UAR Verification methodology using RDC

Shuchi, Ramananda Bhimireddy, Rama Krishna Reddy Lokareddy, Deepak Jindal, Suman Chalana

Qualcomm India Private Limited

AGENDA

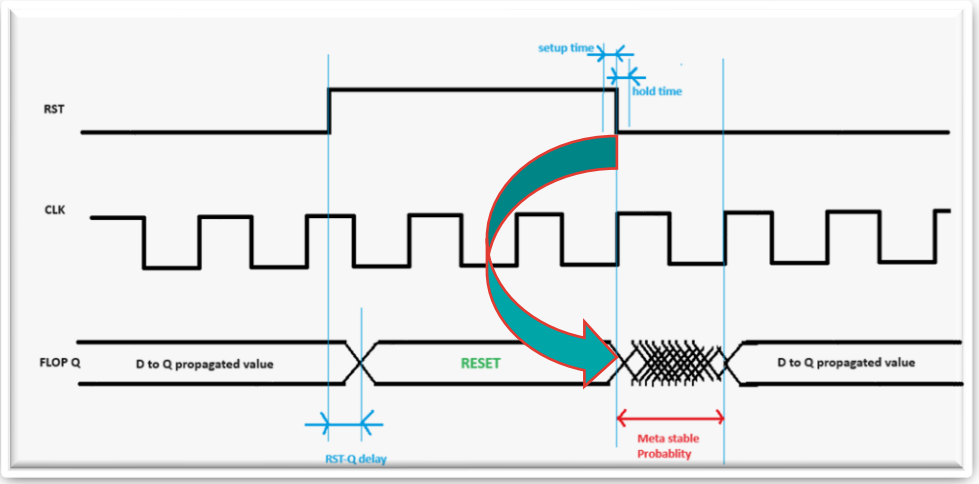
- Introduction
- DV Challenges
- Solution Methodology
- Merits & Summary
- Future Scope
- Acknowledgement

Introduction

Introduction

What is **U**niversal **A**synchronous **R**eset (UAR)?

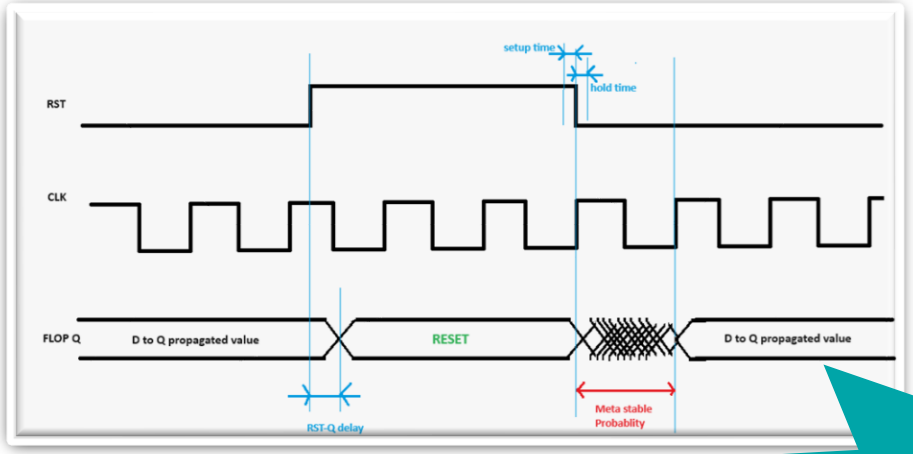
Synchronous Resets	Asynchronous Resets
✔ Path is timed (no metastability)	✘ Can cause metastability in flops on reset de-assertion
✘ Requires an active clock ✘ Incurs clock-related latency	✔ Does not require an active clock ✔ Has lower latency



