Bringing Reset Domains and Power Domains Together – Set/Reset Flops Augmenting Complexities in Power-Aware RDC Verification.

Manish Bhati, Siemens EDA, India, manish_bhati@mentor.com,
Manjunatha Srinivas, Siemens EDA, India, manjunatha_srinivas@mentor.com
Abdul Moyeen, Siemens EDA, India, abdul_moyeen@mentor.com
Inayat Ali, NXP Semiconductors, India, inayat.ali@nxp.com

***Abstract* — Reset designs in complex SOCs are full of challenges. Handling various resets and reset domains is a daunting task. And ignoring the effective interaction between asynchronous resets, clocks, and power domains further adds more complexity leading to intermittent chip failures. Increasing technology (longer battery, high speed, IoT, handheld, etc) and market demands have made power management a priority and the designers have to adopt techniques to manage static/dynamic power consumption. This power management description is instrumented in a design through low power-aware Unified Power Format (UPF) specifications. Any design that incorporates UPF low power elements is prone to erupt and devise reset/clock-related issues. We know that there are two (asynchronous and synchronous) types of resets in hardware. We have several variations among them based on the polarities (active-low and active-high) and based on the signal values (binary 1/0) forced on the flop outputs we categorize them further into Set and Reset. Along with conventional resettable flops, the set-reset flops (also known as Priority flops) are used in digital circuits. In our paper, we have highlighted challenges and analysed design problems that occur due to interaction between power domains and Set/Reset flop in a UPF instrumented design.**

Keywords — Reset Domain Crossing (RDC), Clock Domain Crossing (CDC), Priority flop, Set/Reset (SR) flop, Active Power Management (APM), Power-Aware RDC (PA-RDC), Unified Power Format (UPF), Dynamic Voltage Frequency Scaling (DVFS), Power (UPF) Instrumentation.

I. INTRODUCTION

Designing IC chips with multiple types of resettable flops is an inevitable part of the design process. Having registers with reset function is needed to ensure designs do initialize to a known good state during power-up/booting, avoid unknown and metastability value propagations. Modern high-performance SoCs have multiple subsystems, IPs, peripherals, I/O components, bridge controller blocks, etc that can run at different clock frequencies along with multiple resets such as software resets, watchdog resets, hard resets, power-on resets. Considering the integration of all these blocks from different reset sources we have to ensure that the reset architecture and distribution logic is not affected. Multiple asynchronous resets in a design pose may result in Reset Domain Crossing (RDC) issues when data from one asynchronous source reset domain propagates to either a different asynchronous, synchronous, or no-reset destination domain. While dealing with RDC paths we also need to ensure that the de-assertion of resets does not lead to metastability and that the de-assertion is synchronized in the same clock domain. The reset and RDC issues get augmented when UPF low power elements (such as power domains, power-states, isolation, retention, level shifter cells, etc) are instrumented in the RTL. It is quite natural that the UPF logic added later in the RTL design is likely to disturb the reset architecture and functional behavior which designers are not very much aware of it.

With new generation static verification tools, verification methodologies need to be updated to catch these reset-related issues early in the IC verification cycle. Due to the asynchronous behavior of signal paths crossing different clock/reset/power domains, simulators have difficulty sampling the signal paths. Design and verification engineers have also tried writing assertions to check if the simulators can efficiently sample these signal paths. But the results of verifying such asynchronous paths are not often accurate through simulations. Rare instances of sampling incorrect values will lead to chip failure. This is very hard to predict during the re-spins and ECOs as the root cause of the design sampling incorrect value is difficult to find. Hence need for static design verification tools that employ special algorithms to structurally and functionally identify asynchronous reset problems that account in UPF instrumented designs, is inevitable in the design flow.

