SYNOPSYS®
Silicon to Software™

# VC SpyGlass Constraints Overview

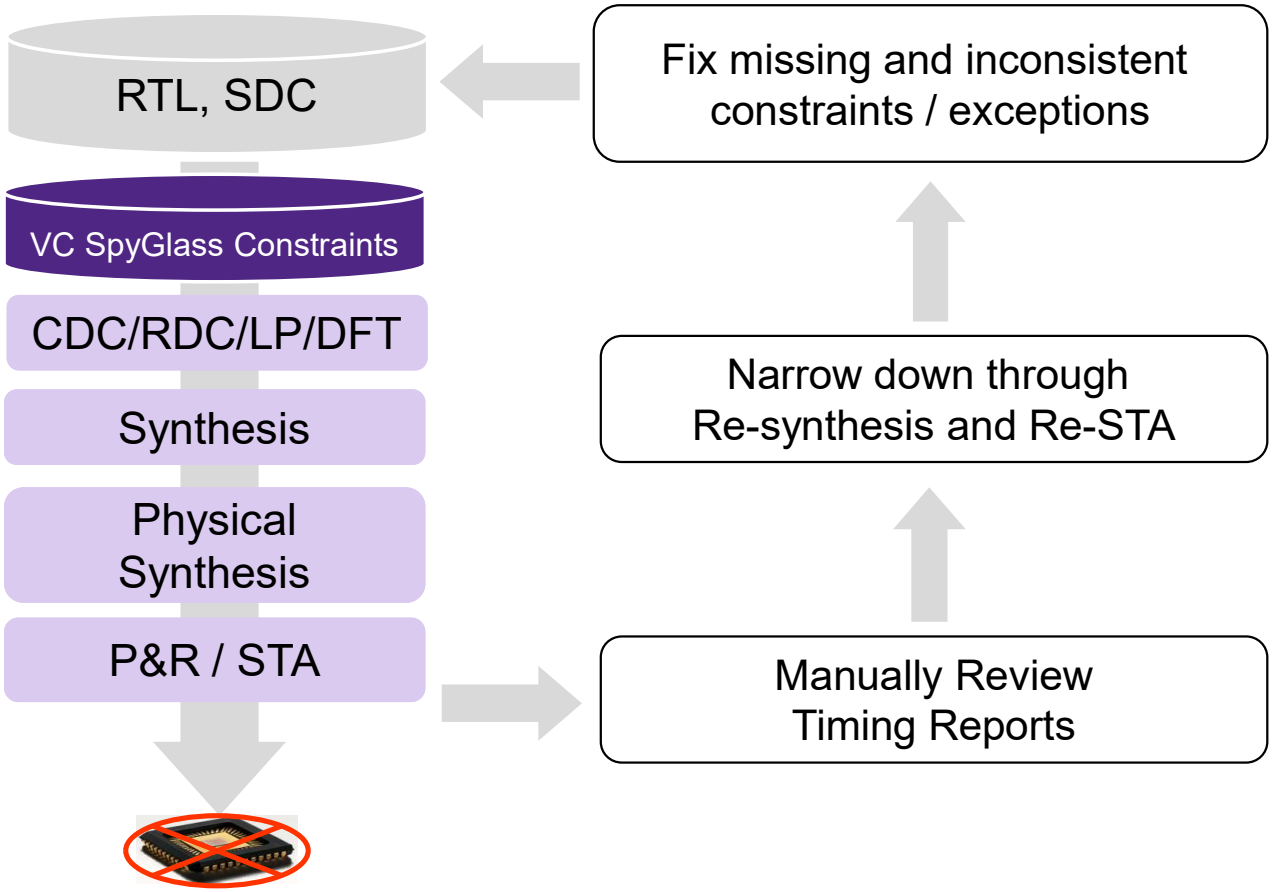
July 2022

# Agenda

- Why Signoff Constraints at RTL?
- VC SpyGlass Platform
- VC SpyGlass Constraints Overview
  - Dynamic Verification of Multi-cycle path and False path
  - DC/PT Compatibility and TCL Support
  - Differential Report Analysis
  - Debug Capabilities

# Why Signoff Constraints at RTL?

# Why Signoff Constraints at RTL?




## Constraints Signoff Upfront Reduces Backend Iterations

# VC SpyGlass Platform

# VC SpyGlass

Industry leading static verification platform for RTL signoff

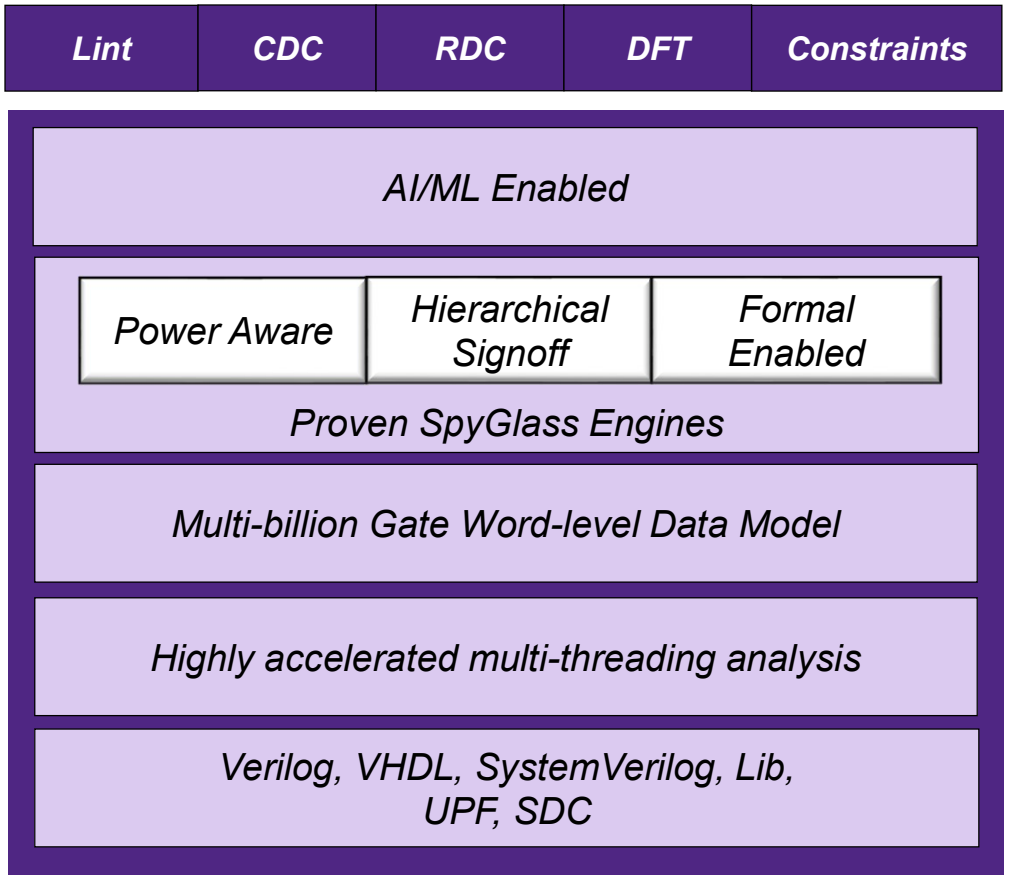
**10x noise reduction with AI vs. competition**