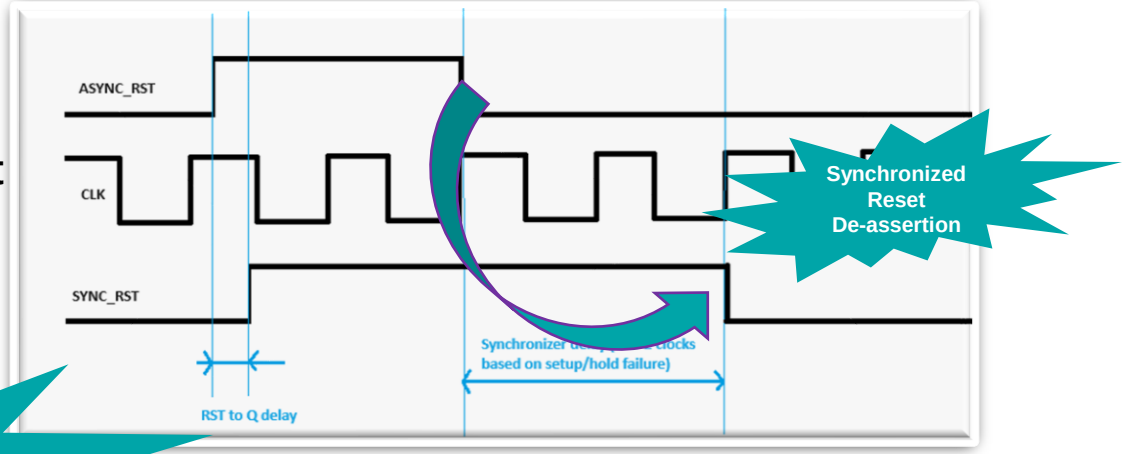
Although the relative timing between clock & async reset can be ignored during reset assertion, **the reset release must be synchronized to clock** to avoid recovery-removal violation and metastability.

Introduction

Async Reset with Reset Synchronizer



Using 2-flop Reset Synchronizer



UAR

Due to rising complexity of desi

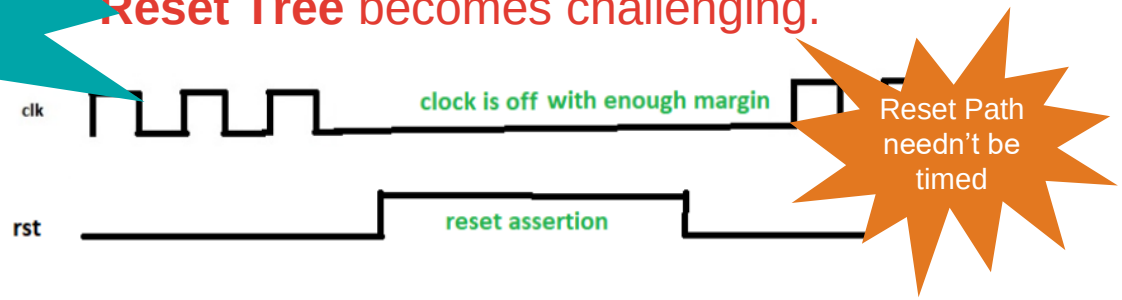
Switching OFF the clock durin

design techniques used widely

metastability issues without using

structure & with relaxed reset timing closure.

Reset Tree becomes challenging.



- Qualcomm CDMA Technologies (QCT) follow **UAR Reset Scheme** where we have 'asynchronous' assertion & 'asynchronous' de-assertion of reset.
- Clock to Flops is kept OFF during Reset duration and Reset Timing is Relaxed.

Challenges

Post-Silicon bugs in recent chips

CHIP1

- CPU boot didn't happen
- CPU came out of reset before PC value is latched correctly (loaded 0x0 value in PC)
- **CPU clock was running on reset de-assertion violating UAR.**
- Incorrect CDC waiver assuming UAR

CHIP2

- Microcontroller hang issue
- When the microcontroller was taken down and brought-up, IRQ indicating done is not generated causing hang.
- The IRQ got cleared (clock was not gated during reset de-assertion) before sending out.

Weeks/Months of debug led to identifying UAR issue

A single miss on async reset path can result in functional error or even overall chip failure.

