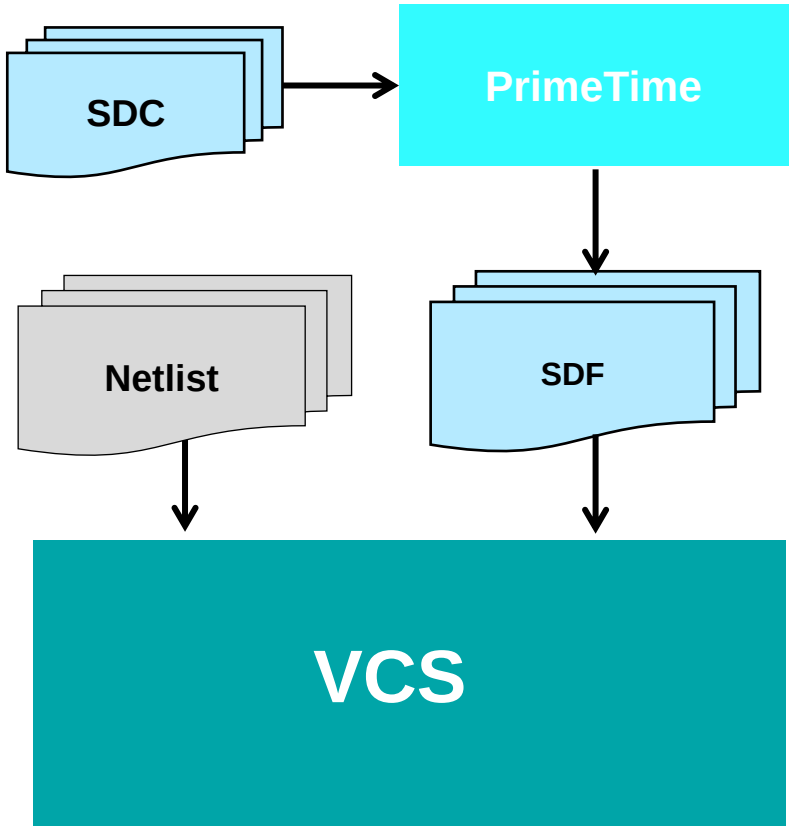
Connectivity Chip Design

The RTL-Level SDC Timing Exception Verification Ecosystem

ChienLin Huang, MediaTek
Evan Yang, Synopsys

Traditional Gate Level Simulation Flow

- Many chip makers do gate level simulations (GLS) in addition to RTL simulations, static and formal verification
- Designers rely on GLS to catch chip-killing bugs that other tools like static timing analysis (STA), assertion-based verification (ABV), static verification and emulation can't catch
- GLS netlist debug is cumbersome
- Becomes active only in the very late stages of the design process when the netlist and standard delay format (SDF) files are available



Existing SDC Verification Approaches

Need a fast and comprehensive solution that can verify SDC intent early in the design cycle

- **Gate-level simulations with full timing (SDF)**
  - Late in the development cycle (too close to tape-out)
  - Huge, slow and high debug effort
  - Often run only a test subset leading to low coverage

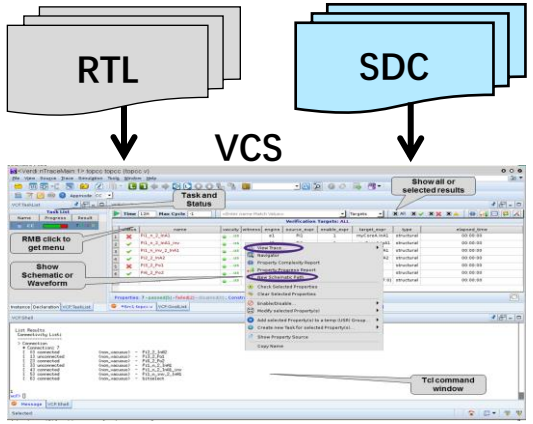
- **Formal/Static methods**
  - Great when they work – testbench-less & exhaustive
  - Typically exhibit capacity limitations and may require a complete environment (i.e., firmware?)
  - May lead to inconclusive results (or require inordinate amount of time)
  - QoR issues with noise

- **Proprietary RTL coding schemes (macros, assertions, …) for simulation**
  - Ad-hoc, non-standard and difficult to implement (instrumentation point)
  - May degrade performance

