Exhaustive Reset Verification Enablement: Client PCIE Design Reset Verification Case Study

Rohit Kumar Sinha, Intel India

Praveen Dornala, Intel India

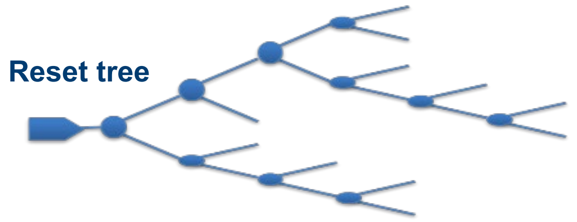
Motivation

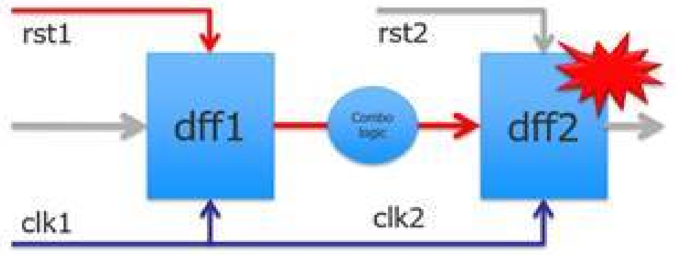
- With the increased complexity of Intel Client SoC design,
  - the hierarchical reset architecture has become very complex
  - increase in reset signaling complexity with the emergence of multiple reset domains
  - create new verification challenges that aren't addressable by RTL simulations.

- Because of the different flavors of IPs
  - independent "reset domains" can be created by complex reset sequences or reset structures
  - metastability and reconvergence issues similar to the failures seen in asynchronous CDC.
  - tangible need to provide automated, exhaustive structural and functional reset signaling checks.

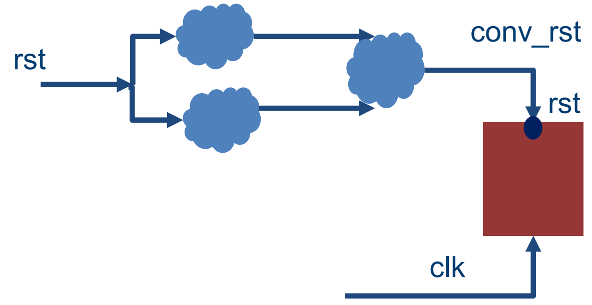
Exhaustive Reset Verification

1. Reset Tree Evaluation
    - Power On Reset, Watchdog Timeout reset
    - Debug reset, Software reset, and Loss of Clock reset.
    - Construction Issues
2. Reset Domain Crossing
    - Application of resets- Sync/Async Behaviour
    - Same reset used as both sync/async in a single module leading to synth vs sim mismatches
    - Combinational Logic in Reset Path
3. Reset Convergence & Reconvergence
    - Reset splitting and converging
    - Reset bus synchronizers reconverging



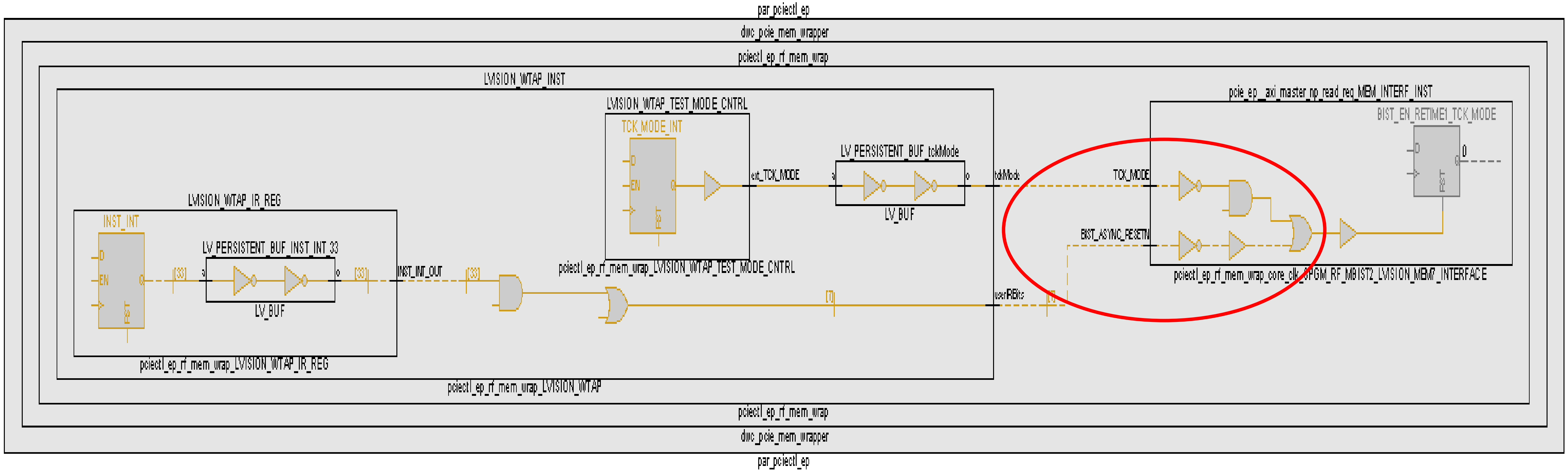


- If rst1 is asserted while rst2 is not asserted, the asynchronous data from dff1 will metastability on dff2



