Automated Constraints Promotion Methodology from IP to SoC Designs for Complex Designs

Mallik Devulapalli & Ajay Daga
Synopsys

Agenda

- Addressing the Need for Timing Constraints Promotion
  - Overview of Synopsys Timing Constraints Manager
- Case Study - SDC Promotion Using Timing Constraints Manager
  - Leveraging Early PCIe Gen 6 Sub System Configuration
  - Getting a Jumpstart on SDC Promotion Using Fusion QIKs
  - Results on Early PCIe Gen 6 Design
- Conclusion
- Q & A

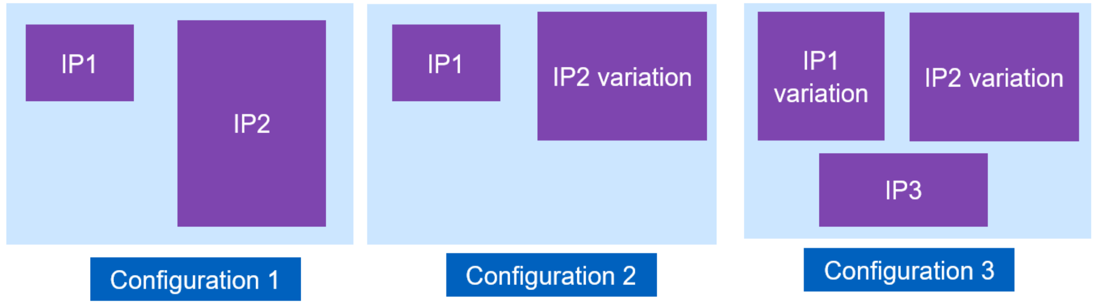
Addressing the Need for Timing Constraints Promotion Before Synthesis

Overview of Synopsys Timing Constraints Manager

Bottom-up Hierarchical Design Integration

What is sub-system integration?

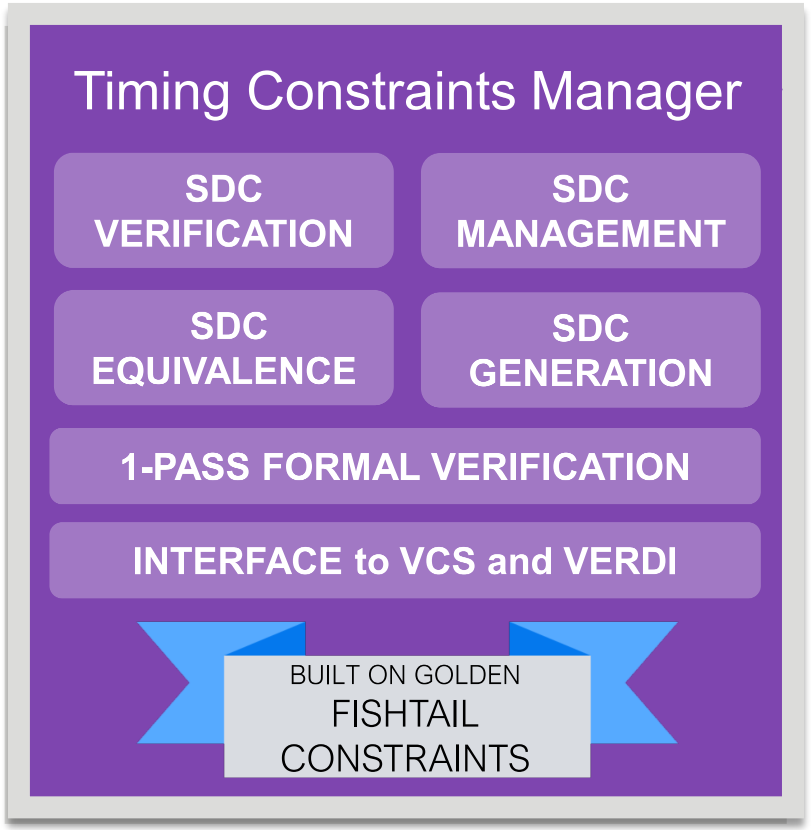
- SoCs involve integration of several sub-systems
  - Ex. Interface IPs: PCI Express (PCIe) or USB
  - Computing IPs: Micro-processors or Graphics processors
  - Communication IPs: Modems
    Timing constraints at individual IPs are well-defined and exercised
- Timing constraints take various shapes when IPs are put together to form sub-system
  - Number of computing units may change
  - Different number of clocks, different interface standards
  - Higher or lower bandwidth for IP variations
    - May demand different bus dimensions
    - Clocks, exceptions need to be manually rethought and verified



Timing constraints don't exist for newer configurations and require manual effort

Synopsys Timing Constraints Manager (TCM)

Built on FishTail's Best-in-Class SDC Constraints Solution


Timing Constraints Manager
SDC VERIFICATION
SDC MANAGEMENT
SDC EQUIVALENCE
SDC GENERATION
1-PASS FORMAL VERIFICATION
INTERFACE to VCS and VERDI
BUILT ON GOLDEN FISHTAIL CONSTRAINTS