---

**3x performance and ½ the memory vs. competition**

---

**Synopsys Design Compiler, PrimeTime, VCS, Verdi compatibility**

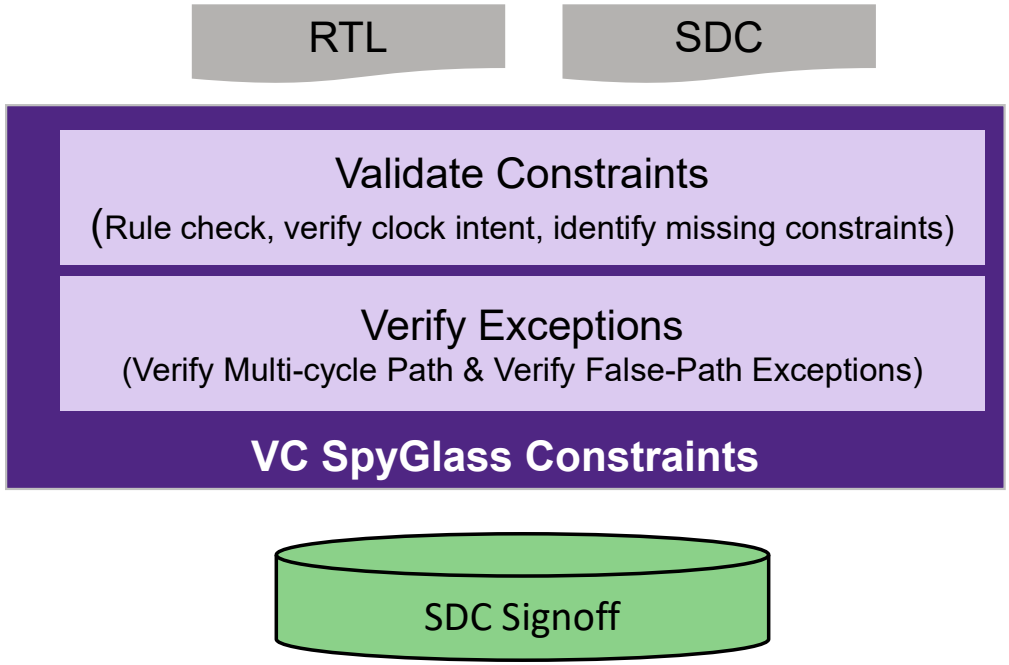


# VC SpyGlass Constraints Overview

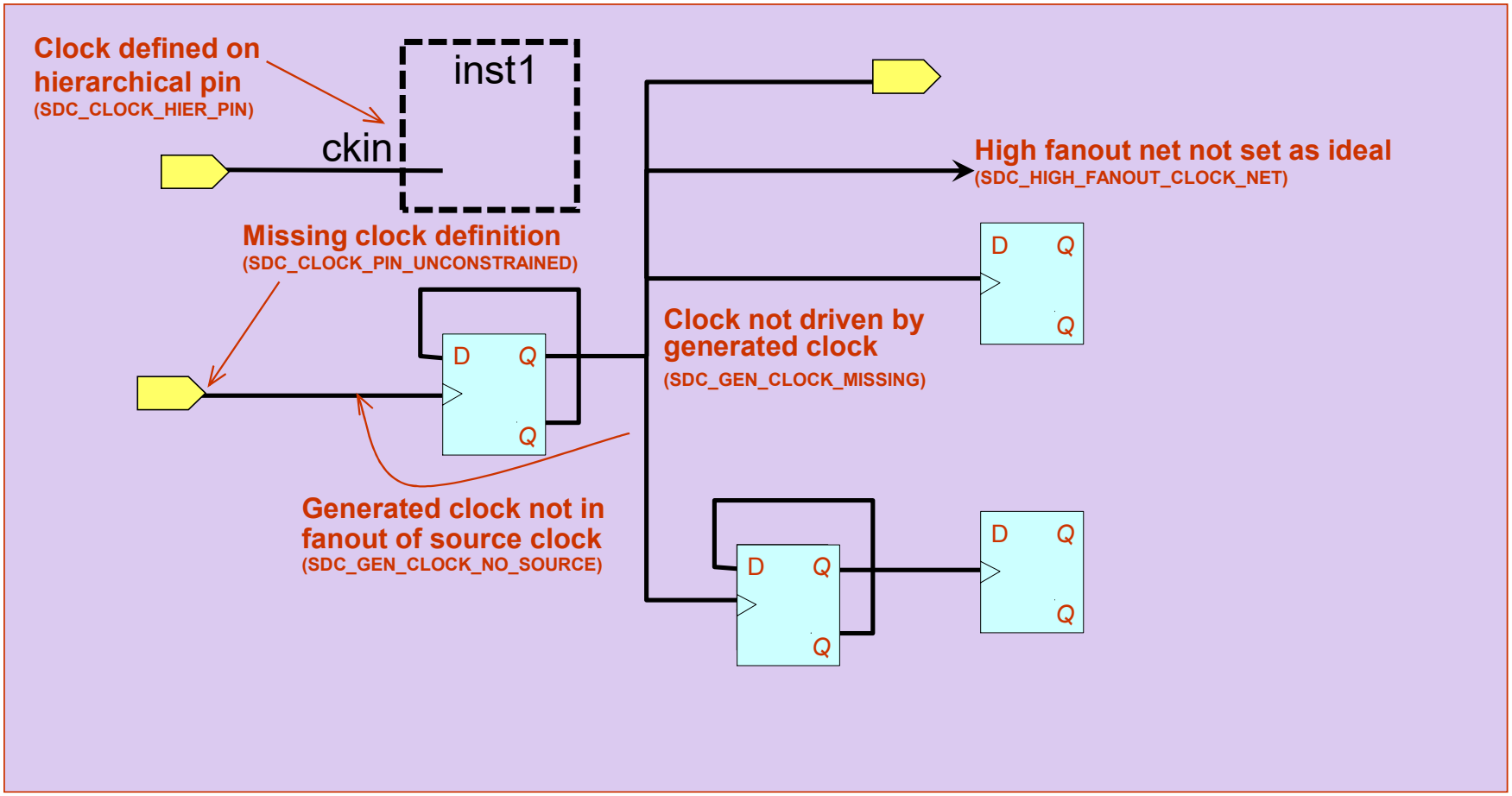
# VC SpyGlass Constraints Overview

## Industry-standard signoff platform accelerating design cycle

- Delivers 3X performance with half the memory and billion+ gate design capacity
- Signoff confidence using exhaustive constraint checks
- Design Compiler and PrimeTime compatibility
- Dynamic and formal verification for multi-cycle and false path
- Debug visibility across abstraction levels



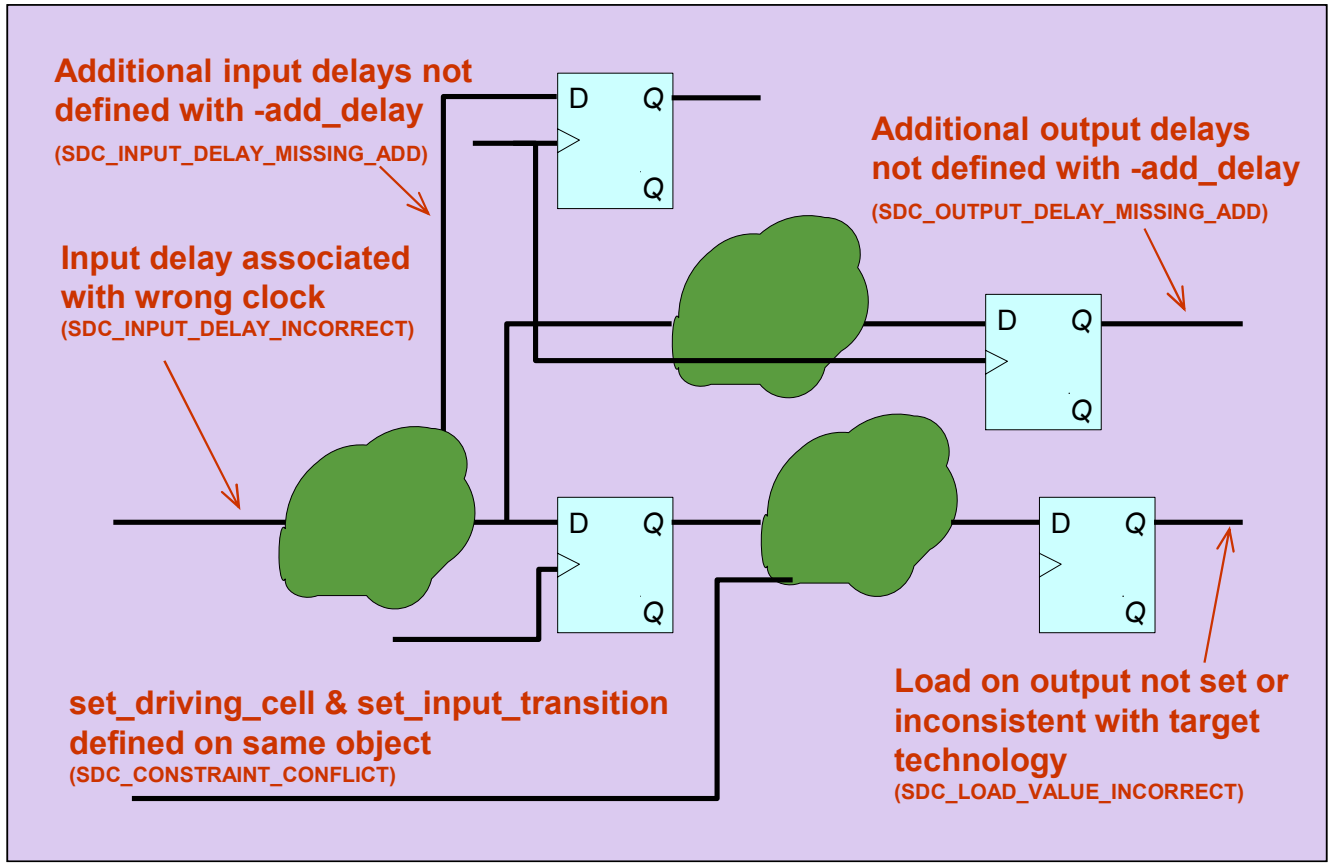
# Example Of Typical Clock Issues



# VC SpyGlass Constraints Tags – Clocks Checks (1)

| Category | VC SpyGlass Tag | Description |
|---|---|---|
| Clock Intent Checks | SDC_VIRTUAL_CLOCK_UNUSED, SDC_CLOCK_UNUSED | Constrained clock not used as a clock |
| | SDC_GEN_CLOCK_NO_SOURCE | A generated clock is not in the fanout of it source clock |
| | SDC_IO_DELAY_CLOCK_PORT | set_input_delay/set_output_delay has been specified on a clock port |
| | SDC_SYNC_CLK_DIFFERENT_ROOT | Some clocks in the same clock domain have a different root clock |
| | SDC_GEN_CLOCK_PORT | Object on which clock is generated should not be a port |
| | SDC_CLOCK_OVERLAP | Clock source pin is in the fanout of another clock, but is not generated by that clock |
| | SDC_CLOCK_PIN_UNCONSTRAINED | Clock not driven by a clock constraint (missing clock constraints) |
| | SDC_UNCONSTRAINED_SEQ<br>SDC_CLOCK_PIN_CONSTANT<br>SDC_GEN_CLOCK_SOURCE_CONSTANT | Clock driven by a constant value or hanging (clocks is blocked by constant/sdt/sense etc. or clock pin of seq is hanging) |
| | SDC_GEN_CLOCK_NO_MASTER | Identifies multiple clocks at the source pin of a generated clock that does not have a master_clock argument |
| | SDC_GEN_CLOCK_MISSING | Identifies clocks for which generated clock is not defined at a design object |
| | SDC_GEN_CLOCK_CONVERGE | Ensures a single path for the master clock to its generated clock |
| | SDC_CLOCK_MULTIPLE_SOURCES | Multiple paths exist from the clock pin of a sequential cell to different clock sources |
| | Assertion Generation Based Check | Incorrectly defined generated clock using dynamic |
| | SDC_CLOCK_CONVERGE | Identifies instances when the same clock is converging through different combinational paths |
| | SDC_CLOCK_ON_MUX_SELECT | Identifies the control pin of a mux in the clock path |