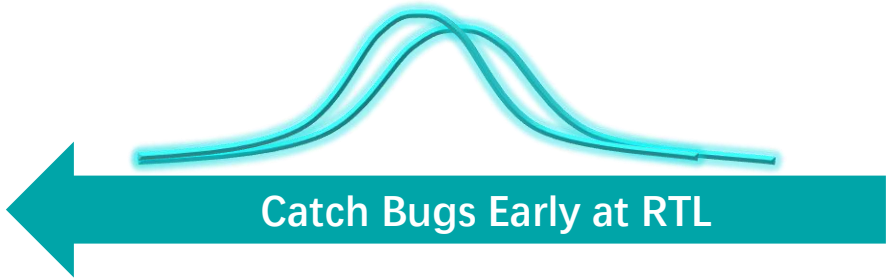
SDC Aware Verification in VCS

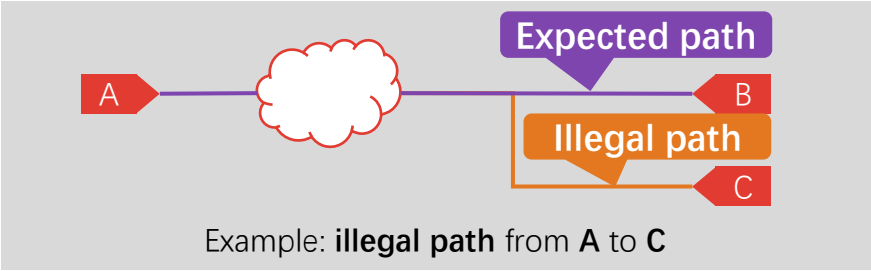
VCS-SDC feature

RTL Simulation



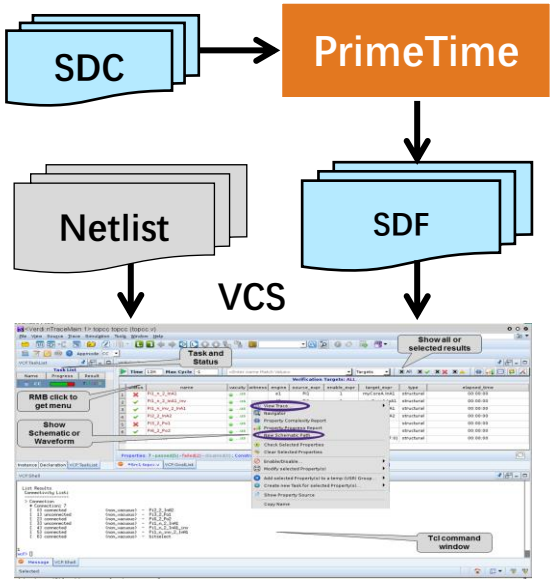
- Catch bugs early at RTL
- Faster runtime
- Leverage Verdi debug
- High coverage





Timing related Bugs in Design or SDC

- False Paths
- Multi Cycle Paths
- Clock Tree
- Asynchronous Paths & Resets



Gate Level Simulation

- Bugs found late
- Longer runtimes
- Hard to debug
- Low coverage

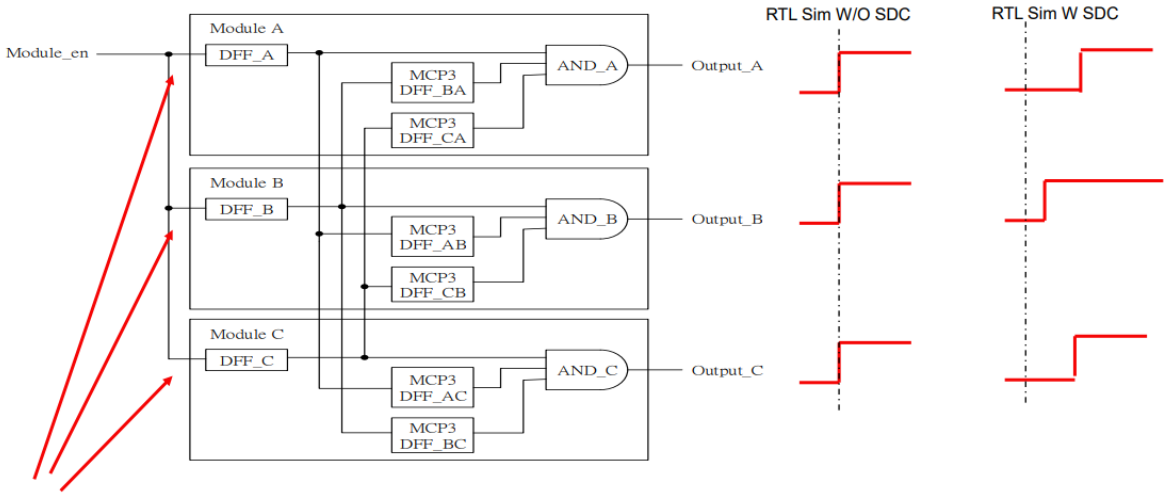
Verify SDC at RTL stage
-> increase 10X test coverage
-> change simulation time from week to days

