Left Shift Mechanism to Mitigate Gate Level Asynchronous Design Challenges

Rohit Kumar Sinha,
rohit.kumar.sinha@intel.com
Kavya Kotha,
kavya.kotha@intel.com

***Abstract*-In the Intel's IOTG(Internet of Things Group)SoC Design, it is often a challenge to signoff gate level Clock Domain Crossing (CDC) or Reset Domain Crossing(RDC) verification because of the lack of IP/Subsystem hierarchal collaterals, lack of clock-reset architectural understanding in back-end domain, waiver importing issues, strict schedule pressure etc. However, with the advent of complex clock-reset architecture and due to the increase in reset signaling complexity with the emergence of multiple reset domains, gate level clock and reset domain crossing verification becomes an absolute need to ensure glitch free reset assertions during various power states. Therefore, there is a tangible need to define a methodology to ensure all the CDC or RDC issues post low power cell insertions, scan insertion and synthesis optimization are left-shifted in the Front-End CDC or RDC. This paper details about complex gate level CDC or RDC challenges in the IOTG SoC design and proposed few simple techniques in the front-end verification to address issues which are mainly related to implementation flows.**

I. INTRODUCTION

The purpose of this paper is to describe the asynchronous design issues which are mainly related to design implementation flows and address such issues with various front-end design verification techniques in the SoC design. Below are some of the challenges leading to potential metastability issues that appear mainly the gate level design.

- Synthesis changes such as optimization as sequential optimization, merging, duplication etc done with respect to the RTL
- Clock-Reset paths changes during synthesis consist CTECHs(Component TECHnologies)
- Synthesis/PNR(Place and Route) introduces any new glitches in the async paths
- ECO(Engineering change order) changes leading to existing or broken CDC paths
- DFT(Design for Testability) Scan insertion changes during synthesis flows
- Introduction of isolation logic can impact clock and reset paths leading to glitches
- Isolation logic on a synchronized CDC path can result in combinational logic in the data path that should be guarded against glitches.
- Enable path of the ICGs can create a new asynchronous path
- Introduction of isolation logic can create new CDC paths if the isolation enable signal is driven by an asynchronous clock domain.
- Retention Save & Restore logic can have asynchronous signals driving save and restore, leading to additional CDC paths.

II. TECHNIQUES TO ADDRESS GATE LEVEL ASYNC ISSUES

First of all, one needs to understand the motivation on why gate level async issues analysis needs to be performed and what's the additional quality enhancement Gate level CDC analysis brings to the table. When the power logic is not instantiated in a design, the power intent is absent in the RTL and is extracted from the UPF file during synthesis. The late implementation of the power intent information into the gate-level design may delay the start of power verification until after the gate-level representation is available, and the power verification may create a critical path late in the design flow. Most design teams are aware of possible design problems that can be introduced by the power control logic implementation. Clock and CDC errors can occur when the power elements are incorrectly inserted in the clock tree, the reset tree, or the CDC paths. These errors may result in incorrect functionality due to data loss or data corruption. In other cases, these errors may result in intermittent errors that may not be reproduced in simulation and are extremely difficult to debug in silicon. These intermittent errors may appear or disappear when the operating conditions, such as temperature, voltage, or frequency change.

A. Low power cell instrumentation leading to New or Broken CDC or RDC paths

Traditionally, following are the various design architectural techniques to handle low power instrumentation related CDC/RDC issues

- Clock -Assertions Used to validate to both source & destination side are off while switching on the power- This mechanism is a standard technique to ensure the safe asynchronous crossing by the means of design architecture
- Synthesis- Launch and Capture clock for signals going through isolation cells.

As per the design review, we need to ensure the power control signal crossing one domain to another, and it is signed off through manual reviews.

- At netlist level, need to ensure clock are gated during transition
- Verification - Review of default isolation value and value after reset deassertion need to validate between Integration and Verification Team
- DFT – Scan review post scan-insertion flow

As per the standard design paranoia review, all the scan paths are reviewed post scan insertion and potential metastability issues are checked accordingly.

