the domain. As a result, the downstream rules may not function properly.

How to Debug and Fix

Check the specifications of the reported power domain.

To fix this violation, specify the `-voltage` argument of the `create_power_nets` command for the main power net of the domain.

Example Code and/or Schematic

Example 1

For the following code, the *CPF_lowpower01* rule reports a violation because the `update_power_domain` command is not specified for the domain `V3`.

```
set_cpf_version 1.0
set_design top

create_power_domain -name V1 -default
create_power_domain -name V3 -instances .inst2
create_power_domain -name V2 -instances .inst1

create_power_nets -nets vdd0 -voltage 1.2
create_power_nets -nets vdd1 -voltage 1.6
create_power_nets -nets vdd2 -voltage 1.8
create_ground_nets -nets vss -voltage 0.0

update_power_domain -name V1 -primary_power_net vdd0
internal_ground_net vss
update_power_domain -name V2 -primary_power_net vdd1

end_design
```

Example 2

Consider the following example:

```
create_power_domain -name V1 -default
create_power_domain -name V3 -instances .inst2
create_power_domain -name V2 -instances .inst1

create_power_nets -nets vdd0
create_power_nets -nets vdd1
create_power_nets -nets vdd2 -voltage 1.8
create_ground_nets -nets vss -voltage 0.0

update_power_domain -name V1 -internal_power_net vdd0
internal_ground_net vss
```

```
update_power_domain -name V2 -internal_power_net vdd1
update_power_domain -name V3 -internal_power_net vdd2
```

For the above code, the *CPF_lowpower01* rule reports the following violation because the `-voltage` argument is not given with `internal_power_net vdd0` for domain `V1`:

```
The '-voltage' field should be given with '-internal_power_net'
'vdd0' specified with command 'update_power_domain' for domain
'V1'
```

Default Severity Label

Fatal

Rule Group

`CPF Check Rules`

Reports and Related Files

None

CPF_lowpower03

Reports unsupported unary expressions specified in CPF commands

When to Use

This is a setup rule and always runs by default.

Description

Currently, for the CPF commands, the expressions only with | and ! (OR and NOT) operators are supported. Expressions with & (AND) operator are considered as complex and are not supported.

The `CPF_lowpower03` rule reports the occurrences of such complex expressions or syntactically incorrect expressions specified with any CPF command.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

Message 1

The following message appears when a complex expression is specified in the *<field-name>* field of the *<cmd-name>* CPF command:

[WARNING] Complex expression is not supported in '<fieldname>' field of '<cmd-name>' command

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the expression specified in the *<field-name>* field of the *<cmd-name>* CPF command is syntactically

incorrect:

```
[WARNING] Syntax error in expression specified in '<field-
name>' field of '<cmd-name>' command
```

Potential Issues

The violation messages appear because you have either:

- used a complex operator in the CPF command or
- specified a syntactically incorrect expression in the field of the CPF command.

Consequences of Not Fixing

Though the SpyGlass Power Verify solution will continue with its analysis, the results may not be as expected.

How to Debug and Fix

The CPF command reported in the violation message is highlighted in the Atrenta Console GUI.

Review the CPF command for complex expressions and syntax errors. Ensure it does not contain an & (AND) operator.

Example Code and/or Schematic

Example 1

In the following snippet, the *CPF_lowpower03* rule reports a violation message because the name field contains a complex unary operator:

```
create_power_domain -name shutOFF -instances t1.inst4 -
shutoff_condition {isosig1 & isosig1}
```

To resolve the violation message reported, specify the command without using an & (AND) operator, as shown in the following snippet:

```
create_power_domain -name shutOFF -instances t1.inst4 -
shutoff_condition {!isosig1}
```

Example 2

Consider the following UPF snippet:

```
create_power_domain -name VA -instances {ua }  -
shutoff_condition {(ein1)ein2}
```

In the above UPF example, this rule reports the following violation message

because there is a syntax error in the given expression:

```
Syntax error in expression specified in 'shuttOff_condition'
field of 'create_power_domain' command
```

Default Severity Label

Warning

Rule Group

CPF Check

Reports and Related Files

None

CPF_lowpower04

Reports the unavailability of switchable pin type information in define_power_switch_cell

When to Use

This is a setup rule and always runs by default.

Description

The `CPF_lowpower04` rule reports a violation message in the following cases:

- When the `-power_switchable` and `-power` arguments are not specified with power switch of type `header`.
- When the `-ground_switchable` and `-ground` arguments are not specified with the power switch of type `footer`.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

CPF

define_power_switch_cell (Mandatory)

Messages and Suggested Fix

The following message appears when the arguments, *<arg1>* and *<agr2>* are not specified with the power switch of type *<type>*:

[WARNING] Arguments '<arg1>' and '<arg2>' should be specified for '<type>' type powerswitch

Where,

- *<arg1>* is `-power_switchable` and *<arg2>* is `-power` when *<type>* is `header`.

- *<arg1>* is `-ground_switchable` and *<arg2>* is `-ground` when *<type>* is `footer`.

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

Though the SpyGlass Power Verify solution will continue with its analysis, the results may not be as expected. Rules using this information may produce wrong results.

How to Debug and Fix

The CPF command reported is highlighted in the Atrenta Console GUI.

Review the power switch type and update the CPF command to include the missing arguments stated in the violation message.

Example Code and/or Schematic

In the following snippet, the *CPF_lowpower04* rule reports a violation because the `-power_switchable` and `-power` arguments are not specified with the power switch of type `header`:

```
define_power_switch_cell -cells "hD8MM hD16MM" -
stage_1_enable SLEEPIN1 -type header
```

To resolve the violation message reported, specify the `-power_switchable` and `-power` arguments, as shown in the following snippet:

```
define_power_switch_cell -cells "XD8MM XD16MM" -
stage_1_enable SLEEPIN1 -type header –power_switchable VDD –
power TVDD
```

Default Severity Label

Warning

Rule Group

`CPF Check`

Reports and Related Files

None

CPF_lowpower05

Reports if set_design is not specified or is specified incorrectly for the top design unit

When to Use

This is a setup rule and always runs by default.

Description

The *CPF_lowpower05* rule reports a violation in the following cases:

- The *set_design* command is not specified in the CPF file for the top-level design unit.
- The top-level design unit specified with the *set_design* command is not the same as the top-level design unit specified in the *current_design* SGDC command.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

- *current_design* (Mandatory)

CPF

- *set_design* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the *set_design* command is not specified in the CPF file for the top-level design unit:

[FATAL]`The top design unit should be specified with 'set_design' command`

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the top-level design unit <*duname*> specified with the *set_design* command is different from the top-level design unit specified as *current_design* in the SGDC file:

[FATAL]`The top design unit '<du-name>' specified with 'set_design' command in CPF does not match with current_design`

Potential Issues

The violation message explicitly states the potential issue.

Consequences of not Fixing

If this is not fixed, it means that our SGDC and CPF file are out of sync as we are specifying different top design unit in both these files.The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

Review the CPF file and ensure that the top-level design unit is specified using the *set_design* command. In addition, ensure the top-level design unit stated in the *set_design* command is the same as that specified through the *current_design* command in the SGDC file.

Example Code and/or Schematic

Suppose, you have defined the top-level design unit in the SGDC file as follows:

```
current_design top
```

In the CPF file, if you haven't defined the top-level design unit, the *CPF_lowpower05* rule reports a violation. In addition, if you have defined the top-level design unit with a different name, as shown in the following snippet, the *CPF_lowpower05* rule reports a violation:

```
set_design top11
```

To resolve the reported violation message, specify the top-level design unit name as is specified in the *current_design* SGDC command.

Default Severity Label

Fatal

Rule Group

CPF Check

Reports and Related Files

None

CPF_lowpower07

Reports create_isolation_rule commands that do not have either '-from' or '-to'

When to Use

This is a setup rule and always runs by default.

Description

The *CPF_lowpower07* rule reports a violation message when an isolation rule created using the *create_isolation_rule* CPF command contains neither the `-from` nor `-to` arguments. If both these options are missing, the isolation rule is ignored.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

CPF

create_isolation_rule (Mandatory)

Messages and Suggested Fix

The following message appears when neither the `-from` nor `-to` argument is specified for the isolation rule <*rule-name*>:

[WARNING] The isolation rule '<rule-name>' is ignored, at least one of arguments '-from' or '-to' should be present

Potential Issues

The violation message explicitly states the potential issue.

Consequences of not Fixing

Since neither `-from` and `-to` options are specified, we are not defining the isolation rule for a specific domain. Therefore, the rule is not associated with any domain. This is similar to not writing an isolation rule

for a domain because the isolation rule will be ignored. The SpyGlass Power Verify analysis will not be as expected.

How to Debug and Fix

The CPF command reported is highlighted in the Atrenta Console GUI.

To fix this violation, specify either the `from` or `to` argument in the *create_isolation_rule* CPF command.

Example Code and/or Schematic

For the following snippet, the *CPF_lowpower07* rule reports a violation message because neither the `from` nor `to` argument is specified:

```
create_isolation_rule -name ISORULE001
```

To resolve the violation message, specify one of the arguments, as shown in the following snippet:

```
create_isolation_rule -name ISORULE001 -from ATR1|PDD1
```

Default Severity Label

Warning

Rule Group

`CPF Check`

Reports and Related Files

None

CPF_lowpower09

Checks to ensure that 'supply' is inferred properly from CPF commands

When to Use

This is a setup rule and always runs by default.

Description

The *CPF_lowpower09* rule reports in the following cases:

- For a power domain, the nets specified with `internal_power_net` or `internal_ground_net` argument of the *update_power_domain* command are not switched supply nets. This is because the nets that do not have either the `internal` or `external_shuttoff_condition` argument specified.
- For an always-on domain, the nets specified with the `internal_power_net` or `internal_ground_net` argument of the *update_power_domain* command are switched supply nets.
- External power/ground net is not specified for power/ground net by the `external_power_net or external_ground_net` argument of the *create_power_switch_rule* command.
- The switched supply net is not associated with any domain.
- For the *create_power_switch_rule* command, the `enable_condition_1` argument is not specified and the:
  - `external_power_net or external_ground_net` argument is used.
  - `enable_condition_2` argument is used.
- The domain specified with the *create_power_switch_rule* command is an always-on domain.
- The net specified with `-external_power_net/-external_ground_net` is a switched net.

Prerequisites

Enable the *LPPLIB17* rule.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

CPF

- *update_power_domain* (Optional)
- *create_power_switch_rule* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when the supply net <*net-name*> specified for the power domain <*dom-name*> is not a switched supply net:

[WARNING] The supply net '<net-name>' specified with powerdomain '<dom-name>' should be switched supply net

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

The domain is defined as power domain. If its supply is not switchable (always-on), the domain will never shutdown. This will be equivalent to defining that domain as an always-on domain and you might get incorrect results.

How to Debug and Fix

Review the CPF file. To resolve this violation message, create a switched supply net. To create a switched supply net, ensure the nets specified with the `internal_power_net` or `-internal_ground_net` argument of the *update_power_domain* command have either the `internal` or `external_shuttoff_condition` argument specified.

Message 2

The following message appears when the supply net <*net-name*>

specified for the always-on domain <*dom-name*> is a switched supply net:

[FATAL] The supply net '<net-name>' specified with always-on domain '<dom-name>' should not be switched net

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

The domain is defined as an always-on domain. This means that it should never shutdown. If its supply is switchable (not always-on), the domain can shutdown and so the point of declaring it as an always-on domain is defeated.

How to Debug and Fix

Review the CPF file. To resolve this violation message, create an always-on supply net. To create an always-on supply net, ensure the nets specified with the `internal_power_net` or `internal_ground_net` argument of the *update_power_domain* command do not have `internal` or `external_shuttoff_condition` argument specified with them.

Message 3

The following message appears when the external <*net-type*> (power/ground) net is not specified by <*arg*> (`external_power_net`/-`external_ground_net`, respectively) argument of the *create_power_switch_rule* command for the supply net <*net-name*> of the power domain <*dom-name*>:

[FATAL] External <net-type> net is not specified by '<arg>' argument of 'create_power_switch_rule' command for supply net '<net-name>' of powerdomain '<dom-name>'

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

You have not specified the external power/ground nets to which the source pin of the power switch should be connected. This will drive the power switch. Since it is a FATAL error, the SpyGlass run will terminate and further checking will not be done.

How to Debug and Fix

Review the CPF file. To resolve this violation message, specify the external (power/ground) net with either the external_power_net or external_ground_net argument of the *create_power_switch_rule* command for the supply net.

Message 4

The following message appears when the switched supply net <*net-name*> is not associated with any domain:

[WARNING] The switched supply net '<net-name>' is not associated with any domain

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

This will not cause any error in the design. However, if you declare a supply, it is expected that it will be related to a domain. It is equivalent to defining an extra supply which is not used.

How to Debug and Fix

Review the CPF file and make sure that each supply is associated with some domain. It may happen that you define a supply net and associate a different supply net with a domain due to spelling mistake.

Message 5

The following message appears when <*fld1-name*> (external_power_net/-external_ground_net or enable_condition_2) argument is specified without the <*fld2-name*> (enable_condition_1) argument, for the *create_power_switch_rule* command:

[FATAL] "Command 'create_power_switch_rule': Field '<fld1-name>' can/should be specified with field '<fld2-name>' only"

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

If this is not fixed, it means you are defining enable_condition_2 for the power switch without defining the enable_condition_1. This will

be for power switches having more than one enables, so we should define enable_condition_1 for first and then enable_condition_2 for second enable. Since this is a FATAL error, the SpyGlass run will terminate and further checking will not be done.

How to Debug and Fix

Review the CPF file. Specify either the external_power_net, external_ground_net, or enable_condition_2 arguments with the enable_condition_1 argument for the *create_power_switch_rule* command.

Message 6

The following message appears when domain <*dom-name*> specified with the *create_power_switch_rule* command is an always-on domain:

[FATAL] Domain '<dom-name>' specified with create_power_switch_rule command should be off-domain

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

If the domain specified with the *create_power_switch_rule* command is not an off-domain, it means the power switch is redundant as the domain will never shutdown.

How to Debug and Fix

Review the CPF file. Make sure that the domain specified with the –domain field of the *create_power_switch_rule* command is a shutoff domain and not an always-on domain.

Message 7

The following message appears when the net <*net-name*> specified with <*arg1*> (external_power_net/-external_ground_net) of *create_power_switch_rule* command is a switched net:

[FATAL] The supply net '<net-name>' specified with '<arg1>' argument of create_power_switch_rule command should be always-on

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

The driver of power switch is not always-on, and so it can shutdown. Since this a FATAL error, the SpyGlass run will terminate and further checking will not be done.

How to Debug and Fix

Review the CPF file and specify a switched net in the `external_power_net or external_ground_net` arguments of the *create_power_switch_rule* command.

Example Code and/or Schematic

Example 1

Consider the following example:

```
create_power_nets -nets {VDD_SW VHIGH_SW} -voltage 1.2

create_power_domain -name V1_VIRTUAL -instances {U6 U5 }  -
shutoff_condition !on_1

update_power_domain -name V1_VIRTUAL -internal_power_net
VDD_SW -internal_ground_net VSS_SW
```

`VDD_SW` as an always-on supply because the `-shutoff` condition is not specified for it. The *CPF_lowpower09* rule reports the following violation message because this net is `internal_power_net` of domain `V1_VIRTUAL`, which is a shut-off domain:

```
The supply net 'VDD_SW' specified with powerdomain 'V1_VIRTUAL'
should be switched supply net
```

Example 2

Consider the following example:

```
create_power_nets -nets VDDB -voltage 1.1 -
external_shutoff_condition {en2}

create_power_domain -name vd2 -instances l3

update_power_domain -name vd2 -internal_power_net VDDB -
internal_ground_net VSSB
```

`VSSB` is a switched supply, but the domain `vd2` is an always-on domain, since no shutoff condition is mentioned for it. The *CPF_lowpower09* rule

reports the following violation message:

```
The supply net 'VDDB' specified with always-on domain 'vd2'
should not be switched net
```

Example 3

Consider the following example:

```
create_power_domain -name vd2 -instances l3 -
shutoff_condition {en1}

update_power_domain -name vd2 -internal_power_net VDDA -
internal_ground_net VSSA

create_power_switch_rule -name PS -domain vd2 -
external_power_net VDD
```

Since, `external_ground_net` in the *create_power_switch_rule* command is not specified, the *CPF_lowpower09* rule reports the following violation message:

```
External ground net is not specified by –
external_ground_net argument of create_power_switch_rule
command for supply net 'VSSA' of powerdomain 'vd2'
```

Example 4

Consider the following example:

```
create_power_nets -nets VDDA -voltage 1.1 -
external_shutoff_condition {en1}
```

A switched power net is defined, but it is not used anywhere. Therefore, the *CPF_lowpower09* rule reports the following violation message:

```
The switched supply net 'VDDA' is not associated with any
domain
```

Example 5

Consider the following example:

```
update_power_switch_rule -name PS4 -enable_condition_1
{!((en3 & en5) | !(!en1))} -enable_condition_2 {en5}
```

Since a complex expression is specified in `enable_condition_1`, it is ignored. Therefore, it is equivalent to specifying `enable_condition_2` without specifying `enable_condition_1`. The *CPF_lowpower09* rule

reports the following violation message:

```
Command update_power_switch_rule: Field '-enable_condition_2'
can/should be specified with field '-enable_condition_1' only
```

Example 6

Consider the following example:

```
create_power_domain -name vd1 –default

create_power_switch_rule -name PS3 -domain vd1 -
external_power_net VDD
```

`vd1` is defined as an always-on domain and used it in the *create_power_switch_rule* command. The *CPF_lowpower09* rule reports the following violation message:

```
Domain 'vd1' specified with create_power_switch_rule command should
be off-domain
```

Example 7

Consider the following example:

```
create_ground_nets -nets VSSB -voltage 0.0 -internal

create_power_switch_rule -name PS6 -domain vd3 -
external_ground_net VSSB
```

`VSSB` is defined as a switchable net and it is used in the `external_ground_net` argument of the *create_power_switch_rule* command. The *CPF_lowpower09* rule reports the following violation message:

```
The supply net 'VSSB' specified with '-external_ground_net'
argument of create_power_switch_rule command should be always-on.
```

Default Severity Label

Warning/Fatal

Rule Group

`CPF Check`

Reports and Related Files

None

CPF_lowpower10

Checks the power switch mentioned in update_power_switch_rule

When to Use

This is a setup rule and always runs by default.

Description

The *CPF_lowpower10* rule reports a violation when the power switch specified with the `cells` argument of the *update_power_switch_rule* command is neither defined by using the *define_power_switch_cell* command nor present in the library.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

CPF

update_power_switch_rule (Mandatory)

define_power_switch_cell (Optional)

Messages and Suggested Fix

The following message appears when the power switch <*pwr-swtch*> is not found:

[WARNING] Power switch '<pwr-swtch>' specified through '-cells' field of command 'update_power_switch_rule' is neither specified through command 'define_power_switch_cell' nor present in the library

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

If this is not fixed, it means that you are updating the power switch rule for

a cell that is either non-existent or not a power switch. The cells specified with the `cells` argument are not identified as a power switch cell. Therefore, the power switch rule may report false violation messages on those cells.

How to Debug and Fix

The CPF command reported is highlighted in the Atrenta Console GUI.

To fix this violation, either ensure the power switch specified through the `cells` argument of the *update_power_switch_rule* command is present in the library with the `coarse_grain` attribute or update the *define_power_switch_cell* command with the correct power switch name. Ensure there are no spelling mistakes.

Example Code and/or Schematic

For the following snippet, the *CPF_lowpower10* rule reports a violation message because the power switch cell specified in the *update_power_switch_rule* command is not the same.

```
define_power_switch_cell -cells {XD1AT XD1BT} -
power_switchable VDD -power TVDD -stage_1_enable SLPIN1 -
stage_1_output SLPOUT1 -type header

update_power_switch_rule -name DSPCre -cells XD1T
```

The *define_power_switch_cell* command refers to `XD1AT`, while the *update_power_switch_rule* command refers to `XD1T`.

Default Severity Label

Warning

Rule Group

`CPF Check`

Reports and Related Files

None

CPFSEM_2

Checks the presence of the power domain in scope

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSEM_2* rule reports a violation if a power domain is not found in scope.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when a power domain is not found in the scope:

[FATAL] `Power domain <power-domain> not found in the scope <scope>`

Potential Issues

The power domain is not created while it is being accessed and specified with different power-strategies.

Consequences of not fixing

The power-intent cannot be analyzed thus no power-verification will be performed.

How to debug and fix

Create the power domain using the `create_power_domain -name <power domain name>` command before using them in different power strategies.

Example Code and/or Schematic

Consider the following CPF snippet:

```
update_power_domain -name XX -internal_power_net VDD
```

For the above CPF snippet, the *CPFSEM_2* rule reports the following violation because the power domain `XX` has not been created:

```
Power domain (XX) not found in the scope (top)
```

To resolve the above violation, create power domain using `create_power_domain` command, as follows:

```
create_power_domain –name XX
```

Default and Severity label

Fatal

Rule Group

`CPF Check`

Reports and Related files

None

CPFSEM_3

Checks if the default power domain is already present in scope

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSEM_3* rule reports a violation when default power domain is created while a default power domain is already present in the scope.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when a default power domain already present in scope:

[FATAL] `Default power domain <power-domain> already exists for scope <scope>`

Potential Issues

Default power domain is created while a default power domain is already present in the scope.

Consequences of not fixing

Relevant checks do not happen correctly for the power domains.

How to debug and fix

Specify only one power domain as default for a scope.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_power_domain -name V0  -default
create_power_domain -name VX  -default
```

For the above CPF snippet, the *CPFSEM_2* rule reports the following violation the default power domain VX is created while the default power domain V0 is created:

```
Default power domain ( V0 ) already exists for scope ( top )
```

To resolve the above violation, specify only one default power domain under a scope.

Default and Severity label

Fatal

Rule Group

CPF Check

Reports and Related files

None

CPFSEM_4

Checks if default power domain is not present in scope

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSEM_4* rule reports a violation when no default power domain is present in the scope.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when no default power domain is present in the scope:

[ERROR] `No default power domain present for design <design-name>`

Potential Issues

No default power domain in the scope.

Consequences of not fixing

The power-intent is not analyzed correctly as there is no default power domain.

How to debug and fix

Create the power domain using the `create_power_domain -name <power domain name> -default` command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_power_domain -name V0 -instances t1/inst3
create_power_domain -name V1 -instances t1/inst4  \
-shutoff_condition !isosig
```

For the above CPF snippet, the *CPFSEM_4* rule reports the following violation because there is no default power domain:

```
No default power domain present for design top
```

To resolve the above violation, specify a default power domain, as follows:

```
create_power_domain -name V0 –default \
-instances t1/inst3
```

Default and Severity label

Error

Rule Group

CPF Check

Reports and Related files

None

CPFSEM_5

Checks the name conflict in the same scope

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSEM_5* rule reports a violation when there is a name conflict in the same scope.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when there is a name conflict in the same scope:

[FATAL] `<string> named object already created in scope <scope> , cannot create power domain with same name in this scope`

Potential Issues

The same strings are used for naming in different CPF commands.

Consequences of not fixing

The power-intent cannot be analyzed thus no power verification is performed.

How to debug and fix

Avoid specifying same string for naming in different CPF commands.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_power_domain -name V0  -default
create_power_domain -name V0 -instances t1/inst3
```

For the above CPF snippet, the *CPFSEM_5* rule reports the following violation because the string `V0` is used twice as the name of power domains:

```
( V0 ) named object already created in scope ( top ) , cannot create power domain with same name in this scope
```

To resolve the above violation, avoid using same string “V0” again as a name of power domains.

Default and Severity label

Error

Rule Group

`CPF Check`

Reports and Related files

None

CPFSEM_6

Checks the presence of the nominal condition

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSEM_6* rule reports a violation when the nominal condition is not present.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when the nominal condition is not present:

[FATAL] `<nominal-condition-name> nominal condition not created earlier`

Potential Issues

The nominal conditions are not specified while they are being accessed by other CPF commands.

Consequences of not fixing

The relevant checks do not happen for the specified nominal conditions in other CPF commands.

How to debug and fix

Specify nominal conditions using the `create_nominal_conditions –name <nominal-condition-name> -voltage <voltage-value>` command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_power_mode -name PM1 \
-domain_conditions{V0@high}
```

For the above CPF snippet, the *CPFSEM_6* rule reports the following violation because the nominal condition `high` was not created previously:

```
'high' nominal condition not created earlier
```

To resolve the above violation, specify the nominal condition `high` as follows:

```
create_nominal_condition –name high
```

Default and Severity label

Fatal

Rule Group

`CPF Check`

Reports and Related files

None

CPFSEM_7

Checks the presence operating corner

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSEM_7* rule reports a violation when an operating corner is not present.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when an operating corner is not present:

[Error] <operating-corner-name> operating corner not created earlier

Potential Issues

The operating corners are not specified while they are being accessed by other CPF commands.

Consequences of not fixing

The relevant checks do not happen for the specified operating corners in other CPF commands.

How to debug and fix

Specify the operating corners using the `create_operating_corner –name <operating-corner-name> -voltage <voltage-value>` command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
reate_analysis_view -name av -mode md \
-domain_corners {V0@high_oc}
```

For the above CPF snippet, the *CPFSEM_7* rule reports the following violation because the operating corner `high_oc` was not created previously:

```
'high_oc' operating corner not created earlier
```

To resolve the above violation, specify the operating corner `high_oc`, as follows:

```
create_operating_corner –name high_oc
```

Default and Severity label

Error

Rule Group

`CPF Check`

Reports and Related files

None

CPFSEM_8

Checks if library set is defined

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSEM_8* rule reports a violation when the library set is not defined.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when the library set is not defined:

[FATAL] `<library-set-name> library set not defined`

Potential Issues

The library set is not created while it is being accessed and specified with other CPF commands.

Consequences of not fixing

Power analysis may not be done correct.

How to debug and fix

Specify the library set using the `define_library_set –name <library-set-name> -libraries <libraries-list>` command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_operating_corner -name high_oc -voltage 1.1 \
-library_set {set1}
```

For the above CPF snippet, the *CPFSEM_8* rule reports the following violation because the library set `set1` was not defined previously:

```
'set1' library set not defined
```

To resolve the above violation, specify library set `set1`, as follows:

```
define_library_set –name set1
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSEM_10

Checks if a ground net is defined

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSEM_10* rule reports a violation when a ground net is not defined.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when a ground net is not defined:

[FATAL] `<ground-net-name> ground net not found in the scope top`

Potential Issues

Ground net is not created while it is being accessed and specified with other CPF commands.

Consequences of not fixing

Relevant checks do not happen for the CPF command specified with the ground net.

How to debug and fix

Specify the ground net using the `create_ground_nets –name <ground-net-name>` command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
update_power_domain -name V1 -primary_power_net VDD \
-primary_ground_net VSSX
```

For the above CPF snippet, the *CPFSEM_10* rule reports the following violation because the ground net VSSX was not defined previously:

```
'VSSX' ground net not found in the scope top
```

To resolve the above violation, specify the ground net VSSX , as follows:

```
create_ground_nets –name VSSX
```

Default and Severity label

FATAL

Rule Group

CPF Check

Reports and Related files

None

CPFSEM_11

Checks the presence of power net

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSEM_11* rule reports a violation when a power net is not found.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when a power net is not found:

[FATAL] <power-net-name> power net not found in the scope top

Potential Issues

Power net is not created while it is being accessed and specified with other CPF commands.

Consequences of not fixing

Relevant checks do not happen for the CPF command specified with the power net.

How to debug and fix

Specify the power net using the `create_ power _nets –name <power -net-name>` command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
update_power_domain -name V1 -primary_power_net VDDX
```

For the above CPF snippet, the *CPFSEM_11* rule reports the following violation because the power net VDDX was not defined previously:

```
'VDDX' power net not found in the scope top
```

To resolve the above violation, specify the power net VDDX, as follows:

```
create_power_nets –name VDDX
```

Default and Severity label

FATAL

Rule Group

CPF Check

Reports and Related files

None

CPFSEM_12

Checks the presence of bias net

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSEM_12* rule reports a violation when a bias net is not found.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when a bias net is not found:

[ERROR] <bias-net-name> bias net not found in the scope top

Potential Issues

Bias net is not created while it is being accessed and specified with other CPF commands.

Consequences of not fixing

Relevant checks do not happen for the CPF command specified with the bias net.

How to debug and fix

Specify the bias net using the `create_ bias _net –name < bias -net-name>` command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
update_power_domain -name V1 -primary_power_net VDD \
-pmos_bias_net VDDX
```

For the above CPF snippet, the *CPFSEM_12* rule reports the following violation because the bias net VDDX was not defined previously:

```
'VDDX' bias net not found in the scope top
```

To resolve the above violation, specify the bias net VDDX, as follows:

```
create_bias_net –name VDDX
```

Default and Severity label

Error

Rule Group

CPF Check

Reports and Related files

None

CPFSEM_13

Checks the presence of isolation rule

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSEM_13* rule reports a violation when an isolation rule is not found.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when an isolation rule is not found:

[FATAL] `<isolation-rule> isolation rule not found in scope <scope>`

Potential Issues

The isolation rule is not specified while it is being accessed and specified with other CPF commands.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed where that isolation rule is specified.

How to debug and fix

Specify isolation rules using the `create_isolation_rule –name <rule-name>` command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
update_isolation_rules -names iso -location to
```

For the above CPF snippet, the *CPFSEM_13* rule reports the following violation because the isolation rule `iso` was not defined previously:

```
'iso' isolation rule not found in scope top
```

To resolve the above violation, specify isolation rule `iso`, as follows:

```
create_isolation_rule –name iso
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSEM_14

Checks the presence of level shifter rule

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSEM_14* rule reports a violation when a level shifter rule is not found.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when a level shifter rule is not found:

[FATAL] `<level-shifter-rule> level shifter rule not found in scope <scope>`

Potential Issues

The level shifter rule is not specified while it is being accessed and specified with other CPF commands.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed where that level shifter rule is specified.

How to debug and fix

Specify isolation rules using the `create_level_shifter_rule -name <rule-name>` command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
update_level_shifter_rules -names ls -location to
```

For the above CPF snippet, the *CPFSEM_14* rule reports the following violation because the level shifter rule `ls` was not defined previously:

```
'ls' level shifter rule not found in scope top
```

To resolve the above violation, specify level shifter rule `ls`, as follows:

```
create_level_shifter_rule –name ls
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSEM_15

Checks the presence of power mode

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSEM_15* rule reports a violation when a power mode is not found.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when a power mode is not found:

[FATAL] `<power-mode> power mode not found`

Potential Issues

Power mode is not specified while it is being accessed and specified with other CPF commands.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed where the power mode is specified.

How to debug and fix

Specify power mode using the `create_power_mode –name <power-mode-name>` command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_analysis_view -name av -mode md \
-domain_corners {V0@high_oc}
```

For the above CPF snippet, the *CPFSEM_15* rule reports the following violation because the power mode `md` was not defined previously:

```
'md' power mode not found
```

To resolve the above violation, specify power mode `md`, as follows:

```
create_power_mode –name md
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSEM_17

Checks the presence power switch rule

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSEM_17* rule reports a violation when the power switch rule is not found.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when the power switch rule is not found:

[FATAL] `<power-switch-rule> power switch rule not found in scope <scope>`

Potential Issues

The power switch rule is not specified while it is being accessed and specified with other CPF commands.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed where the power switch is specified.

How to debug and fix

Specify power switch rule using the `create_power_switch_rule -name <power-switch-name>` command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
update_power_switch_rule -name ps -cells PS1
```

For the above CPF snippet, the *CPFSEM_17* rule reports the following violation because the power switch rule `ps` was not defined previously:

```
'ps' power switch rule not found in scope top
```

To resolve the above violation, specify power switch rule `ps`, as follows:

```
create_power_switch_rule –name ps
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSEM_19

Checks if set_design is preceded by a set_instance command

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSEM_19* rule reports a violation when `set_design` command is not preceded by a `set_instance` command.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when `set_design` command is not preceded by a `set_instance` command:

[FATAL] `set_design <design-name> should be preceded by a set_instance command with argument`

Potential Issues

The `set_instance` command is not specified followed by the `set_design` CPF command.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify the design using the `set_instance <instance-name> -design <design-name>` command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
set_design des
end_design
```

For the above CPF snippet, the *CPFSEM_19* rule reports the following violation because the `set_instance` command is not specified for the design `des`:

```
set_design des should be preceded by a set_instance command
with argument
```

To resolve the above violation, specify the design `des`, as follows:

```
set_design des
set_instance I1 –design des
end_design
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSEM_21

Checks if set_design is specified for the current instance

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSEM_21* rule reports a violation when the `set_design` command is not specified for the current instance.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when the `set_design` command is not specified for the current instance:

[FATAL] `set_design not specified for the current instance <instance-name>`

Potential Issues

Current design is not specified in the `–design` argument in the `set_instance` command.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify the current design using the `set_instance <instance-name> -design <design-name>` command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
set_design top1
set_instance I1
end_design top1
```

For the above CPF snippet, the *CPFSEM_21* rule reports the following violation because the -design argument of the `set_instance` command is not specified for the current instance:

```
set_design not specified for the current instance I1
```

To resolve the above violation, specify current design in the `set_instance` command, as follows:

```
set_instance I1 –design top1
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_1

Checks if mutually exclusive options are specified

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_1* rule reports a violation when mutually exclusive options are specified.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when mutually exclusive options are specified:

[FATAL] `Mutually exclusive options <options> specified in command <command>`

Potential Issues

Mutually exclusive options are specified in CPF commands.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Avoid specifying mutually exclusive options in CPF commands.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_power_switch_rule -name psw1 -domain V1 \
-external_power_net VDD -external_ground_net VSS
```

For the above CPF snippet, the *CPFSTX_1* rule reports the following violation because `-external_power_net` and `-external_ground_net` are mutually exclusive options in the `create_power_switch_rule` command:

```
Mutually exclusive options (-external_power_net and -external_ground_net) specified in command create_power_switch_rule
```

To resolve the above violation, specify either -external_power_net or -external_ground_net option at a time. Avoid specifying both options at once.

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_2

Checks if at least one mandatory argument is specified

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_2* rule reports a violation when at least one mandatory argument is not specified in CPF commands.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when at least one mandatory argument is not specified in CPF commands:

[FATAL] `At least one argument among <arguments> should be specified in command <command>.`

Potential Issues

At least one mandatory argument is not specified in CPF commands.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify at least one mandatory argument in CPF commands.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_power_domain -name V2
```

For the above CPF snippet, the *CPFSTX_2* rule reports the following violation because none of the mandatory arguments are specified in the `create_power_domain` command:

```
At least one argument among (-default, -instances, -
boundary_ports) should be specified in command
create_power_domain
```

To resolve the above violation, specify at least one of the `-default`, `-instances`, or `-boundary_ports` arguments in the `create_power_domain` command.

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_4

Checks the presence of an invalid domain condition

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_4* rule reports a violation when an invalid domain condition is specified in CPF commands.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when an invalid domain condition is specified in CPF commands:

[FATAL] `Invalid domain condition <domain-condition> in command <command>`

Potential Issues

An invalid domain condition is specified in CPF commands.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify a valid domain condition correctly in CPF commands.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_power_mode -name PM2 \
-domain_conditions {TOP@off PD1&on_high}
```

For the above CPF snippet, the *CPFSTX_4* rule reports the following violation because the domain condition `PD1&on_high` is invalid:

```
Invalid domain condition PD1&on_high in command
create_power_mode
```

To resolve the above violation, specify domain condition correctly, as shown below:

```
create_power_mode -name PM2 \
-domain_conditions {TOP@off PD1@on_high}
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_5

Checks the presence of a wrong array naming style

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_5* rule reports a violation when a wrong array naming style is used in the `set_array_naming_style` CPF command.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when a wrong array naming style is used in the `set_array_naming_style` CPF command:

[FATAL] `Wrong array naming style <style> in command set_array_naming_style`

Potential Issues

A wrong array naming style is used in the `set_array_naming_style` CPF command.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify a correct array naming style in the `set_array_naming_style` CPF command using the `[character]%d[character]` format.

Example Code and/or Schematic

Consider the following CPF snippet:

```
set_array_naming_style a&da
```

For the above CPF snippet, the *CPFSTX_5* rule reports the following violation because the array naming style `a&da` is not valid:

```
Wrong array naming style a&da in command set_array_naming_style
```

To resolve the above violation, use correct array naming style, as shown below:

```
set_array_naming_style a%da
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_6

Checks the presence of a wrong hierarchy delimiter

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_6* rule reports a violation when a wrong hierarchy delimiter is used in the `set_hierarchy_separator` CPF command.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when a wrong hierarchy delimiter is used in the `set_hierarchy_separator` CPF command:

[FATAL] `Wrong hierarchy delimiter <delimiter> in command set_hierarchy_separator`

Potential Issues

A wrong hierarchy delimiter is used in the `set_hierarchy_separator` CPF command.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify only supported hierarchy separators in the `set_hierarchy_separator` CPF command. Currently supported hierarchy separators are period (.), slash (/), and colon (:).

Example Code and/or Schematic

Consider the following CPF snippet:

```
set_hierarchy_separator *
```

For the above CPF snippet, the *CPFSTX_6* rule reports the following violation because the specified hierarchy separator `*` is not valid:

```
Wrong hierarchy delimiter * in command set_hierarchy_separator
```

To resolve the above violation, specify only supported hierarchy separators, as shown below:

```
set_hierarchy_separator “.” or
set_hierarchy_separator “/” or
set_hierarchy_separator “:”
```

Default and Severity label

FATAL

Rule Group

CPF Check

Reports and Related files

None

CPFSTX_7

Checks the presence of an invalid value

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_7* rule reports a violation when an invalid type of value is specified in CPF commands.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when an invalid type of value is specified in CPF commands:

[FATAL] `Invalid type <value> for command <command>`

Potential Issues

An invalid type is specified in CPF commands.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify a valid type of value in CPF commands.

Example Code and/or Schematic

Consider the following CPF snippet:

```
set_power_unit x
```

For the above CPF snippet, the *CPFSTX_7* rule reports the following violation because an invalid type of value `x` is specified:

```
Invalid type x for command set_power_unit
```

To resolve the above violation, specify a valid value type, as shown below:

```
set_power_unit pW | nW | uW | mW | W
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_8

Checks the presence of an invalid register naming style

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_8* rule reports a violation when an invalid register naming style is used in the `set_register_naming_style` CPF command.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when an invalid register naming style is used in the `set_register_naming_style` CPF command:

[FATAL] `Invalid register naming style <style> in command set_register_naming_style`

Potential Issues

An invalid register naming style is used in the `set_register_naming_style` CPF command.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify a valid register naming style in the `set_register_naming_style` CPF command, using the `[string%s]` format.

Example Code and/or Schematic

Consider the following CPF snippet:

```
set_register_naming_style reg
```

For the above CPF snippet, the *CPFSTX_8* rule reports the following violation because the register naming style `reg` is not valid:

```
Invalid register naming style reg in command
set_register_naming_style
```

To resolve the above violation, specify a valid register naming style, as shown below:

```
set_register_naming_style reg%s
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_9

Checks if the arguments are specified together

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_9* rule reports a violation when arguments are not specified together or only one is specified in a CPF command.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when arguments are not specified together or only one is specified in a CPF command:

[FATAL] `All arguments <arguments> or none of them, should be present in command <command>`

Potential Issues

Arguments are not specified together or only one is specified in CPF command.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify the arguments together or do not specify any of them in the CPF command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
update_power_switch_rule -name ps -cells PS1
```

For the above CPF snippet, the *CPFSTX_9* rule reports the following violation because only the `-cells` argument is specified:

```
All arguments (-cells and -library_set ) or none of them,
should be present in command update_power_switch_rule
```

To resolve the above violation, specify both arguments, `-cells` and `-library_set`, together or specify none of them, as shown below:

```
update_power_switch_rule -name ps -cells PS1 –library_set
lib_set
```

or

```
update_power_switch_rule -name ps
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_10

Checks if wrong number of arguments are specified

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_10* rule reports a violation when an incorrect number of values are specified for the arguments in CPF commands.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when an incorrect number of values are specified for the arguments in CPF commands:

[FATAL] Incorrect number of values for argument <voltage> in command <command>

Potential Issues

An incorrect number of values are specified for the arguments in CPF commands.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify a valid number of values for the arguments in CPF commands.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_power_nets -nets VDD -voltage 1.1:1.2:1.3
```

For the above CPF snippet, the *CPFSTX_10* rule reports the following violation because number of values is not valid for the `-voltage` argument:

```
Incorrect number of values for argument voltage in command create_power_nets
```

To resolve the above violation, specify a valid number of values for the `-voltage` argument, as shown below:

```
create_power_nets -nets VDD -voltage 1.1:1.2
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_11

Checks if the specified value is out of range

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_11* rule reports a violation when the value specified in the argument is out of range.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when the value specified in the argument is out of range:

[FATAL] `Value for argument <argument> is out of range in command <command>`

Potential Issues

The value specified in the argument is out of range.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify a value for the argument that is in a valid range.

Example Code and/or Schematic

Consider the following CPF snippet:

```
cupdate_power_mode -name PM2 \
-activity_file file1 -activity_file_weight 110
```

For the above CPF snippet, the *CPFSTX_11* rule reports the following violation because the value `110` is out of range:

```
Value for argument activity_file_weight is out of range in command update_power_mode
```

To resolve the above violation, specify a value for the `-activity_file_weight` argument that is in the valid range. A valid range is 0 to 100.

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_13

Checks if an invalid option is specified for command

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_13* rule reports a violation when an invalid option is specified for command.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when an invalid option is specified for command:

[FATAL] `Invalid option <option> for command <command>`

Potential Issues

An invalid option is specified for a CPF command.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify a valid options in CPF commands.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_power_mode -names X1
```

For the above CPF snippet, the *CPFSTX_13* rule reports the following violation because the `-names` option is not valid:

```
Invalid option -names for command create_power_mode
```

To resolve the above violation, specify valid option, as shown below:

```
create_power_mode –name X1
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_14

Checks if no value is specified for argument

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_14* rule reports a violation when no value is specified for an argument in CPF commands.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when no value is specified for an argument in CPF commands:

[FATAL] `No value specified for argument <argument> in command <command>`

Potential Issues

No value is specified for an argument in CPF commands.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify valid values for the arguments in CPF commands.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_power_mode -name \
-domain_conditions {TOP@off PD1@on_high}
```

For the above CPF snippet, the *CPFSTX_14* rule reports the following violation because no value is specified for the `-name` argument:

```
No value specified for argument -name in command
create_power_mode
```

To resolve the above violation, specify valid value for the `–name` argument, as shown below:

```
create_power_mode –name PM1 \
-domain_conditions {TOP@off PD1@on_high}
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_15

Checks if an invalid value type is specified

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_15* rule reports a violation when an invalid value type is specified for an argument in CPF commands.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when an invalid value type is specified for an argument in a CPF command:

[FATAL] Wrong value <value> specified for argument <argument> in command <command>

Potential Issues

An invalid value type is specified for an argument in CPF commands.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify a valid value type for the argument in the CPF command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_isolation_rule -name iso1 -isolation_output x
```

For the above CPF snippet, the *CPFSTX_15* rule reports the following violation because the value `x` is not a valid value:

```
Wrong value x specified for argument -isolation_output in
command create_isolation_rule
```

To resolve the above violation, specify a valid value for the `–isolation_output` argument, as shown below:

```
create_isolation_rule -name iso1 \
-isolation_output high | low | hold | tristate
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_16

Checks if an invalid value is specified for argument

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_16* rule reports a violation when an invalid value is specified for an argument in CPF commands.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when an invalid value is specified for an argument in the CPF command:

[FATAL] `Invalid value <value> specified for argument <argument> in command <command>`

Potential Issues

An invalid value is specified for an argument in CPF commands.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify a valid value for the argument in the CPF command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
set_switching_activity -all -toggle_percentage 110
```

For the above CPF snippet, the *CPFSTX_16* rule reports the following violation because the value `110` is not a valid value:

```
Invalid value "110" specified for argument -toggle_percentage
in command set_switching_activity
```

To resolve the above violation, specify a valid value for the –`toggle_percentage` argument, as shown below:

```
set_switching_activity -all -toggle_percentage 50 (value
must be a float between 0 and 100)
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_17

Checks if a mandatory argument is missing

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_17* rule reports a violation when a mandatory argument is missing in CPF commands.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when a mandatory argument is missing in the CPF command:

[FATAL] `Mandatory argument <argument> missing in command <command>`

Potential Issues

A mandatory argument is missing in the CPF command.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify the mandatory arguments in CPF commands.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_isolation_rule -name iso_r1 -from V1\
-isolation_condition isosig -isolation_output low
```

For the above CPF snippet, the *CPFSTX_17* rule reports the following violation because the mandatory argument `-isolation_condition` missing:

```
Mandatory argument -isolation_condition missing in command
create_isolation_rule
```

To resolve the above violation, specify the mandatory argument `-isolation_condition`, as shown below:

```
create_isolation_rule -name iso_r1 -from V1 \
-isolation_condition isosig -isolation_output low \
-isolation_condition inst1.en
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_19

Checks the presence of a wildcard reference

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_19* rule reports a violation when wildcard characters are specified in an argument in CPF commands.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when a wildcard character is specified in an argument in the CPF command:

[FATAL] `Argument <argument> of <command> cannot contain wildcard character(s)`

Potential Issues

Wildcard characters are specified in an argument in CPF commands.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Avoid specifying wildcard characters in arguments where they are not supported.

Example Code and/or Schematic

Consider the following CPF snippet:

```
update_nominal_condition -name cnd*
```

For the above CPF snippet, the *CPFSTX_19* rule reports the following violation because a wildcard character * is specified in the -name argument:

```
Argument -name of create_nominal_condition cannot contain
wildcard character(s)
```

To resolve the above violation, specify the –name argument without wildcard characters, as shown below:

```
update_nominal_condition -name cnd
```

Default and Severity label

FATAL

Rule Group

CPF Check

Reports and Related files

None

CPFSTX_20

Checks the presence of hierarchical separator

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_20* rule reports a violation when a hierarchical separator is specified in an argument in CPF commands.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when a hierarchical separator is specified in an argument in the CPF command:

[FATAL] `Argument <argument> of <command> cannot contain hierarchical separator character(s)`

Potential Issues

A hierarchical separator is specified in an argument in the CPF command.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Avoid specifying hierarchical separators in arguments where hierarchical separators are not supported.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_isolation_rule -name iso1/x
```

For the above CPF snippet, the *CPFSTX_20* rule reports the following violation because the hierarchical separator / is specified in the `-name` argument:

```
Argument -name of create_isolation_rule cannot contain
hierarchical seperator character(s)
```

To resolve the above violation, specify the `–name` argument without hierarchical separator, as shown below:

```
create_isolation_rule -name iso1x
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_22

Checks the presence of top design

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_22* rule reports a violation when no top design is specified.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when no top design is specified:

[FATAL] `No top design specified`

Potential Issues

No top design is specified.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify the top design using the `set_design <module>` command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
set_cpf_version 1.1
create_power_domain -name TOP
```

For the above CPF snippet, the *CPFSTX_22* rule reports the following violation because the top design is not specified:

```
No top design specified
```

To resolve the above violation, specify the top design, as shown below:

```
set_cpf_version 1.1
set_design top
create_power_domain -name TOP
```

Default and Severity label

FATAL

Rule Group

CPF Check

Reports and Related files

None

CPFSTX_23

Checks the presence of control group

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_23* rule reports a violation when no control group is specified.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when no control group is specified:

[FATAL] `<control-group> control group in scope <scope> not found`

Potential Issues

No control group is specified.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify the control group using the `set_power_mode_control_group -name <group-name>` CPF command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_power_mode -name PM2 -group_modes {group1@off}
```

For the above CPF snippet, the *CPFSTX_23* rule reports the following violation because the control group `group1` was not specified previously:

```
group1 control group in scope top not found
```

To resolve the above violation, specify the `control_group`, as shown below:

```
set_power_mode_control_group –name group1
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_24

Checks the presence of invalid commands in macro model

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_24* rule reports a violation when an invalid command is specified in the macro model.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when an invalid command is specified in the macro model:

[FATAL] `Invalid command <command> inside set_macro_model and end_macro_model block`

Potential Issues

An invalid command is specified in macro model.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify only valid commands inside `set_macro_model`.

Valid commands are:

- create_isolation_rule
- create_mode_transition

- create_nominal_condition
- create_power_domain
- create_power_mode
- create_power_switch_rule
- create_state_retention_rule
- set_floating_ports
- set_input_voltage_tolerance
- set_wire_feedthrough_ports
- update_power_domain

Example Code and/or Schematic

Consider the following CPF snippet:

```
set_macro_model mac1
create_power_nets -nets VDD -voltage 1.0
end_macro_model mac1
```

For the above CPF snippet, the *CPFSTX_24* rule reports the following violation because the `create_power_nets` command is not a valid command inside the macro model:

```
Invalid command "create_power_nets" inside set_macro_model and end_macro_model block
```

To resolve the above violation, specify only valid CPF commands inside macro models.

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_25

Checks the presence of macro model

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_25* rule reports a violation when the macro model specified is not found.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when the macro model specified is not found:

[FATAL] `Macro Model <macro-model> not present, referred from command <command>`

Potential Issues

The specified macro model is not found.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify the macro model using the `set_macro_model <macro_cell>` command before using it.

Example Code and/or Schematic

Consider the following CPF snippet:

```
set_instance inst1 -model mac1
```

For the above CPF snippet, the *CPFSTX_25* rule reports the following violation because the macro model `mac1` was not specified previously:

```
Macro Model mac1 not present, referred from command
"set_instance"
```

To resolve the above violation, specify the macro model, as shown below:

```
set_macro_model mac1
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_26

Checks the presence of the specified file

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_26* rule reports a violation when the file specified is not found.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when the file specified is not found:

[FATAL] `File <file-name> not present or correct permissions are not set`

Potential Issues

The specified file is not found.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify the files in the CPF commands that exist and can be found.

Example Code and/or Schematic

Consider the following CPF snippet:

```
include ./const1.cpf
```

For the above CPF snippet, the *CPFSTX_26* rule reports the following

violation because the specified file `const1.cpf` does not exist:

```
File ./const1.cpf not present or correct permissions are not set
```

To resolve the above violation, specify a file with include command that already exists and is readable.

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_27

Checks the presence of the specified design

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_27* rule reports a violation when the design specified is not found.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when the design specified is not found:

[FATAL] `Design <design-name> not present, referred from command <command>`

Potential Issues

The specified design is not found.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify the design using the `set_design <module>` command before using it.

Example Code and/or Schematic

Consider the following CPF snippet:

```
set_instance inst1 -design inst1
```

For the above CPF snippet, the *CPFSTX_27* rule reports the following violation because the specified design `inst1` was not declared previously:

```
Design inst1 not present, referred from command "set_instance"
```

To resolve the above violation, specify the design before using it, as shown below:

```
set_design inst1
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_28

Checks if an invalid command is specified in control group

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_28* rule reports a violation when an invalid command is specified inside the control group.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when an invalid command is specified inside the control group:

[FATAL] Invalid command <command> inside set_power_mode_control_group and end_power_mode_control_group block

Potential Issues

An invalid command is specified inside the control group.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify only valid commands inside the control group. Valid commands are:

- create_analysis_view

- create_mode_transition
- create_power_mode
- update_power_mode

Example Code and/or Schematic

Consider the following CPF snippet:

```
set_power_mode_control_group -name g1 -domains PD1 -groups g1
create_power_domain -name PD -instances inst2
end_power_mode_control_group
```

For the above CPF snippet, the *CPFSTX_28* rule reports the following violation because the `create_power_domain` is not a valid command inside the `control_group`:

```
Invalid command "create_power_domain" inside
set_power_mode_control_group and end_power_mode_control_group
block
```

To resolve the above violation, specify only valid commands inside the control group.

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_29

Checks if an invalid object name is specified

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_29* rule reports a violation when an invalid object name is specified.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when an invalid object name is specified:

[FATAL] `Invalid name ( NULL ) specified for object <object-name>`

Potential Issues

An invalid value is specifies for an argument in the CPF command.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify a valid name for the CPF object.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_power_mode -name -domain_conditions {TOP@off
```

```
PD1@on_high}
```

For the above CPF snippet, the *CPFSTX_29* rule reports the following violation because NULL is specified for the `–name` option:

```
Invalid name ( NULL ) specified for object power mode
```

To resolve the above violation, specify a valid name for the `-name` option, as shown below:

```
create_power_mode –name PM1 -domain_conditions {TOP@off
PD1@on_high}
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_30

Checks if an invalid group view is specified

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_30* rule reports a violation when an invalid group view is specified in CPF commands.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when an invalid group view is specified in the CPF command:

[FATAL] `Invalid group view <group-view> in command <command>`

Potential Issues

An invalid group view is specified in the CPF command.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify a valid group view in the CPF command using the `group_name@view_name` format.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_analysis_view -name AV1 -group_views {g1&V1}
```

For the above CPF snippet, the *CPFSTX_30* rule reports the following violation because the group view `g1&V1` is not valid:

```
Invalid group view g1&V1 in command create_analysis_view
```

To resolve the above violation, specify a valid group view, as shown below:

```
create_analysis_view -name AV1 -group_views {g1@V1}
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_31

Checks the presence of an analysis view

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_31* rule reports a violation when analysis view specified is not defined within the control group.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when analysis view specified is not defined within the control group:

[FATAL] `Analysis view <analysis-view> is not defined within control group <group-name>`

Potential Issues

The specified analysis view is not defined within the control group.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Define an analysis view within the control group before specifying it in the CPF command.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_analysis_view -name AV1 -group_views {g1@V1} -mode M1
```

For the above CPF snippet, the *CPFSTX_31* rule reports the following violation because the value `110` is not valid:

```
Analysis view V1 is not defined within control group g1
```

To resolve the above violation, specify the analysis view `V1` before using it, as shown below:

```
create_analysis_view -name V1
create_analysis_view -name AV1 -group_views {g1@V1}
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_32

Checks if an incorrect argument format is specified in some switch of command

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_32* rule reports a violation when an incorrect argument format is specified in a switch of a CPF command.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when an incorrect argument format is specified in a switch of a CPF command:

[FATAL] `Incorrect argument format specified in switch <switch-name> of command <command>`

Potential Issues

An incorrect argument format is specified in a switch of CPF commands.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify only valid argument formats in switches of the CPF commands.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_power_domain -name TOP –default -
active_state_conditions x
```

For the above CPF snippet, the *CPFSTX_32* rule reports the following violation because the specified argument format `x` is not correct:

```
Incorrect argument format specified in switch -
active_state_conditions of command create_power_domain
```

To resolve the above violation, specify a correct argument format, as shown below:

```
create_power_domain -name TOP –default \
-active_state_conditions cnd@en
(Correct format is nominal_condition_name@expression)
```

Default and Severity label

FATAL

Rule Group

CPF Check

Reports and Related files

None

CPFSTX_33

Checks if an incorrect design name is specified in end_design

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_33* rule reports a violation when an incorrect design name is specified in end design.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when an incorrect design name is specified in end design:

[FATAL] `Incorrect design name <design-name> in end_design`

Potential Issues

An incorrect design name is specified in end design.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify the correct design name in end design.

Example Code and/or Schematic

Consider the following CPF snippet:

```
Set_design top
```

```
end_design topx
```

For the above CPF snippet, the *CPFSTX_33* rule reports the following violation because the specified design name `topx` is not correct:

```
Incorrect design name "topx" in end_design
```

To resolve the above violation, specify the correct design with end_design, as shown below:

```
set_design top
end_design top
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_34

Checks if an incorrect model name is specified in end_macro_model

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_34* rule reports a violation when an incorrect model name is specified in end_macro_model.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when an incorrect model name is specified in end_macro_model:

[FATAL] `Incorrect model name <model-name> in end_macro_model`

Potential Issues

An incorrect model name is specified in end_macro_model.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify the correct design name in end_macro_model.

Example Code and/or Schematic

Consider the following CPF snippet:

```
set_macro_model mac1
end_macro_model mac
```

For the above CPF snippet, the *CPFSTX_34* rule reports the following violation because the specified model name `mac` is not correct:

```
Incorrect model name "mac" in end_macro_model
```

To resolve the above violation, specify the correct macro model with `end_macro_model`, as shown below:

```
set_macro_model mac1
end_macro_model mac1
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_35

Checks if set_macro_model is specified for the current block

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_35* rule reports a violation when set_macro_model is not specified for the current block.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when set_macro_model is not specified for the current block:

[FATAL] `set_macro_model not specified for the current block`

Potential Issues

The set_macro_model is not specified or the current block.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify the set_macro_model for the current block before ending it.

Example Code and/or Schematic

Consider the following CPF snippet:

```
end_macro_model mac
```

For the above CPF snippet, the *CPFSTX_35* rule reports the following violation because the specified macro model `mac` was not previously set:

```
set_macro_model not specified for the current block
```

To resolve the above violation, specify the macro model before ending it, as shown below:

```
set_macro_model mac
end_macro_model mac
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_36

Checks if an unknown command is specified

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_36* rule reports a violation when an unknown command is specified.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when an unknown command is specified:

[FATAL] Unknown command <command>

Potential Issues

An unknown command is specified.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify only valid and known commands.

Example Code and/or Schematic

Consider the following CPF snippet:

```
command_xx
```

For the above CPF snippet, the *CPFSTX_36* rule reports the following

violation because the specified command `command_xx` is unknown:

```
Unknown command command_xx
```

To resolve the above violation, specify only valid and known commands.

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_38

Checks if power domain is visible inside control group

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_38* rule reports a violation when a specified domain is not visible inside control group.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when a specified domain is not visible inside control group:

[FATAL] `Power domain <domain-name> not visible inside control group <control-group>`

Potential Issues

The specified domain is not visible inside control group.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Only specify domains in the CPF commands that are visible inside control groups.

Example Code and/or Schematic

Consider the following CPF snippet:

```
set_power_mode_control_group -name g1 -domains PD1
create_power_mode -name PM3 \
-domain_conditions {TOP@off}
```

For the above CPF snippet, the *CPFSTX_38* rule reports the following violation because the specified domain `TOP` is not visible inside the control group `g1`:

```
Power domain TOP not visible inside control group .g1
```

To resolve the above violation, specify the domain TOP inside control group, as shown below:

```
set_power_mode_control_group -name g1 -domains TOP
create_power_mode -name PM3 \
-domain_conditions {TOP@off}
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_39

Checks if control group is visible inside control group

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_39* rule reports a violation when a control group is not visible inside a control group.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when a control group is not visible inside a control group:

[FATAL] `Control group <control-group> not visible inside control group <control-group>`

Potential Issues

The control group is not visible inside a control group.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify the control group inside a control group to be visible.

Example Code and/or Schematic

Consider the following CPF snippet:

```
set_power_mode_control_group -name g1 -domains PD1 \
-groups g1
set_power_mode_control_group -name gx -domains PD1 \
-groups gx
create_power_mode -name PM4 -group_modes {g1@PM1}
end_power_mode_control_group
```

For the above CPF snippet, the *CPFSTX_39* rule reports the following violation because the specified control group `g1` is not visible inside the control group `gx`:

```
Control group g1 not visible inside control group .gx
```

To resolve the above violation, specify the control group inside a control group to be visible, as shown below:

```
set_power_mode_control_group -name g1 -domains PD1 \
-groups g1
set_power_mode_control_group -name g1 -domains PD1 \
-groups g1
create_power_mode -name PM4 -group_modes {g1@PM1}
end_power_mode_control_group
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_40

Checks if set_instance without instance name contains parameters

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_40* rule reports a violation when set_instance is specified with parameters without instance.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when set_instance is specified with parameters without instance:

[FATAL] `set_instance without an instance name cannot contain other parameters`

Potential Issues

The set_instance is specified with parameters without instance.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Specify instance option in set_instance before specifying other parameters.

Example Code and/or Schematic

Consider the following CPF snippet:

```
set_instance -model mac1
```

For the above CPF snippet, the *CPFSTX_40* rule reports the following violation because the instance option is not specified with set_instance:

```
set_instance without a instance name cannot contain other parameters
```

To resolve the above violation, specify an instance option with set_instance, as shown below:

```
set_instance instance inst1 -model mac1
```

Default and Severity label

FATAL

Rule Group

CPF Check

Reports and Related files

None

CPFSTX_41

Checks the presence of a command outside set_macro_model and end_macro_model block

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_41* rule reports a violation when a specified command is not allowed outside the macro model.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when a specified command is not allowed outside the macro model:

[FATAL] Command <command> not allowed outside set_macro_model and end_macro_model block

Potential Issues

The specified command is not allowed outside the macro model.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Avoid specifying the command outside macro model.

Example Code and/or Schematic

Consider the following CPF snippet:

```
set_input_voltage_tolerance -ports in
```

For the above CPF snippet, the *CPFSTX_41* rule reports the following violation because the specified command `set_input_voltage_tolerance` is not allowed outside `set_macro_model` and `end_macro_model` blocks:

```
Command set_input_voltage_tolerance not allowed outside
set_macro_model and end_macro_model block
```

To resolve the above violation, specify the `set_input_voltage_tolerance` command inside macro model, as shown below:

```
set_macro_model mac
set_input_voltage_tolerance -ports in
end_macro_model mac
```

Default and Severity label

FATAL

Rule Group

`CPF Check`

Reports and Related files

None

CPFSTX_42

Checks if power net is specified as power supply net to an always on power domain

When to Use

This is a setup rule and always runs by default.

Description

The *CPFSTX_41* rule reports a violation when a switched power net is specified as a power supply net to an always on power domain.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when a switched power net is specified as a power supply net to an always on power domain:

[ERROR] Switched power net <power-net> specified as power supply net to always on power domain <domain-name>

Potential Issues

The switched power net is specified as power supply net to an always on power domain.

Consequences of not fixing

Power intent cannot be analyzed, thus no power verification is performed.

How to debug and fix

Avoid specifying switched power net to an always on domain.

Example Code and/or Schematic

Consider the following CPF snippet:

```
create_power_nets -nets VDD -voltage 1.0 -
external_shutoff_condition {iso}
update_power_domain -name TOP -primary_power_net VDD
```

For the above CPF snippet, the *CPFSTX_42* rule reports the following violation because the switched power net VDD is specified to the always on domain TOP:

```
Switched power net 'VDD' specified as power supply net to
always on power domain TOP
```

To resolve the above violation, avoid specifying switched power nets to the always on domain TOP.

Default and Severity label

Error

Rule Group

CPF Check

Reports and Related files

None

UPF Check Rules

The UPF Check rules are as follows:

Rule	Reports
checkUPF_existence	If design objects (instances/nets/pins) specified with UPF commands do not exist in design
UPF_lowpower01	Reports if top design unit specified with the set_design_top command does not match with current_design in SGDC
UPF_lowpower02	Checks relationship of power nets between commands 'set_domain_supply_net', 'connect_supply_net' and 'add_port_state
UPF_lowpower03	If the default power domain for top design unit is not specified
UPF_lowpower04	If the supply port/net in 'add_pst_state' command is in invalid state.
UPF_lowpower05	Checks the set_isolation_control and map_isolation_cell commands
UPF_lowpower06	Checks the existence of the map_level_shifter_cell command
UPF_lowpower07	Checks the create_power_switch and map_power_switch commands in the library
UPF_lowpower08	Checks the improperly defined create_pst commands
UPF_lowpower09	If supply port or supply net defined in UPF file is either undriven or multiple driven
UPF_lowpower10	If different isolation sense is specified for an isolation signal of a domain
UPF_lowpower11	If supply net is not declared for the associated domain where supply net is used
UPF_lowpower12	When appropriate isolation/level shifter/repeater strategy on domain element is not specified
UPF_lowpower13	If the power switch output port do not have same voltage values as parent supply
UPF_lowpower14	The mismatch between information of supply nets and their states in power state tables is complete
UPF_lowpower15	The multi-supply cells that do not have connect_supply_net

Rule	Reports
UPF_lowpower16	The relationship between the bias net and the primary supply net
UPF_lowpower17	Isolation power nets that are not declared in the domain in which the isolation cell will be inferred
UPF_lowpower18	Retention power/ground net specified in the *set_retention* command with primary power/ground supply net of domain
UPF_lowpower19	Checks for conflict between save/restore information specified in *set_retention* and the retention cells available in library
UPF_lowpower20	Checks for the supported set_port_attributes command options
UPF_lowpower21	If the source or destination supplies are not available in the domain where level shifter cell will be inferred
UPF_lowpower22	Reports if the supply of a repeater is not matching the supply specified in the UPF command
UPF_lowpower23	Checks the set_equivalent UPF command
UPF_lowpower24	Checks the arguments of the set_design_attributes UPF command
UPF_lowpower25	Checks mismatch between voltage _map and connect_supply_net
UPF_lowpower26	Checks inconsistencies between the given liberty file and the corresponding UPF file
UPF_lowpower27	Validates the arguments of the create_power_switch command
UPF_lowpower28	Reports duplicate and conflicting set_related_supply_net constrains
UPFSEM_1	Checks the presence of the specified power domain
UPFSEM_2	Checks the presence of the supply port in scope
UPFSEM_3	Checks the presence of PST
UPFSEM_4	Checks the presence of the supply net
UPFSEM_5	Checks the presence of the isolation in the specified domain
UPFSEM_6	Checks the presence of retention in the specified domain

Rule	Reports
UPFSEM_7	Checks the presence of the acknowledge port specified in power switch
UPFSEM_8	Checks the presence of the level shifter specified for the power domain
UPFSEM_9	Checks the presence of the power switch specified for the domain
UPFSEM_11	Checks the presence of support port/net specified in the PST
UPFSEM_12	Checks the presence of the input supply port for power switch
UPFSEM_13	Checks if an element is already owned by another power domain
UPFSEM_14	Checks if a net already exists
UPFSEM_15	Checks if a port already exists
UPFSEM_16	Checks if the named object already exists
UPFSEM_18	Checks if invalid design elements are found in element list
UPFSEM_19	Checks the presence of a supply set referred in command
UPFSEM_21	Checks if the supply set already exists
UPFSEM_22	Checks the presence of the net referred
UPFSEM_23	Checks if an invalid logic net is specified
UPFSEM_24	Checks if an invalid supply set is specified
UPFSEM_25	Checks if an invalid function-net pair is provided in supply set
UPFSEM_29	Checks if an invalid port is specified, cannot be found in current scope
UPFSEM_31	Checks if same string is specified in both off and full_on tools
UPFSEM_32	Checks if an invalid handle is specified for supply set
UPFSEM_33	Checks if an invalid value is specified for a parameter in command
UPFSEM_34	Checks if an invalid command is specified inside the begin_power_model command

Rule	Reports
UPFSEM_35	Checks whether the end_power_model command is used with the begin_power_model command
UPFSEM_36	Checks if an invalid supply set pair is specified in supply map of the apply_power_model command
UPFSEM_37	Checks if an invalid value is specified in the set_partial_on_translation command
UPFSEM_38	Checks if an invalid supply set handle is provided in the add_power_state command
UPFSEM_39	Checks if UPF versions is specified as 2.1 in the upf_version command
UPFSEM_40	Ensures that no UPF command is used in the sourced Tcl file
UPFSEM_41	Checks the missing ack_port argument in the create_power_switch UPF command
UPFSEM_42	Checks for the command specified in the disallow_upf_command constraint
UPFSEM_43	Checks if a command option is specified in the disallow_upf_command constraint
UPFSEM_45	Checks if the same net is specified as power and ground in the set_isolation command
UPFSEM_46	Checks supply set function and pg_type of the ports given in the -connect option of the connect_supply_set command
UPFSEM_47	Checks for unsupported versions of UPF
UPFSEM_48	Checks for the use of deprecated UPF commands
UPFSEM_49	Checks for the use of deprecated UPF command options
UPFSEM_50	Checks for modification of power states that have been marked as complete
UPFSEM_51	Checks for power states where the named object does not match the specified -supply or -domain option
UPFSEM_52	Checks that the named object already exists when the -update option is used
UPFSTX_1	Checks if an invalid port state is specified
UPFSTX_2	Checks if an state value is specified

Rule	Reports
UPFSTX_4	Checks if pg type is simultaneously specified with mutually exclusive port/pin
UPFSTX_5	Checks if an invalid HDL type is specified in the definition of vct
UPFSTX_6	Checks if an invalid UPF value is specified in the definition of vct
UPFSTX_13	Checks if an invalid signal type is specified in retention strategy
UPFSTX_15	Checks if an invalid switch is specified for command
UPFSTX_16	Checks if no value is specified for argument
UPFSTX_17	Checks if an invalid value type is specified for argument
UPFSTX_18	Checks if a mandatory argument is missing
UPFSTX_19	Checks the presence of an invalid PST state specification
UPFSTX_20	Checks if an invalid port-net pair is specified
UPFSTX_21	Checks if an invalid HDL-UPF value pair is specified
UPFSTX_22	Checks if an invalid port is specified for a power switch
UPFSTX_23	Checks if an invalid state is specified for a power switch
UPFSTX_24	Checks if an invalid ack delay value is specified for a power switch
UPFSTX_25	Checks if an invalid net-edge pair is specified for a retention control
UPFSTX_26	Checks the presence of syntax errors
UPFSTX_27	Checks the presence of an hierarchical separator
UPFSTX_28	Checks the presence of an input file
UPFSTX_29	Checks the presence of an invalid command
UPFSTX_31	Checks the presence of extra arguments in a UPF command
UPFSTX_33	Checks the presence of arguments that cannot be specified together
UPFSTX_34	Checks if at least one required argument is specified in a command

Rule	Reports
UPFSTX_39	Checks the existence of the port state mentioned in the add_power_state command
UPFWRN_1	Checks if port/pin connection is simultaneously specified with mutually exclusive switch
UPFWRN_2	Checks if the presence of the specified vct in scope
UPFWRN_3	Checks if the function is already associated with net
UPFWRN_4	Checks if the argument is redefined
UPFWRN_5	Checks if an invalid resolve strategy is specified at the definition of net
UPFWRN_6	Checks if a command is invoked multiple times
UPFWRN_7	Checks if a list is specified for an argument
UPFWRN_13	Checks if the state name is used multiple times for a port
UPFWRN_14	Checks if PST state is redefined
UPFWRN_16	Checks if the command is supported
UPFWRN_17	Checks if an option in command is not supported
UPFWRN_19	Checks if the value of supply port specified in the add_power_state command is different than in the add_port_state command
UPFINFO_3	Checks if an unsupported UPF version is specified
UPFINFO_7	Checks if the supply is already set for a domain

checkUPF_existence

Reports non-existent design objects specified with UPF commands

When to Use

This is a setup rule and always runs by default.