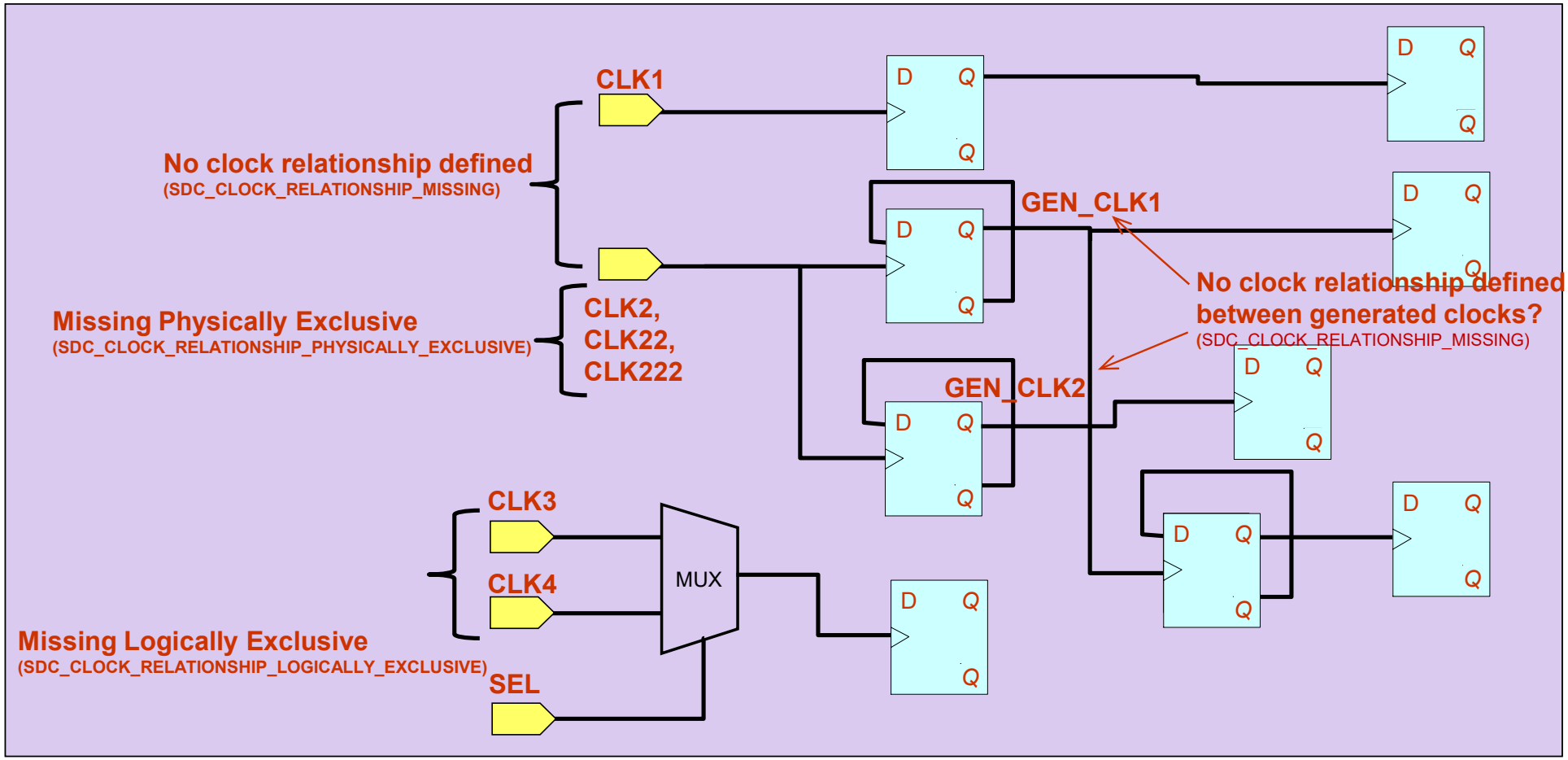
# Example Of Typical I/O Issues



# VC SpyGlass Constraints Tags – I/O & other Categories (2)

| Categorization | VC SpyGlass Tag | Description |
|---|---|---|
| I/O Delay Checks | SDC_INPUT_DELAY_MISSING | Input not constrained by set_input_delay |
| | SDC_OUTPUT_DELAY_MISSING | Output has no set_output_delay constraint |
| | SDC_INPUT_DELAY_INCORRECT | Input constraint associated with wrong (or, incomplete set of) clocks |
| | SDC_INPUT_DELAY_MISSING_ADD | set_input_delay is set on the same input relative to multiple clocks but -add_delay missing |
| | SDC_OUTPUT_DELAY_INCORRECT | Output constraint associated with wrong (or, incomplete set of) clocks |
| | SDC_OUTPUT_DELAY_MISSING_ADD | set_output_delay is set on the same output relative to multiple clocks but -add_delay is missing |
| Clock Uncertainty, Transition & Latency Checks | SDC_CLOCK_UNCERTAINTY_MISSING | Inter clock uncertainty not defined between synchronous clocks |
| | SDC_CLOCK_UNCERTAINTY_NEGATIVE | set_clock_uncertainty is set to negative value |
| | SDC_CLOCK_UNCERTAINTY_INVALID_OBJECT | set_clock_uncertainty constraint set on an object which is not a real or generated clock |
| | SDC_CLOCK_LATENCY_NEGATIVE | set_clock_latency is set to negative value |
| | SDC_INPUT_TRANSITION_MISSING | Input transition or drive or driving cell is not defined for input |
| High Fanout, Load | SDC_LOAD_VALUE_INCORRECT | Load values are outside technology limits |
| | SDC_HIGH_FANOUT_CLOCK_NET, SDC_HIGH_FANOUT_NET | High fanout net (clock/non-clock) identified |
| Overlapping/Conflicting Constraints | SDC_CONSTRAINT_CONFLICT | Reports conflicts between set_driving_cell and set_input_transition defined on same object. |
| | SDC_CONSTRAINT_OVERWRITTEN | Overwritten constraint detected |

# Example Of Typical Clock Relationship Issues



# Clock Relationship Verification – SCG Checks (3)

## Clock relationship checks to verify correctness of set_clock_group specification

| Category | VC SpyGlass Tag | Description |
|---|---|---|
| Clock Relationship Verification | SDC_CLOCK_RELATIONSHIP_INCORRECT | Generated relationships between clocks and set_clock_groups has been defined between them<br>Ex:- Parent generated relationship exists between Clocks <clk1> and <clk2> and set_clock_groups is defined between them<br>Ex:- Generated Clocks <clk1> and <clk2> are from same master and set_clock_groups is defined between them |
| | SDC_CLOCK_RELATIONSHIP_MISSING | Generated clocks that do not have set_clock_groups defined between them |
| | SDC_CLOCK_RELATIONSHIP_CONFLICT | Conflicting set_clock_uncertainty constraints for clocks defined as asynchronous/physically exclusive |
| | SDC_CLOCK_RELATIONSHIP_INCORRECT | Clocks, which have a set_clock_groups defined but share a harmonic relationship<br>Ex:- set_clock_groups is specified between clocks <clk1> and <clk2> sharing a harmonic relationship |
| | SDC_CLOCK_RELATIONSHIP_PHYSICALLY_EXCLUSIVE | Multiple clock definitions at the same node as being physically exclusive |
| | SDC_CLOCK_RELATIONSHIP_MISSING | Missing asynchronous (non-harmonic) set_clock_group between clocks<br><br>set_clock_groups/set_false_path is not specified between async non-harmonic clocks <clk1> and <clk2> |
| | SDC_CLOCK_RELATIONSHIP_ALLOW_PATH | Asynchronous clocks have defined set_clock_groups with -allow_path |
| | SDC_CLOCK_RELATIONSHIP_LOGICALLY_EXCLUSIVE | Missing logically-exclusive set_clock_group between clocks |