Two issues were found at the try-run stage

Saving lots of time and $$

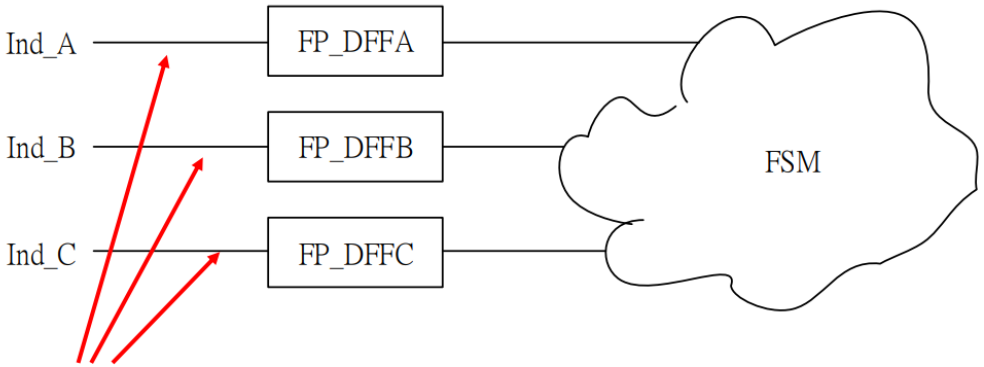
- Case1: CDC Reconvergence
- Case2: False path Reconvergence

"set_multicyc_path -to DFF_* " may cause output_* edge NOT aligned



VCS-SDC can insert different delays on different paths to trigger the design issue to make simulation failed.