Description

The *checkUPF_existence* rule reports when design objects, such as instances, nets, pins, and cells, specified with UPF commands do not exist in design.

The various scenarios, for which different violations are reported, are given along with the violation messages in the *Messages and Suggested fix* section.

Language

Verilog, VHDL

Parameter(s)

- *lp_allow_wildcard_in_UPF_2*: Default value is `1`. Set this parameter to `0` to report `Fatal` messages for wildcards used in the UPF commands, apart from the `find_object` command.
- *lp_skip_existence_check_for_resolve_parallel_net*: Default value is `no`. Set this parameter to `yes` to skip the existence check for supply nets created with the `-resolve parallel` UPF strategy.
- *lp_flag_missing_wildcard_in_set_retention*: Default value is `no`. Set this parameter to `yes` report a violation if any vector signal, when specified in the `elements` list of the `set_retention` command, is not specified with a wildcard.
- *lp_flag_missing_power_switch_lib*: Default value is `no`. Set this parameter to `yes` to report violations at the RTL/Netlist level for missing library cells specified in the `map_power_switch` command.
- *lp_hdlin_enable_upf_compatible_naming*: Default value is `no`. Set this parameter to `1` to support `struct` naming in System Verilog with `"."` to access the member variable of the `struct`.

- *lp_hdlin_sv_union_member_naming*: Default value is no. Set this parameter to 1 (along with the *lp_hdlin_enable_upf_compatible_naming* parameter also set to 1), to support union naming in System Verilog with "." to access the member variable of the union.
- *lp_skip_existence_check_for_virtual_supply_net_without_srsn*: Default value is no. Set this parameter to yes to skip existence check for virtual supplies that are not used in the set_related_supply_net command.
- *lp_set_design_stage*: Default value is rtl. Set this parameter to netlist or pg_netlist to specify the desired design stage.

Constraint(s)

None

Messages and Suggested fix

Message 1

The following message appears when the design object <*obj-name*> <*type*> (instances/nets/pins/cells) specified with a UPF command is not found in the design <*top-du-name*>:

[checkUPF_existence_1][Fatal]'<obj-name>'[<type>] not found on/within module '<top-du-name>'

[checkUPF_existence_2][Warning]'<obj-name>'[<type>] not found on/within module '<top-du-name>'

[checkUPF_existence_5][ERROR]'<obj-name>'[<type>] not found on/within module '<top-du-name>'

NOTE: *The same message is reported with different severity labels, based on the scenarios summarized in the table below.*

For debugging information, click *How to debug and fix*.

The following table shows various scenarios for which this message is

reported:

Scenario	Severity
The scope mentioned in the `scope` command of UPF file is not present in design.	**Fatal / Warning**
The hierarchy mentioned in the `elements` option of the `create_power_domain` UPF command is not present in design.	**Fatal / Warning**
The supply net provided in the `create_supply_net` command is not present in design.	**Fatal / Warning**
The supply net provided in the `input_supply_port` option of the `create_power_switch` UPF command is not present in design.	**Fatal / Warning**
The supply net provided in the `control_port` option of the `create_power_switch` UPF command is not present in design.	**Fatal / Warning**
The supply net provided in the `output_supply_port` option of the `create_power_switch` UPF command is not present in design.	**Fatal / Warning**
The hierarchical boundary/instance terminal provided in the `elements` option of the `set_isolation` is not present in design.	**Fatal / Warning**
The net provided in the `isolation_signal` option of `set_isoaltion/set_isolation_control` UPF command is not present in design.	**Fatal / Warning**
The sequential element or sequential element's output pin or net connected to output pin of the sequential element is not present in the design.	**Fatal / Warning**
The net name provided in the `save_signal` command of the `set_retention/set_retention_control` UPF command is not present in design.	**Fatal / Warning**
The net name provided in the `restore_signal` option of the `set_retention/ set_retention_control` UPF command is not present in design.	**Fatal / Warning**

Scenario	Severity
The instance provided in the `elements` option of the `map_retention_cell` command is not present in design.	**Fatal / Warning**
The hierarchical boundary/instance terminal provided in the `elements` option of the `map_level_shifter_cell` command is not present in design.	**Fatal / Warning**
The hierarchical boundary/instance terminal provided in the `elements` command option of the `set_level_shifter` command is not present in design.	**Fatal / Warning**
The pin provided in the `ports` option of the `connect_supply_net` command is not present in design.	**Warning**
The net provided in the `ack_port` option of the `create_power_switch` command is not present in design.	**Error**
The port/pin provided in the `object_list` option of the `set_related_supply_port` command is not present in design.	**Error**
The port/pin provided in the `ports` option of the `set_port_attributes` command is not present in design.	**Error**

NOTE: *In the table above, where the severity is mentioned as* **Fatal / Warning***, the decision on severity (fatal or warning) depends on if the ungrouping is used during the synthesis of the object/hierarchy. If yes then the severity is warning, otherwise it is fatal.*

Message 2

The following message appears when the cell <*cell-name*> specified with a UPF command is not found in the design and library:

[checkUPF_existence_6][Warning] `Cell '<cell-name>' not found both in design and library`

Message 3

The following message appears when the cell <*cell-name*> specified with a UPF command is not found in the library:

[checkUPF_existence_8][Fatal] `Cell '<cell-name>' not found in library`

[checkUPF_existence_9][Warning] `Cell '<cell-name>' not found in library`

NOTE: *The same message is reported with different severity labels, based on the scenarios summarized in the table below.*

The following table shows various scenarios for which this message is reported:

Scenario	Severity
The cell name provided in the `lib_cells` option of the `map_power_switch` command is not available in library, if the *lp_flag_missing_power_switch_lib* is set to `no` or the design is a PG Netlist design.	**Fatal**
The cell name provided in the `lib_cells` option of the `map_isolation_cell` command is not available in library.	**Warning**
The cell name provided in the `lib_cells` option of the `map_retention_cell` command is not available in library.	**Warning**
The cell name provided in the `lib_cells` option of the `map_level_shifter_cell` command is not available in library.	**Warning**
The cell name provided in the `models` option of the `set_design_attributes` command is not available in library.	**Warning**

Message 4

The following message appears when the pin <*pin-name*> specified with a UPF command is not found on/ or within the module <top-du-name>:

[checkUPF_existence_7][Error] `<pin-Type> pin <pin-name> not`

found on/within module '<top-du-name>'

The following table shows various scenarios for which this message is reported:

Scenario	Severity
The signal pin name provided in the pins option of the set_pin_related_supply command is not found in design.	Error
The power pin name provided in the related_power_pin option of the set_pin_related_supply command is not found in design.	Error
The ground pin name provided in the related_ground_pin option of the set_pin_related_supply command is not found in design.	Error
The signal name provided in the pins option of the set_pin_related_supply command is not found in design.	Error

Message 5

The following message appears when the design object in the UPF2.0 command (apart from find_object) is mentioned using a wildcard character:

[checkUPF_existence_11][Warning] Wildcard used in <obj-name> specified with <option-name> option of <command-name> command

[checkUPF_existence_13][Fatal] Wildcard used in <obj-name> specified with <option-name> option of <command-name> command

NOTE: *The same message is reported with different severity labels. If the wild card is used with connect_supply_net, map_isolation_cell, map_power_switch, map_retention_cell, and map_level_shifter commands then the severity will be warning, otherwise it will be fatal.*

Message 6

The following message appears when the design object in the UPF2.0 command without any option (apart from find_object) is mentioned using a wildcard character:

[checkUPF_existence_12][Warning] Wildcard used in <obj-name> specified with <command-name> command

[checkUPF_existence_14][Fatal] Wildcard used in <obj-name> specified with <command-name> command

NOTE: *The same message is reported with different severity labels. If the wild card is used with connect_supply_net, map_isolation_cell, map_power_switch, map_retention_cell, and map_level_shifter commands then the severity will be warning, otherwise it will be fatal.*

Message 7

The following massage appears if any vector signal provided in the `-elements` argument of the `set_retention` command is not specified with a wildcard:

[checkUPF_existence_16][ERROR] Vector signal '<signal-name>' is not specified with wildcards

Message 8

The following massage appears if element mentioned in the `-elements` argument of the `set_retention` command is a combo element:

[checkUPF_existence_17][ERROR] Element '<element-name>' used in set_retention command is not a sequential element it is a combo element

Message 9

The following massage appears if port direction in the `create_supply_port` command does not match with port direction in the PG design:

[checkUPF_existence_18][ERROR] Direction '<direction1>' of supply port '<port-name>' in upf does not match with direction '<direction2>' in design

The following table shows all the possible violation scenarios:

Port Direction in UPF (<*direction1*>)	**Port Direction in Design (<*direction2*>)**
in	output
inout	input or output
out	input or inout

Message 10

The following massage appears at the rtl stage if a net/port is created in UPF using the `create_logic_net/create_logic_port` commands and a net/port with the same name exists in design:

`[checkUPF_existence_19][ERROR]` `Design element <element-name> given in <command-name> command exist in rtl design`

Potential issues

The violation messages appear when:

- **Message 1**: The design object specified with a UPF command does not exist in the design. Alternatively, this violation message appears when ports/nets specified in *set_retention* command do not exist or do not have a sequential driver.

  The design object can be instances, nets, pins, or cells. In case of instances and nets, a Fatal message is reported.

  In case of pins and ports, Warning or Error messages are reported.
- **Message 2**: The cell name specified with a UPF command do not exist in design and library.
- **Message 3**: When an isolation cell is specified in the UPF in the *set_isolation* command, but does not exist in the Library.
- **Message 4**: This message appears in case of a power, ground, or signal pin specified in the UPF file using the *set_pin_related_supply* command, but does not exist in the design.
- **Message 5 and 6**: Wildcards are used outside the `find_object` command in the UPF.
- **Message 7**: The `lp_flag_missing_wildcard_in_set_retention` parameter is set to `yes`.
- **Message 8**: A combinational element is specified in the set_retention command
- **Message 9**: Wrong port direction specified in the `create_supply_port` command

- **Message 10**: A net/port is created in UPF using the `create_logic_net/create_logic_port` commands and a net/port with the same name exists in design.

Consequences of not Fixing

If this is not fixed, it means we are trying to apply a command on an element which is actually not existing, which is wrong. The command will not be applied and results pertaining to that command will be wrong. For **Message 8**, as a combinational element is specified in the `set_retention` command, so the `set_retention` command will not be applied on that element.

How to debug and fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation messages:

- **Message 1, 3, and 4**: Review the design file and ensure the design object reported in the violation message exists in the design. Check for spelling errors in the UPF and design file.
- **Message 2**: Review the design file and library. Ensure the cell name reported in the violation message exists in the design file and library. Check for spelling errors in the UPF file, design file, and library.
- **Message 5 and 6**: The name of the object, where a wildcard character is used, is displayed in the message. Remove the wildcard character from the object name to fix this issue. To use a wildcard character, you can use the `find_object` command of UPF2.0.
- **Message 7**: The vector signal that is causing the violation should be specified with a wildcard.
- **Message 8**: Review the design file and make sure that any combinational element is not specified in the `set_retention` command.
- **Message 9**: Review the design file and make sure that the port direction in UPF matches with the port direction in design.
- **Message 10**: Ensure that the net/ports created by using the `create_logic_net/create_logic_port` commands have unique names.

Example Code and/or Schematic

Example 1

Suppose `top.i1` exists in the design, but `top.i2` does not exist. For the following UPF command, the *checkUPF_existence* rule reports a violation because `top.i2` does not exist in the design.

```
create_power_domain –instances top.i1 top.i2
```

Example 2

The *checkUPF_existence* rule reports a violation because the isolation cell `ISO_NOT_IN_LIB` does not exist in the library.

```
set_isolation ISO3 -domain PD2   -no_isolation  -elements {
u2_inst/out1 }

set_isolation_control ISO3 -domain PD2 -isolation_signal iso
-isolation_sense low -location self

map_isolation_cell ISO3 -domain PD2 -lib_cells {
ISO_NOT_IN_LIB }
```

Example 3

The *checkUPF_existence* rule reports a violation because the Power pin `VDF` does not exist on the module `PAD01`.

```
set_pin_related_supply PAD01 -pins P1 -related_power_pin VDF
-related_ground_pin VC
```

Example 4

Consider the following command:

```
set_retention ret1 -domain PD1 -retention_power_net VDD1 -
elements {w}
```

For the above example the *checkUPF_existence* rule reports the following message because the vector signal `w` does not contain any wildcard:

```
Vector signal 'top.W' is not specified with wildcards
```

Example 5:

Consider the following UPF snippet:

```
create_supply_port VDD–direction out // Direction is output
```

Consider the following Verilog snippet:

```
module TOP(…,VDD,…)
input VDD;  // Direction is input
```

For the above example the *checkUPF_existence* rule reports the following message because the port direction in UPF is defined as `out`, but in the Verilog it is defined as `input`:

```
Direction '<out>' of supply port '<VDD>' in upf does not match
with direction '<input>' in design
```

Default and Severity label

Warning/Error/Fatal

Rule Group

`UPF Check`

Reports and Related files

None

UPF_lowpower01

Reports if top design unit specified with the set_design_top command does not match with current_design in SGDC

When to Use

This is a setup rule and always runs by default.

Description

The rule reports a violation message when the correct *set_design_top* command is not specified. The rule reports when the top design unit specified with `set_design_top` command is not same as the top design unit specified as `current_design` in the SGDC file.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

UPF Command

- *set_design_top* (Mandatory)

Messages and Suggested Fix

The following message appears when the top design unit specified with the `set_design_top` command in the UPF does not match with `current_design` in SGDC:

[FATAL] `The top design unit '<design-unit-name>' specified with 'set_design_top' command in UPF does not match with current_design`

Potential Issues

There is a mismatch in the top design unit name provided through the SGDC and the `set_design_top` command.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

To resolve this violation, provide consistent design unit names in SGDC and the *set_design_top* command.

Example Code and/or Schematic

Consider the following input files:

SGDC (top.sgdc):

```
current_design TOP
power_data -file const.upf -format upf
```

Project File:

```
read_file -type sgdc top.sgdc
set_design_top U1_inst
```

For the above example, the *UPF_lowpower01* rule reports the following violation message:

```
The top design unit 'U1_inst' specified with 'set_design_top'
command in UPF does not match with current_design
```

Default Severity Label

Fatal

Rule Group

UPF Check

Reports and Related Files

None

UPF_lowpower02

Checks relationship of power nets between set_domain_supply_net, connect_supply_net, and add_port_state commands

When to Use

This is a setup rule and always runs by default.

Description

The *UPF_lowpower02* rule reports a violation in the following cases:

- The *set_domain_supply_net* command is not specified in the UPF file.
- The *connect_supply_net* command is not specified for `primary_power_net/primary_ground_net` of domain.
- None of the ports specified with the *connect_supply_net* command of `primary_power_net` are defined with the *create_supply_port* command.
- The *add_port_state* command is not specified for supply port connected to `primary_power_net/primary_ground_net` of domain.
- The value of supply port specified with the *add_port_state* command is less than or equal to zero for `primary_power_net` and not equal to zero for `primary_ground_net`.
- Any of the ports specified with the *connect_supply_net* command for a non primary power net is not defined with the *create_supply_port* command.
- The *add_port_state* command is not specified for supply port connected to a non-primary power net of a domain.

Language

Verilog, VHDL

Parameter(s)

- *lp_output_pg_pin_as_supply_port*: Default values is `no`. Set this parameter to `yes` to assume the output PG Pin of the library cell as a supply port in the UPF.

- *lp_pg_direction_as_supply_port*: Default values is not set to any value. Set this parameter to `output` or `inout` or both, for the directions of PG pins that need to be considered as valid drivers of supply nets.

Constraint(s)

UPF

- *set_domain_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *add_power_state*

Messages and Suggested Fix

Message 1

The following message appears when the *set_domain_supply_net* command for the domain defined with the *create_power_domain* command is not specified for the power domain *<pd-name>*:

[FATAL] `The 'set_domain_supply_net' command should be specified for power domain '<pd-name>'`

Potential Issues

The violation message appears when no primary power net is specified for the domain. All domains must have a primary power net.

Consequences of Not Fixing

The domain will not have an operating voltage because it derives the operating voltage from the primary power net.

How to Debug and Fix

Specify the missing *set_domain_supply_net* command for the domain created by using the *create_power_domain* command.

Message 2

The following message appears when the *connect_supply_net* command is not specified for primary_power_net *<net-name>*:

[FATAL] `Missing 'connect_supply_net' command for supply net`

'<net-name>'

Potential Issues

The violation message appears when the connection with supply ports is not established in the power intent for the given supply net.

Consequences of Not Fixing

Different states of operation and respective voltage values are not inferred for the primary power net.

How to Debug and Fix

Specify the missing *connect_supply_net* command for the supply net.

Message 3

The following message appears when none of the ports specified for supply net *<net-name>* is defined with the *create_supply_port* command:

[FATAL] At least one of the ports specified for supply net '<net-name>' should be defined with 'create_supply_port' command

Potential Issues

The violation message appears when the supply nets are not connected with UPF supply ports.

Consequences of Not Fixing

Operating voltage values are not inferred for the given power net.

How to Debug and Fix

Specify the missing *create_supply_port* command for the ports mentioned in the *connect_supply_net* command or make sure that the supply net is connected to a supply port which has been defined through the *create_supply_port* command.

Message 4

The following message appears when the *add_port_state* command is not specified for the supply port *<port-name>* connected to the supply net *<net-name>*:

[FATAL] Missing 'add_port_state' command for supply port '<port-name>' connected to supply net '<net-name>'

Potential Issues

The violation message appears when the port states are not defined by *add_port_state* command for a supply port.

Consequences of Not Fixing

The operating voltage value of the supply ports would not be inferred.

How to Debug and Fix

Specify the missing *add_port_state* command for the supply port connected to the given supply net.

Message 5

The following message appears when the value *<port-name-value>* of the supply port *<port-name>* in case of `primary_power_net` specified with the *add_port_state* command is less than or equal to zero:

[FATAL] The value for supply port '<port-name>' should be positive value greater than zero, instead of '<port-name-value>'

Potential Issues

The violation message appears if zero or negative voltage values are specified for the supply port.

Consequences of Not Fixing

The voltage values cannot be inferred for the supply port.

How to Debug and Fix

Specify non-negative voltage values through the *add_port_state* command for the supply port. The `ON` state value should always be greater than zero for ports connected to `primary_power_nets`. The `OFF` state is recognized by `off` string in the command.

Message 6

The following message appears when the *connect_supply_net* command is not specified for the ground net *<gnd-net-name>*:

[WARNING] Missing 'connect_supply_net' command for ground net '<gnd-net-name>'

Potential Issues

The violation comes if `primary_ground_net` is not connected to any port through the *connect_supply_net* command.

Consequences of Not Fixing

If this is not fixed, it means that the ground net is connected to any port, that is, it is not used. Therefore, there is no point in defining that net. This might catch issues where you have forgotten to connect the ports to ground nets.

How to Debug and Fix

Specify the missing *connect_supply_net* command for the ground net.

Message 7

The following message appears when you have not used the *create_supply_port* command to specify at least one of the ports for ground net <gnd-net-name>:

[WARNING] At least one of the ports specified for ground net '<gnd-net-name>' should be defined with 'create_supply_port' command

Potential Issues

The violation message appears when the primary_ground_net is not connected with UPF supply ports.

Consequences of Not Fixing

The primary_ground_net should be connected to its supply port from where it infers its power intent. Therefore, at least one of the ports specified in *connect_supply_net* for primary_ground_net should be a supply port which is properly defined through the *create_supply_port* command. If this is not fixed, the power intent of ground_net is not inferred.

How to Debug and Fix

Make sure that primary_ground_net is connected to at least one supply port, which has been defined by the *create_supply_port* command.

Message 8

The following message appears when you have not specified the *add_port_state* command for the ground port <gnd-port>, which is connected to the ground net <gnd-net-name>:

[WARNING] Missing 'add_port_state' command for ground port '<gnd-port>' connected to ground net '<gnd-net-name>'

Potential Issues

The violation message appears when the port states are not defined by the *add_port_state* command for a ground supply port connected to the primary ground net.

Consequences of Not Fixing

The voltage value for a net is inferred from the corresponding supply port to which it is connected, which in turn is inferred from the states defined in the *add_port_state* command for that port. If this is not fixed, it means the power intent for the `primary_ground_net` cannot be inferred.

How to Debug and Fix

Make sure that the state of the ground supply ports connected to `primary_ground_net` is properly defined in the *add_port_state* command.

Message 9

The following message appears when the value *<gnd-port-value>* of the ground port *<gnd-port>* is not zero:

[WARNING] The value for ground port '<gnd-port>' should be zero, instead of '<gnd-port-value>'

Potential Issues

The violation message appears if you supply a non-zero value to the ground port.

Consequences of Not Fixing

Currently not available

How to Debug and Fix

Update the value of the ground port so that it is zero.

Message 10

The following message appears when you have not used the *add_port_state* command for the power port <port-name>, which is connected to the power net <net-name>:

[WARNING] Missing 'add_port_state' command for power port '<port-name>' connected to power net '<net-name>', assuming voltage as '0.0'

Potential Issues

The violation message appears when the port states are not defined by *add_port_state* command for a power supply port connected to the primary power net.

Consequences of Not Fixing

The voltage value for a net is inferred from the corresponding supply port to which it is connected, which in turn is inferred from the states defined in the *add_port_state* command for that port. If this is not fixed, the power intent for the primary_power_net cannot be inferred.

How to Debug and Fix

Make sure that the state of the power supply ports connected to primary_power_net are properly defined in *add_port_state* command.

Message 11

The following message appears when you have not defined at least one of the ports for the power net *<net-name>* with the *create_supply_port* command:

[WARNING] At least one of the ports specified for power net '<net-name>' should be defined with 'create_supply_port' command

Potential Issues

The violation message appears when the primary_power_net is not connected with UPF supply ports.

Consequences of Not Fixing

Primary_power_net should be connected to its supply port from where it infers its power intent. Therefore, at least one of the ports specified in *connect_supply_net* for primary_power_net should be a supply port which is properly defined through the *create_supply_port* command. If this is not fixed, the power intent of power_net is not inferred.

How to Debug and Fix

Make sure that primary_power_net is connected to at least one supply port, which has been defined by the *create_supply_port* command.

Message 12

The following message appears when you have not defined at least one of the ports for the power net *<net-name>* with the *create_supply_port*

command:

```
[WARNING] Unable to infer valid voltage value for supply net
'<net-name>' using related 'add_port_state | add_power_state'
command
```

Potential Issues

The violation message appears when the voltage value is not specified for the supply net.

Consequences of Not Fixing

The voltage value for a net is inferred from the corresponding supply port to which it is connected, which in turn is inferred from the states defined in the *add_port_state* command for that port or it is inferred from the *add_power_state* command. If this is not fixed, the power intent for the supply specified net cannot be inferred.

How to Debug and Fix

Make sure that the specified supply net is connected to at least one supply port or the state is mentioned in the *add_power_state* command using the `-supply_expr` option.

Message 13

The following message appears when the defined value of the supply port *<supply-port-name>* is out of the supported `range of -9999 to 9999`:

```
[FATAL] The value for supply <port | net> '<supply-port-name |
supply-net-name>'<given in state-name> should lie between
supported range -9999 to 9999, instead of provided value
'<value>'
```

Potential Issues

The violation message appears when the voltage value provided is out of the supported range.

Consequences of Not Fixing

It leads to undefined rule checking behavior.

How to Debug and Fix

Make sure that the specified `add_port_state` or `add_power_state` value is in the range of `-9999 to 9999`.

Message 14

The following message appears when the value corresponding to the *<net-state-name>* for supply net *<net-name>* is not mentioned in the UPF either through the `add_power_state` command for the same net or through the `add_port_state` command for a connected port:

[FATAL] `Unable to find voltage value corresponding to state ‘<net-state-name>’ for supply net ‘<net-name>' in add_port_state command ‘<state-name>' for supply set ‘<supply-set-name>`

Potential Issues

A valid voltage value cannot be inferred corresponding to the net state for the given net.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

Make sure that the net state is provided either through the `add_port_state` command or `add_power_state` command for any of the connected ports.

Message 15

The following message appears when the net corresponding to *<net-name>* is not found in the scope of the supply set for which this command is specified:

[FATAL] `Supply net '<net-name>' mentioned in add_power_state command ‘<state-name>' for supply set ‘<supply-set-name>' not found`

Potential Issues

A net corresponding to the given net name is not found.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

Make sure that the net is created through a `create_supply_net`

command.

Message 16

The following message appears when the expression specified in the `supply_expr` option of the `add_power_state` command for the supply set *<supply-set-name>* cannot be evaluated:

[FATAL] Unable to evaluate expression '<expression>' mentioned in the add_power_state command ‘<state-name>' for Supply Set '<supply-set-name>'

Potential Issues

State of nets in the supply set cannot be determined.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

Make sure that the supply expression is written properly.

Example Code and/or Schematic

Example 1

Consider the following example code:

```
set_design_top mid

# power domain definitions
create_power_domain top
create_power_domain shutOFF -elements t1/inst4

# supply nets
create_supply_port VDD
add_port_state VDD -state {default 1.1 1.2 1.3}
create_supply_net  VDD -domain top

create_supply_port VDD_LP
add_port_state VDD_LP -state {default 1.1 1.2 1.3} -state
```

```
{off_state off}
create_supply_net  VDD_LP -domain shutOFF

connect_supply_net VDD -ports VDD
connect_supply_net VDD_LP -ports VDD_LP


create_supply_port VSS
create_supply_net  VSS -domain top
connect_supply_net VSS -ports VSS

add_port_state VSS -state {default 0.0}


#set_domain_supply_net top -primary_power_net VDD

primary_ground_net VSS

##set_domain_supply_net shutOFF -primary_power_net VDD_LP
primary_ground_net VSS
```

The following *UPF_lowpower02* violations are reported:

```
The 'set_domain_supply_net' command should be specified for
power domain 'top'

The 'set_domain_supply_net' command should be specified for
power domain 'shutOFF'
```

Example 2

Consider the following example code:

```
set_design_top mid
create_power_domain top
create_power_domain shutOFF -elements t1/inst4

create_supply_port VDD
add_port_state VDD -state {default 1.1 1.2 1.3}
create_supply_net  VDD -domain top

create_supply_port VDD_LP
add_port_state VDD_LP -state {default 1.1 1.2 1.3} -state
```

```
{off_state off}
create_supply_net  VDD_LP -domain shutOFF

create_supply_port VSS
create_supply_net  VSS -domain top
connect_supply_net VSS -ports VSS

add_port_state VSS -state {default 0.0}
set_domain_supply_net top -primary_power_net VDD
primary_ground_net VSS
set_domain_supply_net shutOFF -primary_power_net VDD_LP
primary_ground_net VSS
```

The following *UPF_lowpower02* violations are reported:

```
Missing 'connect_supply_net' command for supply net 'VDD'
Missing 'connect_supply_net' command for supply net 'VDD_LP'
```

Example 3

Consider the following example code:

```
create_supply_net VSS_ALIVE -domain VA0
create_supply_port VSS_ALIVE -domain VA0

create_supply_net VDD_ALIVE -domain VA4 -reuse
create_supply_net VSS_ALIVE -domain VA4 -reuse

#add_port_state VDD_ALIVE -state VA4_ON 0.99} #Added
add_port_state VDD_ALIVE -state {VA4_OFF off}

connect_supply_net VVDD1 -ports junk
connect_supply_net VVDD2 -ports VVDD2

add_port_state VVDD2 -state {VA4_ON 0.99} -state
{VA4_OFF off}
##connect_supply_net VVDD3 -ports VVDD3
add_port_state VDD -state {VA_ON -0.99} -state
{VA4_OFF off}
```

The following violation is reported:

```
At least one of the ports specified for supply net 'VVDD1'
should be defined with 'create_supply_port' command
```

Example 4

Consider the following example code:

```
set_design_top mid

# power domain definitions
create_power_domain top
create_power_domain shutOFF -elements t1/inst4

# supply nets
create_supply_port VDD
add_port_state VDD -state {default 1.1 1.2 1.3}
create_supply_net  VDD -domain top

create_supply_port VDD_LP
add_port_state VDD_LP -state {default 1.1 1.2 1.3}
-state {off_state off}
create_supply_net  VDD_LP -domain shutOFF

connect_supply_net VDD -ports VDD
connect_supply_net VDD_LP -ports VDD_LP

create_supply_port VSS
create_supply_net  VSS -domain top
connect_supply_net VSS -ports VSS

set_domain_supply_net top -primary_ground_net VSS
set_domain_supply_net shutOFF -primary_power_net VDD_LP
```

The following violation is reported because the *add_port_state* command is not specified for the supply port VSS connected to the supply net VSS:

```
Missing 'add_port_state' command for power port 'VSS' connected
to power net 'VSS', assuming voltage as '0.0'
```

Example 5

Consider the following example code:

```
create_power_domain shutOFF -elements t1/inst4

create_supply_port VDD_LP
create_supply_net VDD_LP -domain shutOFF add_port_state
VDD_LP -state {default 0.0} -state {off_state off}
connect_supply_net VDD_LP -ports VDD_LP

create_supply_port VSS
create_supply_net VSS -domain top
connect_supply_net VSS -ports VSS
add_port_state VSS -state {default 0.0}

set_domain_supply_net shutOFF -primary_power_net VDD_LP -
primary_ground_net VSS
```

The following violation is reported for supply port VDD_LP (connected to the primary power net VDD_LP of domain shutOFF), because a value less than or equal to 0 has been provided as a port state:

```
The value for supply port 'VDD_LP' should be positive value
greater than zero, instead of '0.0'
```

Example 6

Consider the following UPF snippet:

```
create_power_domain VA0 -include_scope
create_power_domain VA1 -elements uArm9/uCOREDP
create_power_domain VA2 -elements uArm9/uCORECTL
create_power_domain VA3 -elements uArm9/uDbg
create_power_domain VA4 -elements uSysctl

create_supply_net VDD -domain VA0
create_supply_net VSS -domain VA0
create_supply_net VVDD0 -domain VA0
create_supply_port VDD -domain VA0
create_supply_port VSS -domain VA0
create_supply_port VVDD0 -domain VA0

connect_supply_net VDD -ports VDD
```

```
set_domain_supply_net VA3 -primary_power_net VVDD3
primary_ground_net VSS

create_supply_net VDD_ALIVE -domain VA0
create_supply_port VDD_ALIVE -domain VA0
create_supply_net VSS_ALIVE -domain VA0
create_supply_port VSS_ALIVE -domain VA0
```

The following violation messages are reported:

```
Missing 'connect_supply_net' command for ground net 'VSS'

Missing 'add_port_state' command for power port 'VDD' connected
to power net 'VDD', assuming voltage as '0.0'

Missing 'connect_supply_net' command for ground net 'VSS_ALIVE'
```

Example 7

Consider the following UPF snippet:

```
set_design_top mid

 create_power_domain top
 create_power_domain UA -elements inst1
 create_power_domain UB -elements inst2

 create_supply_port VDDA
 create_supply_port VDDB
 create_supply_port VSSA
 create_supply_port VTOP
 create_supply_port VSSB
 create_supply_net  VDDA -domain top
create_supply_net  VDDB -domain top
create_supply_net  VDDA -domain UA -reuse
create_supply_net  VDDB -domain UB -reuse
create_supply_net  VSSA -domain top
create_supply_net  VSSB -domain top
create_supply_net  VSSA -domain UA -reuse
create_supply_net  VSSB -domain UB -reuse
create_supply_net  VTOP -domain top
```

```
connect_supply_net VDDA -ports { pad1/VA }
connect_supply_net VDDB -ports { pad1/VB }
connect_supply_net VDDA -ports VDDA
connect_supply_net VDDB -ports VDDB
connect_supply_net VTOP -ports VTOP
connect_supply_net VSSA -ports { pad1/VCA }
connect_supply_net VSSB -ports { pad1/VCB }

set_domain_supply_net UA -primary_power_net VDDA
primary_ground_net VSSA
set_domain_supply_net UB -primary_power_net VDDB
primary_ground_net VSSB
set_domain_supply_net top -primary_power_net VTOP
primary_ground_net VSSA
```

The following messages are reported:

```
At least one of the ports specified for ground net 'VSSA'
should be defined with 'create_supply_port' command

At least one of the ports specified for ground net 'VSSB'
should be defined with 'create_supply_port' command
```

Example 8

Consider the following UPF snippet:

```
set_design_top test

create_power_domain pd_safe -include_scope
create_power_domain pd_core -elements u1

create_supply_net vddtest02 -domain pd_safe
create_supply_net vddsafe -domain pd_safe
create_supply_net vddcore -domain pd_safe
create_supply_net gnd -domain pd_safe

set_domain_supply_net pd_core -primary_power_net vddcore
primary_ground_net gnd
set_domain_supply_net  pd_safe -primary_power_net vddsafe
-primary_ground_net gnd
```

```
create_supply_port vddsafe -domain pd_safe
add_port_state vddsafe -state {vsafe_on 1.05}
create_supply_port vddcore -domain pd_safe
add_port_state vddcore -state {vcore_on 1.20}

create_supply_port vss -domain pd_safe
#add_port_state vss -state {vcore_on 1.20}
connect_supply_net gnd -ports vss

connect_supply_net vddsafe -ports vddsafe
connect_supply_net vddcore -ports vddcore
create_supply_port vddtest02 -domain pd_safe
```

The following messages are reported:

```
Missing 'add_port_state' command for ground port 'vss'
connected to ground net 'gnd'
```

```
Missing 'add_port_state' command for power port 'vddtest02'
connected to power net 'vddtest02', assuming voltage as '0.0'
```

Example 9

Consider the following UPF snippet:

```
set_design_top test

create_power_domain pd_safe -include_scope
create_power_domain pd_core -elements u1

create_supply_net vddtest02 -domain pd_safe
create_supply_net vddsafe -domain pd_safe
create_supply_net vddcore -domain pd_safe
create_supply_net gnd -domain pd_safe

set_domain_supply_net pd_core -primary_power_net vddcore
primary_ground_net gnd
set_domain_supply_net  pd_safe -primary_power_net vddsafe
-primary_ground_net gnd

create_supply_port vddsafe -domain pd_safe
add_port_state vddsafe -state {vsafe_on 1.05}
```

```
create_supply_port vddcore -domain pd_safe
add_port_state vddcore -state {vcore_on 1.20}
create_supply_port gnd -domain pd_safe
add_port_state gnd -state {off_state off} -state
{on_state   1.2}
connect_supply_net vddsafe -ports vddsafe
connect_supply_net vddcore -ports vddcore
connect_supply_net gnd -ports gnd
#create_supply_port vddtest02 -domain pd_safe
```

The following message is reported because the value of the ground port `gnd` is not zero:

```
The value for ground port 'gnd' should be zero, instead of
'1.2'
```

Example 10

Consider the following UPF snippet:

```
set_design_top test

create_power_domain pd_safe -include_scope
create_power_domain pd_core -elements u1

create_supply_net vddtest02 -domain pd_safe
create_supply_net vddsafe -domain pd_safe
create_supply_net vddcore -domain pd_safe
create_supply_net gnd -domain pd_safe

set_domain_supply_net pd_core -primary_power_net vddcore
-primary_ground_net gnd
set_domain_supply_net  pd_safe -primary_power_net vddsafe
-primary_ground_net gnd

create_supply_port vddsafe -domain pd_safe
add_port_state vddsafe -state {vsafe_on 1.05}
create_supply_port vddcore -domain pd_safe
add_port_state vddcore -state {vcore_on 1.20}
```

```
create_supply_port vss -domain pd_safe
add_port_state vss -state {vcore_on 1.20}
connect_supply_net gnd -ports vss

connect_supply_net vddsafe -ports vddsafe
connect_supply_net vddcore -ports vddcore
create_supply_port vddtest02 -domain pd_safe
```

The following message is reported because the *add_port_state* command has not been used for the power port `vddtest02`, which is connected to the power net `vddtest02`:

```
Missing 'add_port_state' command for power port 'vddtest02'
connected to power net 'vddtest02', assuming voltage as '0.0'
```

Example 11

Consider the following UPF snippet:

```
connect_supply_net VVDD2 -ports VVDD2
add_port_state VVDD2 -state {VA4_ON 0.99} -state {VA4_OFF
off}

##connect_supply_net VVDD3 -ports VVDD3

add_port_state VDD -state {VA_ON -0.99} -state {VA4_OFF
off}
set_domain_supply_net VA4 -primary_power_net VDD_ALIVE
primary_ground_net VSS_ALIVE
set_retention ret_arm_pmk -domain VA1 -
retention_power_net
VDD -retention_ground_net VSS
set_retention_control ret_arm_pmk -domain VA1 -save_signal
{RETEN_1 low } -restore_signal {RETEN_1 high}
map_retention_cell ret_arm_pmk -domain VA1 -lib_cell_type
DRFF

set_retention ret_sec -domain VA2 -retention_power_net VDD
-retention_ground_net VSS
set_retention_control ret_sec -domain VA2 -save_signal
{RETEN_1 low } -restore_signal {RETEN_1 high}
```

```
map_retention_cell ret_sec -domain VA2 -lib_cell_type DRFF

set_retention ret_sec_test -domain VA3
retention_power_net VDD -retention_ground_net VSS
set_retention_control ret_sec_test -domain VA3
save_signal {RETEN_3 low } -restore_signal {RETEN_3 high}
map_retention_cell ret_sec_test -domain VA3 -lib_cell_type
DRFF

create_power_switch psw_head_0 -domain VA0
output_supply_port {psw_head_o_out VVDD0} -
input_supply_port
{VDD VDD_ALIVE} -control_port { SC_0 SC_0 } -on_state {
VA0_ON VDD !SC_0 }
map_power_switch psw_head_0 -domain VA0 -lib_cells
HEADBUF16_X1M
```

The following message is reported because at least one of the ports for the power net VDDD_ALIVE with the *create_supply_port* command has not been defined:

```
At least one of the ports specified for power net 'VDD_ALIVE'
should be defined with 'create_supply_port' command
```

Example 12

Consider the following example code:

```
create_supply_port VDD1
add_port_state VDD1 -state  { on1 12111.11111 }  -state  {
off1 off }
```

The following violation is reported for supply port VDD1 because it's given value is out of the supported range:

```
The value for supply port 'VDD1' given in state 'on1' should
lie between supported range -9999 to 9999, instead of provided
value '12111.1'
```

Example 13

Consider the following example code:

```
add_power_state supply_set1 -state on1 { -supply_expr {power
```

```
== {FULL_ON, 12111.11111}} }
```

The following violation is reported for supply net that is associated as `power` handle of supply set `supply_set1`:

```
The value for supply net 'supply_set1.power' given in state
'FULL_ON' should lie between supported range -9999 to 9999,
instead of provided value '12111.1'
```

Example 14

Consider the following UPF 2.0 snippet:

```
create_supply_set ss
create_supply_set ss1

create_power_domain TOP -include_scope -supply {primary ss}
create_power_domain PD1 -elements {inst1} -supply {primary
ss1}

create_supply_port VDD
create_supply_port VDD1
create_supply_port VSS

create_supply_net VDD
create_supply_net VDD1
create_supply_net VSS

create_supply_set ss -function {power VDD} -function {ground
VSS} -update
create_supply_set ss1 -function {power VDD1} -function
{ground VSS} -update

connect_supply_net VDD -ports {VDD}
connect_supply_net VDD1 -ports {VDD1}
connect_supply_net VSS -ports {VSS}

add_port_state VDD -state {on1 1.2} -state {off1 off}
add_port_state VDD1 -state {on1 1.2} -state {off1 off}
add_port_state VSS -state {on1 0.0}
add_power_state ss -state on1 { -supply_expr {power == on2
```

```
&& ground == on1 } }
```

The following message is reported because state `on2` is not defined for the ports connected to net `power` of supply set `ss`:

```
Unable to find voltage value corresponding to state 'on2' for
supply net 'power' in add_port_state command 'on1' for supply
set 'ss'
```

Example 15

Consider the following UPF 2.0 snippet:

```
create_supply_set ss
create_supply_set ss1

create_power_domain TOP -include_scope -supply {primary ss}
create_power_domain PD1 -elements {inst1} -supply {primary
ss1}

create_supply_port VDD
create_supply_port VDD1
create_supply_port VSS

create_supply_net VDD
create_supply_net VDD1
create_supply_net VSS

create_supply_set ss -function {power VDD} -function {ground
VSS} -update
create_supply_set ss1 -function {power VDD1} -function
{ground VSS} -update

connect_supply_net VDD -ports {VDD}
connect_supply_net VDD1 -ports {VDD1}
connect_supply_net VSS -ports {VSS}

add_port_state VDD -state {on1 1.2} -state {off1 off}
add_port_state VDD1 -state {on1 1.2} -state {off1 off}
add_port_state VSS -state {on1 0.0}
add_power_state ss -state on11 {-supply_expr { VDDx ==
```

```
`{FULL_ON,1.2} } }
```

The following message is reported because VDDx is not created in the UPF:

```
Supply net 'VDDx' mentioned in add_power_state command 'on11'
for supply set 'ss' not found
```

Example 16

Consider the following UPF 2.0 snippet:

```
create_supply_set ss
create_supply_set ss1

create_power_domain TOP -include_scope -supply {primary ss}
create_power_domain PD1 -elements {inst1} -supply {primary
ss1}

create_supply_port VDD
create_supply_port VDD1
create_supply_port VSS

create_supply_net VDD
create_supply_net VDD1
create_supply_net VSS

create_supply_set ss -function {power VDD} -function {ground
VSS} -update
create_supply_set ss1 -function {power VDD1} -function
{ground VSS} -update

connect_supply_net VDD -ports {VDD}
connect_supply_net VDD1 -ports {VDD1}
connect_supply_net VSS -ports {VSS}

add_port_state VDD -state {on1 1.2} -state {off1 off}
add_port_state VDD1 -state {on1 1.2} -state {off1 off}
add_port_state VSS -state {on1 0.0}
add_power_state ss1 -state on3 { -supply_expr { “123x” } }
```

The following message is reported because supply expression 123x cannot

be resolved as it is not related to supply set `ss1`:

```
Unable to evaluate expression '123x' mentioned in the
add_power_state command 'on3' for Supply Set 'ss1'
```

Default Severity Label

Fatal, Warning

Rule Group

UPF Check

Reports and Related Files

None

UPF_lowpower03

Reports unspecified default power domain for the top design unit

When to Use

This is a setup rule and always runs by default.

Description

The *UPF_lowpower03* rule reports a violation message when:

- The default power domain is not specified.
- The unique power domain for top design unit is not specified.
- Power domain not specified for top design unit

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

UPF Command

- *create_power_domain* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the top design unit is not specified:

```
[FATAL] The default power domain for top design unit should be specified with '-include_scope' option
```

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the top design unit is included in the extent of multiple power domains:

```
[FATAL] The top design unit should not be included in the extent of multiple power domains
```

Message 3

The following message appears when the power domain is not specified for the top-level design unit:

[FATAL] `Power Domain specification has not been specified for top design unit`

Potential Issues

Every design must have a top-level power domain, created in the top-level scope with the `–include_scope` option to include everything at the top-level.

You can assign any cell to one power domain only. Therefore, to completely specify the power intent of a design, each cell in the design must be associated with one power domain, and not multiple power domains.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation messages:

- **Message 1**: Specify the `–include_scope` option to the create_power_domain command so that the entire hierarchical level of the domain is included at the top-level.
- **Message 2**: Ensure the top-level design unit is not associated with multiple power domains.
- **Message 3**: Ensure the power domain is specified for the top-level design unit.

Example Code and/or Schematic

Example 1

For the following UPF snippet, the *UPF_lowpower03* rule reports a violation because the default power domain is not specified for the top-level design unit, `TOP`.

```
create_power_domain TOP

create_power_domain sub1 –elements {u1}
```

To resolve the violation message, specify the `-include_scope` argument as shown below:

```
create_power_domain TOP -include_scope

create_power_domain sub1 -elements {u1}
```

Example 2

Consider the following code snippet:

```
    set_design_top mid
    # power domain definitions
    create_power_domain top
    create_power_domain shutOFF -elements t1/inst4
    -include_scope

    # supply nets
    create_supply_port VDD
    add_port_state VDD -state {default 1.2}
    create_supply_net  VDD -domain top

  create_supply_port VDD_LP
  add_port_state VDD_LP -state {default 1.2} -state
  {off_state off}
  create_supply_net  VDD_LP -domain shutOFF

  connect_supply_net VDD -ports VDD
  connect_supply_net VDD_LP -ports VDD_LP

  create_supply_port VSS
  create_supply_net  VSS -domain top
  connect_supply_net VSS -ports VSS
  add_port_state VSS -state {default 0.0}

  set_domain_supply_net top -primary_power_net VDD
  primary_ground_net VSS
  set_domain_supply_net shutOFF -primary_power_net VDD_LP
  primary_ground_net VSS
```

The following message is reported because the top design unit is included

in the extent of multiple power domains:

```
The top design unit should not be included in the extent of
multiple power domains
```

Example 3

For the following UPF snippet, the *UPF_lowpower03* rule reports a violation because power domain is not specified for the top-level design unit, `top`.

```
upf_version 2.0
set_design_top top
```

Default Severity Label

Fatal

Rule Group

`UPF Check`

Reports and Related Files

None

UPF_lowpower04

Reports multiple pst state definitions

When to Use

This is a setup rule and always runs by default.

Description

The *UPF_lowpower04* rule reports when supply state is defined multiple times for a Power State Table (PST) using the *add_pst_state* command.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

UPF

- *add_pst_state (Mandatory)*

Messages and Suggested Fix

The following message appears when for PST *<pst-name>*, the state *<state-name>* is defined multiple times:

[WARNING] For pst '<pst-name>', states '<state-names>' are duplicate

Potential Issues

The listed states are duplicate, which should avoided.

Consequences of Not Fixing

Specifying duplicate states in pst might unnecessarily increase the run time of a verification tool with no added advantage.

How to Debug and Fix

The UPF file that contains the duplicate states is highlighted in the Atrenta Console GUI.

To resolve this violation, review the state entries specified by the `state` argument of the *add_pst_state* command and ensure each entry is unique.

Example Code and/or Schematic

For the following UPF snippet, the *UPF_lowpower04 rule reports a violation because the state of pst0 and pst1 is duplicated.*

```
create_pst PM_pst –supplies {VDD u0/VDD_SUB0_SW VDD_SUB1
VDD_SUB2 }

add_pst_state pst0 –pst PM_pst –state {VDD_N SW_on SUB1_H
SUB2_H}

add_pst_state pst1 –pst PM_pst –state {VDD_N SW_on SUB1_H
SUB2_H}
```

To resolve this violation, change the state to a unique entry, as shown in the following UPF snippet:

```
create_pst PM_pst –supplies {VDD u0/VDD_SUB0_SW VDD_SUB1
VDD_SUB2 }

add_pst_state pst0 –pst PM_pst –state {VDD_N SW_on SUB1_H
SUB2_H}

add_pst_state pst1 –pst PM_pst –state {VDD_N SW_off SUB1_L
SUB2_L}
```

Default Severity Label

Warning

Rule Group

`UPF Check`

Reports and Related Files

None

UPF_lowpower05

Checks set_isolation_control and map_isolation_cell commands

When to Use

This is a setup rule and always runs by default.

Description

The *UPF_lowpower05* rule reports in the following cases:

- If the `no_isolation` argument is not specified with *set_isolation* command and the:
  - *set_isolation_control* command is not given. A control signal has to be provided for each isolation strategy written using the *set_isolation* command. This is valid for UPF 1.0 only.
  - `isolation_signal` is not specified in the *set_isolation* command. In addition, the *set_isolation_control* command for the related strategy is not specified. This is valid for UPF 2.0 only.
  - *map_isolation_cell* command is not given. A particular cell should be specified for each isolation strategy written using the *set_isolation* command or else any isolation cell present in library would be used.
- If the *is_isolation_cell* library attribute is not set for any isolation cell used with the `isolation_cell` or *map_isolation_cell* command.
- If the `clamp_value` has not been specified in the *set_isolation* command
- If the `isolation_enable_condition` attribute is defined for at least one signal pin for a cell in the library.

Language

Verilog, VHDL

Parameter(s)

- *lp_check_map_iso_cmd*:Default values is `1`. Set this parameter to `0` to ignore the missing `map_isolation_cell` command corresponding to a `set_isolation` command in the UPF.

- *lp_match_iso_cell_type_with_clamp_value*:Default values is `no`. Set this parameter to `yes` to report violation if:
  - An AND gate is not specified in the `map_isolation_cell` command and `clamp_value` is given as `0` or no `clamp_value` is specified in the corresponding `set_isolation` command.
  - An OR cell is not specified in the `map_isolation_cell` command and `clamp_value` is given as `1` in the corresponding `set_isolation` command.

Constraint(s)

UPF

- *map_isolation_cell* (Mandatory)
- *set_isolation_control* (Mandatory)
- *set_isolation* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the `isolation_sense` field is not specified for the *set_isolation_control* command:

[UPF_lowpower05_1][WARNING] `Missing isolation_sense in strategy '<iso-strategy>', using default value for isolation_signal '<iso-sig>'`

For information on debugging, click *How to Debug and Fix*.

Message 2

The following message appears when the *set_isolation_control* command is not provided for the rule *<rule-name>* specified with the *set_isolation* command:

[UPF_lowpower05_2][FATAL] `Command 'set_isolation_control' not given for isolation rule '<rule-name>', defined by set_isolation`

For information on debugging, click *How to Debug and Fix*.

Message 3

The following message appears when the *map_isolation_cell* command is not provided for the rule <rule-name> specified with the set_isolation command:

[UPF_lowpower05_3][WARNING] Command 'map_isolation_cell' should be given for isolation rule '<rule-name>' defined by set_isolation, for using a specific isolation cell

For information on debugging, click *How to Debug and Fix*.

Message 4

The following warning message appears when the cell *<cell-name>* specified through the *map_isolation_cell* command does not have an *is_isolation_cell* attribute set in the library:

[UPF_lowpower05_4][WARNING] Cell '<cell-name>' specified as isolation cell, does not have 'is_isolation_cell' attribute in the library cell description

For information on debugging, click *How to Debug and Fix*.

Message 5

The following message appears when the clamp_value is not specified for the isolation strategy *<iso-strategy>*:

[UPF_lowpower05_5][WARNING] Missing clamp_value in strategy '<iso-strategy>', using default clamp value '0' for isolation signal(s) '<iso-sig>'

Message 6

The following info message appears when the isolation_enable_condition attribute is defined in the library for at least one signal pin of a cell instantiated in the design:

[UPF_lowpower05_6][INFO] Unable to determine if a signal connected to pin used in 'isolation_enable_condition' attribute is indeed an isolation control signal, as it is not defined in UPF. Hence typical rules related to isolation control signals will not trigger for such pins

Message 7

The following info message appears when the

isolation_enable_condition attribute is defined in the library for at least one signal pin of a cell instantiated in the design:

[UPF_lowpower05_7][WARNING] Cell <cell_name> specified in map_isolation_cell command <commandp-name> is not an <AND/OR> cell whereas clamp value is specified as <0/1/none>

This message is reported only when the *lp_match_iso_cell_type_with_clamp_value* parameter is set.

Potential Issues

The potential issues related to Message 1 depend on the UPF version.

- **For UPF 1.0**: Message 1 appears when the isolation_sense field is not specified for the *set_isolation_control* command.
- **For UPF 2.0**: Message 1 appears when the isolation_sense field is not specified for the *set_isolation_control* command or when the isolation_sense field is not specified for the related isolation_signal list in the *set_isolation* command.

For Message 7, the violation points to a design error.

The other violation messages explicitly state the potential issues.

Consequences of Not Fixing

An isolation cell is inserted for an isolation strategy written using *set_isolation* command. An enable signal of a isolation cell should be driven by a correct isolation signal to implement proper isolation logic. Therefore, it is necessary to specify isolation signal for each isolation strategy. Also, not specifying *map_isolation_cell* or specifying non-isolation cells in *map_isolation_cell* command might lead incorrect cell being inserted leading to incorrect isolation logic in the design.

For Message 2, the SpyGlass Power Verify solution cannot proceed with its analysis and will exit. For Message 7, if violation is not be fixed, the isolation functionality may not work properly.

For all other messages, the SpyGlass Power Verify solution analysis will continue.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation messages:

- **Message 1**: Specify the `isolation_sense` argument for the *set_isolation_control* command for the rule name reported in the violation message.
- **Message 2**: Specify the *set_isolation_control* command for the rule stated in the violation message. You specified the rule name in the `set_isolation` command.
- **Message 3**: Specify the *map_isolation_cell* command for the rule reported in the violation message. You specified the rule name in the `set_isolation` command.
- **Message 4**: Set the *is_isolation_cell* attribute set in the library for the cell name reported in the violation message. You specified the cell name in the *map_isolation_cell* command.
- **Message 5**: Specify the `clamp_value` in the *set_isolation* command in the UPF file.
- **Message 7:** The `map_isolation_cell` command is given in the violation message. Replace the incorrect cell in the `map_isolation_cell` command with the correct cell.

Example Code and/or Schematic

Example 1

Consider the following code snippet:

```
set_design_top mid
# power domain definitions
create_power_domain top
create_power_domain shutOFF -elements t1/inst4

# supply nets
create_supply_port VDD
add_port_state VDD -state {default 1.2 }
create_supply_net  VDD -domain top

create_supply_port VDD_LP
add_port_state VDD_LP -state {default 1.2 } -state
{off_state off}
create_supply_net  VDD_LP -domain shutOFF
```

```
connect_supply_net VDD -ports VDD
connect_supply_net VDD_LP -ports VDD_LP


create_supply_port VSS
add_port_state VSS -state {gnd_state 0.0}
create_supply_net  VSS -domain top
connect_supply_net VSS -ports VSS

set_domain_supply_net top -primary_power_net VDD
primary_ground_net VSS
set_domain_supply_net shutOFF -primary_power_net VDD_LP
primary_ground_net VSS

# isolation rules
set_isolation iso_r1 -domain shutOFF -clamp_value 0
set_isolation_control iso_r1 -domain shutOFF
isolation_signal isosig -location fanout
```

The following message is reported because the `isolation_sense` field `iso_r1` is not specified:

```
Missing isolation_sense in strategy 'iso_r1', using default
value for isolation_signal 'isosig'
```

Example 2

For the following UPF file snippet, the *UPF_lowpower05* rule reports a violation message because the *set_isolation_control* command has not been specified.

```
set_isolation iso_r1 -domain shutOFF -clamp_value 0
```

Example 3

For the following UPF file snippet, the *UPF_lowpower05* rule reports a violation message because the *map_isolation_cell* command has not been specified.

```
set_isolation iso_r1 -domain shutOFF -clamp_value 0

set_isolation_control iso_r1 -domain shutOFF -
```

```
isolation_signal isosig -location parent
```

Example 4

Consider the following code snippet:

```
set_design_top top
# power domain definitions
create_power_domain top  -include_scope
create_power_domain PD1  -elements inst1
create_power_domain PD2  -elements inst1/buf_cell

create_supply_net  VDD -domain top
create_supply_net  VSS -domain top

create_supply_port  VDD
add_port_state VDD    -state {active_state 1.2}
create_supply_port  VSS
add_port_state VSS -state {gnd_state 0.0 }

connect_supply_net VDD -ports VDD
connect_supply_net VSS -ports VSS

create_supply_net  VDD_SW -domain PD1
create_supply_port  VDD_SW
add_port_state VDD_SW -state {active_state 1.2} -state
{off_state off}
connect_supply_net VDD_SW -ports VDD_SW

set_domain_supply_net top -primary_power_net VDD
primary_ground_net VSS
set_domain_supply_net PD1 -primary_power_net VDD_SW
primary_ground_net VSS
set_domain_supply_net PD2 -primary_power_net VDD_SW
primary_ground_net VSS

set_isolation iso_r1 -domain PD1  -applies_to outputs
set_isolation_control iso_r1 -domain PD1  -isolation_signal
iso
map_isolation_cell iso_r1 -domain PD1 -lib_cells AN2
```

```
set_isolation iso_r2 -domain PD2 -applies_to inputs
clamp_value latch
```

The following message is reported because the clamp value is not specified for the isolation strategy `iso_r1`:

```
Missing clamp_value in strategy 'iso_r1', using default clamp
value '0' for isolation signal(s) 'iso '
```

Example 5

Consider the following UPF Command:

```
set_isolation iso_r1 -domain PD1  -applies_to outputs -
clamp_value 0
set_isolation_control iso_r1 -domain PD1  -isolation_signal /
iso
map_isolation_cell iso_r1 -domain PD1 -lib_cells {ISO_OR_EN}
```

Library definition of ISO_OR_EN cell:

```
cell (ISO_OR_EN) {
      is_isolation_cell : true;
      pin (A) { direction : input; }
      pin (B) { direction : input; isolation_cell_enable_pin
: true; }
      pin (Y) { direction : output; function : "A|B"; }
   }
```

In the above example, `clamp_value` is given as `0`, whereas the isolation cell given in the `map_isolation_cell` command is an OR cell. So the following violation is reported:

```
Cell 'ISO_OR_EN' specified in map_isolation_cell command
'iso_r1' is not an 'AND' cell whereas clamp value is specified
as '0'
```

Default Severity Label

Fatal, Warning, Info

Rule Group

`UPF Check`

Reports and Related Files

None

UPF_lowpower06

Checks the existence of the map_level_shifter_cell command

When to Use

This is a setup rule and always runs by default.

Description

The *UPF_lowpower06* rule reports if the `map_level_shifter_cell` command is not provided with the `set_level_shifter` command. This rule does not report if the `-no_shift` argument is specified.

Language

Verilog, VHDL

Parameter(s)

- *lp_check_map_ls_cmd*:Default values is `1`. Set this parameter to `0` to ignore the missing `map_level_shifter_cell` command corresponding to a `set_level_shifter` command in the UPF.

Constraint(s)

UPF

- *map_level_shifter_cell* (Mandatory)
- *set_level_shifter* (Mandatory)

Messages and Suggested Fix

The following message appears when the *map_isolation_cell* command is not provided for the rule `<rule-name>` specified with the `set_isolation` command:

[INFO] Command 'map_level_shifter_cell' should be given for level shifter rule '<rule-name>' defined by set_level_shifter, for using a specific level shifter cell

For information on debugging, click *How to Debug and Fix*.

Potential Issues

The violation message explicitly states the potential issues.

Consequences of Not Fixing

Not specifying *map_level_shifter_cell* or specifying non-level-shifter cells in *map_level_shifter_cell* command might lead incorrect cell being inserted leading to incorrect level shifting logic in the design.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation message, specify the *map_level_shifter_cell* command for the rule reported in the violation message.

Example Code and/or Schematic

For the following UPF file snippet, the *UPF_lowpower0*6 rule reports a violation message because the *map_level_shifter_cell* command has not been specified.

```
set_level_shifter ls_r1 -domain PD -applies_to inputs -rule
low_to_high -location self
```

```
Command 'map_level_shifter_cell' should be given for level
shifter rule 'ls_r1' defined by set_level_shifter, for using a
specific level shifter cell
```

Default Severity Label

Info

Rule Group

UPF Check

Reports and Related Files

None

UPF_lowpower07

Checks the create_power_switch and map_power_switch commands in the library

When to Use

This is a setup rule that runs automatically for the SpyGlass Power Verify solution.

Description

The *UPF_lowpower07* rule reports violations in the following cases:

- The power switch, specified by using the *map_power_switch* command, does not exist in the library.
- The control pin of the power switch, specified by using the `-control_port` argument of the *create_power_switch* command, does not exist in the library.
- The Power-out supply pin of the power switch, specified by using the `-output_supply_port` argument of the *create_power_switch* command, does not exist in the library.
- The Power-out supply pin of the power switch, specified by using the `-input_supply_port` argument of the *create_power_switch* command, does not exist in the library.
- When there is inconsistency between `on_state` and `off_state` specified in the *create_power_switch* command.
- When there is inconsistency between `off_state/on_state` and the switch function attribute.

Language

Verilog, VHDL

Parameter(s)

- *lp_allow_pin_alias*: Default is `no`. Set this parameter to `yes` to allow pin aliases in the *create_power_switch* command, instead of the actual pins in library cells.

Constraint(s)

None

Messages and Suggested Fix

Message 1

The following message appears when the power switch *<pwr-swtch-name>*, specified by using the *map_power_switch* command, is not found in the library:

[UPF_lowpower07_1][WARNING] Power Switch '<pwr-swtch-name>', specified through 'map_power_switch', is not found in the library

Potential Issues

This violation appears in the following cases:

- A specified cell name may not be correct.
- All the required library files are not provided.
- A typo.

Consequences of Not Fixing

If you do not fix this violation, the rules using the power switch may produce incorrect results.

How to Debug and Fix

To fix this violation:

- Specify the correct library path.
- Define the cell used in the *map_power_switch* command in the library.

Message 2

The following message appears when the Control/Power-out supply/Power-in supply pin, *<pin-name>*, of the power switch *<pwr-swtch-name>*, specified by using the *create_power_switch* command, is not found in the library:

[UPF_lowpower07_2][WARNING] <Control | Power-out supply | Power-in supply> pin '<pin-name>' of the Power Switch '<pwr-swtch-name>', specified through 'create_power_switch' command, is not found in the library

Potential Issues

This violation appears when the pin specified as the control port of the power switch does not exist.

Consequences of Not Fixing

If you do not fix this violation, no pin is considered as the control port for the power switch, since the pin specified as the control port is non-existent.

How to Debug and Fix

To fix this violation:

- Ensure that the Control/Power-out supply/Power-in supply pin is present in the library definition of the cell.
- Specify the correct library path.

Message 3

The following message appears when the `on_state` or `off_state` has been incorrectly specified in the *create_power_switch* command:

```
[UPF_lowpower07_3][WARNING] Incorrect on_state/off_state
specified in create_power_switch command
```

Potential Issues

This violation appears when the ON state and the OFF state defined in the *create_power_switch* command are not consistent with each other.

Consequences of Not Fixing

If you do not fix this violation, the power switch in the design may not function properly.

How to Debug and Fix

To fix this violation, specify the correct ON state and OFF state in *create_power_switch* command.

Message 4

The following message appears when there is an inconsistency between the `off_state/on_state` and the switch function attribute:

```
[UPF_lowpower07_4][WARNING] Inconsistency exists between
'off_state/on_state' and switch_function attribute on pin
'<pin-name>' of cell '<cell-name>'
```

Potential Issues

This violation appears when the OFF state or the ON state is not consistent with the switch function defined in the library cell mapped through the *map_power_switch* command.

Consequences of Not Fixing

If you do not fix this violation, the power switch in the design may not function properly.

How to Debug and Fix

To fix this violation, specify the correct OFF state/ON state and switch function so that they are consistent.

Example Code and/or Schematic

Example 1

For the following code, the *UPF_lowpower07* rule reports a violation because the control pin `SLEP` of the power switch `HEADBUF16_X1M_A9TR`, specified by using the `-control_port` argument of the *map_power_switch* command, does not exist in the library.

```
set_design_top top

# power domain definitions
create_power_domain Vtop
create_power_domain VA -elements ua

# supply nets
create_supply_port VDD
add_port_state VDD -state {on_state 1.0}
create_supply_net VDD -domain Vtop
connect_supply_net VDD -ports VDD

create_supply_port VSS
add_port_state VSS -state {on_state 0.0}
create_supply_net VSS -domain Vtop
connect_supply_net VSS -ports VSS

create_supply_net VDDX -domain VA
```

```
set_domain_supply_net Vtop -primary_power_net VDD
primary_ground_net VSS
set_domain_supply_net VA -primary_power_net VDDX
primary_ground_net VSS

# power switch cells
create_power_switch PS_rule1 -input_supply_port {VDDG VDD}
output_supply_port {VDD VDDX} -control_port {SLEP ENA}
domain VA
map_power_switch PS_rule1 -domain VA -lib_cells {
HEADBUF16_X1M_A9TR }
add_port_state PS_rule1/VDD -state {on_state 1.0} -state
{off_state off}
```

Example 2

For the following code, the UPF_lowpower07 rule reports *Message 3* and *Message 4* because the ON state (!SLEEP) and the OFF state (SLEEP1) are not consistent with each other and the switch-function (!SLEEP) is not consistent with the OFF state (SLEEP1).