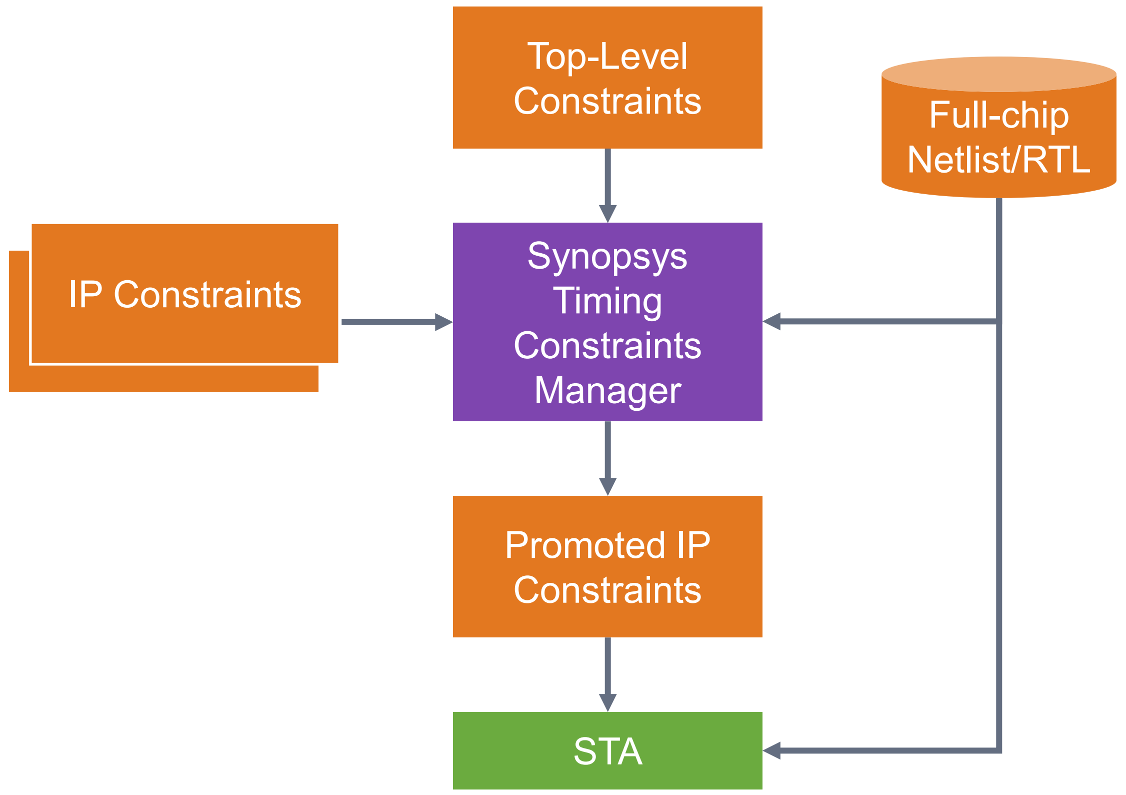
- Comprehensive SDC Timing Constraints Generation, Verification and Management
- SDC Verification with No Noise
  - SDC Signoff at RTL handoff using proprietary formal & assertion technology
- Automated, Accurate SDC Promotion
  - Saves weeks and eliminates bugs from manual reuse of IP constraints in SoCs
- Automated SDC Generation from RTL
  - Improves designer productivity and design PPA

90% Reduction in Designer SDC Review Effort	Auto SDC Gen/Mgmt. Shortens SoC Execution **by 6-8 Weeks**	**Eliminate Silicon Failure** from SDC Bugs

Deployed at top-tier semi companies with production-proven tape-outs

TCM Constraint Promotion Flow

- Promote standalone block-level constraints to chip-level
  - Reduces effort and eliminates errors resulting from maintain two versions of the same constraints (for standalone and in-context use)
  - Automatically stitches block and top-level clocks
  - Preserves user Tcl for timing exceptions



TCM SDC Promotion Methodology on Early PCIe Gen 6

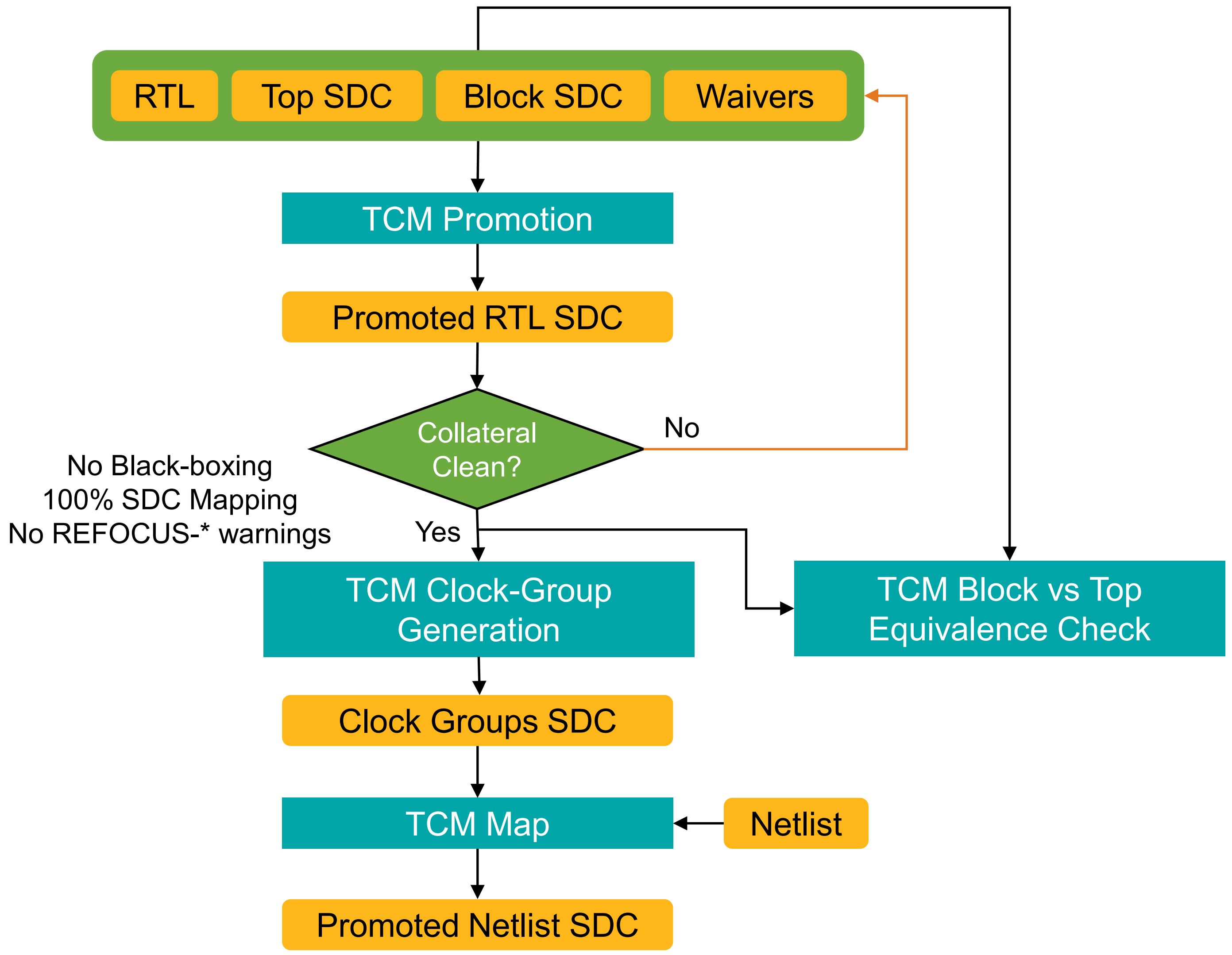
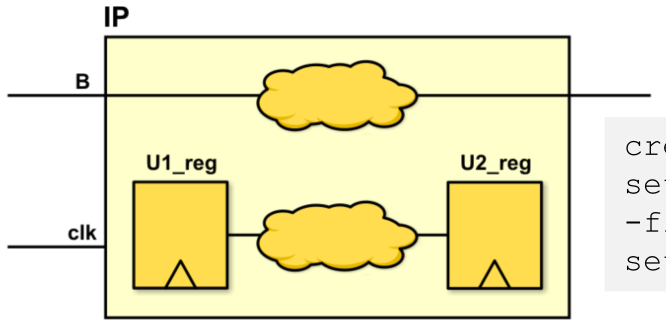
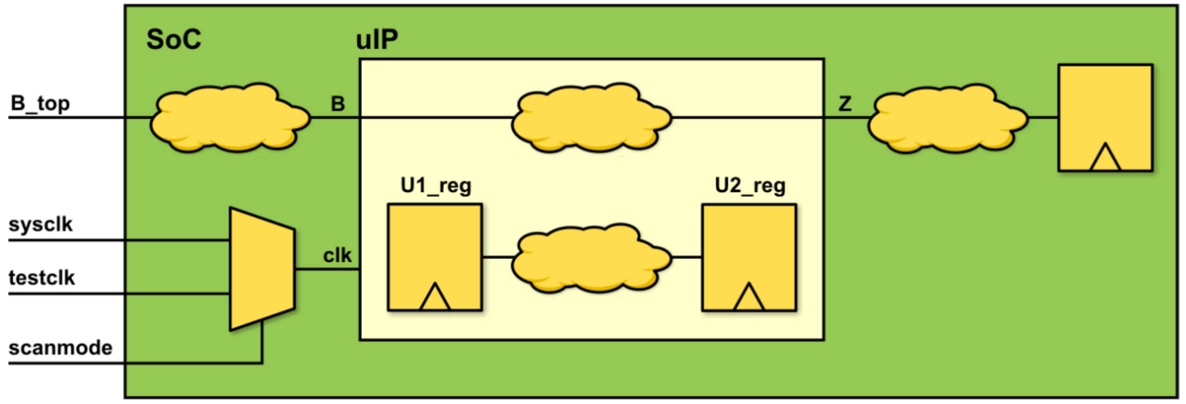