"set_false_path -to FP_DFF*" causes different delays and make FSM hanging

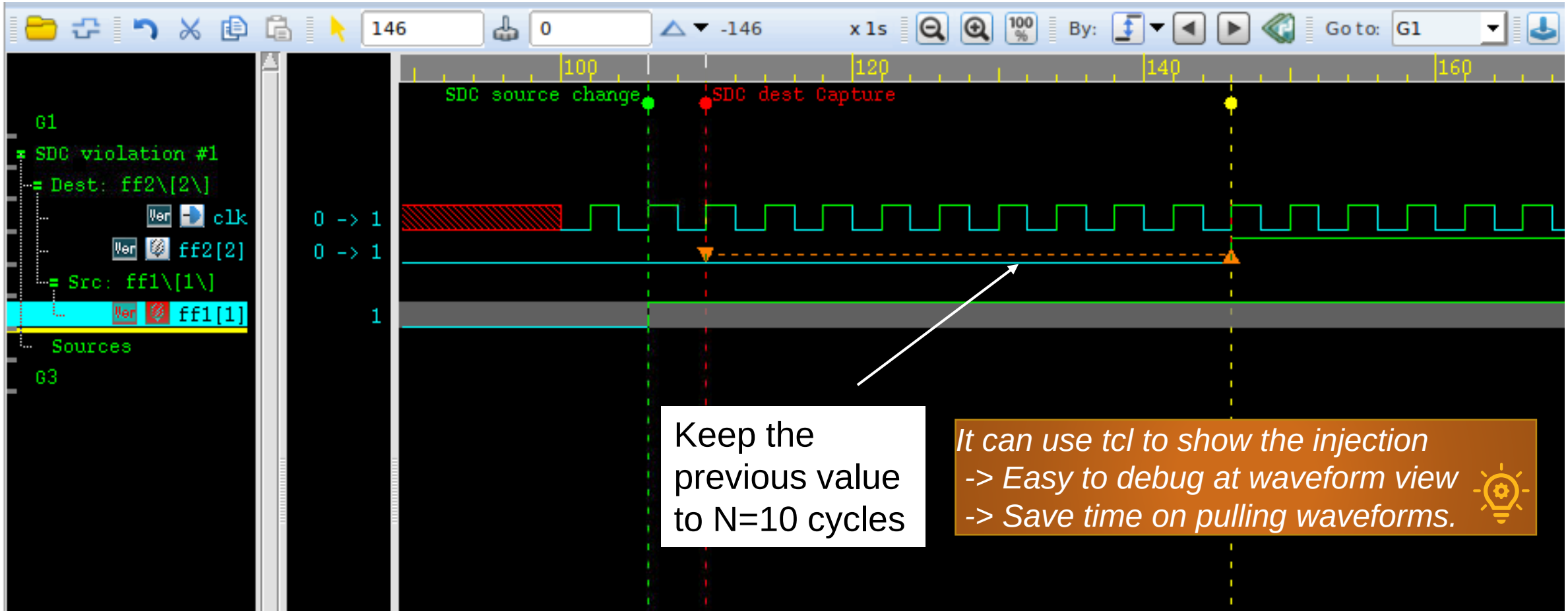


VCS-SDC can insert different delays on different paths to trigger the design issue to make simulation failed.

Debug Fail pattern

Verdi inject marker and DUTRCA feature: Inject marker

- set_multicycle_path 10 –from ff1* -to ff2*

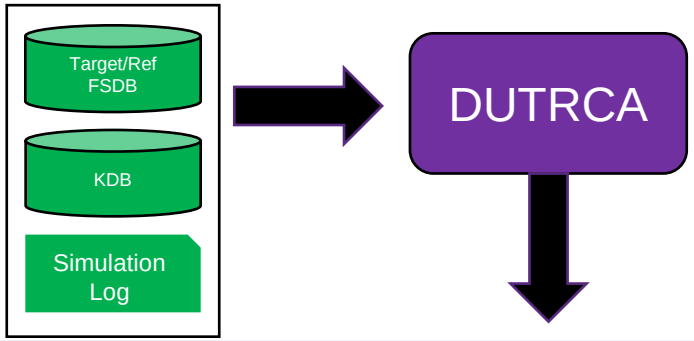


Keep the previous value to N=10 cycles

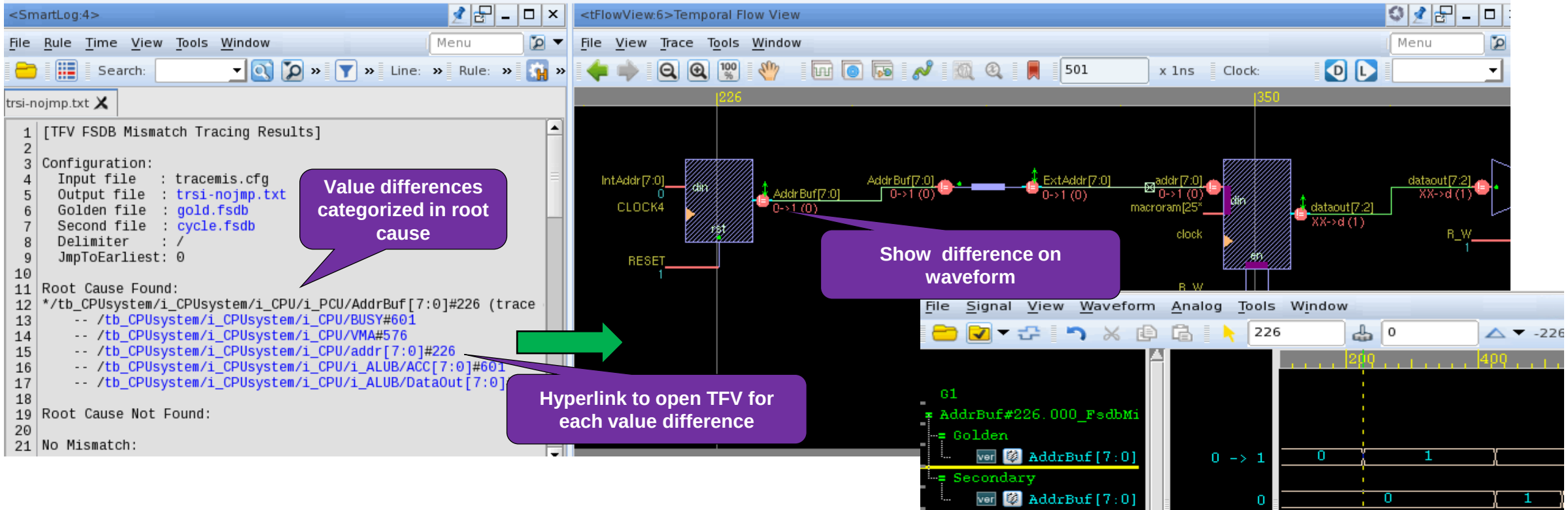
It can use tcl to show the injection
-> Easy to debug at waveform view
-> Save time on pulling waveforms.