Power Switch command in UPF

```
create_power_switch PS_rule2 \
-input_supply_port {vddin VDD0} \
 -output_supply_port {vddout VDDOUT} \
-control_port {SLEEP ENA} -domain VD0 \
 -on_state {ON vddin {  !SLEEP }} \
-off_state {OFF {  SLEEP1  }}

add_port_state PS_rule2/vddout -state {on1 1.5} -state {on2
2.0} -state {off off}
map_power_switch PS_rule2 -domain VD0 -lib_cells {
power_switch }
```

Library cell mapped through above code

```
/*POWER SWITCH CELL*/
```

```
cell (power_switch) {
        is_pad : true;
        switch_cell_type : coarse_grain;
       pg_pin (vddin) { pg_type : primary_power; direction :
input;}
        pg_pin ( vddout ) { pg_type : internal_power;
direction : output ; switch_function : "!SLEEP"; pg_function
: "vddin" ;}
        pg_pin ( vss ) {pg_type : primary_ground;direction :
input;}
        pin ( SLEEP ) { switch_pin : true; related_power_pin
: vddin; related_ground_pin : vss; }
}
```

Default Severity Label

Warning

Rule Group

UPF Check

Reports and Related Files

None

UPF_lowpower08

Reports improperly defined create_pst commands

When to Use

This is a setup rule and always runs by default.

Description

The *UPF_lowpower08* rule reports when PSTs are not properly defined using the *create_pst* command in the provided UPF file.

Language

Verilog, VHDL

Parameter(s)

- *lp_pst_merge_new*: Default is `yes`. Set this parameter to `no` to use the old PST merging flow.
- *lp_pst_merge_no_caching*: Default is `no`. Set this parameter to `yes` to use the PST merging flow without caching of the merged PSTs.
- *lp_use_equivalence_in_pst*: Default is `no`. Set this parameter to yes to use the equivalence information from the *set_equivalent* command in the PST.
- *lp_treat_equivalent_psw_output_supplies_different*: Default is `no`. Set this parameter to `yes` to enable the rule to not consider output supply of power switches with same control signal and same input as equivalent.

Constraint(s)

UPF

- *create_pst* (Mandatory)
- *add_port_state* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the *<state-name>* state of PST *<pst-name>*, value (*<supply-value>*) of the supply *<supply-*

name> does not match with the value (*<related-supply-value>*) of its related supply *<related-supply-name>*:

[UPF_lowpower08_1][FATAL] `In state '<state-name>' of PST '<pst-name>', value of supply '<supply-name>(<supply-value)' does not match with the value of related supply '<related-supply-name>(<related-supply-value>)'`

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation message, review the *create_pst* and *add_port_state* commands and update the value of the supply to match the value of the related supply specified in the violation message.

Message 2

The following message appears when the `<supply-state>` of the supply `<supply-name>` used in the `<pst-state-name>` state of the `<pst-name>` PST is not defined with the *add_port_state* command:

[UPF_lowpower08_2][FATAL] `Undefined state '<supply-state>' of supply '<supply-name>' used in PST '<pst-name>' for state '<pst-state-name>'`

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation message, review the *add_port_state* command and update it to include the state information for the supply port.

Message 3

The following message appears when an error occurs while merging the list of states <state-name-list> of PST <pst-name> to master state table (MST), as no state is found with the required supply values <supply-value-list>:

[UPF_lowpower08_3][ERROR] Error in merging states <state-name> with PST '<pst-name>': No state found with supply values <supply-value-list>

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

The SpyGlass Power Verify solution will ignore the power state *<state-name>* and will continue with the checking.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation message, review the supplies specified in the *create_pst* command. Update the states and the supply values of the *create_pst* and *add_port_state* commands.

Message 4

The following message appears for the state <state-name> of PST <pst1-name> when there is no matching state in PST <pst2-name>:

[UPF_lowpower08_4][WARNING] For state '<state-name>' of PST '<pst1-name>', there is no matching state in PST '<pst2-name>'. Reason: There is no common state of supplies <supply(supply-value)-list> in both the PSTs

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

The SpyGlass Power Verify solution will ignore the power state *<state-name>* of PST *<pst1-name>* and will continue with the checking.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation message, review the state *<state-name>* mentioned in the PST *<pst1-name>*. The supply voltages *<supply(supply-value)-list>* specified in the first PST *<pst1-name>* are not specified in the second PST *<pst2-name>*. The state *<state-name>* would be ignored and it is assumed that the design functions without this ignored power state.

Message 5

The following message appears while merging the PST `<pst-name>` because there is no MST formed after merging the PSTs `<pst-name-list>`:

[UPF_lowpower08_5][FATAL] Error merging PST '<pst-name>', there are no matching state in MST formed after merging the PSTs '<pst-name-list>' respectively

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation message, review the state mentioned in the PST, the supply voltages are specified in the PST is not specified in the TOP level PST/MST. The state would be ignored and it is assumed that the design functions as power states specified in the TOP PST.

Message 6

The following message appears for the state `<state-name>` of PST `<pst1-name>` when there is no matching state in PST `<pst2-name>`:

[UPF_lowpower08_6][ERROR] Error in merging state '<state-name>' of PST '<pst1-name>', there is no matching state in PST '<pst2-name>'. Reason: There is no common state of supplies <supply(supply-value)-list> in both the PST's

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

The SpyGlass Power Verify solution will ignore the power state *<state-name>* of PST *<pst-name>* and will continue with the checking.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation message, review the state mentioned in the PST, the supply voltages are specified in the first PST is not specified in the second PST. The state would be ignored and it is assumed that the design functions without the ignored power state.

Message 7

The following message appears when two nets, *<net1-name>* and *<net2-name>,* are connected to each other through some supply ports using the `connect_supply_net` commands, but are specified at different operating voltages values, *<net1-state-value>* and *<net2_state_value>,* respectively. This message is generated for state *<state-name>* of domain *<domain-name>*.

[UPF_lowpower08_10][FATAL] `Supply '<net1-name>' is at voltage '<net1-state-value> volts' as compared to related supply '<net2-name>' which is at voltage '<net2_state_value> volts' mentioned in state '<state-name>' of add_power_state command for domain '<domain-name>'`

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation message, review the state of the objects specified (power domains or supply sets) in the logic expression of the *add_power_state* command. These objects have conflicting voltage values for the reported nets. Update the object states to fix this.

Message 8

The following message appears when an object corresponding to *<object-name>* cannot be resolved. This message is generated for state *<state-name>* corresponding to power domain *<domain-name>*.

[UPF_lowpower08_11][FATAL] Unable to resolve object '<object-name>' mentioned in logic expression of add_power_state command ‘<state-name>' written for '<domain-name>

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation message, review and update the object name and/or object state name specified in the logic expression of the *add_power_state* command.

Message 9

The following message appears when state *<object-state-name>* of object *<object-name>* cannot be resolved to a valid state. This message is generated for state *<state-name>* of power domain *<domain-name>*.

[UPF_lowpower08_12][FATAL] Unable to resolve state ‘<object-state-name>' of object '<object-name>' mentioned in logic expression of add_power_state command '<state-name>' written for ‘<domain-name>'

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation message, review and update the object name and/or object state name specified in the logic expression of the *add_power_state* command.

Message 10

The following message appears when state multiple `connect_supply_net` commands are given for a leaf-level pin:

```
[UPF_lowpower08_13][ERROR] Multiple connect_supply_net commands <list of [filename:line-number] of connect_supply_net commands> specified for pin '<pin-name>'. Command specified at [<filename>:<line-number>] will be considered
```

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

The SpyGlass Power Verify solution will ignore the all the `connect_supply_net` commands for *<pin-name>* except the one reported in the message, and will continue with the checking.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation messages, review and update the various *connect_supply_net* commands for the reported pin.

Message 11

The following message appears when the *<state-name>* state of PST *<pst-name>*, value (*<supply-value>*) of the supply *<supply-name>* does not match with the value (*<related-supply-value>*) of its related supply *<related-supply-name>*:

```
[UPF_lowpower08_14][WARNING] Ignoring state '<state-name>' of PST '<pst-name>' while PST merging, value of supply '<supply-name>(<supply-value)' does not match with the value of related supply '<related-supply-name>(<related-supply-value>)'
```

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

The SpyGlass Power Verify solution will ignore the reported power state and continue with the checking.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation message, review the *add_pst_state* command and update the value of the supply to match the value of the related supply specified in the violation message.

Message 12

The following message appears when an illegal *<state-name>* state of domain *<domain-name>* is defined in the *add_pst_state* command:

[UPF_lowpower08_15][WARNING] Ignoring state '<power-state>' of domain '<domain-name>' while PST merging, as state '<power-state>' of object '<supply-set>' is mentioned as illegal in the corresponding add_power_state command

Potential Issues

An illegal state will not be considered for checking, as it should not be reached in any design state.

Consequences of Not Fixing

If an illegal state is mentioned then no checking takes place, based on that state.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation message, make the state as legal or remove the state, in the *add_pst_state* command.

Message 13

The following message appears when an illegal *<state-name>* state of the supply set is defined in the *add_pst_state* command:

[UPF_lowpower08_16][WARNING] Ignoring state '<state-name>' of domain '<domain>' while PST merging, as state '<state-name>' of object '<object-name>' is mentioned as illegal in the corresponding add_power_state command

Potential Issues

An illegal state will not be considered for checking, as it should not be reached in any design state.

Consequences of Not Fixing

If an illegal state is mentioned then no checking takes place, based on that state.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation message, make the state as legal or remove the state, in the *add_pst_state* command.

Message 14

The following message appears when **Message 11** is reported for all the states of the power state table (PST) <*PST-name*>:

[UPF_lowpower08_19][ERROR] PST '<PST-name>' has been ignored as all its states were ignored due to voltage mismatch between related supplies

Potential Issues

The PST will not be considered during the rule checking.

Consequences of Not Fixing

The SpyGlass Power Verify solution will ignore the reported PST and continue with the checking.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation message, review the *create_pst* and *add_pst_state* commands.

Message 15

The following message appears when state multiple *connect_supply_set* commands are given for a leaf-level instance:

[UPF_lowpower08_17][ERROR] Multiple connect_supply_set commands <list-of-[filename:line-number]-of-connect_supply_set-commands> specified for pin '<pin-name>'. Command specified at [<filename>:<line-number>] will be considered

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

The SpyGlass Power Verify solution will ignore all the *connect_supply_set* commands for instances except the one reported in the message, and will continue with the checking.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation message, review and update the various `connect_supply_set` commands for the reported instance.

Message 16

The following message appears when same pg_type is associated with more than one supply net for any given instance using the *connect_supply_set* command:

[UPF_lowpower08_18][ERROR] Trying to connect pg_type, '<pg-type>' of instance '<instance-name>' to nets created in commands <list-of [filename:line-number]>. Net created at [<filename>:<line-number>] will be considered

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

The SpyGlass Power Verify solution will ignore the all the nets for instances except the one reported in the message, and will continue with the checking.

How to Debug and Fix

To resolve the violation message, review and update the various `connect_supply_set` commands for the reported instance.

Message 17

The following message appears when supply nets from both *connect_supply_set* and *connect_supply_net* commands are connected to the same leaf level pin:

[UPF_lowpower08_20][ERROR] connect_supply_net is defined for the Pin, '<pin-name>' at [<filename>:<line-number>]. Net connected from connect_supply_net will be considered

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

The SpyGlass Power Verify solution will ignore the net connected from the connect_supply_set command for leaf level pins except the one reported in the message, and will continue with the checking.

How to Debug and Fix

To resolve the violation message, review and update the various connect_supply_set commands for the reported pin.

Message 18

The following message appears when pins of pg_type specified in the -connect argument of the *connect_supply_set* command is not present in the instances specified for the -elements option:

[UPF_lowpower08_21][ERROR] Pins of PG Type 'pg_type' are not present in element '<instance-name>'

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

The SpyGlass Power Verify solution will ignore the -connect option of the connect_supply_set command for the reported pg_type, and will continue with the checking.

How to Debug and Fix

To resolve the violation messages, review and update the various connect_supply_set commands for the reported instance.

Message 19

The following message appears when the lp_pst_merge_message.csv report is generated:

[UPF_lowpower08_22][Info] Refer spreadsheet report for PST states which could not be merged with any PST because of absence of common state of their supplies in PSTs

You can click the message to open the *lp_pst_merge_message* report.

Example Code and/or Schematic

Example 1

For the following UPF file snippet, the *UPF_lowpower08* rule reports a violation message because the state information has not been specified for the supply port:

```
create_pst P_pst –supplies {VDD u01/VDD_SB0_SW VDD_SB1
VDD_SB2}

add_pst_state pst0 –pst P_pst
```

To resolve the violation, specify the state information as shown in the following snippet:

```
create_pst P_pst –supplies {VDD u01/VDD_SB0_SW VDD_SB1
VDD_SB2}

add_pst_state pst0 –pst P_pst –state {VDD_N SW_on SB1_H
SB2_H}
```

Example 2

Consider the following code snippet:

```
source main.upf
create_pst pst_top -supplies {VDDA VDDB}
add_pst_state ab11 -pst pst_top -state {A_on  B_on}
add_pst_state ab10 -pst pst_top -state {A_on  B_off}

create_pst pst_top_2 -supplies {VDDC VDDB}
add_pst_state cb11 -pst pst_top_2 -state {C_on  B_on}
add_pst_state cb01 -pst pst_top_2 -state {C_off  B_on}
add_pst_state cb1L -pst pst_top_2 -state {C_on  B_low}
```

The following messages is reported because an error occurred while merging the list of states `ab10(pst_top)` with PST `pst_top_2`, as no state is found with the required supply values:

```
Error in merging states 'ab10(pst_top)', with PST 'pst_top_2':
```

```
No state found with supply values VDDB(off)
```

Example 2

Consider the following code snippet:

```
source main.upf

create_pst pst_top -supplies {VDDA VDDB}
add_pst_state ab11 -pst pst_top -state {A_on  B_on}
add_pst_state ab10 -pst pst_top -state {A_on  B_off}

create_pst pst_top_2 -supplies {VDDA VDDB}
add_pst_state ab11 -pst pst_top_2 -state {A_off  B_on}
```

The following message is reported while merging the PST `pst_top_2` because there is no MST formed after merging the PSTs `pst_top`:

```
Error merging PST '/pst_top_2', there are no matching state in
MST formed after merging the PSTs '/pst_top' respectively
```

Example 3

Consider the following UPF2.0 snippet:

```
create_supply_set ss1
create_power_domain TOP -include_scope -supply { primary ss1
}

create_supply_net VSS -domain TOP
create_supply_port VSS -domain TOP
connect_supply_net VSS  -ports VSS
add_port_state VSS -state {FULL_ON 0.0} -state { OFF off }

create_supply_net VDD -domain TOP
create_supply_port VDD -domain TOP
connect_supply_net VDD  -ports VDD
add_port_state VDD -state {FULL_ON 1.2} -state { OFF off }

create_supply_set ss1 -function { ground VSS } -function {
power VDD } -update

connect_supply_net VDD -ports { VDD }
```

```
connect_supply_net VSS -ports { VSS }

add_power_state ss1 \
                    -state on1 { -supply_expr {power == FULL_ON
&& ground == FULL_ON}} \
                     -state off {-supply_expr {power == OFF &&
ground == OFF } }

add_power_state TOP \
                     -state top1 { -logic_expr { PD == s1 &&
primary == on1 } } \
                     -state top2 { -logic_expr { PD == s2 &&
primary == off } }
```

The following message is reported because there is no supply set or power domain named PD in the scope of power domain TOP:

```
Unable to resolve object 'PD' mentioned in logic expression of
add_power_state command 'top1' written for 'TOP'
```

Example 4

Consider the following UPF2.0 snippet:

```
create_supply_set ss1
create_power_domain TOP -include_scope -supply { primary ss1
}

create_supply_net VSS -domain TOP
create_supply_port VSS -domain TOP
connect_supply_net VSS  -ports VSS
add_port_state VSS -state {FULL_ON 0.0} -state { OFF off }

create_supply_net VDD -domain TOP
create_supply_port VDD -domain TOP
connect_supply_net VDD  -ports VDD
add_port_state VDD -state {FULL_ON 1.2} -state { OFF off }

create_supply_set ss1 -function { ground VSS } -function {
power VDD } -update
```

```
connect_supply_net VDD -ports { VDD }
connect_supply_net VSS -ports { VSS }

add_power_state ss1 \
                    -state on1 { -supply_expr {power == FULL_ON
&& ground == FULL_ON}} \
                     -state off {-supply_expr {power == OFF &&
ground == OFF } }

add_power_state TOP \
                     -state top1 { -logic_expr {primary ==
on2 } } \
                    -state top2 { -logic_expr {primary == off
} }
```

The following message is reported because state `on2` is not specified for supply set `/ss1`:

```
Unable to resolve state 'on2' of object '/ss1' mentioned logic
expression of add_power_state command 'top1' written for 'TOP'
```

Example 5

Consider the following UPF2.0 snippet:

```
upf_version 2.0

set_design_top top

create_supply_port VDDp
create_supply_net  VDDn
connect_supply_net VDDn -ports {VDDp}

create_supply_port VDD1p
create_supply_net  VDD1n
connect_supply_net VDD1n -ports {VDD1p}

create_supply_port VDD2p
create_supply_net  VDD2n
```

```
connect_supply_net VDD2n -ports {VDD2p inst1/and1/VDD}

create_supply_port VDD3p
create_supply_net  VDD3n
connect_supply_net VDD3n -ports {VDD3p inst1/and1/VDD}

create_supply_port VSSp
create_supply_net VSSn
connect_supply_net VSSn -ports {VSSp}

create_supply_set ss -function {power VDDn} -function {ground
VSSn}
create_supply_set ss1 -function {power VDD1n} -function
{ground VSSn}
create_supply_set ss2 -function {power VDD2n} -function
{ground VSSn}
create_supply_set ss3 -function {power VDD3n} -function
{ground VSSn}

create_power_domain TOP -include_scope -supply {primary ss}
create_power_domain PD1 -elements {inst1} -supply {primary
ss1}
create_power_domain PD2 -elements {inst2} -supply {primary
ss2}
create_power_domain PD3 -elements {inst3} -supply {primary
ss3}

add_port_state VDDp -state {on1 1.1}
add_port_state VDD1p -state {on1 1.1}
add_port_state VSSp -state {def 0.00}
add_port_state VDD2p -state {on1 1.1}
add_port_state VDD3p -state {on1 1.1}

create_pst ps1 -supplies {VDDn VDD1n VDD2n VDD3n VSSn}
add_pst_state s1 -pst ps1 -state {* * * * * }
```

The following message is reported because the connect_supply_net command is specified multiple times for the pin `top.inst1.and1.VDD`:

```
Multiple connect_supply_net commands [const.upf:16],
[const.upf:20] specified for pin 'top.inst1.and1.VDD'. Command
specified at [const.upf:16] will be considered
```

Example 6

Consider the following commands in the UPF snippet:

```
add_power_state ss1
-state on{ -supply_expr { VDD1 == on1 && VSS == on }}
-state on1 {-supply_expr { VDD1 == on2 && VSS == on }
illegal}

add_power_state PD1
-state s1 { -logic_expr {ss1 == on && ss2 == on} -illegal}
-state s2 { -logic_expr {ss1 == on1 && ss2 == on1} }
```

For the above example, the following message is reported because state `s1` of domain `PD1` is illegal:

```
Ignoring state 's1' of domain 'PD1' while PST merging, it is
mentioned as illegal in the add_power_state command
```

Example 7

Consider the following commands in the UPF snippet:

```
add_power_state ss1
-state on{ -supply_expr { VDD1 == on1 && VSS == on }}
-state on1 {-supply_expr { VDD1 == on2 && VSS == on }
illegal}

add_power_state PD1
-state s1 { -logic_expr {ss1 == on && ss2 == on} -illegal}
-state s2 { -logic_expr {ss1 == on1 && ss2 == on1} }
```

For the above example, the following message is reported because the state `on1` of the supply set `ss1` is illegal:

```
Ignoring state 's2' of domain 'PD1' while PST merging, as state
'on1' of object 'ss1' is mentioned as illegal in the
corresponding add_power_state command
```

Example 8

Consider the following UPF snippet:

```
upf_version 2.0

set_design_top top
create_power_domain TOP -include_scope
create_power_domain PD1 -elements {inst1}
create_power_domain PD2 -elements {inst2}
create_power_domain PD3 -elements {inst3}

create_supply_port VTOP
create_supply_net  VTOP
connect_supply_net VTOP -ports {VTOP}
create_supply_net  VDD2 -domain PD2
connect_supply_net VDD2 -ports {VTOP}

create_supply_net  VDD1 -domain PD1
connect_supply_net VDD1 -ports {VTOP}

create_supply_net  VDD3 -domain PD3
connect_supply_net VDD3 -ports {VTOP}

create_supply_port VSS
create_supply_net VSS
create_supply_net VSS -domain PD2 -reuse
create_supply_net VSS -domain PD1 -reuse
create_supply_net VSS -domain PD3 -reuse
connect_supply_net VSS -ports VSS

set_domain_supply_net TOP -primary_power_net VTOP -
primary_ground_net VSS
set_domain_supply_net PD2 -primary_power_net VDD2 -
primary_ground_net VSS
set_domain_supply_net PD1 -primary_power_net VDD1 -
primary_ground_net VSS
set_domain_supply_net PD3 -primary_power_net VDD3 -
primary_ground_net VSS
```

```
add_port_state VTOP -state {on1 1.0} -state {on2 1.1}  -state
{on2 1.2}  -state {on3 1.3}  -state {on4 1.4}  -state {off1
off}
add_port_state VSS -state {def 0.00}

create_pst ps1 -supplies {VTOP VDD1 VDD2 VDD3}
add_pst_state s1 -pst ps1 -state {on1 on2 on3 on4}
add_pst_state s2 -pst ps1 -state {on1 off1 on3 on4}
```

For the above example, the following messages are reported because VTOP, VDD1, VDD2, and VDD3 are connected to each other but have different port states in the PST states s1 and s2:

```
Ignoring state 's1' of pst '/ps1' while PST merging, value of
supply '/VDD1(1.10)' does not match with the value of related
supply '/VDD2(1.30)'

Ignoring state 's2' of pst '/ps1' while PST merging, value of
supply '/VDD1(off)' does not match with the value of related
supply '/VDD2(1.30)'

PST '/ps1' has been ignored as all its states were ignored due
to voltage mismatch between related supplies
```

Example 9

Consider the following UPF snippets:

File: main.upf

```
upf_version 2.0
set_design_top top

create_power_domain TOP -include_scope
create_supply_port VDD
create_supply_net VDD -domain TOP
connect_supply_net VDD -ports VDD

create_supply_port VSS
create_supply_net VSS -domain TOP
connect_supply_net VSS -ports VSS
```

```
set_domain_supply_net TOP -primary_power_net VDD -
primary_ground_net VSS

add_port_state VDD -state {on 1.0 } -state { on1 1.1 } -state
{ on2 1.2 } -state { off off }
add_port_state VSS -state {gnd 0} -state { off off}

load_upf ip.upf -scope inst1
load_upf ip.upf -scope inst2

set_scope /

connect_supply_net VDD -ports { inst1/VDD inst2/VDD }
connect_supply_net VSS -ports { inst1/VSS inst2/VSS }



create_pst mainpst -supplies            {   VDD    VSS }
add_pst_state s1 -pst mainpst -state    {   on     gnd }
add_pst_state s2 -pst mainpst -state    {   on1    gnd }
add_pst_state s3 -pst mainpst -state    {   off    gnd }
```

File ip.upf

```
upf_version 2.0
create_power_domain PD  -include_scope

create_supply_port VDD -domain PD
create_supply_net VDD  -domain PD
connect_supply_net VDD -ports VDD

create_supply_net VSS  -domain PD

add_port_state VDD -state { on 1.0  } -state { on1 1.1 } -
state { on2 1.2 }
add_port_state VDD -state { off off }

add_port_state VSS -state { on 0 }
```

```
connect_supply_net VSS -ports VSS

create_supply_net VDDg -domain PD
set_domain_supply_net PD -primary_power_net VDDg -
primary_ground_net VSS

create_power_switch psw -domain PD -input_supply_port { VDD
VDD } -output_supply_port { VDDg VDDg } -control_port { SLEEP
SLEEP}
map_power_switch psw -domain PD -lib_cells { PS1 }
add_port_state psw/VDDg -state {on 1.0 } -state {off off} -
state { on1 1.1 } -state { on2 1.2 }

set_isolation iso1 -domain PD -applies_to inputs -
isolation_power_net VDD -isolation_ground_net VSS -
clamp_value 0 -location self
set_isolation_control iso1 -domain PD -isolation_signal
SLEEP -isolation_sense low

create_pst pst1 -supplies            {   VDD   VDDg  VSS }
add_pst_state s1 -pst pst1 -state    {  on2    on2    gnd }
add_pst_state s2 -pst pst1 -state    {   on1    off   gnd }
add_pst_state s3 -pst pst1 -state    {   off    off   gnd }
```

For the above example, the following violation messages are reported because some states of VDD used in top level PST are not used in the IP level PST and vice versa:

```
Error in merging state 's1' of PST '/mainpst', there is no
matching state in PST '/inst1/pst1'. Reason: There is no common
state of supplies 'VDD(1.00), VSS(0.0)' in both the PST's

For state 's1' of PST '/inst1/pst1', there is no matching state
in PST '/mainpst'. Reason: There is no common state of supplies
'VDD(1.20), VSS(0.0)' in both the PST's

For state 's1' of PST '/inst2/pst1', there is no matching state
in PST '/mainpst'. Reason: There is no common state of supplies
'VDD(1.20), VSS(0.0)' in both the PST's
```

Default Severity Label

Error/Fatal/Warning

Rule Group

UPF Check

Reports and Related Files

None

UPF_lowpower09

Checks if supply port or supply net defined in UPF file is either undriven or multiple driven

When to Use

These are setup rules and are enabled by default.

Description

The *UPF_lowpower09* rule reports in the following situations:

- Top level supply port (with direction input) or power switch output port does not have an active ON state defined.
- A supply port is neither connected to a top level supply port nor a power switch output port.
- A supply port is either connected to multiple top level supply ports or power switch output ports.
- A supply net is neither connected to a top level supply port nor a power switch output port.
- A supply net is either connected to multiple top level supply ports or power switch output ports.

A supply port, or a supply net connecting two ports, if specified to be driven with multiple ports in the UPF file may lead to design error.

Specify the top-level supply ports with direction input and power switch output ports with at least one ON state using *add_port_state* command. Also connect all the other supply ports in the design to either a top level supply port or a power-switch output port using *connect_supply_net* command.

The PG pins of the given library cell, specified in the *set_supply_node* SGDC constraint, are considered as valid drivers.

Language

Verilog, VHDL

Parameter(s)

- *lp_output_pg_pin_as_supply_port*: Default values is no. Set this parameter to yes to assume the output PG Pin of the library cell as a supply port in the UPF.

- *lp_pg_direction_as_supply_port*: Default values is not set to any value. Set this parameter to `output` or `inout` or both, for the directions of PG pins that need to be considered as valid drivers of supply nets.
- *lp_pg_type_as_supply_port*: Default values is not set to any value. Set this parameter to `internal_power` or `internal_ground` or both, for the type of PG pins that need to be considered as valid drivers of supply net.

Constraint(s)

SGDC

set_supply_node: Use this constraint to specify PG pin of a library cell that needs to be considered as a valid driver.

Messages and Suggested Fix

Message 1

The following message appears when the top-level supply port *<port-name>* does not have any state defined as active `ON`:

[UPF_lowpower09_1][FATAL] `Supply port '<port-name>' does not has an active ON state`

Potential Issues

This violation message appears if there is power intent error or supply ports do not have an active on state.

Consequences of Not Fixing

Not fixing this violation may result that active state values cannot be inferred for associated supply nets. This impacts the downstream rules functioning.

How to Debug and Fix

To fix this violation, add an active on state for the supply port in the *add_port_state* command.

Message 2

The following message appears when the power-switch output port *<port-name>* does not have any state defined as active `ON`:

[UPF_lowpower09_2][FATAL] `Power switch output port '<port-`

name>' does not has an active ON state

Potential Issues

This violation message appears if there is power intent error or power switch output port does not have an active ON state.

Consequences of Not Fixing

Not fixing this violation may result that active state values cannot be inferred for associated supply nets. This impacts the downstream rules functioning.

How to Debug and Fix

To fix this violation, add an active on state for the supply port in the *add_port_state* command.

Message 3

The following message appears when the supply port *<port-name>* is neither connected to a top-level supply port nor to a power-switch output port:

[UPF_lowpower09_7][ERROR] Supply port '<port-name>', connected to supply net '<net-name>', is not connected to a top level input supply port or power switch output port

Potential Issues

This violation message appears if every supply port is not routed to top level supply port or a power switch output port.

Consequences of Not Fixing

Not fixing this violation may result in redundant supply ports.

How to Debug and Fix

To debug this violation, specify a *connect_supply_net* command for this port to establish a connection with either a top-level supply port or power switch output port.

Message 4

The following message appears when the scope level supply port <port-name> is not driven by o a top-level supply port or a power-switch output Port:

[UPF_lowpower09_3][WARNING] Supply port <port-name> is not connected to a top level input supply port or power switch

output port

Potential Issues

This violation message appears if there are undriven supply ports in the power intent.

Consequences of Not Fixing

Voltage value of the supply port is not inferred.

How to Debug and Fix

To fix this violation, specify a driver for this supply port in the power Intent file.

Message 5

The following message appears when the supply net *<net-name>* is neither connected to a top-level supply port nor a power-switch output port:

[UPF_lowpower09_4][INFO] Supply net '<net-name>' is not connected to a top level input supply port or power switch output port

Potential Issues

This violation message appears if there are undriven supply nets in the power content.

Consequences of Not Fixing

Voltage value of the supply net is not inferred.

How to Debug and Fix

To fix this violation, specify the connection of the supply net using the *connect_supply_net* command to the concerned supply port or power switch output port.

Message 6

The following message appears when the supply net *<net-name>* is connected to multiple top-level supply ports and/or power-switch output ports:

[UPF_lowpower09_5][WARNING] Supply net '<net-name>' is connected to multiple top level input supply ports and/or power switch output ports <list-of-ports>

Potential Issues

This violation message appears if there are multiple driven supply net in the power intent. It may result in functional errors in the design.

Consequences of Not Fixing

Not fixing this violation may result that voltage values from the first power port in the connection list are inferred for the supply net.

How to Debug and Fix

To fix this violation, review the power intent file and remove the extra *connect_supply_net* commands written for the supply net.

Message 7

The following message appears when the supply net *<net-name>* is unused:

[UPF_lowpower09_6][INFO] Supply net '<net-name>' is not used, neither defined as primary net of any domain nor connected to any leaf instance pin using 'connect_supply_net' command

A supply net is considered as unused if it is not used as a primary net of a domain, not connected to any leaf level pin using the `connect_supply_net` command, not used as supply net in the `set_related_supply_net` command or not used as a supply net in the `set_port_attributes` command.

Potential Issues

This violation message appears if there are redundant supply nets.

Consequences of Not Fixing

Not fixing this violation may result that supply net is missed from using or in redundant supply net.

How to Debug and Fix

To fix this violation, review the power intent and see whether the mentioned supply net was meant to be used in the first place. Else remove the *create_supply_net* command which created this supply net.

Message 8

The following message appears if a supply net is defined with `-resolve {parallel}` option in the UPF file but it is driven by only one driver:

[UPF_lowpower10][Warning] Supply net <net-name> has only one driver <port_name>,is declared with resolve parallel

Potential Issues

This violation message appears if a supply net is defined with `-resolve {parallel}` option in the UPF file but it is driven by only one driver.

Consequences of Not Fixing

Not fixing this violation may result in incorrect rule checking.

How to Debug and Fix

To fix this violation, review the power intent file and ensure that either multiple drivers are specified or the `-resolve {parallel}` option is not specified.

Example Code and/or Schematic

Example 1

For the following snippet, the *UPF_lowpower09* rule reports a violation for supply port `IN`.

```
create_supply_port IN -direction in
create_supply_net VDD_B_net -domain PD_top
connect_supply_net VDD_B_net -ports {VDD_B IN}
```

Therefore, Message 1 is reported.

Example 2

For the following snippet, the *UPF_lowpower09* rule reports a violation for power switch output port `psw/pswOut`.

```
create_power_domain PD_top -include_scope
create_supply_net pswOut_net -domain PD_top
create_power_switch psw -domain PD_top -output_supply_port
{pswOut pswOut_net} -input_supply_port {pswIn1 VDD1_net} -
input_supply_port {pswIn2 VDD2_net}
```

Therefore, Message 2 is reported.

Example 3

For the following snippet, the *UPF_lowpower09* rule reports a violation for supply port `A/VDD1`.

```
create_power_domain PD_top -include_scope
set_scope A
```

```
create_supply_port VDD1
add_port_state VDD1 -state {ON 1.0} -state {OFF off}
```

Therefore, Message 3 is reported.

Example 4

For the following snippet, the *UPF_lowpower09* rule reports a violation for supply net `VDD_B_net`.

```
create_power_domain PD_top –include_scope
create_supply_port VDD_B
add_port_state VDD_B -state {ON 1.0} -state {OFF off}

create_supply_port IN -direction in
create_supply_net VDD_B_net -domain PD_top
connect_supply_net VDD_B_net -ports {VDD_B IN}
```

Therefore, Message 6 is reported.

Example 5

For the following snippet, the *UPF_lowpower09* rule reports a violation for supply net `IN_net`.

```
create_supply_port IN
add_port_state IN -state {ON 2.0} -state {OFF off}
create_supply_net IN_net  -domain PD_top
connect_supply_net IN_net  -ports IN
```

Therefore, Message 7 is reported.

Default Severity Label

Warning, Info, Fatal, Error

Rule Group

`UPF Check`

Reports and Related Files

None

UPF_lowpower10

Reports if isolation control signal of a domain is specified with different isolation senses

When to Use

This is a setup rule and it runs by default.

Description

The *UPF_lowpower10* rule reports a violation if you specify different isolation senses to the same isolation control signal for a domain by using the `-isolation_sense` argument of the *set_isolation_control* command.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

None

Messages and Suggested Fix

The following message appears when different isolation sense is specified for the isolation signal *<sig-name>* of the domain *<dom-name>*:

[WARNING] Isolation signal '<sig-name>' is specified with different isolation sense for domain '<dom-name>'

Potential Issues

This violation appears if two or more isolation strategies are written for a domain and same isolation signal is specified, but the isolation senses are different.

Consequences of Not Fixing

If you do not fix this violation, the logic value reaches the enable pin of the isolation cell, which may lead to incorrect output of isolation. This causes improper isolation of domain ports.

How to Debug and Fix

To fix this violation, ensure that if you specify more than one isolation sense for the same isolation signal, they should have same value.

Example Code and/or Schematic

For the following code, the *UPF_lowpower10* rule reports a violation because two different isolation senses, high and low, are specified for the same isolation signal, ISOSIG, for the domain VD1:

```
set_isolation ISO1 -domain VD1 -applies_to outputs

set_isolation_control ISO1 -domain VD1 -isolation_signal
ISOSIG -isolation_sense low

set_isolation ISO2 -domain VD1 -applies_to inputs

set_isolation_control ISO2 -domain VD1 -isolation_signal
ISOSIG -isolation_sense high
```

Default Severity Label

Warning

Rule Group

UPF Check

Reports and Related Files

None

UPF_lowpower11

Reports undeclared supply net for associated domain

When to Use

These are setup rules and by default enabled.

Description

The *UPF_lowpower11* rule reports when a supply net is not declared in the domain where it is being used.

The rule reports violation messages when the:

- primary power/ground net of a domain is not declared for that domain.
- supply net connected to a power switch input/output port is not declared in the associated domain.
- supply net connected to a power switch input/output port is not defined using the *create_supply_net* command.
- It is mandatory to redefine a supply net for each of the associated domains using `reuse` option of the *create_supply_net* command.
- power and ground supply net specified in the *set_related_supply_net* command are not declared in the associated domain.
- the `set_related_supply_net` command is specified on hierarchical ports on the domain boundary.

Language

Verilog, VHDL

Parameter(s)

- *lp_csn_on_ground_pg_pin*: Default value is `0`. Set this parameter to `1` to check the availability of ground supply net mentioned in the `set_related_supply_net` command, in the associated domain.

Constraint(s)

UPF

- *create_supply_net* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the primary *<net-type>* (power/ground) net *<net-name>* of domain *<dom-name>* is not defined for that domain:

[WARNING] Primary <net-type> net '<net-name>' of domain '<dom-name>' is not declared using command 'create_supply_net' for associated domain

Potential Issues

A supply net declared in the UPF file is created in the domain specified. A primary power and ground net specified as primary nets should have a corresponding *create_supply_net* with the domain specified using –domain field.

Consequences of Not Fixing

In physical implementation the supply net would not be created in desired domain hierarchy.

How to Debug and Fix

Create the supply net in the domain by using the *create_supply_net* and `-domain` options.

Message 2

The following message appears when the supply net *<net-name>* of power switch *<port-type>* (input/output) port is not available in the domain *<domain-name>*:

[WARNING] Supply net '<net-name>' connected to the power switch <port-type> port is not available in domain '<domain-name>'

Potential Issues

A supply net connected to the supply output port of specified using the *create_power_switch* should be created in the domain specified. A net connected to the output supply port should have a corresponding *create_supply_net* with the domain specified using `–domain` field.

Consequences of Not Fixing

Not fixing this violation may result that in physical implementation the supply net would not be created in desired domain hierarchy.

How to Debug and Fix

To debug this violation specify the supply net connected to power switch output using command *create_supply_net* and specify the power switch domain in `-domain` field.

Message 3

The following fatal message appears when the supply net *<net-name>* of power switch *<port-type>* (input/output) port *<port-name>* is not created using the *create_supply_net* command:

```
[FATAL] Supply net '<net-name>' connected to the power switch
<port-type> port '<port-name>' should be created with command
'create_supply_net'
```

Potential Issues

Violation message appears if net connected to the power switch output port are not inferred as the supply net and also the net connected to input port of the power switch are not inferred as the master supply net of the power switch output net.

Consequences of Not Fixing

Not fixing this violation may result that the voltage value associated with the power switch output port are not associated with any domain & possibly the master child relationship among the supply nets are not established. This causes functional errors in the downstream rules.

How to Debug and Fix

To fix this violation, specify the net connected with the power switch port with the *create_supply_net* command.

Message 4

The following message appears if the net *<net-name>* mentioned in the `set_related_supply_net` command is not available in the domain *<domain-name>*:

```
[WARNING] <Power/Ground> supply net <net-name> used in
set_related_supply_net command for object <port-name> is not
‘declared’ for domain <domain-name>
```

NOTE: *The violation for the ground pin is reported if the lp_csn_on_ground_pg_pin parameter is set to 1 or yes.*

Potential Issues

Violation message appears if a net mentioned in the `set_related_supply_net` command is not defined using the `create_supply_net` command in the associated domain.

Consequences of Not Fixing

Not fixing this violation may result in an inappropriate connection of data pins with the supply net.

How to Debug and Fix

To fix this violation, specify the net mentioned in the `set_related_supply_net` command using the *create_supply_net* command in the associated domain.

Message 5

The following message appears if the supply net *<net-name>* mentioned in the `connect_supply_net` command is not available in the domain *<domain-name>*:

[WARNING] Supply net '<net-name>' used in connect_supply_net command for object '<object-name>' is not ‘declared’ for domain '<domain-name>'

Potential Issues

Violation message appears if a net mentioned in the `connect_supply_net` command is not available in the domain.

Consequences of Not Fixing

Not fixing this violation may result in an inappropriate connection of data pins with the supply net.

How to Debug and Fix

To fix this violation, specify the net mentioned in the `connect_supply_net` command using the *create_supply_net* command in the associated domain.

Message 6

The following message appears if the supplies provided in the *set_related_supply_net* command do not match with the driver or receiver supply:

[WARNING] Supply '<supply-name>' given in set_related_supply_net command for port(s) (<port-list>) does

not match with <driver | receiver> supply '<inferred-supply-name>'

Potential Issues

An incorrect supply is provided in the *set_related_supply_net* command.

Consequences of Not Fixing

The user intent in the set_related_supply_net command will not get applied.

How to Debug and Fix

To fix this violation, make sure the supplies given in the set_related_supply_net command are connected to the receiver or driver supplies.

Message 7

The following message appears when the *lp_srsn_info* report is generated:

[INFO] Report lp_srsn_info.rpt has been generated

Message 8

The following message appears if the supply net specified in the *connect_supply_set* command is not available in the domain:

[WARNING] Supply net '<supply-net-name> ('<supply-function>' net of supply set '<supply-set-name>'>') used in connect_supply_set command for object '<object-name>' is not declared for domain '<domain-name>'

Potential Issues

A net mentioned in the connect_supply_set command is not available in the domain.

Consequences of Not Fixing

Not fixing this violation may result in an inappropriate connection of data pins with the supply net.

How to Debug and Fix

To fix this violation, specify the net mentioned in the *connect_supply_set* command using the *create_supply_net* command in the associated domain.

Example Code and/or Schematic

Example 1

Consider the following code snippet:

```
upf_version 1.0

create_power_domain PD_top -include_scope
create_power_domain PD_mid -elements Scope_A

create_supply_port VDD
add_port_state VDD -state {ON 1.0} -state {OFF off}

create_supply_port GND
add_port_state GND -state {ON 0.0}

create_supply_net VDD_net -domain PD_top
create_supply_net GND_net -domain PD_top

connect_supply_net VDD_net -ports VDD
connect_supply_net GND_net -ports GND

set_domain_supply_net PD_top -primary_power_net VDD_net
primary_ground_net GND_net
set_domain_supply_net PD_mid -primary_power_net VDD_net
primary_ground_net GND_net
```

The following messages are reported because primary power net `VDD_net` and primary ground net `GND_net` of domain `PD_mid` are not defined for that domain:

```
Primary power net 'VDD_net' of domain 'PD_mid' is not declared
using command 'create_supply_net' for associated domain

Primary ground net 'GND_net' of domain 'PD_mid' is not declared
using command 'create_supply_net' for associated domain
```

Example 2

Consider the following code snippet:

```
upf_version 1.0
create_power_domain PD_top -include_scope
```

```
create_power_domain PD_mid -elements Scope_A

create_supply_port VDD
add_port_state VDD -state {ON 1.0}
create_supply_port GND
add_port_state GND -state {ON 0.0} -state {OFF off}

create_supply_net VDD_net -domain PD_top
create_supply_net GND_net -domain PD_top
create_supply_net GND_net -domain PD_mid -reuse
create_supply_net VDD_off  -domain PD_top
create_supply_net VDD  -domain PD_top

connect_supply_net VDD_net -ports VDD
connect_supply_net VDD -ports VDD
connect_supply_net GND_net -ports GND

set_domain_supply_net PD_top -primary_power_net VDD_net
primary_ground_net GND_net
set_domain_supply_net PD_mid -primary_power_net VDD_off
primary_ground_net GND_net

# power switch cells
create_power_switch PS_rule1 -input_supply_port {VDDI VDD}
control_port {ON ENA[0:1]} -output_supply_port {VDDO
VDD_off} -domain PD_mid
map_power_switch PS_rule1 -domain PD_mid -lib_cells  { PWRSW
}
add_port_state PS_rule1/VDDO -state {on_state 1.0} -state
{off_state off}
```

The following message is reported because supply net `VDD_net` connected to the power switch input port is not available in domain `PD_mid`:

```
Supply net 'VDD' connected to the power switch input port is
not 'available' in domain 'PD_mid'
```

The following message is reported because the supply net `VDD_off` connected to the power switch output port is not available in domain

PD_mid:

```
Supply net 'VDD_off' connected to the power switch output port
is not 'available' in domain 'PD_mid'
```

Example 3

In the following example, supply net VDD connected to the power switch port VDDI is not defined using *create_supply_net:*

```
upf_version 1.0

create_power_domain PD_top -include_scope
create_power_domain PD_mid -elements Scope_A

create_supply_port VDD
Add_port_state VDD -state {ON 1.0}

create_supply_port GND
add_port_state GND -state {ON 0.0} -state {OFF off}

create_supply_net VDD_net -domain PD_top
create_supply_net GND_net -domain PD_top
create_supply_net GND_net -domain PD_mid -reuse
create_supply_net VDD_off  -domain PD_mid

connect_supply_net VDD_net -ports VDD
connect_supply_net GND_net -ports GND

set_domain_supply_net PD_top -primary_power_net VDD_net
primary_ground_net GND_net
set_domain_supply_net PD_mid -primary_power_net VDD_off
primary_ground_net GND_net

# power switch cells
create_power_switch PS_rule1 -input_supply_port {VDDI
VDD_net}
control_port {ON ENA[0:1]} -output_supply_port {VDDO
VDD_off} -domain PD_mid
map_power_switch PS_rule1 -domain PD_mid -lib_cells  { PWRSW
```

```
}
add_port_state PS_rule1/VDDO -state {on_state 1.0} -state
{off_state off}
```

In the above example, supply net VDD_net connected to the power switch port VDDI is not defined by using *create_supply_net*. Therefore, the rule will violate this one. To avoid rule violation, the following line has to be added in the UPF file:

```
create_supply_net VDD_net -domain PD_mid -reuse
```

Example 4

Consider the following code snippet:

```
upf_version 2.0
set_design_top top

create_supply_port vccG_upf
create_supply_port vccC_upf
create_supply_port vss

add_port_state vccC_upf -state {ON 0.75} -state {OFF off}
add_port_state vccG_upf -state {SWITCH_ON 0.9000} -state
{SWITCH_OFF off}
add_port_state vss -state {OFF 0.0}

create_power_domain my_domain -elements {my_levelshifter}

create_supply_net vccG_upf
create_supply_net vccC_upf -domain my_domain
create_supply_net vss
create_supply_set SS_G -function {power vccG_upf} -function
{ground vss}
create_supply_set SS_C -function {power vccC_upf} -function
{ground vss}

## top_domain
create_power_domain top_domain -include_scope -supply
{primary SS_G} -available_supplies {}
```

```
create_power_domain my_domain -supply {primary SS_C} -update

## my_domain
connect_supply_net vccG_upf -ports {vccG_upf}
connect_supply_net vss -ports {vss}
connect_supply_net vccC_upf -ports {vccC_upf}

set_related_supply_net -object_list in1 -power vccC_upf
ground vss


set_level_shifter my_ls -domain my_domain \
      -elements {my_levelshifter/in1 my_levelshifter/
in2my_levelshifter/out1} \

      -applies_to inputs \
      -rule both \
      -location automatic
```

The following message is reported because the power supply net `vccC_upf` mentioned in the `set_related_supply_net` is not available in the domain `top_domain`:

```
Power supply net 'vccC_upf' used in set_related_supply_net
command for object 'top.in1' is not 'declared' for in domain
'top_domain'
```

Example 5

Consider the following code snippet:

```
set_related_supply_net -object_list my_levelshifter/in1 -
power vccG_upf -ground vss
```

The following message is reported because an incorrect supply `vccG_upf` is given in the *set_related_supply_net* command for the port `top.my_levelshifter.in1`:

```
Supply ‘vccG_upf’ given in set_related_supply_net command for
port(s) (top.my_levelshifter.in1) does not match with source
supply ‘vcc’
```

Default Severity Label

Warning, Fatal, Info

Rule Group

UPF Check Rules

Reports and Related Files

lp_srsn_info: Lists the ports on which correct supply information using the *set_related_supply_net* command is specified.

UPF_lowpower12

Reports when appropriate isolation / level shifter / repeater strategy on domain element is not specified

When to Use

This is a setup rule and always runs by default.

Description

The *UPF_lowpower12* rule reports a violation when proper isolation/level shifter/repeater strategy is not defined for all the domain elements using *set_isolation*/*set_level_shifter*/*set_repeater* commands, respectively.

This rule reports following scenarios:

- Element specified in the *set_isolation*/*set_level_shifter*/*set_repeater* command is not on any domain boundary or is on an incorrect domain boundary.
- Element specified in the *set_isolation*/*set_level_shifter*/*set_repeater* command is an INOUT port/pin.
- Multiple isolation/level shifter/repeater strategies are specified for the same element, all inputs, or all outputs of a domain.
- Element specified in the *set_isolation*/*set_level_shifter*/*set_repeater* command contains an incorrect power domain.
- Element specified in the *set_isolation*/*set_level_shifter*/*set_repeater* command has multiple source or sink domains.
- Multiple isolation strategies, which have the same precedence, are applicable for the element specified in the *set_isolation*/*set_level_shifter*/*set_repeater* command.
- Incorrect source or sink is specified for the element specified in the *set_isolation*/*set_level_shifter*/*set_repeater* command.

Language

Verilog, VHDL

Parameter(s)

- *lp_skip_aon_buf*: Default value is `1`. Set the parameter to `0` to not skip always-on buffers to find an isolation crossing.

- *lp_flag_iso_ls_strategy_on_domain_sub_hier*: Default value is `no`. Set the parameter to `yes` to report solation/level shifter strategy specified on domain sub-hierarchy.

Constraint(s)

UPF

- *set_isolation*
- *set_level_shifter*
- *set_repeater*

Messages and Suggested Fix

Message 1

The following message appears when the element *<pin-name>* specified in *<cmd-name>* (*set_isolation*/*set_level_shifter*/*set_repeater*) command is not on any domain boundary:

[ERROR] Element '<pin-name>' specified in <cmd-name> command [<file-name>:<line-no>] is not on domain boundary. No <chk-type> checks performed on this element

where, *<chk-type>* can be isolation, repeater, or level shifter.

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when an isolation/level shifter/repeater strategy is specified on an inout port *<port-name>* using *<cmd-name>* (*set_isolation*/*set_level_shifter*/*set_repeater*) command:

[WARNING] Inout port '<port-name>' specified in <cmd-name> command [<file-name>:<line-no>]. No <chktype> checks performed on this element

where, *<chk-type>* can be isolation, repeater, or level shifter.

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when multiple *<chk-type>* (*isolation*/*level shifter/repeater*) strategies are specified for

<reported-element(s)> of the *<dom-name>* domain:

```
[ERROR] Multiple <chk-type> strategies '<rulename1>' [<file-name1>:<line-num1>], '<rule-name2>' [<file-name2>:<line-num2>] specified for '<reported-element(s)>' of domain '<dom-name>'. No <chk-type> checks performed on <reported-element>
```

where, *<reported-element(s)>* is the hierarchical name of a domain element or *all inputs* or *all outputs* when multiple isolation/level shifter/repeater specifications are given for an element or all inputs or all outputs of the domain, respectively.

For debugging information, click *How to Debug and Fix*.

Message 4

The following message appears for the specified *<strategy-name>* strategy when the *<element-hier-name>* element specified in the *set_isolation*/*set_level_shifter*/*set_repeater* command contains an incorrect *<domain-name>* power domain:

command contains an incorrect *<domain-name>* power domain:

```
[ERROR] Strategy (set_isolation | set_level_shifter | set_repeater) <strategy-name> [<upf-file-name>:<upf-line-number>] is ignored on element <element-hier-name> as element does not belong to specified domain '<domain-name>'
```

For debugging information, click *How to Debug and Fix*.

Message 5

The following message appears when the *<element-hier-name>* element specified in the *set_isolation*/*set_repeater* command has multiple source or sink domains *<domain-name-list>:*

```
[ERROR] Element '<element-hier-name>' has multiple <source | sink> domains (<domain-name-list>). No isolation strategy is applied on this element
```

For debugging information, click *How to Debug and Fix*.

Message 6

The following message appears when multiple isolation/repeater strategies *<isolation/repeater-strategy-list>* that have the same precedence are applicable for the *<element-hier-name>* element

specified in the *set_isolation*/*set_repeater* command:

[ERROR] Multiple <isolation | repeater> strategies (<isolation/repeater-strategy-list>' having same precedence are applicable for element '<element-hier-name>'. No isolation strategy is applied on this element

For debugging information, click *How to Debug and Fix*.

Message 7

The following message appears when an incorrect source or sink is specified for the *<element-hier-name>* element specified in the *set_isolation*/*set_repeater* command:

[ERROR] Incorrect <source | sink> '<supply-set-name>' specified for element '<element-hier-name>' in <set_isolation | set_repeater> command '<upf-isolation-command-name>'. No <isolation | repeater> strategy is applied on this element

For debugging information, click *How to Debug and Fix*.

Message 8

The following message appears when the *<element-hier-name>* element specified in the *set_level_shifter* command has multiple source or sink domains *<domain-name-list>:*

[ERROR] Element '<element-hier-name>' has multiple <source | sink> domains (<domain-name-list>). No level shifter strategy is applied on this element

For debugging information, click *How to Debug and Fix*.

Message 9

The following message appears when multiple `level shifter` strategies *<lshifter-strategy-list>* that have the same precedence are applicable for the *<element-hier-name>* element specified in the *set_level_shifter* command:

[ERROR] Multiple level shifter strategies (<lshifter-strategy-list>' having same precedence are applicable for element '<element-hier-name>'. No level shifter strategy is applied on this element

For debugging information, click *How to Debug and Fix*.

Message 10

The following message appears when an incorrect source or sink is specified for the *<element-hier-name>* element specified in the *set_level_shifter* command:

[ERROR] Incorrect <source | sink> '<domain-name>' specified for element '<element-hier-name>' in set_level_shifter command '<upf-lshifter-command-name>'. No level shifter strategy is applied on this element

Potential Issues

This violation message explicitly specify the potential issues.

Consequences of Not Fixing

SpyGlass Power Verify does not apply an isolation/level shifter strategy to an element for which this rule is reporting a violation. Since no isolation/level shifter strategy is being applied on an element, subsequent SpyGlass Power Verify rules (which run after this rule) perform rule checking based assuming there is no isolation/level shifter strategy on that element.

For example, if an isolation cell is inserted for an element on which you have written an isolation strategy and that strategy is reported by this rule. The strategy is not applied to that element. This causes an isolation cell to be redundant and different checks, such as checking its location, its functionality, steady state value on its output etc. are not performed by SpyGlass Power Verify rules.

How to Debug and Fix

Ensure that design has properly specified isolation/level shifter strategy.

Example Code and/or Schematic

Example 1

For the following snippet, the UPF_lowpower12 rule reports a violation because `u1` is the domain boundary and not `u1/u2`, hence element `u1/u2/out1` is not on domain boundary.

```
create_power_domain PD1 –elements {u1}
set_isolation ISO1 –domain PD1 –elements {u1/u2/out1}…..
```

Therefore, Message1 is reported.

Example 2

For the following snippet, the UPF_lowpower12 rule reports a violation because `inout1` is an inout port of domain `PD1`.

```
set_isolation ISO1 –domain PD1 –elements {u1/inout1}…..
```

Therefore, Message 2 is reported.

Example 3

For the following snippet, the UPF_lowpower12 rule reports a violation because multiple isolation strategies (ISO1 and ISO2) have been specified for same element {u1/out1}.

```
set_isolation ISO1 –domain PD1 –elements {u1/out1} –
no_isolation

set_isolation ISO2 –domain PD1 –elements {u1/out1} –
clamp_value 0
```

Therefore, Message 3 is reported.

Example 4

For the following snippet, the *UPF_lowpower12* rule reports a violation because the isolation strategy `ISO1` is specified for domain `PD1`. However, element `{u2/out1}` specified in the strategy belongs of domain `PD2 (u2)`.

```
create_power_domain PD1 –elements {u1}

create_power_domain PD2 –elements {u2}

set_isolation ISO1 –domain PD1 –elements {u2/out1} –
clamp_value 0
```

Therefore, Message 4 is reported.

Example 5

Suppose an output port `Y` of domain `PD1_domain` has multiple fan-outs or sink domains: `PD2_domain` and `VD_domain`.

UPF

```
associate_supply_set PD1_domain_set -handle
PD1_domain.primary

associate_supply_set PD2_domain_set -handle
```

```
PD2_domain.primary

set_isolation ISO2 -domain PD1_domain -isolation_power_net
PD1_domain_supply -source PD1_domain_set -sink
PD2_domain_set
```

In this case, the rule considers Y as a non-uniform net. Therefore, no strategy is applied on it and Message 5 is reported.

Example 6

Suppose an output port Y of domain PD1_domain has PD1_domain as source domain and PD2_domain as destination domain.

UPF

```
associate_supply_set PD1_domain_set -handle
PD1_domain.primary

set_isolation ISO1 -domain PD1_domain -isolation_power_net
PD1_domain_supply -source PD1_domain_set -clamp_value 0


associate_supply_set PD2_domain_set -handle
PD2_domain.primary

set_isolation ISO2 -domain PD1_domain -isolation_power_net
PD1_domain_supply -sink PD2_domain_set -clamp_value 0
```

In this case, since both strategies, ISO1 and ISO2, are applicable on output Y and they have the same precedence, Message 6 is reported.

Example 7

Suppose the following isolation strategy is written on the output element A of domain VC. The *UPF_lowpower12* rule reports a violation if the sink of element A is different from VDDB.

```
set_isolation S1 -domain VC -elements A -sink VDDB
```

This strategy will not be applied to element A and Message 7 is reported.

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPF_lowpower13

Reports power switch output ports that do not have the same voltage values as the parent supply

When to Use

This is a setup rule and always runs by default.

Description

The *UPF_lowpower13* rule reports a violation when the states specified for the power switch output ports using the *add_port_state* command is not valid.

The states specified with the *add_port_state* command are considered invalid for a power switch output port when:

- No `off` state is defined.
- The state is not defined for the parent supply.
- Any of the states declared for parent supply is not defined.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

UPF

- *add_port_state* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when no `off` state is defined for the power switch output port *<port-name>*:

[WARNING] Power switch output port '<port-name>' has no 'off' state

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the state *<state-name>* of power switch output port *<port-name>* is not defined for the parent supply *<net-name>*:

```
[WARNING] State '<state-name>(<voltage-value> V)' of power
switch output port '<port-name>' is not defined for the parent
supply '<net-name>'
```

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when the state *<state-name>* of parent supply is not defined for the power switch output port *<port-name>:*

```
[ERROR] State '<state-name>(<voltage-value> V)' of parent
supply '<net-name>' is not used in the power switch output port
'<port-name>'
```

Potential Issues

The *add_port_state* command defines state information. The state information consists of a named state and voltage information for that state. An `off` state is also required so that a supply network can be shut down off chip.

Consequences of Not Fixing

The SpyGlass Power Verify solution cannot proceed with its analysis and will exit.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To resolve the violation messages:

- **Message 1**: Define an off state for the power switch output port.
- **Message 2**: The power switch output have same voltage level as the parent supply net connected to power switch input port. If there is a state defined for power switch output port but there is no corresponding state for power switch input port, then it is an error.
- **Message 3**: The power switch output have same voltage level as the parent supply net connected to power switch input port. If there is a

state defined for supply port connected to power switch input port but there is no corresponding state for power switch output port, then it is an error.

Example Code and/or Schematic

Example 1

For the following UPF snippet, neither the `off` state nor the state of the parent supply is defined for the power switch output port `VDD`. Therefore, the *UPF_lowpower13* rule reports violation messages:

```
add_port_state VDD

add_port_state VDD_SUB1 -state {SUB1_H 0.89} -state {SUB1_L
0.69}

add_port_state VDD_SUB2 -state {SUB2_H 0.89} -state {SUB2_L
0.69}
```

To resolve these violation messages, define an `off` state for the power switch output port and define the state of the parent supply for VDD, as shown below:

```
add_port_state VDD -state {VDD_N 0.99}

add_port_state VDD_SUB1 -state {SUB1_H 0.89} -state {SUB1_L
0.69}

add_port_state VDD_SUB2 -state {SUB2_H 0.89} -state {SUB2_L
0.69}

add_port_state GND -state {default 0}
```

Example 2

Consider the following code snippet:

```
set_design_top top

# power domain definitions
create_power_domain Vtop
create_power_domain VA -elements ua

# supply nets
create_supply_port VDD
```

```
add_port_state VDD -state {on_state 1.0}
create_supply_net  VDD -domain Vtop
connect_supply_net VDD -ports VDD

create_supply_port VSS
add_port_state VSS -state {on_state 0.0}
create_supply_net  VSS -domain Vtop
connect_supply_net VSS -ports VSS

create_supply_net  VDDX -domain VA

set_domain_supply_net Vtop -primary_power_net VDD
primary_ground_net VSS
set_domain_supply_net VA -primary_power_net VDDX
primary_ground_net VSS

# power switch cells
create_power_switch PS_rule1 -input_supply_port {VDDI
VDD} control_port {ON ENA[0:1]} -output_supply_port {VDDO
VDDX} -domain VA
map_power_switch PS_rule1 -domain VA -lib_cells  { PWRSW }
add_port_state PS_rule1/VDDO -state {on_state 1.1} -state
{off_state off}
```

For **Message 2**, the following violation is reported because the state `on_state(1.100 V)` of power switch output port `/PS_rule1/VDDO` is not defined for the parent supply `VDD`:

```
State 'on_state(1.100 V)' of power switch output port '/
PS_rule1/VDDO' is not defined for the parent supply 'VDD'
```

For **Message 3**, the following violation is reported because the state `on_state(1.000 V)` of parent supply `VDD` is not defined for the power switch output port `/PS_rule1/VDDO`:

```
State 'on_state(1.000 V)' of parent supply 'VDD' is not used in
the power switch output port '/PS_rule1/VDDO'
```

Default Severity Label

Warning

Rule Group

UPF Check

Reports and Related Files

None

UPF_lowpower14

Reports the mismatch between information of supply nets and their states in power state tables is complete

When to Use

This is a setup rule and always runs by default.

Description

The *UPF_lowpower14* rule reports a violation when there is a mismatch between information of supply nets and their states specified using *create_supply_net*/*add_port_state* commands and the power state tables for a defined scope.

This rule reports in the following situations:

- Any supply net created with the *create_supply_net* command for a scope is not used in the scope level PST(s).
- Any state of supply port specified in the *add_port_state* command is not used in the scope level PST(s).

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

UPF

- *create_supply_net* (Mandatory)
- *add_port_state* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the supply *<supply1>* created for the scope *<scope1>* is not defined in the scope level PST(s):

[UPF_lowpower14_1][WARNING] The supply '<supply1>' created in

scope '<scope1>' is not used in the scope level PST(s)

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when all the states in the list *<statelist>*, defined for the supply port *<port-name>*, are not used in the scope level PST(s):

[UPF_lowpower14_2][WARNING] The states '{<state-list>}' of the supply port '<port-name>' are not used in the scope level PST(s)

Potential Issues

This violation messages appear because:

- **Message 1**: The supply net created with the *create_supply_net* command for a scope is not defined in the scope level PST(s).
- **Message 2**: The supply port specified in the *add_port_state* command is not used in the scope level PST(s).

Consequences of Not Fixing

The results of SpyGlass Power Verify rules depend on the relationship between various supplies as mentioned in the power state table (PST). For example, whether isolation is needed between two cells powered by two supplies depends on the relationship between those two supplies in the PST. If any supply is not mentioned or any supply is partially represented in the PST, the complete relationship between the supplies cannot be determined which in turn might affect the results of SpyGlass Power Verify rules.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To fix these violations, perform the following:

- **Message 1**: Use unused supply reported in the message in the scope level PST.
- **Message 2**: Use all states of a supply in the power state table.

Example Code and/or Schematic

Example 1

For the following snippet, the *UPF_lowpower14* rule reports a violation because VDDY is not used in TOP_PST.

```
create_supply_port VDD
create_supply_net VDD –domain PD1
connect_supply_net VDD –ports {VDD}

create_supply_port VSS
create_supply_net VSS –domain PD1
connect_supply_net VSS –ports {VSS}

create_supply_port VDDY
create_supply_net VDDY –domain PD1
connect_supply_net VDDY –ports {VDDY}

create_pst TOP_PST –supplies {VDD VSS}
```

Therefore, *Message 1 is reported.*

Example 2

For the following snippet, the *UPF_lowpower14* rule reports a violation because off state of VDDX is not used in PST_1.

```
create_supply_port VDDX
add_port_state VDDX -state {VDDX_0 5.0} -state {VDDX_1 5.1} -
state {VDDX_2 5.2} -state {OFF off}
create_supply_net VDDX -domain PD1
connect_supply_net VDDX -ports {VDDX}
create_pst PST_1 -supplies {VSS VDDX}
add_pst_state s0 -pst PST_1 -state {VSS_0 VDDX_0}
```

```
add_pst_state s1 -pst PST_1 -state {VSS_2 VDDX_1}
add_pst_state s2 -pst PST_1 -state {VSS_2 VDDX_2}
```

Therefore, Message 2 is reported.

Default Severity Label

Warning

Rule Group

UPF Check

Reports and Related Files

None

UPF_lowpower15

Reports multi-supply cells that do not have connect_supply_net

When to Use

This is a setup rule and always runs by default.

Description

The *UPF_lowpower15* rule automatically generates a report when multi-supply cells that do not have an associated *connect_supply_net* command are identified. Use the report to ensure every multi-supply cells are handled correctly in the UPF file by associating them with *connect_supply_net* command. Therefore, no cells are overlooked.

NOTE: *The UPF_lowpower15 rule ignores multiple supply macro cells when all the signal pins are associated to the single power/ground supply pin.*

Language

Verilog, VHDL

Parameter(s)

- *lp_dump_ls_in_multi_supply_rpt*: Default is `1` or `yes`. Set the value as `0` to not to provide level shifter information in the *lp_multi_supply_instance* report.
- *lp_dump_ps_in_multi_supply_rpt*: Default is `1` or `yes`. Set this parameter as `0` to not provide power switch information in the *lp_multi_supply_instance* report.
- *lp_dump_unrelated_macro_cell_pgpin_in_multi_supply_rpt*: Default is `no`. Set this parameter as `yes` to report unconnected PG pins of macro cells that are not related to data pins.
- *lp_check_csn_on_internal_pg_pin*: Default is `yes`. Set this parameter as `no` to not report the missing `connect_supply_net` command on an internal power/ground pin for multi-supply cell.
- *lp_generate_missing_csn_violation*: Default is `no`. Set this parameter to `yes` to enable the *UPF_lowpower15* rule to report violations instead of generating report.

- *lp_report_drc_blockgroup_for_missing_csn*: Default is `yes`. Set this parameter as `no` to not include the *drc_blockgroup* section in the generated *lp_multi_supply_instance* report.

Constraint(s)

None

Messages and Suggested Fix

Message 1

If the value of the *lp_generate_missing_csn_violation* parameter is set to `no`, the following message appears when multi-supply instances that do not have a *connect_supply_net* command are identified and a `lp_multi_supply_instance.rpt` report is generated:

[ERROR] One or more multi supply instances with missing supply net command found. Please refer to the 'lp_multi_supply_instance.rpt' file for more details <file-location>

Message 2

If the value of the *lp_generate_missing_csn_violation* parameter is set to `yes`, the following message appears when multi-supply instances that do not have a *connect_supply_net* command are identified and no report is generated:

[ERROR] Found multi supply instance <'instance-name'> (Cell: <cell-name>, Domain: <domain-name>) with missing connect_supply_net command for pin(s) : <'pin-names'>

Potential Issues

The violation messages explicitly state the potential issue.

Consequences of Not Fixing

Cells with multiple power/ground pins should have *connect_supply_net* command specified in UPF so that pins, such as back up power, should be driven by the correct supply. Connecting wrong supply can lead to unsteady states in the design, which can result in design failure.