Power aware designs specify power intent through UPF and rely on synthesis to correctly insert the power elements in design. All power elements are available in the netlist at the gate level and allows to perform CDC verification. However, CDC verification can be considered overdue at the gate level stage because there might be new CDC paths introduced in the design, which may lead to multiple iterations to fix the RTL/UPF. The time it takes to fix the RTL and run the verification again can impact the design schedules. Thus, it is highly recommended to verify power aware CDC at RTL level.

Proposed Flow:

As per the new proposed flow, we need to ensure that post Front end CDC and structural Low power signoff, UPF aware CDC is enabled, and all the violations post low power cell instrumentation is reviewed and signed off.

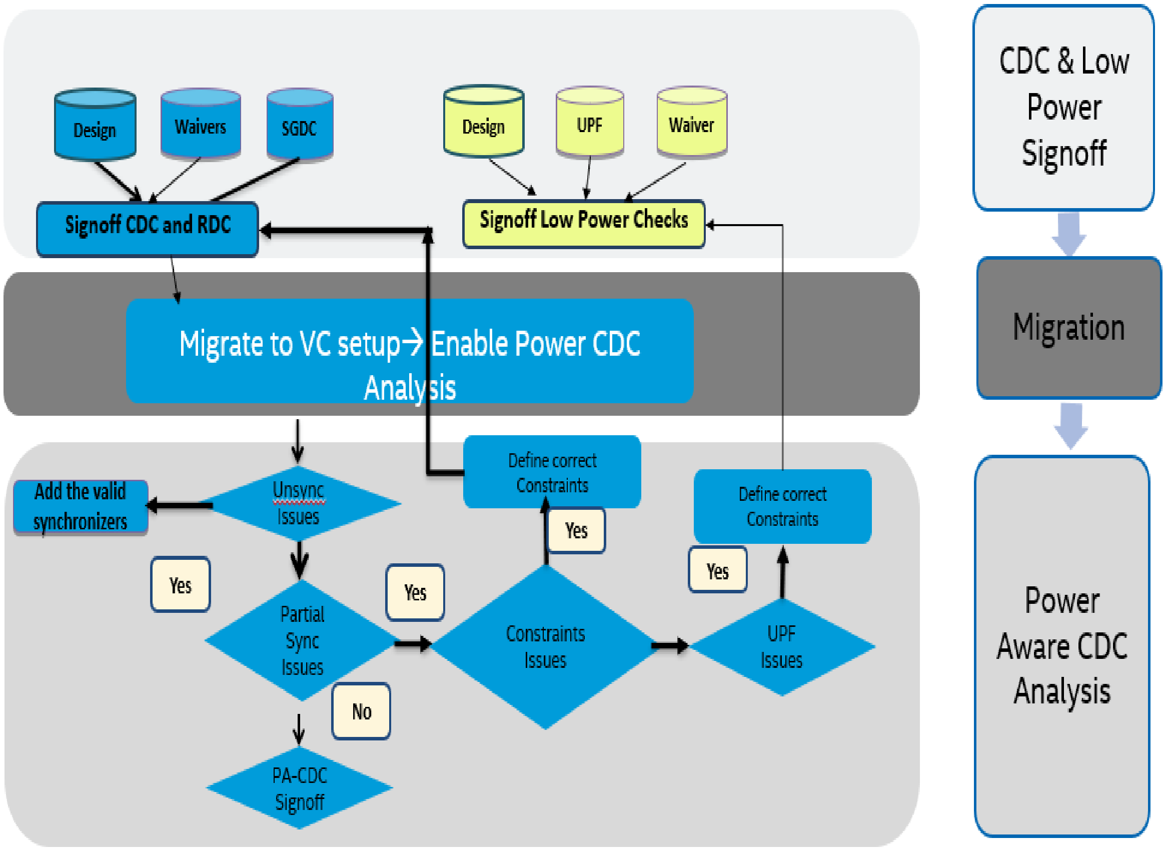


Figure 1. proposed flow to handle low power Instrumentation

B. Glitches Created by Synthesis Optimization

1) Insertion of Clock Gating Logic

Issue related to clock-gating insertions can only be detected at the netlist level CDC verification but it is mainly ensured using the design architecture that asynchronous path from clock enable circuitry is not prone to metastability threats