No defined DV check/flow exists to ensure 100% UAR compliance in whole design

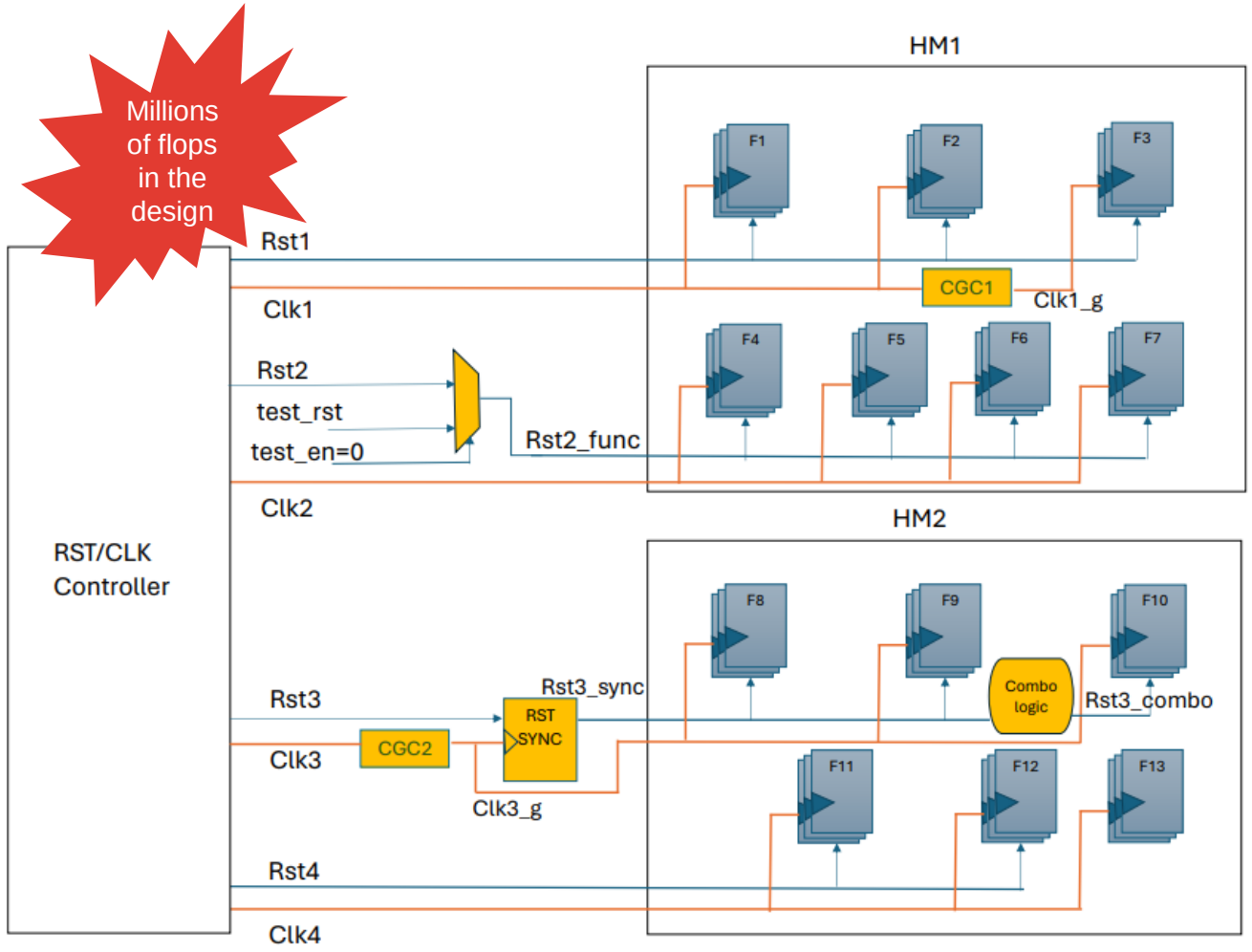
Is clock reaching every flop in the design GATED during Reset ?

Functional DV Simulations will not fail even if UAR is not followed.

DV Challenges

DV Challenges

Extracting flop-level clock-reset pairs



- DV is a challenge for checking each Flop
- Clocks coming from Primary-inputs/Clock controller/CGC's/CLKLIB elements/etc.
- Resets coming from Primary-inputs/Reset controller/Generated resets/etc.
- Validate SV assertions using dynamic simulations to check UAR compliance on every flop's clock and reset in the design
  - Kills simulation time

How do we get flop-level clock, reset information from design?