How to Debug and Fix

Review and update the UPF file for the instances mentioned in the *lp_multi_supply_instance* report. To view the report, navigate to the path mentioned in the violation message.

Example Code and/or Schematic

Example 1

For the following snippet, the *UPF_lowpower15* rule reports a violation and generates the *lp_multi_supply_instance* report.

```
connect_supply_net vddsafe -ports {vddsafe /sph_c3883/vddo}
```

Here, *connect_supply_net* command is missing for the `inst2/vddo` pin, therefore *lp_multi_supply_instance* will have a violation based on the type of cell:

```
connect_supply_net <supply_net> -ports top/inst2/vddo
#Master cell = HS45_LS_ONBFISOX18; Power Domain = TOP
```

The report lists the instances that do not have a *connect_supply_net* command. To resolve the violation, update the instances stated in the UPF file, as shown in the following:

```
connect_supply_net vddsafe -ports {vddsafe /sph_c3883/vddo
inst2/vddo}
```

Example 2

Consider the following UPF snippet:

```
connect_supply_net vddsafe -ports {vddsafe /sph_c3883/vddo}
```

When the value of the *lp_generate_missing_csn_violation* parameter is set to `yes`, the following violation appears:

```
Found multi supply instance 'top.sph_c3883' (Cell:
HS45_LS_ONBFISOX9, Domain: TOP) with missing
connect_supply_net command for pin(s) :' gndo'
```

To resolve the violation, update the instances stated in the UPF file, as shown below:

```
connect_supply_net gndsafe -ports {gndsafe /sph_c3883/gndo}
```

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

lp_multi_supply_instance

UPF_lowpower16

Checks the relationship between the bias net and the primary supply net

When to Use

This is a setup rule and always runs by default.

Description

The *UPF_lowpower16* rule reports a violation when there is a power state in which the bias (nwell) supply is OFF, but the primary supply is ON.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_set* (Mandatory)
- *create_supply_port* (Mandatory)
- *create_supply_net* (Mandatory)
- *create_pst* (Mandatory)
- *set_domain_supply_net* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when an incorrect bias net is specified in the supply set:

```
[ERROR] Incorrect bias 'power/ground' net specified in supply
set '<sup-set-name>' : Bias net '<bias-net-name>' is less
always-on than primary supply net '<supply-net-name>'
```

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when an incorrect bias net is specified for a domain:

```
[ERROR] Incorrect bias 'power/ground' net specified for domain
'<dom-name>' : Bias net '<bias-net-name>' is less always-on
than primary supply net '<supply-net-name>'
```

Potential Issues

The violation messages explicitly state the potential issue.

Consequences of Not Fixing

This is a design error that can lead to silicon failure.

How to Debug and Fix

The UPF command reported is highlighted in the Atrenta Console GUI.

To fix these violations, perform the following:

- **Message 1**: Ensure that the bias net specified in the supply set is more always-on.
- **Message 2**: Ensure that the bias net specified for the domain is more always-on.

Example Code and/or Schematic

In this snippet, an incorrect bias net is specified for both the domain and the supply set. Therefore, the *UPF_lowpower16* rule reports both violations.

```
…
create_supply_port VDTOP
add_port_state VDTOP -state {VT 1.2}
create_supply_port VDDB
add_port_state VDDB -state {V2 1.4} -state {V2_off off}
create_supply_set S1 -function {nwell VDDB}
associate_supply_set S1 -handle V1.primary
set_domain_supply_net V1 -primary_power_net VDTOP -
```

```
primary_ground_net VSS
create_supply_set S2 -function {power VDTOP} -function {nwell
VDDB}
…
```

Message 1 is reported for supply set S2 because error is at the creation of the supply set. While Message 2 is reported for domain V1 because the error is in the association of this domain with supply set S1.

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPF_lowpower17

Reports isolation power nets that are not declared in the domain in which the isolation cell will be inferred

When to Use

This is a setup rule and always runs by default.

Description

The *UPF_lowpower17* rule reports isolation power nets that are not declared in the domain in which the isolation cell will be inferred.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_supply_set* (Mandatory)
- *set_isolation* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when an isolation power net is not available in a domain:

```
[ERROR] Isolation power net '<iso-pwr-net-name>' specified in
set_isolation strategy '<iso-str-name>' is not available in a
domain '<domain-and-supply-name>' in which isolation cell will
be inferred for this strategy
```

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the location field of the set_isolation UPF command is set to automatic:

[WARNING] Unable to determine if Isolation power net '<iso-pwr-net-name>' specified in set_isolation strategy '<iso-str-name>' is available in the domain in which isolation cell will be inferred. Location field is set to 'automatic' in the set_isolation strategy

Message 3

The following message appears when isolation supply is not specified through isolation_power_net, isolation_ground_net, or isolation_supply_set:

[ERROR] Isolation supplies not specified for isolation strategy '<strategy-name>' for domain '<domain-name>'. Specify either with option 'isolation_supply_net/isolation_ground_net' or 'isolation_supply_set'

For debugging information, click *How to Debug and Fix*.

Potential Issues

- **Message 1**: Isolation power net should be either created or reused in the domain in which strategy infers isolation cell.
- **Message 2**: If the `-location` field is set to automatic, it is difficult to determine the domain in which isolation cell will be inferred. Therefore, it is not possible to check whether an isolation power net is either created or reused in that domain.
- **Message 3**: Isolation supply for the isolation strategy is not specified.

Consequences of Not Fixing

- **Message 1**: SpyGlass requires you to resolve this violation.
- **Message 2**: Design may not function as desired.
- **Message 3**: SpyGlass requires you to resolve this violation.

How to Debug and Fix

The *set_isolation* UPF command is highlighted in the Atrenta Console GUI. To fix these violations, perform the following:

- **Message 1**: Declare the isolation power net in a domain, as stated in the message.
- **Message 2**: The `-location` field of the *set_isolation* UPF command is set to automatic. Specify the `-location` field.

- **Message 3**: Specify the isolation supply for the isolation strategy using the isolation_power_net, isolation_ground_net, or isolation_supply_set commands.

Example Code and/or Schematic

Suppose the following isolation strategies are specified:

```
set_isolation ISO1 -domain A -isolation_power_net VA -
location self …
```

```
set_isolation ISO2 -domain A -isolation_power_net VA -
location automatic …
```

The *UPF_lowpower17* rule reports Message 1 if the isolation power net in domain A is not declared. In addition, Message 2 is reported because the `-location` field is set to automatic.

To resolve the violation messages, declare the isolation power net in domain A, as shown below:

```
create_supply_net VA -domain A -reuse
```

In addition, change the setting of the `-location` field:

```
set_isolation ISO2 -domain A -isolation_power_net VA -
location self …
```

Default Severity Label

Error/Warning

Rule Group

`UPF Check`

Reports and Related Files

None

UPF_lowpower18

Checks retention power/ground net specified in set_retention with primary power/ground supply net of domain

When to Use

This is a setup rule and always runs by default.

Description

The *UPF_lowpower18* rule reports violation messages when the power/ground net or supply set specified in the *set_retention* command is not more always-on than the primary power/ground net of the domain. In addition, it checks if the power net is not specified in the *set_retention* command using either *-retention_supply_set <retention_supply_name>* or *-retention_power_net <net_name>*.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

UPF Commands

- *set_retention* (Mandatory)
- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when the *-retention_supply_set <retention_supply_name>* or the *-retention_power_net <net_name>*

specified is not more always-on than the primary power net of the domain:

[ERROR] Incorrect <power net | supply set> specified in field <retention_supply_set | retention_power_net> of command set_retention '<strategy-name>': Retention power net '<ret-power-name>' is not more on than primary power net '<primary-supply-name>'

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when the *-retention_power_net <net_name>* is not specified in the *set_retention* command:

[ERROR] Power net not specified in set_retention command '<strategy-name>' using either retention_supply_set or retention_power_net

Message 3

The following message appears when the retention supply net *<net-name>* is not available in the domain *<domain-name>*:

[ERROR] Power net '<net-name>' specified in set_retention command '<command>' is not 'available' in domain '<domian-name>

Message 4

The following message appears when the retention supply and domain supply of a corresponding design element are switched off together:

[ERROR] Retention supply ‘<supply-name>' for the retention strategy '<strategy-name>' for domain '<domain-name>' is not more ON than the domain supply ‘<supply-name>'

NOTE: *This message is not reported if the entire chip is switched off, that is, all the power supplies are in off state.*

Message 5

The following message appears if the element mentioned in `-elements` argument of the `set_retention` command is not present in the same domain, as specified in the `-domain` argument of the `set_retention` command:

[ERROR] Power domain '<PD-name>' of element '<element-name>' is not same as the domain '<domain-name>' given in set_retention command

Potential Issues

The violation messages explicitly state the potential issues.

Consequences of Not Fixing

SpyGlass requires you to resolve these violations.

How to Debug and Fix

The *set_retention* UPF command is highlighted in the Atrenta Console GUI. To fix these violations, perform the following:

- **Message 1**: Specify the *-retention_supply_set <retention_supply_name>* or the *-retention_power_net <net_name>* such that it is more always-on than the primary power net of the domain.
- **Message 2**: Specify the *-retention_power_net <net_name>* argument of the *set_retention* command.
- **Message 3**: Specify the retention supply net for the domain in the *set_retention* UPF command.
- **Message 4**: Make sure that the retention supply and domain supply are not switched off at the same time.
- **Message 5**: Make sure that the element specified in the `set_retention` command is present in the same domain, as specified in the `-domain` argument of the `set_retention` command.

Example Code and/or Schematic

Example 1

This example illustrates when **Message 1** is reported. In this example, the power net specified in the *-retention_power_net <net_name>* argument, `VDD`, is less always-on than the primary power net, `VDD_LP`, of the `V2` domain.

```
# Defining the Power Domains
create_power_domain V1
create_power_domain V2 -elements {mid_inst}
# Defining the supply nets
create_supply_port VDD
add_port_state VDD -state {on_state 1.2} -state {off_state
```

```
off}
create_supply_net  VDD -domain V1
create_supply_port VDD_LP
add_port_state VDD_LP -state {on_state 1.2}
create_supply_net  VDD_LP -domain V1

set_domain_supply_net V2 -primary_power_net VDD_LP -
primary_ground_net VSS
set_retention RET1 -domain V2 -retention_power_net VDD
```

Example 2

This example illustrates when **Message 2** is reported. In this example, the *-retention_power_net <net_name>* argument of the *set_retention* command has not been specified.

```
set_retention RET1 -domain V2
set_retention_control RET1 -domain V2 -save_signal {rst high}
-restore_signal {rst low }
```

Example 3

This example illustrates when **Message 4** is reported. In this example, the `-retention_power_net <VDD1>` argument of the `set_retention` command has been specified for domain `VD1`. But domain `VD1`'s primary supply is `VDD`. So, `VDD` and `VDD1` are switched off together in state `s1`.

```
create_power_domain VD1 -elements {inst1}
set_domain_supply_net VD1 -primary_power_net VDD
primary_ground_net VSS

set_retention ret1 -domain VD1 -retention_power_net VDD1
retention_ground_net VSS

add_port_state VDD -state {on1 0.7} -state {off1 off}
add_port_state VDD1 -state {on1 0.85} -state {off1 off}
```

```
create_pst ps1 -supplies {VDD VDD1}
add_pst_state s1 -pst ps1 -state {off1 off1}}
```

Example 4

Consider the following example snippet:

```
create_power_domain PD1 -elements {mid/inst1 mid}
create_power_domain PD2 -elements mid/inst2

set_retention RET2 -domain PD2   -elements {mid/inst1/fd1
mid/inst2/a1}
map_retention_cell RET2 -domain PD2 -elements {mid/inst1/
fd1}  -lib_cells FD
```

For the above example, the UPF_lowpower18 rule reports the following violation because the `set_retention` command is written for domain PD2, but `element mid/inst1/fd1` is present in domain PD1:

```
Power domain 'PD1' of element 'top.mid.inst1.fd1' is not same
as the domain 'PD2' given in set_retention command
```

Default Severity Label

Error

Rule Group

`UPF Check`

Reports and Related Files

None

UPF_lowpower19

Checks for conflict between save/restore information specified in set_retention and the retention cells available in library

When to Use

This is a setup rule and always runs by default.

Description

Rule *UPF_lowpower19* reports missing *map_retention_cell* command.

The rule reports violation messages when:

- the library does not have any retention cells.
- the retention cell specified in *map_retention_cell* is not found in library.
- there is a conflict between save/restore information specified in *set_retention* and retention cells available in library.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

UPF Commands

- *set_retention* (Mandatory)
- *map_retention_cell* (Mandatory)
- *set_retention_control* (Mandatory)
- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)

Messages and Suggested Fix

Message 1

The following message appears when a retention cell is not found in the library:

```
[ERROR] UPF retention strategies could not be applied: No retention cell found in library
```

For debugging information, click *How to Debug and Fix*.

Message 2

The following message appears when a cell is specified in the *map_retention_cell* command of UPF, but the cell does not have any `retention_cell` attribute in the library file:

```
[ERROR] Cell '<cell-name>' specified as retention cell, does not have 'retention_cell' attribute in the library cell description
```

For debugging information, click *How to Debug and Fix*.

Message 3

The following message appears when a UPF retention strategy <strategy-name> could not be applied:

```
[ERROR] UPF retention strategy <strategy-name> could not be applied: <reason>
```

Message 4

The following message appears when the *map_retention_cell* command is missing for a UPF retention strategy command:

```
[INFO] Command 'map_retention_cell' should be given for retention rule '<upf-strategy-name>' defined by set_retention, for using a specific retention cell
```

Potential Issues

The violation messages explicitly state the potential issues. In addition for Message 2, the <reason> can be any of the following:

- Specified retention cell <ret-cell-name> not found in library.

- Distinct save and restore signals specified but retention cell `<ret-cell-name>` has single save-restore pin.
- Both save and restore signals are not specified but retention cell `<ret-cell-name>` has distinct save and restore pins.
- Distinct save and restore signals specified but no retention cell in Library has distinct save and restore pins.
- Both save and restore signals are not specified but no retention cell in Library has a single save-restore pin.
- No retention cell found in library with cell type `<ret-cell-name>`.

Consequences of Not Fixing

Since you cannot apply retention cells in case of conflicting information, SpyGlass requires you to resolve these violations.

How to Debug and Fix

To fix the Message 1 violation, specify the retention cell in the library.

To fix the Message 2 violation, ensure compatibility of the retention cell information specified in the power intent and the library.

To fix the Message 4 violation, ensure that the *map_retention_cell* command is specified for the retention rule reported in the violation message.

Example Code and/or Schematic

Example 1

This example illustrates some scenarios in which the *UPF_lowpower19* rule reports violation message. Consider the following UPF code snippet:

```
set_retention RET1 -domain UA -retention_power_net VDDB -
retention_ground_net VSS

set_retention_control RET1 -domain UA -save_signal {save
high} -restore_signal {rst low }

map_retention_cell RET1 -domain UA -lib_cells RET_DFF
```

In this example, set_retention is specified. If no retention cell is in the library, Message 1 is reported:

```
UPF retention strategies could not be applied: No retention
cell found in library
```

In the snippet, distinct save and restore signals are specified. If the retention cell RET_DFF is not found in library, the following violation message is reported:

```
UPF retention strategy RET1 could not be applied: Specified retention cell RET_DFF not found in library
```

In the snippet, the distinct save and restore signals are specified. If the retention cell RET_DFF does not have distinct save and restore signals, the following violation message is reported.

```
UPF retention strategy RET1 could not be applied: Distinct save and restore signals specified but retention cell RET_DFF has single save-restore pin
```

Example 2

Consider the following snippet:

```
set_retention RET1 -domain UA -retention_power_net VDDB -retention_ground_net VSS

set_retention_control RET1 -domain UA -save_signal {save high} -restore_signal {rst low }
```

In the snippet, the distinct save and restore signals are specified, but no map_retention_cell is specified. If none of the retention cells in the library have distinct save and restore signals, the following violation message is reported.

```
UPF retention strategy RET1 could not be applied: Distinct save and restore signals specified but no retention cell in Library has distinct save and restore pins
```

Example 2

Consider the following retention strategy provided in the UPF file:

```
set_retention ret1 domain VD1 elements {inst1/ret1} retention_power_net VDD retention_ground_net VSS
```

For the above example, the *UPF_lowpower19* rule reports the following violation because the *map_retention_cell* command is not present in the UPF for this retention strategy:

```
Command 'map_retention_cell' should be given for retention rule 'ret1' defined by set_retention, for using a specific retention cell
```

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPF_lowpower20

Checks for the supported set_port_attributes command options

When to Use

This is a setup rule and always runs by default.

Description

This rule does the sanity checks for `driver_supply`, `receiver_supply`, `model`, `pg_type`, `attribute` and `clamp_value` arguments of the `set_port_attributes` command and reports the violations accordingly.

Language

Verilog, VHDL

Parameter(s)

- *lp_check_driver_receiver_supply*: Default value is `no`. Set this parameter to `yes` to report a top level port with missing driver_supply/ receiver_supply options of the *set_port_attribute* command.

Constraint(s)

UPF Commands

- *set_port_attributes*

Messages and Suggested Fix

Message 1

The following message is reported when an argument (receiver_supply/ driver_supply) mentioned in *set_port_attributes* command is ignored for a set of top level ports:

[WARNING] Argument <argument-name> is used for top level <input|output> ports in set_port_attributes command. This argument is ignored for rule checking

The message reports an argument that is inapplicable to that port type (input/output). The violation is reported only when `-applies_to` is used

in the *set_port_attributes* command to mention the direction of top level port on which the argument is to be applied.

Message 2

The following message is reported when an expected argument (receiver_supply/driver_supply) is not provided in *set_port_attributes* command for specific port type (input/output):

[WARNING] Argument <argument-name> is missing for top level <input|output> ports in set_port_attributes command

The violation is reported only when `-applies_to` is used in the *set_port_attributes* command to mention the direction of top level port on which the argument is to be applied.

Message 3

The following message is reported when, an argument (receiver_supply/ driver_supply) mentioned in *set_port_attributes* command is ignored for a specific port based on its direction:

[WARNING] Argument <argument-name> is used for top level <input|output> port <port-name> in set_port_attributes command. This argument is ignored for this port

This message reports an argument that is inapplicable to the port type (input/output). The violation is reported only when `-ports` is used in the *set_port_attributes* command to mention one or more than one top level ports.

Message 4

The following message is reported when any string or hierarchical name other than a dot (.) is used in the –elements field in the *set_port_attributes* command:

[WARNING] Hierarchical name <hier-name> is used with -elements argument in set_port_attributes command. This argument is ignored for rule checking

Message 5

The following message is reported if driver_supply/receiver_supply arguments of the `set_port_attribute` command are missing for a top level port:

[WARNING] Missing set_port_attribute -<attribute-name> command

`for top level port <port-name>`

The rule reports this violation under the lp_check_driver_receiver_supply parameter.

The rule reports the violation depending on the type of top level port, as given below:

- For INPUT port, missing driver_supply option is reported.
- For OUTPUT port, missing receiver_supply option is reported.
- For INOUT port, depending on the usage of the port in the design, the missing driver/receiver is reported.
- If INOUT port is used only as INPUT port, the missing driver_supply is reported.
- If INOUT port is used only as OUTPUT port, the missing receiver_supply is reported.
- If INOUT port is used both as INPUT and OUTPUT port, one violation each is reported for the following:
  - Missing driver_supply option
  - Missing receiver_supply option

Message 6

The following message is reported when anything other than name of a library cell is used in the *model* argument of the *set_port_attributes* command:

[WARNING] `Library cell is not specified in -model option. The command is ignored for rule checking`

Message 7

The following message is reported when the *pg_type* argument is specified for non PG pins in the *set_port_attributes* command:

[WARNING] `Non supply pin(s) <non-PG-pin-list> specified with pg_type option in set_port_attributes command. Non supply pin(s) are ignored for rule checking`

Message 8

The following message is reported when the *related_power_port* or/and *related_ground_port* arguments are specified for PG pins in the

set_port_attributes command:

[WARNING] Supply pin(s) <PG-pin-list> specified with related_power_port/ related_ground_port option in set_port_attributes command. Supply pin(s) are ignored for rule checking

Message 9

The following message is reported when an unsupported argument is used along with the *model* argument in the *set_port_attributes* command:

[WARNING] Option 'model' is not supported with <argument-name> option. The command is ignored for rule checking

Arguments that are not supported together with the model argument are:

- driver_supply
- receiver_supply
- repeater_supply

The elements argument is also not supported with model argument, but for this, SpyGlass Power Verify reports *UPFSTX_31*.

Message 10

The following message is reported when any attribute other than the allowed attributes is used in the *attribute* argument of the *set_port_attributes* command:

[WARNING] Attribute <attribute-name> is not supported. The command is ignored for rule checking

The allowed attributes to be used in the attribute argument are:

- UPF_clamp_value
- UPF_driver_supply
- UPF_receiver_supply
- UPF_pg_type
- UPF_related_power_port
- UPF_related_ground_port

Message 11

The following message is reported when an unsupported value is used for *pg_type* or *clamp_value* argument in the *set_port_attributes* command:

[WARNING] Value <value> is not supported under attribute <argument-name>. The command is ignored for rule checking

Message 12

The following message is reported when the pg_type, unconnected, or feedthrough attribute is specified without the model attribute, since these attributes can be set only for the pins of library cells:

[WARNING] pg_type, unconneted, feedthrough options are only supported with -model option. The command is ignored for rule checking as -model is not specified

Message 13

The following message is reported when more than one input pin is specified under -ports with the -feedthough attribute, as a feedthrough connection can be made only between one input pin and one or more output pins:

[WARNING] More than one input pin is specified with -feedthrough option. The command is ignored for rule checking

Message 14

The following message is reported when only input pin(s) or only output pin(s) are specified under -ports, as both type of pins should be specified to make a feedthrough connection:

[WARNING] No <input | output> type of pins are specified with -feedthrough attribute. The command is ignored for rule checking as no feedthrough connection possible

Message 15

The following message is reported when -unconnected and/or -feedthrough is specified without -ports as ports need to be feedthrough or unconnected must be specified in the command itself:

[WARNING] <feedthrough | unconnected> is specified without port list. The command is ignored for rule checking

Message 16

The following message is reported when the -feedthrough attribute is specified with ports that are not internally connected as to make a feedthrough it is a must that the ports involve in feedthrough connection should be connected internally:

[WARNING] No Feedthrough connection possible between input pin <input-pin-name> and output pin <output-pin-name> as those are not directly connected in functional description of model <model-name>

Message 17

The following message is reported when the -feedthrough attribute is specified for output pin(s) that are internally connected to input pin(s) other than the input pin that is specified in the command:

[WARNING] No Feedthrough connection possible between input pin <input-pin-name> and output pin(s) <output-pin-names> as functional description of output pin(s) <output-pin-names> contains input pin(s) other than <input-pin-name>

Message 18

The following message is reported when the -unconnected attribute is specified for pins that are connected internally:

[WARNING] Unconnected attribute is written on pin(s) <pin-names> that are connected in functional description. Unconnected attribute will be ignored on those pin(s)

Potential Issues

Incorrect or unsupported options in the set_port_attribute UPF command may lead to power related issues in design.

Consequences of Not Fixing

Intended supply (driver/receiver) is not associated with the required port. Intended overwriting of pg_type, related_power_port, related_ground_port of library cell pins does not get applied. As a result isolation/level shifting rules might give false/missing violations.

How to Debug and Fix

For Message 1, Message 3, Message 6, Message 7, Message 8, Message 9,

Message 10, and Message 12, Message 13, Message 14, Message 15, the corresponding commands are not applicable and must be removed.

For Message 2, and Message 11, provide the required argument with valid value as pointed out in the message.

For Message 4, only top level ports are supported, hierarchical ports other than in top hierarchy must not be mentioned.

For Message 5, correct *set_port_attributes* command must be provided for the violating port, if required.

For Massage 16, Massage 17, and Massage 18, the command is not applicable on reported pins and reported pins must be removed from the command.

Example Code and/or Schematic

Example 1

Consider the following UPF snippet that illustrates some scenarios where the *UPF_lowpower20* rule reports violation messages:

```
create_supply_port top_supply -domain TOP -direction in
create_supply_set SS_TOP -function {power top_supply} -
function {ground VSS}
set_port_attributes {-elements {"."} -applies_to inputs} -
receiver_supply SS_TOP
```

For the above example, the *UPF_lowpower20* rule reports the following violations because the `-receiver_supply` argument is provided for top level input ports that is not required and the `–driver_supply` argument is missing:

```
Argument 'receiver_supply' is used for top level input ports in
set_port_attributes command. This argument is ignored for rule
checking

Argument 'driver_supply' is missing for top level input ports
in set_port_attributes command
```

Example 2

Consider the following UPF snippet where three top level ports IN1, IN2 and OUT1 are present and the *set_port_attributes* command is only provided for the top level input port IN1:

```
set_port_attributes -ports {IN1} -driver_supply SS_VD
```

For the above example, the *UPF_lowpower20* rule reports the following violations because the *set_port_attributes* command is not provided for the other two ports:

```
Missing set_port_attribute -driver_supply command for top level port 'IN2'

Missing set_port_attribute –receiver_supply command for top level port 'OUT1'
```

Example 3

In the following UPF commands, `Top` is a verilog module that has two input ports, in1, in2 and AND_2 is a library cell that has signal pins A, B, Y, and PG pins VDD, VDDC1, VSS.

```
set_port_attributes -model Top –ports {in1 in2} -
related_power_port VDD -related_ground_port VSS
```

This command leads to the following violation because `Test1` is a verilog module and only library cells are supported for the `model` argument:

```
Library cell is not specified in -model option. The command is ignored for rule checking
```

```
set_port_attributes -model AND_2 -ports {A B Y VDDC1} -
applies_to outputs -pg_type primary_power
```

This command leads the following violation because A,B, and Y are signal pins:

```
Non supply pins(s) ' A B Y ' specified with pg_type option in set_port_attributes command. Non supply pin(s) are ignored for rule checking
```

```
set_port_attributes -model AND_2 –ports {A B} -driver_supply ss1
```

This command leads the following violation message because to `driver_supply` and `model` arguments are not supported together:

```
Option '-model' is not supported with 'driver_supply' option.
```

```
The command is ignored for rule checking
```

```
set_port_attributes -model AND_2 –ports {A B Y VDDC1} -
attribute UPF_related_power_port VDD -related_ground_port
VSS
```

This command leads to the following violation message because VDDC1 is a PG pin:

```
Supply pins(s) 'VDDC1' specified with related_power_port/
related_ground_port option in set_port_attributes command.
Supply pin(s) are ignored for rule checking
```

```
set_port_attributes –model AND_2 –ports {A} –attribute
UPF_source_off_clamp_value 1
```

This command leads to the following violation because the UPF_source_off_clamp_value attribute is not supported for the attribute argument:

```
Attribute 'UPF_source_off_clamp_value' is not supported. The
command is ignored for rule checking
```

Example 4

In the following UPF commands, AND_1B is a library cell that has PG pin VDDC.

UPF Command:

```
set_port_attributes -model AND_1B -ports {VDDC} -pg_type xx
```

The above command leads to the following violation message because xx is not a valid value supported by the pg_type option of the set_port_attributes command:

```
Value 'xx' is not supported under attribute 'pg_type'. The
command is ignored for rule checking
```

UPF Command:

```
set_port_attributes -ports {inst1/out1} -clamp_value y
```

The above command leads to the following violation message because `y` is not a valid value supported by the `clamp_value` option of the `set_port_attributes` command:

```
Value 'y' is not supported under attribute 'clamp_value'. The command is ignored for rule checking
```

Example 5

Consider the following UPF command:

```
set_port_attributes -ports {A Y1 Y2 Y3} -feedthrough
```

The above command leads to the following violation message because the feedthrough argument is specified but the model argument is not specified, and feedthrough is only applicable for the pins of library cells:

```
pg_type, unconneted, feedthrough options are only supported with -model option. The command is ignored for rule checking as -model is not specified
```

Example 6

Consider the following library description for the cell CELL:

```
cell (CELL) {
        is_macro_cell : true;
      pg_pin (VDD) { pg_type : primary_power; }
      pg_pin (VSS) { pg_type : primary_ground; }
      pin (A) { direction : input; }
      pin (B) { direction : input; }
      pin (C) { direction : input; }
      pin (D) { direction : input; }
      pin (E) { direction : input; }
      pin (F) { direction : input; }

        pin (Y1) { direction : output; function : "A"; }
        pin (Y2) { direction : output; function : "B"; }
        pin (Y3) { direction : output; function : "C"; }
        pin (Y4) { direction : output; function : "D & A"; }
        pin (Y5) { direction : output; function : "E"; }
        pin (Y6) { direction : output; function : "F"; }
   }
```

Consider the following UPF command:

```
set_port_attributes -model CELL -ports {A B Y1 Y2 Y3} -
feedthrough
```

The following violation will be reported for the above `set_port_attributes` command due to the `-feedthrough` attribute is applicable only if one input type of pin is specified with one or more output pins in `-ports` option but here both 'A' and 'B' are input pins:

```
More than one input pin is specified with -feedthrough option.
The command is ignored for rule checking
```

Example 7

Consider the following library description for the cell CELL:

```
cell (CELL) {
        is_macro_cell : true;
      pg_pin (VDD) { pg_type : primary_power; }
      pg_pin (VSS) { pg_type : primary_ground; }
      pin (A) { direction : input; }
      pin (B) { direction : input; }
      pin (C) { direction : input; }
      pin (D) { direction : input; }
      pin (E) { direction : input; }
      pin (F) { direction : input; }

        pin (Y1) { direction : output; function : "A"; }
        pin (Y2) { direction : output; function : "B"; }
        pin (Y3) { direction : output; function : "C"; }
        pin (Y4) { direction : output; function : "D & A"; }
        pin (Y5) { direction : output; function : "E"; }
        pin (Y6) { direction : output; function : "F"; }
   }
```

Consider the following UPF command:

```
set_port_attributes -model CELL -ports {Y1 Y2 Y3} -
feedthrough
```

The following violation will be flagged for the above `set_port_attributes` command due to `-feedthrough` is applicable only if both one input and output pins are specified in -ports option but

here both no input pin is specified in command above:

```
No 'input' type of pins are specified with -feedthrough
attribute. The command is ignored for rule checking as no
feedthrough connection possible
```

Example 8

Consider the following library description for the cell CELL:

```
cell (CELL) {
        is_macro_cell : true;
      pg_pin (VDD) { pg_type : primary_power; }
      pg_pin (VSS) { pg_type : primary_ground; }
      pin (A) { direction : input; }
      pin (B) { direction : input; }
      pin (C) { direction : input; }
      pin (D) { direction : input; }
      pin (E) { direction : input; }
      pin (F) { direction : input; }

        pin (Y1) { direction : output; function : "A"; }
        pin (Y2) { direction : output; function : "B"; }
        pin (Y3) { direction : output; function : "C"; }
        pin (Y4) { direction : output; function : "D & A"; }
        pin (Y5) { direction : output; function : "E"; }
        pin (Y6) { direction : output; function : "F"; }
   }
```

Consider the following UPF command:

```
set_port_attributes -model CELL -feedthrough
```

The following violation will be flagged for the above `set_port_attributes` command due to -feedthrough is applicable only if the ports need to be feedthrough is specified but in command above -ports is missing:

```
'feedthrough' is specified without port list. The command is
ignored for rule checking
```

Example 9

Consider the following library description for the cell CELL:

```
cell (CELL) {
        is_macro_cell : true;
      pg_pin (VDD) { pg_type : primary_power; }
      pg_pin (VSS) { pg_type : primary_ground; }
      pin (A) { direction : input; }
      pin (B) { direction : input; }
      pin (C) { direction : input; }
      pin (D) { direction : input; }
      pin (E) { direction : input; }
      pin (F) { direction : input; }

        pin (Y1) { direction : output; function : "A"; }
        pin (Y2) { direction : output; function : "B"; }
        pin (Y3) { direction : output; function : "C"; }
        pin (Y4) { direction : output; function : "D & A"; }
        pin (Y5) { direction : output; function : "E"; }
        pin (Y6) { direction : output; function : "F"; }
   }
```

Consider the following UPF command:

```
set_port_attributes -model CELL -ports {A Y1 Y2} -feedthrough
```

The following violation will be flagged for the above `set_port_attributes` command due to -feedthrough is applicable only if the specified pins have an internal connection but pin 'A' and 'Y2' is not connected in library description above:

```
No Feedthrough connection possible between input pin 'A' and
output pin 'Y2' as those are not directly connected in
functional description of model 'CELL'
```

Example 10

Consider the following library description for the cell CELL:

```
cell (CELL) {
        is_macro_cell : true;
      pg_pin (VDD) { pg_type : primary_power; }
      pg_pin (VSS) { pg_type : primary_ground; }
      pin (A) { direction : input; }
      pin (B) { direction : input; }
      pin (C) { direction : input; }
```

```
      pin (D) { direction : input; }
      pin (E) { direction : input; }
      pin (F) { direction : input; }

        pin (Y1) { direction : output; function : "A"; }
        pin (Y2) { direction : output; function : "B"; }
        pin (Y3) { direction : output; function : "C"; }
        pin (Y4) { direction : output; function : "D & A"; }
        pin (Y5) { direction : output; function : "E"; }
        pin (Y6) { direction : output; function : "F"; }
  }
```

Consider the following UPF command:

```
set_port_attributes -model CELL -ports { a Y1 Y2 Y4} -
feedthrough
```

The following violation will be flagged for the above `set_port_attributes` command due to `-feedthrough` is applicable only if the specified output pins(s) have internal connection only with the input pin specified but pin 'Y4' has excess pin 'D' in its function:

```
No Feedthrough connection possible between input pin 'A' and
output pin(s) 'Y4 ' as functional description of output pin(s)
Y4 contains input pin(s) other than 'A'
```

Example 11

Consider the following library description for the cell CELL:

```
cell (CELL) {
        is_macro_cell : true;
      pg_pin (VDD) { pg_type : primary_power; }
      pg_pin (VSS) { pg_type : primary_ground; }
      pin (A) { direction : input; }
      pin (B) { direction : input; }
      pin (C) { direction : input; }
      pin (D) { direction : input; }
      pin (E) { direction : input; }
      pin (F) { direction : input; }

        pin (Y1) { direction : output; function : "A"; }
        pin (Y2) { direction : output; function : "B"; }
```

```
        pin (Y3) { direction : output; function : "C"; }
        pin (Y4) { direction : output; function : "D & A"; }
        pin (Y5) { direction : output; function : "E"; }
        pin (Y6) { direction : output; function : "F"; }
    }
```

Consider the following UPF command:

```
set_port_attributes -model CELL -ports { A Y1 Y2 Y4} -
feedthrough
```

The following violation will be flagged for above set_port_attributes command due to -unconnected is applicable only for pins that do not have internal connections but pin 'A', 'Y1', 'Y2', 'Y4' have internal connections according to library description above:

```
Unconnected attribute is written on pin(s) 'A Y1 Y2 Y4 ' that
are connected in functional description. Unconnected attribute
will be ignored on those pin(s)
```

Default Severity Label

Warning

Rule Group

UPF Check

Reports and Related Files

None

UPF_lowpower21

Checks if the source or destination supplies are not available in the domain where level shifter cell will be inferred

When to Use

This is a setup rule and always runs by default.

Description

This rule reports a violation message if source or destination supplies are not declared in the domain, in which level shifter cell will be inferred. Source and destination supplies should be either created or reused in the domain in which the strategy infers level shifter cell.

This rule does not perform checking for a level shifter strategy that has:

- –no_shift
- –location automatic
- –location fanin

This rule does not perform checking when source or destination supplies are operating at the same voltage (strategy is redundant).

Language

Verilog, VHDL

Parameter(s)

- *lp_skip_buf*: Default value is `1`. Set the value to 0 to consider buffers generated during design synthesis.
- *lp_skip_pwr_gnd*: Default value is `1`. Set the value to 0 to consider nets connected to power/ground supply.
- *lp_flag_undriven_nets*: Default value is `0`. Set the parameter to 1 to consider nets that are not driven at voltage crossings.
- *lp_flag_unconnected_nets*: Default value is `0`. Set the parameter to 1 to consider unconnected nets at voltage crossings.
- *lp_skip_blackbox_checking*: Default value is `0`. Set the parameter to `1` to skip checking for black boxes.

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_level_shifter* (Mandatory)
- *map_level_shifter_cell* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when source or destination supplies are not available in the domain where level shifter cell will be inferred:

[ERROR] Level shifter strategy ‘<strategy-name>’ infers level shifter cell in a domain ‘<domain-name>’ between source ‘<source-hier-path>’ (<source-domain>:<source-supply>) and destination ‘<destination-hier-path>’ (<destination-domain>:<destination-supply>). [Source | Destination] supply is not available in the domain

NOTE: *If the source or destination supply is coming through a power switch (Gated Supply), then gated supply of power switch will be considered available in domain in which level shifter cell will be inferred only if new PST merge flow is used, using the lp_pst_merge_new parameter.*

Message 2

The following violation message is reported when source supply does not match the `input_supply_set` and/or destination supply does not match `output_supply_set` for a given level shifter strategy:

[ERROR] Level shifter strategy ‘<strategy-name>’ infers level shifter cell between source ‘<source-node>’ (<source-domain-

name>) and destination ‘<destination-node>’ (<destination-domain-name>). <input_supply_set | output_supply_set> ‘<supply_set_name>’ is not same as <source | destination> supply

Potential Issues

Consider the following scenario:

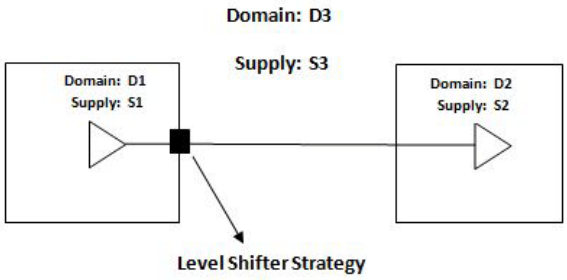


FIGURE 214. Level shifter strategy

A level shifter strategy is written at the output of the D1 domain. Since location is self, a level shifter cell is inserted inside the D1 domain. For this level shifter cell to perform voltage shifting correctly, both source and destination supplies should be available in the D1 domain. Since source supply is S1 and primary supply of the D1 domain is also the same (S1), hence source supply is available in the D1 domain. But if the destination supply is not available in the D1 domain, then an implementation tool is not able to route power supplies and hence level shifting is not performed.

Consequences of Not Fixing

An implementation tool is not able to route power supplies and hence level shifting is not performed.

How to Debug and Fix

The *set_level_shifter* UPF command is highlighted in the Atrenta Console GUI. To fix this violation, perform the following:

Declare the source or destination supply in a domain that will infer level shifter cell, as stated in the message.

Example Code and/or Schematic

Example 1

Consider the following UPF1.0 code snippet:

```
create_supply_net VDD10 –domain PD1
create_supply_net VDD21 –domain PD2
set_level_shifter ls2 -domain PD1 -location self -applies_to
outputs -rule low_to_high
```

For the above UPF snippet, the *UPF_lowpower21* rule reports the following violation because the destination supply VDD21 is not available in the domain PD1 that will infer level shifter cell:

```
Level shifter strategy 'ls2' infers level shifter cell in a
domain 'PD1' between source 'top.inst1.mac1.Y' (PD1(supply
VDD10:0.850)) and destination 'top.inst2.mac1.A' (PD2(supply
VDD21:1.050)). Destination supply is not available in the
domain
```

To resolve the above violation, create the destination supply VDD21 in the PD1 domain using the following command:

```
create_supply_net VDD21 –domain PD1 –reuse
```

The following is the schematic for this example:

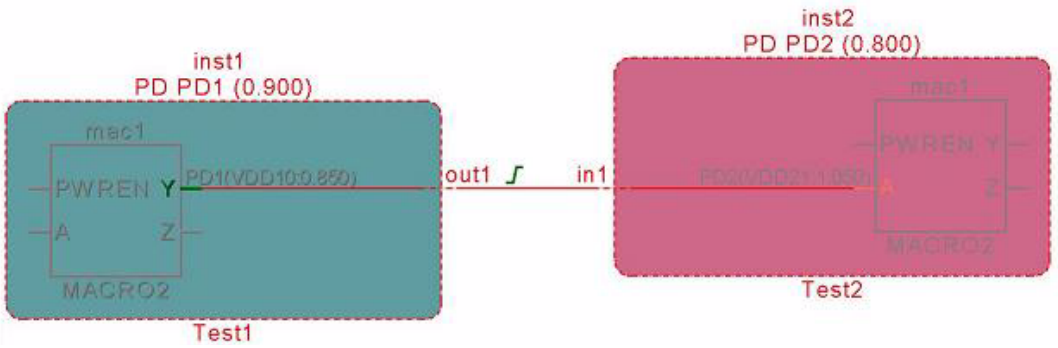


FIGURE 215. Incremental schematic

Example 2

Consider the following UPF snippet:

```
create_supply_set ss1 -function {power VDD1} -function
{ground VSS} -update
create_supply_set ss2 -function {power VDD2} -function
{ground VSS} –update
set_level_shifter ls1 -domain PD1 -input_supply_set ss2 -
output_supply_set ss1
```

The following violation message is reported because the input supply set `ss2` does not match the source supply:

```
Level shifter strategy 'ls1' infers level shifter cell between
source 'top.inst1.and1.Y' (PD1(supply VDD1:0.900)) and
destination 'top.inst2.and1.A' (PD2(supply VDD2:0.800)).
input_supply_set 'ss2' is not same as source supply
```

The following is the schematic for this example:

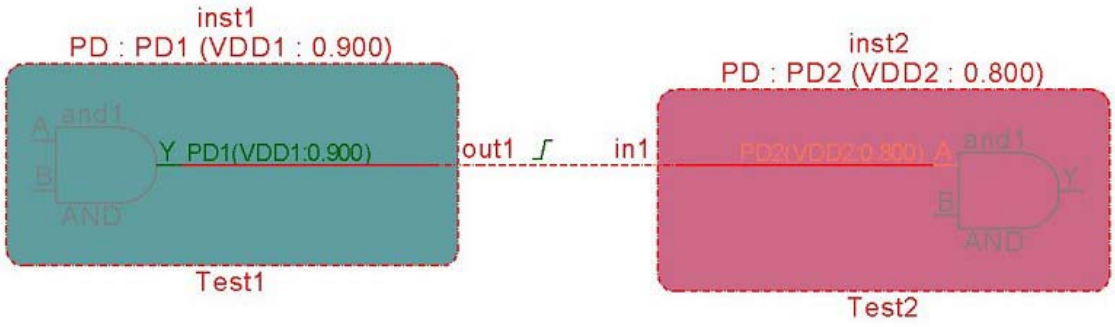


FIGURE 216. Incremental schematic

Example 3

Consider the following UPF snippet:

```
set_scope pd1
create_power_domain VA -include_scope
create_supply_port VDD
create_supply_net  VDD -domain VA
connect_supply_net VDD -ports VDD
```

```
set_domain_supply_net VA -primary_power_net VDD
primary_ground_net VSS
set_level_shifter ls -domain VA  -rule both  -location self
-applies_to outputs

set_scope /pd2
create_power_domain VB -include_scope
create_supply_port VDD
create_supply_net  VDD -domain VB
create_supply_net  VDDG -domain VB
connect_supply_net VDD -ports VDD
set_domain_supply_net VB -primary_power_net VDDG
primary_ground_net VSS
create_power_switch PS_rule1 -input_supply_port {VDD VDD}
output_supply_port {VDD_gt VDDG} -control_port {SLEEP /pd2
ENA} -domain VB
```

Consider a crossing in which a level shifter cell will be inferred in domain VA.
Destination supply (VDDG, domain VB) will be available in domain VA only if the new PST merge flow is used, using the *lp_pst_merge_new* parameter. Hence, the UPF_lowpower21 rule will not report non-availability of destination supply.

The following is the schematic for this example:

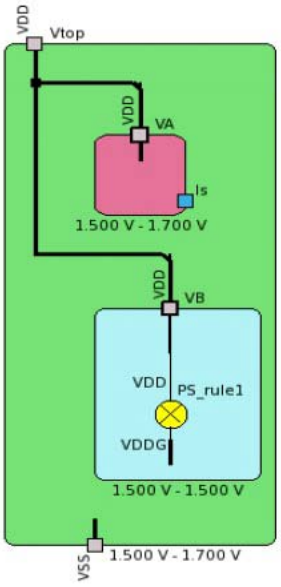


FIGURE 217. Incremental schematic

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPF_lowpower22

Reports if the supply of a repeater is not matching the supply specified in the UPF command

When to Use

Use this rule for:

- *RTL description files with or without library files*
- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*
- *DEF files and their associated LEF files*

Description

This rule reports a message in the following scenarios:

- RTL stage:
  - If there is a conflict between the driver supply and the repeater supply in the *set_port_attribute* command on a port in the UPF.
  - If the supply specified for a repeater in UPF is not available in repeater location domain.
- Netlist/PG Netlist stage:
  - If the supply of a repeater (through connect_supply_net or Standard Cell Main Rail) does not match the repeater supply specified in the UPF. User can set the repeater supply through:
    - `set_port_attribute -repeater_supply`
    - `set_repeater -repeater_supply_set`
  - If repeater is not found corresponding to repeater attribute specified in the UPF.
  - If repeater name does not follow the name prefix/suffix specified in the `set_repeater -name_prefix/-name_suffix` command in the UPF

Language

Verilog, VHDL

Parameter(s)

lp_set_design_stage: Default value is `rtl`. Set this parameter to `netlist` or `pg_netlist` to specify the desired design stage.

Constraint(s)

UPF Commands

- *set_port_attributes* (Mandatory)
- *set_repeater* (Mandatory)

Messages and Suggested Fix

Message 1

The following message is reported when the driver supply *<driver-supply>* does not match the repeater supply *<repeater-supply>* specified on the same port in the UPF:

[ERROR] Driver supply ‘<driver-supply>’ is not consistent with repeater supply ‘<repeater-supply>’ for port ‘<port-name>’ in UPF

Message 2

The following message is reported when the supply of a repeater does not match the repeater supply *<repeater-supply-name>* specified on the port, in the UPF:

[ERROR] Repeater supply mismatch. For supply pin '<supply-pin-name>' of repeater '<repeater-name>', supply '<supply-name >' specified is different from repeater supply '<repeater-supply-name>' specified on port '<port-name>’ in UPF

Message 3

The following message is reported when the repeater supply *<repeater-supply-name>* specified using the *set_port_attributes* command for port *<port-name>* is not available at the location of the repeater cell:

[ERROR] Repeater supply '<repeater-supply-name>' specified on port '<port-name>' is not 'available' in repeater location domain '<domain-name>’

Message 4

The following message is reported when the supply of a SpyGlass generated buffer acting as a repeater does not match the repeater supply *<repeater-supply-name>* specified on the port, in the UPF. This message is reported only when the lp_skip_buf parameter is set to 0.

[ERROR] Repeater supply mismatch. For repeater '<repeater-name>', supply '<supply>' is different from repeater supply '<supply-name>' specified on port '<port-name>' in UPF

Message 5

The following message is reported when the repeater is not found corresponding to the repeater attribute specified on port *<port-name>* in the UPF:

[ERROR] Repeater not found corresponding to repeater attribute specified on port '<port-name>' in UPF

Message 6

The following message is reported when the repeater name *<repeater-name>* does not match the prefix/suffix convention specified in the corresponding strategy of *<strategy-name>* on port *<port-name>* in the UPF:

[WARNING] Repeater name '<repeater-name>' does not match the name <prefix | suffix> specified in corresponding strategy '<strategy-name>' specified on port '<port-name>' in UPF

Potential Issues

The violation messages explicitly states the potential issues.

Consequences of Not Fixing

The design may fail if the repeater cell is not connected to the expected supply as specified in the *set_port_attribute* command, in the UPF.

Message 6 only shows that the naming prefix/suffix convention has not been followed and does not cause the design to fail.

How to Debug and Fix

To fix this violation, perform the following:

- **Message 1:** Make sure that driver supply and repeater supply for violating port must be the same.

- **Message 2 and 4:** Make sure that the supply of a repeater cell matches the repeater supply specified for the port, in the UPF.
- **Message 3:** Make sure the supply specified in the repeater supply is available in the repeater location.
- **Message 5:** Make sure repeater cell is present in design corresponding to repeater attribute provided for port, in the UPF.
- **Message 6:** Make sure repeater name corresponds to the prefix/suffix convention specified in the corresponding strategy of the specified port, in the UPF.

Example Code and/or Schematic

Example 1

Consider the following UPF code snippet:

```
connect_supply_net VDD2 -ports {bf1/VDD}
create_supply_set ss1 -function {power VTOP} -function
{ground VSS} -function {nwell VTOP} -function {pwell VSS}
create_supply_set ss2 -function {power VDD1} -function
{ground VSS} -function {nwell VDD1} -function {pwell VSS}

set_port_attributes -ports {inst1/in1} -driver_supply ss2 -
repeater_supply ss1
```

For the above UPF snippet, the *UPF_lowpower22* rule reports the following violations because the driver supply VDD1 is not consistent with repeater supply VTOP and for the supply pin 'VDD' of repeater 'top.bf1', supply 'VDD2' is different from repeater supply 'VTOP' specified on port 'top.inst1.in2':

```
Driver supply 'VDD1' is not consistent with repeater supply
'VTOP' for port 'top.inst1.in1' in UPF


Repeater supply mismatch. For supply pin 'VDD' of repeater
'top.bf1', supply 'VDD2' is different from repeater supply
'VTOP' specified on port 'top.inst1.in1' in UPF
```

The following is the schematic for this example:

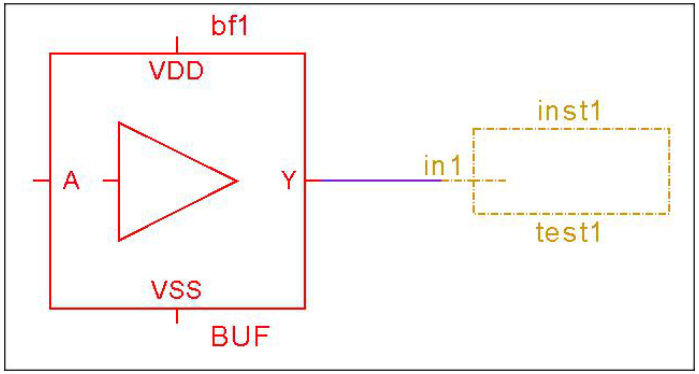


FIGURE 218. Incremental schematic

Example 2

Consider the following UPF code snippet:

```
create_supply_set ss1 -function {power VTOP} –function
{ground VSS}

create_supply_set ss2 -function {power VDD1} –function
{ground VSS}
create_supply_set ss3 -function {power VDD2} –function
{ground VSS}
create_power_domain TOP –include_scope -supply {primary ss1 }
create_power_domain PD1 –elements {inst1} -supply {primary
ss2 }
set_port_attributes -ports {inst1/out} -repeater_supply ss1
```

For the above UPF snippet, the *UPF_lowpower22* rule reports the following violations because the repeater supply `VTOP` is not available in the repeater domain `PD1`. Also, no repeater corresponding to repeater strategy specified for `top.inst1.out` is present in the design.

```
Repeater supply 'VTOP' specified on port 'top.inst1.out' in UPF
is not 'available' in repeater location domain 'PD1'


Repeater not found corresponding to repeater attribute
```

```
specified on port 'top.inst1.out' in UPF
```

The following is the schematic for this example:

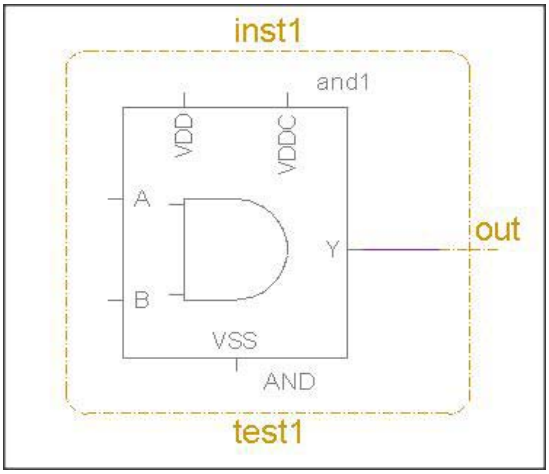


FIGURE 219. Incremental schematic

Example 3

Consider the following UPF code snippet:

```
upf_version 2.0
create_power_domain TOP -include_scope
create_supply_port VDD
create_supply_net VDD -domain TOP
connect_supply_net VDD -ports {VDD}
create_supply_port VSS
create_supply_net VSS -domain TOP
connect_supply_net VSS -ports {VSS}
add_port_state VDD -state {active 1.0} -state {Off off}
add_port_state VSS -state {sleep 0}

create_power_domain PD1 -elements {/sub1}
create_supply_port VDD1 -domain PD1
```

```
create_supply_net VDD1 -domain PD1
connect_supply_net VDD1 -ports {VDD1}
create_supply_net VSS -domain PD1 -reuse

create_supply_port VDD2 -domain PD1
create_supply_net VDD2 -domain PD1
connect_supply_net VDD2 -ports {VDD2}

add_port_state VDD1 -state {active 1.0}
add_port_state VDD2 -state {active 1.0}


set_domain_supply_net TOP -primary_power_net VDD
primary_ground_net VSS
set_domain_supply_net PD1 -primary_power_net VDD1
primary_ground_net VSS

set_isolation iso -domain PD1 -elements {/sub1/in}
applies_to inputs -isolation_power_net VDD1
isolation_ground_net VSS
set_isolation_control iso -domain PD1  -isolation_signal
en_iso -location self

create_supply_set ss1 -function {power VDD1} -function
{ground VSS}
create_supply_set ss2 -function {power VDD2} -function
{ground VSS}
#set_port_attributes {-elements {/sub1} -applies_to inputs}
-repeater_supply ss1
set_repeater R1 -elements /in -domain TOP
repeater_supply_set ss1 -name_prefix buf -name_suffix 2
#set_repeater R1 -elements /sub1/in -domain PD1
repeater_supply_set ss1
#set_port_attributes -ports {/sub1/in} -repeater_supply ss1
connect_supply_net VDD2 -pins {/buf1/vdd}
```

For the above UPF snippet, the *UPF_lowpower22* rule reports the following violations because the repeater supply `top.buf1` does not match the suffix `suffix('2')` specified in corresponding strategy `R1` of on port

`in`, in the UPF.

```
Repeater name 'top.buf1' does not match the name suffix('2')
specified in corresponding strategy 'R1' specified on port 'in'
in UPF
```

The following is the schematic for this example:

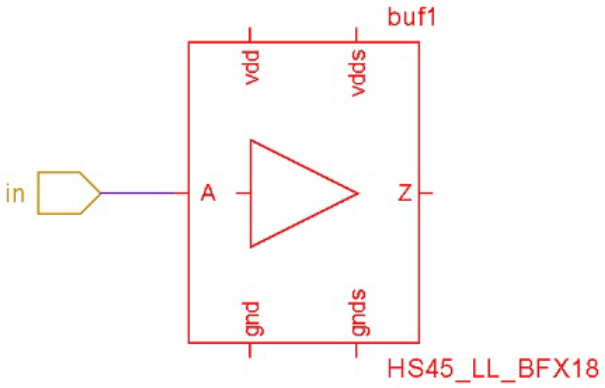


FIGURE 220. Incremental schematic

Default Severity Label

Error, Warning

Rule Group

UPF Check

Reports and Related Files

None

UPF_lowpower23

Checks the set_equivalent UPF command

When to Use

This is a setup rule and always runs by default.

Description

The *UPF_lowpower23* rule checks the validity of the *set_equivalent* command.

This rule reports violations in the following scenarios:

- When the supply nets/ports given for the `set_equivalent` command with the `-function_only` option have different power states in PST.
- When the supply nets/ports are given for the `set_equivalent` command without the `-function_only` option that are not connected.
- When the supply sets given for the `set_equivalent` command with the `-function_only` option have power/ground nets with different power states in PST.
- When the supply sets given for the `set_equivalent` command without the `-function_only` option have power/ground nets that are not connected.
- When the supply sets given for the `set_equivalent` command don not have the same functions.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)

- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_equivalent* (Mandatory)
- *create_supply_set* (Optional)
- *create_pst* (Optional)
- *add_pst_state* (Optional)

Messages and Suggested Fix

Message 1

The following message appears when the supply nets/ports given for the *set_equivalent* command with the `-function_only` option have different power states in PST:

[ERROR] Supply net/port '<net1-name | port1-name>' and supply net/port '<net2-name | port2-name>' are declared as equivalent, but they have different power states

Message 2

The following message appears when the supply nets/ports are given for the *set_equivalent* command without the `-function_only` option that are not connected:

[ERROR] Supply net/port '<net1-name | port1-name>' and supply net/port '<net-name2 | port2-name>' are declared as electrically equivalent, but they are not connected

Message 3

The following message appears when the supply sets given for the *set_equivalent* command with the `-function_only` option have power/ground nets with different power states in PST:

[ERROR] Supply sets '<supply-set1-name>' and '<supply-set2-name>' are declared as equivalent, but they have power/ground/nwell/pwell nets with different power states

Message 4

The following message appears when the supply sets given for the *set_equivalent* command without the `-function_only` option have power/ground nets that are not connected.

```
[ERROR] Supply sets '<supply-set1-name>' and '<supply-set2-name>' are declared as electrically equivalent, but their power/ground  nets are not connected
```

Message 5

The following message appears when the supply sets given for the `set_equivalent` command do not have the same functions:

```
[ERROR] Supply sets '<supply-set1-name>' and '<supply-set2-name>' are declared as equivalent, but they don't have the same set of functions
```

Message 6

The following violation message appears when input and output elements of a power switch are declared as electrically equivalent:

```
[ERROR] Input supply net/port '<net1-name | port1-name>' and output supply net/port '<net2-name | port2-name>' of power switch '<power-switch-name>' are declared as electrically equivalent
```

Potential Issues

The violation messages explicitly states the potential issues.

Consequences of Not Fixing

SpyGlass requires you to resolve these violations.

How to Debug and Fix

The set_equivalent UPF command is highlighted in the Atrenta Console GUI. To fix these violations, perform the following:

- **Message 1**: Make sure that supply nets/ports given for the *set_equivalent* command with the `-function_only` option have the same power states in PST.
- **Message 2**: Make sure that the supply nets/ports are given for the `set_equivalent` command without the `-function_only` option that are connected.

- **Message 3**: Make sure that the supply sets given for the `set_equivalent` command with the `-function_only` option have power/ground nets with the same power states in PST.
- **Message 4**: Make sure that the supply sets given for the `set_equivalent` command without the `-function_only` option have power/ground nets that are connected.
- **Message 5**: Make sure that supply sets given to the *set_equivalent* command have the same set of functions.
- **Message 6**: Make sure that the input and output elements of a power switch are not declared as equivalent.

Example Code and/or Schematic

Example 1

Consider the following set of UPF commands:

```
create_supply_port VDD
create_supply_net VDD
connect_supply_net VDD –ports {VDD}

create_supply_port VDD1
create_supply_net VDD1
connect_supply_net VDD1 –ports {VDD1}

create_supply_port VSS
create_supply_net VSS
connect_supply_net VSS –ports {VSS}

create_supply_set ss0 –function {power VDD} –function {ground
VSS}
create_supply_set ss1 –function {power VDD1} –function
{ground VSS}

set_equivalent –nets {VDD VDD1}
set_equivalent –sets {ss0 ss1}
```

For the above example, violations are reported for the both *set_equivalent* commands, because VDD and VDD1 are not connected.
To resolve this, make VDD and VDD1 driven by the same port, as given

below:

```
connect_supply_net VDD –ports {VDD}
connect_supply_net VDD1 –ports {VDD}
```

Example 2

Consider the following set of UPF commands:

```
add_port_state VDD –state {a 1.2} –state {b off}
add_port_state VDD1 –state {a 1.2} –state {b off}
add_port_state VSS –state {def 0.0}

create_pst pst1 –supplies {VDD VDD1 VSS}
add_pst_state s1 –pst pst1 –state { a a def}
add_pst_state s2 –pst pst1 –state { a b def}

set_equivalent -function_only –nets {VDD VDD1}
set_equivalent -function_only –sets {ss0 ss1}
```

In the above example, both of the *set_equivalent* commands trigger violations because VDD and VDD1 have different behaviors in PST.
To resolve this, make sure that VDD and VDD1 are similar in PST, as given below:

```
add_pst_state s1 –pst pst1 –state { a b def}
add_pst_state s2 –pst pst1 –state { a b def}
```

Example 3

Consider the following set of UPF commands:

```
create_supply_set ss3 -function {power VDD} -function {ground
VSS}
create_supply_set ss4 -function {power VDD1}
set_equivalent -sets {ss3 ss4}
```

In the above example, the *UPF_lowpower23* rule reports the following violation message because supply sets `ss3` and `ss4` do not have the same set of functions:

```
Supply sets 'ss3' and 'ss4' are declared as equivalent, but
they don't have the same set of functions
```

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPF_lowpower24

Checks the arguments of the set_design_attributes UPF command

When to Use

This is a setup rule and always runs by default.

Description

This rule does the sanity checks for `models`, `is_macro_cell`, `is_leaf_cell`, and `attribute` arguments of the `set_design_attributes` UPF command and reports the violations accordingly.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)
- *set_domain_supply_net* (Mandatory)
- *set_design_attributes* (Mandatory)

Messages and Suggested Fix

Message 1

The following message is reported when any attribute other than `UPF_is_macro_cell` or `UPF_is_leaf_cell` is used for the `–attribute` argument in the *set_design_attributes* command:

[WARNING] Attribute <attribute-name> is not supported and it is ignored for rule checking

Message 2

The following message is reported when -is_macro_cell or -is_leaf_cell or –attribute arguments are not specified in the *set_design_attributes* command:

[WARNING] No attribute is specified. The command is ignored for rule checking

Message 3

The following message is reported when the –models argument is not used and -is_macro_cell argument is used in the *set_design_attributes* command:

[WARNING] Option '-models' is not specified. The command is ignored for rule checking

Message 4

The following message is reported when anything other than the name of a library cell is used for the -models argument in the *set_design_attributes* command:

[WARNING] <cell-name> specified under -models is not a Library cell and it is ignored for rule checking since only Library cells are supported under -models option

Message 5

The following message is reported when the -elements argument is specified and -is_macro_cell argument is used but -is_leaf_cell argument is not used in the *set_design_attributes* command:

[WARNING] Option '-elements' is not supported with attribute is_macro_cell. Option -elements is ignored for rule checking

Message 6

The following message is reported when value of an attribute-value pair given in the -attribute argument is not valid for the given attribute:

[WARNING] Value '<value>' is not supported under attribute <attribute-name>. Attribute is ignored for rule checking

Potential Issues

The `-elements` argument is not supported with currently supported attribute `is_macro_cell`.

Consequences of Not Fixing

Intended overwriting of the `is_macro_cell` attribute of library cells will not be applied. As a result rules that interpret lib cells as macro cell will be impacted.

How to Debug and Fix

For **Message 1** and **Massage 5**, corresponding options are not applicable and must be removed. For **Message 2**, **Message 3** and **Message 4**, the corresponding commands are not applicable and must be removed. For **Massage 6**, the value specified for the given attribute is not applicable and a valid value must be given for the attribute.

Example Code and/or Schematic

In the following examples, `Test1` is a Verilog module and `AND_1` and `ISO1` are library cells.

Example 1

Consider the following UPF command:

```
set_design_attributes -models AND_1 -attribute
lower_domain_boundary  FALSE
```

For the above command, the *UPf_lowpower24* rule reports the following violation because `lower_domain_boundary` is not supported in the `-attribute` argument:

```
Attribute 'lower_domain_boundary' is not supported and it is
ignored for rule checking
```

Example 2

Consider the following UPF command:

```
set_design_attributes -models AND_1
```

For the above command, the *UPf_lowpower24* rule reports the following violation because neither `–is_macro_cell` nor `–attribute` arguments are specified in the command:

```
No attribute is specified. The command is ignored for rule checking
```

Example 3

Consider the following UPF command:

```
set_design_attributes -elements {.} -is_macro_cell TRUE
```

For the above command, the *UPf_lowpower24* rule reports the following violation because the `–models` argument is not specified:

```
Option '-models' is not specified. The command is ignored for rule checking
```

Example 4

Consider the following UPF command:

```
set_design_attributes -models Test1 -is_macro_cell TRUE
```

For the above command, the *UPf_lowpower24* rule reports the following violation because `Test1` is a Verilog module and only library cells are supported in the `–models` argument:

```
'Test1' specified under -models is not a Library cell and it is ignored for rule checking since only Library cells are supported under -models option
```

Example 5

Consider the following UPF command:

```
set_design_attributes -elements {.} -is_macro_cell TRUE
```

For the above command, the *UPf_lowpower24* rule reports the following

violation because the `-elements` and `-is_macro_cell` arguments are not supported together:

```
Option '-elements' is not supported with attribute
is_macro_cell. Option -elements is ignored for rule checking
```

Example 6

Consider the following UPF command:

```
set_design_attributes -models {ISO1} -attribute
{UPF_is_macro_cell xx}
```

For the above command, the *UPf_lowpower24* rule reports the following violation because the value `xx` is not a valid value for the `UPF_is_macro_cell` attribute:

```
Value 'xx' is not supported under attribute
'UPF_is_macro_cell'. Attribute is ignored for rule checking
```

Default Severity Label

Warning

Rule Group

UPF Check

Reports and Related Files

None

UPF_lowpower25

Checks mismatch between voltage _map and connect_supply_net

When to Use

Use this rule for:

- *Gate-level netlist files and their associated library files*
- *Post-Layout Netlist (Verilog) files and their associated physical libraries (PLIB/LEF) and gate libraries (LIB)*

Description

The *UPF_lowpower25* rule reports a violation when the pin voltage in the `voltage_map` attribute does not match with supply value connected to that pin from the `connect_supply_net` command.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_power_domain* (Mandatory)
- *create_supply_port* (Mandatory)
- *add_port_state* (Mandatory)
- *create_supply_net* (Mandatory)
- *connect_supply_net* (Mandatory)

Messages and Suggested Fix

Message 1

The following message is reported when there is inconsistency between the value of the `voltage_map` attribute provided from the library file for a pin and the supply value connected to it from *connect_supply_net* command

(only for the port states of port connected to net that are used in PST) in the UPF file:

[WARNING] Inconsistency between voltage_map value, '<voltage-map-value>' provided for pin '<pin-name>' of cell '<cell-name>' from library file and voltage, '<supply-value-connected-from connect_supply_net>' provided from connect_supply_net command in upf

Message 2

The following Info massage is reported when the *connect_supply_net* command is written for a leaf level pin, which has the voltage_map attribute set on it, and the port connected to this supply net has extra states that are not used in port state table, which do not match with the voltage_map value provided for the pin:

[INFO] Supply net connected to pin '<pin-name>' of cell '<cell-name>' using connect_supply_net has extra state(s) 'supply-port-states' which are not matching with voltage_map value, '<voltage-map-value>' provided for pin from library

Potential Issues

The violation messages explicitly state the potential issue.

Consequences of Not Fixing

Since the voltage_map value of the pin and supply connected from the UPF are different, this can lead to unsteady states in the design, which can result in design failure.

How to Debug and Fix

Review and update the UPF file for the pins mentioned in the reported violation massage.

Example Code and/or Schematic

Example 1

Consider the following example.