# VC SpyGlass Constraints: Performance Results

| Design | Run-time | | | Memory | | |
|---|---|---|---|---|---|---|
| | VC SpyGlass | SpyGlass | Gain (X) | VC SpyGlass (GB) | SpyGlass (GB) | Gain (X) |
| SoC 1 (4B) | 14 hours (30 rules) | SpyGlass doesn't complete | - | 357 | - | - |
| Block 1 | 245 sec | 679 sec | **2.8** | 1.7 | 6 | **3.3** |
| Block 2 | 483 sec | 1982 sec | **4.1** | 4.4 | 8 | **1.8** |
| Block 3 | 271 sec | 2018 sec | **7.4** | 2.1 | 7.8 | **3.7** |
| Block 4 | 272 sec | 821 sec | **3** | 1.7 | 5.5 | **3.2** |
| Block 5 | 262 sec | 437 sec | **1.7** | 1.8 | 5.4 | **3** |

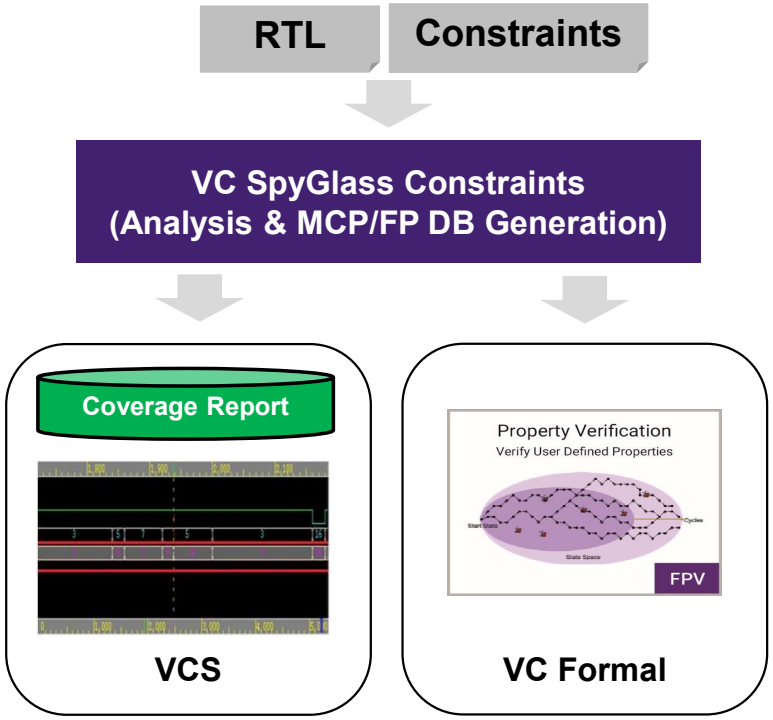
4 cores: Multi-core rule engines for both design read and constraints analysis

# Dynamic Verification for Multi-cycle and False Path

# Timing Exception Verification Flow

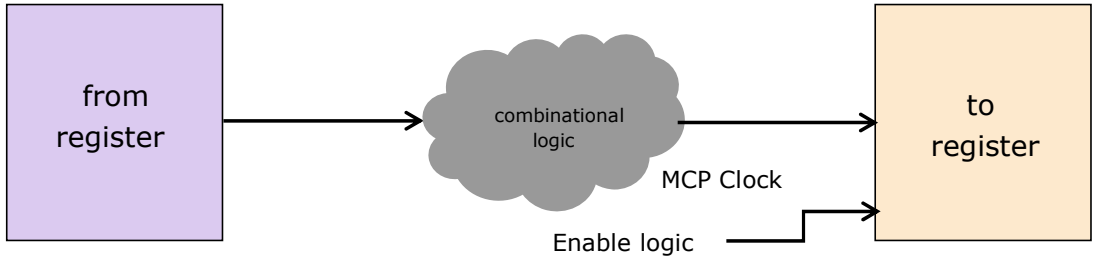
## Static Verification Links to Formal and Dynamic Simulation

- VC SpyGlass Constraints to generate MCP/FP DB
  - Structural checker bedrock for Formal and Dynamic analysis
  - Filtering based on specific cells
- VC Formal (block-level)
  - Bug hunting to identify MCP and FP violations
  - Utilizes MCP/FP database (2-step flow)
- VCS (block & SoC level)
  - Enable inconclusive assertions in simulation
  - Failure using waveform viewer



# Assertion Generation and Application

- The following diagram shows how SVA is generated

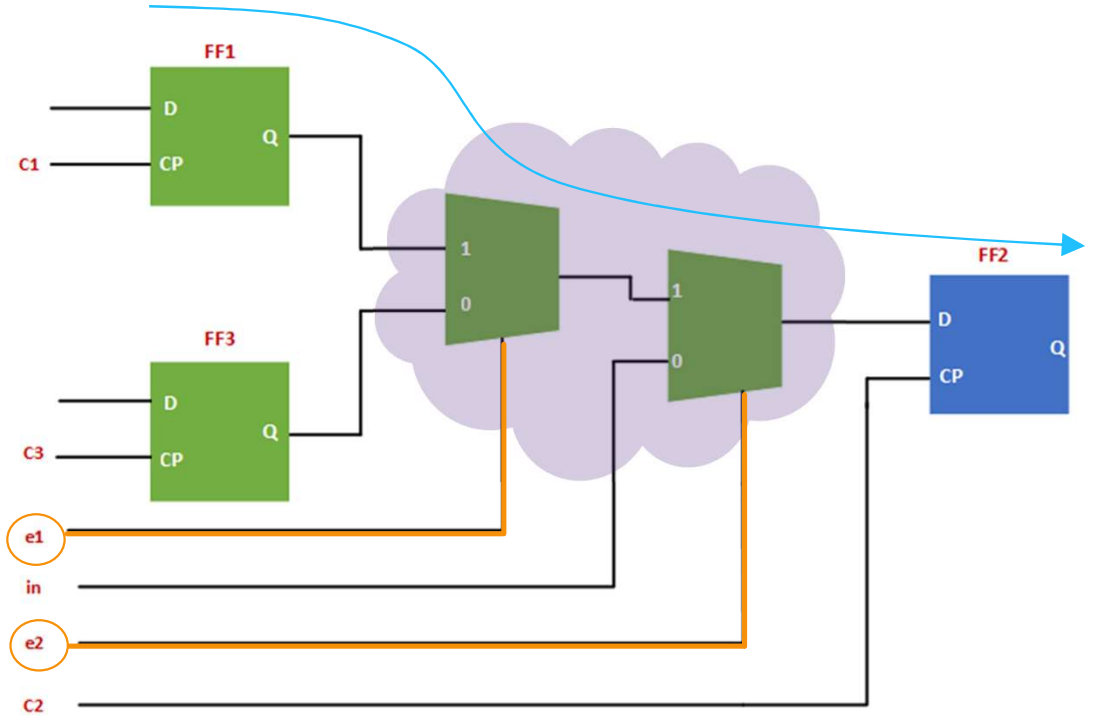



- Benefits
  - Exhaustively identifying all possible paths from an MCP constraint
  - An enable logic which determines change of data in destination register
    - Enable logic formed by evaluating all control signals which affect data path
  - Reducing required number of assertions/MCP by identifying common enable logic governing many flat paths

# “Enable Condition” Extraction

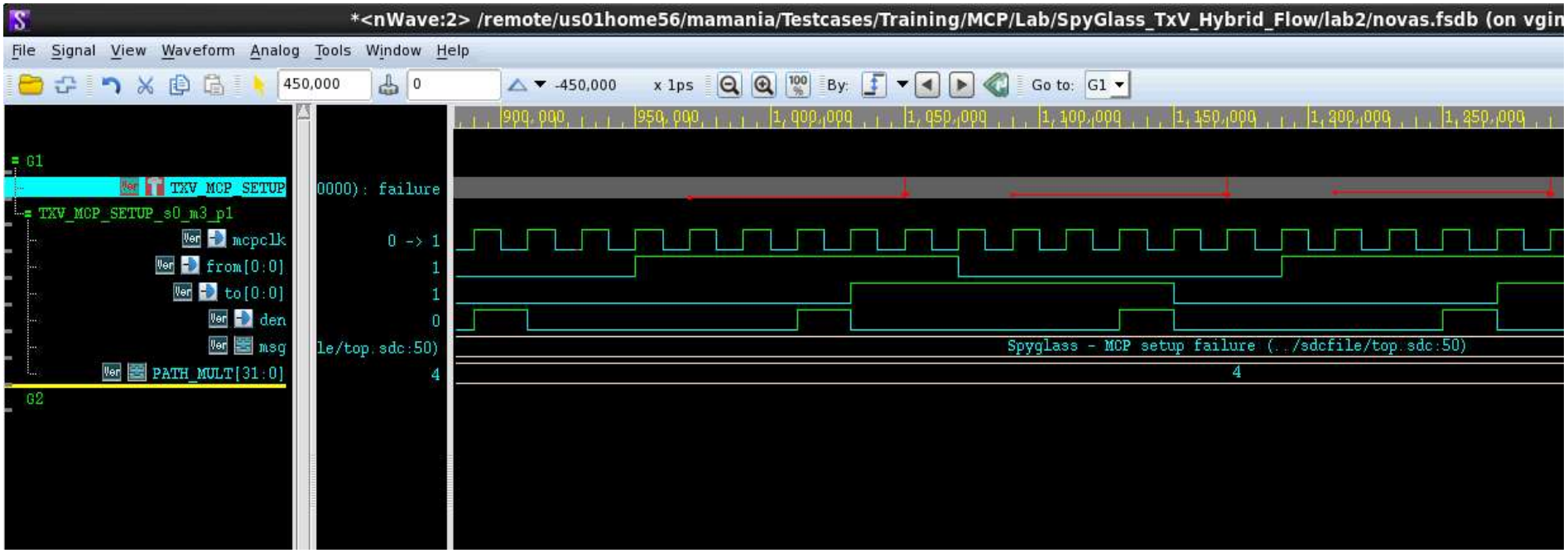
## Statically Identify enabling conditions for MCP paths