Initial DV Approaches & their Limitations

Qualcomm snug

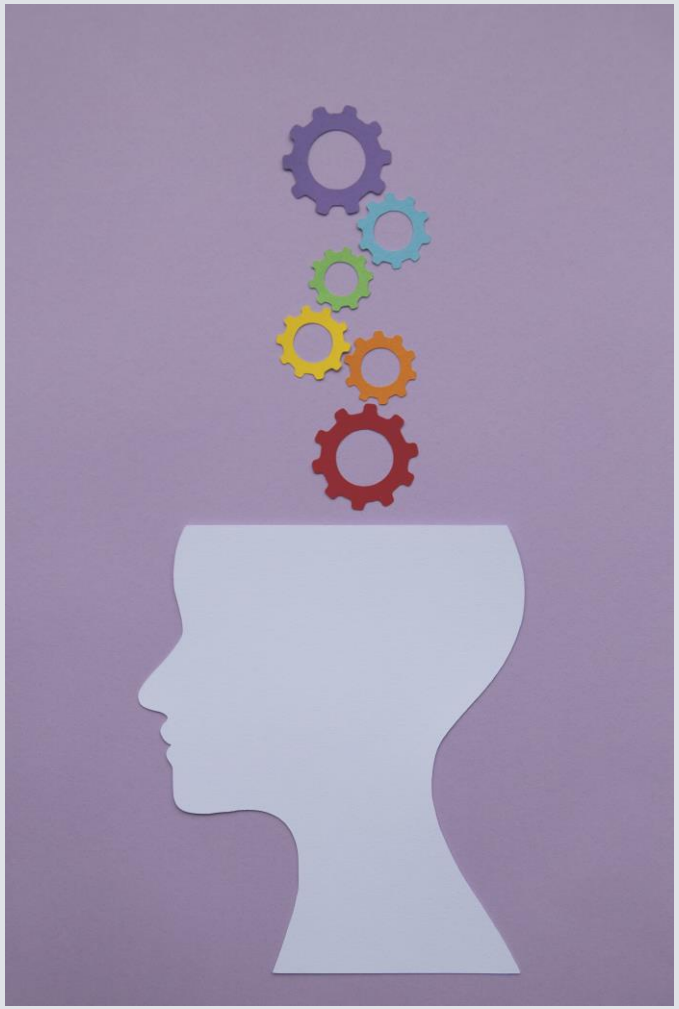
CDC

- **User constraints, waivers**
  Existing CDC checks report violations based on quality of designer constraints.
- **Accuracy**
  Incorrect reset port constraint or improper CDC waiver tags can suppress valid CDC errors.
- **False errors**
  CDC considers the root clock instead of actual CGC output which can result into false UAR violations
- **Failure analysis**
  Once UAR SVA fails, designer needs to manually trace & analyze the load of that clock-reset pair.
- **Incomplete list of pairs**
  We don't have flop level clock-reset pairs extracted.
- **Hierarchical flow**
  Can miss out many clock-reset pairs

Verdi getmodIO utility

- **Name-based clock/reset extraction**
  Ports having "clk/clock" or "rst/ares/reset" in port name
- **Accuracy**
  No guarantee of missing or extracting wrong ports
- **Incomplete set**
  We don't have flop level clock-reset pairs extracted.
- **Failure Analysis**
  Once UAR SVA fails, designer needs to manually trace & analyze the load of that clock-reset pair.
- **False errors**
  Doesn't give clock-reset pair mapping reaching to each flop.

Current static tools can't prove UAR, it must be verified in Dynamic DV simulations with SVA.