Illustration of Constraint Promotion



SDC for IP

```
create_clock clk -period 6 -name clk
set_false_path -from [get_pins -of [get_cells -hier
-filter full_name=~*U1*] -filter is_clock_pin==true]
set_multicycle_path -from clk
```



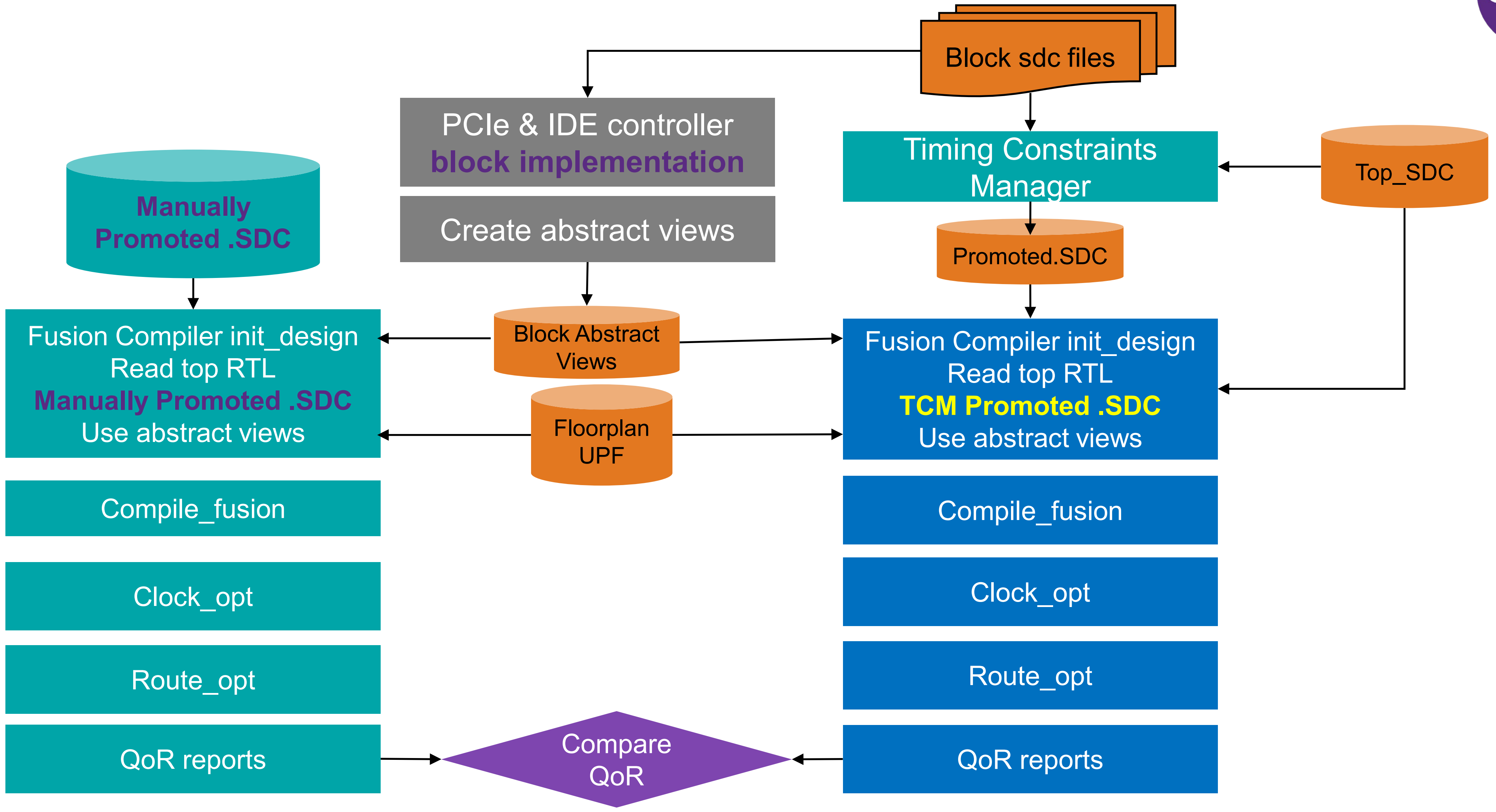
SDC for SoC

```
create_clock sysclk -period 6
create_clock testclk -period 6
set_case_analysis 0 scanmode
set_clock_groups -async \
-group sysclk -group testclk
```