- Analyze all multi-cycle paths
  - Consider all combinational enables:
    - MUX, AND, NAND, OR, NOR
- Compute enable condition
- Scalable Solution
  - Single Assertion for multiple destinations of a source with same enable condition



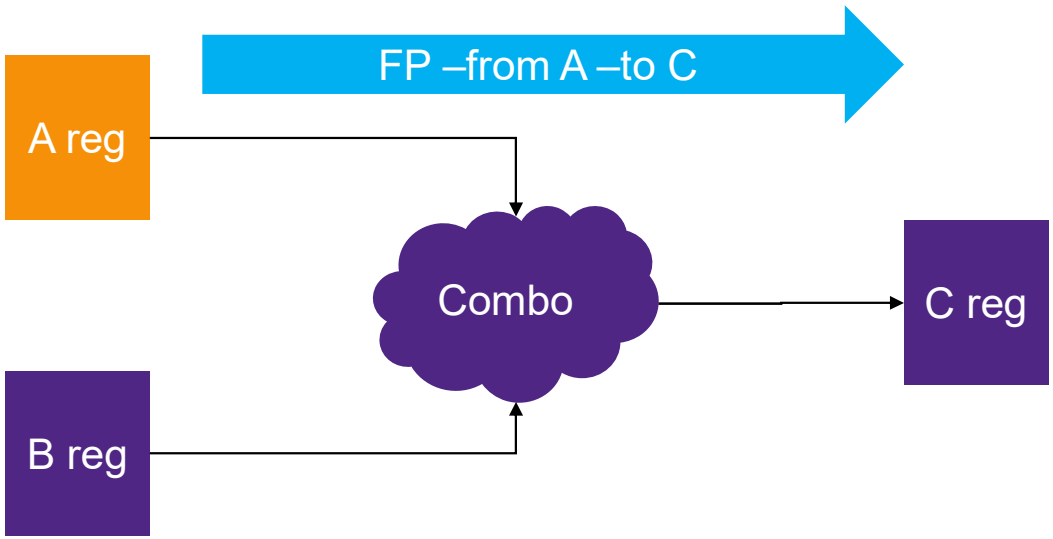

MCP path: C1 → C2
data path: FF1 → FF2
Enable condition: e1 & e2

# MCP: Failure debug in Verdi




- Assertion trigger shows failure
- Data in signal “from” is launched on a clock edge and is captured in “to” signal after 4 cycles violating MCP of cycle count 5

# False Path Assertion Verification




**Supported False Paths**

- Clock to clock
- Flop to flop
- Based on Through
- Both rise/fall

# DC/PT Compatibility and TCL Support

# VC SpyGlass Accelerates Productivity

## Reuse DC/PT Setup

```
vc_static_shell> set search_path "<space separated list of
directories>"
vc_static_shell> set link_library "<space separated list
of libs>"

vc_static_shell> analyze  -format vhdl -work lib_name
file.vhd
vc_static_shell> analyze  -format verilog -work lib_name
file.v
…
vc_static_shell> elaborate -work lib_name DUT
vc_static_shell> read_sdc top.sdc


vc_static_shell> check_sdc
vc_static_shell> report_sdc

vc_static_shell> start_gui
```

Setup

RTL Design

Check

Report GUI

Reuses Design Compiler and PrimeTime Setup with SDC and non-SDC commands

## Custom TCL Checks

```
#Unconstrained Clock Pins
sizeof_collection [filter_collection $clockpins "sizeof(clocks)==0"]

#Seqs receiving one clock
sizeof_collection [filter_collection $clockpins "sizeof(clocks)==1"]

#Multi-Clock Seqs
sizeof_collection [filter_collection $clockpins "sizeof(clocks) > 1"]

#Constant data flops
set datapins [get_pins -of_objects [all_registers] -filter "is_data_pin
&& (case_value==0 || case_value==1)"]

foreach_in_collection p $datapins {
       redirect -file "constant_cells.rpt" {echo "[get_object_name
[get_cells -of_objects $p]]"}
}
```

TCL procedure for Custom checks on RTL

# Compatible with Design Compiler and PrimeTime

| SDC Commands | | | Non-SDC Commands |
|---|---|---|---|
| all_clocks | set_min_capacitance | set_max_area | all_connected |
| all_inputs | set_min_delay | set_max_capacitance | all_fanin |
| all_outputs | set_multicycle_path | set_max_delay | all_fanout |
| all_registers | set_wire_load_selection_group | set_case_analysis | all_instances |
| create_clock | set_max_fanout | set_clock_gating_check | get_attribute |
| create_generated_clock | set_max_time_borrow | set_clock_groups | get_clock_network_objects |
| create_voltage_area | set_max_transition | set_clock_latency | get_clock_relationship |
| current_design | set_timing_derate | set_clock_sense | get_clocks |
| delay_port_pin_list | set_voltage | set_clock_transition | add_to_collection |
| set_min_pulse_width | set_wire_load_min_block_size | set_clock_uncertainty | remove_from_collection |
| get_cells | set_false_path | set_data_check | append_to_collection |
| get_lib_cells | set_fanout_load | set_disable_timing | define_user_attribute |
| get_lib_pins | set_ideal_latency | set_drive | get_timing_arcs |
| get_libs | set_ideal_network | set_driving_cell | get_timing_paths |
| get_nets | set_ideal_transition | set_operating_conditions | set_sense |
| get_pins | set_input_delay | set_output_delay | set_cell_mode |
| get_ports | set_input_transition | set_port_fanout_number | |
| group_path | set_level_shifter_strategy | set_propagated_clock | |
| set_level_shifter_threshold | set_load | set_resistance | |
| max_case_voltage | set_logic_one | set_wire_load_mode | |
| set_wire_load_model | set_logic_zero | | |

# Analysis using get_trace_paths

Usage: **get_trace_paths** # Creates a collection of tracing paths for custom reporting and other processings.

| | |
|---|---|
| **-from <signal>** | (Traces specified signal) |
| **[-mode <mode>]** | (Mode with options 'user' for SCA & 'all' for everything ) |
| **[-to <signal>]** | (Use this option with -from to specify end point of tracing) |
| **[-through <signal>]** | (Use with both -from and -to to specify through which signal to trace) |
| **[-back]** | (Performs a backtrace. This option cannot be used with -to /-through) |
| **[-level <signal_level>]** | (Maximum number of level for signals in traverse: Value >= 0) |
| **[-max_path <n>]** | (Maximum number of paths to be reported: Value >= 1) |
| **[-quiet]** | (Suppresses all messages. Syntax error messages are not suppressed) |
| **[-psw_mode]** | (Traverse in switch enable mode) |
| **[-through_sequential]** | (Traverse through sequentials) |
| **[-through_icg]** | (Traverse through ICG cells) |
| **[-exhaustive_mode]** | (Get all paths) |