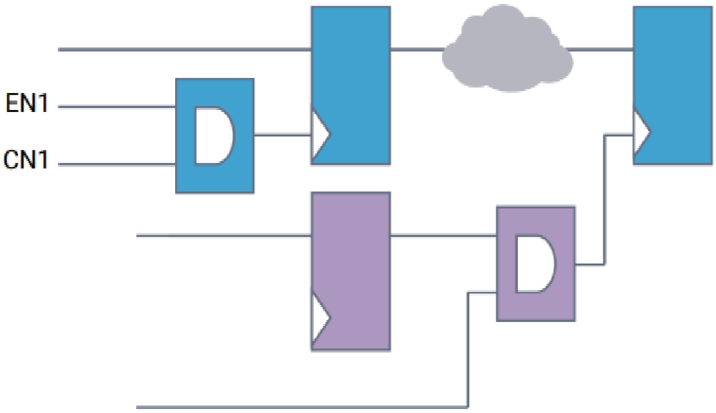


Figure 2 . Clock gating logic Insertion

2) Retiming during synthesis/APR

Issue related to retiming can only be analyzed at the gate-level, however retiming is only done for the synchronous paths and movement of logic across asynchronous paths are not done during the retiming process.

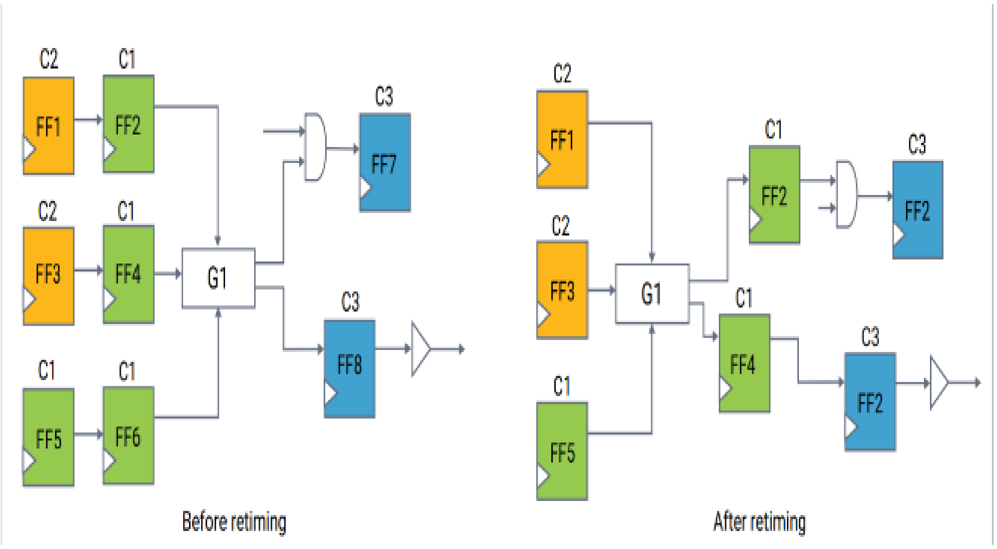


Figure 3. Before and after Retiming

3) Synthesis Optimization of muxes in the async path

Such method is used to identify potential glitches source in the async paths and providing the utility to RTL designer to follow design practices which will prevent the mux decomposition which could potentially lead to glitches at the gate level netlist.