Promoted SDC

```
current_instance uIP
set_false_path -from [get_pins -of [get_cells -hier -filter full_name=~*U1*]
-filter is_clock_pin==true]
current_instance
create_generated_clock -divide_by 1 -master sysclk  -name sysclk_1 uIP/clk
set_multicycle_path -from sysclk_1
set_clock_groups -async -group {sysclk_1} -group {testclk}
```

Synopsys Fusion Compiler Flow with TCM Promotion



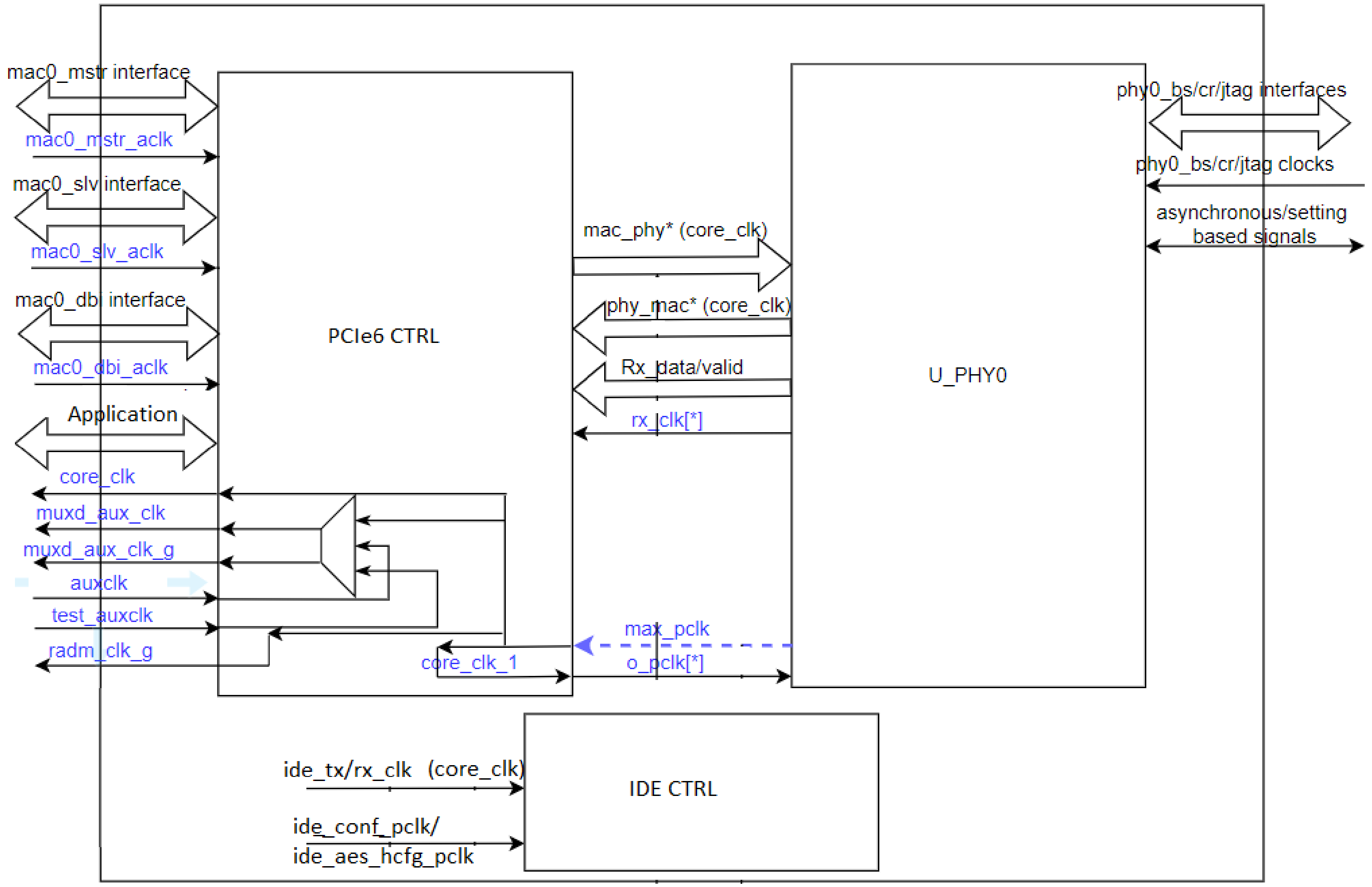
SDC Promotion Using TCM

Case Study Leveraging Early PCIe Gen 6 Subsystem Configuration

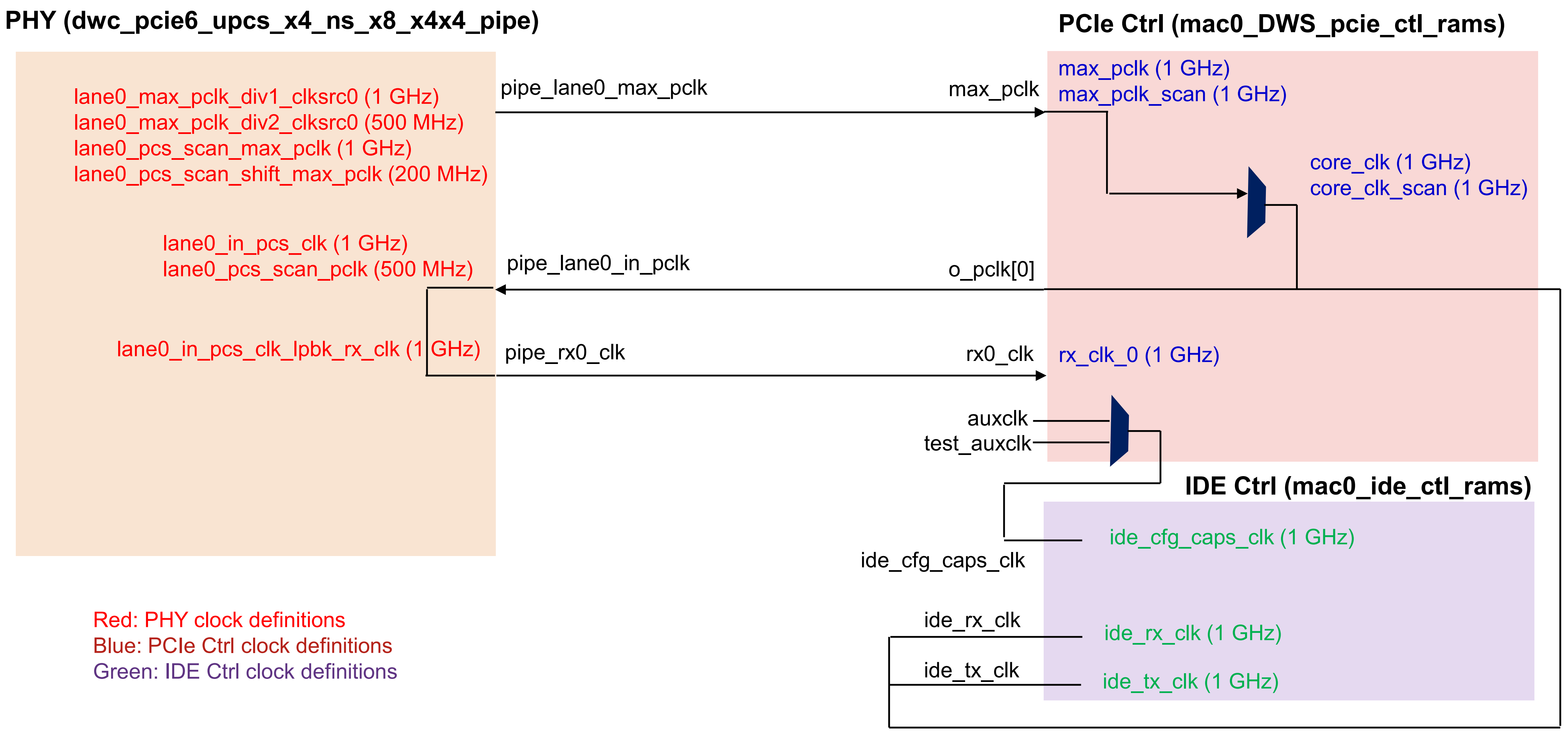
Design Specifics of PCIe Top SS

Gen6 top SS early release

- PCIe controller & IDE controller blocks are implemented as Hard macro



Design Specifics - PCIe Gen6 Clock Definitions


PHY (dwc_pcie6_upcs_x4_ns_x8_x4x4_pipe)
lane0_max_pclk_div1_clksrc0 (1 GHz)
lane0_max_pclk_div2_clksrc0 (500 MHz)
lane0_pcs_scan_max_pclk (1 GHz)
lane0_pcs_scan_shift_max_pclk (200 MHz)
lane0_in_pcs_clk (1 GHz)
lane0_pcs_scan_pclk (500 MHz)
lane0_in_pcs_clk_lpbk_rx_clk (1 GHz)
pipe_lane0_max_pclk
pipe_lane0_in_pclk
pipe_rx0_clk
PCIe Ctrl (mac0_DWS_pcie_ctl_rams)
max_pclk
max_pclk (1 GHz)
max_pclk_scan (1 GHz)
core_clk (1 GHz)
core_clk_scan (1 GHz)
o_pclk[0]
rx0_clk
rx_clk_0 (1 GHz)
auxclk
test_auxclk
IDE Ctrl (mac0_ide_ctl_rams)
ide_cfg_caps_clk
ide_cfg_caps_clk (1 GHz)
ide_rx_clk
ide_rx_clk (1 GHz)
ide_tx_clk
ide_tx_clk (1 GHz)

Red: PHY clock definitions