Paper [1] presents verification challenges due to power instrumentation in reset paths using normal resettable flops. Complexity increases many folds when designers use Set/Reset flops also known as Priority flops. A Priority flop is a storage element whose asynchronous reset behavior is controlled by two signals generally known as Set and Reset. Designers avoid Set/Reset flops in their design, but sometimes complex scenarios end up getting

synthesized as Priority flops and it becomes inevitable to avoid such design structures. A Priority flop increases the complexity in the reset tree only to add frustration to a verification engineer who is cleaning the design from RDC issues. During power instrumentation, if the RDC path consists of Tx as Priority flop, Rx as Priority flop, or both, the challenge to verify such path is humungous [2].

In this paper we have presented some RDC cases in which the designer might miss some key reset assertion orders involving Priority flops which may lead to few corner case scenarios which are difficult to find out, debug, analyse or handle. These scenarios are hard to predict or catch in simulation because silicon and simulation behavior is not the same when handling Priority flops especially when there are chances of Set and Reset toggling in the same clock cycle. Lightweight formal functional capabilities are required to be built in current generation static verification tools to identify such issues well ahead of time to save on ECO nightmares.

II. STATIC ANALYSIS OF BASIC RDC STRUCTURES WITH AND WITHOUT UPF LOGIC

In this section, we present some of the basic RDC structures that will help understand the complex cases of RDC paths which we have discussed in section III. We will use these basic structures to correlate/interpret how complicated the reset domain groups and RDC paths can become during utilization of Set/Reset flops in UPF instrumented RTL when compared to simple resettable flops.

A. Simple resettable flops RDC structure without UPF instrumentation:

Consider Figure.1 schematic below. RDC structure in a non-UPF RTL with source register controlled by asynchronous active_low reset, tx_rst and destination register controlled by asynchronous active_low reset, rx_rst signal. Different colour codes in the schematics indicate that the resets are in different reset domains. In this case, the RDC problem or metastability at destination flop can occur when tx_rst reset asserts asynchronously and causes the source/launch flop output to change its value within the setup-hold window of clock clk while destination/capture flop is in a non-reset state.

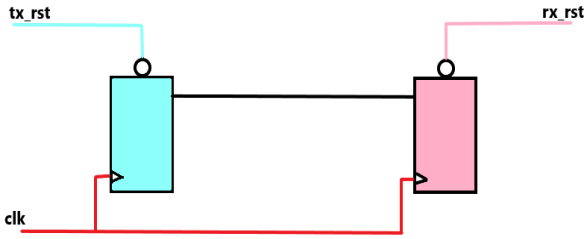


Figure 1. Simple resettable flops RDC structure in non-UPF RTL

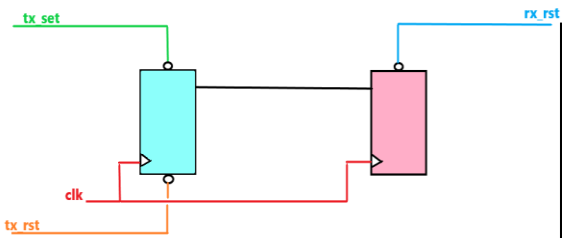
B. Set/Reset (SR) flop RDC structure without UPF instrumentation:

Consider Figure.2 schematic below. RDC structure with SR flop in a non-UPF RTL with source register controlled by two signals

a.) tx_rst (being treated as priority reset based on if-else condition in always block) asynchronous reset

b.) tx_set asynchronous set signal.

The destination register is controlled by the rx_rst asynchronous reset signal. In this case, based on the relationship between the reset pairs tx_set/tx_rst, the RDC problem at destination flop can be determined. An **important factor** to note here is the number of resets has increased when compared to the simple resettable flop structure (as seen in Figure.1) which increases the probability of RDC problems and complexity of the designer's analysis effort [2].




```
always @(posedge clk or negedge tx_rst or negedge tx_set)
  if (~tx_rst)
    tx_reg<=1'b0;
  else if (~tx_set)
    tx_reg<=1'b1;
  else
    tx_reg<=in;
```

Figure 2. Set/Reset flop RDC structure in non-UPF RTL and RTL code snippet for the same