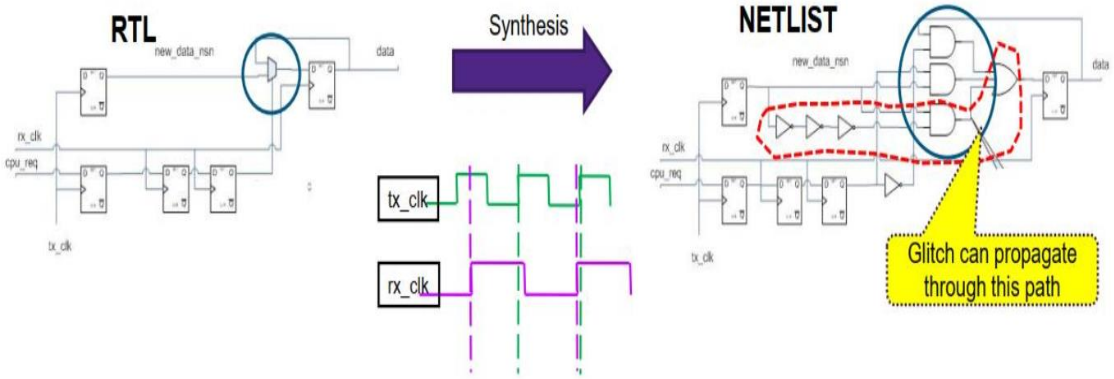


Figure 4. propagation of glitch through CDC path

Vendor tool first reports all the muxes present in the synchronized crossings using these two schemes

a. Recirculation MUX Synchronization Scheme

This scheme marks those clock crossings as synchronized where the first flip-flop in the destination clock domain is driven by a MUX. The clock domain crossing happens through one of the MUX input pins and the other MUX input pin is driven by the destination flip-flop output.

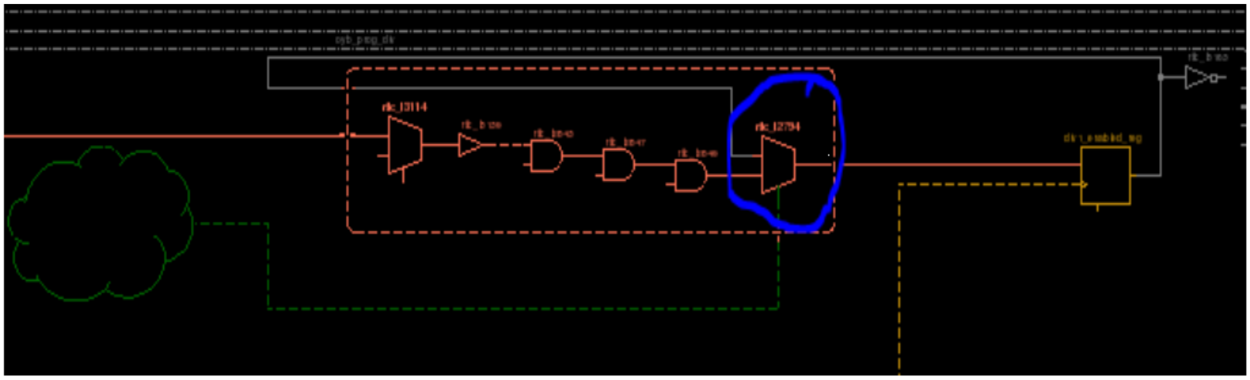


Figure 5 Recirculation MUX Synchronization Scheme

b. MUX-Select Sync (Without Recirculation) Synchronization Scheme

This scheme is the same as the Recirculation MUX Synchronization Scheme except for the MUX need not be a recirculation MUX.