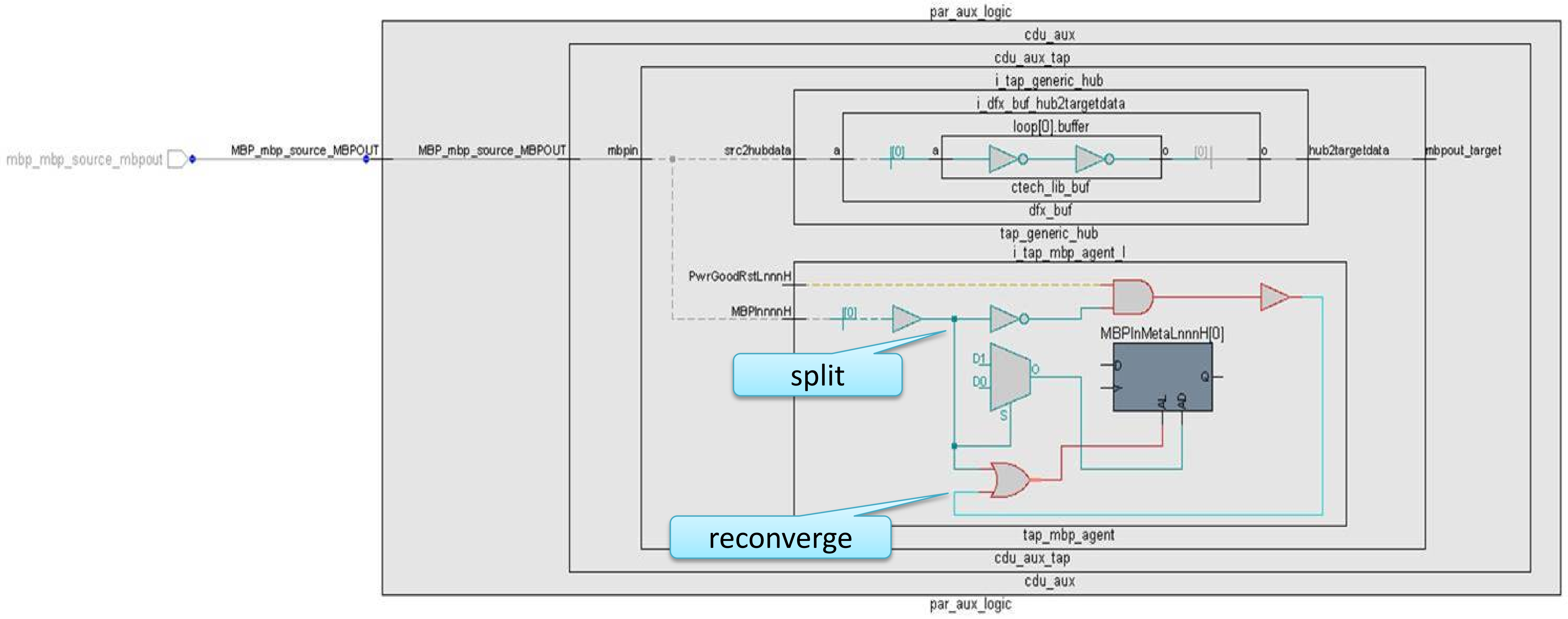
Evidence: Additional Metastability Risk Uncovered

1. RESET combo glitch: This could lead to a glitch in the reset path due to combo logic


par_pciectl_ep
dwc_pcie_mem_wrapper
pciectl_ep_rf_mem_wrap
LVISION_WTAP_INST
LVISION_WTAP_TEST_MODE_CNTRL
TCK_MODE_INT
D
EN
Q
RST
ext_TCK_MODE
LV_PERSISTENT_BUF_tckMode
a
o
LV_BUF
tckMode
TCK_MODE
pcie_ep__axi_master_np_read_req_MBM_INTERF_INST
BIST_EN_RETIME1_TCK_MODE
D
Q
RST
BIST_ASYNC_RESETN
pciectl_ep_rf_mem_wrap_core_clk_PGM_RF_MBIST2_LVISION_MBM7_INTERFACE
LVISION_WTAP_IR_REG
INST_INT
D
EN
Q
RST
LV_PERSISTENT_BUF_INST_INT_33
[33]
a
o
[33]
LV_BUF
INST_INT_OUT
[33]
pciectl_ep_rf_mem_wrap_LVISION_WTAP_TEST_MODE_CNTRL
[7]
userIRBits
[7]
pciectl_ep_rf_mem_wrap_LVISION_WTAP_IR_REG
pciectl_ep_rf_mem_wrap_LVISION_WTAP
pciectl_ep_rf_mem_wrap
dwc_pcie_mem_wrapper
par_pciectl_ep