C. Set/Reset (SR) flop RDC structure with UPF instrumentation:

Consider Figure.3 schematic below. In the below scenario RDC structure further becomes complex when RTL is instrumented with low power elements. The source flop is controlled by asynchronous Set/Reset signals (tx_rst & tx_set) which are present in the source power domain (PD_src). The destination flop is controlled by asynchronous reset rx_rst which is present in destination power domain (PD_dest). The low power element added to the RTL is controlled by power control logic which consists of signals driven by a different asynchronous reset domain rst_pwr. In this RDC path, we are now dealing with multiple resets in four different asynchronous reset domains (tx_set, tx_rst, rx_rst, rst_pwr). Now we are not just looking at the four different reset domains but also we need to consider the impact of two different power domains which brings in the necessity of concurrent multi-domain verification. Verifying such RDC paths can be painful for designers when dealing with huge SoCs/IPs where reset sequence and relationship between these multiple resets in multiple power domains should be accounted for.

Please note the below Figure.3 schematic depicts just a simple straightforward scenario. Similarly, we may have many such cases with corner situations in the design. In this paper, we have studied and explored the potential RDC problems with SR flops in power-aware RTL as detailed in section III.

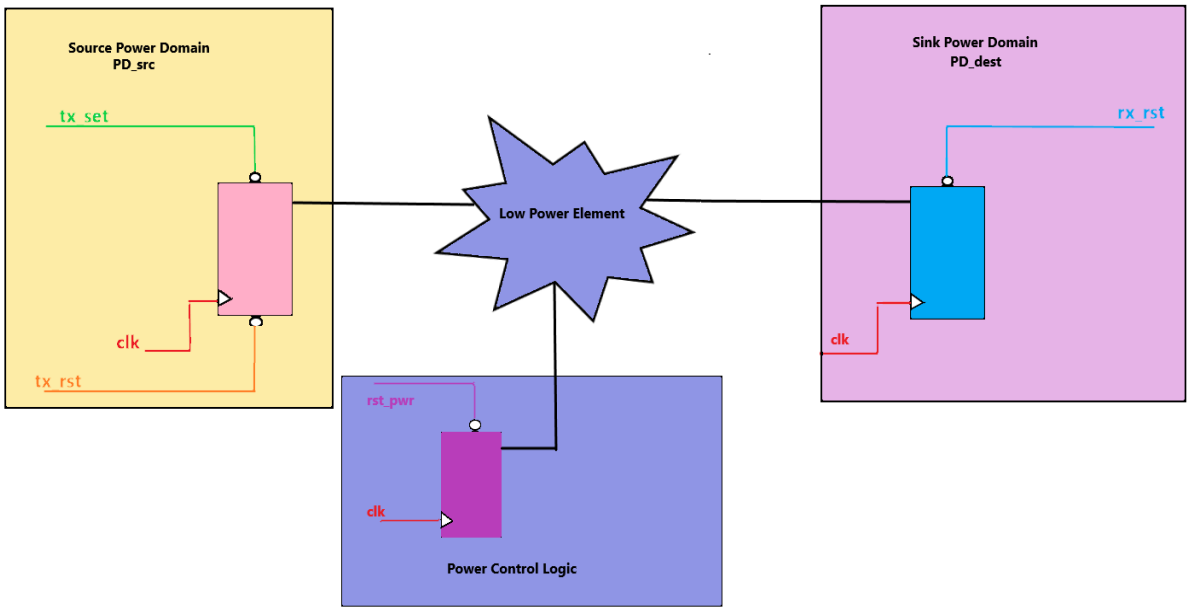


Figure 3. Set/Reset flop RDC structure in UPF RTL

III. PRIORITY FLOP STRUCTURES LEADING TO RDC ISSUES IN UPF DESIGNS

To overcome power consumption problems designers build power architectures adopting various techniques (such as partitioning blocks into power domains, multi-voltage grouping, clock-gating, power-gating, DVFS, etc). To achieve this the original RTL design is reinforced with extra power control logic structures that consist of isolation logic, retention logic, level shifters, power domains, power states, supply-sets, etc. Below are some of the Priority Set/Reset flop structure cases closely studied that can lead to new RDC problems or disrupt existing paths when low power elements are instrumented in the design.