Blue: PCIe Ctrl clock definitions
Green: IDE Ctrl clock definitions

PCIe Gen 6 TCM Constraints Promotion Flow

Outline of steps

- Required input collateral for running TCM:
  - Full RTL design file list
    - Include entire design subsystem which would include IP RTL
  - Liberty db files (standard cells and/or memory cells)
    - Verilog behavioral models can be used if liberty models are unavailable
  - Sub design SDC constraints in Tcl format
    - SDC format is acceptable, Tcl is preferred
  - Top level clocks file
  - Any waiver files provided by IP verification by IP provider

Flow Steps

1. Flow inputs
2. Pre-requisites
3. Additional changes
4. Preparing config file
5. Run promotion
6. Review output logs
7. Viewing results
8. Writing waivers
9. Formal equivalence

Prerequisites of TCM Promotion Flow

Applicable to all sub-designs or IP cores

- Ensure all sub designs/IP cores are run through TCM SDC verification flow
  - Read Tcl constraints, RTL and perform SDC verification
    - Avoid issues faced in promotion flow having origin in the block-level constraints
    - Separate training resources are available

- Ensure block-level constraints are signed off
  - Example : A clock is not propagating to output of mux, we would not expect it to be promoted
  - If a constraint or Tcl structure is not friendly with TCM, user has opportunity to fix it first