UPF:

```
create_supply_port VDDn
create_supply_net  VDDn
connect_supply_net VDDn -ports {VDDn}
```

```
connect_supply_net VDDn -ports {inst1/and1/VDD}
add_port_state VDDn -state {on1 1.2}
```

Library:

```
voltage_map(state1,1.1)
cell (AND) {
      pg_pin (VDD) { pg_type : primary_power; voltage_name :
state1;}
      pg_pin (VDDC) { pg_type : backup_power; voltage_name :
state4;}
     pg_pin (VSS) { pg_type : primary_ground; voltage_name :
state5;}
      pin (A) { direction : input;related_power_pin: VDD;
related_ground_pin: VSS; }
      pin (B) { direction : input;related_power_pin: VDD;
related_ground_pin: VSS; }
      pin (Y) { direction : output; function :
"A&B";related_power_pin: VDD;related_ground_pin: VSS;  }
   }
```

For the above example, the UPF_lowpower25 rule reports the following violation because the supply connected to pin `inst1/and1/VDD` from UPF and `voltage_map` value of it is different:

```
Inconsistency between voltage_map value, '1.100000' provided
for pin 'VDD' of cell 'top.inst1.and1' from library file and
voltage, '1.200000' provided from connect_supply_net command in
upf
```

Example 2

Consider the following example.

UPF:

```
create_supply_port VDDn
create_supply_net VDDn
connect_supply_net VDDn -ports {VDDn}
connect_supply_net VDDn -ports {inst1/and1/VDD}
add_port_state VDDn -state {on1 1.1} -state {on2 1.3} -state
{off1 off}
create_pst ps1 -supplies {VDDn VDD1n VDD2n VSSn}
```

```
add_pst_state s1 -pst ps1 -state {on1 on1 on1 def }
add_pst_state s3 -pst ps1 -state {off1 on1 on1 def }
```

Library:

```
voltage_map(state1,1.1)
cell (AND) {
pg_pin (VDD) { pg_type : primary_power; voltage_name :
state1;}
pg_pin (VDDC) { pg_type : backup_power; voltage_name :
state4;}
pg_pin (VSS) { pg_type : primary_ground; voltage_name :
state5;}
pin (A) { direction : input;related_power_pin: VDD;
related_ground_pin: VSS; }
pin (B) { direction : input;related_power_pin: VDD;
related_ground_pin: VSS; }
pin (Y) { direction : output; function :
"A&B";related_power_pin: VDD;related_ground_pin: VSS; }
}
```

For the above example, the *UPF_lowpower25* rule reports the following violation because the supply connected to pin `inst1/and1/VDD` from UPF and `voltage_map` value of it is different for state `on2` of supply `VDDn` and this state is not used in PST:

```
Supply net connected to pin 'VDD' of cell 'top.inst1.and1'
using connect_supply_net has extra state(s) '1.3' which are not
matching with voltage_map value, '1.100000' provided for pin
from library
```

Default Severity Label

Warning, Info

Rule Group

UPF Check Rules

Reports and Related Files

None

UPF_lowpower26

Checks inconsistencies between the given liberty file and the corresponding UPF file

When to Use

Use this rule to check inconsistencies between cell description in liberty file and UPF file.

Description

The *UPF_lowpower26* rule checks the inconsistencies present between the given liberty file and the corresponding UPF file and reports violations accordingly. This rule reports in following cases:

- When there is no port declared in the UPF file corresponding to any PG pin in the liberty file.
- When there is no PG pin(s) declared in the liberty file corresponding to supply port in the UPF file considering internal PG pins.
- When there is direction mismatch between the PG pins and corresponding UPF supply ports.
- When related supply defined through the `set_related_supply_net/ set_port_attributes` command or domain primary supplies in the UPF, does not match the related supply defined in the liberty file.
- When the voltage value for the PG pin does not match with any of the states defined in the UPF file.
- When the signal pin does not have an `is_isolated` attribute in liberty file but the UPF file infers the isolation rule defined for corresponding pin in the UPF file.
- When the signal pin does not have isolation rule in UPF file but the `is_isolated` attribute is defined in liberty file.
- When the signal pin has `no_isolation` strategy in UPF file but `is_isolated` attribute is defined in the liberty file.
- When there is no voltage map defined corresponding to PG pin in the liberty file.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

SGDC

set_lib_name: Used to specify the cell name defined in the corresponding liberty file.

UPF Commands

- *create_supply_port*
- *create_supply_net*
- *connect_supply_net*
- *add_port_state*
- *create_supply_set*
- *create_power_domain*
- *create_power_switch*
- *set_port_attributes*
- *set_isolation*
- *set_related_supply_net*
- *set_domain_supply_net*

Messages and Suggested Fix

Message 1

The following message is reported when the PG pins defined in the library do not have a corresponding supply port declared in the UPF file:

[WARNING] PG Pin(s) '<pin-name-list>' defined in liberty does not have a corresponding supply port declaration in the UPF file

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

Power verification may be incorrect.

How to Debug and Fix

Make sure that the PG pins defined in the library do have a corresponding supply port declared in the UPF file.

Message 2

The following message is reported when a supply port does not have a corresponding PG pin declared in the liberty file:

[WARNING] Supply port(s) '<port-name-list>' does not have a corresponding pg_pin declaration in the liberty file

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

Power verification may be incorrect.

How to Debug and Fix

Make sure that the supply ports have corresponding PG pins declared in the liberty file.

Message 3

The following message is reported when the direction defined for PG pin in liberty does not match with the direction defined for corresponding supply port in the UPF file:

[WARNING] Direction ‘<direction1-name>’ defined for pg_pin ‘<pin-name>’ in liberty does not match with direction ‘<direction2-name>’ defined for corresponding supply port in the UPF file

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

Power verification may be incorrect.

How to Debug and Fix

Make sure that the direction defined for PG pin in liberty matches with the

direction defined for corresponding supply port in the UPF file.

Message 4

The following message is reported when the related supply power pin defined for signal pin in the liberty file does not match with related supply defined for the corresponding pin in the UPF file:

[WARNING] Related supply power pin of ‘<pin-name>’ defined for signal pin ‘<signal-pin-name>’ in liberty does not match with related supply ‘<supply-name>’ defined for corresponding pin in the UPF file

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

Power verification may be incorrect.

How to Debug and Fix

Make sure that the related supply power pin defined for signal pin in the liberty file matches with related supply defined for the corresponding pin in the UPF file.

Message 5

The following message is reported when the related supply ground pin defined for signal pin in the liberty file does not match with related supply defined for the corresponding pin in the UPF file:

[WARNING] Related supply ground pin of ‘<pin-name>’ defined for signal pin ‘<pin-name>’ in liberty does not match with related supply ‘<supply-name>’ defined for corresponding pin in the UPF file

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

Power verification may be incorrect.

How to Debug and Fix

Make sure that the related supply ground pin defined for signal pin in the liberty file matches with related supply defined for the corresponding pin in the UPF file.

Message 6

The following message is reported when the voltage value for PG pin in the liberty file does not match with any of the state values defined in the UPF file:

[WARNING] Voltage value ‘<value>’ for pg_pin ‘<pin-name>’ in liberty does not match with any of the state values defined in the UPF file i.e '<value>'

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

Power verification may be incorrect.

How to Debug and Fix

Make sure that the voltage value for PG pin in the liberty file matches with any of the state values defined in the UPF file.

Message 7

The following message is reported when the signal pin does not have the `is_isolated` attribute in the liberty file but the UPF file infers the isolation rule defined for corresponding pin in the UPF file:

[WARNING] Signal pin ‘<pin-name>’ does not have attribute is_isolated in liberty but the UPF file infers the isolation rule ‘<rule-name>’ is defined for corresponding pin in the UPF file

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

Power verification may be incorrect.

How to Debug and Fix

Make sure that the signal pin has the `is_isolated` attribute defined in the liberty file.

Message 8

The following message is reported when the signal pin does not have isolation rule in UPF file but the `is_isolated` attribute is defined in the liberty file:

[WARNING] Signal pin '<pin-name>' does not have isolation rule in UPF file but is_isolated attribute is defined in liberty file

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

Power verification may be incorrect.

How to Debug and Fix

Make sure that the signal pin has isolation rule in the UPF file.

Message 9

The following message is reported when the signal pin has the no_isolation strategy in the UPF file but the is_isolated attribute is defined in the liberty file:

[WARNING] Signal pin '<pin-name>' have no_isolation strategy in UPF file but is_isolated attribute is defined in liberty file

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

Power verification may be incorrect.

How to Debug and Fix

Make sure that the signal pin has the no_isolation strategy in the UPF file.

Message 10

The following message is reported when the set_lib_name constraint is not specified in the SGDC file:

[WARNING] set_lib_name is not specified in SGDC file, it is mandatory for the rule to run

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

Power verification may be incorrect.

How to Debug and Fix

Make sure that the `set_lib_name` constraint is specified in the SGDC file.

Message 11

The following message is reported when voltage map corresponding to pin is not defined in the liberty file:

[WARNING] `Voltage map corresponding to pin '<pin-name>' is not defined in liberty file`

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

Power verification may be incorrect.

How to Debug and Fix

Make sure that the voltage map corresponding to pin is defined in the liberty file.

Message 12

The following message is reported when the *lp_related_pin_data_for_consistency_check* is generated:

[INFO] `Please refer report '<report-path>' for information on how related supplies have been interpreted for all pins`

Potential Issues

None

Consequences of Not Fixing

None

How to Debug and Fix

None

Example Code and/or Schematic

Example 1

Consider the following example.

UPF:

```
create_supply_port vddB
create_supply_port vssB
create_supply_port vddC
create_supply_port vssC
create_supply_net vddB_int
create_supply_net vssB
create_supply_net vddB
create_supply_net vssC
create_supply_net vddC
connect_supply_net vddB -ports { vddB }
connect_supply_net vddC -ports { vddC }
connect_supply_net vddB -ports { vddB }
connect_supply_net vssB -ports { vssB }
connect_supply_net vssC -ports { vssC }
create_power_switch pswA
...
-output_supply_port {out vddB_int}
-input_supply_port {in vddB}
create_power_switch pswB
...
-output_supply_port {out vddA_int}

-input_supply_port {in vddB}
```

Library:

```
library(libCellA){
cell ( cellA ) {
pg_pin ( vssA ) {
voltage_name : vssA
pg_type : primary_ground;
}
pg_pin ( vssB ) {
voltage_name : state3;
pg_type : primary_ground;
}
```

```
pg_pin ( vddA ) {
voltage_name : vddA;
pg_type : primary_power;
}
pg_pin ( vddB ) {
voltage_name : vddB;
pg_type : primary_power;
}
pg_pin ( vddB_int ) {
voltage_name : vddB_int;

pg_type : internal_power;
pg_funcion: "vddB";
}
pg_pin ( vddA_int ) {
voltage_name : vddA_int;
pg_type : internal_power;
pg_funcion: "vddA";
}
}
}
```

For the above example, the *UPF_lowpower26* rule reports the following violation messages because in the liberty file there are two PG pins named `vssA` and `vddA` for which there is no matching `create_supply_port` command in the UPF:

```
PG Pin vssA defined in liberty does not have a corresponding
supply port declaration in the UPF file
PG Pin vddA defined in liberty does not have a corresponding
supply port declaration in the UPF file
```

For the same example, the *UPF_lowpower26* rule reports the following violation messages because in the UPF file there are two supply ports, vssC and vddC, for which there is no matching PG pin in liberty file:

```
Supply port vddC does not have a corresponding pg_pin
declaration in the liberty file
```

```
Supply port vssC does not have a corresponding pg_pin
declaration in the liberty file
```

Also, for the same example, the *UPF_lowpower26* rule reports the following violation messages because in the direction of the PG pin vddA_int does not match with the direction of the supply port declared in the UPF file:

```
Direction input defined for pg_pin vddA_int in liberty does not
match with direction output defined for corresponding supply
port in the UPF file
```

Example 2

Consider the following example.

UPF:

```
set_related_supply_net -object_list{X} -power vddB -ground
vssB
set_related_supply_net -object_list{Y} -power vddB_int -
ground vssB
```

Library:

```
pin ( X ) {
direction: input;
is_isolated: true;
related_ground_pin: vssB;
related_power_pin: vddB;
}
pin ( Y ) {
direction: input;
related_ground_pin: vssB;
```

```
related_power_pin: vddB;
}
```

For the above example, the UPF_lowpower26 rule reports the following violation messages because the related supply for the signal pin Y (vddB) does not match with the related supply (vddB_int) defined in the UPF file using the set_related_supply_net command:

```
Related supply power pin of vddB defined for signal pin Y in
liberty does not match with related supply vddB_int defined for
corresponding pin in the UPF file
```

Example 3

Consider the following example.

UPF:

```
create_supply_port vddB

add_port_state -state {High 1.0} -state {Low 0.8} -state {off
off}
```

Library:

```
library(libCellA){
libvoltage_map(vddB, 1.2);
cell(cellA){
...
pg_pin(vddB){
pg_type: primary_power;
voltage_name: vddB;
}
...
}
}
```

For the above example, the *UPF_lowpower26* rule reports the following

violation messages because the nominal voltages defined for the supply port vddB in the UPF are 1.0 and 0.8. None of these match with the nom voltage value 1.2 defined using the voltage_map command in the liberty file:

```
Voltage value 1.2 for pg_pin vddB in liberty does not match
with any of the state values defined in the UPF file i.e 1.0 0.8
```

Example 4

Consider the following example.

UPF:

```
create_power_domain PD_cellA -include_scope
set_isolation internal_iso -domain PD_cellA -elements {X Y} -
location self
set_isolation iso1 -domain TOP_cellA -elements { W } -

no_isolation
```

Library:

```
library(libCellA){
cell(cellA){
...
pin ( X ) {
direction: input;
is_isolated: true;
}
pin ( Y ) {
direction: input;
}
pin ( Z ) {
is_isolated: true;
direction: input;
```

```
pin ( W ) {
direction: input;
is_isolated: true;
}}...
}
}
```

For the above example, the UPF_lowpower26 rule reports the following violation messages because the signal pin `Y` does not have the `is_isolated` attribute in the liberty file, but the pin `Y` is defined in the UPF file to have the isolation with location as self:

```
Signal pin Y does not have attribute is_isolated in liberty but the UPF file infers the isolation rule internal_iso is defined for corresponding pin in the UPF file

Signal pin z does not have isolation rule in UPF file but is_isolated attribute is defined in liberty file

Signal pin w have no_isolation strategy in UPF file but is_isolated attribute is defined in liberty file
```

Example 5

Consider the following example.

UPF:

```
create_supply_port vddA
add_port_state -state {High 1.0} -state {Low 0.8} -state {off off}
```

Library:

```
library(lib){
…
Cell(cellA){
pg_pin(vddA){
pg_type:primary_power;
direction:input;
voltage_name:vddC;
```

```
}
}
…
}
```

For the above example, the *UPF_lowpower26* rule reports the following violation messages because the voltage map corresponding to the pin `vddA` is not defined in the liberty file:

```
Voltage map corresponding to pin vddA is not defined in liberty file
```

Default Severity Label

Warning

Rule Group

UPF Check Rules

Reports and Related Files

lp_related_pin_data_for_consistency_check

UPF_lowpower27

Checks -ack_delay and -supply_set arguments of the create_power_switch command

When to Use

Use this rule to check `-ack_delay` and `-supply_set` arguments in the `create_power_switch` command.

Description

The *UPF_lowpower27* rule checks the following arguments in the `create_power_switch` command:

- `-ack_delay [-ack_delay {port_name delay}]*`
  - The `-ack_delay` argument specifies the acknowledge delay for a given acknowledge port. This option has no impact on Power Verify checks.
- `-supply_set [-supply_set supply_set_ref]`
  - The `-supply_set` argument specifies a supply set associated with a switch. `supply_set_ref` is a rooted name of a supply set or a supply set handle. It is an error if `supply_set_ref` is not specified.

Prerequisites

The `create_power_switch` command must be specified in the UPF.

Language

Verilog, VHDL

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_supply_port*
- *create_supply_net*

- *connect_supply_net*
- *add_port_state*
- *create_supply_set*
- *create_power_domain*
- *create_power_switch*
- *map_power_switch*
- *set_port_attributes*
- *set_isolation*
- *create_pst*
- *add_pst_state*
- *set_domain_supply_net*

Messages and Suggested Fix

Message 1

The following violation message is reported if the `supply_set` reference specified using the *`create_power_switch`* command is not defined in the active scope:

[ERROR] Supply set ‘<supply-set-name>’ specified in create_power_switch command ‘<power-switch-command-name>’ is not defined in active scope

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

Power switch specified may not work properly.

How to Debug and Fix

Make sure that the supply set is defined in the same scope as the `create_power_switch` command.

Message 2

The following message is reported if the `supply_set` argument is missing while the `ack_port` argument is specified in the `create_power_switch` command:

[ERROR] supply_set option missing for create_power_switch command ‘<power-switch-name>’, ack_port option provided for this command

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

Power switch specified may not work properly.

How to Debug and Fix

Make sure that the `supply_set` argument is also specified when the `ack_port` argument is specified in the `create_power_switch` command.

Message 3

The following message is reported if the supply set is not available in the domain mentioned in the `–domain` argument of the `create_power_switch` command:

[ERROR] supply net ‘<supply-net-name>’ specified in create_power_switch command ‘<power-switch-name>’ is not available in domain ‘<domain-name>’

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

Power switch specified may not work properly.

How to Debug and Fix

Make sure that the power supply net specified in the supply set is available in the domain mentioned in the `–domain` argument of the `create_power_switch` command.

Message 4

The following message is reported if the power net specified in supply set reference provided using the `create_power_switch` command is not always-on:

[ERROR] supply net ‘<supply-net-name>’ specified in create_power_switch command ‘<power-switch-name>’ is not

always-on

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

Power switch specified may not work properly.

How to Debug and Fix

Make sure that the power net specified in supply set reference provided using the `create_power_switch` command is always-on.

Example Code and/or Schematic

Consider the following UPF snippet.

```
set_scope inst1
create_supply_set ss1 -function {power VDD} -function {ground
VSS}
set_scope ../

create_supply_set ss2 -function {power VDDX} -function
{ground VSS}

create_power_switch psw1 -domain VD0 -input_supply_port
{VDDG VDD1} -output_supply_port {VDD VDDX} -control_port
{SLEEP en1} -control_port {SLEEP1 en2} -ack_port {SLEEPOUT
a1} -ack_port {SLEEPOUT1 b1} -supply_set inst1/ss1 -on_state
{on1 VDDG {SLEEP1&SLEEP2}} -off_state {off1
{!SLEEP1|!SLEEP2}}

map_power_switch psw1 -domain VD0 -lib_cells {PS2}

create_power_switch psw2 -domain VD0 -input_supply_port
{VDDG VDD2} -output_supply_port {VDD VDDX} -control_port
{SLEEP en3} -control_port {SLEEP1 en4} -ack_port {SLEEPOUT
en2} -ack_port {SLEEPOUT1 b2} -supply_set ss2 -on_state {on1
VDDG {en1&en2}} -off_state {off1 {!en1&!en2}}

map_power_switch psw2 -domain VD0 -lib_cells {PS2}
```

```
create_power_switch psw3 -domain VD0 -input_supply_port
{VDDG VDD2} -output_supply_port {VDD VDDX} -control_port
{SLEEP en3} -control_port {SLEEP1 en4} -ack_port {SLEEPOUT
en2} -ack_port {SLEEPOUT1 b2}

map_power_switch psw3 -domain VD0 -lib_cells {PS2}


add_port_state psw1/VDD -state {on1 1.0} -state {off1 off}
add_port_state psw2/VDD -state {on1 1.0} -state {off1 off}
```

For the above example:

- The following violation message is reported because the supply set `ss1` specified is not defined in the active scope:

```
Supply set 'ss1' specified in create_power_switch command
'psw1' is not defined in active scope
```

- The following violation message is reported because the supply net `VDDX` specified in supply set `ss2` specified in the `create_power_switch` command `psw2` is not always on:

```
supply net 'VDDX' specified in create_power_switch command
'psw2' is not always-on
```

- The following violation message is reported because the supply net `VDDX` specified in supply set `ss2` (specified in the `create_power_switch` command `psw2)` is not present in the domain `VD0` (specified in the `–domain` argument of the `create_power_switch` command `psw2)`:

```
supply net 'VDDX' specified in create_power_switch command
'psw2' is not available in domain 'VD0'
```

- The following violation message is reported because the `ack_port` argument is provided and the `supply_set` argument is not provided for the `create_power_switch` command `psw3`:

```
supply_set option missing for create_power_switch command
'psw3', ack_port option provided for this command
```

Default Severity Label

Error

Rule Group

UPF Check Rules

Reports and Related Files

None

UPF_lowpower28

Reports duplicate and conflicting set_related_supply_net constrains

When to Use

Use this rule to find duplicate and conflicting `set_related_supply_net` constrains.

Description

When there are two or more set_related_supply_net commands written for same port/pin in UPF and supply given from two commands are different, SpyGlass considers the last one. The UPF_lowpower28 rule reports this overriding and gives the actual *set_related_supply_net* command that applied was on a port/pin.

Prerequisites

By default, the *UPF_lowpower28* rule will not run. To enable this rule, select the *UPF_lowpower28* rule in the Atrenta Console GUI

Rule Exceptions

If the same supply (power and ground) is getting applied in multiple *set_related_supply_net* commands, then violation will not be reported.

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_supply_port*
- *create_supply_net*
- *connect_supply_net*
- *add_port_state*
- *create_power_domain*
- *set_related_supply_net*

Messages and Suggested Fix

The following message is reported if two or more set_related_supply_net commands written for same Port/Pin in upf and supply given from two commands are different:

[WARNING] Multiple set_related_supply_net commands< [file-name1,line-number1] [file-name2,line-number2]> specified for pin '<pin-name>'. Command specified at <[file-name,line-number]> will be considered for rule checking '

Potential Issues

The violation message explicitly states the potential issue.

Consequences of Not Fixing

The SpyGlass Power Verify solution will ignore the all the set_related_supply_net commands for <pin-name> except the one reported in the message, and will continue with the checking.

How to Debug and Fix

Back referencing is available for the command that got applied. To resolve the violation messages, review and update the various set_related_supply_net commands for the reported pin.

Example Code and/or Schematic

Example 1

Consider the following UPF snippet.

```
create_supply_set ss
create_supply_set ss1
create_supply_set ss2
create_power_domain TOP -include_scope -supply {primary ss}
create_power_domain PD1 -elements {inst1} -supply {primary
ss1}
create_power_domain PD2 -elements {inst2} -supply {primary
ss2}

create_supply_port VDD
add_port_state VDD    -state {active_state 1.4}
```

```
create_supply_net VDD
connect_supply_net VDD -ports VDD

create_supply_port VDDX
add_port_state VDDX    -state {active_state 1.4}
create_supply_net VDDX
connect_supply_net VDDX -ports VDDX
set_related_supply_net -object_list {inst2/and1/Y} -power
VDD
set_related_supply_net -object_list {inst2/and*/Y} -power
VDDX
```

For the above example, the *UPF_lowpower28* rule reports the following violation messages:

```
Multiple set_related_supply_net commands [ const.upf,31 ] [
const.upf,32 ]  specified for pin 'top.inst2.and1.Y'. Command
specified at [ const.upf,32 ]  will be considered for rule
checking
```

Default Severity Label

Warning

Rule Group

UPF Check Rules

Reports and Related Files

None

UPF_lowpower29

Checks if the connect_supply_net command for PG pins are written correctly

When to Use

Use this rule to find the incorrect *connect_supply_net* command used in a UPF file.

Description

This rule checks all the `connect_supply_net` commands and reports a violation when a ground net is connected to power pin or a power net is connected to ground pin. The rule checks only the leaf level PG pins given in the `-ports` option of the `connect_supply_net` command and reports a violation on the basis of `pg_type` liberty attribute written for that pin.

Violations are reported in the following scenarios:

Pin Type	Supply Net Type	Violation
primary_power internal_power backup_power	ground	Yes
primary_ground internal_ground backup_ground	power	Yes

Parameter(s)

None

Constraint(s)

UPF Commands

- *create_supply_port*
- *create_supply_net*
- *connect_supply_net*

- *add_port_state*
- *create_power_domain*
- *set_related_supply_net*

Messages and Suggested Fix

The following message is reported if a ground net is given in the *connect_supply_net* command for the power pin and vice versa:

```
[WARNING] <net_type> net <net_name> is connected with <pin type> pin <pin name>(cell : <cell name>) using connect_supply_net command
```

Potential Issues

Incorrect connection is specified.

Consequences of Not Fixing

If this connection is not fixed, the connection made by inferring these commands will also be incorrect.

How to Debug and Fix

The violation message gives the line number and file name of wrong *connect_supply_net* command. To resolve the violation messages, review and update that line by changing the net to correct type of net in `connect_supply_net` command.

Example Code and/or Schematic

Consider the following UPF snippet.

```
connect_supply_net VSS -ports {VSS inst2/bbgen/vana}
create_supply_set ss -function {power VDD} -function {ground VSS}
```

For the above example, the *UPF_lowpower29* rule reports the following violation message because a ground net VSS (inferred from *connect_supply_net* definition) is connected to a power pin vana (inferred from library definition):

```
'Ground' net 'VSS' is connected with 'power' pin 'vana'(cell : 'BBGEN') using connect_supply_net command
```

Default Severity Label

Warning

Rule Group

UPF Check Rules

Reports and Related Files

None

UPF_lowpower30

Checks association of the elements specified in set_retention_elements command with the retention strategy

When to Use

Use this rule to check the association of elements specified in the *set_retention_elements* command with the retention strategy.

Description

The UPF_lowpower30 rule checks whether all elements specified under *set_retention_elements* command are retained under a strategy. There can be four scenarios:

- **All the elements are retained under the same strategy**: In this case no violation will be reported
- **All the elements are retained but under different strategies**: In this case a warning message is generated, which informs that elements in the same element list are retained under different strategies
- **Some of the elements are retained**: An error message is reported with mention of the list of elements that are not retained under any strategy
- **None of the elements are retained**: An error message is reported with mention of the list of elements that are not retained under any strategy

Parameter(s)

None

Constraint(s)

UPF Commands

- *set_retention*
- *set_retention_elements*

Messages and Suggested Fix

Message 1

The following message is reported when no elements specified in the *set_retention_elements* command are associated with a retention strategy:

[UPF_lowpower30_1][ERROR] None of the elements '<element-list>' specified in set_retention_elements command '<command-name>' are associated with a retention strategy

Message 2

The following message is reported when elements specified in the *set_retention_elements* command are associated with multiple retention strategies:

[UPF_lowpower30_2][ERROR] Elements specified in set_retention_elements command '<command-name>' are associated with multiple retention strategies '<retention-strategy-names>'

Message 3

The following message is reported when elements specified in the *set_retention_elements* command are not associated any retention strategies:

[UPF_lowpower30_3][ERROR] Elements '<element-list>' specified in set_retention_elements command '<command-name>' are not associated with any retention strategy

Message 4

The following message is reported when elements specified in the exclude list of the *set_retention_elements* command are associated a retention strategy:

[UPF_lowpower30_4][INFO] Element(s) '<element-list>' specified in exclude element list of set_retention_elements command '<command-name>' are associated with retention strategy

Message 5

The following message is reported when all elements specified in the the *set_retention_elements* command are associated a retention strategy:

[UPF_lowpower30_5][INFO] All element(s) specified in set_retention_elements command '<command-named>' are associated with retention strategy '<retention-strategy-name>'

Potential Issues

Potential issues are explicitly mentioned in the violation messages.

Consequences of Not Fixing

The desired retention of elements may not be honored, which may lead to signal corruption.

How to Debug and Fix

To resolve the violation messages, add the necessary retention commands.

Example Code and/or Schematic

Consider the following UPF snippet.

```
set_retention_elements list1 -elements { inst* }  -
exclude_elements { inst1/buf1 }

set_retention ret -domain TOP -retention_power_net VDDB -
retention_ground_net VSS -elements {inst1/buf1} -save_signal
{ save high } -restore_signal { restore high }
set_retention ret1 -domain TOP -retention_power_net VDDB -
retention_ground_net VSS -elements {inst1/buf2} -save_signal
{ save high } -restore_signal { restore high }
```

For the above example, the *UPF_lowpower30* rule reports the following violation messages because:

- The reported elements are not associated with the retention strategy
- The reported element is associated with the retention strategy but is also specified in the exclude element list

```
Elements 'top.inst1.buf3 top.inst1.buf4 ' specified in
set_retention_elements command 'list1' are not associated with
any retention strategy
```

```
Element(s) 'top.inst1.buf1' specified in exclude element list
of set_retention_elements command 'list1' associated with
retention strategy
```

Default Severity Label

Error, Info

Rule Group

UPF Check Rules

Reports and Related Files

None

UPF_lowpower31

Checks the connectivity of logic nets (used as control signals) in upf

When to Use

Use this rule to check connectivity-related issues if the control nets used in isolation/power switch/retention strategies are defined as logic net by using the `create_logic_net` UPF command.

Description

The `UPF_lowpower31` rule flags violation in below scenarios:

- If a logic net that is used as control signal is not driven by a top level logic port. Connectivity tracing is done in UPF only.
- If a logic net that is used as control signal is driven by more than one driving logic port.
- If a logic port is created on a module hierarchy which is not a domain boundary.

Parameter(s)

None

Constraint(s)

UPF Commands

- `create_logic_net`
- `create_logic_port`
- `connect_logic_net`
- *set_isolation*
- *create_power_switch*
- *set_retention*

Messages and Suggested Fix

Message 1

The following message is reported if a logic net that is used as a control signal is not driven by a top level logic port:

[Error] Logic net <net-name> is not connected to top level input logic port

Message 2

The following message is reported if a logic net that is used as control signal is driven by more than one driving logic port:

[Error] Logic net <net-name> is driven by multiple input ports

Message 3

The following message is reported if a logic port is created on a module hierarchy which is not a domain boundary:

[Error] Port <Port-name> given in create_logic_port command is not on domain boundary. No checks will be performed on this port

Potential Issues

These violations indicate that the control signals in the design have improper connectivity. Due to improper connectivity of control signals functioning of power cells (Isolation/Power switch/Retention) may be incorrect.

Consequences of Not Fixing

Incorrect functioning of isolation/power switch/retention cells may lead to design issues.

How to Debug and Fix

To resolve the violation messages, correct the connectivity issues pointed out by violation messages.

Example Code and/or Schematic

Consider the following UPF snippet which has the top level domain TOP and nested domains A (Element A1) and power domain B (element within B: A1/B1). By default scope is top level.

```
create_logic_port temp -direction in
create_logic_net temp
set_scope A1
create_logic_port temp1 -direction in
create_logic_net temp1
connect_logic_net temp1 -ports temp1
set_scope B1
create_logic_port temp11 -direction in
create_logic_net temp11
connect_logic_net temp11 -ports temp11
set_isolation iso1 -domain B -applies_to outputs -
isolation_supply_set ss1
set_isolation_control iso1 -domain B -isolation_signal
temp11 -location parent
set_scope ..
connect_logic_net temp1 -ports { B1/temp11}
set_scope
connect_logic_net temp -ports {A1/temp}
```

For the above example, the *UPF_lowpower31* rule reports the following violation messages because the connection between top-level design port temp and top-level design net temp is missing:

```
Logic net 'B1/temp11' is not connected to top level logic input
port
```

Default Severity Label

Error

Rule Group

UPF Check Rules

Reports and Related Files

None

UPFSEM_1

Checks the presence of the specified power domain

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_1* rule reports a violation message when the power domain specified is not found in the UPF file.

Messages and Suggested Fix

The following message appears when a power domain *<power-domain-name>* specified is not found in the UPF file:

[FATAL] Specified power domain <power-domain-name> not found

Potential Issues

Power-Domain is not created while it is being accessed and specified with different power-strategies.

Consequence of Not Fixing

Power intent cannot be analyzed, so no power verification is performed.

How to Debug and Fix

Create the power domains using create_power_domain *<power-domain-name>* command before using them in different power-strategies.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_supply_port vdd1 –domain pd_top
create_supply_net vdd1 –domain pd_top
```

For the above UPF snippet, the *UPFSEM_1* rule reports the following violation because the power domain pd_top has not been created:

```
Specified power domain ( pd_top ) not found
```

To resolve the above violation, create the power domain using the

creat_power_domain command, as shown below:

```
create_power_domain pd_top
```

Default Severity Label

Fatal

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_2

Checks the presence of the supply port in scope

When to Use

This is a setup rule and always runs by default.

Description

The UPFSEM_2 rule reports a supply port that is not found in the scope.

Messages and Suggested Fix

The following message appears when a supply port *<supply-port-name>* is not found in the scope:

[FATAL] `Invalid supply port <supply-port-name> specified, cannot find in current scope`

Potential Issues

Supply Port is not created while it is being accessed and specified with different power strategies.

Consequence of Not Fixing

Power intent cannot be analyzed, so no power verification is performed.

How to Debug and Fix

Create the supply port using *create_supply_port* *<supply-port-name>* command before use them in different power-strategies.

Example Code and/or Schematic

Consider the following UPF snippet:

```
connect_supply_net vdd –ports VDD
connect_supply_net vdd1 –ports VDD
```

For the above UPF snippet, the *UPFSEM_2* rule reports the following violation because the supply port VDD has not been created:

```
Invalid supply port ( VDD ) specified, cannot find in current scope
```

To resolve the violation, create a supply port using *create_supply_port* command, as shown below:

```
create_supply_port VDD –domain PD
```

Default Severity Label

Fatal

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_3

Checks the presence of PST

When to Use

This is a setup rule and always runs by default.

Description

The UPFSEM_3 rule reports a violation when a PST is not found in the UPF.

Messages and Suggested Fix

The following message appears when an invalid pst *<pst-name>* is specified:

[FATAL] Invalid pst <pst-name> specified

Potential Issues

The PST specification is missing the UPF.

Consequence of Not Fixing

Power verification does not happen correctly as PST states cannot be linked to the relevant PSTs.

How to Debug and Fix

Create the PST using *create_pst* *<pst-name>* command before using them in the *add_pst_state*.

Example Code and/or Schematic

Consider the following UPF snippet:

```
add_pst_state s1 -pst pst2 -state {vnn_top_on}
```

For the above UPF snippet, the *UPFSEM_3* rule reports the following violation because the pst2 is not created in UPF.

```
Invalid pst ( pst2 ) specified
```

To resolve the violation, specify the PST using *create_pst* UPF command, as shown below:

```
create_pst pst2 -supplies {vnn_top }
```

Default Severity Label

Fatal

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_4

Checks the presence of the supply net

When to Use

This is a setup rule and always runs by default.

Description

The UPFSEM_4 rule reports a violation when a supply net is not found in the UPF file.

Messages and Suggested Fix

The following message appears when an invalid supply net *<supply-net-name>* specified is not found:

[FATAL] `Invalid supply net <supply-net-name> specified`

Potential Issues

The supply net is not found in the active scope.

Consequence of Not Fixing

Power intent is not analyzed correctly as the supply net is not specified but getting used in various UPF commands.

How to Debug and Fix

Create a supply net in the UPF using *create_supply_net* command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
connect_supply_net vnn_top -ports {vnn_top}
```

For the above UPF snippet, the *UPFSEM_4* rule reports the following violation because the supply net vnn_top is not found in the UPF:

```
Invalid supply net ( vnn_top ) specified
```

To resolve the violation, create a supply net using *create_supply_net* UPF command, as shown below:

```
create_supply_net vnn_top
```

Default Severity Label

Fatal

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_5

Checks the presence of the isolation in the specified domain

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_5* rule reports a violation when the isolation strategy is not found in the domain specified.

Messages and Suggested Fix

The following message appears when isolation <*isolation-name*> is not found in the domain <*domain-name*> specified:

[ERROR] Invalid isolation name <isolation-name> specified in domain <domain-name>

Potential Issues

The isolation strategy corresponding to isolation-control specified is missing.

Consequence of Not Fixing

Isolation control is not applied since the strategy is missing.

How to Debug and Fix

Specify the isolation strategy corresponding to the isolation control.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_isolation_control iso2 \
            -domain PD \
            -isolation_signal iso1
```

For the above UPF snippet, the *UPFSEM_5* rule reports the following violation because the isolation strategy not specified but relevant isolation control is specified in the UPF:

```
Invalid isolation name ( iso2 ) specified in domain ( PD )
```

To resolve the violation, specify the isolation strategy using the

set_isolation UPF command, as shown below:

```
set_isolation iso2 –domain PD
```

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_6

Checks the presence of retention in the specified domain

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_6* rule reports a violation when retention is not found in the domain specified.

Messages and Suggested Fix

The following message appears when retention *<retention-strategy-name>* is not found in the domain *<domain-name>* specified:

[ERROR] Invalid retention strategy name <retention-strategy-name> specified in domain <domain-name>

Potential Issues

The retention strategy is not created while it is being specified with *set_retention_control* or *map_retention_cell* commands.

Consequence of Not Fixing

The *set_retention_control* or *map_retention_cell* commands are not applied.

How to Debug and Fix

Specify the retention strategy before using it in a UPF command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_retention_control ret2 \
     -domain PD2 \
     -save_signal {in1 low}
```

OR

```
map_retention_cell ret2\
–domain PD2 \
```

```
-lib_cells {RET1}
```

For the above UPF snippet, the *UPFSEM_6* rule reports the following violation because the retention strategy is not specified and relevant *set_retention_control*/*map_retention_cell* is specified in the UPF:

```
Invalid retention strategy name ( ret2 ) specified in domain ( PD2 )
```

To resolve the violation, specify the retention strategy using the *set_retention* UPF command, as shown below:

```
set_retention ret2 -domain PD2 \
              -retention_power_net VPD
```

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_7

Checks the presence of the acknowledge port specified in power switch

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_7* rule reports a violation when the acknowledge port specified in a power switch is not found.

Messages and Suggested Fix

The following message appears when an invalid acknowledge port <*ack-port-name*> is found in power switch <*power-switch-name*>:

[ERROR] Invalid acknowledgement port <ack-port-name> found in power switch <power-switch-name> definition in power domain <power-domain-name>

Potential Issues

The acknowledge port specified in UPF for power switch is not found in the design or the design does not have any relevant power switch instance.

Consequence of Not Fixing

The power analysis may not be done correctly.

How to Debug and Fix

Specify the power switch in the design or specify the acknowledge port with power switch instance specified in the UPF.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_switch psw2 -domain PD2 \
        -input_supply_port {VDDG VDD} \
        -output_supply_port {VDD VDDX} \
        -ack_port {SLEEPOUT Ack_psw2}  \
        -ack_delay {VDDP 0.9}
```

OR

```
create_power_switch psw2 -domain PD2 \
        -input_supply_port {VDDG VDD} \
        -output_supply_port {VDD VDDX} \
        -ack_delay {VDDP 0.9}
```

For the above UPF snippet, the *UPFSEM_7* rule reports the following violation because the acknowledge port specified with *create_power_switch* command is missing:

Invalid acknowledgement port (VDDP) found in power switch (psw1) definition in power domain (PD)

To resolve the violation, specify the acknowledge port in the design corresponding to *create_power_swtch* command.

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_8

Checks the presence of the level shifter specified for the power domain

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_8* rule reports a violation when the level shifter specified for the power domain is not found.

Messages and Suggested Fix

The following message appears when level shifter *<level-shifter-name>* specified for the power domain *<power-domain-name>* is not found:

[ERROR] `Invalid level shifter <level-shifter-name> specified at power domain <power-domain-name>`

Potential Issues

The level shifter strategy is not created while it is being accessed and specified with the UPF commands.

Consequence of Not Fixing

The power analysis may not be done correctly.

How to Debug and Fix

Create level shifter strategy before using it in the power UPF commands using the *set_level_shifter* *<level shifter name>*.

Example Code and/or Schematic

Consider the following UPF snippet:

```
map_level_shifter_cell ls1 -domain PD1 -lib_cells {LS1}
```

For the above UPF snippet, the *UPFSEM_8* rule reports the following violation because the level shifter strategy ls1 is not specified and the relevant *map_level_shifter_cell* is specified in UPF:

```
Invalid level shifter ( ls1 ) specified at power domain (PD)
```

To resolve the violation, specify the level shifter strategy using the *set_level_shifter* UPF command, as shown below:

```
set_level_shifter ls1 –domain PD
```

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_9

Checks the presence of the power switch specified for the domain

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_9* rule reports a violation when power switch specified for the domain is not found.

Messages and Suggested Fix

The following message appears when power switch *<power-switch-name>* specified for the domain *<domain-name>* is not found:

[ERROR] Power switch <power-switch-name> not found in specified domain <domain-name>

Potential Issues

The power switch is not created but is getting used on the UPF commands.

Consequence of Not Fixing

The power switch functionality is not analyzed or checked. The missing power switch may lead to a design failure if it is driving supply to power domain.

How to Debug and Fix

Define a power switch using the *create_power_switch* *<power-switch-name>* command before using the power switch in the commands.

Example Code and/or Schematic

Consider the following UPF snippet:

```
map_power_switch psw2 -domain PD2 -lib_cells {PS1}
```

For the above UPF snippet, the *UPFSEM_9* rule reports the following violation because the power switch psw2 is not specified but relevant switch is mapped using map_power_switch in UPF:

```
Power switch ( psw2 ) not found in specified domain ( PD2 )
```

To resolve the violation, specify power-switch using create_power_switch UPF command, as shown below:

```
create_power_switch psw2 -domain PD2 \
        -input_supply_port {VDDG VDD} \
        -output_supply_port {VDD VDDX} \
        -control_port {SLEEPOUT Ack_psw2}
```

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_11

Checks the presence of supply port/net specified in the PST

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_11* rule reports a violation when supply port/net specified in the PST is not found.

Messages and Suggested Fix

The following message appears when supply port/net specified in the PST *<pst-name>* is not found:

[FATAL] `Invalid element <element-name> specified in pst <pst-name>`

Potential Issues

The supply net/port in the UPF is missing, which is specified in PST.

Consequence of Not Fixing

The functionality of blocks/modules that are operating on this supply is hindered.

How to Debug and Fix

Specify the supply-net/port before using it in the PST table.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_pst ps1 -supplies {VAON VVD VPD VPD2X VSS}
```

For the above UPF snippet, the *UPFSEM_11* rule reports the following violation because the supply-net/port VPD2X specified in PST state is not found in the UPF:

```
Invalid element ( VPD2X ) specified in pst ( ps1 )
```

To resolve the above violation, specify supply-net/port before using it in the PST, as shown below:

```
create_supply_net VPD2X
```

or

```
create_supply_port VPD2X
```

Default Severity Label

Fatal

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_12

Checks the presence of the input supply port for power switch

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_12* rule reports a violation when input supply port is not found for the power switch.

Messages and Suggested Fix

The following message appears when input supply port *<input-supply-port-name>* is not found for the power switch *<power-switch-name>* specification:

[FATAL] Input supply port <input-supply-port-name> not found for power switch <power-switch-name>

Potential Issues

The supply port that is used in *input_supply_port* of power switch strategy is not created using the *create_supply_port* command.

Consequence of Not Fixing

The power switch functionality is not analyzed or checked. The missing power switch may lead to a design failure if it is driving supply to a power domain.

How to Debug and Fix

Create the supply port using the *create_supply_port <port-name>* before using it in the *input_supply_port* option in power switch strategy.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_switch psw1 -domain PD \
-input_supply_port {VDD XX} \
-output_supply_port {VDDG VPDY} \
-ack_port {SLEEPOUT VDDX} \
```

```
-on_state {on1 XX {on}}
```

For the above UPF snippet, the *UPFSEM_12* rule reports the following violation because the supply port XX used in the *input_supply_port* has not been created, which has been reused in the *on_state* option:

```
Input supply port ( XX ) not found for power switch ( psw1 )
```

To resolve the violation, create supply port using the *create_supply_port* UPF command, as shown below:

```
create_supply_port XX
```

Default Severity Label

Fatal

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_13

Checks if an element is already owned by another power domain

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_13* rule reports a violation when an element is already owned by another power domain.

Messages and Suggested Fix

The following message appears when element *<element-name>* is already owned by another power domain *<power-domain-name>*:

[ERROR] Element <element-name> already owned by the power domain <power-domain-name>

Potential Issues

The same instance is associated with different power domains.

Consequence of Not Fixing

The power intent cannot be analyzed, thus no power verification is performed.

How to Debug and Fix

Avoid assigning the same instance for multiple power domains.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_domain PD1 -elements {inst1/inst2}
create_power_domain PD2 -elements {inst2 inst1/inst2}
```

For the above UPF snippet, the *UPFSEM_13* rule reports the following violation because the instance inst1/inst2 is associated with both PD1 and PD2:

```
Element ( inst1/inst2 ) already owned by the power domain ( / PD1 )
```

To avoid this violation remove instance inst1/inst2 from PD1 or PD2.

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_14

Checks if a net already exists

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_14* rule reports a violation when the net specified already exists.

Messages and Suggested Fix

The following message appears when the net *<net-name>* specified already exists:

[ERROR] Net <net-name> already exists

Potential Issues

The supply net is created with the same name that is already created in the scope.

Consequence of Not Fixing

None

How to Debug and Fix

Avoid creating supply nets that are already created with the same name under the same scope.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_supply_net VPD2 -domain PD2
create_supply_net  VPD2 -domain PD2
```

For the above UPF snippet, the *UPFSEM_14* rule reports the following violation because supply net VPD2 is created twice in the same scope:

```
Net ( VPD2 ) already exists
```

To avoid this violation, only one supply net should be created under the same scope with the same name.

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_15

Checks if a port already exists

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_15* rule reports a violation when the port specified already exists.

Messages and Suggested Fix

The following message appears when the port specified already exists:

[ERROR] `Port <port_name> already exists`

Potential Issues

The supply port is created with the same name that is already created in same scope.

Consequence of Not Fixing

None

How to Debug and Fix

Avoid creating supply ports that are already created with the same name under the same scope.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_supply_port  VPD -domain PD2
create_supply_port  VPD -domain PD2
```

For the above UPF snippet, the *UPFSEM_15* rule reports the following violation because supply net VPD2 is created twice in the same scope:

```
Port ( VPD ) already exists
```

To avoid this violation only one supply port should be created under same scope with same name.

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_16

Checks if the named object already exists

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_16* rule reports a violation when the named object already exists in the same scope.

Messages and Suggested Fix

The following message appears when the named object *<object-name>* already exists in the same scope:

[ERROR] Object <object-name> already exists in the same scope

Potential Issues

An object is specified multiple times in the same scope.

Consequence of Not Fixing

The check does not happen for all the objects.

How to Debug and Fix

Specify only one object with the same name under the same scope.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_switch psw1 -domain PD -output_supply_port
{VDDG VPDY} -ack_port {SLEEPOUT VDDX}
create_power_switch psw1 -domain PD -output_supply_port
{VDDG VPDY} -ack_port {SLEEPOUT VDDX}
```

For the above UPF snippet, the *UPFSEM_16* rule reports the following violation because power switch psw1 is specified twice under the top scope:

```
UPF_POWER_SWITCH ( psw1 ) already exists in the same scope
```

To resolve this violation, specify the second power switch with a different name.

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_18

Checks if invalid design elements are found in element list

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_18* rule reports a violation when an invalid design element is specified in the element list.

Messages and Suggested Fix

The following message appears when an invalid design element *<element-name>* is specified in the element list:

[ERROR] `Illegal design element list specified : <element-name>`

Potential Issues

An illegal design element is specified in the `–elements` argument of a UPF command.

Consequence of Not Fixing

The required checks do not take place for such UPF commands.

How to Debug and Fix

Specify only legal design elements inside the `–elements` argument.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_domain PD -elements {inst ..}
```

For the above UPF snippet, the *UPFSEM_18* rule reports the following violation because the design element “..” is illegal:

```
Illegal design element list specified : ..
```

To avoid this violation, remove the illegal design element “..” from the `–elements` argument.

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_19

Checks the presence of a supply set referred in command

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_19* rule reports a violation when the supply set referred in a command is not found.

Messages and Suggested Fix

The following message appears when the supply set *<supply-set-name>* referred in a command *<command>* is not found:

[ERROR] Supply Set <supply-set-name> referred in command <command> not found

Potential Issues

The supply set is not created while it is specified with the other UPF commands.

Consequence of Not Fixing

The power analysis may not be done correctly.

How to Debug and Fix

Create supply set using the *create_supply_set* *<supply-set-name>* before specifying it in other UPF commands.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_supply_set ss1 -update
```

For the above UPF snippet, the *UPFSEM_19* rule reports the following violation because supply set ss1 has not been created previously in the UPF:

```
Supply Set ( ss1 ) referred in command ( create_supply_set ) not found
```

To resolve the violation, create supply set using create_supply_set UPF

command, as shown below:

```
create_supply_set ss1 \
–function {power vdd} \
–function {ground vss}
```

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_21

Checks if the supply set already exists

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_21* rule reports a violation when the supply set specified already exists.

Messages and Suggested Fix

The following message appears when the supply set *<supply-set-name>* specified already exists:

[ERROR] Supply Set <supply-set-name> already exists

Potential Issues

The supply set created is already existing in the same scope.

Consequence of Not Fixing

The power analysis may not be done correctly.

How to Debug and Fix

Avoid creating duplicate supply sets under the same scope.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_supply_set ss1 -function {power VAON} -function
{ground VSS}
create_supply_set ss1 -function {power VAON} -function
{ground VSS}
```

For the above UPF snippet, the *UPFSEM_21* rule reports the following violation because supply set ss1 is already created in the same scope:

```
Supply Set ( ss1 ) already exists
```

To resolve this violation, avoid create duplicate supply sets under the same scope.

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_22

Checks the presence of the net referred

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_22* rule reports a violation when the net referred in a command does not exist.

Messages and Suggested Fix

The following message appears when the net *<net-name>* referred in a command *<command>* does not exist:

[ERROR] Net <net-name> referred in command <command> does not exist

Potential Issues

The supply net is not created while it is being accessed and specified with different UPF commands.

Consequence of Not Fixing

The power analysis may not be done correctly.

How to Debug and Fix

Create the supply net using *create_supply_net* *<supply-net-name>* command before using it in other UPF commands.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_supply_set ss1 -function {power VAONx} -function
{ground VSSx}
```

For the above UPF snippet, the *UPFSEM_22* rule reports the following violation because supply net VAONx is not created previously using create_supply_net command:

```
Net ( VAONx ) referred in ( ss1 ) does not exist
```

To resolve the violation, create the supply net using the creat_supply_net

command, as shown below:

```
create_supply_net VAONx
```

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_23

Checks if an invalid logic net is specified

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_23* rule reports a violation when an invalid logic net is specified.

Messages and Suggested Fix

The following message appears when an invalid logic net *<net-name>* is specified:

[ERROR] `Invalid logic net <net-name> specified`

Potential Issues

Undefined logic net is used in a UPF command.

Consequence of Not Fixing

The required checks do not take place for such UPF commands.

How to Debug and Fix

Specify the logic net using the `create_logic_net <net-name>` command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
connect_logic_net xx -ports VSS
```

For the above UPF snippet, the *UPFSEM_23* rule reports the following violation because the logic net `xx` is not a predefined logic net:

```
Invalid logic net ( xx ) specified
```

To avoid this violation, specify the logic net `xx` as shown below:

```
create_logic_net xx
```

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_24

Checks if an invalid supply set is specified

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_24* rule reports a violation when an invalid supply set is specified.

Messages and Suggested Fix

The following message appears when an invalid supply set *<supply-set-name>* is specified:

[ERROR] Invalid supply set <supply-set-name> specified

Potential Issues

The supply set is not created while it is being accessed and specified with different UPF commands.

Consequence of Not Fixing

The power analysis may not be done correctly.

How to Debug and Fix

Create the supply set using the *create_supply_set* *<supply-set-name>* command before using them in different commands.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_isolation iso1 -domain PD2 -elements {inst1/in1 inst1/
in1} -isolation_supply_set ss1
```

For the above UPF snippet, the *UPFSEM_24* rule reports the following violation because supply set ss1 has not been created before use it:

```
Invalid supply set ( ss1 ) specified
```

To resolve this violation, create supply set using the *create_supply_set* UPF command, as shown below:

```
create_supply_set ss1
```

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_25

Checks if an invalid function-net pair is provided in supply set

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_25* rule reports a violation when an invalid function-net pair is provided in the supply set.

Messages and Suggested Fix

The following message appears when an invalid function-net pair *<net-name>* is provided in the supply set *<supply-set-name>*:

[ERROR] Invalid function net pair <net-name> provided in supply set <supply-set-name>

Potential Issues

Specifies an invalid function-net pair in supply set `-function` option.

Consequence of Not Fixing

The power analysis may not be done correctly.

How to Debug and Fix

Add a valid function-net pair in the supply set using the `-function {<function name> <net-name>}` command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_supply_set ss1 -function {power VAON} -function {ground VSS} -function {power VSS VAON}
```

For the above UPF snippet, the *UPFSEM_25* rule reports the following violation because function-net pair {power VSS VAON} is invalid:

```
Invalid function net pair ( power VSS VAON ) provided in supply set ( ss1 )
```

To resolve this violation, specify valid function-net pair in the `–function`

option of supply set, as shown below:

```
create_supply_set ss1 -function {power VAON} -function
{ground VSS} -function {power VSS}
```

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_29

Checks if an invalid port is specified, cannot be found in current scope

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_29* rule reports a violation when an invalid port is specified and cannot be found in the current scope.

Parameters

lp_report_port_state_on_non_supply_port: Default value is no. Set this parameter to `yes` to report the specification of the *add_port_state* command on non-supply ports that are not already created with the *create_supply_port* command.

Messages and Suggested Fix

The following message appears when an invalid port *<port-name>* is specified and cannot be found in current scope:

[ERROR] `Invalid port <port-name> specified, cannot find in current scope`

Potential Issues

Undefined supply port is specified in a UPF command.

Consequence of Not Fixing

The required checks do not take place for such UPF commands.

How to Debug and Fix

Specify the supply port using the `create_supply_port <port-name>` command.

Example Code and/or Schematic

Example 1

Consider the following UPF snippet:

```
set_port_attributes -ports {in} -related_power_port VAON -
related_ground_port XXX
```

For the above UPF snippet, the *UPFSEM_29* rule reports the following violation because an invalid port `XXX` is specified that cannot be found in the current scope:

```
Invalid port ( XXX ) specified, cannot find in current scope
```

To avoid this violation, specify the supply port `XXX`, as shown below:

```
create_supply_port XXX
```

Example 2

Consider the following UPF snippet:

```
create_supply_net VDD
add_port_state VDD –state {ON 1.0}
```

For the above UPF snippet, when the *lp_report_port_state_on_non_supply_port* parameter is set to `yes`, the *UPFSEM_29* rule reports the following violation message because the `VDD` port is not created using the *create_supply_port* command:

```
Invalid port ( VDD ) specified, cannot find in current scope
```

Default Severity Label

Error

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSEM_31

Checks if same string is specified in both off and full_on tools

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_31* rule reports a violation when same string is specified in both off and full_on tools.

Messages and Suggested Fix

The following message appears when same string <*string*> is specified in both off and full_on tools:

```
[ERROR] Same string <string> is specified in both off and
full_on tools
```

Potential Issues

The same string is specified for both `–full_on_tools` and `-off_tools` options in the *set_partial_on_translation* command.

Consequence of Not Fixing

The power analysis may not be done correctly.

How to Debug and Fix

Specify different strings for `full_on_tools` and `off_tools` options in the *set_partial_on_translation* command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_partial_on_translation OFF -full_on_tools {tool1} -
off_tools {tool1}
```

For the above UPF snippet, the *UPFSEM_31* rule reports the following violation because string tool1 is specified in both `–full_on_tools` and `-off_tools` options:

```
Same string ( tool1 ) is specified in both off and full_on
```

`tools`

To resolve this violation, specify different strings in `-full_on_tools` and `-off_tools` options of the *set_partial_on_translation* command, as shown below:

```
set_partial_on_translation OFF -full_on_tools {tool1} -
off_tools {tool2}
```

Default Severity Label

Error

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSEM_32

Checks if an invalid handle is specified for supply set

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_32* rule reports a violation when an invalid handle is specified for a supply set.

Messages and Suggested Fix

The following message appears when an invalid handle *<handle-name>* is specified for a supply set *<supply-set-name>*:

[ERROR] Invalid Handle <handle-name> specified for supply set <supply-set-name>

Potential Issues

The same supply set handle is specified multiple times.

Consequence of Not Fixing

The power analysis may not be done correctly.

How to Debug and Fix

Avoid specifying the same supply set handle multiple times.

Example Code and/or Schematic

Consider the following UPF snippet:

```
associate_supply_set ss1 -handle AON.primary -handle
AON.primary
associate_supply_set ss1 -handle AON.primary -handle
AON.primary
```

For the above UPF snippet, the *UPFSEM_32* rule reports the following violation because supply set ss1 handles *AON.primary* twice:

```
Invalid handle ( AON.primary ) specified for supply set ( ss1 )
```

To resolve the violation, avoid specifying duplicate supply set handles.

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_33

Checks if an invalid value is specified for a parameter in command

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_33* rule reports a violation when an invalid value is specified for a parameter in a command.

Messages and Suggested Fix

The following message appears when an invalid value is specified for a parameter <*parameter-name*> in a command <*command*>:

[FATAL] `Invalid value <value> specified for parameter <parameter-name> in command <command>`

Potential Issues

An invalid value is specified for a parameter in a UPF command.

Consequence of Not Fixing

The required checks do not take place for such UPF commands.

How to Debug and Fix

Specify only valid values for the parameters in UPF commands.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_isolation iso1 -domain PD2 -elements {inst1/in1} -
isolation_supply_set ss1 -isolation_signal iso -
isolation_sense {xx}
```

For the above UPF snippet, the *UPFSEM_33* rule reports the following violation because value `xx` is not a valid value for the parameter `isolation_sense`:

```
Invalid value ( xx ) specified for parameter ( isolation_sense
) in command ( set_isolation )
```

To avoid this violation, specify only valid values {`low | high`} for the parameter `isolation_sense`.

Default Severity Label

Fatal

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSEM_34

Checks if an invalid command is specified inside the begin_power_model command

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_34* rule reports a violation when any of the following command(s) is(are) specified inside the `begin_power_model` command:

- name_format
- save_upf
- set_scope
- load_upf -scope
- begin_power_model
- end_power_model

Messages and Suggested Fix

The following message appears when any of the above command(s) is(are) specified inside the `begin_power_model` command:

[FATAL] Command (<command-name>) not allowed within begin_power_model command

Potential Issues

Specified command is not allowed inside `begin_power_model` command.

Consequence of Not Fixing

The relevant checks are not performed properly for the UPF command.

How to Debug and Fix

To fix this violation, do not use any of the command(s), listed in the description section above, inside the `begin_power_model` command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
begin_power_model
...
set_scope  u1
...
end_power_model
```

For the above UPF snippet, the *UPFSEM_34* rule reports the following violation because the `set_scope` command is specified in the `begin_power_model` command:

```
Command (set_scope) not allowed within begin_power_model
command
```

Default Severity Label

Fatal

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSEM_35

Checks whether the end_power_model command is used with the begin_power_model command

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_35* rule reports a violation when the `end_power_model` command is not used (in pair) with the `begin_power_model` command.

Messages and Suggested Fix

The following message appears when the `end_power_model` command is not used (in pair) with the `begin_power_model` command:

[FATAL] `Command end_power_model not used with begin_power_model command`

Potential Issues

The `end_power_model` command is not used with the `begin_power_model` command.

Consequence of Not Fixing

The relevant checks are not performed properly for the UPF command.

How to Debug and Fix

To fix this violation, always end the `begin_power_model` command scope with `end_power_model` command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
begin_power_model
...
set_scope  u1
...
```

```
<end of file> -> end_power_model command missing
```

For the above UPF snippet, the *UPFSEM_35* rule reports the following violation because the `end_power_model` command is not used with the `begin_power_model` command:

```
Command end_power_model not used with begin_power_model command
```

Default Severity Label

Fatal

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSEM_36

Checks if an invalid supply set pair is specified in supply map of the apply_power_model command

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_36* rule reports a violation when an invalid supply set pair is specified in the `-supply_map` option of the `apply_power_model` command.

Messages and Suggested Fix

The following message appears when an invalid supply set pair is specified in the `-supply_map` option of the `apply_power_model` command:

[FATAL] `Invalid supply set pair provided in -supply_map of apply_power_model command (<power_model_name>)`

Potential Issues

The `-supply_map` option does not have two values specified for it.

Consequence of Not Fixing

The relevant checks are not performed properly for the UPF command.

How to Debug and Fix

To fix this violation, always specify both lower scope handle and upper scope supply set in the `-supply_map` option of the `apply_power_model` command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
apply_power_model PM1 -supply_map {LH1} ' Only one value specified in apply_power_model command
```

For the above UPF snippet, the *UPFSEM_36* rule reports the following violation because an invalid supply set pair is specified in the –

supply_map option of the apply_power_model command:

```
Invalid supply set pair provided in -supply_map of
apply_power_model command (PM1)
```

Default Severity Label

Fatal

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_37

Checks if an invalid value is specified in the set_partial_on_translation command

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_37* rule reports a violation when an invalid value is specified in the `set_partial_on_translation` command.

Messages and Suggested Fix

The following message appears when an invalid value is specified in the `set_partial_on_translation` command:

[FATAL] `Invalid argument <argument value> provided in command (set_partial_on_translation)`

Potential Issues

The `set_partial_on_translation` command does not have a valid value.

Consequence of Not Fixing

The relevant checks are not performed properly for the UPF command.

How to Debug and Fix

To fix this violation, always specify either "OFF" or "FULL_ON" in the `set_partial_on_translation` command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_partial_on_translation FULL_OFF
```

For the above UPF snippet, the *UPFSEM_37* rule reports the following violation because an when invalid value, `FULL_OFF`, is specified in the `set_partial_on_translation` command:

```
Invalid argument (FULL_OFF) provided in command
(set_partial_on_translation)
```

Default Severity Label

Fatal

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_38

Checks if an invalid supply set handle is provided in the add_power_state command

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_38* rule reports a violation when a supply set handle provided in the `add_power_state` command is not valid. Some of the valid handles are power, ground, pwell, or nwell.

Messages and Suggested Fix

The following message appears when supply set handle provided in the `add_power_state` command is not valid:

[FATAL] `Invalid supply set handle '<handle-name>' specified in command add_power_state for supply set '<supply-set-name>'`

Potential Issues

An invalid value is provided in the `add_power_state` command as a supply set handle.

Consequence of Not Fixing

The power intent cannot be analyzed thus no power verification is performed.