A. Isolation Logic

Case 1 – *RDC path formed due to insertion of isolation cell along the data path*

Consider schematic of Figure. 4. The color-coding in the schematic indicates different power domains for the block instances in the design. U_f1 is a priority flop. Here the notations “AL” and “AD” indicated for the pins in the u_f1 register is the synthesis output structure for the SR priority flop RTL code (as seen in Figure.2) which means “Asynchronous Load” and “Asynchronous Data”. Pins “AL” and “AD” are active based on the values of rst and set. Flop will either be in Reset or Set state which will be determined when “AL” pin will be active. But flop will only be in Set state when “AD” is also active along with “AL”. When “AL” is not active SR flop is in functional mode.

Without insertion of isolation cell, RDC path crossing exists between source u_f1 SR (rst1 – active_low asynchronous reset; set1 – active_low asynchronous set) register and u_f2 simple resettable (rst2 active_low asynchronous reset) register. Asynchronous assertion of either rst1 or set1 at u_f1 can cause metastability at the flop u_f2 instance when rst2 is released (i.e. u_f2 is in functional state). With the insertion of an isolation cell along the data path between u_f1 and u_f2, we have a new reset domain rst3 (active_low asynchronous reset signal) driven through register u_f3.q driving the isolation enable signal. This will result in a new RDC path during the data transfer mode (both source u_f1 and destination u_f2 power domains are ON) or in isolation mode (source u_f1 is OFF and destination u_f2 is ON) when rst3 reset asserts asynchronously while rst2 is in the de-assertion state.

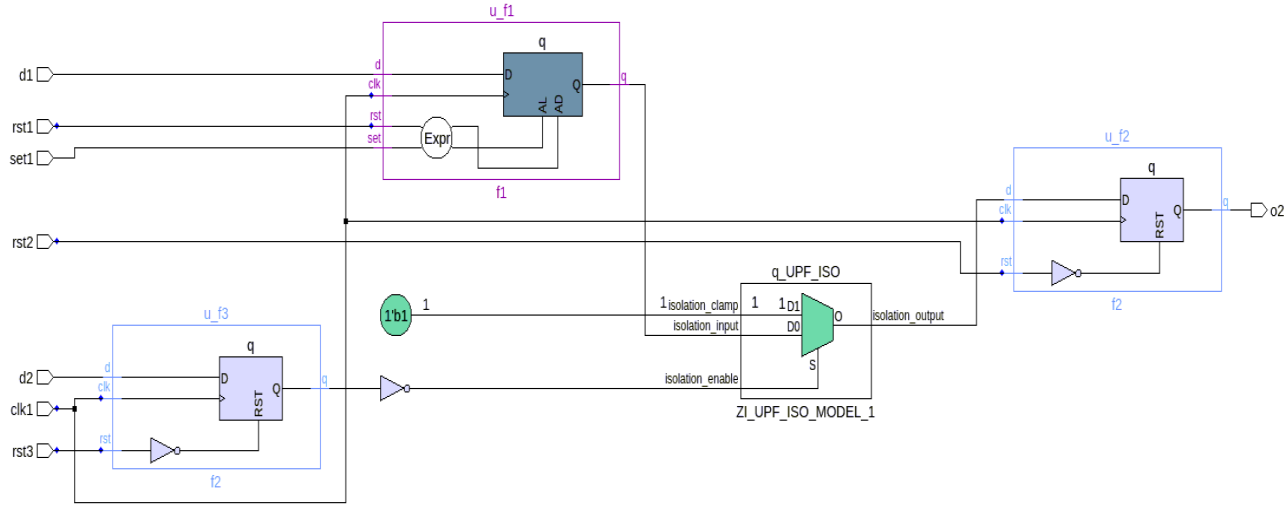


Figure 4. Isolation enables in different reset domains inserted along the data path