- Verilog file lists along with `defines, and necessary variables to resolve RTL
  - set verilog_defines “SYNTHESIS” ;# recommend to add SYNTHESIS to all Synopsys cores
  - set verilog_parameters “rtl_parameters.cfg”
  - set mapping_include_files “rtl_variables.config.tcl”

Before Starting TCM Promotion Flow

Applicable to all sub-designs or cores

- Prepare a top-level clocks file
  - This file should include
    - All primary and virtual clocks used purely at top level

      `If we expect the clock to connect with promoted clock, it is essential to maintain same clock period`
    - Generated clocks used purely at top level
    - Clock groupings

      `Any known asynchronous clock relationships`
    - IO delays, if required at this level
- Treat all IP blocks as a sub-block for constraint promotion flow
  - All constraints that need to be promoted should be at block level scope
  - Top level contains only instantiation of blocks and some glue logic but not IP blocks
  - This approach keeps the constraints space very clean and helps promotion of constraints

Preparing Configuration Files

Customizing template

- Download template config file and cheat sheet from TCM installation
  - `Unix %> cp $FISHTAIL_HOME/cfs_config_files/rtl_constraint_promotion.cfg`
  - `Unix %> cp $FISHTAIL_HOME/doc/tcm/doc/cheat_sheets/RTLConstraintPromotionCheatSheet.pdf`
- Customize sections in rtl_constraint_promotion.cfg (Example)

```
set design_name "pcie6_snps_hardening_dm8sx8_gen6_axi_subsys"
set file_list "verilog_lists/pcie6_snps_hardening_dm8sx8_gen6_axi_subsys.add_mux.lst"
set block_list {
mac0_ide_ctl_rams
mac0_DWS_pcie_ctl_rams
dwc_pcie6_upcs_x4_ns_x8_x4x4_pipe
}
set block_sdc_files {
tcl_constraints/mac0_ide_ctl_rams.tcl
tcl_constraints/mac0_DWS_pcie_ctl_rams.mod.core_clk.add_scan.tcl
tcl_constraints/pcie6_snps_hardening_dm8sx8_gen6_axi_subsys.phy.tcl
}
set top_level_clocks_for_promotion {
tcl_constraints/top_clocks.tcl
tcl_constraints/Uncertainity.tcl
}
set lib_models_are_dbs true
set lib_file_dirs {
.. }
```

Running Promotion

Customizing template

- After config file is suitably edited, run these commands
  - `Unix %> create_fishtail_scripts -config_file rtl_constraint_promotion.cfg`
  - `Unix %> cd fishtail_rtl_constraint_promotion/constraint_promotion/`
  - `Unix %> promote_constraints`
- New SDC file promoted.sdc is generated
  - Comments indicate which lines of source sdc files were used to promote

```
# SDC file generated by FishTail Refocus version U-2022.12-SP5-1
# From /mac0_ide_ctl_rams.tcl line 13 for instance u_mac_subsys/u_ide_ctl_rams
create_generated_clock -name ide_aes_hcfg_pclk_ft_ide_aes_hcfg_pclk_ft_I_u_ide_ctl_rams -master_clock
ide_aes_hcfg_pclk -add [get_pins { u_mac_subsys/u_ide_ctl_rams/ide_aes_hcfg_pclk }] -source [get_ports {
ide_aes_hcfg_pclk }] -divide_by 1 -combinational
..
set_clock_uncertainty -fall_from [get_clocks { phy0_pll0_word_clk_o }] -rise_to [get_clocks {
phy0_pll0_word_clk_o }] $::fishtail::phy0_pll0_word_clk_o_setup_fall_from_rise_to_uncertainty -setup

..
# Auto-Generated clock group because multiple tool inserted div-1 generated clocks are defined on the same
pin
set_clock_groups -physically_exclusive -group { core_clk_ft_I_u_phy } -group { core_clk_ft_lane0_in_pcs_clk
} -group { core_clk_scan_ft_I_u_phy } -group { core_clk_scan_ft_lane0_pcs_scan_pclk }

set_false_path -through [get_pins { u_mac_subsys/u_pcie_ctl_rams/rx_lane_flip_en }]
```

Verifying Output and Resolving Issues

What to watch out for

- Un-resolved modules: ensure that the hand mapped cells are found
- Constraints that were not parsed due to incompatible tcl
  - Example: Don't include implementation tool (like Design Compiler or Fusion Compiler) attributes in tcl constraints
- What to ensure
  - The constraint mapping is 100% or close enough
  - Look at run_refocus.log
    - **`Info: Constraint mapping percentage: 99%`**
    - If it is not 100% check what constraints were skipped, and fix the constraints