Debug Fail pattern

Verdi inject marker and DUTRCA feature: DUTRCA

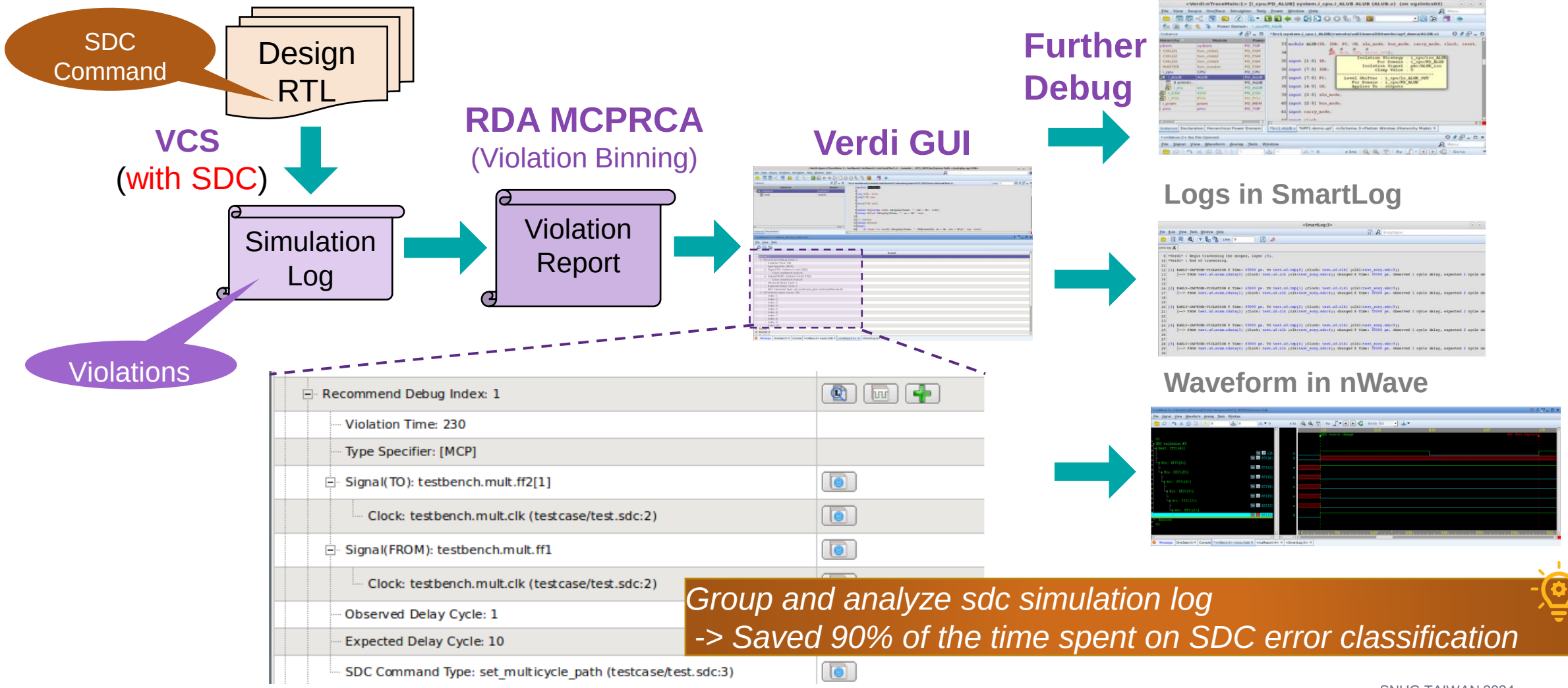


wo/w SDC delay injection
-> Save 90% designer debug effort
-> Save 50% DV root cause analysis time