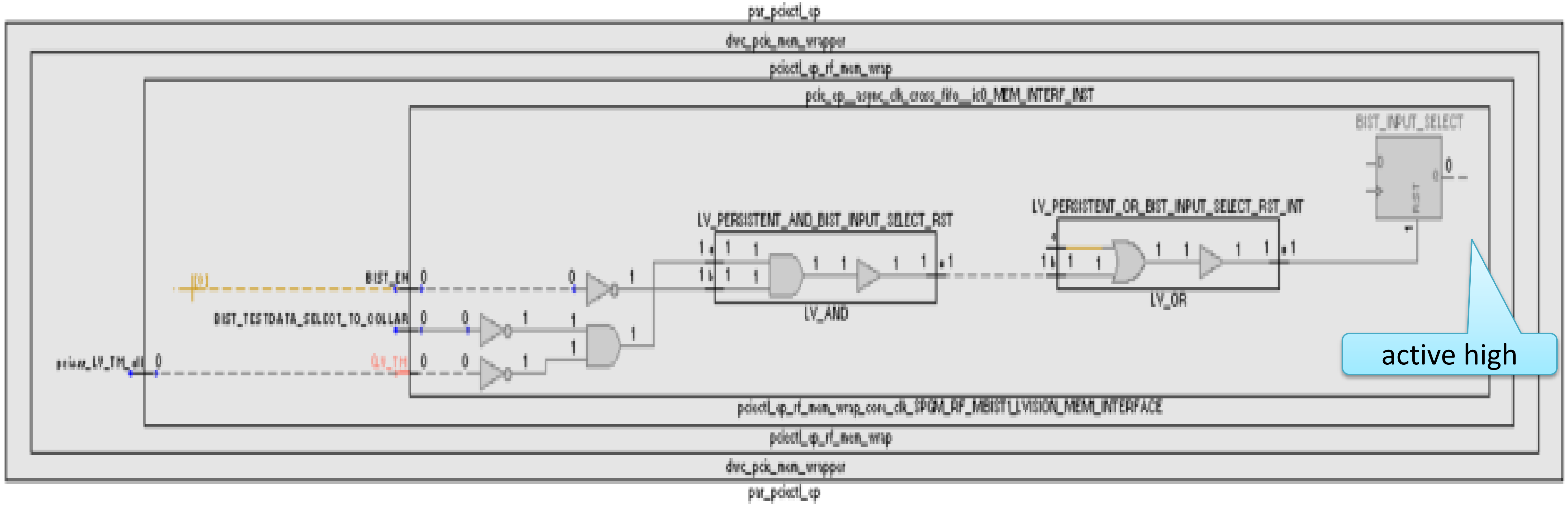
Evidence: Additional Matastability Risk Uncovered

2. RESET convergence Issue appears when reset get split and converge



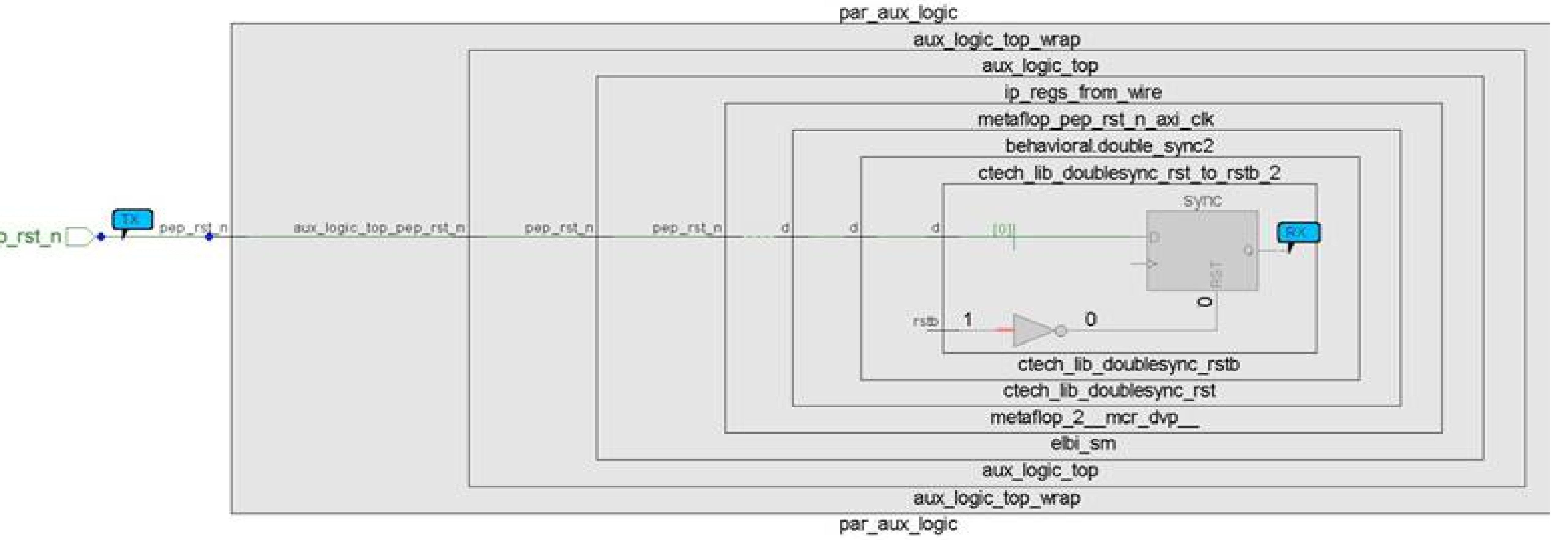
Evidence: Additional Matastability Risk Uncovered