Need of the hour

A robust & **comprehensive Flow & solution** is needed for UAR verification

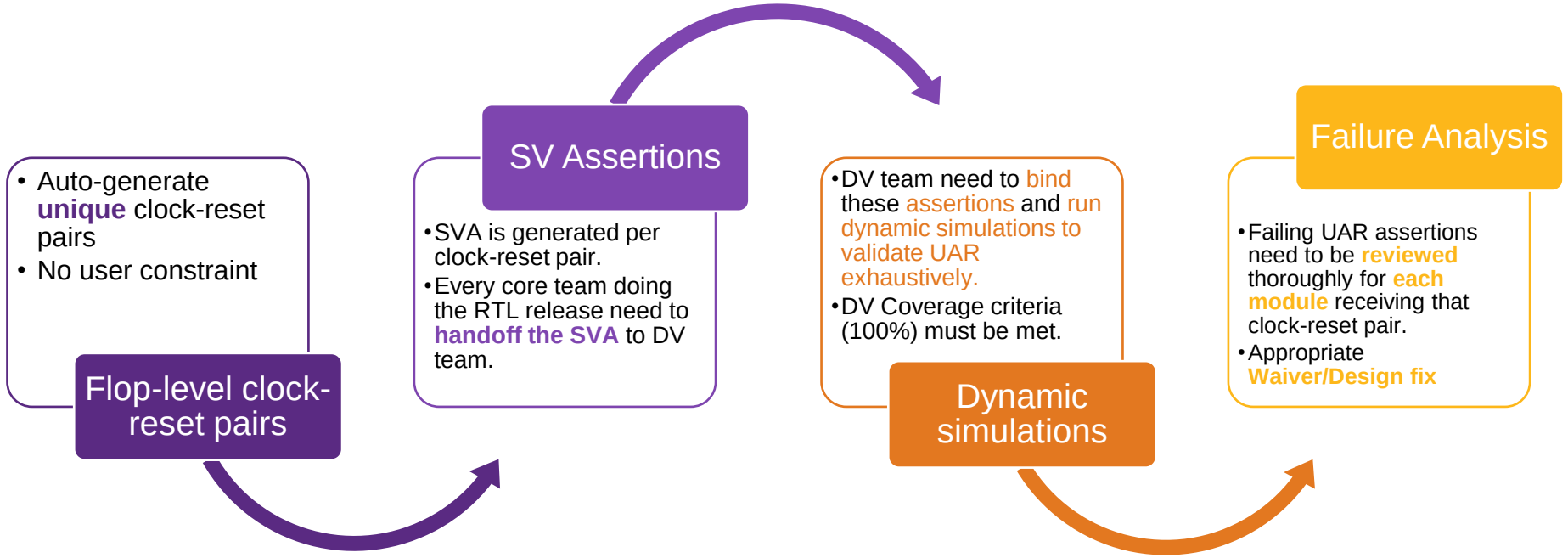
- Millions of flops -> Millions of SVA -> Drastic increase in simulation runtime & convergence issues
  - Need a flow to generate a **unique representative set** of clock-reset pairs for 100% coverage of entire design
- Dependency on **user input constraints** needs to be removed
- Need to **avoid False Errors**.
  - 1000's of False errors lead to Inefficient & error-prone analysis.
- Need to **facilitate failure Impact Analysis.**
  - Reviewer should get complete information of all impacted flops in the design from the tool

Solution Methodology

Solution Methodology

New RDC-based UAR Verification flow

- RDC flow is a static tool to report the crossing between 2 reset domains.
- Idea leverages the tool's capability of knowing Clk-Rst information of every flop in the design.
- We worked with Synopsys team to **enhance the tool** and generate a **representative** (reduced) set of **unique clock-reset pairs** covering whole design.



Solution Methodology (cont…)

Implementation Details & Challenges (Results on a QCT sub-system)

- Tool enhancement to back-trace clock & reset path till it hits **combinational / sequential logic / primary input**
- For e.g.: CGC, Reset-synchronizer, etc.

- Identified **test clocks/resets** and filtered using **DFT constraints**
- Identified **non-resettable/hard-tied reset flops** and dumped in a separate report

- Identified cells where **DFT constraints were not applied properly**
- Identified pairs where **same clock-reset pair** was reported at **multiple level of hierarchy**