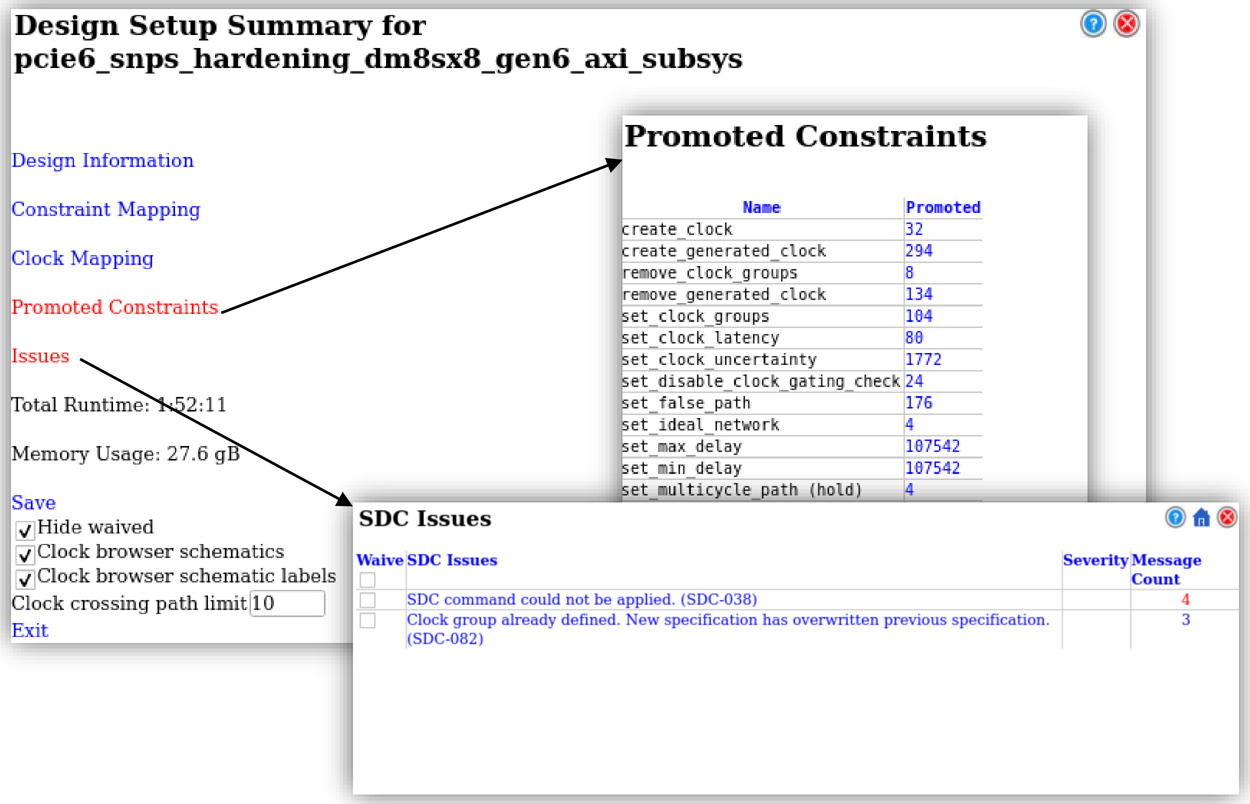
```
grep skipped run_refocus.log
Info: 0 create_clock command(s) were skipped
Info: 0 create_clock (virtual) command(s) were skipped
Info: 0 create_generated_clock command(s) were skipped
Info: 0 set_clock_groups command(s) were skipped
Info: 0 set_clock_latency command(s) were skipped
Info: 4 set_clock_uncertainty command(s) were skipped
Info: 0 set_false_path command(s) were skipped
Info: 0 set_ideal_network command(s) were skipped
Info: 0 set_max_delay command(s) were skipped
Info: 0 set_min_delay command(s) were skipped
Info: 0 set_multicycle_path (hold) command(s) were skipped
Info: 0 set_multicycle_path (setup) command(s) were skipped
Info: 0 set_sense command(s) were skipped
```

```
Info: 32 create_clock command(s) were written to the output file(s)
Info: 294 create_generated_clock command(s) were written to the output file(s)
Info: 8 remove_clock_groups command(s) were written to the output file(s)
Info: 134 remove_generated_clock command(s) were written to the output file(s)
Info: 104 set_clock_groups command(s) were written to the output file(s)
Info: 80 set_clock_latency command(s) were written to the output file(s)
Info: 1772 set_clock_uncertainty command(s) were written to the output file(s)
Info: 24 set_disable_clock_gating_check command(s) were written to the output file(s)
Info: 176 set_false_path command(s) were written to the output file(s)
Info: 4 set_ideal_network command(s) were written to the output file(s)
Info: 107542 set_max_delay command(s) were written to the output file(s)
Info: 107542 set_min_delay command(s) were written to the output file(s)
Info: 4 set_multicycle_path (hold) command(s) were written to the output file(s)
Info: 4 set_multicycle_path (setup) command(s) were written to the output file(s)
Info: 58 set_sense command(s) were written to the output file(s)
```

Viewing the Results of the Promotion

Using the graphical user interface

- Bring up the results in UI
  - After promotion of clocks is run, browse results and take action to fix anything
    - `Unix %> module load tcm ; module load verdi`
    - `Unix %> cd fishtail_rtl_constraint_promotion/constraints_promotion`
    - `Unix %> view_fishtail_results -verdi`
- Browse design issues, add any waivers until all issues are resolved



Design Setup Summary for pcie6_snps_hardening_dm8sx8_gen6_axi_subsys

Design Information

Constraint Mapping

Clock Mapping

Promoted Constraints

Issues

Total Runtime: 1:52:11

Memory Usage: 27.6 gB

Save

✓ Hide waived

✓ Clock browser schematics

✓ Clock browser schematic labels

Clock crossing path limit 10

Exit

Promoted Constraints

Name	Promoted
create_clock	32
create_generated_clock	294
remove_clock_groups	8
remove_generated_clock	134
set_clock_groups	104
set_clock_latency	80
set_clock_uncertainty	1772
set_disable_clock_gating_check	24
set_false_path	176
set_ideal_network	4
set_max_delay	107542
set_min_delay	107542
set_multicycle_path (hold)	4

SDC Issues

Waive	SDC Issues	Severity	Message Count
☐			
☐	SDC command could not be applied. (SDC-038)		4
☐	Clock group already defined. New specification has overwritten previous specification. (SDC-082)		3