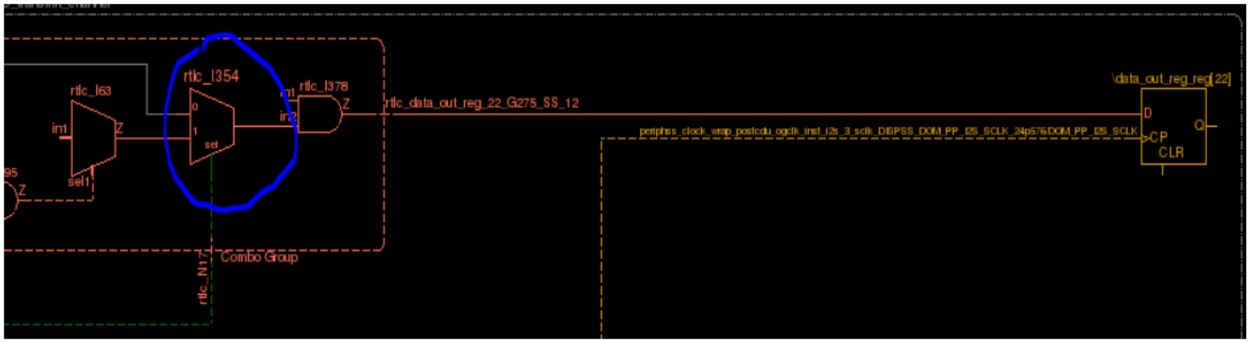

rtlc_I63
in1
Z
sel1
95
Z
rtlc_I354
0
1
sel
rtlc_I378
in1
in2
Z
rtlc_data_out_reg_22_G275_SS_12
periphss_clock_wrap_postcdu_ogclk_inst_i2s_3_sclk_DISPSS_DOM_PP_I2S_SCLK_24p576DOM_PP_I2S_SCLK
\data_out_reg_reg[22]
D
CP
CLR
Q
Combo Group
rtlc_N17

Figure 6. MUX-Select Synchronization Scheme

- Add //infer_mux_override in the RTL of the MUX that were reported by above two schemes

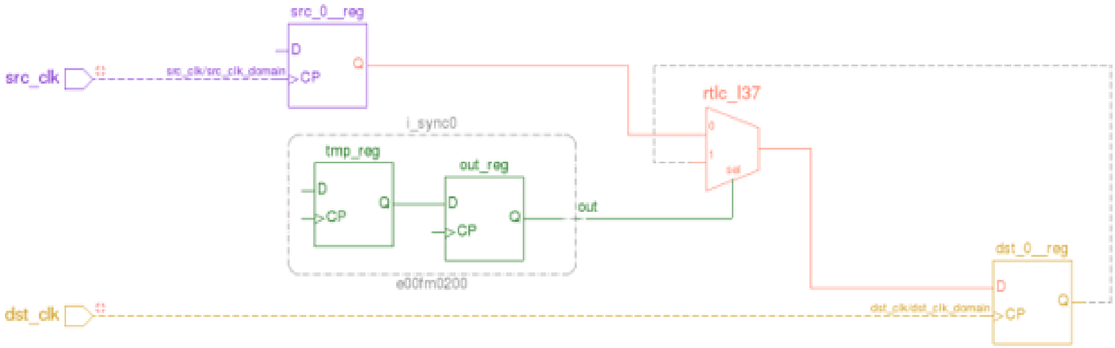


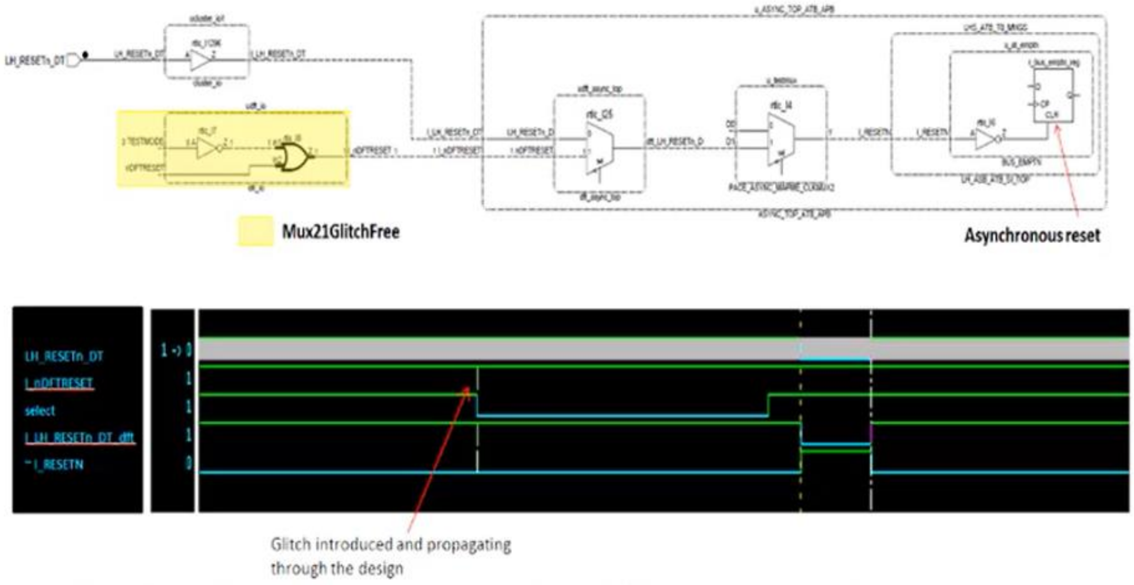
Figure 7. pragma to identify muxes in CDC paths