Like PT Command analyze_paths, Use Path Tracing Capability for Debug

# Path Report Using report_trace_paths

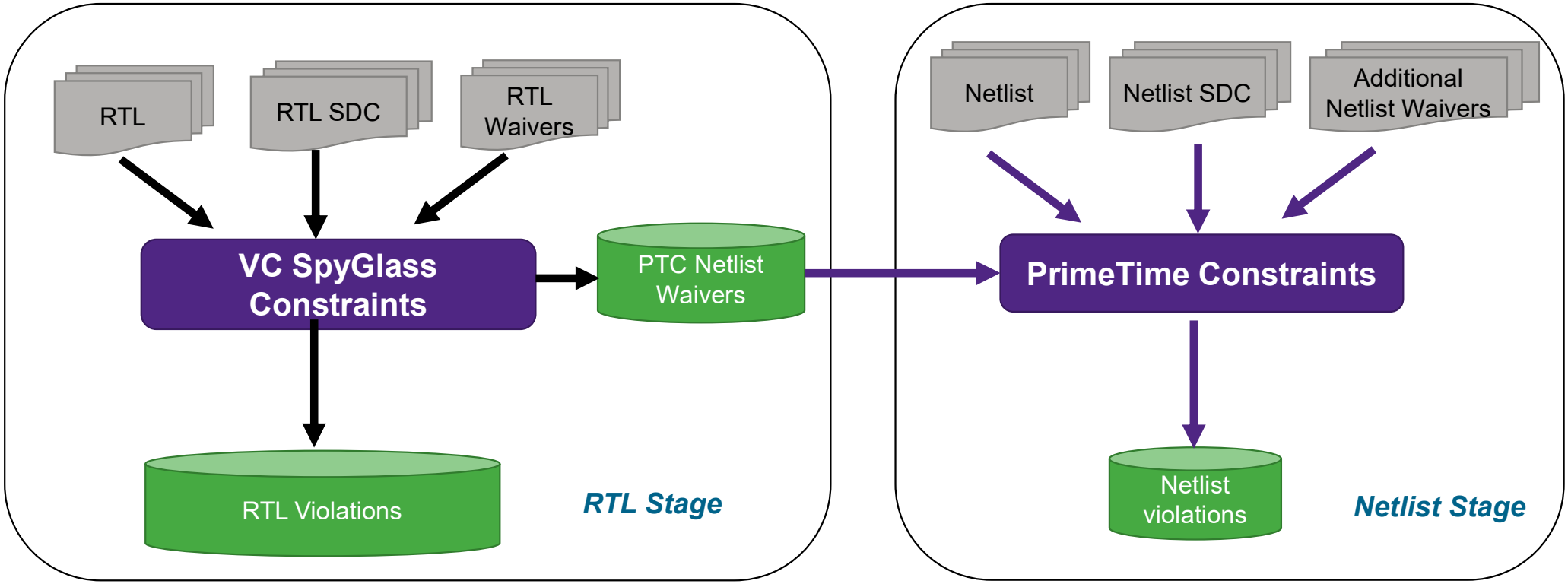
```
Point                                                                                Type
-------------------------------------------------------------------------------     ------------
U_DWC_mshc_clk_gate/I_AND_clk_gated/OUT
      BITWISE_AND
U_DWC_mshc_clk_gate/clk_gated                                                        output pin
U_DWC_mshc_uhs2/bclk                                                                 input pin
U_DWC_mshc_uhs2/U_DWC_mshc_uhs2_tlu/bclk                                             input pin
U_DWC_mshc_uhs2/U_DWC_mshc_uhs2_tlu/U_DWC_mhsc_uhs2_tlu_tpsm/bclk                    input pin
U_DWC_mshc_uhs2/U_DWC_mshc_uhs2_tlu/U_DWC_mhsc_uhs2_tlu_tpsm/dcmd_ebsy_wait/CLK      SEQ_FF
---------------------------------------------------------------------
```

**vc_static_shell**> report_trace_paths [get_trace_paths -from U_DWC_mshc_clk_gate/enable_scan]

[Info] REPORT_TRACE_PATHS_HEADER: path info:

# Waivers Translation to PTC

# VC SpyGlass & PrimeTime Constraints Flow




Message based RTL waivers translation to PT based  Netlist waiver commands

# VC SpyGlass to PrimeTime Constraints Waiver File Format

**VC SpyGlass RTL Waiver Examples**

*waive_sdc –add waiver1 -tag “SDC_CLOCK_UNUSED" -filter { ClockName =~ CLK* }*

*waive_sdc –add waiver2 –tag “SDC_GEN_CLOCK_NO_MASTER” -filter {ClockName==gClk1 }*

**Translated PrimeTime Constraints Waiver Examples**

create_waiver -rule CLK_0026 -condition [list "clock"\
[get_clocks CLK*]]

create_waiver -rule CLK_0028 -condition [list "clock"\
[get_clocks gClk1]]

- Waivers translated for cases involving only registers and ports
  - Different naming between RTL and Netlist for other logic
  - Regular expression supported in waivers
- Unmapped waivers generated in comment form in translated PTC waiver file

# Differential Report Analysis

# Differential Report Analysis for Incremental Updates

## Delivers faster comparisons between two RTL or constraints

- Easy comparison of the results of two runs
  - 2 SDCs on the same RTL
  - 2 incremental RTL with the same SDC

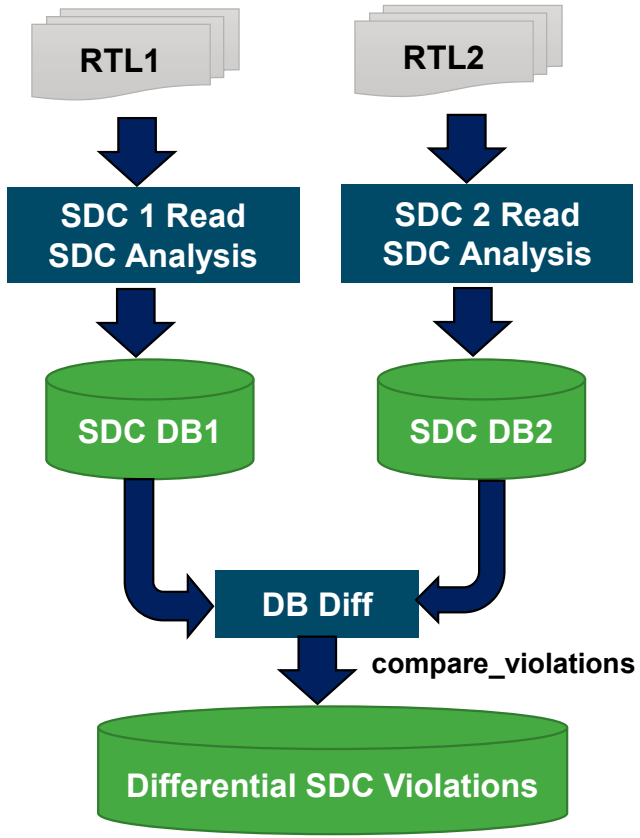
**Run1:**

```
Severity  Stage   Tag                          Count
--------  -----  ---------------------------  -----
error   SYNC VERIF_CLK_SYNC_CLK_DIFFERENT_ROOT      26
--------  -----  ---------------------------  -----
```

**Run2:**

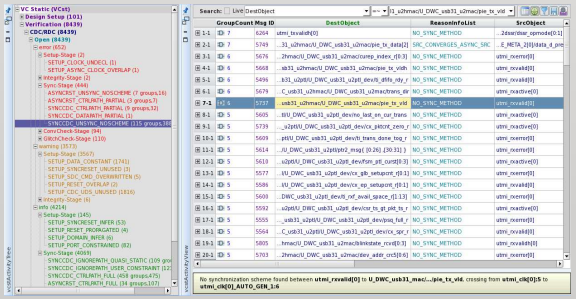
```
Severity  Stage   Tag                          Count
--------  -----  ---------------------------  -----
error VERIF_CLK_SYNC_CLK_DIFFERENT_ROOT       24
--------  -----  ---------------------------  -----
```

2 extra violations which are shown in detailed report for user review

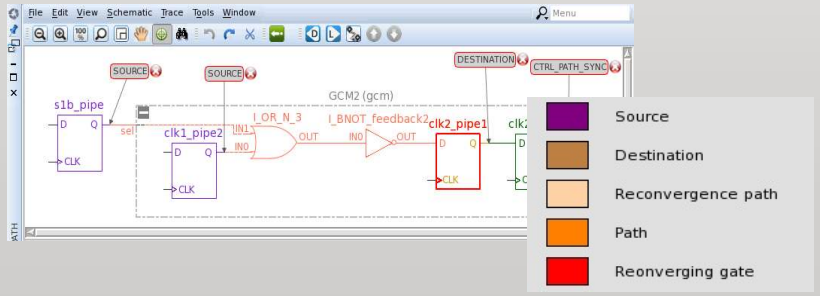


# Debug Capabilities

# Intuitive Verdi Debug to Accelerate Root Cause Analysis

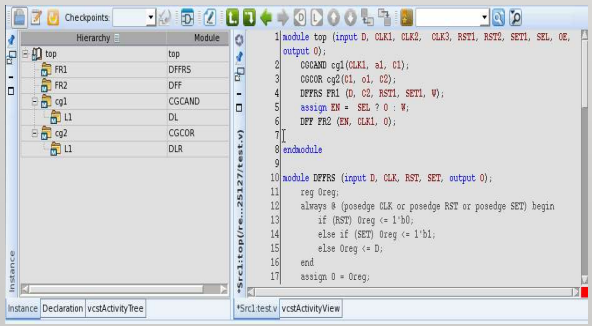


*Easy violation group & filter*
*Priority-ordered violation tree*

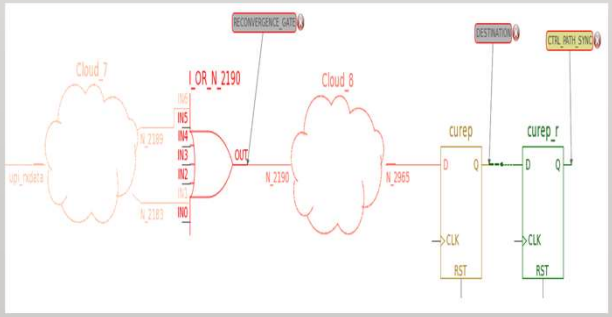



*Locators pointing to relevant logic*
*Domain-specific coloring*
*Debug properties probing*