Review the VCS-SDC simulation log

MCPRCA feature: Bins view



Group and analyze sdc simulation log
-> Saved 90% of the time spent on SDC error classification

Review the VCS-SDC simulation log

MCPRCA feature: SDC annotation in nTrace

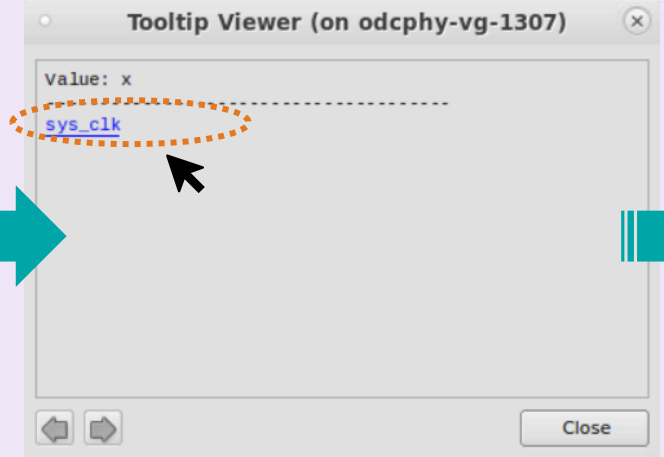
In Verdi GUI

nTrace

```
3  reg[7:0] ff1,ff2;
4  wire[7:0] mout;
5
6  // MCP source flop
7  always @(posedge clk or negedge rst)
                sys_clk
8          if (!rst) ff1 <= 0;
```

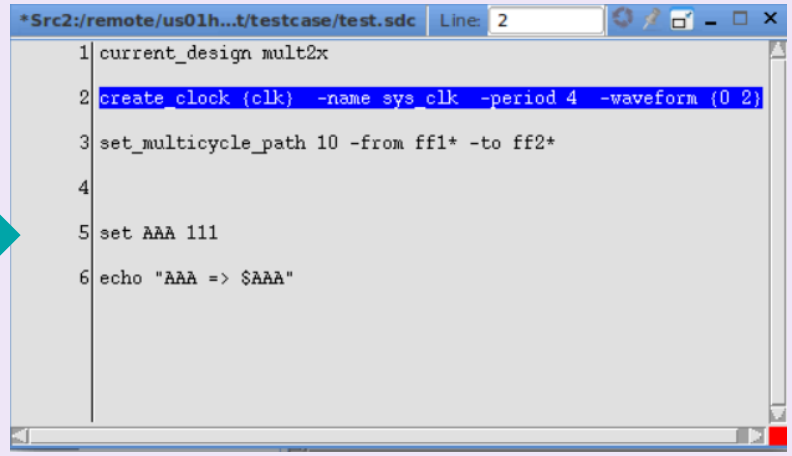
1. Press "shift" to open tool tip viewer

Tooltip Viewer



2. Click hyperlink to show SDC in Tooltip

nTrace (open in new tab)



3. The sdc command is shown in nTrace

All SDC information can be annotated
-> Very helpful for debugging
-> Easy to trace the clock

Review the VCS-SDC coverage

URG Coverage feature

- MCP Command:
  - Set_multicycle_path from ff2* to ff5*
  - Totally, there are four MCP paths covered by the MCP command
- 4-level signoff criterion (From coarse to fine)