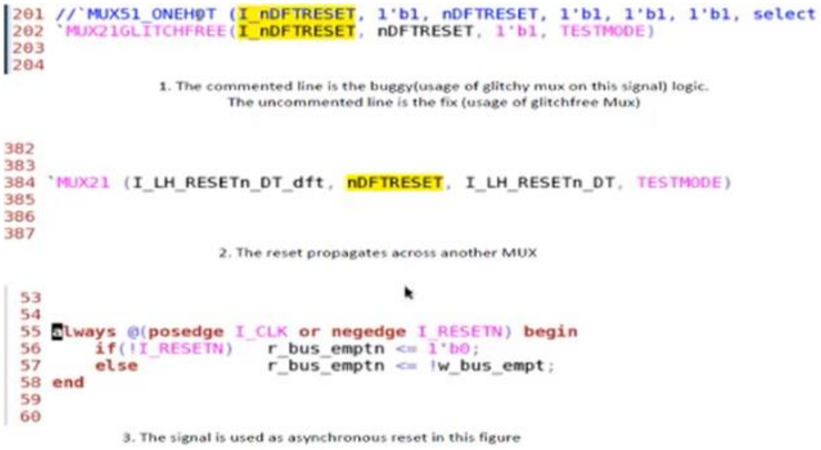
EDA tools will check mux_inferencing RTL code to determine if an pragma //infer_mux_override exists via Ac_cross_analysis rule

- CDC tool will check the mux-inferencing RTL code to determine if an //infer_mux_override pragma exists and if it is the correct place via the Ac_cross_analysis rule
- Seeing this pragma synthesizable tool can in distinguish the CDC muxes
- If designers do not correctly specify "infer_mux_override" on MUXes on CDC paths, DC may infer glitch MUX structure

C. Glitch in Reset Paths

Glitches in the reset paths are mainly introduce due to incorrect modelling in the RTL as shown in Fig8. Sometimes synthesis tool instantiates regular mux instead of the glitch-free mux resulting in the unexpected glitches in the reset path.





```
//`MUX51_ONEH@T (I_nDFTRESET, 1'b1, nDFTRESET, 1'b1, 1'b1, 1'b1, select
`MUX21GLITCHFREE(I_nDFTRESET, nDFTRESET, 1'b1, TESTMODE)
```

1. The commented line is the buggy(usage of glitchy mux on this signal) logic.
The uncommented line is the fix (usage of glitchfree Mux)

```
`MUX21 (I_LH_RESETn_DT_dft, nDFTRESET, I_LH_RESETn_DT, TESTMODE)
```

2. The reset propagates across another MUX

```
always @(posedge I_CLK or negedge I_RESETN) begin
    if(!I_RESETN)    r_bus_emptn <= 1'b0;
    else             r_bus_emptn <= !w_bus_empt;
end
```

3. The signal is used as asynchronous reset in this figure

Figure 8. Glitches in Reset paths

D. Design Practices

Following are some of the design practices which are adopted to ensure that metastability issues are not specifically introduced after synthesis and APR stage and front-end CDC analysis covers all the asynchronous design challenges.

- Avoiding complex synthesis change done with respect to the RTL. This is the standard step that we should not be enabling any complex synthesis optimization to
- Clock-Reset paths only consist of CTECHs
- Synthesis/PNR doesn't introduce any new glitches
- Ensuring ECO changes don't introduce any new glitches
- DFT Full-Chip GLS is done with Test_Enable
- Val Full-Chip Power Aware GLS with zero-delay, SDF-max, SDF min

III. RESULTS

The techniques mentioned in Section II were first implemented in ARM-based SoC design and it is currently in the POR stage. In the hierarchical SoC design, the flow was integrated at p05 milestone itself and all the CDC/RDC related issues that are introduced during the implementation flows are analyzed in the front-end asynchronous analysis using techniques such as power aware CDC, glitch analysis, following standardized design practices or extensive design reviews.