3. RESET is always active HIGH


psr_pcisctl_cp
dwc_pcie_mem_wrapper
pcisctl_cp_rf_mem_wrap
pcie_cp_async_clk_cross_fifo_ic0_MEM_INTERF_INST
BIST_INPUT_SELECT
LV_PERSISTENT_AND_BIST_INPUT_SELECT_RST
LV_PERSISTENT_OR_BIST_INPUT_SELECT_RST_INT
LV_AND
LV_OR
BIST_EN
BIST_TESTDATA_SELECT_TO_COLLAR
pcisw_LV_TM_all
LV_TM
RST
active high
pcisctl_cp_rf_mem_wrap_core_clk_SPGM_RF_MBIST1_LVISION_MEM_INTERFACE
pcisctl_cp_rf_mem_wrap
dwc_pcie_mem_wrapper
psr_pcisctl_cp

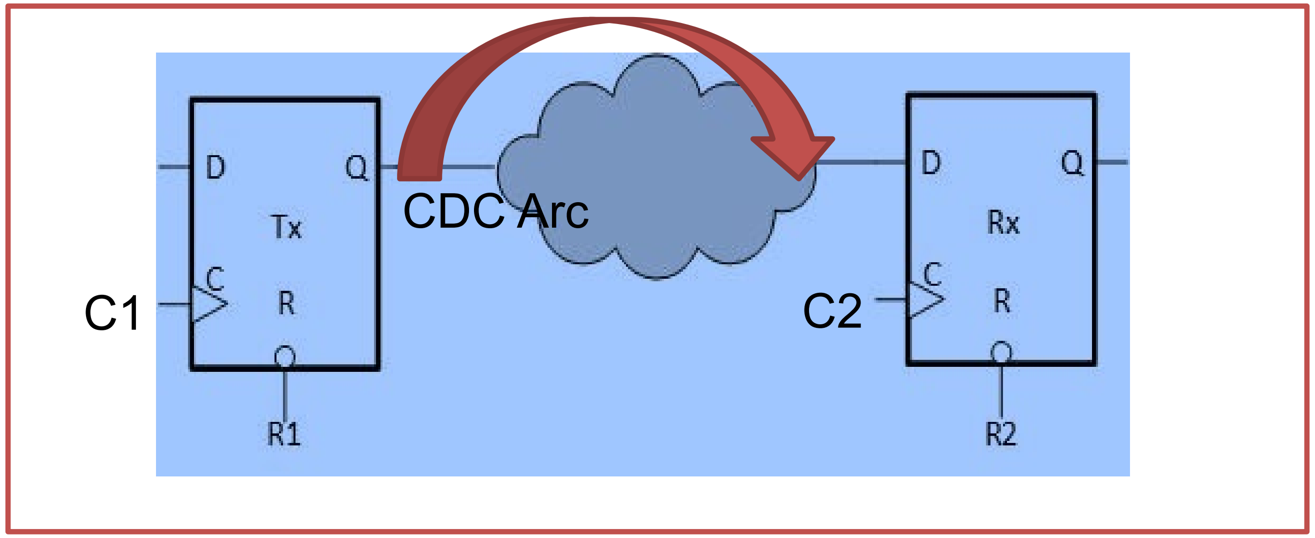
Evidence: Additional Matastability Risk Uncovered

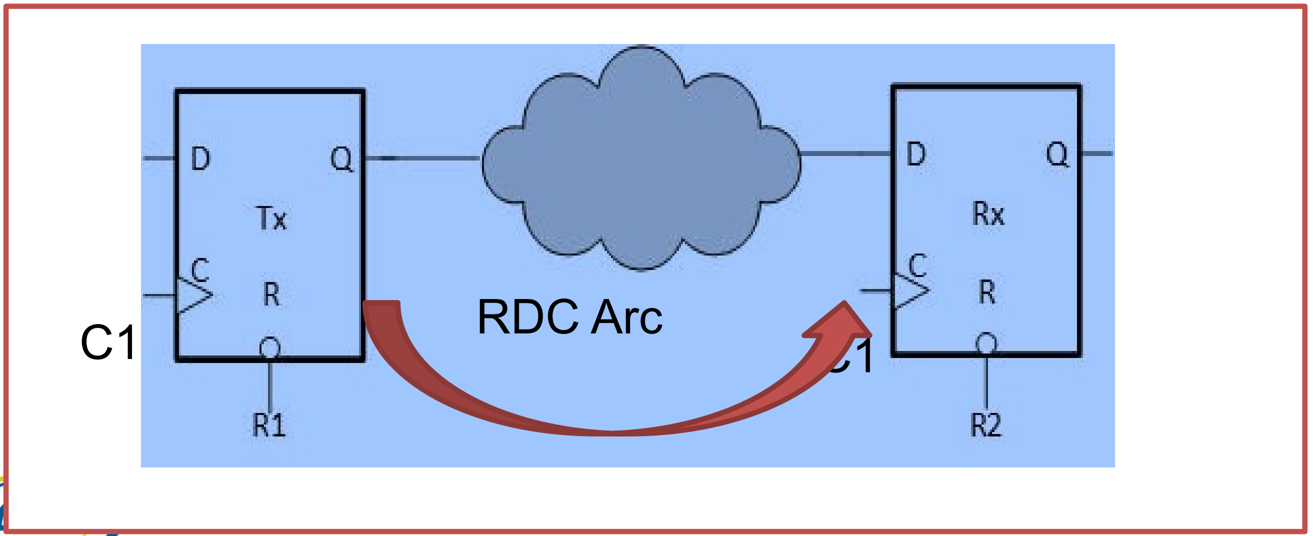
4. Asynchronous wrongly connected to data without synchronizer