Iteration 1 → **Iteration 2** → **Iteration 3** → **Iteration 4**

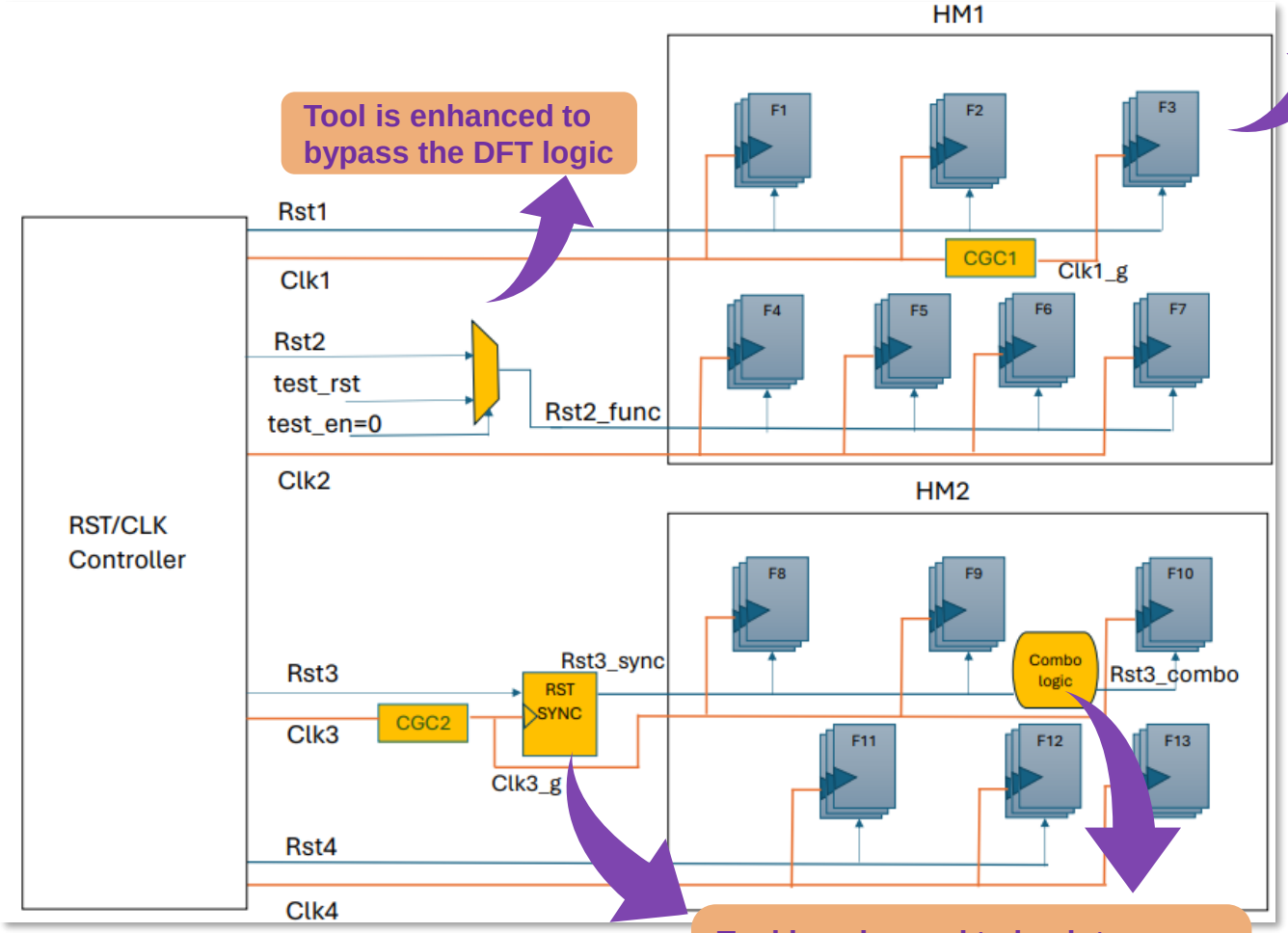
Iteration 1	Iteration 2	Iteration 3	Iteration 4
1 million flops	37k pairs	8k pairs	4k pairs

4 UAR related design violations/ issues caught

Solution Methodology (cont…)

Unique Clk-Rst Pair Extraction

Qualcomm snug



Tool is enhanced to bypass the DFT logic

Tool is enhanced to back-trace clocks & resets till sequential/ combinational logic/primary input

Fig. - Sample

Multiple clock & reset domains

Unique representative set of clock-reset pairs is extracted

CLK source (hierarchy)	RST source (hierarchy)	Destination flop hierarchy	Special key
Top.Clk1	Top.Rst1	Top.F1, Top.F2	
Top.CGC1.Clk1_g	Top.Rst1	Top.F3	
Top.Clk4	NA	Top.F4	RESETLESS FLOP
Top.CGC2.Clk3_g	Top.Rst3_sync. Rst3_combo	Top.F10	
Top.Clk2	Top.Rst2	Top.F4, Top.F5, Top.F6, Top.F7	
Top.CGC2.Clk3_g	Top.Rst3_sync	Top.F8, Top.F9	SYNCRONIZED RESET