How to Debug and Fix

Make sure that a valid supply set handle/function is provided in the `add_power_state` command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
upf_version 2.0
set_design_top top

create_supply_port VDD1
create_supply_port VSS
```

```
create_supply_net VDD1
create_supply_net VSS

create_supply_set ss1 -function {power VDD1} -function {pwell
VSS}

connect_supply_net VDD1 -ports {VDD1}
connect_supply_net VSS -ports {VSS}

add_port_state VDD1 -state {NOM 0.9}
add_port_state VSS -state {NOM 0.0}


create_power_domain TOP -include_scope -supply {primary ss1}

add_power_state ss1 -state NOM {-simstate NORMAL -supply_expr
{ss2.power == NOM1 && pwell == `{NOM , 0.0} } }
```

For the above UPF snippet, the *UPFSEM_38* rule reports the following violation because the supply set handle `ss2.power` specified for supply set `ss1` is not valid:

```
Invalid supply set handle 'ss2.power' specified in command
add_power_state for supply set 'ss1'
```

Default Severity Label

Fatal

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSEM_39

Checks if UPF versions is specified as 2.1 in the upf_version command

This rule is obsolete and will be removed in a future release, as UPF version 2.1 is now supported.

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_39* rule reports a violation when the UPF version specified to in the `upf_version` command is 2.1, which is not supported in SpyGlass. To allow specific 2.1 commands, set the `upf_version` to 2.0 and set the *lp_allow_UPF_21_commands_options* to `yes`.

Messages and Suggested Fix

The following message appears when an UPF version is specified as 2.1:

```
[FATAL] UPF version given as upf_version=2.1 is not supported. To allow specific 2.1 commands, please set the upf_version to 2.0 and set the parameter 'lp_allow_UPF_21_commands_options' to yes. Please refer the documentation of this parameter
```

Potential Issues

Unsupported version 2.1 is specified in the `upf_version` command.

Consequence of Not Fixing

SpyGlass will assume the version as 1.0 and continue the execution.

How to Debug and Fix

Set `upf_version` command to 2.0 and set the `lp_allow_UPF_21_commands_options` parameter to `yes`.

Example Code and/or Schematic

Consider the following UPF snippet:

```
upf_version 2.1
```

For the above UPF snippet, the *UPFSEM_39* rule reports the following violation message because the UPF version is specified as 2.1:

```
UPF version given as upf_version=2.1 is not supported. To allow
specific 2.1 commands, please set the upf_version to 2.0 and
set the parameter 'lp_allow_UPF_21_commands_options' to yes.
Please refer the documentation of this parameter
```

Default Severity Label

Fatal

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_40

Ensures that no UPF command is used in the sourced Tcl file

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_40* rule reports if any UPF command has been used in sourced Tcl file, using `source` Tcl command from a UPF file.

Parameters

lp_report_upf_command_in_source_file: Default value is no. Set the value of this parameter to `yes` to report the UPF command used in the sourced TCL file.

Messages and Suggested Fix

The following message appears when any UPF command has been used in sourced Tcl file using `source` Tcl command:

[ERROR] UPF command '<command-name>' has been used in sourced TCL file. Please use 'load_upf' command for UPF commands

Potential Issues

The violation message explicitly states the potential issues.

Consequence of Not Fixing

None

How to Debug and Fix

Make sure that the `load_upf` command is used to load UPF commands file.

Example Code and/or Schematic

Consider the top level UPF file *const.upf*, as follows:

```
upf_version 2.0

source upf_files/temp.tcl
```

```
load_upf upf_files/const1.upf

create_power_domain TOP
create_power_domain PD1_domain -elements " ${VAR} "
create_power_domain PD2_domain -elements " /u_PD2 "
create_power_domain VD_domain  -elements " /u_VD  "

create_supply_port top_supply -domain TOP -direction in
create_supply_port PD1_domain_supply  -domain TOP -direction
in
create_supply_port PD2_domain_supply -domain TOP -direction
in
```

In the above example, the `temp.tcl` file has been sourced. And a UPF file `const1.upf` has been sourced using the `load_upf` command. Therefore any UPF commands used in the `temp.tcl` file should get reported.

Consider the *temp.tcl* file, as follows:

```
create_power_domain PD3_domain -elements " /u_PD2
"create_supply_net VDD -domain PD3_domain
create_supply_port VDD -domain PD3_domain

connect_supply_net VDD -ports {VDD}

set_related_supply_net -power VDD -object_list {inst1/VDD}
```

The following violation message is reported because commands are used in the sourced `temp.tcl` file:

```
UPF command 'create_power_domain' has been used in sourced TCL
file. Please use 'load_upf' command for UPF commands
```

Default Severity Label

Error

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSEM_41

Checks the missing ack_port argument in the create_power_switch UPF command

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_41* rule reports a violation if the `create_power_switch` UPF command does not have `ack_port` argument specified.

Parameters

lp_report_missing_ack_port_option: Default value is no. Set the value of this parameter to `yes` to report the missing `ack_port` argument in the `create_power_switch` command.

Messages and Suggested Fix

The following message appears when the `create_power_switch` UPF command does not have `ack_port` argument specified:

[WARNING] `Missing 'ack_port' specification in command create_power_switch '<command-name>`

Potential Issues

The violation message explicitly states the potential issues.

Consequence of Not Fixing

None

How to Debug and Fix

Make sure that the ack_port argument has been specified in the `create_power_switch` command.

Example Code and/or Schematic

Consider the following snippet:

```
create_power_switch psw3 \
-domain VD0 \
```

```
-input_supply_port {VDD VDD1} \
-output_supply_port {VDDG VDDX} \
-control_port {a en1}\
 -control_port {b en2} \
 -on_state {on1 VDD {en1}}\
 -off_state {off1 {!en1}}
```

The following violation message is reported because the `create_power_switch` UPF command does not have the `ack_port` argument specified:

```
Missing 'ack_port' specification in command create_power_switch
'psw3'
```

Default Severity Label

Warning

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSEM_42

Checks for the command specified in the disallow_upf_command constraint

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_42* reports a violation when the command specified in the *disallow_upf_command* SGDC constraint has been used in the UPF.

Messages and Suggested Fix

The following message appears when the command specified in the *disallow_upf_command* SGDC constraint has been used in the UPF:

`[FATAL] Command ( <command-name> ) not supported`

Potential Issues

The specified command is mentioned in the *disallow_upf_command* SGDC constraint.

Consequence of Not Fixing

The relevant checks are not performed properly for the UPF command.

How to Debug and Fix

To fix this violation, do not use the command specified in the *disallow_upf_command* SGDC constraint.

Example Code and/or Schematic

Consider the following example:

SGDC:

```
disallow_upf_command -name map_level_shifter_cell
```

UPF:

```
set_level_shifter LS1 -domain UA -applies_to outputs -
location fanout -applies_to inputs  -transitive -force_shift
map_level_shifter_cell LS1 -lib_cells BASIC_LS -domain UA
```

For the above example, the *UPFSEM_42* rule reports the following violation

because the `map_level_shifter_cell` command is used in the UPF, which is also mentioned in the `disallow_upf_command` SGDC constraint:

```
Command ( map_level_shifter_cell ) not supported
```

Default Severity Label

Fatal

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSEM_43

Checks if a command option is specified in the disallow_upf_command constraint

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_43* rule reports a violation when a command option is also specified in the *disallow_upf_command* SGDC constraint.

Messages and Suggested Fix

The following message appears when a command option is also specified in the *disallow_upf_command* SGDC constraint:

[WARNING] Option (<option-name>) not supported for command (<command-name>)

Potential Issues

The specified option in a command is disallowed using the *disallow_upf_command* SGDC constraint.

Consequence of Not Fixing

The option is ignored and does not have any impact.

How to Debug and Fix

To fix this violation, do not use the option that is specified in the *disallow_upf_command* SGDC constraint.

Example Code and/or Schematic

Consider the following example:

SGDC:

```
disallow_upf_command -name set_level_shifter -options
location  transitive
```

UPF:

```
set_level_shifter LS1 -domain UA -applies_to outputs -
applies_to inputs  -transitive -force_shift
```

For the above example, the *UPFSEM_43* rule reports the following violation because the `transitive` option is disallowed in SGDC for the `set_level_shifter` command:

```
Option ( -transitive ) not supported for command ( set_level_shifter )
```

Default Severity Label

Warning

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSEM_44

Reports if multiple elements are specified in the create_power_domain command

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_44* rule reports a violation when multiple elements are specified in the *create_power_domain* command, if the *lp_flag_multi_elements_in_create_power_domain* parameter is set to `yes`.

Parameters

lp_flag_multi_elements_in_create_power_domain: Default value is `no`. Set this parameter to `yes` to report if multiple elements have been specified in the `create_power_domain` UPF command.

Messages and Suggested Fix

The following message appears when if multiple elements are specified in the `create_power_domain` command:

[ERROR] `Multiple elements '<element-list>' are specified in command 'create_power_domain'`

Potential Issue

None

Consequence of Not Fixing

None

How to Debug and Fix

Provide only single element in the *create_power_domain* command to avoid this violation.

Example Code and/or Schematic

Consider the following example:

```
create_power_domain VD1 -elements {inst1/and1 inst2/and1
```

```
inst3/and1 inst4/and1 inst5/and1}
```

For the above example, the *UPFSEM_44* rule reports the following violation because multiple elements are specified in the *create_power_domain* command:

```
Multiple elements 'inst1/and1,inst2/and1,inst3/and1,inst4/
and1,inst5/and1' are specified in command 'create_power_domain'
```

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_45

Checks if the same net is specified as power and ground in the set_isolation command

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_45* rule reports a violation if the same net is specified in both `isolation_power_net` and `isolation_ground_net` in the same *set_isolation* command.

Messages and Suggested Fix

The following message appears when the same net is specified in both `isolation_power_net` and `isolation_ground_net`:

[ERROR] Same net (<net-name>) specified as power and ground in set_isolation command (<command-name>)

Potential Issue

The same net will be used as power and ground net for isolation cells.

Consequence of Not Fixing

This can lead to unexpected functioning of isolation cell resulting in design failure.

How to Debug and Fix

The violation message points to the UPF file and corresponding line number where the *set_isolation* command has the same net specified as both power and ground in the `isolation_power_net` and `isolation_ground_net` arguments.

Example Code and/or Schematic

Consider the following example:

```
set_isolation ISO1 –domain PD1 –isolation_power_net VDD –
isolation_ground_net VDD
```

For the above example, the *UPFSEM_45* rule reports the following violation

because the same net VDD is specified in both `isolation_power_net` and `isolation_ground_net` arguments:

```
Same net (VDD) specified as power and ground in set_isolation
command (ISO1)
```

Default Severity Label

Error

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSEM_46

Checks supply set function and pg_type of the ports given in the -connect option of the connect_supply_set command

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_46* rule reports a violation if there is mismatch between supply set function and pg_type of the ports given in -connect option of *connect_supply_set* command.

The following table shows the scenarios in which a violation is reported:

- YES: Violation
- NO: No Violation as these are expected cases

Ports	power	ground	pweel	nweel	deeppwell	deepnwell
primary_power	NO	YES	YES	YES	YES	YES
backup_power	NO	YES	YES	YES	YES	YES
internal_power	NO	YES	YES	YES	YES	YES
primary_ground	YES	NO	YES	YES	YES	YES
backup_ground	YES	NO	YES	YES	YES	YES
internal_ground	YES	NO	YES	YES	YES	YES
pwell	YES	YES	NO	YES	YES	YES
nwell	YES	YES	YES	NO	YES	YES
deeppwell	YES	YES	YES	YES	NO	YES
deepnwell	YES	YES	YES	YES	YES	NO

Parameters

None

Messages and Suggested Fix

The following message appears if incorrect pair of supply set function and pg_type are given in the *connect_supply_set* command:

[ERROR] Incompatible pair of supply set function <function_name> and pg_type <pg_type> specified in (-connect) argument of command 'connect_supply_set'

Potential Issue

Incorrect connection created in the design.

Consequence of Not Fixing

The PG connections of various cells will not be proper, which may lead to design failure.

How to Debug and Fix

The violation message points to the UPF file and corresponding line number where the *connect_supply_set* command with incorrect pair of function and pg_type is written. Review this and correct the command.

Example Code and/or Schematic

Consider the following example:

```
connect_supply_set ss2 -connect {power backup_ground} -
connect {ground primary_ground} -elements FF1
```

For the above example, the *UPFSEM_46* rule reports the following violation:

```
Incompatible pair of supply set function 'power' and pg_type
'backup_ground' specified in ( -connect ) argument of command
'connect_supply_set'
```

Default Severity Label

Error

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_47

Checks for unsupported versions of UPF

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_47* rule reports a violation when an invalid UPF version is used either in the `power_data` SGDC constraint or in the `load_power_data` project file command.

Parameters

None

Messages and Suggested Fix

The following message appears when an invalid UPF version is used:

[FATAL] `Project file with unsupported version ( <unsupported-version> )`

or

[FATAL] `Project file with unsupported version ( <unsupported-version> )`

Potential Issue

The violation message explicitly states the potential issues.

Consequence of Not Fixing

SpyGlass cannot determine which version of UPF is to be used.

How to Debug and Fix

To fix this violation, specify a supported version of UPF (1.0, 2.0 or 2.1).

Example Code and/or Schematic

Consider the following example:

```
power_data -file "test.upf" -version "2.2" -format "upf"
```

For the above example, the *UPFSEM_47* rule reports the following violation

because 2.2 is not a supported version of UPF:

```
SGDC file power_data constraint with unsupported version ( 2.2 )
```

Default Severity Label

Fatal

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_48

Checks for the use of deprecated UPF commands

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_48* rule reports a violation when a deprecated UPF command is used.
Several commands have been deprecated in UPF version 2.1, including:

- add_domain_elements
- map_isolation_cell
- map_level_shifter_cell
- set_isolation_control
- set_pin_related_supply
- set_retention_control

NOTE: *Refer to the IEEE 1801-2013 "Standard for Design and Verification of Low-Power Integrated Circuits" Appendix D for more details.*

Parameters

None

Messages and Suggested Fix

The following message appears when a deprecated UPF command is used:

```
[WARNING] The command ( <deprecated-command> ) is deprecated.
<suggested-replacement-if-applicable>
```

Potential Issue

The deprecated commands still operate. However their intended replacements should be preferred since they are more general and allow more progressive refinement.

Consequence of Not Fixing

Deprecated commands may be removed from a future version of SpyGlass.

How to Debug and Fix

To fix this violation, use the recommended replacement command.

Example Code and/or Schematic

Consider the following example:

SGDC:

```
power_data -file "test.upf" -version "2.1" -format "upf"
```

UPF:

```
set_isolation iso1 -domain PD1 -applies_to both -
isolation_power_net VDD -isolation_ground_net VSS -
clamp_value 0 -location parent
set_isolation_control iso1 -domain PD1 -isolation_signal iso
-isolation_sense low
```

For the above example, the *UPFSEM_48* rule reports the following violation because the `set_isolation` command is a deprecated command in the UPF version 2.1:

```
The command ( set_isolation_control ) is deprecated. Please use
"set_isolation [-update]" instead.
```

Default Severity Label

Warning

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSEM_49

Checks for the use of deprecated UPF command options

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_49* rule reports a violation when a deprecated UPF command option is used. In the UPF version 2.1, numerous command options have been deprecated.

Command	Deprecated option(s)
connect_supply_net	-pins *pin_list*, -rail_connection *rail_type*
create_power_domain	-include_scope, -scope *scope*
map_power_switch	-domain *domain_name*
save_upf	-version *string*
set_isolation	-location fanin | faninout | sibling, -clamp_value any, -sink_off_clamp {<0 | 1 | any | Z | latch | value> [simstate_list]} -source_off_clamp {<0 | 1 | any | Z | latch | value> [simstate_list]} -transitive [<TRUE | FALSE>]
set_level_shifter	-location fanin | faninout | sibling, -transitive [<TRUE | FALSE>], -threshold *list*
set_partial_on_translation	-full_on_tools {string_list}, -off_tools {string_list}
set_port_attributes	-domains domain_list [-applies_to <inputs | outputs | both>], exclude_domains domain_list [-applies_to <inputs | outputs | both>], -repeater_supply supply_set_ref, -transitive [<TRUE | FALSE>]
set_retention_elements	-expand [<TRUE | FALSE>]

NOTE: *Refer to the IEEE 1801-2013 "Standard for Design and Verification of Low-Power Integrated Circuits" Appendix D for more details.*

Parameters

None

Messages and Suggested Fix

The following message appears when a deprecated UPF command option is used:

[WARNING] The option (<deprecated option>) of command (<command-name>) is deprecated. <Suggested replacement if applicable>

Potential Issue

Most of the options have been deprecated to improve clarity or because the old values did not make sense.

Consequence of Not Fixing

Deprecated command options may be removed from a future version of SpyGlass.

How to Debug and Fix

To fix this violation, in most cases it is necessary to replace the command/option with a different construct. In a few cases it is possible to modify the UPF as suggested in the warning message.

Example Code and/or Schematic

Consider the following example:

SGDC:

```
power_data -file "test.upf" -version "2.1" -format "upf"
```

UPF:

```
create_power_switch psw1 -domain PD -output_supply_port
{VDDG VPDY} -ack_port {S
LEEPOUT VDDX} -ack_delay {VDDP 8.0}
map_power_switch psw1 -domain PD2 -lib_cells {PS1}
```

For the above example, the *UPFSEM_49* rule reports the following violation

because “-domain” is a deprecated option of the command “map_power_switch” in UPF 2.1:

```
The option ( -domain ) of command ( map_power_switch ) is
deprecated.
```

Default Severity Label

Warning

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_50

Checks for modification of power states that have been marked as complete

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_50* rule reports a violation when there is an attempt to update the power states of an object that has already been flagged as complete:

Parameters

None

Messages and Suggested Fix

The following message appears when trying to add power states to an object flagged as complete:

[FATAL] `add_power_state called after -complete has been specified`

Potential Issue

Power state tables that have been marked complete cannot be modified. Commands that attempt to modify a complete power state table have no effect.

Consequence of Not Fixing

The power state table may not be properly defined and may be missing states.

How to Debug and Fix

To fix this violation, ensure that `-complete` is used only on the last use of the `add_power_state` command for an object.

Example Code and/or Schematic

Consider the following example:

UPF:

```
upf_version 2.1
add_power_state PDtop -state {ON -supply_expr {(power ==
{FULL_ON 0.8}) && (ground == {FULL_ON 0})}}
add_power_state PDtop -domain -complete
add_power_state PDtop -domain -state {OFF -supply_expr
{(power == {FULL_OFF off}) && (ground == {FULL_OFF off})}}
```

For the above example, the *UPFSEM_50* rule reports the following violation because the power states for domain `PDtop` are modified after being flagged as complete:

```
add_power_state called after -complete has been specified.
```

Default Severity Label

Fatal

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSEM_51

Checks for power states where the named object does not match the specified –supply or –domain option

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_51* rule reports a violation when the `add_power_state` command is used with either:

- supply and an object that is a power domain
- domain and an object that is a supply set handle

Parameters

None

Messages and Suggested Fix

Message 1

The following message appears when the object is not a supply set handle:

[FATAL] `Command add_power_state used with option ( -supply ) but a power domain ( <power-domain-name> ) was detected`

Message 2

The following message appears when the object is not a power domain:

[FATAL] `add_power_state used with option ( -domain ) but a supply set ( <supply-set-handle> ) was detected`

Potential Issue

The UPF version 2.1 allows you to explicitly specify whether power states apply to a power domain or to a supply set handle. In previous UPF versions, the type was implied from the provided named object.

Consequence of Not Fixing

Potentially, the power state tables may not be properly defines, with states being added to the wrong named object

How to Debug and Fix

To fix this violation, ensure that `-supply` or `-domain` applies to an object of the correct type.

Example Code and/or Schematic

Consider the following example:

UPF:

```
upf_version 2.1
create_supply_set top_SS -function {power vdd} -function
{ground vss}
associate_supply_set top_SS -handle PDtop.primary
add_power_state PDtop.primary -domain -state {ON -
supply_expr {(power == {FULL_ON 0.8}) && (ground == {FULL_ON
0})}}
```

For the above example, the *UPFSEM_51* rule reports the following violation because the `-domain` option of `add_power_state` is used with `PDtop.primary`, which is a supply set handle:

```
Command add_power_state used with option ( -domain ) but a
supply set ( PDtop.primary ) was detected
```

Default Severity Label

Fatal

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSEM_52

Checks that the named object already exists when the –update option is used

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_52* rule reports a violation when there is an attempt to use a progressive refinement command using `–update` that refers to an object that has not yet been created.

Parameters

None

Messages and Suggested Fix

The following message appears when an invalid UPF version is used:

[FATAL] `Named object ( <object-name> ) not found for command ( <command-name> ) with -update`

Potential Issue

Since the object being updated does not exist, the update cannot be applied and the effect of the update is lost.

Consequence of Not Fixing

The UPF is not properly formed.

How to Debug and Fix

Ensure that the named object has been created before it is modified with –update.

Example Code and/or Schematic

Consider the following example:

UPF:

```
upf_version 2.1
```

```
# no previous call to create_power_switch for PS_rule1
create_power_switch PS_rule1 -update -instance { /top/ua/U1}
```

For the above example, the *UPFSEM_52* rule reports the following violation because power switch *PS_Rule1* does not exist when the update option is applied to it:

```
Named object ( PS_rule1 ) not found for command (
create_power_switch ) with –update
```

Default Severity Label

Fatal

Rule Group

UPF Check

Reports and Related Files

None

UPFSEM_54

Checks for conflicting values specified for an argument of a command specified along with the -update argument

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_54* rule reports a violation if conflicting or contradicting values are specified for an argument of the `set_isolation`, `set_level_shifter`, and the `set_retention` commands along with the `-update` argument.

Parameters

None

Messages and Suggested Fix

The following message appears when an invalid UPF version is used:

[ERROR] `Invalid argument <argument-name> specified in upf command <command-name>`

Potential Issue

The value specified with an argument of the `set_isolation`, `set_level_shifter`, or the `set_retention` command contradicts the previously specified value of the argument.

Consequence of Not Fixing

The UPF is not properly formed.

How to Debug and Fix

Ensure that the specified value of the argument is not contradicting the previously specified value of the argument.

Example Code and/or Schematic

Consider the following example:

UPF:

```
set_level_shifter ls1 -domain TOP -elements {in1 en1 }
-applies_to inputs
set_level_shifter ls1 -domain TOP -elements {out1 out2}
-update -applies_to outputs
```

In the above example, the *UPFSEM_54* rule reports the following violation because the `-applies_to` argument contradicts the previously defined value of the argument.

```
Invalid argument <applies_to> specified in upf command
<set_level_shifter>
```

Default Severity Label

Error

Rule Group

`UPF Check`

Reports and Related Files

No

UPFSEM_55

Checks if the same port name occurs in the port_list of multiple connect_logic_net commands with different net_name arguments

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSEM_55* rule reports a violation if a logic port already connected to some logic net is reconnected to different logic net.

NOTE: *The create_logic_net and set_isolation/set_retention/create_power_switch are order dependent. If the set_isolation/set_retention/create_power_switch commands are specified before the create_logic_net command, rule checking may produce incorrect results.*

NOTE: *It is expected that net/port names to have simple names. The rule does not support hierarchical or bus select type or escaped names. Names must not contain the '[' ']' ':' characters. If these characters are specified in the created names, violation will not be flagged.*

Parameters

None

Messages and Suggested Fix

The following message appears if a logic port already connected to some logic net is reconnected to different logic net:

[ERROR] Logic port <port name> previously connected to logic net <net name> through connect_logic_net command

Potential Issue

A logic port already connected to some logic net is reconnected to different logic net.

Consequence of Not Fixing

The command is ignored and the UPF is not properly formed.

How to Debug and Fix

Ensure that the logic ports are connected correctly.

Example Code and/or Schematic

Consider the following example:

UPF:

```
Connect_logic_net net1 -ports {p1}
Connect_logic_net net2 -ports {p1}
```

In the above example, the first `connect_logic_net` command connects the `p1` logic port to the `net1` logic net. In addition, the second `connect_logic_net` command tries to connect the `p1` logic port to the `net2` logic net. In this case, the *UPFSEM_55* rule reports the following violation:

```
Logic port p1 previously connected to logic net net1 through
connect_logic_net command
```

Default Severity Label

Error

Rule Group

`UPF Check`

Reports and Related Files

No

UPFSTX_1

Checks if an invalid port state is specified

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_1* rule reports a violation when an invalid port state is specified.

Messages and Suggested Fix

The following message appears when an invalid port state <*state-name*> is specified:

[SYNTAX] Invalid state specification for state <state-name>

Potential Issues

An invalid or no value is specified for the state of `add_port_state` command.

Consequence of Not Fixing

The power intent cannot be analyzed, thus no power verification is performed.

How to Debug and Fix

Specify valid values in the state using the `add_port_state <state_name value {nom | min max | min nom max | off}>` command, where min<nom<max.

Example Code and/or Schematic

Consider the following UPF snippet:

```
add_port_state mod1supply -state {on_state } -state
{off_state off}
```

For the above UPF snippet, the *UPFSTX_1* rule reports the following violation because no value is specified for the `on_state` state in the `add_port_state` command:

```
Invalid state specification for state ( on_state  )
```

To resolve the violation, specify valid value with the state option in the `add_port_state` command, as shown below:

```
add_port_state mod1supply -state {on_state 0.88 }
```

Default Severity Label

Syntax

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSTX_2

Checks if an state value is specified

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_2* rule reports a violation when an invalid state value is specified.

Messages and Suggested Fix

The following message appears when an invalid state value <*state-value*> is specified:

[SYNTAX] `Invalid state value <state-value> specified at state <state-name>`

Potential Issues

An invalid value is specified for the state of the `add_port_state` command.

Consequence of Not Fixing

The power intent cannot be analyzed, thus no power verification is performed.

How to Debug and Fix

Specify valid values in the state using the `add_port_state <state_name value {nom | min max | min nom max | off}>` command, where min<nom<max.

Example Code and/or Schematic

Consider the following UPF snippet:

```
add_port_state vnn_top -state {vnn_top_on 0.7 0.5 0.9}
```

For the above UPF snippet, the *UPFSTX_2* rule reports the following violation because invalid values are specified for the `on_state` state in the `add_port_state` command:

```
Invalid state value ( 0.5 ) specified at state ( vnn_top_on )
```

To resolve the violation, specify valid values that satisfy min<nom<max argument, as shown below:

```
add_port_state vnn_top -state {on_state 0.5 0.7 0.9}
```

Default Severity Label

Syntax

Rule Group

UPF Check

Reports and Related Files

None

UPFSTX_4

Checks if pg type is simultaneously specified with mutually exclusive port/pin

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_4* rule reports a violation when a pg type is simultaneously specified with mutually exclusive port/pin.

Messages and Suggested Fix

The following message appears when a pg type is simultaneously specified with mutually exclusive port/pin:

[SYNTAX] `Invalid use of pg type in connection specification for net <net-name>`

Potential Issues

Both, pg_type and pin/port arguments are used in the `connect_supply_net` command simultaneously.

Consequence of Not Fixing

The power intent cannot be analyzed, thus no power verification is performed.

How to Debug and Fix

Use only pg_type or pin/port argument at a time when making connection for the nets.

Example Code and/or Schematic

Consider the following UPF snippet:

```
connect_supply_net gnd -pins {ground} -pg_type ground
```

For the above UPF snippet, the *UPFSTX_4* rule reports the following violation because both pg_type and pin arguments are specified in the `connect_supply_net` command:

```
Invalid use of pg type in connection specification for net (
```

```
gnd )
```

To resolve the violation, specify only pin or pg_type argument at a time.

Default Severity Label

Syntax

Rule Group

UPF Check

Reports and Related Files

None

UPFSTX_5

Checks if an invalid HDL type is specified in the definition of vct

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_5* rule reports a violation when an invalid HDL type is specified in the definition of vct.

Messages and Suggested Fix

The following message appears when an invalid HDL type is specified in the definition of the vct <*vct-name*>:

[SYNTAX] Invalid hdl type <hdl-type-name> specified at the definition of vct <vct-name>

Potential Issues

Wrong HDL type is specified for the vct.

Consequence of Not Fixing

The power intent cannot be analyzed, thus no power verification is performed.

How to Debug and Fix

Specify the currently available HDL types using the `-hdl_type <vhdl|sv type_name>` command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_hdl2upf_vct vlog2upf_onoffstate -hdl_type {ve
std_logic} -table {{OFF 'X'} {ON '1'}}
```

For the above UPF snippet, the *UPFSTX_5* rule reports the following violation because *ve* is not a valid HDL type:

```
Invalid hdl type ( ve ) specified at the definition of vct (
vlog2upf_onoffstate )
```

To resolve the violation, specify a valid HDL type for the `-hdl_type` argument, as shown below:

```
create_hdl2upf_vct vlog2upf_onoffstate -hdl_type {vhdl std_logic}
```

Default Severity Label

Syntax

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSTX_6

Checks if an invalid UPF value is specified in the definition of vct

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_6* rule reports a violation when an invalid UPF value is specified in the definition of vct.

Messages and Suggested Fix

The following message appears when an invalid UPF value is specified in the definition of vct <*vct-name*>:

[SYNTAX] Invalid upf value provided in the definition of the vct <vct-name>

Potential Issues

Wrong values are specified in the vct table instead of the predefined values.

Consequence of Not Fixing

The power intent cannot be analyzed, thus no power verification is performed.

How to Debug and Fix

Add predefined values for table values using the `-table {<from value> <to value>}` command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_upf2hdl_vct xx -hdl_type {sv reg} -table {{x c}}
```

For the above UPF snippet, the *UPFSTX_6* rule reports the following violation because undefined values {x c} are specified in the `-table` argument of the `create_upf2hdl_vct` command:

```
Invalid upf value provided in the definition of the vct ( xx )
```

To resolve the violation, specify valid values for the `-table` option of the `create_hdl2upf_vct` command, as shown below:

```
create_hdl2upf_vct vlog2upf_vs -hdl_type {sv reg} \
-table {{X OFF} {0 FULL_ON} {1 OFF} {Z PARTIAL_ON}}
```

Default Severity Label

Syntax

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSTX_13

Checks if an invalid signal type is specified in retention strategy

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_13* rule reports a violation when an invalid signal type is specified in the retention strategy.

Messages and Suggested Fix

The following message appears when an invalid signal type is specified in the retention strategy *<retention-strategy-name>*:

[SYNTAX] Invalid signal specified at <location-name> in retention strategy <retention-strategy-name>

Potential Issues

An invalid value is specified for the sensitivity of retention save_signal or restore_signal.

Consequence of Not Fixing

The power intent cannot be analyzed, thus no power verification is performed.

How to Debug and Fix

Use valid values for the sensitivity of the retention save_signal and restore_signal using the `[-save_signal {{logic_net <high | low | posedge | negedge>}} -restore_signal {{logic_net <high | low | posedge | negedge>}}]` command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_retention ret1 -domain PD -retention_power_net VPD   -save_signal {iso x}
```

For the above UPF snippet, the *UPFSTX_13* rule reports the following

violation because invalid value `x` has been specified for save_signal:

```
Invalid signal specified at -save_signal in retention strategy
ret1
```

To resolve the violation, specify valid value for the sensitivity of isolation signal, as shown below:

```
set_retention ret1 -domain PD \
-retention_power_net VPD  \
-save_signal {iso high}
```

Default Severity Label

Syntax

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSTX_15

Checks if an invalid switch is specified for command

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_15* rule reports a violation when an invalid switch is specified for a UPF command.

Messages and Suggested Fix

The following message appears when an invalid switch is specified for a command <*command*>:

[SYNTAX] Invalid argument <argument-name> specified in upf command <command>

Potential Issues

An invalid argument is specified in the UPF command.

Consequence of Not Fixing

The power intent cannot be analyzed, thus no power verification is performed.

How to Debug and Fix

Specify a valid argument for the UPF command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_isolation iso1 -domain PD2 -isolation_power_net VPD2 -
clamp_value 0 arg
```

For the above UPF snippet, the *UPFSTX_15* rule reports the following violation because argument `arg` is not a valid argument in the `set_isolation` command:

```
Invalid argument ( arg ) specified in upf command (
set_isolation )
```

To resolve the violation, specify only valid arguments with the

`set_isolation` command.

Default Severity Label

Syntax

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSTX_16

Checks if no value is specified for argument

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_16* rule reports a violation when no value is specified for an argument in a UPF command.

Messages and Suggested Fix

The following message appears when no value is specified for an argument *<argument-name>* in command *<command>*:

[SYNTAX] Invalid or no value provided for argument <argument-name> in command <command>

Potential Issues

An invalid or no value is specified for an argument in a UPF command.

Consequence of Not Fixing

The power intent cannot be analyzed, thus no power verification is performed.

How to Debug and Fix

Specify a valid value for the argument in the UPF command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_isolation_control iso1 -domain pd_top -isolation_signal
iso -isolation_sense
```

For the above UPF snippet, the *UPFSTX_16* rule reports the following violation because no value is specified for the `isolation_sense` argument:

```
Invalid or no value provided for argument ( -isolation_sense )
in command ( set_isolation_control )
```

To resolve the violation, specify a valid value for the `isolation_sense`

argument, as shown below:

```
set_isolation_control iso1 -domain pd_top -isolation_signal
iso -isolation_sense low
```

Default Severity Label

Syntax

Rule Group

UPF Check

Reports and Related Files

None

UPFSTX_17

Checks if an invalid value type is specified for argument

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_17* rule reports a violation when an invalid value type is specified for an argument in a UPF command.

Messages and Suggested Fix

The following message appears when an invalid value type is specified for an argument *<argument-name>* in a command *<command>*:

[SYNTAX] `Invalid enum value <enum-value> specified for argument <argument-name> in command <command>`

Potential Issues

An invalid enum value is specified for an argument in a UPF command.

Consequence of Not Fixing

The power intent cannot be analyzed, thus no power verification is performed.

How to Debug and Fix

Specify a valid enum value for the argument.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_isolation iso1 -domain PD2  -isolation_power_net VPD2 -applies_to x
```

For the above UPF snippet, the *UPFSTX_17* rule reports the following violation because invalid enum value `x` is specified for the `–applies_to` argument:

```
Invalid enum value ( x ) specified for argument ( -applies_to ) in command ( set_isolation )
```

To resolve the violation, specify a valid enum value for the `–applies_to`

argument, as shown below:

```
set_isolation iso1 -domain PD2  -isolation_power_net VPD2 -
applies_to outputs
```

Default Severity Label

Syntax

Rule Group

UPF Check

Reports and Related Files

None

UPFSTX_18

Checks if a mandatory argument is missing

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_18* rule reports a violation when a mandatory argument is missing in a UPF command.

Messages and Suggested Fix

The following message appears when a mandatory argument *<argument-name>* is missing in a UPF command *<command>*:

[SYNTAX] Required argument <argument-name> not specified in command <command>

Potential Issues

A mandatory argument is not specified for a UPF command.

Consequence of Not Fixing

The power intent cannot be analyzed, thus no power verification is performed.

How to Debug and Fix

Specify the mandatory arguments for the UPF command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_isolation iso1   -isolation_power_net VPD2 -applies_to
outputs
```

For the above UPF snippet, the *UPFSTX_18* rule reports the following violation because the mandatory `-domain` argument is missing for the `set_isolation` command:

```
Required argument ( -domain ) not specified in command (
set_isolation
```

To resolve the violation, specify the mandatory `–domain` argument for the `set_isolation` command, as shown below:

```
set_isolation iso1 –domain PD1  -isolation_power_net VPD2 -
applies_to outputs
```

Default Severity Label

Syntax

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSTX_19

Checks the presence of an invalid PST state specification

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_19* rule reports a violation when an invalid PST state is specified.

Messages and Suggested Fix

The following message appears when an invalid state <*state-name*> in PST <*pst-name*> is specified:

[SYNTAX] `Invalid pst state specification for state <state-name> in PST <pst-name>`

Potential Issues

An incorrect PST state is specified.

Consequence of Not Fixing

The power intent cannot be analyzed, thus no power verification is performed.

How to Debug and Fix

Specify a correct PST state for the power state tables.

Example Code and/or Schematic

Consider the following UPF snippet:

```
add_port_state VVD -state {on1 0.6} -state {off1 off}
create_pst ps1 -supplies {VAON VVD VPD VSS}
add_pst_state s1 -pst ps1 -state {on1 on1 on1x on1  def}
```

For the above UPF snippet, the *UPFSTX_19* rule reports the following violation because there are four supplies in the PST ps1 but five states are specified for state s1:

```
Invalid pst state specification for state ( s1 ) in pst ( ps1 )
```

To resolve the violation, specify the PST state s1 correctly with four port

states, as shown below:

```
add_pst_state s1 -pst ps1 -state {on1 on1 on1 def}
```

Default Severity Label

Syntax

Rule Group

UPF Check

Reports and Related Files

None

UPFSTX_20

Checks if an invalid port-net pair is specified

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_20* rule reports a violation when an invalid port-net pair is specified.

Messages and Suggested Fix

The following message appears when an invalid port-net pair *<pair-name>* is specified:

[SYNTAX] `Invalid port net pair <pair-name> specified`

Potential Issues

An invalid port-net pair is specified.

Consequence of Not Fixing

The power intent cannot be analyzed, thus no power verification is performed.

How to Debug and Fix

Specify a valid port-net pair for an argument of the UPF command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
bind_checker bc -module top -ports {{port1 a b}}
```

For the above UPF snippet, the *UPFSTX_20* rule reports the following violation because port-net pair `{{port1 a b}}` specified is not valid:

```
Invalid port net pair (  port1 a b  ) specified
```

To resolve the violation, specify a valid port-net pair, as shown below:

```
bind_checker bc -module top -ports {{port1 a}}
```

Default Severity Label

Syntax

Rule Group

UPF Check

Reports and Related Files

None

UPFSTX_21

Checks if an invalid HDL-UPF value pair is specified

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_21* rule reports a violation when an invalid HDL-UPF value pair is specified.

Messages and Suggested Fix

The following message appears when an invalid HDL-UPF value pair *<pair-name>* is specified:

[SYNTAX] `Invalid hdl upf value pair <pair-name> provided`

Potential Issues

An invalid HDL-UPF value pair is specified.

Consequence of Not Fixing

The power intent cannot be analyzed, thus no power verification is performed.

How to Debug and Fix

Specify a valid HDL-UPF value pair.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_upf2hdl_vct xx -hdl_type {sv} -table {{FULL_ON a b}}
```

For the above UPF snippet, the *UPFSTX_21* rule reports the following violation because an invalid HDL-UPF value pair `{{FULL_ON a b}}` is specified:

```
Invalid hdl upf value pair (  FULL_ON a b  ) provided
```

To resolve the violation, specify a valid HDL-UPF value pair, as shown below:

```
create_upf2hdl_vct xx -hdl_type {sv} -table {{FULL_ON a }}
```

Default Severity Label

Syntax

Rule Group

UPF Check

Reports and Related Files

None

UPFSTX_22

Checks if an invalid port is specified for a power switch

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_22* rule reports a violation when an invalid port is specified for a power switch.

Messages and Suggested Fix

The following message appears when an invalid port is specified for a power switch *<power-switch-name>*:

[SYNTAX] Invalid port specification <port-specification-name> for power switch <power-switch-name> provided

Potential Issues

An invalid input/output supply port is specified for a power switch.

Consequence of Not Fixing

The power intent cannot be analyzed, thus no power verification is performed.

How to Debug and Fix

Specify valid input/output supply ports for power switches, using:

```
–input_supply_port {<port name> <supply net name>}
–output_supply_port {<port name> <supply net name>}
```

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_switch psw1 -domain PD -input_supply_port {VDD
XX 11} -output_supply_port {VDDG VPDY} -ack_port {SLEEPOUT
VDDX} -on_state {on YY}
```

For the above UPF snippet, the *UPFSTX_22* rule reports the following violation because `input_supply_port` specification `{VDD XX 11}` is not valid:

```
Invalid -input_supply_port specification ( VDD XX 11 ) for
power switch ( psw1 )  provided
```

To resolve the violation, specify a valid `input_supply_port`, as shown below:

```
create_power_switch psw1 -domain PD -input_supply_port {VDD
VDDA} -output_supply_port {VDDG VPDY} -ack_port {SLEEPOUT
VDDX} -on_state {on YY}
```

Default Severity Label

Syntax

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSTX_23

Checks if an invalid state is specified for a power switch

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_23* rule reports a violation when an invalid state is specified for a power switch.

Messages and Suggested Fix

The following message appears when an invalid state is specified for a power switch *<power-switch-name>*:

[SYNTAX] Invalid state specification <state-specification-name> for power switch <power-switch-name>

Potential Issues

An invalid or no boolean function is specified for on/off state of a power switch.

Consequence of Not Fixing

The power intent cannot be analyzed, thus no power verification is performed.

How to Debug and Fix

Specify a valid boolean function for on/off state of the power switch.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_switch psw1 -domain PD -input_supply_port {VDD VVD} -output_supply_port {VDDG VPDY} -ack_port {SLEEPOUT VDDX} -control_port {SLEEP in} -on_state {on VDD}
```

For the above UPF snippet, the *UPFSTX_23* rule reports the following violation because no boolean function is specified for `on_state`:

```
Invalid ( -on_state ) state specification : ( on YY ) for power switch ( psw1 )
```

To avoid this violation, specify a valid boolean function, as shown below:

```
create_power_switch psw1 -domain PD -input_supply_port {VDD
VVD} -output_supply_port {VDDG VPDY} -ack_port {SLEEPOUT
VDDX} -control_port {SLEEP in} -on_state {on VDD}
```

Default Severity Label

Syntax

Rule Group

UPF Check

Reports and Related Files

None

UPFSTX_24

Checks if an invalid ack delay value is specified for a power switch

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_24* rule reports a violation when an invalid ack delay value is specified for a power switch.

Messages and Suggested Fix

The following message appears when an invalid ack delay value *<ack-delay-value>* is specified for a power switch *<switch-name>*:

[SYNTAX] Invalid ack delay value specified <ack-delay-value> for switch <switch-name>

Potential Issues

An invalid value is specified for the `ack_delay` argument in the `create_power_switch` UPF command.

Consequence of Not Fixing

No check takes place for the `create_power_switch` command.

How to Debug and Fix

Specify a valid value for the `ack_delay` argument using `–ack_delay {<port-name> delay}` command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_switch psw1 -domain PD \
-input_supply_port {VDD VVD} \
-output_supply_port {VDDG VPDY} \
-ack_port {SLEEPOUT VDDX} \
-control_port {SLEEP} -on_state {on VDD SLEEP} \
-off_state {on1 SLEEP } -ack_delay {SLEEPOUT x}
```

For the above UPF snippet, the *UPFSTX_24* rule reports the following violation because the delay value “x” specified for ack_delay argument is not valid:

```
Invalid ack delay value specified ( SLEEPOUT x ) for switch ( psw1 )
```

To avoid this violation, specify a valid delay value in the `–ack_delay` argument, as shown below:

```
-ack_delay {SLEEPOUT 0.5}
```

Default Severity Label

Syntax

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSTX_25

Checks if an invalid net-edge pair is specified for a retention control

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_25* rule reports a violation when an invalid net-edge pair is specified for a retention control in a power object.

Messages and Suggested Fix

The following message appears when an invalid net-edge pair <*net-edge-pair-name*> is specified for a retention control <*retention-control-name*> in a power object <*power-object-name*>:

[SYNTAX] Invalid net edge pair <net-edge-pair-name> for <retention-control-name> provided in power object <power-object-name>

Potential Issues

An invalid net-edge pair is specified for the `set_retention_control` UPF command.

Consequence of Not Fixing

The power intent cannot be analyzed thus no power verification is performed.

How to Debug and Fix

Specify a valid net-edge pair for the `set_retention_contro` command using:

```
[-save_signal {{logic_net <high | low | posedge | negedge>}}
-restore_signal {{logic_net <high | low | posedge |
negedge>}}]
```

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_retention_control ret1 -domain pd_top -save_signal {en1
posedge && negedge } -restore_signal {en1 negedge}
```

For the above UPF snippet, the *UPFSTX_25* rule reports the following violation because an invalid net-edge pair `{en1 posedge && negedge }` is specified for the `set_retention_control` command:

```
Invalid net edge pair ( en1 posedge && negedge ) for ( -
save_signal ) provided in power object ( ret1 )
```

To resolve the violation, specify valid net-edge pair, as shown below:

```
set_retention_control ret1 -domain pd_top -save_signal {en1
posedge} -restore_signal {en1 negedge}
```

Default Severity Label

Syntax

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSTX_26

Checks the presence of syntax errors

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_26* rule reports a violation when a UPF command is specified with syntax errors.

Messages and Suggested Fix

The following message appears when a UPF command <*command*> is specified with syntax error:

[SYNTAX] `Command <command> ignored due to syntax error(s)`

Potential Issues

A UPF command is specified with syntax errors.

Consequence of Not Fixing

The power intent cannot be analyzed thus no power verification is performed.

How to Debug and Fix

Specify the UPF command without syntax errors.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_isolation_control iso1 -domain pd_top -isolation_signal
iso -isolation_sense
```

For the above UPF snippet, the *UPFSTX_26* rule reports the following violation because no value is specified for the `isolation_sense` argument:

```
Command ( set_isolation_control ) ignored due to syntax
error(s)
```

To resolve the violation, specify a valid value for the `isolation_sense` argument, as shown below:

```
set_isolation_control iso1 -domain pd_top -isolation_signal
iso -isolation_sense low
```

Default Severity Label

Syntax

Rule Group

UPF Check

Reports and Related Files

None

UPFSTX_27

Checks the presence of an hierarchical separator

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_27* rule reports a violation when hierarchical separator is present in a UPF command.

Messages and Suggested Fix

The following message appears when hierarchical separator is present:

`[SYNTAX] Hierarchical character present in <name> of <name>`

Potential Issues

An invalid use of hierarchical character “/” in a UPF command argument.

Consequence of Not Fixing

No check takes place for the UPF command.

How to Debug and Fix

Use the hierarchical separator “/” correctly in the UPF commands.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_supply_port inst1/VVD -domain VD
```

For the above UPF snippet, the *UPFSTX_27* rule reports the following violation because the domain VD is not created under scope `inst1` where “inst1/VVD” is used to create supply port:

```
Hierarchical character present in portName of
create_supply_port
```

To avoid this violation, remove scope from the `create_supply_port` command, as shown below:

```
create_supply_port VVD –domain VD
```

Default Severity Label

Syntax

Rule Group

UPF Check

Reports and Related Files

None

UPFSTX_28

Checks the presence of an input file

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_28* rule reports a violation when the input file is not found.

Messages and Suggested Fix

The following message appears when the input file <*file-name*> is not found:

[SYNTAX] File <file-name> does not exist or does not have read permission

Potential Issues

A UPF file is loaded with load_upf that does not exist or that does not have a read permission.

Consequence of Not Fixing

The power intent cannot be analyzed thus no power verification is performed.

How to Debug and Fix

Load a UPF file using load_upf <file name> that exists and that has a read permission.

Example Code and/or Schematic

Consider the following UPF snippet:

```
load_upf ./const1.upf
```

For the above UPF snippet, the *UPFSTX_28* rule reports the following violation because *const1.upf* file does not exist:

```
File const1.upf does not exist or does not have read permission
```

To resolve the violation, specify a UPF file that exists.

Default Severity Label

Syntax

Rule Group

UPF Check

Reports and Related Files

None

UPFSTX_29

Checks the presence of an invalid command

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_29* rule reports a violation when an invalid UPF command is specified.

Messages and Suggested Fix

The following message appears when an invalid command is found:

[SYNTAX] `Internal Tcl Interpreter error in file <file-name> : <error>`

Potential Issues

An invalid UPF command is specified.

Consequence of Not Fixing

The power intent cannot be analyzed thus no power verification is performed.

How to Debug and Fix

Specify a valid UPF command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
xx
```

For the above UPF snippet, the *UPFSTX_29* rule reports the following violation because xx is not a valid UPF command:

```
invalid command name "xx" while executing "xx"
```

To resolve the violation, specify valid UPF commands.

Default Severity Label

Syntax

Rule Group

UPF Check

Reports and Related Files

None

UPFSTX_31

Checks the presence of extra arguments in a UPF command

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_31* rule reports a violation when extra arguments are specified in a UPF command.

Messages and Suggested Fix

The following message appears when an extra argument <*argument-name*> is specified in a UPF command <*upf-command*>:

[SYNTAX] <argument-name> is specified along with <argument-name> in upf command <upf-command>

Potential Issues

An extra argument is specified with an argument, which is not valid.

Consequence of Not Fixing

The power intent cannot be analyzed thus no power verification is performed.

How to Debug and Fix

Avoid specifying any extra argument, which is not valid.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_isolation iso1 -domain PD2  -isolation_power_net VPD2 -applies_to outputs -source ss1
```

For the above UPF snippet, the *UPFSTX_31* rule reports the following violation because the `–applies_to` argument is specified with the `–source` argument, which is not valid:

```
( -applies_to ) is specified along with ( -source ) in upf command ( set_isolation )
```

To avoid this violation, do not specify the `-applies_to` argument with the `-source` argument in the `set_isolation` command.

Default Severity Label

Syntax

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSTX_33

Checks the presence of arguments that cannot be specified together

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_33* rule reports a violation when multiple arguments that are not allowed together are specified in a UPF command.

Messages and Suggested Fix

The following message appears when multiple arguments *<arguments-list>* that are not allowed together are specified in a command:

[SYNTAX] Arguments <arguments-list> cannot be specified at same time in command <command-name>

Potential Issues

Arguments that cannot be specified together are specified in a UPF commands.

Consequence of Not Fixing

The power intent cannot be analyzed thus no power verification is performed.

How to Debug and Fix

Avoid specifying arguments that cannot be specified together in UPF commands.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_isolation iso1 -domain pd_top -isolation_power_net
vnn_top -clamp_value 0 -no_isolation
```

For the above UPF snippet, the *UPFSTX_33* rule reports the following violation because the `no_isolation` and `isolation_power_net` arguments are specified together:

```
Arguments ( -no_isolation and -isolation_power_net ) cannot be
specified at same time in command ( set_isolation )
```

To avoid this violation, only specify one, `no_isolation` or `isolation_power_net`, argument at a time.

Default Severity Label

Syntax

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSTX_34

Checks if at least one required argument is specified in a command

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_34* rule reports a violation when none of the required arguments is specified in a UPF command.

Messages and Suggested Fix

The following message appears when none of the required arguments is specified in a UPF command <*command*>:

[ERROR] At least one argument of these arguments <arguments-list> is required in command <command>

Potential Issues

At least one required argument is not specified in the UPF command.

Consequence of Not Fixing

The power intent cannot be analyzed thus no power verification is performed.

How to Debug and Fix

Specify at least one required argument in the UPF command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_isolation iso1 -domain pd_top -applies_to outputs -
clamp_value 0
```

For the above UPF snippet, the *UPFSTX_34* rule reports the following violation because at least one, `no_isolation` or `-isolation_power_net`, `-isolation_ground_net`, or `-isolation_supply_set` argument is not specified in the `iso1` isolation strategy:

```
At least one argument of these arguments ( -no_isolation or -
```

```
isolation_power_net or -isolation_ground_net or -
isolation_supply_set ) is required in command ( set_isolation )
```

To avoid the violation, specify any of the above arguments in the `iso1` strategy.

Default Severity Label

Error

Rule Group

`UPF Check`

Reports and Related Files

None

UPFSTX_39

Checks the existence of the port state mentioned in the add_power_state command

When to Use

This is a setup rule and always runs by default.

Description

The *UPFSTX_39* rule reports a violation when the port state mentioned in the `add_power_state` command does not exist.

Messages and Suggested Fix

The following message appears when the port state mentioned in the `add_power_state` command does not exist:

[FATAL] Unable to find voltage value corresponding to state '<state-name>' for supply net '<supply-net-name>' in add_power_state command '<add-power-state-name>' for supply set '<supply-set-name>'

Potential Issues

The violation message explicitly states the potential issue.

Consequence of Not Fixing

The power intent cannot be analyzed thus no power verification is performed.

How to Debug and Fix

Make sure that the port state mentioned exists in the `add_power_state` command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
upf_version 2.0
set_design_top top

create_supply_port VDD1
create_supply_port VSS
```

```
create_supply_net VDD1
create_supply_net VSS

create_supply_set ss1 -function {power VDD1} -function {pwell
VSS}

connect_supply_net VDD1 -ports {VDD1}
connect_supply_net VSS -ports {VSS}

add_port_state VDD1 -state {NOM 0.9}
add_port_state VSS -state {NOM 0.0}


create_power_domain TOP -include_scope -supply {primary ss1}

add_power_state ss1 -state NOM {-simstate NORMAL -supply_expr
{power == NOM1 && pwell == `{NOM , 0.0} } }
```

For the above UPF snippet, the *UPFSTX_39* rule reports the following violation because the state specified for supply set function `power (supply port VDD1)` does not exist:

```
Unable to find voltage value corresponding to state 'NOM1' for
supply net 'power' in add_power_state command 'NOM' for supply
set 'ss1'
```

Default Severity Label

Fatal

Rule Group

`UPF Check`

Reports and Related Files

None

UPFWRN_1

Checks if port/pin connection is simultaneously specified with mutually exclusive switch

When to Use

This is a setup rule and always runs by default.

Description

The *UPFWRN_1* reports a violation when a port/pin connection is simultaneously specified with mutually exclusive switch.

Messages and Suggested Fix

The following message appears when a port/pin connection is simultaneously specified with mutually exclusive switch:

[WARNING] Port/Pin connection is simultaneously specified in connection for net <net-name>

Potential Issues

The port/pin connection is simultaneously specified with mutually exclusive switch.

Consequence of Not Fixing

The relevant UPF command does not work properly.

How to Debug and Fix

The port/pin connection should not be specified simultaneously with mutually exclusive switch.

Example Code and/or Schematic

Consider the following UPF snippet:

```
connect_supply_net VAON -ports {VAON} \
-pins {inst1/and1/Y} -cells {AND}
```

For the above UPF snippet, the *UPFWRN_1* rule reports the following violation because port/pin connection is specified with mutually exclusive argument `cells`:

```
Port/Pin connection is simultaneously specified in connection
for net ( VAON )
```

To avoid the violation, avoid specifying port/pin connection with mutually exclusive argument `cells`.

Default Severity Label

Warning

Rule Group

`UPF Check`

Reports and Related Files

None

UPFWRN_2

Checks if the presence of the specified vct in scope

When to Use

This is a setup rule and always runs by default.

Description

The *UPFWRN_2* reports a violation when a specified VCT is not found in the scope.

Messages and Suggested Fix

The following message appears when the vct is not found in the scope:

[WARNING] Specified vct <vct-name> not found

Potential Issues

The specified vct is not found in scope.

Consequence of Not Fixing

The relevant UPF command does not work properly.

How to Debug and Fix

Specify the vct using the `create_upf2hdl_vct <vct_name>` command before use.

Example Code and/or Schematic

Consider the following UPF snippet:

```
connect_supply_net VAON -ports {VAON} -vct {vct_x}
```

For the above UPF snippet, the *UPFWRN_2* rule reports the following violation because the vct, `vct_x`, was not previously defined:

```
Specified vct ( vct_x ) not found
```

To avoid the violation, specify the vct, `vct_x`, before using it, as shown below:

```
create_upf2hdl_vct vct_x
```

Default Severity Label

Warning

Rule Group

UPF Check

Reports and Related Files

None

UPFWRN_3

Checks if the function is already associated with net

When to Use

This is a setup rule and always runs by default.

Description

The *UPFWRN_3* reports a violation when a function specified is already associated with the net.

Messages and Suggested Fix

The following message appears when the specified function is already associated with the net:

[WARNING] Function <function-name> is already associated with net <net-name>

Potential Issues

The specified function is already associated with net.

Consequence of Not Fixing

The relevant checks are not applied for the supply set.

How to Debug and Fix

Different supply nets should not be specified for the same function of supply sets.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_supply_set ss1 -function {power VVD} \
 -function {power VPD}
```

For the above UPF snippet, the *UPFWRN_3* rule reports the following violation because the function power is already associated with the supply net VVD:

```
Function ( power ) is already associated with net ( VVD )
```

To avoid the violation, different supply nets VVD and VPD should not be

specified for the same function `power` of supply set `ss1`.

Default Severity Label

Warning

Rule Group

`UPF Check`

Reports and Related Files

None

UPFWRN_4

Checks if the argument is redefined

When to Use

This is a setup rule and always runs by default.

Description

The *UPFWRN_4* reports a violation when an argument is redefined.

Messages and Suggested Fix

The following message appears when an argument is redefined:

[WARNING] Argument <argument-name> cannot be redefined in <name>

Potential Issues

An argument is redefined.

Consequence of Not Fixing

The relevant checks are not performed for the UPF command.

How to Debug and Fix

Avoid redefining an argument in the UPF command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_domain PD -elements inst1
create_power_domain PD -scope inst1 –update
```

For the above UPF snippet, the *UPFWRN_4* rule reports the following violation because the `scope` argument cannot be redefined in the `create_power_domain` UPF command:

```
Argument (-scope) cannot be redefined in (create_power_domain)
```

To avoid the violation, avoid redefining the `scope` argument in the `create_power_domain` UPF command.

Default Severity Label

Warning

Rule Group

UPF Check

Reports and Related Files

None

UPFWRN_5

Checks if an invalid resolve strategy is specified at the definition of net

When to Use

This is a setup rule and always runs by default.

Description

The *UPFWRN_5* reports a violation when an invalid resolve strategy is specified at the definition of net. The default strategy is set in this case.

Messages and Suggested Fix

The following message appears when an invalid resolve strategy is specified at the definition of net:

[WARNING] Invalid resolve strategy <resolve-strategy-name> specified at definition of net <net-name>. Setting default strategy (unresolved) for this net.

Potential Issues

An invalid resolve strategy is specified at the definition of net.

Consequence of Not Fixing

The relevant checks are not performed properly for the net.

How to Debug and Fix

Specify a valid resolve strategy at the definition of the net.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_supply_net  VAON -domain AON \
-resolve parallel_one_hot
```

For the above UPF snippet, the *UPFWRN_5* rule reports the following violation because the resolve strategy `parallel_one_hot` is not valid in UPF 1.0:

```
Invalid resolve strategy (parallel_one_hot) specified at
definition of net (VAON). Setting default strategy ( unresolved
) for this net
```

To avoid the violation, specify only valid resolve strategies for the net VAON.

Default Severity Label

Warning

Rule Group

UPF Check

Reports and Related Files

None

UPFWRN_6

Checks if a command is invoked multiple times

When to Use

This is a setup rule and always runs by default.

Description

The *UPFWRN_6* reports a violation when a command is invoked multiple times.

Messages and Suggested Fix

The following message appears when a command is invoked multiple times:

[WARNING] Command <command-name> is invoked more than once

Potential Issues

A UPF command is invoked multiple times.

Consequence of Not Fixing

The relevant checks are not performed on the UPF command.

How to Debug and Fix

Avoid specifying a UPF command multiple times.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_partial_on_translation OFF
set_partial_on_translation FULL_ON
```

For the above UPF snippet, the *UPFWRN_6* rule reports the following violation because the `set_partial_on_translation` UPF command is specified multiple times:

```
Command ( set_partial_on_translation ) is invoked more than once
```

To avoid the violation, avoid specifying the `set_partial_on_translation` UPF command multiple times.

Default Severity Label

Warning

Rule Group

UPF Check

Reports and Related Files

None

UPFWRN_7

Checks if a list is specified for an argument

When to Use

This is a setup rule and always runs by default.

Description

The *UPFWRN_7* reports a violation when a list is specified for an argument. In this case, the first value in the list is considered as the final value and the remaining are ignored.

Messages and Suggested Fix

The following message appears when a list is specified for an argument:

[WARNING] List cannot be specified for argument <argument-name>. First value in list will be considered as final value, rest will be ignored

Potential Issues

A list is specified for an argument.

Consequence of Not Fixing

The relevant checks are not performed on the UPF command.

How to Debug and Fix

List should not be specified for a UPF command.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_design_top {mod mod1}
```

For the above UPF snippet, the *UPFWRN_7* rule reports the following violation because a list is specified for the `set_design_top` UPF command:

```
List cannot be specified for argument (instName). First value in list will be considered as final value, rest will be ignored
```

To avoid the violation, avoid specifying a list for the `set_design_top` UPF command.

Default Severity Label

Warning

Rule Group

UPF Check

Reports and Related Files

None

UPFWRN_13

Checks if the state name is used multiple times for a port

When to Use

This is a setup rule and always runs by default.

Description

The *UPFWRN_13* reports a violation when the same state names are used multiple times for a port.

Messages and Suggested Fix

The following message appears when the same state names are used multiple times for a port:

[WARNING] Same state name <state-name> used multiple times for port <port-name>

Potential Issues

The state names are used multiple times for a port.

Consequence of Not Fixing

The relevant checks are not performed properly for the UPF command.

How to Debug and Fix

State names specified should not be the same for a supply port.

Example Code and/or Schematic

Consider the following UPF snippet:

```
add_port_state VAON -state {on1 1.0} -state {on1 1.0}
```

For the above UPF snippet, the *UPFWRN_13* rule reports the following violation because the state name `on1` is used twice in the `add_port_state` command for supply port `VAON`:

```
Same state name ( on1 ) used multiple times for port ( VAON )
```

To avoid the violation, the state name `on1` should not be specified multiple times for supply port `VAON`.

Default Severity Label

Warning

Rule Group

UPF Check

Reports and Related Files

None

UPFWRN_14

Checks if PST state is redefined

When to Use

This is a setup rule and always runs by default.

Description

The *UPFWRN_14* reports a violation when PST state is redefined.

Messages and Suggested Fix

The following message appears when PST state is redefined:

[WARNING] Re-definition of pst state <state-name>. Ignoring...

Potential Issues

The PST state is redefined.

Consequence of Not Fixing

The relevant checks are not performed properly for the PST state.

How to Debug and Fix

PST state should not be redefined.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_pst ps1 -supplies {VAON VVD VPD VPD2 VSS}
add_pst_state s1 -pst ps1 -state {on1 on1 on1 on1 def}
add_pst_state s1 -pst ps1 -state {on1 on1 on1 on1 def}
```

For the above UPF snippet, the *UPFWRN_14* rule reports the following violation because the PST state `s1` is redefined:

```
Re-definition of pst state ( s1 ). Ignoring...
```

To avoid the violation, the PST state `s1` should not be redefined.

Default Severity Label

Warning

Rule Group

UPF Check

Reports and Related Files

None

UPFWRN_16

Checks if the command is supported

When to Use

This is a setup rule and always runs by default.

Description

The *UPFWRN_16* reports a violation when the command specified is not supported.

Messages and Suggested Fix

The following message appears when the command specified is not supported:

[WARNING] Command <command-name> not supported

Potential Issues

The specified command is not supported.

Consequence of Not Fixing

The relevant checks are not performed properly for the UPF command.

How to Debug and Fix

The unsupported command is highlighted in the Atrenta Console GUI.

For the list of unsupported commands, refer to *Unsupported UPF Commands*.

For the list of commands that do not have any impact on verification, refer to *Inconsequential UPF Commands*.

To fix this violation, use a command that is supported.

Example Code and/or Schematic

Consider the following UPF snippet:

```
query_net_ports VSS
```

For the above UPF snippet, the *UPFWRN_16* rule reports the following violation because the `query_net_ports` command is not supported:

```
Command ( query_net_ports ) not supported
```

To avoid the violation, avoid specifying the `query_net_ports`

command, which is not supported.

Default Severity Label

Warning

Rule Group

UPF Check

Reports and Related Files

None

UPFWRN_17

Checks if an option in command is not supported

When to Use

This is a setup rule and always runs by default.

Description

The *UPFWRN_17* reports a violation when an option in the command specified is not supported.

Messages and Suggested Fix

The following message appears when an option in the command specified is not supported:

[WARNING] Option <option-name> not supported for command <command-name>

Potential Issues

The specified option in a command is not supported.

Consequence of Not Fixing

The option is ignored and does not have any impact. Only the first specification of the option is reported and all later specifications are ignored.

How to Debug and Fix

Avoid specifying unsupported options in UPF commands.

Example Code and/or Schematic

Consider the following UPF snippet:

```
create_power_domain PD -elements inst1 \
-exclude_elements {inst2}
```

For the above UPF snippet, the *UPFWRN_17* rule reports the following violation because the `exclude_elements` option is not supported for the `create_power_domain` UPF command:

```
Option ( -exclude_elements ) not supported for command ( create_power_domain )
```

To avoid the violation, avoid specifying option `-exclude_elements`, which is not supported in the `create_power_domain` UPF command.

Default Severity Label

Warning

Rule Group

`UPF Check`

Reports and Related Files

None

UPFWRN_19

Checks if the value of supply port specified in the add_power_state command is different than in the add_port_state command

When to Use

This is a setup rule and always runs by default.

Description

The *UPFWRN_19* reports a violation when the value of supply port specified in the `add_power_state` command is different than the one provided in the `add_port_state` command. In such a case, the value specified in the `add_power_state` command will be preferred.

Messages and Suggested Fix

The following message appears when the value of supply port specified in the `add_power_state` command is different than in the `add_port_state` command:

[WARNING] Conflict in voltage value corresponding to state '<state-name>' for supply net '<supply-net-name>' in add_power_state command '<add-power-state-name>' for supply set '<supply-set-name>', updating value given in add_power_state command

Potential Issues

The violation message explicitly states the potential issue.

Consequence of Not Fixing

The value specified in the `add_power_state` command will be used.

How to Debug and Fix

Make sure that the same values for supply port are specified in the `add_power_state` and `add_port_state` commands.

Example Code and/or Schematic

Consider the following UPF snippet:

```
upf_version 2.0
```

```
set_design_top top

create_supply_port VDD1
create_supply_port VSS

create_supply_net VDD1
create_supply_net VSS

create_supply_set ss1 -function {power VDD1} -function {pwell
VSS}

connect_supply_net VDD1 -ports {VDD1}
connect_supply_net VSS -ports {VSS}

add_port_state VDD1 -state {NOM 0.9}
add_port_state VSS -state {NOM 0.0}


create_power_domain TOP -include_scope -supply {primary ss1}

add_power_state ss1 -state NOM {-simstate NORMAL -supply_expr
{power == `{NOM, 1.2} && pwell == `{NOM , 0.0} } }
```

For the above UPF snippet, the *UPFWRN_19* rule reports the following violation because the value specified for supply set function `power (supply port VDD1)` is not the same in `add_power_state` and `add_port_state` commands:

```
Conflict in voltage value corresponding to state 'NOM' for
supply net 'power' in add_power_state command 'NOM' for supply
set 'ss1', updating value given in add_power_state command
```

Default Severity Label

Warning

Rule Group

UPF Check

Reports and Related Files

None

UPFINFO_3

Checks if an unsupported UPF version is specified

When to Use

This is a setup rule and always runs by default.

Description

The *UPFINFO_3* reports a violation when an unsupported UPF version is specified.

Messages and Suggested Fix

The following message appears when an unsupported UPF version is specified:

[INFO] `UpfAnalyzer does not support version <version-name>`

Potential Issues

An unsupported UPF version is specified.

Consequence of Not Fixing

The relevant are not performed properly for the UPF commands and the default version is used.

How to Debug and Fix

Avoid specifying unsupported UPF versions.

Example Code and/or Schematic

Consider the following UPF snippet:

```
upf_version 1.1
```

For the above UPF snippet, the *UPFINFO_3* rule reports the following violation because the UPF version 1.1 is not supported:

```
UpfAnalyzer does not support version ( 1.1 )
```

To avoid the violation, avoid specifying unsupported versions in the `upf_verion` UPF command.

Default Severity Label

Info

Rule Group

UPF Check

Reports and Related Files

None

UPFINFO_7

Checks if the supply is already set for a domain

When to Use

This is a setup rule and always runs by default.

Description

The *UPFINFO_7* reports a violation when the supply is already set for a domain.

Messages and Suggested Fix

The following message appears when the supply is already set for a domain:

[INFO] Argument <argument-name> of domain <domain-name> was set previously with supply net <net-name>

Potential Issues

The supply is already set for the domain.

Consequence of Not Fixing

The relevant are not performed properly for the UPF commands.

How to Debug and Fix

Specify only one supply for a domain as primary power/ground.

Example Code and/or Schematic

Consider the following UPF snippet:

```
set_domain_supply_net AON -primary_power_net VAON \
-primary_ground_net VSS
set_domain_supply_net AON -primary_power_net VVD \
-primary_ground_net VSS
```

For the above UPF snippet, the *UPFINFO_7* rule reports the following violation because the supply AON was set previously:

```
Argument (-primary_power_net) of domain (AON) was set
previously with supply net (VAON)
```

To avoid the violation, specify only one supply for a domain as primary

power/ground.

Default Severity Label

Info

Rule Group

UPF Check

Reports and Related Files

None

Debug Rules

The debug rules are as follows:

Rule	Purpose
LP_SDC_PARSE_DEBUG	Reports SDC output to help debug complex hierarchical paths

LP_SDC_PARSE_DEBUG

Reports SDC output to help debug complex hierarchical paths

When to Use

When the hierarchical path for an element in UPF is not found by SpyGlass (existence check error reported).

Description

This rule reports the SDC output to help debug complex hierarchical paths inferred by SpyGlass. Sometimes SpyGlass reports an existence check error for the design elements written through complex structures like nested generate blocks or SV constructs. This rule reports the hierarchical path inferred in such cases by SpyGlass, which can be specified in UPF.

NOTE: *User needs to specify 'sdc_puts' in SDC file for this rule to report the hierarchical paths, which are reported on screen output.*

Only the following SDC constraints that report hierarchical paths are supported:

- get_ports
- get_pins
- get_nets
- get_cells
- get_lib_pins
- get_lib_cells

The remaining constraints are ignored even if they are present in your SDC file. The ignored constraints are listed in the `ignored_commands.txt` file.

Prerequisites

This rule is not enabled by default. You need to specify the following:

```
set_option rules LP_SDC_PARSE_DEBUG
```

Parameter(s)

lp_ignore_tc_commands: Default value is `null`. Set the value of this parameter as the name of the file that contains the list of SDC constraints

to be ignored.

Messages and Suggested Fix

None

Example Code and/or Schematic

For the following SDC commands, the *LP_SDC_PARSE_DUBG* rule reports hierarchical names of the objects on the screen output.

SDC Commands:

```
set c0 [get_cells -hier *]
sdc_puts "c0 = $c0"
set c1 [get_cells -hier out1_reg]
sdc_puts "c1 = $c1"
set c2 [get_cells -hier *_reg]
sdc_puts "c2 = $c2"
set c3 [get_cells -hier u_PD2/b*]
sdc_puts "c3 = $c3"
set c4 [get_cells -hier t1*]
sdc_puts "c4 = $c4"
```

Output:

```
Checking Rule LP_SDC_PARSE_DEBUG(C-Rule: VsLpSDCParseDebug)
of VSDU Type (Rule 127 of total 229) c0 = u_PD1 u_PD1/u_PD1_1
u_PD1/u_PD1_1/buf1 iso1 u_PD2 u_PD2/buf2 u_VD c1=
c2 =
c3 = u_PD2/buf2
c4 =
```

Default Severity Label

None

Rule Group

Debug Rules

Reports and Related Files

None

SoC Abstraction Rules

The SOC abstraction rules are as follows:

Rule	Purpose
PV_Abstract01	Generates power model for a block in the form of liberty or UPF file, to be used for SoC power verification
PV_AbstractReadInfo	Reports if the power model for a block in the form of liberty or UPF file has been read

PV_Abstract01

Generates power model for a block in the form of liberty or UPF file, to be used for SoC power verification

When to Use

This rule is applicable to all design phases and works only in the UPF format.

Prerequisite:

Use the `power_new` license to run this rule.

Description

The *PV_Abstract01* generates a model for a block such that all the information needed for SoC power verification is available at the interface of the model. This rule can abstract power intent of a block in the form of a library (.lib) or a UPF file. For power verification of the SoC, the abstracted power models of these blocks can be used for quicker turnaround.

Make sure that the block being abstracted does not have any violations. The model would still be generated if there are violations. But the supply information available on interfaces might not be correct, leading to unreliable power reports.

Liberty file based power model generation flow

In the LIB flow, the liberty file is generated as `<modulename>_lib_power_abstract.lib`. The generated abstractions also includes an auxiliary UPF file (`<modulename>_upf_power_intent.upf`), along with the LIB file. This UPF contains basic power intent details based on the UPF version 1.0.

The following attributes are supported in the generated liberty file:

Liberty attribute	Scope	Purpose
related_ground_pin	pin	Ground pin associated to functional pin
voltage_map	library	Assign voltage value to library rail
voltage_name	pg_pin	Assign voltage value to library pin

Liberty attribute	Scope	Purpose
related_power_pin	pin	Power pin associated to functional pin
pg_type	pin	Specify the type of power and ground
pg_pin	cell	Declare the PG pin
direction	pin, pg_pin	Declare direction such as input or output
is_isolated	pin	Declare pin as isolated
isolation_enable_condition	pin	Declare enable condition on the isolation enable pin (generated only when the `isolation_signal` is connected directly to the top level port, with buffer or inverter in between)
is_macro_cell	cell	Declare whether a cell is a macro cell
pg_function	pg_pin	Present in output power pin
switch_cell_type	cell	Declare cell as power switch
switch_function	pg_pin	Expression of enable pin
switch_pin	pin	Power switch enable pin
function	pin	Defines value in terms of input or inout pins
retention_pin	pin	Defines the retention pin of a cell
always_on	pin	Identifies always on pins

By default, the vector ports are not bit-blasted in the generated liberty file. To enable bit-blasting in generated model, the parameter *lp_gen_block_abstraction_in_bit_blasted_way* should be set to `1`.

The following commands are supported in the auxiliary UPF generated with `.lib` model.

Command	Purpose
create_power_domain	Define a power domain and its characteristics
create_power_switch	Define a power switch
set_domain_supply_net	Set the default power and ground supply nets for a domain
create_supply_net	Create a supply net
create_supply_port	Create a supply port on a instance
connect_supply_net	Connect a supply net to supply ports
add_port_state	Add states to a port

UPF-based power model generation flow

In the UPF flow, the UPF file is generated as `<modulename>_upf_power_abstract.upf`, with the `begin_power_model` and `end_power_model` commands.

This approach has support for the following UPF commands. UPF version is 2.0 and the constructs are from the version 2.1.

Command	Purpose
create_power_domain	Define a power domain and its characteristics
create_power_switch	Define a power switch
create_supply_set	Create or update a supply set, or update a supply set handle
create_supply_net	Create a supply net
create_supply_port	Create a supply port on a instance
connect_supply_net	Connect a supply net to supply ports
add_port_state	Add states to a port
create_pst	Create a power state table (PST)

Command	Purpose
add_pst_state	Define the states of each of the supply nets for one possible state of the design
set_port_attribute	Define information on ports
begin_power_model	Define a power model
end_power_model	Terminate the definition of a power model

In both the flows, as described above, the abstracted files are generated in a separate directory. The default directory is `spyglass_reports/lowpower/abstract_view`. The directory location can be set to any desired path by using the *lp_block_abstraction_dir* parameter.

Rule Exceptions

- In case design has pins having \ as part of its name, then the generated abstract model in the form of `.lib` will also have the same names. But it is not allowed to have backslash character in the pin names in `.lib`. You will get fatal parsing error while reading this library.

  Therefore, the `.lib` generation flow using the *PV_Abstract01* rule cannot be used for designs that have backslash in their pin names.

- If a block, which is abstracted, has an `inout` port and has the `set_port_attributes` command specified with both driver and receiver supplies, then while reading the abstract model of such a block SpyGlass will randomly choose supply for that port (between the receiver and driver supply). This means, either the driver supply or receiver supply will be considered.

Configuring the Liberty/UPF Power Model

You can configure the liberty or UPF power model generation by providing an input file to SpyGlass. This input file should be specified through the *lp_abstraction_config_file* parameter. The following table lists the liberty attributes (LIB flow) and power intent commands (UPF flow), which you can customize as per the requirement. The parameters and their valid values that can be passed through the configuration file are also listed in the following table.

Configuration	IV	DOMAIN	SRSN	ISO_POWER_NET	CSN
soc_default_pin_supply	Default	Y	Y	N	N
soc_ls_pin_supply	Default	Y	Y	N	N
soc_isolated_pin_supply	Default	Y	Y	Y	N
soc_unconnected_pin_supply	N	Default	Y	N	N
soc_tied_pin_supply	N	Default	Y	N	Y
soc_skip_lib_buf_for_attribute_propagation	Boolean (True default)				
soc_skip_lib_buf_for_feedthrough	Boolean (False default)				

In the above table, the headings represent the following:

- IV: Internal voltage.
- DOMAIN: Domain voltage.
- SRSN: The related supply of the port specified through the `set_related_supply_net` or `set_port_attributes` UPF constraints.
- ISO_POWER_NET: The isolation power net. Valid only for crossings that have isolation strategy in the path.
- CSN: Connection specified in the HDL file for the port.

Parameter(s)

- *lp_gen_block_abstraction_format*: Default value is `lib`. Set the value of this parameter to `upf` to generate the abstracted UPF file.
- *lp_set_abstract_power_model_name*: By default, this parameter is not set and the name of the power model generated would be `upf_<module-name>`. Set the value of this parameter to desired power model name to be generated.

- *lp_block_abstraction_dir*: Default value is `spyglass_reports/lowpower/abstract_view/`. Set this parameter to a desired directory name to be created for the generated SoC abstractions.
- *lp_gen_block_abstraction_in_bit_blasted_way*: Default value is `0`. Set this parameter to `1` to enable bit-blasting of vector ports in generated liberty file.
- *lp_abstraction_config_file*: Default value is `null`. Set this parameter to the path of configuration file to be used for generating the abstraction model.

Messages and Suggested Fix

Message 1

The following message appears when the lib or UPF abstraction model has been generated:

[PV_Abstract01_1][INFO] <LIB abstraction | UPF abstraction > for design ‘<module-name>’ successfully created

In the LIB abstraction model, the following additional message is also reported:

Auxiliary UPF corresponding to LIB model for design ‘<module-name>’ successfully created

Message 2

The following message appears when the wrong value is provided for the *lp_gen_block_abstraction_format* parameter:

[PV_Abstract01_2][INFO] Specify the parameter value as 'lib' or 'upf' instead of '<parameter-value>'

Message 3

The following message appears when the license is not found:

[PV_Abstract01_3][FATAL] Rule 'PV_Abstract01' not run due to unavailability of 'lowpower_soc_abstraction, soc_abstraction' license feature

Potential Issues

The violation message explicitly states the potential issue.

Consequence of Not Fixing

The SpyGlass Power Verify solution will not run.

How to Debug and Fix

Make sure that you have provided the `lowpower_soc_abstraction` or `soc_abstraction` license for running this rule.

Message 4

The following message appears when path specified in *lp_block_abstraction_dir* parameter is invalid:

[PV_Abstract01_4][INFO] Abstract model could not be generated at desired location ‘<location>’

Message 5

The following message appears when a signal pin has multiple fanouts in the design:

[PV_Abstract01_5][ERROR] Pin '<pin-name>' of block '<block-name>' is going to multiple destinations. Generated abstract model might be incorrect

Message 6

The following message appears when a signal pin has multiple fanins in the design:

[PV_Abstract01_6][ERROR] Pin '<pin-name>' of block '<block-name>' is coming from multiple sources. Generated abstract model might be incorrect

Message 7

The following message appears when an invalid option is specified in configuration file:

[PV_Abstract01_7][ERROR] Invalid option '<invalid-configuration-name>' provided in configuration file

Message 8

The following message appears when an invalid value is specified for a parameter in configuration file:

[PV_Abstract01_8][ERROR] Invalid value '<value>' given for option '<configuration-name>' in configuration file. Valid values are : [< valid-value-list>]

Message 9

The following message appears when the parent supply is missing for an internal power supply. In this case, the pg_type is shown as ''primary'':

[PV_Abstract01_9][ERROR] Generating abstract model with supply '<supply-name>' as primary supply. Parent supply not inferred for this supply

Example Code and/or Schematic

Consider a block, B1, in the following UPF file, B1.upf:

```
upf_version 2.0
set_design_top B1
create_supply_port vddA
create_supply_net vddA
connect_supply_net vddA -ports {vddA}
add_port_state vddA -state {on1 0.8} -state {off1 off}
create_supply_port vssA
create_supply_net vssA
connect_supply_net vssA -ports {vssA}
add_port_state vssA -state {on1 0.0}
…
create_supply_set ss1 -function {power vddA} -function
{ground vssA}
…
set_level_shifter ls1 -domain TOP_cellA -elements {W}
input_supply_set ss1 -rule low_to_high  -location self
set_port_attributes -ports {X} -driver_supply ss2
create_pst pst1 -supplies {vddA vddB vddB_int vssA vssB}
add_pst_state s1 -pst pst1 -state {* * * * * }
```

To abstract the B1 block, the following inputs are used.

SGDC Commands:

```
current_design B1
power_data –format UPF –file B1.upf
```

Project file snippet:

```
set_option top B1
define_goal test_goal –policy {lowpower}
```

```
set_goal_option addrules PV_Abstract01
set_parameter lp_gen_block_abstraction_format upf
```

For the above example, the following messages are reported because the `lp_gen_block_abstraction_format` parameter is set to UPF:

```
UPF abstraction for design ‘B1’ successfully created
```

The following UPF file is generated:

B1_upf_power_abstract.upf

When the `lp_gen_block_abstraction_format` parameter is set to LIB:

Project file snippet:

```
set_option top B1
define_goal test_goal –policy {lowpower}
set_goal_option addrules PV_Abstract01
set_parameter lp_gen_block_abstraction_format upf
```

For the above example, the following messages are reported because the `lp_gen_block_abstraction_format` parameter is set to LIB:

```
LIB abstraction for design ‘B1’ successfully created
```

The following library file is generated:

B1_lib_power_abstract.lib

The following UPF file is generated along with the above library file:

B1_upf_power_intent.upf

Default Severity Label

Info

Rule Group

SoC Abstraction Rules

Reports and Related Files

None

PV_AbstractReadInfo

Reports if the power model for a block in the form of liberty or UPF file has been read

When to Use

This rule is applicable to all design phases and works only in the UPF format.

Description

The *PV_AbstractReadInfo rule* reports that the abstraction model in the form of liberty or UPF has been read in the SOC flow.

Parameter(s)

None

Messages and Suggested Fix

Message 1

The following message appears when the UPF abstraction model has been read:

[PV_AbstractReadInfo_1][INFO] UPF Abstracted view has been read for instance '<instance-name>' [Master : '<power-model-name>']. 'stop' will be inferred automatically for this instance

Message 2

The following message appears when the LIB abstraction model has been read:

[PV_AbstractReadInfo_2][INFO] LIB Abstracted view has been read for instance '<instance-name>' [Master : '<master-cell-name>']

Example Code and/or Schematic

Example 1

Consider the following UPF snippet:

```
begin_power_model upf_macroA -for {cellA}
...
end_power_model
```

```
apply_power_model upf_macroA -elements {I2} -supply_map { {
PD.SSAH PDTop.SSH1 } {PD.SSBH PDTop.SSH2} }
```

For the above example, the following message is reported:

```
UPF Abstracted view has been read for instance 'top.I2' [Master
: 'upf_macroA']. 'stop' will be inferred automatically for this
instance
```

Example 2

Consider the following library snippet generated in the abstraction flow where a special library attribute `spyglass_pv_gen_lib_cell` has been defined:

```
define(spyglass_pv_gen_lib_cell,cell,boolean);
cell (cellA) {
    spyglass_pv_gen_lib_cell : true;
is_macro_cell : true;
      pin (W) { direction : input;related_power_pin: vddA;
related_ground_pin: vssA; }
      pin (X) { direction : input;related_power_pin: vddA;
related_ground_pin: vssA; is_isolated:true;
isolation_enable_condition : "W"}
      pin (Y) { switch_pin : true; direction :
input;related_power_pin: vddB; related_ground_pin: vssB; }
      pin (Z) { direction : input;related_power_pin: vddB;
related_ground_pin: vssB; }
      pin (Z1) { direction : output; function :
"W&X";related_power_pin: vddB_int;related_ground_pin: vssB;
power_down_function : "!vddB + vssB"; }
      pin (Z2) { direction : output; function :
"Y&Z";related_power_pin: vddB;related_ground_pin: vssB;  }
   }
```

Consider the following Verilog snippet where instance `I3` of the lib cell `cellA`, given above, has been instantiated:

```
cellA  I3 (.W(in3), .X(w1), .Y(in2), .Z(w2), .Z1(w3),
.Z2(w4));
```

For the above example, the following message is reported:

```
LIB Abstracted view has been read for instance 'top.I3' [Master
```

```
: 'cellA']
```

Default Severity Label

Info

Rule Group

SoC Abstraction Rules

Reports and Related Files

None

Appendix: CPF Commands

The SpyGlass Power Verify solution provides support for (Common Power Format) CPF commands Version 1.0e. You can specify power specifications in a CPF file and provide it as an input. SpyGlass will read and parse the CPF commands and based on the power intent information provided with these commands, performs the rule checking.

Supported SGDC Commands

For the CPF flow, SpyGlass also honors the following SGDC commands:

- *activity*
- *always_on_buffer*
- *assume_path*
- *assertion_signal*
- *cell_hookup*
- *cell_pin_info*
- *cell_tie_class*
- *clock*
- *multivt_lib*
- *non_pd_inputcells*
- *power_data*

- *power_down*
- *power_down_sequence*
- *pg_cell*
- *pg_pins_naming*
- *ram_instance*
- *ram_switch*
- *set_case_analysis*
- *special_cell*
- *switchoff_wrapper_instance*

CPF Check Rules

Specifying Constraints

Example of Using CPF / UPF Commands

CPF Commands Supported by SpyGlass Power Verify

This section provides details of CPF commands supported by the SpyGlass Power Verify solution. The commands have been categorized in the following categories:

- *Commands to Specify Voltage/Power Domain*
- *Commands to Specify Supply Information*
- *Commands to Specify Power Mode*
- *Commands to Specify Isolation Details*
- *Commands to Specify Level Shifter*
- *Commands to Specify Power Switch*
- *Commands to Specify Retention*
- *Commands to Specify Always On Cells*
- *Commands to Specify Power Clamp Cell (Diode Cells) Details*
- *Other Commands*

NOTE: *This document provides complete syntax of the CPF commands supported by SpyGlass. However, the arguments that do not have any effect on SpyGlass rule behavior are highlighted in bold.*

CPF Check Rules

Specifying Constraints

Example of Using CPF / UPF Commands

Commands to Specify Voltage/Power Domain

Following are the commands under this category:

create_power_domain

Syntax

```
create_power_domain
  -name <power_domain>
[ -instances instance_list]
[ -boundary_ports pin_list]
[ -default]
[ -shutoff_condition <expression> |
  -external_controlled_shutoff ]
[ -default_restore_edge <expression> |
    -default_save_edge <expression>   |
  -default_restore_edge <expression>
    -default_save_edge <expression>   |
  -default_restore_level <expression>
   -default_save_level <expression> ]
[ -power_up_states <high|low|random> ]
[ -default_isolation_condition expression ]
[ -active_state_conditions active_state_condition_list]
[ -secondary_domains domain_list]
```

Description

The `create_power_domain` command creates a power domain.

Arguments

The `create_power_domain` command has the following arguments:

-name <power-domain>

Specifies the name of the voltage/power domain.

-default

(Optional) Specifies that the top domain name specified with the `set_design` command should be taken as the name of the top-level design unit belonging to the voltage/power domain being specified.

-instances <instance_list>

(Optional) Space-separated name list of instances belonging to the voltage/power domain being specified.

-boundary_ports <pin_list>

(Optional) Specifies a space-separated name list of top level ports and terminal of leaf-level design units working on a different voltage level from their parent module.

-shutoff_condition <expression>

(Optional) Specifies the condition when the power domain is shut off. If this condition is given, the domain is treated as power domain.

-external_controlled_shutoff

(Optional) Specifies that the power domain is shut off by a power switch, which is controlled by an external signal.

-default_restore_edge <expression>

(Optional) Specifies the condition when the state of the sequential elements is to be restored for state retention rules.

NOTE: *Simple expressions or complex expressions only with | and ! (OR and NOT) operators are supported currently. For example,* `<restore-sig-name>` *or* `!<restore-sig-name>` *or* `<restore-sig-name>|!<restore-sig-name>`*.*

-default_save_edge <expression>

(Optional) Specifies the condition when the state of the sequential elements is to be saved for state retention rules.

NOTE: *Simple expressions or complex expressions only with | and ! (OR and NOT) operators are supported currently. For example,* `<save-sig-name>` *or* `!<save-sig-name>` *or* `<save-sig-name>|!<save-sig-name>`*.*

The following argument of the `create_power_domain` command that do not have any effect on the rule behavior.

- -power_up_states <high | low | random>

- -default_isolation_condition <expression>
- -default_restore_level <expression>
- -default_save_level <expression>
- -active_state_conditions active_state_condition_list
- -secondary_domains domain_list

update_power_domain

Syntax

```
update_power_domain
-name <domain>
-equivalent_power_nets list_of_power_nets
-equivalent_ground_nets list_of_ground_nets
{ -primary_power_net net |
  -primary_ground_net net |
  -pmos_bias_net net |
  -nmos_bias_net net |
  -user_attributes string_list |
  -transition_slope [float:]float |
  -transition_latency {from_nom latency_list} |
-transition_cycles {from_nom cycle_list clock_pin}}...
```

Description

The `update_power_domain` command updates a power domain.

Arguments

The `update_power_domain` command has the following arguments:

`-name <domain>`

Specifies the name of the voltage/power domain.

`-equivalent_power_nets list_of_power_nets`

Specifies a set of ground nets that are equivalent to the primary ground net of the power domain.

`-equivalent_ground_nets list_of_ground_nets`

Specifies a set of power nets that are equivalent to the primary power net of the power domain.

-primary_power_net <net>

Specifies the name of the primary power rail (port) associated with this domain. The voltage value of this rail is taken as the operating voltage value of this domain.

NOTE: *Prior to CPF version 1.0e, the name of this option was* `-internal_power_net`.

-primary_ground_net <net>

Specifies the name of ground rail (port) associated with this domain.

NOTE: *Prior to CPF version 1.0e, the name of this option was* `-internal_ground_net`.

The following arguments of the `update_power_domain` command that do not have any effect on the rule behavior:

- -pmos_bias_net <net>
- -nmos_bias_net <net>
- -user_attributes <string_list>
- -transition_slope [float:]float
- -transition_latency {from_nom latency_list}
- -transition_cycles {from_nom cycle_list clock_pin}

Commands to Specify Supply Information

Following are the commands under this category:

- *create_power_nets*
- *create_ground_nets*
- *create_global_connection*

CPF Check Rules

Specifying Constraints

Example of Using CPF / UPF Commands

create_power_nets

Syntax

```
create_power_nets
-nets <net_list>
[-voltage {float | voltage_range}]
[-external_shutoff_condition <expression> | -internal]
[-user_attributes <string_list>]
[-peak_ir_drop_limit <float>]
[-average_ir_drop_limit <float>]
```

Description

The `create_power_nets` command creates a list of power nets.

Arguments

The `create_power_nets` command has the following arguments:

`-nets <net-list>`

Simple name of the power port (post-synthesis pin).

`-voltage {float | voltage_range}`

(Optional) States the voltage value at the specified port. You can also provide a voltage range as follows:

`lower_bound:upper_bound`

Specifying the lower bound and upper bound of the voltage range, respectively.

`-external_shutoff_condition <expression>`

(Optional) Specifies the condition when the power source can be switched off, if the specified power nets are powered by an external power source. If this argument is specified, the power supply is assumed to be switchable power supply.

If this argument is not specified, the power source is assumed to be an always-on power supply.

-internal

(Optional) If this argument is specified, the power supply is assumed to be switchable power supply.

If this argument is not specified, the power source is assumed to be an always-on power supply.

The following arguments of the `create_power_nets` command that do not have any effect on the rule behavior:

- -user_attributes <string_list>
- -peak_ir_drop_limit <float>
- -average_ir_drop_limit <float>

create_ground_nets

Syntax

```
create_ground_nets
-nets net_list
[-voltage {float | voltage_range}]
[-external_shutoff_condition <expression> | -internal]
[-user_attributes <string_list>]
[-peak_ir_drop_limit <float>]
[-average_ir_drop_limit <float>]
```

Description

The `create_ground_nets` command creates a list of ground nets.

Arguments

The `create_ground_nets` command has the following arguments:

`-nets <net-list>`

Simple name of the ground port (post-synthesis pin).

`-voltage {float | voltage_range}`

(Optional) States the voltage value at the specified port. You can also provide a voltage range as follows:

`lower_bound:upper_bound`

Specifying the lower bound and upper bound of the voltage range, respectively.

`-external_shutoff_condition <expression>`

(Optional) If a specified ground net is connected to an external source, use this option to specify the condition under which the external source can be switched off.

`-internal`

(Optional) Specifies that the ground supply is a switchable ground

supply.

NOTE: *If you do not specify any of* `-external_shutoff_condition` *or* `-internal` *option, the ground source is assumed to be always on or off-chip controlled external switchable.*

The following arguments of the `create_ground_nets` command that do not have any effect on the rule behavior:

- -user_attributes <string_list>
- -peak_ir_drop_limit <float>
- -average_ir_drop_limit <float>

create_global_connection

Syntax

```
create_global_connection
 -net net
 -pin <pin_list>
[-domain domain | -instances <instance_list>]
```

Description

The `create_global_connection` command creates a connection of a global net (data net, bias net, power net or ground net) to the specified pins. However, if a pin specified in *pin_list*, is already connected, that pin is ignored for connection.

Arguments

The `create_global_connection` command has the following arguments:

`-nets net`

Specifies the name of the global net to be connected.

`-pin <pin_list>`

Specifies a list of LEF pins to be connected to the global net. If several pins of the same instance have names that match the specified names, all such pins will be connected to the specified global net.

You can use wildcards (`*`) to specify the pin names.

`-domain domain`

(Optional) Filters the given pin list to the pins that belong to the specified power domain.

`-instances <instance_list>`

(Optional) Filters the given pin list to the pins that belong to the specified instances.

Commands to Specify Power Mode

Following are the commands under this category:

- *create_power_mode*
- *create_nominal_condition*

CPF Check Rules

Specifying Constraints

Example of Using CPF / UPF Commands

create_power_mode

Syntax

```
create_power_mode
 -name <name>
{-domain_conditions <domain_condition_list> |
 -group_modes group_mode_list |
 -domain_conditions <domain_condition_list>
  -group_modes group_mode_list }
[-default]
```

Description

The `create_power_mode` command creates a power mode.

Arguments

The `create_power_mode` command has the following arguments:

-name <name>

Specifies the name of the power mode (power state).

-domain_conditions <domain_condition_list>

Specifies the nominal condition for a power domain. It is specified as *<domain-name>@<nominal-condition>*

NOTE: *If a domain has a nominal condition specified with voltage value as 0 or it is not specified in the <domain_condition_list>, that domain is considered as power domain (switched off).*

The following argument of the `create_power_mode` command that does not have any effect on the rule behavior:

- -default
- -group_modes group_mode_list

create_nominal_condition

Syntax

```
create_nominal_condition
-name <name>
-voltage <float>
[-ground_voltage float]
[-state {on | off | standby}]
[-pmos_bias_voltage <float>] [-nmos_bias_voltage <float>]
```

Description

The `create_nominal_condition` command infers voltage values for domains in a power state.

Arguments

The `create_nominal_condition` command has the following arguments:

`-name <name>`

Specifies the name of the nominal condition.

`-voltage <float>`

Specifies a voltage value for the nominal condition.

The following arguments of the `create_nominal_condition` command that do not have any effect on the rule behavior:

- -pmos_bias_voltage <float>
- -nmos_bias_voltage <float>
- -ground_voltage float
- -state {on | off | standby}

Commands to Specify Isolation Details

Following are the commands under this category:

- *define_isolation_cell*
- *create_isolation_rule*
- *update_isolation_rules*

CPF Check Rules

Specifying Constraints

Example of Using CPF / UPF Commands

define_isolation_cell

Syntax

```
define_isolation_cell
-cells <cell_list>
[-library_set <library_set>]
[-always_on_pins <pin_list>]
[ {-power_switchable <LEF_power_pin> | -ground_switchable
<LEF_ground_pin>}
-power <LEF_power_pin> -ground <LEF_ground_pin> ]
[-valid_location < from | to | on |off>]
[-non_dedicated]
{-enable pin | -no_enable {high|low|hold} }
```

Description

The `define_isolation_cell` command identifies the library cells in the .lib files that can be used as isolation cells.

NOTE: *The information specified with the* `define_isolation_cell` *command takes precedence over the library information. For details of the library attributes, refer to Using Constraints in the SpyGlass Power Verify Solution section.*

Arguments

The `define_isolation_cell` command has the following arguments:

`-cells <cell_list>`

Specifies a list of isolation cells.

`-enable <pin>`

Specifies enable pin of isolation cells.

`-no_enable {low|high|hold}`

Specifies that the isolation cell does not have an enable pin. This option also specifies the output of the cell when the supply for the switchable power (or ground) pin is powered down.

The following arguments of the `define_isolation_cell` command that do not have any effect on the rule behavior:

- -library_set <library_set>
- -always_on_pins <pin_list>
- -power_switchable <LEF_power_pin>
- -ground_switchable <LEF_ground_pin>
- -power <LEF_power_pin>
- -ground <LEF_ground_pin>
- -valid_location < from | to >
- -non_dedicated

create_isolation_rule

Syntax

```
create_isolation_rule
-name <name>
[-isolation_condition expression | -no_condition]
{-pins <pin_list> | -from <power_domain_list> | -to
<power_domain_list>}...
[-isolation_target <from|to>]
[-isolation_output <high | low | hold | tristate>]
[-exclude <pin_list>]
[-secondary_domain <power_domain>]
```

Description

The `create_isolation_rule` command creates a rule for adding an isolation cell.

Arguments

The `create_isolation_rule` command has the following arguments:

`-name <name>`

Specifies name of the isolation rule to be created.

`-isolation_condition <expression>`

Specifies the isolation condition for power domains.

NOTE: *Simple expressions or complex expressions only with | and ! (OR and NOT) operators are supported currently. For example,* `<isolation-sig-name>` *or* `!<isolation-sig-name>` *or* `<isolation-sig-name>|!<isolation-sig-name>`.

`-pins <pin_list>`

Specifies a list of input and output pins of power domains to be isolated.

-from <power_domain_list>

Specifies a list of power domains for which isolation checking is to be done at the output side.

-to <power_domain_list>

Specifies a list of power domains for which isolation checking is to be done at the input side

-isolation_target <from|to>

Specifies when this isolation rule applies.

If you specify the isolation target as `from`, the isolation rule is applied when power domain of the drivers of the specified pins is switched off. If you specify the isolation target as `to`, the isolation rule is applied when power domain of the loads of the specified pins is switched off.

[-isolation_output <high | low | hold>]

Specifies the expected steady-state value of power domain inputs or outputs.

[-exclude <pin_list>]

Specifies a list of pins that are already isolated. No isolation checking will be done on these pins.

The following arguments of the `create_isolation_rule` command that do not have any effect on the rule behavior:

- -no_condition
- -secondary_domain <power_domain>

update_isolation_rules

Syntax

```
update_isolation_rules
-names <rule_list>
{ -location <from | to>
| -cells <cell_list>
| -within_hierarchy instance
| -prefix <prefix>
| -open_source_pins_only}...
```

Description

The update_isolation_rules command updates an isolation rule.

Arguments

The update_isolation_rules command has the following arguments:

-names <rule_list>

Specifies list of isolation rule names to be updated.

-cells <cell_list>

Specifies a list of cell names to be used as isolation logic.

-within_hierarchy instance

Specifies where the level shifters should be located in the specified instance. The power domain of the specified instance must match the power domain of the selected location.

The following arguments of the update_isolation_rules command that do not have any effect on the rule behavior:

- location <from | to>
- -prefix <prefix>
- -open_source_pins_only

Commands to Specify Level Shifter

Following are the commands under this category:

- *define_level_shifter_cell*
- *create_level_shifter_rule*
- *update_level_shifter_rules*

CPF Check Rules

Level Shifter Rules

Specifying Constraints

Example of Using CPF / UPF Commands

define_level_shifter_cell

Syntax

```
define_level_shifter_cell
-cells <cell_list>
[-library_set <library_set>]
[-always_on_pins <pin_list>]
{ -input_voltage_range {voltage | voltage_range}
  -output_voltage_range {voltage | voltage_range}
| -ground_input_voltage_range {voltage | voltage_range}
  -ground_output_voltage_range {voltage|voltage_range}
| -input_voltage_range {voltage | voltage_range}
  -output_voltage_range {voltage | voltage_range}
  -ground_input_voltage_range {voltage | voltage_range}
 -ground_output_voltage_range {voltage|voltage_range}}
[-direction <up|down|bidir>]
[-enable <pin>]
[-input_power_pin <LEF_power_pin> ]
[-output_power_pin <LEF_power_pin>]
[-input_ground_pin LEF_ground_pin]
[-output_ground_pin LEF_ground_pin]
[-ground <LEF_ground_pin>]
[-power LEF_power_pin]
[-valid_location < from | to | either>
```

Description

The `define_level_shifter_cell` command identifies the library cells in the .lib files that can be used as level shifter cells.

NOTE: *A level shifter with an enable pin (clamp level shifter cell) that is tied to a constant value is not treated as an isolation cell.*

NOTE: *The information specified with the* `define_level_shifter_cell` *command takes precedence over the library information. For details of the library attributes, refer to Using Constraints in the SpyGlass Power Verify Solution section.*

Arguments

The `define_level_shifter_cell` command has the following arguments:

`-cells <cell_list>`

Specifies a list of cell names to be used as level shifter.

`-always_on_pins <pin_list>]`

Specifies a list of cell pins which must always be driven.

NOTE: *Prior to CPF version 1.0e, the name of this option was* `-always_on_pin`.

`-input_voltage_range {<voltage> | <voltage_range>}`

Identifies either a single input voltage or a range for the input power supply voltage that can be handled by the specified level shifter. You can also provide a voltage range as follows:

`lower_bound:upper_bound [:step]`

Specifying the lower bound, upper bound of the voltage range and voltage increment step, respectively. The voltage increment is optional.

`-output_voltage_range {<voltage> | <voltage_range>}`

Identifies either a single output voltage or a range for the output power supply voltage that can be handled by the specified level shifter. You can also provide a voltage range as follows:

`lower_bound:upper_bound [:step]`

Specifying the lower bound, upper bound of the voltage range and voltage increment step, respectively. The voltage increment is optional.

`-direction <up|down|bidir>`

Specifies whether the level shifter can be used between a lower and higher voltage, or vice versa.

`-input_power_pin <LEF_power_pin>`

Specifies the input-side supply terminal of level shifter cell.

-output_power_pin <LEF_power_pin>

Specifies the output-side supply terminal of level shifter cell.

-ground <LEF_ground_pin>

Identifies the name of the GROUND pin in the corresponding LEF cell.

The following arguments of the `define_level_shifter_cell` command that do not have any effect on the rule behavior:

- -library_set <library_set>
- -ground_input_voltage_range
- -ground_output_voltage_range
- -enable <pin>
- -valid_location < from | to | either >
- -power LEF_power_pin
- -input_ground_pin LEF_ground_pin
- -output_ground_pin LEF_ground_pin

create_level_shifter_rule

Syntax

```
create_level_shifter_rule
-name <rule-name>
{-pins pin_list | -from <domain_list> | -to <domain_list>}...
[-exclude pin_list]
```

Description

The `create_level_shifter_rule` command creates a rule for adding a level shifter.

Arguments

The `create_level_shifter_rule` command has the following arguments:

`-name <rule-name>`

Specifies name of the level shifter rule.

`-from <domain_list>`

Specifies a list of source domain names.

`-to <domain_list>`

Specifies a list of destination domain names.

NOTE: *User needs to specify both -to and -from for a level shifter else it would be an error.*

The following arguments of the `create_level_shifter_rule` command that do not have any effect on the rule behavior:

- -pins <pin_list>
- -exclude <pin_list>

update_level_shifter_rules

Syntax

```
update_level_shifter_rules
-names <rule_list>
{ -location <from | to>
| -within_hierarchy instance
| -cells <cell_list>
| -prefix <prefix>
```

Description

The update_level_shifter_rules command updates the details of a level shifter rules.

NOTE: *A level shifter with an enable pin (clamp level shifter cell) that is tied to a constant value is not treated as an isolation cell.*

Arguments

The update_level_shifter_rules command has the following arguments:

-names <rule_list>

Specifies a list of level shifter rules to be updated.

-location <from | to>

Specify the location of the level shifter. The default value is to.

-within_hierarchy instance

Specifies where the level shifters should be located in the specified instance. The power domain of the specified instance must match the power domain of the selected location.

-cells <cell_list>

Specifies a list of cell names to be used as level shifter.

The following arguments of the update_level_shifter_cell

command that do not have any effect on the rule behavior:

- -prefix <prefix>

Commands to Specify Power Switch

Following are the commands under this category:

- *define_power_switch_cell*
- *create_power_switch_rule*
- *update_power_switch_rule*

CPF Check Rules

Supply Rules

Specifying Constraints

Example of Using CPF / UPF Commands

define_power_switch_cell

Syntax

```
define_power_switch_cell
-cells <cell_list> [-library_set <library_set>]
-stage_1_enable <expression> [-stage_1_output <expression>]
[-stage_2_enable <expression> [-stage_2_output
<expression>]]
-type <footer|header>
[ -enable_pin_bias [float:]float]
[ -gate_bias_pin LEF_power_pin]
[ -power_switchable <LEF_power_pin> -power <LEF_power_pin>
| -ground_switchable <LEF_ground_pin> -ground
<LEF_ground_pin> ]
[ -on_resistance <float>]
[ -stage_1_saturation_current <float>] [ -
stage_2_saturation_current <float>]
[ -leakage_current <float> ]
```

Description

The `define_power_switch_cell` command identifies the library cells in the .lib files that can be used as power switch cells.

Arguments

The `define_power_switch_cell` command has the following arguments:

`-cells <cell_list>`

Specifies a list of cell names to be used as power switch cells.

`-stage_1_enable <expression>`

Specifies the stage 1 enable pin and its value.

NOTE: *Simple expressions or complex expressions only with | and ! (OR and NOT) operators are supported currently. For example,* `<enable-pin-name>` *or*

`!<enable-pin-name>` *or* `<enable-pin-name>|!<enable-pin-name>`.

-stage_2_enable <expression>

Specifies the stage 2 enable pin and its value.

NOTE: *Simple expressions or complex expressions only with | and ! (OR and NOT) operators are supported currently. For example,* `<enable-pin-name>` *or* `!<enable-pin-name>` *or* `<enable-pin-name>|!<enable-pin-name>`.

-power_switchable <LEF_power_pin>

Name of the power-out terminal.

-power <LEF_power_pin>

Name of the power-in terminal.

-ground_switchable <LEF_ground_pin>

Name of the power-out terminal.

-ground <LEF_ground_pin>

Name of the power-in terminal.

NOTE: *At a time, any one of the -power_switchable and -power or -ground_switchable and -ground arguments are given.*

-type <footer|header>

When the type is `footer`, `-ground_switchable_pin` is treated as the power-out terminal and `-ground` as the power-in terminal and the `-power_switchable_pin` and `-power` arguments are ignored.

When the type is `header`, `-power_switchable_pin` is treated as the power-out terminal and `-power` as power-in terminal.

The following arguments of the `define_power_switch_cell` command that do not have any effect on the rule behavior:

- -library_set <library_set>
- -stage_1_output <expression>

- -stage_2_output <expression>
- -on_resistance <float>
- -stage_1_saturation_current <float>
- -stage_2_saturation_current <float>
- -leakage_current <float>
- -enable_pin_bias [float:]float
- -gate_bias_pin LEF_power_pin

create_power_switch_rule

Syntax

```
create_power_switch_rule
-name <name>
-domain <power_domain>
{-external_power_net net | -external_ground_net net}
```

Description

The `create_power_switch_rule` command specifies how a single power switch must connect the external and internal power or ground nets for the specified power domain.

Arguments

The `create_power_switch_rule` command has the following arguments:

`-name <name>`

Specifies name of the power switch rule to be created.

`-domain <power_domain>`

Specifies name of the domain under which you are specifying the power switch cells.

`-external_power_net <net>`

Specifies the name of supply net used to infer parent always-on supply for the domain.

`-external_ground_net <net>`

Specifies the name of supply net used to infer parent always-on supply for the domain.

update_power_switch_rule

Syntax

```
update_power_switch_rule
-name <rule-name>
{ -enable_condition_1 <expression>
  [-enable_condition_2 <expression>]
  | -acknowledge_receiver <expression>
  | -cells <cell_list>
  | -gate_bias_net power_net
  | -prefix <string>
  | -peak_ir_drop_limit <float>
  | -average_ir_drop_limit <float>
}...
```

Description

The update_power_switch_rule command updates a power switch rule to add power switch logic.

Arguments

The update_power_switch_rule command has the following arguments:

-name <rule-name>

Name of the power switch rule to be updated.

-enable_condition_1 <expression>

Specifies an enable condition and enable value for the stage 1 enable pin of the power switch.

NOTE: *Simple expressions or complex expressions only with | and ! (OR and NOT) operators are supported currently. For example,* <enable-condition> *or* !<enable-condition> *or* <enable-condition>|!<enable-condition>.

-enable_condition_2 <expression>]

Specifies an enable condition and enable value for the stage 2 enable pin of the power switch.

NOTE: *Simple expressions or complex expressions only with | and ! (OR and NOT) operators are supported currently. For example,* `<enable-condition>` *or* `!<enable-condition>` *or* `<enable-condition>|!<enable-condition>`.

The following arguments of the `update_power_switch_rule` command that do not have any effect on the rule behavior:

- -acknowledge_receiver <expression>
- -cells <cell_list>
- -gate_bias_net power_net
- -prefix <string>
- -peak_ir_drop_limit <float>
- -average_ir_drop_limit <float>

Commands to Specify Retention

Following are the commands under this category:

- *define_state_retention_cell*
- *create_state_retention_rule*
- *update_state_retention_rules*

CPF Check Rules

Specifying Constraints

Example of Using CPF / UPF Commands

define_state_retention_cell

Syntax

```
define_state_retention_cell
-cells <cell_list> [-library_set <library_set>]
[-cell_type <string>]
[-always_on_pins <pin_list>]
[-clock_pin <pin>]
{ -restore_function expression |
  -save_function expression |
  -restore_function expression
   -save_function expression }
[-restore_check <expression>]
[-save_check <expression>]
[-always_on_components <component_list>]
[ {-power_switchable <LEF_power_pin> |
   -ground_switchable <LEF_ground_pin> |
   -power_switchable <LEF_power_pin>
    -ground_switchable <LEF_ground_pin> }
   -power <LEF_power_pin> -ground <LEF_ground_pin> ]
```

Description

The `define_state_retention_cell` command identifies the library cells in the .lib files that can be used as state retention cells.

NOTE: *The information specified with* `define_state_retention_cell` *command takes precedence over the library information. For details of the library attributes, refer to Using Constraints in the SpyGlass Power Verify Solution section.*

Arguments

The `define_state_retention_cell` command has the following arguments:

`-cells <cell_list>`

Specifies a list of cell names to be used as state retention cells.

-always_on_pins <pin_list>

Specifies names of the always-on pins of the state retention cell.

NOTE: *Prior to CPF version 1.0e, the name of this option was* `-always_on_pin`.

-cell_type <string>

Specifies a group name for the set of retention cells with same retention behavior to be clubbed into one group.

-clock_pin <pin>

Specifies name of the clock pin of the state retention cell.

-restore_function <expression>

Specifies a condition when the states of the registers need to be restored.

NOTE: *Simple expressions or complex expressions only with | and ! (OR and NOT) operators are supported currently. For example,* `<restore-pin-name>` *or* `!<restore-pin-name>` *or* `<restore-pin-name>|!<restore-pin-name>`.

-save_function <expression>

Specifies a condition when the states of the registers need to be saved.

NOTE: *Simple expressions or complex expressions only with | and ! (OR and NOT) operators are supported currently. For example,* `<save-pin-name>` *or* `!<save-pin-name>` *or* `<save-pin-name>|!<save-pin-name>`.

-power_switchable <LEF_power_pin>

Specifies the names of the Vdd pin (the pin to be connected to the switchable supply).

-power <LEF_power_pin>

Specifies the names of the Vddc pin (the pin to be connected to the always-on supply).

The following arguments of the `define_state_retention_cell` command that do not have any effect on the rule behavior:

- -library_set <library_set>
- -restore_check <expression>
- -save_check <expression>
- -ground_switchable <LEF_ground_pin>
- -ground <LEF_ground_pin>
- -always_on_components <component_list>

create_state_retention_rule

Syntax

```
create_state_retention_rule
-name <rule-name>
{ -domain <power_domain> | -instances <instance_list> }
[ -exclude instance_list ]
[ -restore_edge expr |
  -save_edge expr |
  -restore_edge expr -save_edge expr |
  -restore_level expr -save_level expr ]
[ -restore_precondition expr]
[-save_precondition expr]
[-target_type {flop | latch | both}]
[-secondary_domain domain]
```

Description

The `create_state_retention_rule` command creates a rule to replace selected or all registers in the given power domain with state retention registers.

Arguments

The `create_state_retention_rule` command has the following arguments:

`-name <rule-name>`

Specifies of the state retention rule.

`-domain <power_domain>`

Specifies name of the domain under which you are specifying the retention cell instances.

`-instances <instance_list>`

Specifies space-separated hierarchical instance name list (name of any instance or the top-level design unit) indicating the regions to be

checked.

-restore_edge <expression>

Specifies the condition when the states of the registers need to be restored. The expression is a function of pins. When the expression changes from `false` to `true`, the states are restored.

-save_edge <expression>

Specifies the condition when the states of the registers need to be saved. The expression is a function of pins. When the expression changes from `false` to `true`, the states are saved.

The following arguments of the `create_state_retention_rule` command that do not have any effect on the rule behavior:

- -restore_level expr -save_level expr
- -restore_precondition expr
- -save_precondition expr
- -target_type {flop|latch|both}
- -secondary_domain domain
- -exclude instance_list

update_state_retention_rules

Syntax

```
update_state_retention_rules
  -names <rule-list>
  {-cells <lib-cell> |
   -cell_type <string> |
   -set_reset_control }
```

Description

The update_state_retention_rules command updates the state retention rules.

Arguments

The update_state_retention_rules command has the following arguments:

-names <rule-list>

Specifies the names of the rules to be updated. The name can contain wildcards.

-cells <lib-cell>

Specifies the library cell to be used to map the flops.

NOTE: *Prior to CPF version 1.0e, the name of this option was* -cell.

-cell_type <string>

Specifies the class of library cells that can be used to map the flops.

The following arguments of the update_state_retention_rules command that do not have any effect on the rule behavior:

- -set_reset_control

Commands to Specify Always On Cells

Following is the command under this category:

- *define_always_on_cell*

CPF Check Rules

Specifying Constraints

Example of Using CPF / UPF Commands

define_always_on_cell

Syntax

```
define_always_on_cell
-cells <cell_list>
[-library_set <library_set>]
[ {-power_switchable LEF_power_pin |
   -ground_switchable LEF_ground_pin |
   -power_switchable LEF_power_pin
     -ground_switchable LEF_ground_pin}
  -power <LEF_power_pin> -ground <LEF_ground_pin> ]
```

Description

The `define_always_on_cell` command identifies the library cells in the .lib files that can be used as always-on cells.

NOTE: *The information specified with* `define_always_on_cell` *command takes precedence over the library information. For details of the library attributes, refer to Using Constraints in the SpyGlass Power Verify Solution section.*

Arguments

The `define_always_on_cell` command has the following arguments:

`-cells <cell_list>`

Specifies a list of cell names to be used as always-on cells.

NOTE: *All the buffers given using the command define_always_on_cell will be used as AON buffers as CPF does not have a separate AON buffer command.*

`-power <LEF_power_pin>`

Specifies the name of Vddc pin (the pin to be connected to the always-on supply).

NOTE: *This argument is used only for buffers.*

The following arguments of the `update_level_shifter_cell` command that do not have any effect on the rule behavior:

- -library_set <library_set>
- -power_switchable <LEF_power_pin>
- -ground_switchable <LEF_ground_pin>
- -ground <LEF_ground_pin>

Commands to Specify Power Clamp Cell (Diode Cells) Details

Following is the command under this category:

- *define_power_clamp_cell*

CPF Check Rules

Specifying Constraints

Example of Using CPF / UPF Commands

define_power_clamp_cell

Syntax

```
define_power_clamp_cell
  -cells <cell-list>
  -data <pin-name>
  -power <pin-name> [ -ground <pin-name> ]
  [ -library_set <library-set> ]
```

Description

The `define_power_clamp_cell` command specifies a list of diode cells used for power clamp control.

Arguments

The `define_power_clamp_cell` command has the following arguments:

`-cells <cell-list>`

Identifies the specified cells as diode cells.

The following arguments of the `define_power_clamp_cell` command that do not have any effect on the rule behavior:

- -data <pin-name>
- -power <pin-name>
- -ground <pin-name>
- -library_set <library-set>

Other Commands

Following are the commands under this category:

- *set_design*
- *end_design*
- *set_instance*
- *set_macro_model*
- *end_macro_model*
- *set_power_mode_control_group*
- *end_power_mode_control_group*
- *get_parameter*
- *include*
- *define_related_power_pins*

CPF Check Rules

Specifying Constraints

Example of Using CPF / UPF Commands

set_design

Syntax

```
set_design
 module
[-ports port_list]
[-honor_boundary_port_domain]
[-parameters parameter_value_list]
```

Description

The `set_design` command identifies the relationship between the power pins and data pins for cells that have multiple set of power and ground pins.

Arguments

The `set_design` command has the following arguments:

`module`

Specifies the name of the module to which the power information in the current CPF file applies.

`-ports port_list`

Specifies a list of virtual ports in the specified module.

Virtual ports do not exist in the RTL of this module but will be needed for the control signals of the low power logic such as isolation logic, state-retention logic, and so on.

`-honor_boundary_port_domain`

Specifies to treat each boundary port domain assignment as a design constraint at the top level after the block is instantiated at the top. At the top level, each corresponding hierarchical pin becomes a virtual leaf level driver or leaf level load and its power domain corresponds to the power domain definition for the boundary port. In this case, the netlist traversal stops at this pin

-parameters parameter_value_list

Specifies a list of cell names for which the relationship between the power pins and data pins is defined.
Use the following format for each parameter:

```
{parameter_name default_value}
```

The default value can be a number of a string.

end_design

Syntax

```
end_design
```

Description

The `end_design` command is used with a `set_design` command groups a number of CPF commands that apply to the current design or top design.

set_instance

Syntax

```
set_instance
[{instance
 [-design design | -model macro_model]
 |-of_macro macro_cell -model macro_model
 |-of_macro macro_cell }
 [-port_mapping port_mapping_list]
 [-domain_mapping domain_mapping_list]
 [-parameter_mapping parameter_mapping_list] ]
```

Description

The `set_instance` sets the scope to the specified instance or links a previously defined CPF model to the specified instance.

In case you have not specified `-design` or `-model` option, you need to specify `set_design` command or `set_macro_model` command also after the `set_instance` command. In that case, the instance specified with the `set_instance` command must be an instantiation of the module name specified with the `set_design` command or the cell name specified with the `set_macro_model` command.

Arguments

The `set_instance` command has the following arguments:

`instance`

Specifies an instance to which the current scope is to be set.

`-design design`

Specifies design with a previously loaded CPF description to be used for the specified instance.

`-model macro_model`

Specifies macro model with a previously loaded CPF description to be used

for the specified instance.

-port_mapping port_mapping_list

Specifies the mapping of the virtual ports specified in the `set_design` command to the parent-level drivers. Use the following format to specify a port mapping:

`{virtual_port parent_level_driver}`

-domain_mapping domain_mapping_list

Specifies the mapping of the domains in the current scope to the domains for the specified design or macro model. Use the following format to specify a domain mapping:

`{domain_in_child_scope domain_in_parent_level_scope}`

-parameter_mapping parameter_mapping_list

Specifies the mapping of the parameters specified in the `set_design` command to the local values. Use the following format to specify a parameter mapping:

`{parameter_name local_value}`

The following arguments of the `set_instance` command that do not have any effect on the rule behavior:

- -of_macro macro_cell

set_macro_model

Syntax

```
set_macro_model
 macro_cell
```

Description

The `set_macro_model` indicates the start of the CPF content of a custom IP. The following CPF commands are allowed in a macro model definition:

- create_isolation_rule
- create_nominal_condition
- create_power_domain
- create_power_mode
- create_power_switch_rule
- create_state_retention_rule
- update_power_domain

Arguments

The `set_macro_model` command has the following arguments:

`macro_cell`

Specifies the name of the macro for which the CPF description follows.

end_macro_model

Syntax

```
end_macro_model
```

Description

The `end_macro_model` is used with `set_macro_model` command, groups a number of CPF commands that apply to the described macro cell.

set_power_mode_control_group

Syntax

```
set_power_mode_control_group
 -name group
{ -domains domain_list |
  -groups group_list |
  -domains domain_list -groups group_list}
```

Description

The `set_power_mode_control_group` groups a list of power domains and other power mode control groups.

This command together with the end_power_mode_control_group command groups a set of CPF commands that define the power modes and power mode transitions that apply to this group only.

Arguments

The `set_power_mode_control_group` command has the following arguments:

`-name group`

Specifies the name of the power mode control group.

`-domains domain_list`

Specifies the list of power domains controlled by the power control manager associated with the specified group

`-groups group_list`

Specifies the list of power mode control groups whose power control manager is controlled by the power control manager associated with the specified group.

end_power_mode_control_group

Syntax

```
end_power_mode_control_group
```

Description

The `end_power_mode_control_group` is used with a `set_power_mode_control_group` command, groups a set of CPF commands that define the power modes and power mode transitions that apply to the group defined by the preceding `set_power_mode_control_group` command.

get_parameter

Syntax

```
get_parameter
 parameter_name
```

Description

The `get_parameter` returns the value of a predefined parameter in the current design. An error message will be given if the parameter is not defined for the current design.

Arguments

The `get_parameter` command has the following arguments:

`parameter_name`

Specifies a parameter name.

The parameter must have been defined with the `-parameters` option of the `set_design` command for the current design.

include

Syntax

```
include
 file
```

Description

The `include` includes a CPF file or a Tcl file within a CPF file.

The path name to the file can contain a period (.) to refer to current directory of the CPF file being read and ".." to refer to the parent directory of the current directory of the CPF file being read.

Note: This command differs from the source command in Tcl syntax where a period (.) always refers to the directory of the top level CPF file being read and ".." refers to the parent directory of the top level CPF file being read.

Arguments

The `include` command has the following arguments:

`file`

Specifies the path name of the file to be included.

define_related_power_pins

Syntax

```
define_related_power_pins
-data_pins pin_list
-cells cell_list
[ -library_set library_set ]
{ -power LEF_power_pin |
  -ground LEF_ground_pin  |
  -power LEF_power_pin -ground LEF_ground_pin }
```

Description

The `define_related_power_pins` command identifies the relationship between the power pins and data pins for cells that have multiple set of power and ground pins.

Arguments

The `define_related_power_pins` command has the following arguments:

`-cells <cell_list>`

Specifies a list of cell names for which the relationship between the power pins and data pins is defined.

`-data_pins <pin_list>`

Specifies a list of input or output data pins.

`-ground LEF_ground_pin`

Specifies the GROUND pin of the corresponding LEF cell.

`-power LEF_power_pin`

Specifies the POWER pin of the corresponding LEF cell.

The following arguments of the `define_related_power_pins` command that do not have any effect on the rule behavior:

- -library_set <library_set>

Appendix: UPF Commands

You can specify power specifications in a UPF file and provide it as an input. SpyGlass will read and parse the UPF commands and based on the power intent information provided with these commands, performs the rule checking.

UPF Version

The SpyGlass Power Verify solution provides support for (Unified Power Format) UPF commands Version 1.0, 2.0, and 2.1. By default, the UPF version is 1.0. However, you should set it to 2.0/2.1 when using UPF 2.0/2.1 commands by any one of the following:

- the *upf_version* command.
- version argument of the *load_upf* command.
- version argument of the *load_upf_protected* command.
- version argument of the *power_data* constraint.
- version argument of the `read_power_data` command of the SpyGlass project file.

You can set the `-version` argument in the *power_data* constraint as 1 or 1.0 (for UPF 1.0) and 2 or 2.0 (for UPF 2.0)/2.1 (for UPF 2.1). This would set the default version of the UPF. In case you have specified the `upf_version` command in the UPF, then this option will be overridden.

Using the find_objects UPF Command

The SpyGlass Power Verify solution supports the `-pattern`, `-object_type`, `-direction`, and `-transitive` arguments of the *find_objects* UPF command. To understand how to use this command, review the following UPF command specification:

```
create_power_domain PD1_domain
-elements [find_objects inst1 -pattern u_PD1 -object_type
inst]
```

In this case, the SpyGlass Power Verify solution searches for the `u_PD1` instance in `inst1`. If `u_PD1` is found, the `find_objects` command is replaced in the `-elements` argument, as shown in the following:

```
create_power_domain PD1_domain -elements [inst1/u_PD1]
```

If it is not found, an empty string replaces the `find_objects` command in the `-elements` argument.

There is support for wildcards. For example, for the following command, the SpyGlass Power Verify solution searches for port `Y`:

```
set_isolation ISO2 -domain PD1_domain -elements
[find_objects inst*/u_PD* -pattern Y -object_type port] -
isolation_power_net PD2_domain_supply
```

Suppose port `Y` is in `inst11/u_PD22`, the `find_objects` command is replaced as shown:

```
set_isolation ISO2 -domain PD1_domain
-elements [inst11/u_PD22] -isolation_power_net
PD2_domain_supply
```

The *find_objects* command has an attribute `-direction`. It is used to specify direction of the port and can only be specified when `object_type` is a port. It can have three values that are case insensitive:

```
-direction < in| out | inout >
```

NOTE: *Direction will be undefined if -object_type is not port.*

Example:

```
set_level_shifter ls1 -domain VD0 -applies_to outputs \
- elements \
[find_objects inst1 -pattern A* -object_type port -direction
in] \
-rule both  -location self
```

In the above example, only input ports that match the pattern “A*” will be in the elements list.

The *find_objects* command has following attributes:

- `-transitive`: By default, the value of this attribute is set to `FALSE` and the *find_objects* command searches only the current scope of the logic hierarchy. If the value of this attribute is specified as `TRUE`, the *find_objects* command searches the current scope and the entire dependant hierarchical sub-tree.

  ```
  -transitive < TRUE | FALSE >
  ```

  Example:

  ```
  foreach {elem} [find_objects u_PD1 -pattern buf* -
  object_type
  inst -transitive TRUE] {
  connect_supply_net VDD1 –ports {$elem/VDD}
  }
  ```

  In the above example, the value of the `–transitive` attribute is specified as `TRUE`, so the *find_objects* command searches for instances whose name starts with `buf` at instances that are present in `u_PD1` or at all hierarchical levels below `u_PD1`.

- `-leaf_only`: If this option is specified, then SpyGlass searches for those leaf level instances whose name matches with name specified in the `–pattern` option.
- `-non_leaf`: If this option is used, then SpyGlass searches for those non-leaf level instances whose name matches with the name specified in the `–pattern` option.
- `-model`: If this option is used, then SpyGlass searches for those instances whose master name matches with the name specified in `–pattern` option.

Supported SGDC Commands

For the UPF flow, SpyGlass also honors the following SGDC commands:

- *activity*
- *always_on_buffer*
- *assume_path*
- *assertion_signal*
- *cell_hookup*
- *cell_pin_info*
- *cell_tie_class*
- *clock*
- *multivt_lib*
- *non_pd_inputcells*
- *power_data*
- *power_down*
- *power_down_sequence*
- *pg_cell*
- *pg_pins_naming*
- *ram_instance*
- *ram_switch*
- *set_case_analysis*
- *special_cell*
- *switchoff_wrapper_instance*

UPF Check Rules

Example of Using CPF / UPF Commands

UPF Commands Supported by SpyGlass Power Verify

This section provides details of UPF commands supported by the SpyGlass Power Verify solution. The commands have been categorized in the following categories:

- *Commands to Specify Power Domain*
- *Commands to Specify Power Model*
- *Commands to Specify Isolation Details*
- *Commands to Specify Level Shifter*
- *Commands to Specify Retention Details*
- *Commands to Specify Power States*
- *Commands to Specify Power Switch*
- *Miscellaneous Commands*
- *Unsupported UPF Commands*
- *Inconsequential UPF Commands*

NOTE: *This document provides complete syntax of the UPF commands supported by SpyGlass. However, the arguments that do not have any effect on SpyGlass rule behavior or that are not supported by SpyGlass are not documented.*

UPF Check Rules

Example of Using CPF / UPF Commands

Commands to Specify Power Domain

Use the following UPF commands to specify power domain:

- *create_power_domain*
- *add_domain_elements*
- *set_domain_supply_net*
- *set_related_supply_net*
- *associate_supply_set*
- *create_supply_port*
- *create_supply_net*
- *create_supply_set*
- *connect_supply_net*
- *connect_supply_set*
- *set_pin_related_supply*
- *set_port_attributes*
- *set_repeater*
- *set_equivalent*
- *set_design_attributes*

UPF Check Rules

Example of Using CPF / UPF Commands

create_power_domain

Syntax

```
create_power_domain <domain_name>
[-elements <list>]
[-exclude_elements <list>]
[-include_scope]
[-scope <instance_name>]
[-supply <supply_set_handle supply_set_ref>]
[-available_supplies <supply_set_ref_list>]
[-update]
```

NOTE: *To use the arguments in blue, change the UPF Version to 2.0.*

Description

The `create_power_domain` command creates a power domain.

Arguments

The `create_power_domain` command has the following arguments:

<domain-name>

(Mandatory) Specifies the name of the voltage/power domain.

-elements <list>

(Optional) Space-separated name list of instances belonging to the voltage/power domain being specified.

-exclude_elements <list>

(Optional) Space-separated name list of instances to be excluded from this voltage/power domain being specified.

-include_scope

(Optional) Specifies that the top domain name specified with the `set_design_top` command should be taken as the name of the top-level design unit belonging to the voltage/power domain being specified.

-scope <instance_name>

(Optional) Specifies that domain is to be created with the specified hierarchy.

-supply <supply_set_handle supply_set_ref>

(Optional) Specifies the supply set handle and reference of the associated supply set. The supply set should be created with the *create_supply_set* command.

-available_supplies <supply_set_ref_list>

(Optional) Specifies a list of additional supply sets that are available for use by implementation tools to power cells inserted in this domain.

-update

(Optional) Signifies that the `domain_name` has already been defined.

add_domain_elements

Syntax

```
add_domain_elements <domain_name>
-elements <list>
```

Description

The `add_domain_elements` command adds design elements to a power domain.

Arguments

The `add_domain_elements` command has the following arguments:

`<domain_name>`

(Mandatory) Specifies the name of the voltage/power domain.

`-elements <list>`

(Mandatory) Space-separated name list of instances to be added to the voltage/power domain.

set_domain_supply_net

Syntax

```
set_domain_supply_net <domain_name>
-primary_power_net <supply_net_name>
-primary_ground_net <supply_net_name>
```

Description

The `set_domain_supply_net` command specifies a primary power and ground supply nets for a power domain.

Arguments

The `set_domain_supply_net` command has the following arguments:

<domain_name>

(Mandatory) Specifies the name of the voltage/power domain.

-primary_power_net <supply_net_name>

(Mandatory) Name of the primary power supply net.

-primary_ground_net <supply_net_name>

(Mandatory) Name of the primary ground supply net.

set_related_supply_net

Syntax

```
set_related_supply_net
-object_list <pin/port list>
[ -power <supply_net> ]
[ -ground <ground_net> ]
```

Description

The `set_related_supply_net` command relates leaf level or top level ports to power/ground supply.

This command takes precedence over the library attributes settings and `connect_supply_net` UPF command. In case of conflicting information, this command overwrites the specified port/pin relation with the `supply_net/ground_net` supplies.

Arguments

The `set_related_supply_net` command has the following arguments:

`-object_list <pin/port list>`

(Mandatory) Specifies a list of pins/port names to be related to the power/ground supply.

`-power <supply-net>`

(Optional) Specifies name of the power supply. If you do not specify the power supply, you must specify the ground supply. Alternatively, you can specify both the power and ground supply.

`-ground <ground-net>`

(Optional) Specifies name of the ground supply. If you do not specify the ground supply, you must specify the power supply. Alternatively, you can specify both the power and ground supply.

associate_supply_set

Syntax

```
associate_supply_set <supply-set-ref>
-handle <supply-set-handle>
```

Description

The `associate_supply_set` command associates a supply set to a power domain.

Arguments

The `associate_supply_set` command has the following arguments:

`<supply-set-ref>`

(Mandatory) Specifies the rooted name of the supply set to associate.

`-handle <supply-set-handle>`

(Mandatory) Specifies the supply set handle.

create_supply_port

Syntax

```
create_supply_port <port_name>
[-domain <domain_name>]
[-direction <in |out>]
```

Description

The `create_supply_port` command creates a supply port.

Arguments

The `create_supply_port` command has the following arguments:

<port_name>

(Mandatory) Specifies name of the supply port.

-domain <domain_name>

(Optional) Specifies the name of the voltage/power domain.

-direction <in | out>

(Optional) Specifies the direction of the supply port to be created.

create_supply_net

Syntax

```
create_supply_net <net_name>
[-domain <domain_name>]
[-reuse]
[-resolve <unresolved | one_hot | parallel |
parallel_one_hot>]
```

Description

The `create_supply_net` command creates a supply net.

Arguments

The `create_supply_net` command has the following arguments:

<net_name>

(Mandatory) Simple name of the supply net.

-domain <domain_name>

(Optional) Specifies the domain in which the supply net is created.

-reuse

(Optional) Specifies that a supply net that already exists needs to be reused.

-resolve <unresolved | one_hot | parallel | parallel_one_hot>

(Optional) Specifies the resolution mechanism that determines the state and voltage of the supply net when the net has multiple supply sources. By default, the behavior for resolution is the same as for unresolved.

create_supply_set

Syntax

```
create_supply_set <set_name>
[-function func-name [net_name]]
[-update]
```

Description

The `create_supply_set` command creates the supply set name within the active scope in the UPF name space.

Arguments

The `create_supply_set` command has the following arguments:

<set_name>

(Mandatory) Signifies the name of the supply set.

<function> func-name [net_name]

(Optional) Defines the function a supply net provides for this supply set.

<update>

(Optional) Signifies that this `create_supply_set` refers to a supply set that was previously defined.

connect_supply_net

Syntax

```
connect_supply_net <net_name>
[-ports <list>]
[-pins <list>]
[-pg_type <pg_type_list>]
[-cells <list>]
[-domain <domain_name>]
```

Description

The `connect_supply_net` command connects a supply net with a supply port.

Arguments

The `connect_supply_net` command has the following arguments:

<net_name>

(Mandatory) Specifies name of the supply net.

-ports <list>

(Optional) Specifies a list of port names to be connected to the supply net. For port names that have been specified twice, the port name that is specified with a wildcard would be given lower preference than the port name specified without the wildcard.

-pins <list>

(Optional) Specifies a list of pin names to be connected to the supply net.

-pg_type <pg_type_list>

(Optional) Specifies power/ground pin type.

-cells <list>

(Optional) Specifies a list of cells to be used for the specified –pg_type.

-domain <domain_name>

(Optional) Specifies domain names to be used for the specified –pg_type.

connect_supply_set

Syntax

```
connect_supply_set supply_set_ref
{-connect {supply_function {pg_type_list}}}*
[-elements element_list]
[-exclude_elements exclude_list]
[-transitive <TRUE | FALSE>]
```

Description

The `connect_supply_set` command connects a supply set to the specified elements.

Arguments

The `connect_supply_set` command has the following arguments:

<supply_set_ref>

(Mandatory) Specifies the rooted name of the supply set or a supply_set_handle.

-connect {supply_function <pg_type_list>}

(Mandatory) Defines automatic connectivity for a `supply_function` of the `supply_set_ref` as ports having the specified `pg_type_list` attributes.

-elements <element_list>

(Optional) Specifies a list of design element names to be added.

set_pin_related_supply

Syntax

```
set_pin_related_supply <library_cell>
-pins <list>
-related_power_pin <supply_pin>
-related_ground_pin <supply_pin>
```

Description

The `set_pin_related_supply` command defines the related power/ground pair for a library cell.

NOTE: *Related power pin / related ground pin information of the pins not provided in the library files can be specified using* `set_pin_related_supply` *command. Also, if you want to override the specified information, use this command.*

Arguments

This command has the following arguments:

`library_cell`

(Mandatory) Specifies the library cell where the supply nets are to defined.

NOTE: *You can also specify the top design name as the* `library_cell` *for* `set_pin_related_supply` *command to define the top ports working on a different power domain than the top power domain.*

`-pins <list>`

(Mandatory) Specifies a list of signal pins that is to have a related power/ground supply defined.

NOTE: *For a vector pin, part select of the bus is not supported for this argument.*

`-related_power_pin <supply_pin>`

(Mandatory) Specifies the supply pin of the cell to which the pin is related.

-related_ground_pin <supply_pin>

(Mandatory) Specifies the supply pin of the cell to which the pin is related.

set_port_attributes

Syntax

```
set_port_attributes
[-model <model_name>]
[-pg_type <pg_type_value>]
[-applies_to <inputs | outputs | both>]
[-feedthrough]
[-unconnected]
[-ports <port-list>]
[-related_power_port <port>]
[-related_ground_port <port>]
[-elements {<port_list|.>}]
[-driver_supply <supply_set_ref>]
[-receiver_supply <supply_set_ref>]
[-repeater_supply <supply_set_ref>]
```

NOTE: *To use the arguments in blue, change the UPF Version to 2.0.*

Description

The `set_port_attributes` command specifies information relevant to ports on the power domains interface. This is used to specify related power port, receiver supply and/or driver supply information for top-level ports.

Arguments

The `set_port_attributes` command has the following arguments:

`-model <model_name>`

(Optional) Specifies a module or library cell whose ports are to be attributed.

`-pg_type <pg_type_value>`

(Optional) Specifies the value of the `pg_type` for the port.

-applies_to <inputs | outputs | both>

(Optional) Indicates whether the specified input ports, output ports, or both are to be attributed.

-feedthrough

(Optional) Indicates that the specified ports are connected together internally to form a feedthrough.

-unconnected

(Optional) Indicates that the specified ports are not connected at all internally.

-ports <port_list>

(Optional) Specifies the list of ports to be attributed.

-related_power_port <port>

(Optional) Specifies the related power port for the attributed ports.

-related_ground_port <port>

(Optional) Specifies the related ground port for the attributed ports.

-elements {<port-list|.>} -applies_to both <inputs|outputs|both>

(Optional) Specifies the list of elements to be attributed.

-driver_supply <supply_set_ref>

(Optional) Specifies the supply set used by the drivers of the port.

-receiver_supply <supply_set_ref>

(Optional) Specifies the supply set used by the receivers of the port.

-repeater_supply <supply_set_ref>

(Optional) Specifies the supply set used by the repeater driving the port.

set_repeater

Syntax