Hierarchy | Modules | Groups | Asserts | SDC | Statistics | Tests

*

Name	Line	Type	Status	Score	Total	Active	Inactive	Violating
./Source/urg_SDC/mult_des_cov.sdc				0.00%	4	0	0	4
MCP (./Source/urg_SDC/mult_des_cov.sdc:4) -from ff1 -to ff2	4	MCP	VIOLATING	12.50%	8	1	6	1
MCP (./Source/urg_SDC/mult_des_cov.sdc:5) -from ff1... -to ff2	5	MCP	VIOLATING	12.50%	8	1	6	1
MCP (./Source/urg_SDC/mult_des_cov.sdc:10) -from b1 -to b3	10	MCP	VIOLATING	50.00%	12	6	4	2
MCP (./Source/urg_SDC/mult_des_cov.sdc:11) -from b2 -to b3	11	MCP	VIOLATING	41.67%	12	5	4	3

Four coverage model levels available
-> Reuse urg coverage system
-> An indicator of SDC verification level

Timing Constraints Management System

TCM SDC verification feature

Verification

- Formal Constraint Lint
- Generated Clock Verification
- Timing Exception Verification
- Disable Timing Verification

Management

- Constraint Mapping
- Constraint Promotion
- Constraint Demotion
- Mode Merging
- SDC Equivalence

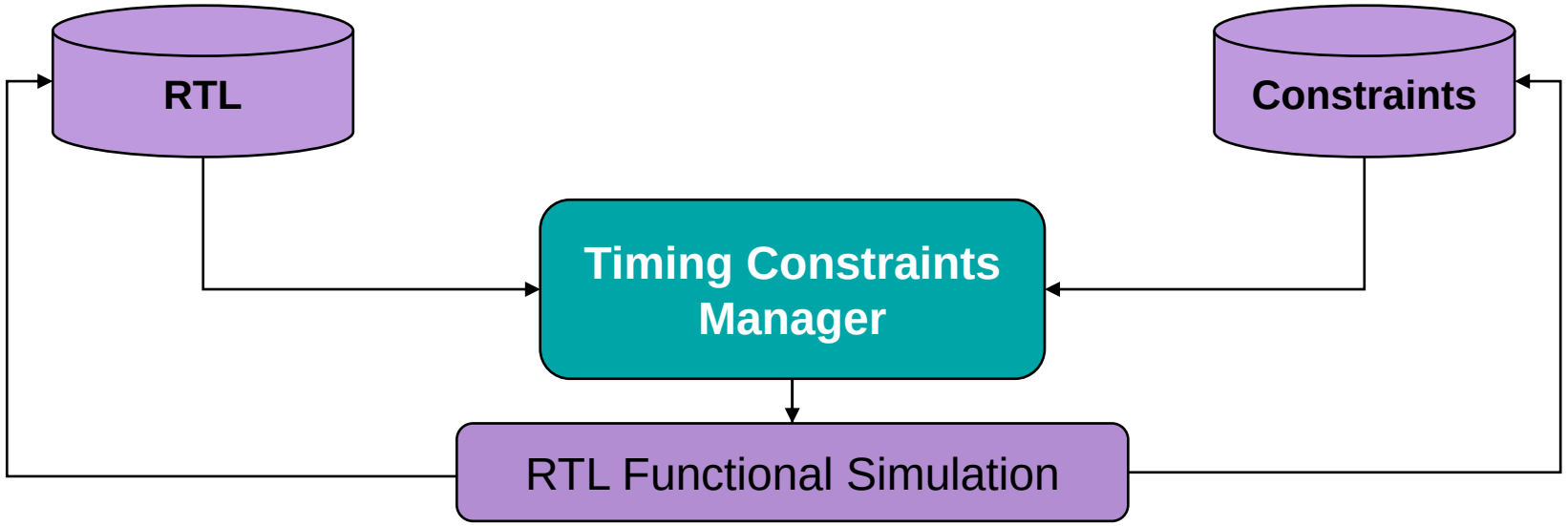
Generation

- Clock Generation
- Clock Group Generation
- Clock Sense Generation
- Timing Exception Generation
- Lib Generation

Constraint Lint and Verification are very useful
-> No need debug mapping issue at VCS-SDC
-> VCS-SDC becomes a means for TCM SDC verification

Timing Constraints Management System

TCM SDC Verification Feature



Use TCM formal verification
-> reduce 90% simulation coverage
-> get lots of design assumptions