Reset Domain Crossing

Introduction using a simple Transmit and Receive Flop



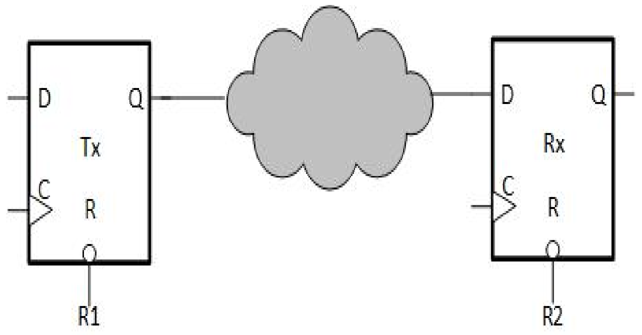


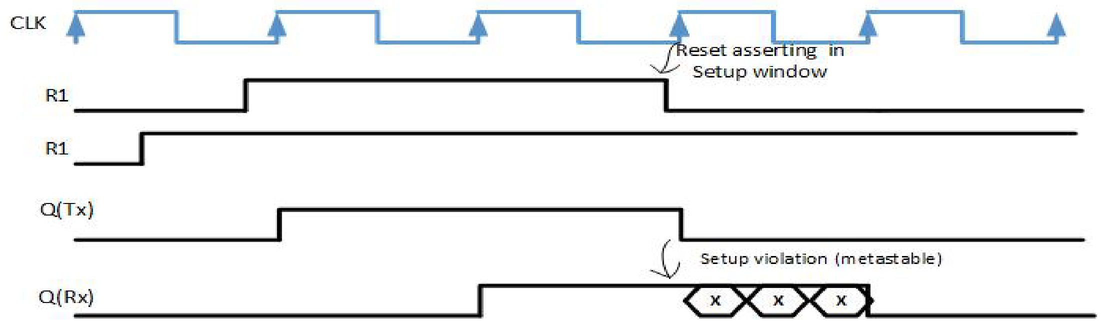
- ✓ Consider 2 flops Tx and Rx on the different and same clock domain.
- ✓ All resets have to assert 0 to clear the flops.
- ✓ Tx flop is on reset (R1)
- ✓ Rx flop is on reset (R2)
- ✓ Reset (R1) asserts when R2 is de-asserted

Reset Domain Crossing

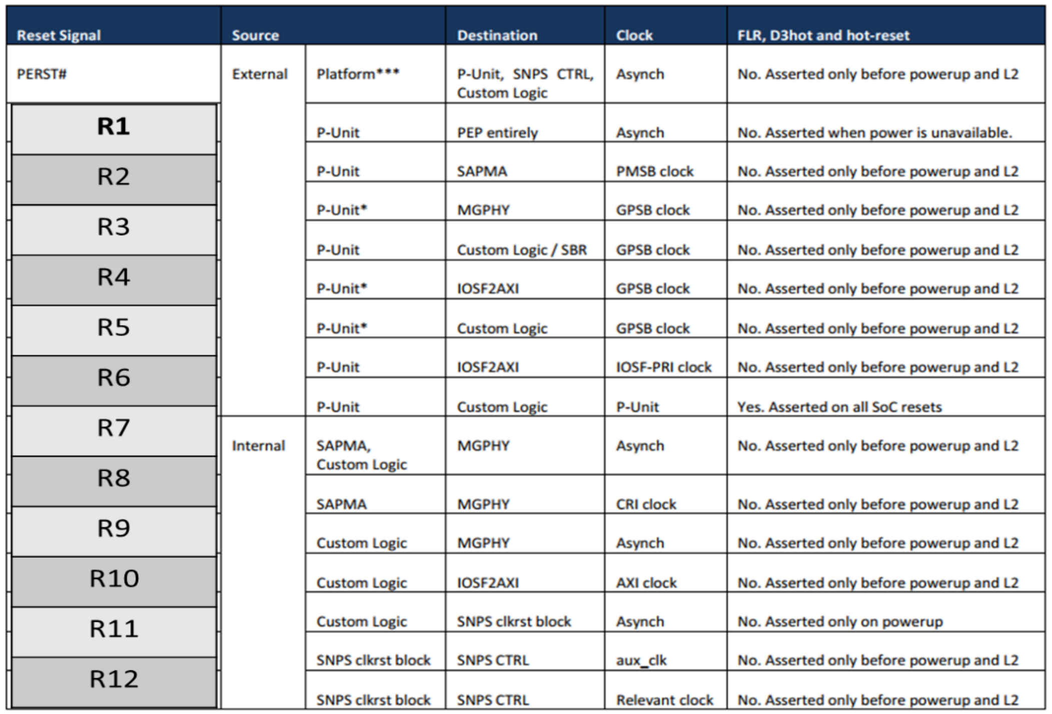
Concept

Reset Domain Crossing is the phenomena when two logic blocks (or flops) are reset using two different asynchronous resets. When we are going into reset, the first reset can assert and make the first block (or first flop) to change asynchronously. This change can make the second block (or the receiver flop) to go meta-stable. This meta-stability caused due to resets is the problem that we are trying to avoid. The cause of this kind of meta-stability is what is known as reset domain crossing