100% coverage

Fig. - Some example RDC-based clock-reset pairs

Results

Sample Testchip result

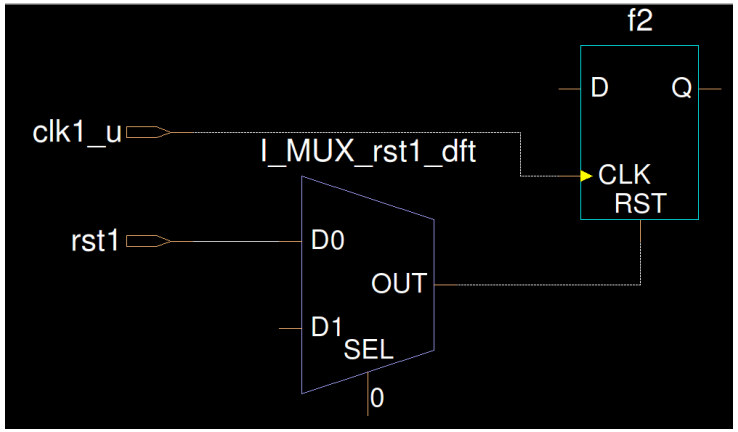


Fig. - example schematic

```
UAR_SVA #(0)  UAR_1 (
                .reset(test.rst1),
                .clock(test.clk1_u)
);
```

Fig. - example generated SVA

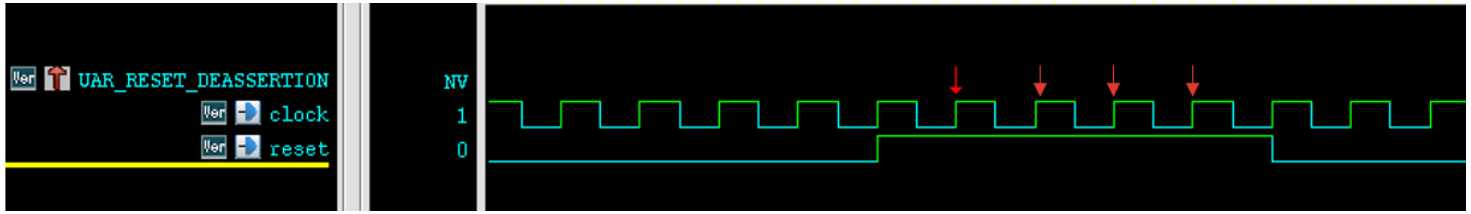


Fig. - example waveform snippet

- SV Assertion is generated for each clock-reset pair dumped by the tool.
- The assertion checks for clock to be at logic LOW in the duration of reset being asserted.
- Among the 4k assertions ran on a QCT sub-system, we found **20 UAR violations**.
  - Out of these 20, 4 were design issues & 16 were TB/Test sequence issues
- The flow is able to catch the post-Si issues found on the previous QCT chipsets.
- Simulation time hits by a mere ~ 5% increase with these assertions.

Merits & Summary

Merits & Summary

0 input constraints

Removes user dependency and generates assertions automatically.

Unique clock-reset pairs

Optimized list of assertions

Flat v/s Hierarchical

Hierarchical CDC can suppress some of the assertions due to no visibility internal to black-boxed modules. RDC-UAR is flat run.

Assured genuine UAR errors

Example run on a QCT subsystem: ***1 million flops*** *reduced to* ***4k clock-reset pairs*** *reported* ***genuine 4 UAR violations.***

Coverage of Millions of flops with limited checks

Due to back-propagation algorithm, millions of clock-reset pairs are reduced to 1000's, ensuring 100% coverage. Hence, reduced runtime & increased efficiency.

Debuggability & Completeness

The tool dumps out the list of flops receiving each clock-reset pair; making it easier for analyzing reported violations.

Future Scope

Future Scope

- Enhance waiver capability feature:
  - Clock-reset pair based waivers
  - Single clock with multiple resets based waivers
  - Single reset with multiple clocks based waivers
  - Module based waivers

- Enhance input constraints:
  - Bypass modules for clock-reset pair extraction

Acknowledgement

Big Thanks!

Our sincere thanks to **Venu Gopal Tata, Vishnu Gumme, Kanika Ghai Bansal and MSS team** from Qualcomm India Private Limited, for their support and encouragement.

We would like to acknowledge **Sanketh & Brijesh** from Synopsys for their continuous support during this work.

Qualcomm snug

THANK YOU

Our Technology, **Your Innovation™**