Figure 9 shows the example of a broken clock issue found in the design. A clock was originally gated with latch- based clock gating. When the clock was crossing the power domain, an isolation control signal was added. Due to the isolation control signal, an additional gating signal was created in the clock path when it reaches to the second power domain, and became prone to glitch.

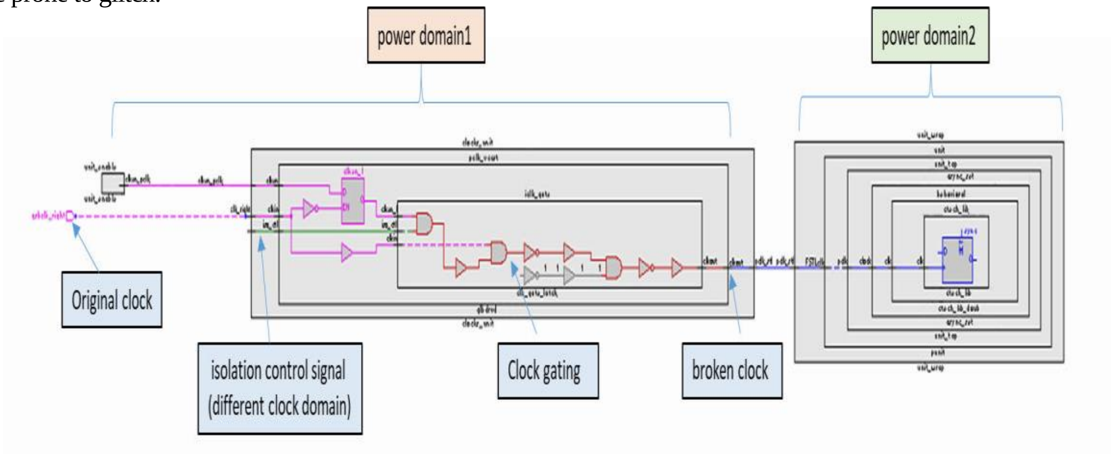


Figure 9. Broken clock path due to isolation control signal

Figure 10 captures an example of new CDC path creation due to the presence of isolation control signal. Here, when the signal was crossing the power domain, an isolation MUX was added to the path. Due to this addition, UPF inserts an isolation MUX that is driven from an asynchronous clock domain. The additional path between asynchronous clock domains is unexpected in the design and can cause meta-stability on the receiver side.