Reset Assertion Ordering

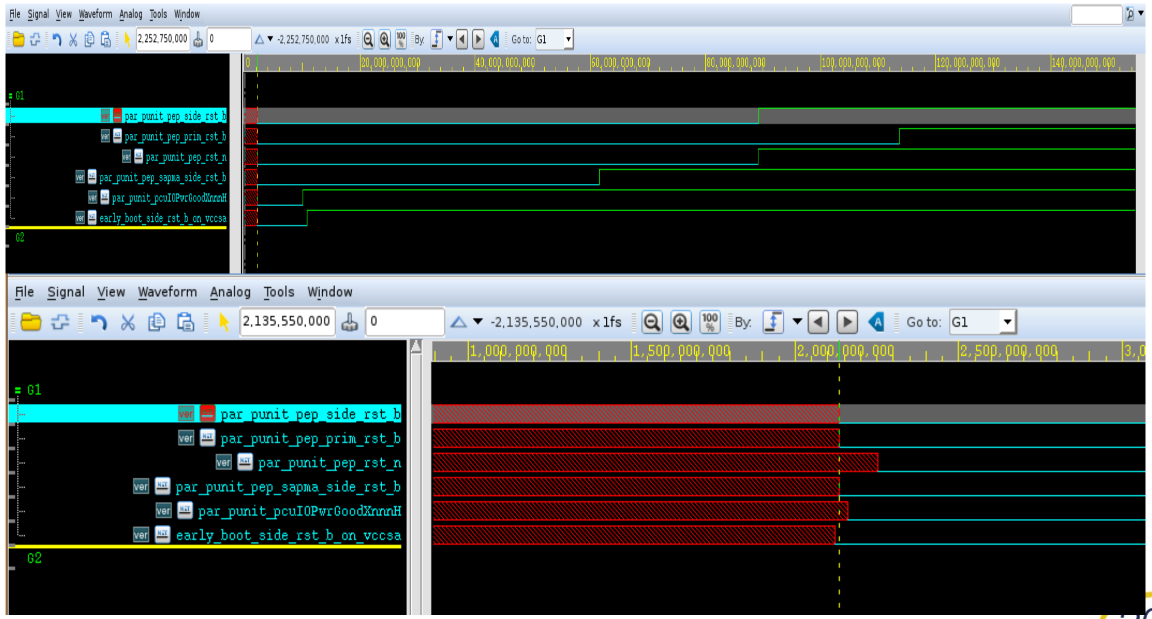


Reset Signal	Source		Destination	Clock	FLR, D3hot and hot-reset
PERST#	External	Platform***	P-Unit, SNPS CTRL, Custom Logic	Asynch	No. Asserted only before powerup and L2
R1		P-Unit	PEP entirely	Asynch	No. Asserted when power is unavailable.
R2		P-Unit	SAPMA	PMSB clock	No. Asserted only before powerup and L2
R3		P-Unit*	MGPHY	GPSB clock	No. Asserted only before powerup and L2
R4		P-Unit	Custom Logic / SBR	GPSB clock	No. Asserted only before powerup and L2
R5		P-Unit*	IOSF2AXI	GPSB clock	No. Asserted only before powerup and L2
R6		P-Unit*	Custom Logic	GPSB clock	No. Asserted only before powerup and L2
R7		P-Unit	IOSF2AXI	IOSF-PRI clock	No. Asserted only before powerup and L2
R8		P-Unit	Custom Logic	P-Unit	Yes. Asserted on all SoC resets
R9	Internal	SAPMA, Custom Logic	MGPHY	Asynch	No. Asserted only before powerup and L2
R10		SAPMA	MGPHY	CRI clock	No. Asserted only before powerup and L2
R11		Custom Logic	MGPHY	Asynch	No. Asserted only before powerup and L2
R12		Custom Logic	IOSF2AXI	AXI clock	No. Asserted only before powerup and L2
		Custom Logic	SNPS clkrst block	Asynch	No. Asserted only on powerup
		SNPS clkrst block	SNPS CTRL	aux_clk	No. Asserted only before powerup and L2
		SNPS clkrst block	SNPS CTRL	Relevant clock	No. Asserted only before powerup and L2