```
set_repeater <repeater_name>
-domain <domain_name>
[-elements <list>]
[-source <source_supply> | <domain_name>]
[-sink <sink_supply | domain_name>]
[-use_equivalence < TRUE | FALSE>]
[-applies_to <inputs | outputs | both>]
[-repeater_supply_set <supply_set_ref>]
[-name_prefix <string>] [-name_suffix <string>]
[-update]
```

Description

The `set_repeater` command specifies a repeater (buffer) strategy.

Arguments

The `set_repeater` command has the following arguments:

<repeater_name>

(Mandatory) Specifies the name of the isolation rule to be mapped.

-domain <domain_name>

(Mandatory) Specifies the name of the power domain.

-elements <list>

(Optional) Specifies the hierarchical names of buffers to be repeated.

-source <source_supply | domain_name>

(Optional) Specifies the source supply or power domain. Use this argument to filter the ports receiving a net that are driven by logic powered by the supply set. When a domain name is used, it represents

the primary supply of that domain.

-sink <sink_supply | domain_name>

(Optional) Specifies the sink supply or power domain. Use this argument to filter the ports driving a net that fans out to logic powered by the supply set. When a domain name is used, it represents the primary supply of that domain.

-use_equivalence <TRUE | FALSE>

(Optional) Indicates whether to consider supply set equivalence. If `-use_equivalence` is not specified at all, the default is `-use_equivalence TRUE`; if `-use_equivalence` is specified without a value, the default value is `TRUE`.

-applies_to <inputs | outputs | both>

(Optional) Specifies the input ports, output port, or both types of ports of the power domain for which the strategy applies.

-repeater_supply_set <supply_set_ref>

(Optional) Specifies the supply set that powers the inserted buffer.

-name_prefix <string>

(Optional) Specified the name format (prefix) for inserted buffer cell instances or nets related to implementation of the strategy.

-name_suffix <string>

(Optional) Specified the name format (suffix) for inserted buffer cell instances or nets related to implementation of the strategy.

-update

(Optional) Signifies that the `set_repeater` command refers to the same `strategy_name` and `domain_name` that were previously defined.

set_equivalent

Syntax

```
set_equivalent
[-function_only]
[-nets <supply_net_name_list>]
[-sets <supply_net_name_list>]
```

Description

The `set_equivalent` command specifies that supply nets or supply sets are electrically or functionally equivalent.

Arguments

The `set_equivalent` command has the following arguments:

<function_only>

Specifies that the supplies are functionally equivalent rather than electrically equivalent.

-nets <supply_net_name_list>

Specifies the list of supply port and/or supply net names that are equivalent.

-sets <supply_net_name_list>

Specifies the list of supply set names that are equivalent.

set_design_attributes

Syntax

```
set_design_attributes
[-models <model_list>]
[-elements <element_list>]
[-exclude_elements exclude_list]
[-attributes {name value}]
[-is_macro_cell[<TRUE| FALSE>]]
[-is_leaf_cell[<TRUE| FALSE>]]
```

Description

The `set_design_attributes` command sets the specified attributes for models or elements.

Arguments

The `set_design_attributes` command has the following arguments:

`-models [model_list]`

Specifies a list of models to be attributed.

`-elements [element_list]`

Specifies a list of rooted names; instances, named processes, sequential regs, or signal names.

`-exclude_elements [element_list]`

Specifies a list of rooted names, for example, instances, named processes, sequential regs, or signal names to exclude from the effective_element_list.

`-attributes {name value}`

For the specified models or elements, associates the attribute `name` with the value of `value`.

-is_macro_cell [<TRUE| FALSE>]

- If -is_macro_cell is not specified at all, the default is FALSE.
- If -is_macro_cell is specified without a value, the default value is TRUE.
- Equivalent to -attribute {UPF_is_macro_cell value}.

-is_leaf_cell [<TRUE| FALSE>]

- If -is_leaf_cell is not specified at all, the default is FALSE.
- If -is_leaf_cell is specified without a value, the default value is TRUE.
- Equivalent to -attribute {UPF_is_leaf_cell value}.

Commands to Specify Power Model

Use the following UPF commands to specify power model:

- *begin_power_model*
- *apply_power_model*
- *end_power_model*

UPF Check Rules

Example of Using CPF / UPF Commands

begin_power_model

Syntax

```
begin_power_model power_model_name
[-for <model_list>]
```

Description

The `begin_power_model` command defines a power model.

Arguments

The `begin_power_model` command has the following arguments:

`[power_model_name]`

Specifies the name of the power model.

`-for [model_list]`

Specifies the names of the hard IP or macro cells to which the power model applies.

apply_power_model

Syntax

```
apply_power_model power_model_name
[-elements <element_list>]
[-supply_map {{lower_scope_handle upper_scope_supply_set}*}]
```

Description

The apply_power_model command connects the interface supply set handles of a previously loaded power model.

Arguments

The apply_power_model command has the following arguments:

[power_model_name]

Specifies the name of a previously defined power model.

-for [model_list]

Specifies the list of instances to which the specified power model applies.

-supply_map [{{lower_scope_handle upper_scope_supply_set}*}]

Specifies how the interface supply handles of the corresponding power model connect with the actual supply sets or supply set handles in the current scope.

Example

```
apply_power_model cellA –elements {I2} –supply_map {SSAH
PDTop.SS1} {SSBH PDTop.SS2}
```

In this example:

- Elements in the power model can be referenced by adding element name I2. For example, to reference the domain PD in the model at SoC level, I2/PD should be used.

- Element specified `I2` should be a leaf level object (or else "`set_option stop <module name of I2>`" should be used.
- Power model supply set `I1/PD.SSAH` is equivalent to SoC supply set referenced using `PDTop.SS1`.
- Power model Supply set `I2/PD.SSBH` is equivalent to SoC supply set referenced using `PDTop.SS2`.

end_power_model

Syntax

```
end_power_model
```

Description

The `apply_power_model` command terminates the definition of a power model. It returns a `1` if successful or raises a `TCL_ERROR` if not.

Commands to Specify Isolation Details

Use the following UPF commands to specify isolation details:

- *set_isolation*
- *set_isolation_control*
- *map_isolation_cell*
- *set_required_input_isolation*
- *use_interface_cell*

UPF Check Rules

Example of Using CPF / UPF Commands

set_isolation

Syntax

```
set_isolation <isolation_name>
-domain <domain_name>
[-isolation_power_net <net_name>]
[-isolation_ground_net <net_name>]
[-no_isolation]
[-force_isolation]
[-elements <list>]
[-clamp_value {< 0 | 1 | any | Z | latch | value>*}]
[-applies_to <inputs | outputs | both>]
[-name_prefix <string>] [-name_suffix <string>]
[-use_equivalence < TRUE | FALSE>]

[-source <source_domain_name | source_supply_ref>
[-sink <sink_domain_name | sink_supply_ref>]
[-isolation_supply_set <list_of_supply_sets>]
[-isolation_sense {<high | low>}]
[-location <automatic | self | other | fanout | fanin |
faninout | parent | sibling>]
[-diff_supply_only < TRUE | FALSE> ]
[-update]
```

NOTE: *To use the arguments in blue, change the UPF Version to 2.0.*

Description

The `set_isolation` command creates an isolation rule.

Arguments

The `set_isolation` command has the following arguments:

<isolation_name>

(Mandatory) Specifies the name of the isolation rule to be mapped.

-domain <domain_name>

(Mandatory) Specifies the name of the power domain.

-elements <list>

(Optional) Specifies the hierarchical names of pins to be isolated.

-name_prefix <string>

(Optional) Specified The name format (prefix) for generated isolation instances or nets related to implementation of the isolation strategy.

-name_suffix <string>

(Optional) Specified The name format (suffix) for generated isolation instances or nets related to implementation of the isolation strategy.

-source <source_domain_name | source_supply_ref>

(Optional) Specifies the source supply or power domain. Use this argument to filter the ports driving a net that fans out to logic powered by the supply set. When a domain name is used, it represents the primary supply of that domain.

-sink <sink_domain_name | sink_supply_ref>

(Optional) Specifies the sink supply or power domain. Use this argument to filter the ports driving a net that fans out to logic powered by the supply set. When a domain name is used, it represents the primary supply of that domain.

-use_equivalence <TRUE | FALSE>

(Optional) Indicates whether to consider supply set equivalence. If `-use_equivalence` is not specified at all, the default is `-use_equivalence TRUE`; if `-use_equivalence` is specified without a value, the default value is `TRUE`.

-applies_to <inputs | outputs | both>

(Optional) Specifies if the input ports, output port, or both types of ports of the power domain are to be isolated.

-isolation_supply_set <list_of_supply_sets>

(Optional) Defines the supply sets for the isolation logic inferred by this strategy.

NOTE: *Only the first supply set mentioned is used.*

-isolation_power_net <net_name>

(Optional) Defines the power supply net for the isolation logic inferred by this strategy.

NOTE: *An explicit supply connection specified for isolation logic using connect_supply_net is given higher priority than the isolation_power_net specified here.*

-isolation_ground_net <net_name>

(Optional) Defines the ground supply net for the isolation logic inferred by this strategy.

-no_isolation

(Optional) Specifies that isolation logic is not required for the elements specified with the `-elements` argument.

-force_isolation

(Optional) If `-force_isolation` is specified, then isolation is inferred for each port in the `effective_element_list` and the inferred isolation cells are not to be optimized away, even if such optimization does not change the behavior of the design. If neither `-no_isolation` nor `-force_isolation` is specified, then isolation is inferred for each port in the `effective_element_list`, and implementation tools are free to optimize away isolation cells that are redundant. It shall be an error if both `-force_isolation` and `-no_isolation` are specified.

-isolation_sense <high | low>

(Optional) Specifies the state of the signal list specified with

-isolation_signal argument. The default value is high for UPF version 1.0 and 2.0.

-clamp_value <0 | 1 | latch | Z>

(Optional) Specifies the expected steady-state value of power domain inputs or outputs.

NOTE: *SpyGlass does not support the value, Z, for this field.*

-location <automatic | self | fanout | parent | sibling>

(Optional) Specifies where the isolation cell is to be placed in the logical hierarchy. The default value is automatic.

NOTE: *The* -isolation_signal, -isolation_sense, *and* -location *options have been added in UPF Version 2.0.*

-diff_supply_only <TRUE | FALSE>

(Optional) Specifies the isolation behavior between the driver and receiver supply sets.

-update

(Optional) Signifies that the set_isolation command refers to an isolation_name that was previously defined.

set_isolation_control

Syntax

```
set_isolation_control <isolation_name>
-domain <domain_name>
-isolation_signal <signal_name>
[-isolation_sense <high | low>]
[-location <automatic | self | fanout | parent | sibling>]
```

Description

The `set_isolation_control` command specifies the control signal for an isolation rule.

Arguments

The `set_isolation_control` command has the following arguments:

<isolation_name>

(Mandatory) Specifies the name of the isolation rule.

-domain <domain_name>

(Mandatory) Specifies the name of the power domain.

-isolation_signal <signal_name>

(Mandatory) Specifies name of an isolation control signal.

-isolation_sense <high | low>

(Optional) Specifies the active level of isolation control signal.

-location <automatic | self | fanout | parent | sibling>

(Optional) Specifies where the isolation cell is to be placed in the logical hierarchy. The default value is `automatic`.

map_isolation_cell

Syntax

```
map_isolation_cell <isolation_name>
-domain <domain_name>
[-lib_cells <list>]
```

Description

The `map_isolation_cell` command maps an isolation rule to a library cell or range of library cells.

Arguments

The `map_isolation_cell` command has the following arguments:

`<isolation_name>`

(Mandatory) Specifies the name of the isolation rule to be mapped.

`-domain <domain_name>`

(Mandatory) Specifies the name of the power domain.

`-lib_cells <list>`

(Optional) Space-separated name list of library cells to be used as isolation cells.

set_required_input_isolation

Syntax

```
set_required_input_isolation -domain <domain_name>
```

Description

The `set_required_input_isolation` command specifies a domain that needs input isolation and hence it is mandatory to have its inputs isolated.

Arguments

The `set_required_input_isolation` command has the following argument:

<domain_name>

Specifies the name of the domain whose inputs need to be isolated

use_interface_cell

Syntax

```
use_interface_cell <interface_implementation_name>
-strategy <list_of_isolation_level_shifter_strategies>
-domain <domain_name>
-lib_cells <lib_cell_list>
```

Description

The `use_interface_cell` command specifies the implementation choices for isolation and level-shifting strategies through the -lib_cells argument.

Arguments

The `use_interface_cell` command has the following argument:

`-strategy <list_of_isolation_level_shifter_strategies>`

Specifies the isolation or level-shifter strategy, or a pair of isolation and level-shifter strategies, as defined by set_isolation and set_level_shifter.

`-domain <domain_name>`

Specifies the domain in which the strategies are defined.

`-lib_cells <lib_cell_list>`

Specifies a list of library cell names.

Commands to Specify Level Shifter

Use the following UPF commands to specify level shifters:

- *map_level_shifter_cell*
- *set_level_shifter*

UPF Check Rules

Example of Using CPF / UPF Commands

map_level_shifter_cell

Syntax

```
map_level_shifter_cell <level_shifter_name>
-domain <domain_name>
[-lib_cells <list>]
```

Description

The `map_level_shifter_cell` command maps a particular level shifter strategy to a library cell or range of library cells. If you have specified library cells in library name/cell name format, the library part is ignored.

NOTE: *A level shifter with an enable pin (clamp level shifter cell) that is tied to a constant value is not treated as an isolation cell.*

Arguments

The command has the following arguments:

`<level_shifter_name>`

(Mandatory) Specifies the name of the level shifter strategy specified in a *set_level_shifter* command for the specified domain.

`-domain <domain-name>`

(Mandatory) Specifies the name of the domain where the strategies are to be applied.

`-lib_cells <list>`

(Optional) Specifies the list of library cells to be used.

set_level_shifter

Syntax

```
set_level_shifter <level_shifter_name>
-domain <domain_name>
[-elements <list>]
[-applies_to <inputs | outputs | both>]
[-threshold <value> | <list>]
[-rule <low_to_high | high_to_low | both>]
[-location <self | parent | sibling | fanout | automatic>]
[-no_shift] [-name_prefix <string>] [-name_suffix <string>]
[-use_equivalence < TRUE | FALSE>]

[-update]
[-source <source_domain_name | source_supply_ref>]
[-sink <sink_domain_name | sink_supply_ref>]
[-input_supply_set <supply_set_ref>]
[-output_supply_set <supply_set_ref>]
[-force_shift]
```

NOTE: *To use the arguments in blue, change the UPF Version to 2.0.*

Description

The `set_level_shifter` command specifies a level shifter strategy.

Arguments

The command has the following arguments:

`level_shifter_name`

(Mandatory) Specifies the name of the level shifter strategy.

`-domain <domain_name>`

(Mandatory) Specifies the name of the domain where the specified

strategy is to be applied.

-elements <list>

(Optional) Specifies a list of design elements, input ports/pins, output ports/pins, and nets for which the specified strategy is to be applied.

-applies_to <inputs | outputs | both>

(Optional) Specifies whether the domain's input ports, output ports, or both are to be level shifted. The default is both.

-threshold value

(Optional) Specifies the voltage threshold (in volts) for determining when level shifters are required. The default is 0.

-rule <low_to_high | high_to_low | both>

(Optional) Specifies the type of level shifters that are required. The default is both.

-location <self | parent | sibling | fanout | automatic

(Optional) Specifies where the level shifter is to be placed in the logic hierarchy. The default is `automatic`.

-no_shift

(Optional) Specified with the `–elements` argument to prevent the insertion of level shifters on the specified ports/pins and nets. This can also be used with the options `-rule` and `-applies_to`.

-name_prefix <string>

(Optional) Specified The name format (prefix) for generated level-shifter instances or nets related to implementation of the shifting strategy.

-name_suffix <string>

(Optional) Specified The name format (suffix) for generated level-shifter instances or nets related to implementation of the shifting strategy.

-use_equivalence <TRUE | FALSE>

(Optional) Indicates whether to consider supply set equivalence. If -use_equivalence is not specified at all, the default is –use_equivalence TRUE; if -use_equivalence is specified without a value, the default value is TRUE.

-update

(Optional) Signifies that the set_level_shifter command refers to an level_shifter_name that was previously defined.

-source <source_domain_name | source_supply_ref>

(Optional) Specifies the source supply or power domain. Use this argument to filter the ports driving a net that fans out to logic powered by the supply set. When a domain name is used, it represents the primary supply of that domain.

-sink <sink_domain_name | sink_supply_ref>

(Optional) Specifies the sink supply or power domain. Use this argument to filter the ports driving a net that fans out to logic powered by the supply set. When a domain name is used, it represents the primary supply of that domain.

-input_supply_set <supply_set_ref>

(Optional) Specifies the supply set used to power the input portion of the level-shifter.

-output_supply_set <supply_set_ref>

(Optional) Specifies the supply set used to power the output portion of the level-shifter.

-force_shift

(Optional) Specifies unconditional insertion of a level-shifter.

NOTE: *The force_shift argument is supported with upf_version 1.0 when the lp_allow_force_shift_in_UPF1 parameter is set to yes.*

Commands to Specify Retention Details

Use the following UPF commands to specify retention details:

- *set_retention*
- *set_retention_control*
- *set_retention_elements*
- *map_retention_cell*

UPF Check Rules

Example of Using CPF / UPF Commands

set_retention

Syntax

```
set_retention <retention_name>
-domain <domain_name>
[-elements <list>]
[-save_signal {<net_name> <high | low | posedge | negedge>}]
[-restore_signal {<net_name> <high | low | posedge |
negedge>}]
[-update]
[-retention_power_net <net_name>]
[-retention_ground_net <net_name>]
[-retention_supply_set <retention_supply_name>]
[-no_retention]
```

NOTE: *To use the arguments in blue, change the UPF Version to 2.0.*

Description

The `set_retention` command creates an retention rule.

Arguments

The `set_retention` command has the following arguments:

`<retention_name>`

(Mandatory) Specifies the name of the retention rule.

`-domain <domain_name>`

(Mandatory) Specifies the name of the power domain.

`[-elements <list>]`

(Optional) Specifies space-separated hierarchical instance name list indicating the regions to be checked.

-save_signal {<net_name> <high | low | posedge | negedge>}

(Optional) Specifies the signal name and its active value, which causes the state of the sequential elements to be saved for state retention rule. The default sensitivity is `high`.

-restore_signal {<net_name> <high | low | posedge | negedge>}

(Optional) Specifies the signal name and its active value, which causes the state of the sequential elements to be restored for state retention rule. The default sensitivity is `high`.

NOTE: *The* `-save_signal` *and* `-restore_signal` *options have been added in UPF Version 2.0.*

-update

(Optional) Signifies that the `set_retention` command refers to a `retention_name` that was previously defined.

-retention_supply_set <retention_supply_name>

(Optional) Defines the supply set used to power the logic inferred by the `retention_name` strategy.

-retention_power_net <net_name>

(Optional) Defines the supply net used as the power for the retention logic inferred by this strategy.

-retention_ground_net <net_name>

(Optional) Defines the supply net used as the power for the retention logic inferred by this strategy.

-no_retention

(Optional) Prevents the inference of retention cells on the specified elements regardless of any other specifications.

set_retention_control

Syntax

```
set_retention_control <retention_name>
-domain <domain_name>
-save_signal {<net_name> <high | low | posedge | negedge>}
-restore_signal {<net_name> <high | low | posedge | negedge>}
```

Description

The `set_retention_control` command specifies a control signal and assertion for an retention rule.

Arguments

The `set_retention_control` command has the following arguments:

`<retention_name>`

(Mandatory) Specifies the name of the retention rule.

`-domain <domain_name>`

(Mandatory) Specifies the name of the power domain.

`-save_signal {<net_name> <high | low | posedge | negedge>}`

(Mandatory) Specifies the signal name and its active value which causes the state of the sequential elements to be saved for state retention rule.

`-restore_signal {<net_name> <high | low | posedge | negedge>}`

(Mandatory) Specifies the signal name and its active value which causes the state of the sequential elements to be restored for state retention rule.

set_retention_elements

Syntax

```
set_retention_elements retention_list_name
[-elements element_list]
[-exclude_elements exclude_list]
```

Description

The `set_retention_elements` command specifies a list of elements that can be used *set_retention* and *map_retention_cell* commands.

Arguments

The `set_retention_elements` command has the following arguments:

`retention_list_name`

(Mandatory) Specifies a simple name that is unique within the current scope.

`-elements element_list`

(Optional) Specifies a list of rooted names, for example, instances, named processes, or sequential regs.

`-exclude_elements exclude_list`

(Optional) Specifies a list of rooted names, for example, instances, named processes, or sequential regs.

map_retention_cell

Syntax

```
map_retention_cell <retention_name>
-domain <domain_name>
[-elements <list>]
[-lib_cells <list>]
[-lib_cell_type <lib_cell_type>]
```

Description

The `map_retention_cell` command maps a retention rule to a library cell or range of library cells.

Arguments

The `map_retention_cell` command has the following arguments:

`<retention_name>`

(Mandatory) Specifies the name of the retention rule to be mapped.

`-domain <domain_name>`

(Mandatory) Specifies the name of the voltage/power domain.

`-elements <list>`

(Optional) Specifies space-separated hierarchical instance name list indicating the regions to be checked.

`-lib_cells <list>`

(Optional) Space-separated name list of library cells to be used as state retention cells.

`-lib_cell_type <lib_cell_type>`

(Optional) Identifies cells that have retention behavior.

Commands to Specify Power States

Use the following UPF commands to specify power states:

- *create_pst*
- *add_pst_state*
- *add_port_state*
- *add_power_state*

UPF Check Rules

Example of Using CPF / UPF Commands

create_pst

Syntax

```
create_pst <table_name>
-supplies <list>
```

Description

The `create_pst` command creates a Power State Table (PST) by using a specific order of supply nets.

It is an error if the specified supply net has not already been created by the *create_supply_net* command.

Arguments

The `create_pst` command has the following arguments:

`<table_name>`

(Mandatory) Specifies the name of the PST.

`-supplies <list>`

(Mandatory) Specifies the list of supply nets or ports to include in each power state of the design.

add_pst_state

Syntax

```
add_pst_state <state_name>
  -pst <table_name>
  -state <supply_states>
```

Description

The add_pst_state command defines the states of each of the supply nets for one possible state of the design.

It is an error if the number of supply state names is different from the number of the supply nets within the Power State Table (PST).

This command also supports the * wild card character, as shown in the following example:

```
create_pst PST1 –supplies { VDD1 VDD2 GND}
add_pst_state state1 –pst PST1 { * * on }
```

The wild card expands to all the states mentioned in the add_port_state command for the supply.

Arguments

The add_pst_state command has the following arguments:

<state_name>

(Mandatory) Specifies the power state

-pst <table_name>

(Mandatory) Specifies the PST to which this state applies.

-state <supply_states>

(Mandatory) Specifies the list of states of the supply nets (specified through the -state argument of the add_pst_state command) in the corresponding order of the -supplies argument of the *create_pst* command.

add_port_state

Syntax

```
add_port_state <port_name>
-state {<name> <nom | <min nom max> | off>}
```

Description

The `add_port_state` command applies a state to a port.

Arguments

The `add_port_state` command has the following arguments:

<port_name>

(Mandatory) Specifies name of a port for which a state is to be added.

-state {<name> <nom | <min nom max> | off>}}

(Mandatory) Specifies the name of the state and value.

The state, `off`, specifies that the port can be switched off.

add_power_state

Syntax

```
add_power_state object_name
[-supply | -domain]
{-state state_name {[-supply_expr {boolean_function}][-
logic_expr {boolean_function}][-legal | -illegal]}
```

Description

The `add_power_state` command applies power state(s) to a power domain or supply set.

Arguments

The `add_power_state` command has the following arguments:

`object_name`

(Mandatory) Specifies simple name of a power domain or supply set.

`-state <state_name>`

(Mandatory) Specifies simple name of the state being defined or refined.

`-supply | -domain`

These arguments specify the kind of object to which this command applies. If `-supply` is specified, the `object_name` shall be the name of a supply set. If -domain is specified, the `object_name` shall be the name of a power domain. If neither is specified, the type of `object_name` determines the kind of object to which the command applies.

`-supply_expr {boolean_function}`

Specifies a boolean expression defined in terms of supply nets that evaluates to True when the object is in the state being defined.

-logic_expr {boolean_function}

Specifies a boolean expression defined in terms of supply sets or power domains that evaluate to True when the object is in the state being defined.

-legal | -illegal

Specifies the legality of the state as either legal or illegal, the default is -legal.

NOTE: *At least one of -supply_expr or -logic_expr is mandatory.*

Commands to Specify Power Switch

Use the following UPF commands to specify power switches:

- *create_power_switch*
- *map_power_switch*

UPF Check Rules

Supply Rules

Example of Using CPF / UPF Commands

create_power_switch

Syntax

```
create_power_switch <switch_name>
-output_supply_port {<port_name> <supply_net_name>}
{-input_supply_port {<port-name> <supply-net-name>}}*
{-control_port {<port-name> <net-name>}}
[-domain <domain_name>]
```

Description

The `create_power_switch` command defines the power switch in the power domain.

Arguments

The command has the following arguments:

<switch_name>

(Mandatory) Specifies the name of the power switch.

-output_supply_port <port-name> <supply-net-name>

(Mandatory) Specifies the output supply port of the power switch and the supply net to which the output port is connected.

-input_supply_port <port-name> <supply-net-name>

(Mandatory) Specifies the input supply port of the power switch and the supply net to which the input port is connected

-control_port {<port-name> <net-name>}

(Mandatory) Specifies the enable port of the power switch and the signal name (on signal) to which this enable port is connected.

NOTE: *SpyGlass supports only one input supply port and up to two control ports.*

-domain <domain_name>

(Optional) Specifies the name of the domain containing the switch.

The following arguments of the `create_power_switch` command do not have any effect on the rule behavior:

- -on_state {<state-name> <input-supply-port> {<boolean-function>}}*
- -on_state {<state-name> <input-supply-port> {<boolean-function>}}*
- -ack_delay {<port-name> <delay>}]*
- -off_state {<state-name> {<boolean-function>}}]*
- -error_state {<state-name> {<boolean-function>}}]*

map_power_switch

Syntax

```
map_power_switch <switch_name>
-domain <domain_name>
-lib_cells <list>
```

Description

The `map_power_switch` command specifies the cell that should be used as power switch.

Arguments

The command has the following arguments:

`<switch-name>`

(Mandatory) Specifies the switch name (as already defined by the *create_power_switch* command).

`-domain <domain-name>`

(Mandatory) Specifies the name of the domain where the power switch was created.

`-lib_cells <list>`

(Mandatory) Specifies the list of library cells to be used.

Miscellaneous Commands

The following UPF commands are also supported:

- *load_upf*
- *load_upf_protected*
- *set_design_top*
- *set_scope*
- *spyglass_run*
- *upf_version*
- *upf_extension*

UPF Check Rules

Example of Using CPF / UPF Commands

load_upf

Syntax

```
load_upf <upf_file_name>
[-scope <instance_name>]
[-version <upf_version>]
```

NOTE: *To use the arguments in blue, change the UPF Version to 2.0.*

Description

The `load_upf` command sets the scope to the specified instance and executes the specified UPF commands.

Arguments

The command has the following arguments:

`<upf_file_name>`

(Mandatory) Specifies the UPF file that needs to be executed.

`-scope <instance_name>`

(Optional) Defines the scope in which the UPF commands need to be executed.

`-version <upf_version>`

(Optional) Defines the *UPF Version* of the `upf_file_name`.

load_upf_protected

Syntax

```
load_upf_protected <upf_file_name>
[-scope <scope_name>]
[-version <upf_version>]
```

NOTE: *To use the arguments in blue, change the UPF Version to 2.0.*

Description

The `load_upf_protected` command loads a UPF file in a protected environment so that corruption of existing variables is prevented.

Arguments

The command has the following arguments:

`<upf_file_name>`

(Mandatory) Specifies the UPF file that needs to be sourced.

`-scope <scope_name>`

(Optional) Defines the scope for sourcing the file.

`-version <upf_version>`

(Optional) Defines the *UPF Version* of the `upf_file_name`.

set_design_top

Syntax

```
set_design_top <root>
```

Description

The `set_design_top` command specifies the root of the design.

Arguments

The command has the following arguments:

<root>

(Mandatory) Specifies the root of the design.

set_scope

Syntax

```
set_scope
  [<instance>]
```

Description

The `set_scope` command specifies the active UPF scope.

Arguments

The command has the following arguments:

`<instance>`

(Optional) Specifies the instance that, after completion of the command, becomes the active UPF scope. If you do not specify this argument, the scope is set to the top-level of the design.

spyglass_run

Syntax

```
if {[info exists spyglass_run] && $spyglass_run} {
#Commands to be executed with spyglass
}
```

Description

The `spyglass_run` Tcl variable is used with any UPF standard command. You can specify the above conditional block to specify the commands that should be executed only with SpyGlass. The `spyglass_run` Tcl variable is set to 1 by default, therefore, the IF block is evaluated as true.

upf_version

Syntax

```
upf_version <upf_ver_num>
```

Description

The `upf_version` command specifies the *UPF Version* for processing subsequent UPF commands.

Arguments

The command has the following arguments:

<upf_ver_num>

(Mandatory) Specifies the *UPF Version* for processing subsequent UPF commands.

upf_extension

Syntax

```
if
{[info exists upf_extension] && $upf_extension}}
 set_related_supply_net -power {VDD_IO} -object_list {PAD_i}
}
```

Description

The `upf_extension` Tcl variable is used with any non-UPF standard command, such as *set_related_supply_net*. You should wrap the non-UPF and UPF extension commands in the IF block, as shown above. The `upf_extension` Tcl variable is set to 1, by default, therefore, the IF block is evaluated as true.

Unsupported UPF Commands

The following UPF commands are not supported by the SpyGlass Power Verify solution:

- `create_composite_domain`
- `describe_state_transition`
- `merge_power_domains`
- `set_power_switch`
- `set_retention_elements`
- `use_interface_cell`
- `set_partial_on_translation`

UPF Check Rules

Example of Using CPF / UPF Commands

Inconsequential UPF Commands

The following UPF commands do not have any impact on verification:

- bind_checker
- connect_logic_net
- create_hdl2upf_vct
- create_logic_net
- create_logic_port
- create_upf2hdl_vct
- load_simstate_behavior
- name_format
- save_upf
- set_partial_on_translation
- set_simstate_behavior

UPF Check Rules

Example of Using CPF / UPF Commands

Example of Using CPF / UPF Commands

This section provides you with an example which allows you to create a CPF and UPF file, and use these files on the given design.

Consider a design, as shown in the following figure:

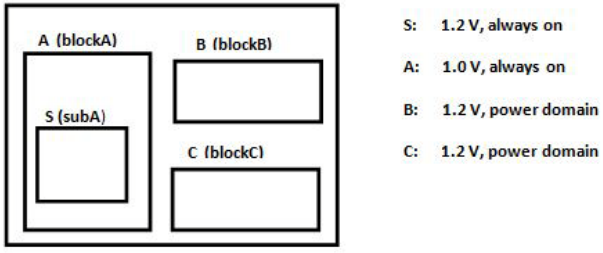


FIGURE 1.

Here, the majority of the design runs at 1.2 V, but one block, `blockA`, runs at a lower voltage, 1.0V. The sub-block, `subA`, of `blockA` is connected to the top-level voltage. Also consider that `blockB` and

`blockC` can be switched off.

You can now specify power specifications for the above design in a CPF or UPF file. Let's discuss each case in detail.

Checking the Design Using a CPF File

In order to check this design with a CPF file, you first need to define the hierarchical instance names of the voltage/power domains and their voltage values in the CPF file.

To define the hierarchical instance names of the voltage/power domains, perform the following steps:

1. Specify the default power domain

   Define the top-level design unit by using the `set_design` command. This top-level design unit is considered as the default power domain.

NOTE: *There should be only one default power domain in the design.*

2. Specify the other power domain(s)

   Define the other power domains by using the `-instances` option of the *create_power_domain* command.

To define voltage values for various power domains, create power modes by using the *create_power_mode* command. While creating a power mode, you can specify different voltage values for various power domains.

NOTE: *There is at least one power mode defined for RTL/gate-level netlist designs.*

Alternatively, you can specify the voltage values for various power domains through the main internal power net for power domain by using the `-internal_power_net` option of *update_power_domain* command.

You can also specify various rules for level-shifter checking, isolation logic, state retention, and power switch in the CPF file by using the *create_level_shifter_rule*, *create_isolation_rule*, *create_state_retention_rule*, and *create_power_switch_rule* commands, respectively. In addition, you also can specify cell information for various cells in the CPF file.

A sample CPF file is given below:

```
set_design top
# domain definitions
create_power_domain -name Vmain -instances A.S -default
create_power_domain -name VA -instances A
create_power_domain -name VB -instances B -shutoff_condition
{enB} -default_restore_edge
restorenet -default_save_edge !savenet
create_power_domain -name VC -instances C
```

```
-shutoff_condition {enC}

# level shifter rules
create_level_shifter_rule -name LS10_12_rule -from VA -to
Vmain
update_level_shifter_rules -names LS10_12_rule -cells
LS10_12
create_level_shifter_rule -name LS12_10_rule -from
Vmain -to VA
update_level_shifter_rules -names LS12_10_rule -cells
LS12_10

# isolation rules
create_isolation_rule -name VB_rule -from VB
-isolation_condition !enB -isolation_target from
create_isolation_rule -name VC_rule -from VC
-isolation_condition !enC -isolation_target from
update_isolation_rules -names {VB_rule VC_rule} -location
from -cells AND2X1_ISO

# state retention rule
create_state_retention_rule -name VB_SR_rule -domain VB
update_state_retention_rules -names VB_SR_rule -cell
DFFX1_RET

# power modes
create_nominal_condition -name high -voltage 1.2
create_nominal_condition -name low -voltage 1.0
create_nominal_condition -name off -voltage 0.0

create_power_mode -name all_on -domain_conditions  {
Vmain@high VA@low VB@high VC@high }
create_power_mode -name transmit -domain_conditions  {
Vmain@high VA@low VB@high VC@off  }
create_power_mode -name receive  -domain_conditions  {
Vmain@high VA@low VB@off  VC@high }

# isolation cells
```

```
define_isolation_cell -cells AND2X1_ISO -enable en

# level shifters
define_level_shifter_cell -cells LS10_12 -input_voltage_range
1.0 -output_voltage_range 1.2 -input_power_pin VDDI -ground
VSS
define_level_shifter_cell -cells LS12_10 -input_voltage_range
1.2 -output_voltage_range 1.0 -input_power_pin VDDI -ground
VSS

# state retention cells
define_state_retention_cell -cells DFFX1_RET -restore_function
!RES -save_function SAVE

# always on cells
define_always_on_cell -cells BUFX2_AON

end_design
```

Checking the Design Using a UPF File

Like CPF, to check this design with a UPF file, you first need to define the hierarchical instance names of the voltage/power domains and their voltage values in the UPF file.

To define the hierarchical instance names, perform the following steps:

1. Specify the default power domain

   Define the top-level design unit by using the `set_design_top` command. This top-level design unit is considered as the default power domain.

2. Specify the other power domain(s)

   Define the other power domains by using the `-elements` option of the *create_power_domain* command.

To specify the voltage value for various power domains, perform the following steps:

1. Create supply ports for the current power domain by using the *create_supply_port* command.
2. Specify the state information to the supply port by using the *set_pin_related_supply* command.
3. Create supply nets for power domains in the design by using the *create_supply_net* command.
4. Connect supply nets with supply port by using the *connect_supply_net* command.
5. Set the default primary power and ground supply nets for a power domain by using the *set_domain_supply_net* command.

NOTE: *A power domain is functional only if the primary power and ground supply nets are specified with the* `set_domain_supply_net` *command.*

You can also specify various rules for isolation-logic checking, state retention, and power switch in the UPF file. In addition, you can specify cell information for various cells.

A sample UPF file is given below:

```
set_design_top top

# power domain definitions
```

```
create_power_domain Vmain -elements A/S -include_scope
create_power_domain VA -elements A
create_power_domain VB -elements B
create_power_domain VC -elements C

# create supply ports
create_supply_port VDDH
add_port_state VDDH -state {on_state 1.2}

create_supply_port VDDS
add_port_state VDDS -state {on_state 1.2} -state {off_state
off}

create_supply_port VDDL
add_port_state VDDL -state {on_state 1.0}

create_supply_port VSS
add_port_state VSS -state {on_state 0.0}

# create supply nets
create_supply_net VDDH -domain Vmain
create_supply_net VDDL -domain VA
create_supply_net VDDSB -domain VB
create_supply_net VDDSC -domain VC
create_supply_net VSS -domain Vmain

# connect supply nets
connect_supply_net VDDH -ports VDDH
connect_supply_net VDDSB -ports VDDS
connect_supply_net VDDSC -ports VDDS
connect_supply_net VDDL -ports VDDL

# set primary supply nets
set_domain_supply_net Vmain –primary_power_net VDDH \
-primary_ground_net VSS
set_domain_supply_net VA -primary_power_net VDDL \
-primary_ground_net VSS
set_domain_supply_net VB -primary_power_net VDDSB \
```

```
-primary_ground_net VSS
set_domain_supply_net VC -primary_power_net VDDX \
-primary_ground_net VSS

# Defining Isolation Stategies :
Case1 : General strategy on domain-outputs
++++++++++++++++++++++++++++++++++++++++
set_isolation VB_rule -domain VB -applies_to outputs
set_isolation_control VB_rule \
            -domain VB \
            -isolation_signal enB \
             -isolation_sense low
map_isolation_cell VB_rule -domain VB -lib_cells AND2X1_ISO

Case2 : Strategy via supply-set:
++++++++++++++++++++++++++++++
create_supply_set ss1 -function {power VDDH} \
                      -function {ground VSS}
associate_supply_set ss1 -handle Vmain.primary
set_isolation VC_rule -domain VC \
                      -applies_to output \
                      -source ss1
OR
set_isolation VC_rule1 -domain VC -applies_to output -sink
ss1
OR
set_isolation VC_rule2 -domain VC \
                       -applies_to output \
                       -sink ss1 \
                       -diff_supply_only true

Defining Level-shifter Strategies:
Case1 : General strategy
set_level_shifter ls1 -domain VC \
                      -applies_to output \
                      -rule both \
                      -location fanout
Case2 :  Strategy via source/sink
```

```
Note  :power-domain to be specified with sourc/sink(UPF
2.0)
     power-domain or supply-set can be specified with
     source/sink (UPF 2.1)
set_level_shifter ls1 -domain VC \
                      -applies_to output \
                      -rule both \
                      -location fanout \
                       -source VA
OR
set_level_shifter ls1 -domain VC \
                      -applies_to output \
                      -rule both \
                      -location fanout \
                      -sink VA \
                      -source VC
```

Defining Retention Strategies:

```
# retention rules
set_retention VB_SR_rule -domain VB
set_retention_control VB_SR_rule -domain VB \
                 -save_signal {savenet low} \
                  -restore_signal {restorenet high}
map_retention_cell VB_SR_rule -domain VB \
                       -lib_cells DFFX1_RET
```

 # power-switch

```
create_power_switch psw1 \
              -output_supply_port {VDDG VDDX}   \
              -input_supply_port {VDD VDDH}    \
              -control_port {SLEEP ON1}   \
                      -ack_port {SLEEPOUT psw_ack}     \
                      -domain VC
add_port_state psw1/VDDG -state {on_state 1.0} -state
{off_state off}
map_power_switch psw1 -domain VC -lib_cells {PS_SWITCH}

# PST table - specifying power-switch via supply-net
```

```
create_pst pt1 -supplies {VDDH VDDL VDDSB VDDSC VSS VDDX}
add_pst_state s1 -pst pt1 {on_state on_state on_state
on_state on_state off_state}

# PST table - specifying power-switch via switch supply-port
#create_pst pt1 -supplies {VDDH VDDL VDDS psw1/VDDG VSS}
#add_pst_state s1 -pst pt1 {on_state on_state on_state
off_state on_state}
```

Appendix: List of Rules With Applicable Design Stages

The following sections list the rules along with their respective applicable design stages.

- *Debug Rules*
- *SoC Abstraction Rules*

Level Shifter Rules

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LPPLIB04	NO	NO	YES	YES	YES	YES	YES
LPPLIB05	NO	NO	YES	YES	NO	NO	YES
LPPLIB07	NO	NO	YES	YES	YES	YES	YES
LPPLIB08	NO	NO	YES	YES	NO	NO	YES
LPLSH01	YES	YES	YES	YES	YES	YES	YES
LPLSH02	YES	YES	YES	YES	YES	YES	YES
LPLSH03	YES	YES	YES	YES	YES	YES	YES
LPLSH04	YES	YES	YES	YES	YES	NO	NO
LPLSH05	YES	YES	YES	YES	YES	NO	NO
LPLSH05A	YES	YES	YES	YES	YES	NO	NO
LPLSH05B	YES	YES	YES	YES	YES	NO	NO
LPLSH06	YES	YES	YES	YES	YES	NO	NO
LPLSH07	YES	YES	YES	YES	YES	NO	NO
LPLSH08	YES	YES	YES	YES	YES	NO	NO
LPSVM04	NO	YES	YES	YES	YES	YES	YES
LPSVM04A	NO	YES	YES	YES	YES	YES	YES
LPSVM04B	NO	YES	YES	YES	YES	YES	YES
LPSVM04C	NO	YES	YES	YES	YES	YES	YES
LPSVM04D	NO	YES	YES	YES	YES	YES	YES
LPSVM04E	NO	YES	YES	YES	YES	YES	YES
LPSVM17	NO	YES	YES	YES	YES	YES	YES
LPSVM24	NO	YES	YES	YES	YES	YES	YES
LPLIB_check04	YES	YES	YES	YES	YES	YES	YES

Isolation Logic Rules

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LPISO01	YES	YES	YES	YES	YES	YES	YES
LPISO02	YES	YES	YES	YES	YES	YES	YES
LPISO04	YES	YES	YES	YES	YES	YES	YES
LPISO04A	YES	YES	YES	YES	YES	YES	YES
LPISO04B	YES	YES	YES	YES	YES	YES	YES
LPISO04C	YES	YES	YES	YES	YES	YES	YES
LPISO04D	YES	YES	YES	YES	YES	YES	YES
LPISO05	YES	YES	YES	YES	YES	YES	NO
LPISO05A	YES	YES	YES	YES	YES	YES	NO
LPISO05B	YES	YES	YES	YES	YES	YES	NO
LPISO06	YES	YES	YES	YES	YES	YES	YES
LPISO06A	YES	YES	YES	YES	YES	YES	YES
LPISO06B	YES	YES	YES	YES	YES	YES	YES
LPISO07	YES	YES	YES	YES	YES	NO	NO
LPSVM08	NO	YES	YES	YES	YES	YES	YES
LPSVM08A	NO	YES	YES	YES	YES	YES	YES
LPSVM08B	NO	YES	YES	YES	YES	YES	YES
LPSVM08C	NO	YES	YES	YES	YES	YES	YES
LPSVM09	NO	YES	YES	YES	YES	YES	YES
LPSVM10	NO	YES	YES	YES	YES	YES	YES
LPSVM12	NO	YES	YES	YES	YES	YES	YES
LPSVM12A	NO	YES	YES	YES	YES	YES	YES
LPSVM12B	NO	YES	YES	YES	YES	YES	YES
LPSVM15	NO	YES	YES	YES	NO	NO	YES
LPSVM22	NO	YES	YES	YES	YES	YES	YES
LPSVM26	NO	YES	YES	YES	YES	YES	YES
LPSVM28	NO	YES	YES	YES	NO	NO	YES

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LPSVM31	NO	YES	YES	YES	YES	YES	YES
LPSVM47	NO	YES	YES	YES	YES	YES	YES
LPSVM48	NO	YES	YES	YES	NO	NO	YES
LPSVM50	NO	YES	YES	YES	NO	NO	YES
LPSVM51	NO	YES	YES	YES	YES	YES	YES
LPSVM52	NO	YES	YES	YES	NO	NO	YES
LPSVM55	NO	YES	YES	YES	NO	NO	YES
LPSVM60	NO	YES	YES	YES	YES	YES	YES
LPLIB_check01	YES	YES	YES	YES	YES	YES	YES
LPLIB_check02	YES	YES	YES	YES	YES	YES	YES
LPLIB_check03	YES	YES	YES	YES	YES	YES	YES

Always-on Logic Rules

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LPAON01	NO	YES	YES	YES	YES	NO	YES
LPAON02	NO	YES	YES	YES	YES	NO	YES
LPPLIB11	NO	NO	YES	YES	YES	NO	YES
LPSVM53	NO	YES	YES	YES	NO	NO	YES
LPSVM54	YES	YES	YES	YES	NO	NO	YES
LPSVM40	NO	YES	YES	YES	YES	NO	YES

State Retention Rules

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LPPLIB10	NO	NO	YES	YES	YES	YES	YES
LPSVM38	NO	YES	YES	YES	YES	YES	YES
LPSVM56	NO	YES	YES	YES	YES	YES	YES

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LPSVM56A	NO	YES	YES	YES	NO	YES	YES
LPSVM56B	NO	YES	YES	YES	YES	YES	YES
LPSVM57	NO	YES	YES	YES	YES	YES	YES
LPSVM58	YES	YES	YES	YES	NO	NO	YES
LPSVM59	NO	YES	YES	YES	YES	YES	YES
LPRET01	YES	YES	YES	YES	YES	NO	NO
LPRET02	YES	YES	YES	YES	YES	NO	NO
LPRET03	YES	YES	YES	YES	YES	NO	NO
LPRET03A	YES	YES	YES	YES	YES	NO	NO
LPRET03B	YES	YES	YES	YES	YES	NO	NO
LPRET04	YES	YES	YES	YES	YES	NO	NO
LPRET04A	YES	YES	YES	YES	YES	NO	NO
LPRET04B	YES	YES	YES	YES	YES	NO	NO

Connection Rules

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LPCONN01	NO	NO	YES	YES	YES	YES	YES
LPCONN02	YES	YES	YES	YES	YES	NO	NO
LPCONN03	YES	YES	YES	YES	YES	YES	NO
LPCONN04	YES	YES	YES	YES	YES	NO	NO
LPCONN04A	YES	YES	YES	YES	YES	NO	NO
LPCONN04B	YES	YES	YES	YES	YES	NO	NO
LPCONN04C	YES	YES	YES	YES	YES	NO	NO
LPCONN04D	YES	YES	YES	YES	YES	NO	NO
LPCONN04E	NO	YES	YES	YES	YES	NO	NO
LPCONN05	YES	YES	YES	YES	YES	NO	NO
LPCONN05A	YES	YES	YES	YES	YES	NO	NO

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LPCONN05B	YES	YES	YES	YES	YES	NO	NO
LPCONN05C	YES	YES	YES	YES	YES	NO	NO
LPCONN06	YES	YES	YES	YES	YES	NO	NO
LPCONN07	YES	YES	YES	YES	YES	NO	NO
LPCONN07A	YES	YES	YES	YES	YES	NO	NO
LPCONN07B	YES	YES	YES	YES	YES	NO	NO
LPCONN08	YES	YES	YES	YES	YES	NO	NO
LPCONN09	YES	YES	YES	YES	YES	NO	NO
LPSVM49	NO	YES	YES	YES	NO	NO	YES
LP_SPECIAL_PIN_CONNECTION	YES	YES	YES	YES	NO	NO	NO

Supply Rules

Rule	RTL	NETLIST	PG NETLIST	DEF / LEF	UPF	CPF	SGDC
LPPLIB06	NO	NO	YES	YES	YES	YES	YES
LPPLIB13	NO	NO	YES	YES	NO	NO	YES
LPPLIB14	NO	NO	YES	YES	YES	YES	YES
LPPLIB15	NO	NO	YES	YES	YES	YES	YES
LPPLIB16	NO	NO	YES	YES	YES	YES	YES
LPPLIB17	NO	NO	YES	YES	YES	YES	YES
LPPLIB18	NO	NO	YES	YES	YES	NO	YES
LPPLIB18A	NO	NO	YES	YES	YES	NO	YES
LPPLIB18B	NO	NO	YES	YES	YES	NO	YES
LPPLIB19	NO	NO	YES	YES	YES	NO	NO
LPPLIB19A	NO	NO	YES	YES	YES	NO	NO
LPPLIB20	NO	YES	YES	YES	NO	NO	YES

Rule	RTL	NETLIST	PG NETLIST	DEF / LEF	UPF	CPF	SGDC
LPSUP01	NO	YES	YES	YES	NO	NO	YEs
LPSUP03	NO	YES	YES	NO	YES	NO	NO

Automatic Fix Rules

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LPSVM23	YES	NO	NO	NO	NO	NO	YES
LPSVM30	YES	NO	NO	NO	NO	NO	YES

Fine Grain Power Gate (MTCMOS) Rules

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LPSVM33	NO	YES	YES	YES	NO	NO	YES
LPSVM33A	NO	YES	YES	YES	NO	NO	YES
LPSVM34	NO	YES	YES	YES	NO	NO	YES
LPSVM35	NO	YES	YES	YES	NO	NO	YES
LPSVM36	NO	YES	YES	YES	NO	NO	YES

Special Purpose Rules

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LPPLIB12	NO	NO	YES	YES	NO	NO	YES
LPPSW01	NO	YES	YES	YES	YES	YES	YES
LPPSW02	NO	YES	YES	YES	YES	YES	NO
LPPSW03	NO	YES	YES	YES	YES	YES	YES
LPPSW04	NO	YES	YES	YES	YES	YES	YES
LPSVM29	NO	YES	YES	YES	NO	NO	YES
LPSVM37	NO	YES	YES	YES	NO	NO	YES

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LPSVM41	NO	YES	YES	YES	NO	NO	YES
LPSVM42	YES	YES	YES	YES	NO	NO	YES
LPSVM43	YES	YES	YES	YES	NO	NO	YES
LPSVM44	NO	YES	YES	YES	NO	NO	YES
LPSVM45	NO	YES	YES	YES	YES	YES	YES
LPSVM46	NO	YES	YES	YES	YES	YES	YES
LPTIE01	NO	YES	YES	YES	YES	YES	YES
LPTIE02	NO	YES	YES	YES	YES	YES	YES
LP_BLACKBOX_CHECK	NO	YES	YES	YES	YES	YES	YES
LP_BLACKBOX_CHECK	YES	NO	NO	NO	YES	NO	NO
LP_MULTI_DOMAIN_CROSSING_CHECK	YES	YES	NO	NO	YES	NO	NO
LP_INTERMEDIATE_DOMAIN_CROSSING_CHECK	YES	YES	YES	NO	YES	NO	NO

Powerdata Checking Rules

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LP_POWERDATA_CHECK	YES	YES	YES	YES	YES	YES	YES
LP_POWERDATA_INFO	YES	YES	YES	YES	YES	YES	YES
LP_POWERDATA_READ				YES	YES	YES	YES

Detailed Reporting Rules

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LP_DECOMPILE_CONSTR	YES	YES	YES	YES	YES	YES	YES
LP_CROSSING_DATA	YES	YES	YES	YES	YES	NO	YES

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LP_LIB_DATA	YES	YES	YES	YES	YES	NO	YES
LpWildCardMatchReport	YES	YES	YES	YES	YES	NO	NO
PairWiseVDCrossing	YES	YES	YES	YES	YES	NO	YES
vdPDInfo	YES	YES	YES	YES	YES	NO	NO
LP_ISO_REPORT	YES	YES	YES	YES	YES	NO	NO
LP_LSH_REPORT	YES	YES	YES	YES	YES	NO	NO

Constraints Checking Rules

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LP_CHECK_CONSTR	YES	YES	YES	YES	YES	NO	YES
LP_SGDC_CHECKS	YES	YES	YES	YES	YES	NO	YES
LpParamSanityCheck	YES	YES	YES	YES	YES	YES	YES
SGDC_lowpower05	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower06	YES	YES	YES	YES	NO	NO	YES
SGDC_voltagedomain01	YES	YES	YES	YES	NO	NO	YES
SGDC_voltagedomain02	YES	YES	YES	YES	NO	NO	YES
SGDC_voltagedomain03	YES	YES	YES	YES	NO	NO	YES
SGDC_voltagedomain04	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower07	YES	YES	YES	YES	NO	NO	YES
SGDC_voltagedomain05	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower09	YES	YES	YES	YES	NO	NO	YES
SGDC_voltagedomain06	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower12	YES	YES	YES	YES	NO	NO	YES
SGDC_voltagedomain07	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower15	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower17	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower18	YES	YES	YES	YES	NO	NO	YES

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
SGDC_lowpower19	YES	YES	YES	YES	NO	NO	YES
SGDC_voltagedomain08	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower23	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower24	YES	YES	YES	YES	NO	NO	YES
SGDC_supply01	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower30	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower31	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower32	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower40	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower47	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower48	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower52	YES	YES	YES	YES	NO	NO	NO
SGDC_lowpower59	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower60	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower61	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower62	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower65	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower66	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower67	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower68	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower69	YES	YES	YES	YES	NO	NO	NO
SGDC_lowpower71	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower72	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower75	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower77	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower78	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower82	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower85	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower86	YES	YES	YES	YES	NO	NO	YES

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
SGDC_lowpower87	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower89	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower90	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower91	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower92	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower93	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower94	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower95	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower96	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower97	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower98	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower99	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower100	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower101	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower103	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower104	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower105	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower107	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower108	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower109	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower110	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower112	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower113	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower114	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower115	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower116	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower118	YES	YES	YES	YES	NO	NO	YES
SGDC_lowpower_RuleReq	YES	YES	YES	YES	NO	NO	NO

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
SGDC_powerdomainoutputs01	YES	YES	YES	YES	NO	NO	YES
SGDC_powerdomainoutputs02	YES	YES	YES	YES	NO	NO	YES

Electrical Checks Rules

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LPERC01	NO	NO	YES	YES	YES	NO	YES
LPERC01A	NO	NO	YES	YES	YES	NO	YES
LPERC01B	NO	NO	YES	YES	YES	NO	YES
LPERC01C	NO	NO	YES	YES	YES	NO	YES
LPERC02	NO	NO	YES	NO	NO	NO	YES
LPERC02A	NO	NO	YES	NO	NO	NO	YES
LPERC02B	NO	NO	YES	NO	NO	NO	YES
LPERC03	NO	NO	YES	NO	NO	NO	YES
LPERC03A	NO	NO	YES	NO	NO	NO	YES
LPERC04	NO	NO	YES	NO	YES	NO	YES
LPERC04A	NO	NO	YES	NO	YES	NO	YES
LPERC04B	NO	NO	YES	NO	YES	NO	YES
LPERC05	NO	YES	YES	YES	YES	NO	NO
LPERC06	NO	NO	YES	YES	YES	YES	YES

CPF Check Rules

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
checkCPF_existence	YES	YES	YES	YES	NO	YES	NO
CPF_lowpower01	YES	YES	YES	YES	NO	YES	NO
CPF_lowpower03	YES	YES	YES	YES	NO	YES	NO

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
CPF_lowpower04	YES	YES	YES	YES	NO	YES	NO
CPF_lowpower05	YES	YES	YES	YES	NO	YES	NO
CPF_lowpower07	YES	YES	YES	YES	NO	YES	NO
CPF_lowpower09	YES	YES	YES	YES	NO	YES	NO
CPF_lowpower10	YES	YES	YES	YES	NO	YES	NO
CPFSEM_2	YES	YES	YES	YES	NO	YES	NO
CPFSEM_3	YES	YES	YES	YES	NO	YES	NO
CPFSEM_4	YES	YES	YES	YES	NO	YES	NO
CPFSEM_5	YES	YES	YES	YES	NO	YES	NO
CPFSEM_6	YES	YES	YES	YES	NO	YES	NO
CPFSEM_7	YES	YES	YES	YES	NO	YES	NO
CPFSEM_8	YES	YES	YES	YES	NO	YES	NO
CPFSEM_10	YES	YES	YES	YES	NO	YES	NO
CPFSEM_11	YES	YES	YES	YES	NO	YES	NO
CPFSEM_12	YES	YES	YES	YES	NO	YES	NO
CPFSEM_13	YES	YES	YES	YES	NO	YES	NO
CPFSEM_14	YES	YES	YES	YES	NO	YES	NO
CPFSEM_15	YES	YES	YES	YES	NO	YES	NO
CPFSEM_17	YES	YES	YES	YES	NO	YES	NO
CPFSEM_19	YES	YES	YES	YES	NO	YES	NO
CPFSEM_21	YES	YES	YES	YES	NO	YES	NO
CPFSTX_1	YES	YES	YES	YES	NO	YES	NO
CPFSTX_2	YES	YES	YES	YES	NO	YES	NO
CPFSTX_4	YES	YES	YES	YES	NO	YES	NO
CPFSTX_5	YES	YES	YES	YES	NO	YES	NO
CPFSTX_6	YES	YES	YES	YES	NO	YES	NO
CPFSTX_7	YES	YES	YES	YES	NO	YES	NO
CPFSTX_8	YES	YES	YES	YES	NO	YES	NO
CPFSTX_9	YES	YES	YES	YES	NO	YES	NO

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
CPFSTX_10	YES	YES	YES	YES	NO	YES	NO
CPFSTX_11	YES	YES	YES	YES	NO	YES	NO
CPFSTX_13	YES	YES	YES	YES	NO	YES	NO
CPFSTX_14	YES	YES	YES	YES	NO	YES	NO
CPFSTX_15	YES	YES	YES	YES	NO	YES	NO
CPFSTX_16	YES	YES	YES	YES	NO	YES	NO
CPFSTX_17	YES	YES	YES	YES	NO	YES	NO
CPFSTX_19	YES	YES	YES	YES	NO	YES	NO
CPFSTX_20	YES	YES	YES	YES	NO	YES	NO
CPFSTX_22	YES	YES	YES	YES	NO	YES	NO
CPFSTX_23	YES	YES	YES	YES	NO	YES	NO
CPFSTX_24	YES	YES	YES	YES	NO	YES	NO
CPFSTX_25	YES	YES	YES	YES	NO	YES	NO
CPFSTX_26	YES	YES	YES	YES	NO	YES	NO
CPFSTX_27	YES	YES	YES	YES	NO	YES	NO
CPFSTX_28	YES	YES	YES	YES	NO	YES	NO
CPFSTX_29	YES	YES	YES	YES	NO	YES	NO
CPFSTX_30	YES	YES	YES	YES	NO	YES	NO
CPFSTX_31	YES	YES	YES	YES	NO	YES	NO
CPFSTX_32	YES	YES	YES	YES	NO	YES	NO
CPFSTX_33	YES	YES	YES	YES	NO	YES	NO
CPFSTX_34	YES	YES	YES	YES	NO	YES	NO
CPFSTX_35	YES	YES	YES	YES	NO	YES	NO
CPFSTX_36	YES	YES	YES	YES	NO	YES	NO
CPFSTX_38	YES	YES	YES	YES	NO	YES	NO
CPFSTX_39	YES	YES	YES	YES	NO	YES	NO
CPFSTX_40	YES	YES	YES	YES	NO	YES	NO
CPFSTX_41	YES	YES	YES	YES	NO	YES	NO
CPFSTX_42	YES	YES	YES	YES	NO	YES	NO

UPF Check Rules

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
checkUPF_existence	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower01	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower02	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower03	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower04	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower05	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower06	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower07	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower08	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower09	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower10	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower11	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower12	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower13	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower14	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower15	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower16	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower17	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower18	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower19	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower20	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower21	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower22	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower23	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower24	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower25	NO	YES	YES	YES	YES	NO	NO
UPF_lowpower26	YES	YES	YES	YES	YES	NO	NO

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
UPF_lowpower27	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower28	YES	YES	YES	YES	YES	NO	NO
UPF_lowpower29	YES	YES	YES	YES	YES	NO	NO
UPFSEM_1	YES	YES	YES	YES	YES	NO	NO
UPFSEM_2	YES	YES	YES	YES	YES	NO	NO
UPFSEM_3	YES	YES	YES	YES	YES	NO	NO
UPFSEM_4	YES	YES	YES	YES	YES	NO	NO
UPFSEM_5	YES	YES	YES	YES	YES	NO	NO
UPFSEM_6	YES	YES	YES	YES	YES	NO	NO
UPFSEM_7	YES	YES	YES	YES	YES	NO	NO
UPFSEM_8	YES	YES	YES	YES	YES	NO	NO
UPFSEM_9	YES	YES	YES	YES	YES	NO	NO
UPFSEM_11	YES	YES	YES	YES	YES	NO	NO
UPFSEM_12	YES	YES	YES	YES	YES	NO	NO
UPFSEM_13	YES	YES	YES	YES	YES	NO	NO
UPFSEM_14	YES	YES	YES	YES	YES	NO	NO
UPFSEM_15	YES	YES	YES	YES	YES	NO	NO
UPFSEM_16	YES	YES	YES	YES	YES	NO	NO
UPFSEM_18	YES	YES	YES	YES	YES	NO	NO
UPFSEM_19	YES	YES	YES	YES	YES	NO	NO
UPFSEM_21	YES	YES	YES	YES	YES	NO	NO
UPFSEM_22	YES	YES	YES	YES	YES	NO	NO
UPFSEM_23	YES	YES	YES	YES	YES	NO	NO
UPFSEM_24	YES	YES	YES	YES	YES	NO	NO
UPFSEM_25	YES	YES	YES	YES	YES	NO	NO
UPFSEM_29	YES	YES	YES	YES	YES	NO	NO
UPFSEM_31	YES	YES	YES	YES	YES	NO	NO
UPFSEM_32	YES	YES	YES	YES	YES	NO	NO
UPFSEM_33	YES	YES	YES	YES	YES	NO	NO

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
UPFSEM_34	YES	YES	YES	YES	YES	NO	NO
UPFSEM_35	YES	YES	YES	YES	YES	NO	NO
UPFSEM_36	YES	YES	YES	YES	YES	NO	NO
UPFSEM_37	YES	YES	YES	YES	YES	NO	NO
UPFSEM_38	YES	YES	YES	YES	YES	NO	NO
UPFSEM_39	YES	YES	YES	YES	YES	NO	NO
UPFSEM_40	YES	YES	YES	YES	YES	NO	NO
UPFSEM_41	YES	YES	YES	YES	YES	NO	NO
UPFSEM_42	YES	YES	YES	YES	YES	NO	NO
UPFSEM_43	YES	YES	YES	YES	YES	NO	NO
UPFSEM_44	YES	YES	YES	YES	YES	NO	NO
UPFSEM_45	YES	YES	YES	YES	YES	NO	NO
UPFSEM_46	YES	YES	YES	YES	YES	NO	NO
UPFSEM_47	YES	YES	YES	YES	YES	NO	NO
UPFSEM_48	YES	YES	YES	YES	YES	NO	NO
UPFSEM_49	YES	YES	YES	YES	YES	NO	NO
UPFSEM_50	YES	YES	YES	YES	YES	NO	NO
UPFSEM_50	YES	YES	YES	YES	YES	NO	NO
UPFSEM_52	YES	YES	YES	YES	YES	NO	NO
UPFSTX_1	YES	YES	YES	YES	YES	NO	NO
UPFSTX_2	YES	YES	YES	YES	YES	NO	NO
UPFSTX_4	YES	YES	YES	YES	YES	NO	NO
UPFSTX_5	YES	YES	YES	YES	YES	NO	NO
UPFSTX_6	YES	YES	YES	YES	YES	NO	NO
UPFSTX_13	YES	YES	YES	YES	YES	NO	NO
UPFSTX_15	YES	YES	YES	YES	YES	NO	NO
UPFSTX_16	YES	YES	YES	YES	YES	NO	NO
UPFSTX_17	YES	YES	YES	YES	YES	NO	NO
UPFSTX_18	YES	YES	YES	YES	YES	NO	NO

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
UPFSTX_19	YES	YES	YES	YES	YES	NO	NO
UPFSTX_20	YES	YES	YES	YES	YES	NO	NO
UPFSTX_21	YES	YES	YES	YES	YES	NO	NO
UPFSTX_22	YES	YES	YES	YES	YES	NO	NO
UPFSTX_23	YES	YES	YES	YES	YES	NO	NO
UPFSTX_24	YES	YES	YES	YES	YES	NO	NO
UPFSTX_25	YES	YES	YES	YES	YES	NO	NO
UPFSTX_26	YES	YES	YES	YES	YES	NO	NO
UPFSTX_27	YES	YES	YES	YES	YES	NO	NO
UPFSTX_28	YES	YES	YES	YES	YES	NO	NO
UPFSTX_29	YES	YES	YES	YES	YES	NO	NO
UPFSTX_31	YES	YES	YES	YES	YES	NO	NO
UPFSTX_33	YES	YES	YES	YES	YES	NO	NO
UPFSTX_34	YES	YES	YES	YES	YES	NO	NO
UPFSTX_39	YES	YES	YES	YES	YES	NO	NO
UPFWRN_1	YES	YES	YES	YES	YES	NO	NO
UPFWRN_2	YES	YES	YES	YES	YES	NO	NO
UPFWRN_3	YES	YES	YES	YES	YES	NO	NO
UPFWRN_4	YES	YES	YES	YES	YES	NO	NO
UPFWRN_5	YES	YES	YES	YES	YES	NO	NO
UPFWRN_6	YES	YES	YES	YES	YES	NO	NO
UPFWRN_7	YES	YES	YES	YES	YES	NO	NO
UPFWRN_13	YES	YES	YES	YES	YES	NO	NO
UPFWRN_14	YES	YES	YES	YES	YES	NO	NO
UPFWRN_16	YES	YES	YES	YES	YES	NO	NO
UPFWRN_17	YES	YES	YES	YES	YES	NO	NO
UPFWRN_19	YES	YES	YES	YES	YES	NO	NO
UPFINFO_3	YES	YES	YES	YES	YES	NO	NO
UPFINFO_7	YES	YES	YES	YES	YES	NO	NO

Debug Rules

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
LP_SDC_PARSE_DEBUG	YES	YES	YES	YES	YES	NO	NO

SoC Abstraction Rules

Rule	RTL	NETLIST	PG NETLIST	DEF /LEF	UPF	CPF	SGDC
PV_Abstract01	YES	YES	YES	YES	YES	NO	NO
PV_AbstractReadInfo	YES	YES	YES	YES	YES	NO	NO

Appendix: SGDC Constraints

SpyGlass Design Constraints (SGDC) provides additional design information that is not apparent in an RTL.

In addition, you can restrict SpyGlass analysis to certain objects in a design by specifying these objects by using SGDC commands.

The following table lists the SGDC commands used by the SpyGlass Power Verify solution:

always_on_cell	*always_on_pin*	*always_on_buffer*
clock	*assume_path*	*cell_hookup*
isolation_cell	*ignore_crossing*	*input_isocell*
multivit_lib	*levelshifter*	*assertion_signal*
domain_signal	*non_pd_inputcells*	*power_down_sequence*
pin_voltage	*pg_cell*	*pg_pins_naming*
domain_outputs	*power_down*	*domain_inputs*
ram_instance	*power_switch*	*power_state*
retention_cell	*ram_switch*	*retention_instance*
cell_tie_class	*set_case_analysis*	*cell_pin_info*
switchoff_wrapper_instance	*special_cell*	*supply*

ignore_supply_pin	*voltage_domain*	*power_data*
antenna_cell	*isolation_wrapper*	*aon_buffered_signals*
reset	*lp_ignore_cells_for_erc*	*associate_lib*
set_supply_node	*set_lib_name*	*disallow_upf_command*
make_mandatory_upf_commands_options	*reference_toplevel_isolation_signal*	

List of Topics