Customizing Waivers

Guidance and refinement with waivers

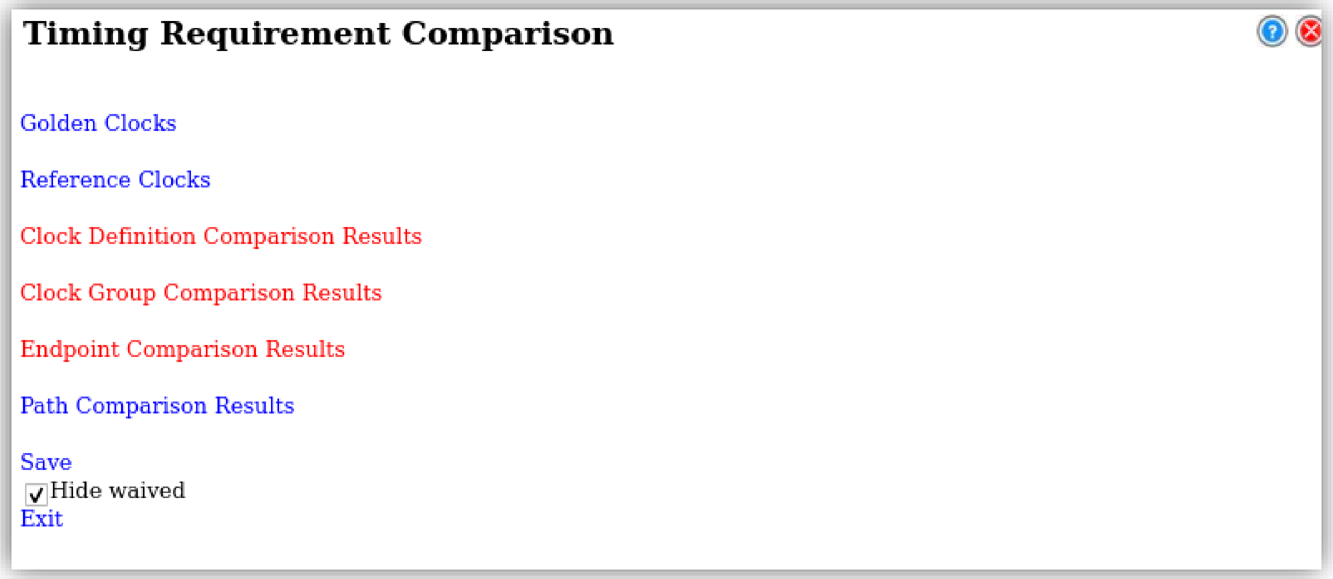
- Waivers file can be specified in the config for promotion flow
  - Example : When the clock in the block should be recognized same as the top level clock
- When multiple clocks arrive from top level at an input on the IP, TCM needs guidance
- Examples:
  - Config file rtl_constraint_promotion.cfg contains:
    - `set external_waive_tcl_file "my_waive.tcl"`

```
#my_waive.tcl
map_top_clock_to_block_clock -instance { u_mac_subsys/u_ide_ctl_rams } -block_clock { ide_aes_hcfg_pclk } -top_clock { ide_aes_hcfg_pclk }

map_top_clock_to_block_clock -instance { u_mac_subsys/u_pcie_ctl_rams } -block_clock { max_pclk } -top_clock { lane0_max_pclk_div1_clksrc0
lane0_max_pclk_div2_clksrc0 }

map_top_clock_to_block_clock -instance {  u_mac_subsys/u_pcie_ctl_rams  } -block_clock {rx_clk_0 } -top_clock { lane0_phy_rx_serdes_clk
lane0_phy_rx_serdes_clk_dser_int_clk lane0_rx_scan_ms_clk lane0_rx_scan_shift_clk lane0_in_pcs_clk_lpbk_rx_clk}
 map_top_clock_to_block_clock -instance {  u_mac_subsys/u_pcie_ctl_rams  } -block_clock {rx_clk_1 } -top_clock { lane1_phy_rx_serdes_clk
lane1_phy_rx_serdes_clk_dser_int_clk lane1_rx_scan_ms_clk lane1_rx_scan_shift_clk lane1_in_pcs_clk_lpbk_rx_clk}
..
```

Equivalence Checking of Promoted Constraints

Validation of IP vs promoted constraints

- TCM verifies IP constraints (golden) vs promoted constraints (reference)
  - After promotion of clocks is run, next step is to run the equivalency check
    - `Unix %> cd fishtail_rtl_constraint_promotion/equivalence_checking`
    - `Unix %> check_equivalence`
  - This step runs clock groups generation, does formal equivalency check
- What to ensure?
  - Endpoint comparison is 100%
  - Clock definition comparison is OK
  - Path Comparison is 100%
- What next?
  - Inspect and debug results
    - For mismatches
    - Add waivers if needed


Timing Requirement Comparison

Golden Clocks

Reference Clocks

Clock Definition Comparison Results

Clock Group Comparison Results

Endpoint Comparison Results

Path Comparison Results

Save
[✓] Hide waived
Exit

Getting a Jumpstart on SDC Promotion using Fusion QIKs

How can implementation engineers benefit?

Additional Automation for Ease of Use & PPA

Quickstart Implementation Kit (QIK Flow)

- Synopsys Fusion Compiler™ users can get a head start with QIK automation
  - Easily generate configuration file(s) and customize based on familiar Fusion Compiler RM 2.0 setup
  - Verify sub blocks
  - Apply waivers (manual analysis)
  - Fix any constraints issues
  - Run promotion flow

Contact Synopsys to download Fusion Compiler QiK setup for Timing Constraints Manager

Viewing and Verifying Results on Early PCIe Gen 6 Design

Results on Early Release of PCIe Gen 6 SS

Timing promotion and Fusion Compiler implementation

- Promotion
  - Runtime 5 hrs; Memory 37 GB
  - 39 map_top_clock_to_block_clock waivers
- Block vs Top Equivalence
  - Runtime 2 hrs; Memory 43 GB
  - Showed promotion results were correct
- Clock-Group Generation
  - Runtime 10 minutes; Memory 36 GB
  - 134 clock groups generated
- Netlist Mapping
  - Runtime 3 hrs; Memory 15 GB
  - 87% mapping; registers optimized out, abstracted models

- Fusion compiler QoR results show similar PPA
- Formal equivalence was established with Synopsys Formality®
- Proves that the design is equivalent

route_opt	WNS(ns) R2R	TNS(ns) R2R
FC Baseline (manual promotion)	-0.447 ns	377 ns
FC with TCM (TCM flow)	-0.417 ns	345 ns

Conclusion

- Need for solution
  - Timing constraints for subsystems/SoCs need to be created through reuse of IP constraints
  - Current manual approach is time-consuming, introduces mistakes with manual promotion of constraints
- Synopsys Timing Constraints Manager
  - Comprehensive SDC management
    - TCM offers flows for SDC verification, Promotion, Demotion, SDC Equivalence among others
  - Customer success
    - Deployed at top tier semi companies with production-proven tape-outs
  - Scalable solution
    - Can be targeted to variety of IPs and configurations
  - Great turnaround time with integrity in IP constraint reuse
    - Cuts the time to solution from weeks to days.
- Fusion Compiler QiK is available to help with productivity

snug

THANK YOU

YOUR
INNOVATION
YOUR
COMMUNITY