Reset Ordering from High Level Architecture Definition

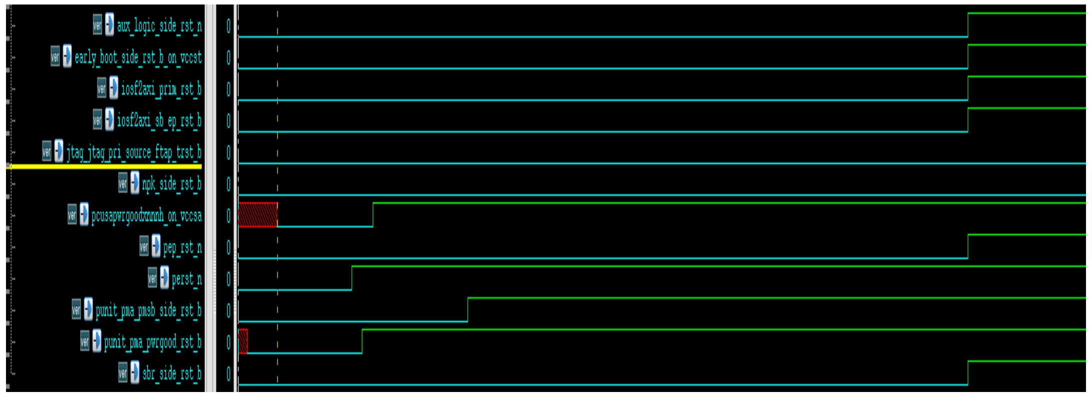
Pciess Resets	Soc Resets
R1	SoC R1_1
R2	SOC R2_1
R3	SOC R3_1
R4	SOC R4_1
R5	SOC R5_1
R6	SOC R6_1
R7	SOC R7_1

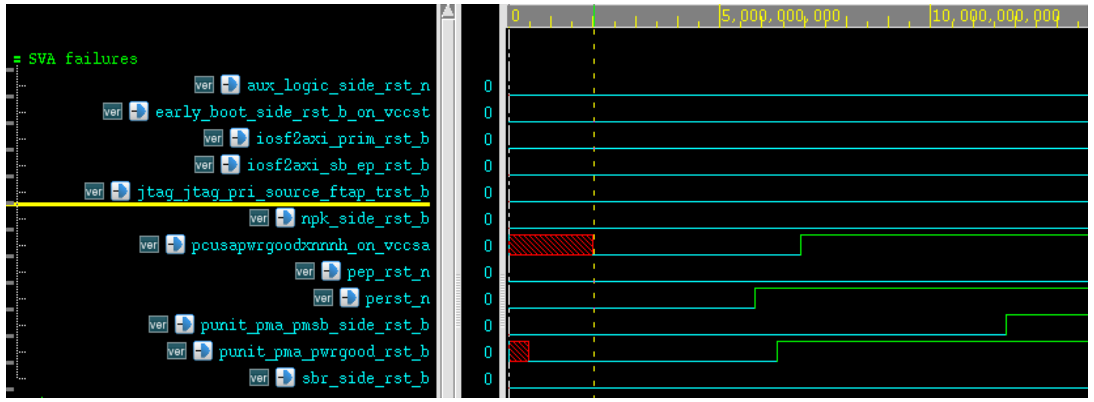
Mapping of Pciess resets with SOC


File Signal View Waveform Analog Tools Window
2,135,550,000
Go to: G1
G1
par_punit_pep_side_rst_b
par_punit_pep_prim_rst_b
par_punit_pep_rst_n
par_punit_pep_sapma_side_rst_b
par_punit_pcuIOPwrGoodXnnnH
early_boot_side_rst_b_on_vccsa
G2

FSDB dump of Pciess resets from PUNIT

Reset Assertion Ordering





Mapping of Pciess resets with SOC

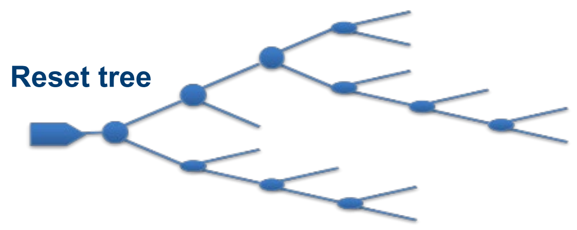
The TX reset punit_pma_pwrgood_rst_b from the waveform is being asserted after the assertion of the all the above RX resets, there by avoid metastability to propagate to the RX Flop.

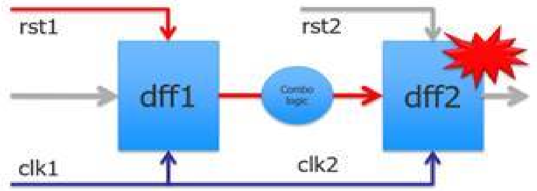
In Reset check tool, the assertion order of these resets can be given by the following directive.