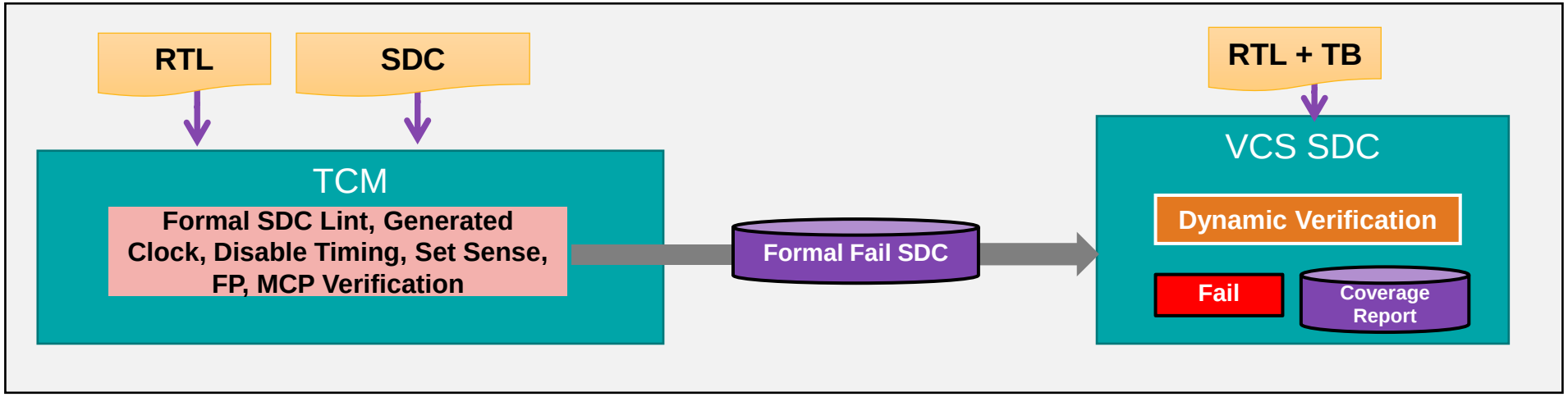
Formal verification of SDC constraints

- Flags constraint/design issues that are not caught by other tools
- Eliminates the need for gate-level sims and the risk of silicon failure from bad constraints
- Assertions generated for timing exceptions that fail formal proof
  - ✓Use of RTL simulation to verify generated assertions is recommended– reduces review effort
- RTL input is recommended
  - ✓RTL is not synthesized by the tool, allowing compact assertions to be generated for simulation
- Refocus maps gate-level signoff constraints to RTL

TCM + VCS-SDC

= Timing Exception Verification Ecosystem

An early stage of SDC verification
-> Caught design/SDC bugs at the RTL stage
-> Save 90% DE and DV effort



Best in Class Technology for Formal + Simulation Based Verification of SDC constraints

- Complete formal verification of all SDC constraints using TCM
- Timing-exceptions that fail formal proof are provided as input to VCS SDC
  - ✓Large majority of timing exceptions are proven formally with minimal user intervention
  - ✓Knowledge of static nets on a design is key to high formal pass percentage
- Fast, dynamic path sensitization-based, verification of formal failures in RTL simulation
  - ✓Native SDC support in simulation is more powerful and easier to manage than assertions
  - ✓Ease of debug and MCP/FP specific coverage

Conclusions

- VCS-SDC is a VCS feature that can model post-simulation behavior, allowing users to use more patterns at the RTL stage to **increase confidence in SDC verification**.
- DUTRCA and MCPRCA, as part of the VCS-SDC toolchain, can **significantly reduce the effort** required to debug failures caused by VCS-SDC injected delays.
- The URG-based SDC coverage system provides a familiar coverage interface, enabling users to review and waive, as well as **choose different levels of coverage models to control the amount of coverage.**
- The TCM SDC Lint feature can greatly reduce the effort needed to debug SDC mapping in VCS-SDC, and also **saves DV engineers from spending time learning SDC syntax-related knowledge**.
- The TCM SDC verification feature can **effectively increase the coverage rate and efficiently reduce the pattern count** required by VCS-SDC.
- Through the combination of TCM and VCS-SDC features, Synopsys provides a very powerful means of SDC verification at the RTL stage, allowing us to **significantly reduce the risk of SDC bugs within limited time and manpower**.

THANK YOU

Our
Technology,
Your
Innovation™