*Cross-probe signals between views*




*Source code exploration*

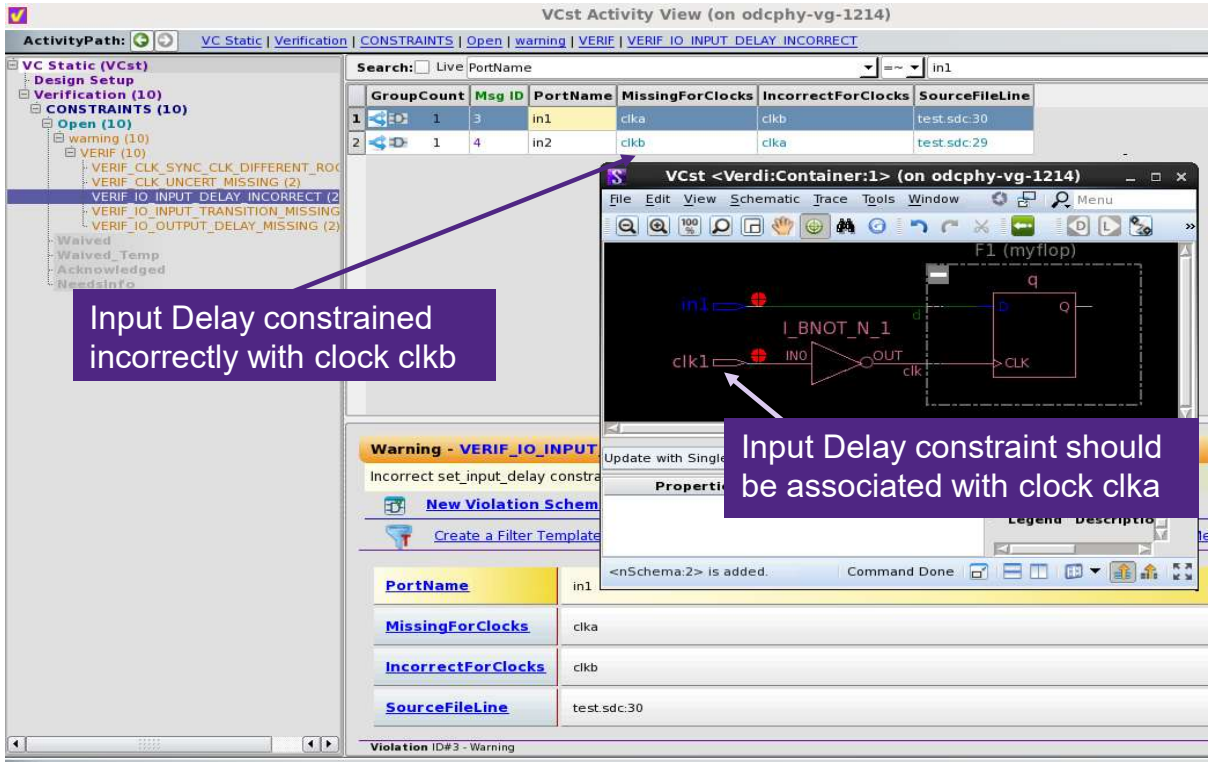



*Abstract irrelevant logic in cloud*

# VC SpyGlass Constraints Offers Intuitive Insights For Every Check

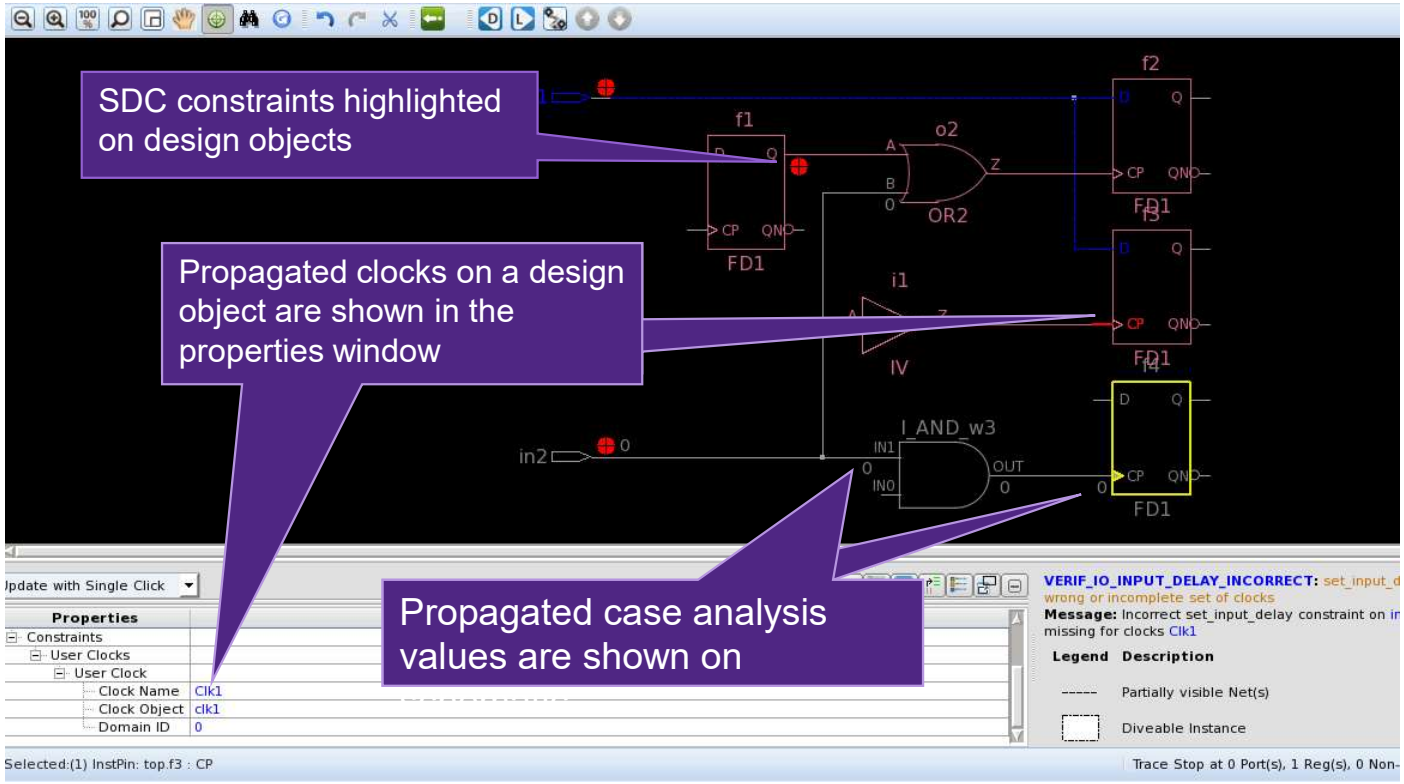
## Input and Output Delay Check – VERIF_IO_INPUT_DELAY_INCORRECT

- Report when Input delay is specified, but associated with wrong clock
- If missed, may lead to incorrect slack assumptions during synthesis
  - Timing closure may fail



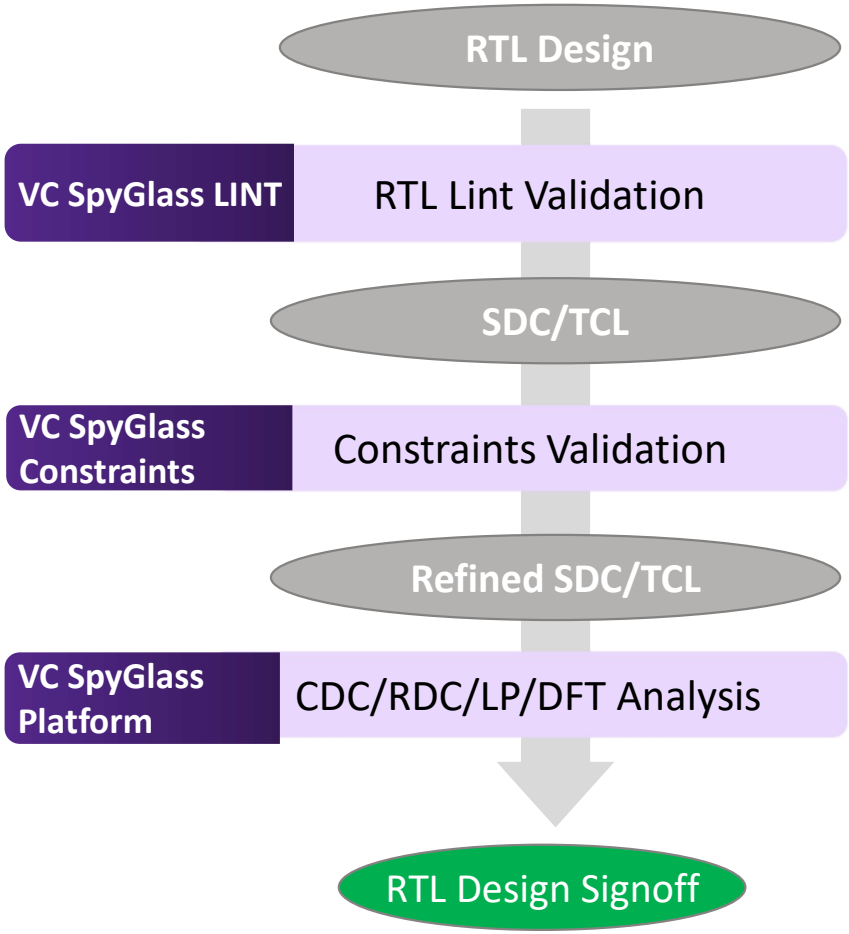
# VC SpyGlass Constraints Provides Intuitive Annotation in Verdi

## Constant and Clock Propagation Annotation in Schematic




- Propagated constant values shown on ports/pins in schematic
- Propagated clock information available in properties window for selected pin/nets in schematic
- SDC constraints highlighted with red circled + sign on design object