resetcheck order assert -from R1 -to R2

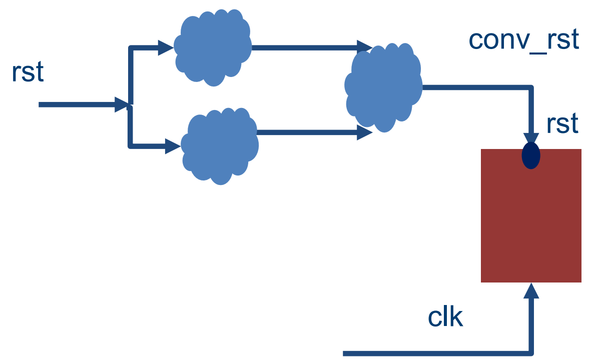
resetcheck order assert -from R3 -to R4

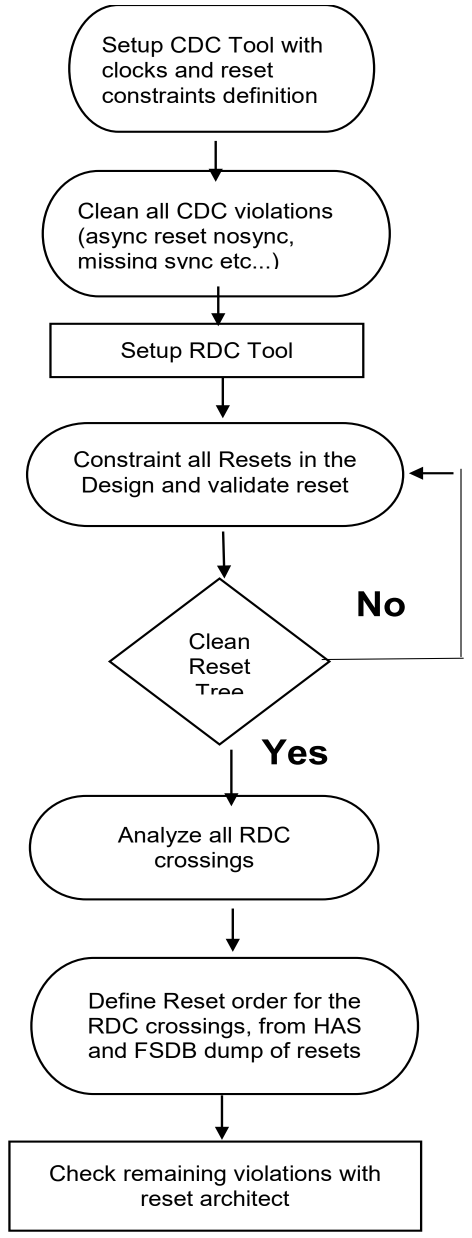
RESET DOMAIN CROSSING





- If rst1 is asserted while rst2 is not asserted, the asynchronous data from dff1 will metastability on dff2





Setup CDC Tool with clocks and reset constraints definition

Clean all CDC violations (async reset nosync, missing sync etc...)

Setup RDC Tool

Constraint all Resets in the Design and validate reset

Clean Reset Tree

No

Yes

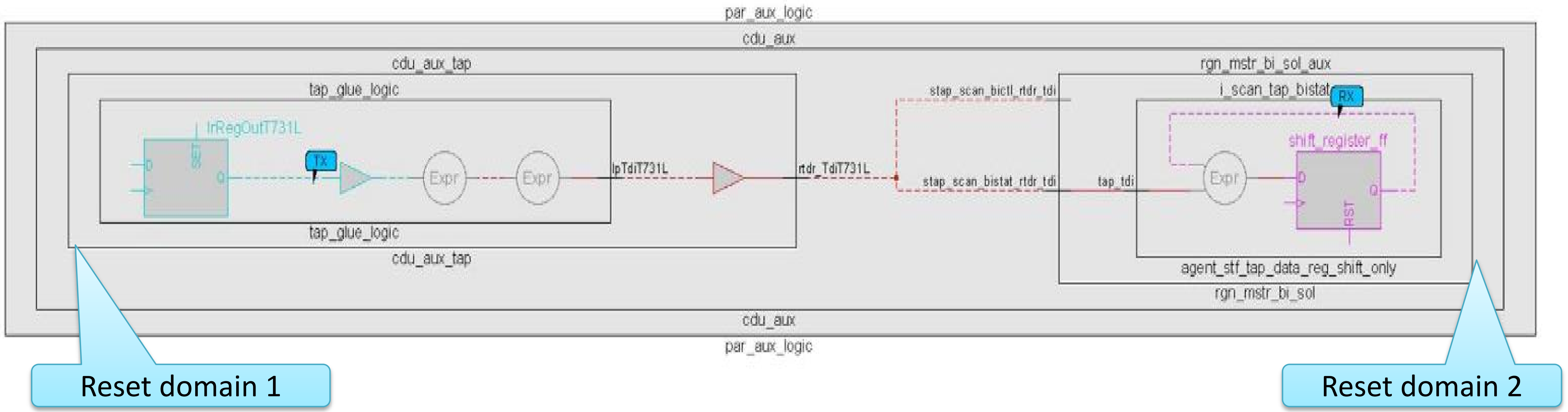
Analyze all RDC crossings

Define Reset order for the RDC crossings, from HAS and FSDB dump of resets

Check remaining violations with reset architect

Evidence: Additional Matastability Risk Uncovered

6. Data crossing reset domains



Summary: Bugs found due to exhaustive reset verification in PCIE Subsystem

- First Implementation of exhaustive reset verification done on PCIE Subsystem for Client SoC
- Additional checks improved the Quality Sign Off
- **Bugs Uncovered after reset verification**
  - Wrong Reset Propagation
  - Power Control logic interpreted as Reset
  - Reset is always active high
  - 2nd Flop with no set/reset pin
  - Reset wrongly used as reset pin
  - Set to Set domain crossing not synchronized
  - Set to Reset domain crossing not synchronized
  - Combinational Logic in reset path
  - Async reset used in sync mode

Partition	#Cell Count	Number of SIPs	Number of HIPs
aux_logic	2577	7	0
Fabric	3108	13	0
PHY	3696	9	5(clock compensator, phy, clock control unit)
Controller	17554	16	18(HIPs)

Questions

Finalize slide set with questions slide