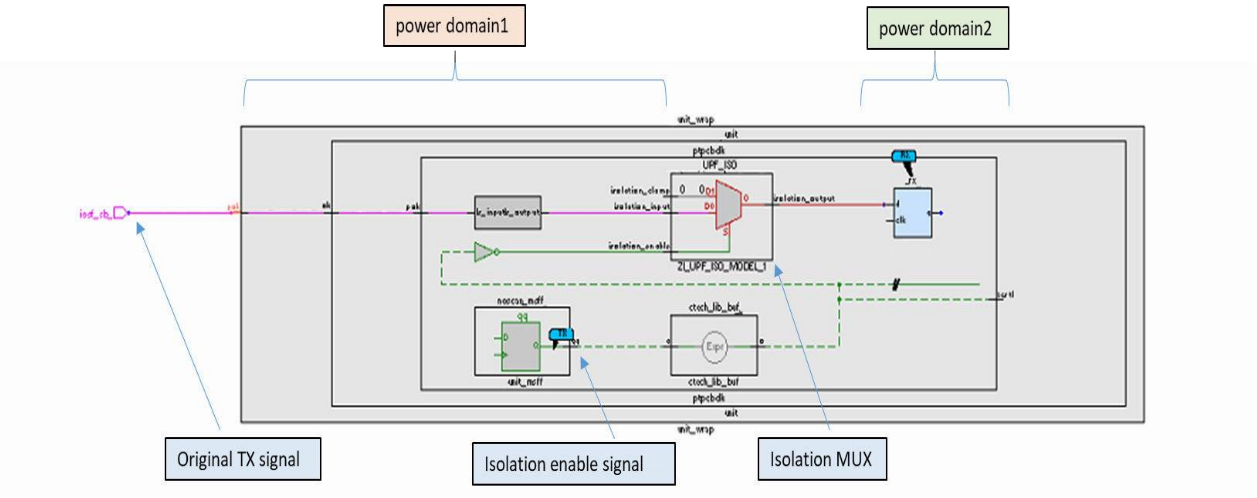


Figure 10. New CDC crossing created due to isolation control signal

Figure 11 shows an example of a broken CDC synchronizer due to the presence of power logic in the CDC path. A 2-DFF synchronizer was introduced in the CDC path between known asynchronous clocks. UPF inserted an isolation MUX between Tx and Rx to the isolation signal crossing power domain. UPF also inserted an unexpected isolation MUX that created a combinational logic which was driven from multiple clock domains. Such combinational logics are undesirable on the CDC paths. Designers must update the UPF to ensure extra isolation from asynchronous clock domain is not inserted in the CDC path.

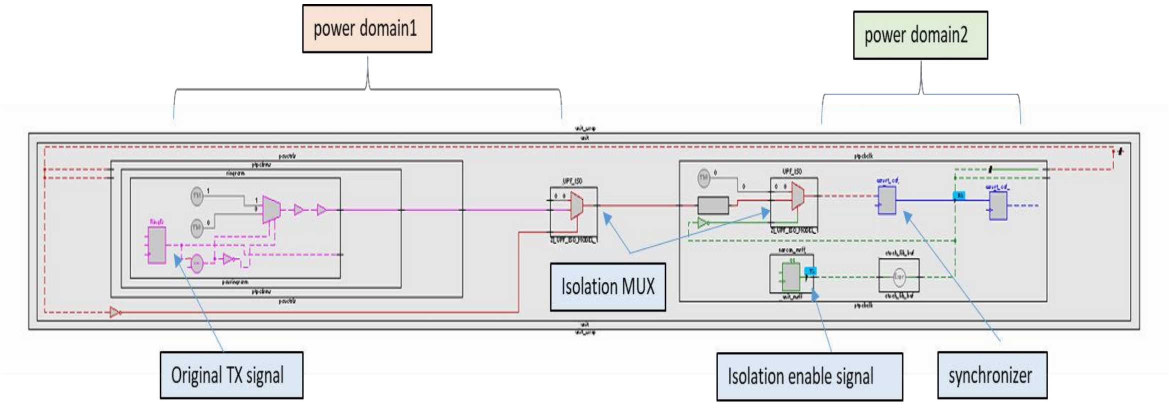


Figure 11. Two-DFF CDC synchronizer broken due to additional combinational logic in the TX-Rx path

In the Fig12, violation results of IOTG subsystem has been mentioned and as per the data, these are the additional UPF aware CDC analysis that have been to be done before CDC signoff is done at the front-end domain

Number of UPF instrumented	**Sync CDC issues missing synchronizer**	**Sync CDC issues with partial synchronization**	**Setup Clock Glitch**	**SYNCCDC_DDATAPATH_FULL**
243	95	76	72	17

IV. SUMMARY

In this paper, we presented several gate level CDC/RDC verification challenges, and a proposed various solution to address the problems. We have also demonstrated the solution through verifying one of our SoC design, and presented the issues discovered. We showed that many of these issues are hard to catch bugs at the front-end domain that cannot be caught easily with simulation or traditional verification techniques. Left shifting gate level challenges and ensuring all aspects of gate level CDC/RDC verification is covered, requires a full suite of methods involving enabling various flows, design practices, RTL change etc. Also, techniques such as power aware CDC or glitch detection at the RTL stage can be scalable to IPs, Subsystems, or all flavor of SoCs.

REFERENCES

[1] Chris Kwok, Priya Viswanathan, Ping Yeung, “Addressing the Challenges of Reset Verification in SoC Designs”, DVCon USA 2015

[2] Questa ResetCheck, HYPERLINK "https://www.mentor.com/products/fv/questa-reset-check" https://www.mentor.com/products/fv/questa-reset-check