# Drive RTL Signoff With Quality Constraints

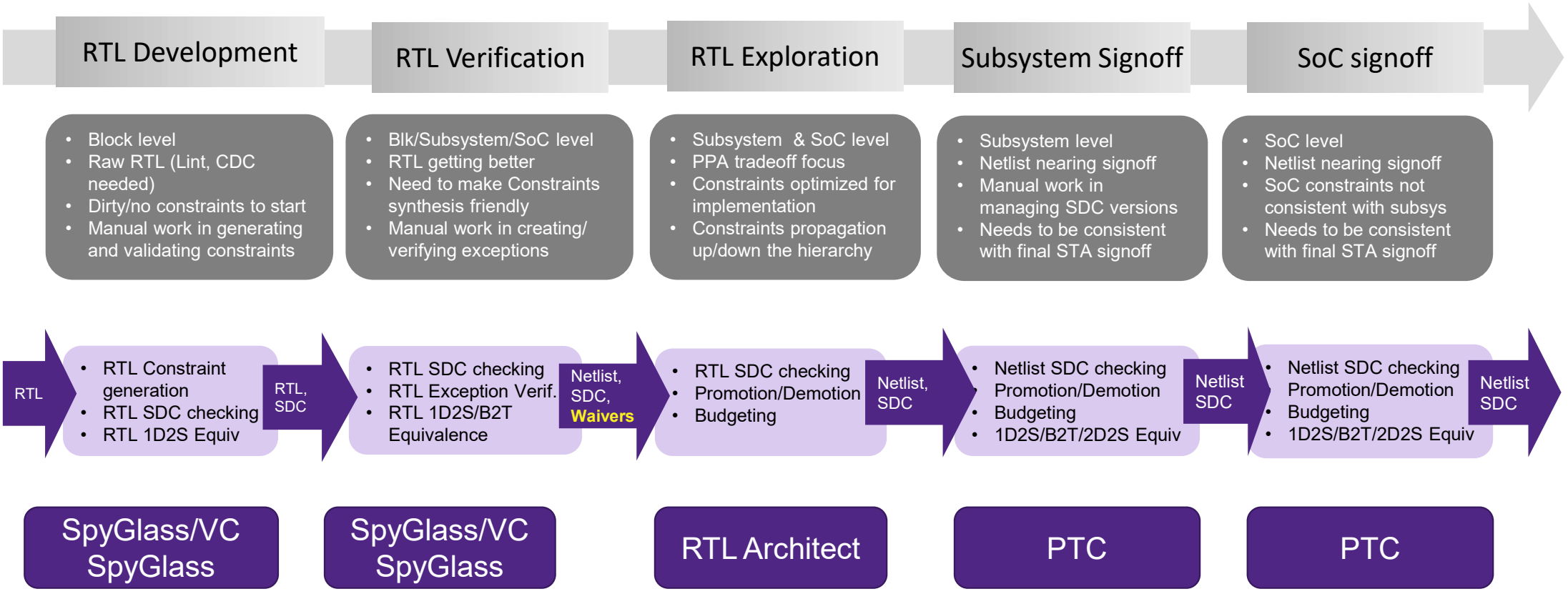



- Reduce Long Setup Cycles
  - Eliminate interactive setup and manual corrections
  - Automatic extraction of clocks
- Identify and Resolve SDC Issues
  - Clock domain conflicts
  - Clock waveform issues
  - Set case analysis issues
  - General correctness and completeness
- CDC/RDC/LP/DFT Analysis Relies On Clean Clock Constraints for Accurate Analysis
  - Alternative, manage noise in analysis
  - Missed bugs!

SYNOPSYS®
Silicon to Software™

# Thank You

# Recommended Synopsys Constraints Solution



# SpyGlass Rules Mapping with PTC (1)

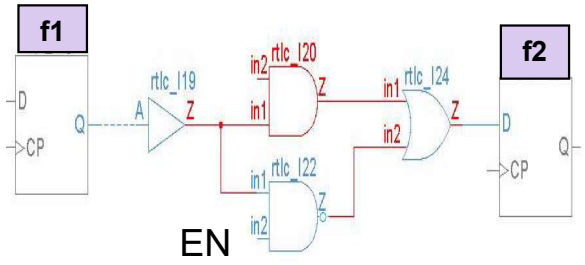
| SG Rule Name | Description | PTC Rule Name |
|---|---|---|
| Clk_Gen01a | Missing primary clock source | UDEF_0001 |
| SDC_209 | Generated clock with no master clock (dead clock) | CLK_0003 |
| Clk_Gen26 | Generated clock with ambiguous master clocks of different periods | CLK_0028 |
| SDC_257 | master_clock not present at -source object (no path from -master_clock, no path from -source) | CLK_0016 |
| Clk_Gen03 | create_generated_clock has no logic path from -source object (path exist from other clock sources) | CLK_0016 |
| Clk_Gen03 | create_generated_clock has no logic path from -source object (path do not exist from other clock sources) | CLK_0016 |
| Clk_Gen01b | application point of clock is a constant | CLK_0006 |
| Clk_Gen01b | source point of a generated clock is a constant | CLK_0007 |
| Clk_Gen01b | Floating or Undriven register clock pins | DES_0002 |
| Clk_Gen38 | clock propagation overlaps with user-applied case analysis | CLK_0042 |
| Clk_Gen09 | create_clock blocks clock propagation | CLK_0043 |
| SCG05 | Missing physically-exclusive set_clock_group between clocks | CGR_0007 |
| SCG08 | Missing logically-exclusive set_clock_group between clocks | UDEF_0014 |
| SCG06 | Missing asynchronous (non-harmonic) set_clock_group between clocks | CGR_0010 |
| SCG02 | Missing physically-exclusive:inherit set_clock_group between clocks | CGR_0006 |
| SCG01 | Incorrect interclock set_clock_group/set_false_path for various categories of sync clock pairs | CGR_0002 |
| SCG01 | Incorrect interclock set_clock_group/set_false_path for various categories of sync clock pairs | CGR_0001 |
| SCG01 | Incorrect set_clock_groups , source sync clocks/Incorrect interclock false path,<br>source-synchronous. | UDEF_018G |
| Clk_Gen38 | set_case_analysis or propagated-constant register clock pins<br>This rule identifies register clock pins that are driven by a constant. | CLK_0042 |
| Clk_Gen02 | Clock has no sink | CLK_0021 |
| Clk_Gen34 | clock is overridden by another clock with same name | UIC-034 |
| Op_Del16 | Source-Sync Output Data port not constrained to source-sync clock | UDEF_add_rule6 |
| Inp_Del08/Op_Del08 | input output delay constrains is overriden because another input output delay constrains at same point without -add | UIC-043 |
| False_Path12 | Clock-Clock False path -setup not covered by -hold | UDEF_add_rule3 |
| MCP09 | Clock-Clock Multicycle path -setup not covered by -hold | UDEF_add_rule4 |
| Clk_Lat08 | negative latency value | UIC-028 |

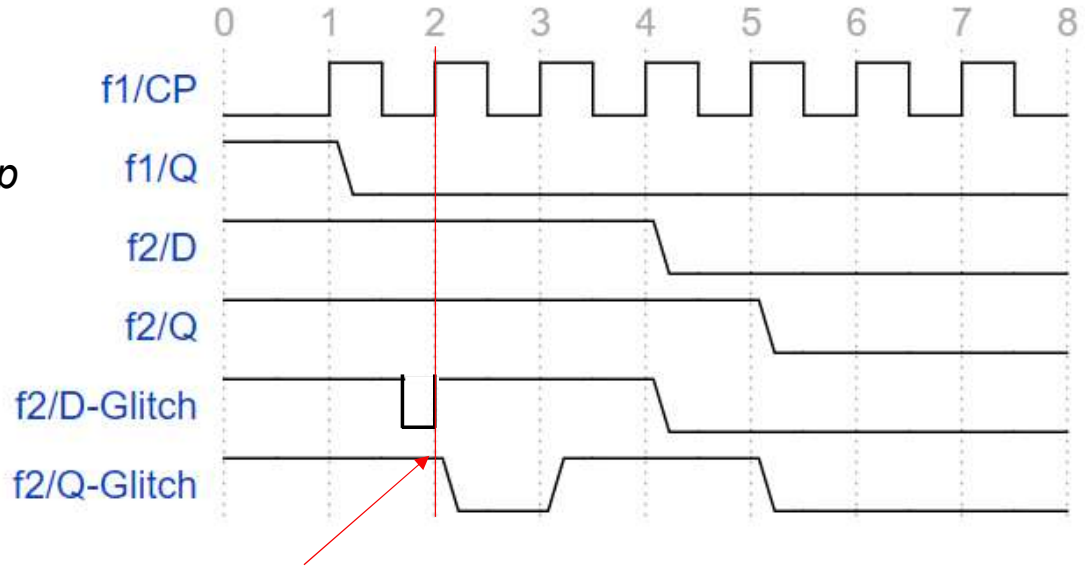
# MCP Glitch Verification

- MCP paths can have glitches due to reconverging logic that can be sampled
  - Spurious transition at sink flop

- Potential glitch with and without MCP hold specification

*set_multicycle_path 3 –from f1/CP -to f2/D -setup*




Glitch sampled before 3 setup delay