Case 2 – *Insertion of isolation cell creating an inference of multiple-reset domains*

Consider Figure.5 below the schematic. Before UPF instrumentation of isolation cell along the reset path, the RDC crossing exists between flops u_f3 (rst2 asynchronous reset domain) and u_f1 (driven by SR flop f2 with set1 and rst1 asynchronous reset domains). If the assertion orders are correctly specified between rst1/set1 and rst2, then is crossing is guaranteed to be a safe RDC crossing. But with the insertion of isolation cell along the reset path for u_f1 flop the tool infers multiple-reset domains (rst1, set1, rst3). This makes crossing unsafe as now there is an unsafe path between u_f3 (rst2 reset domain) and u_f1 (driven by multiple-reset domains) flop resulting in disruption of the reset assertion order specified earlier (between rst1/set1 and rst2 pairs). Also, during initialization of the u_f2 flop i.e. when rst3 asserts, the clamp value 1 deactivates the u_f1.rst reset and chances of u_f1 flop suffering metastability are high when rst2 reset asserts.

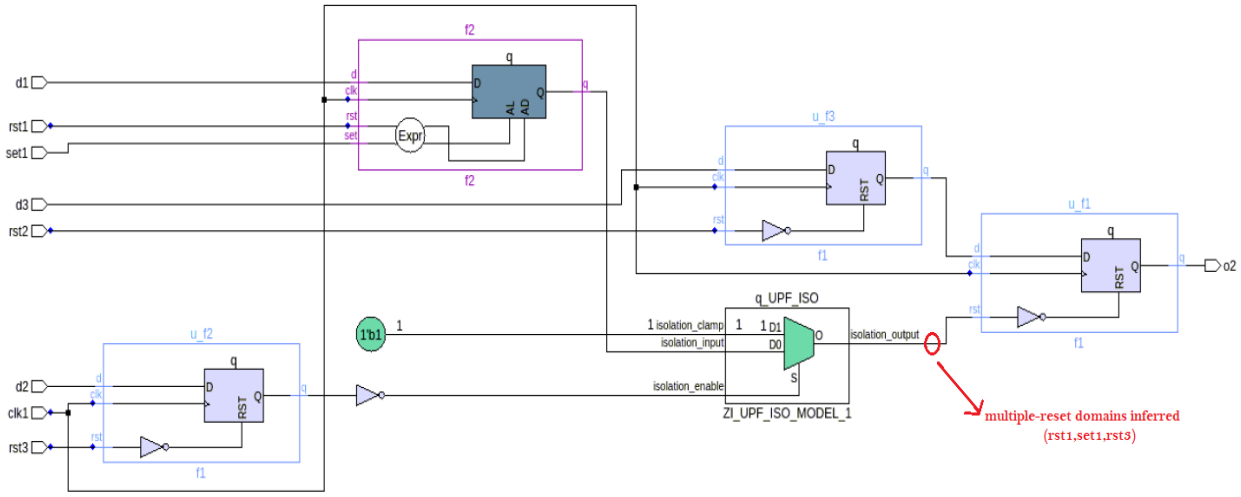


Figure 5. Isolation enables in different reset domain inserted along reset path

Case3 – *Isolation cell disrupts RDC paths controlled by data isolation strategy*

Data isolation strategy is a special technique to control or synchronize the safe data transfer between two reset domains. It defines that the isolation signal pre-emptively isolates the Rx domain logic by anticipating the Tx asynchronous reset activation and pre-emptively deactivates the Rx register, so the receiver logic "freezes" at the current values until the Tx output changes and the isolation signal returns to its original value.

Consider schematic of Figure.6. Without UPF isolation cell the RDC crossing between f2 (set1, rst1 asynchronous domains) SR flop and u_wrap_rx.u_f2 (rst3 asynchronous reset domain) flop is controlled by data isolation signal "data_ctrl". The "data_ctrl" signal isolates the output of f2 flop when either of its resets (set1, rst1) asserts thus avoiding asynchronous data change from f2.q causing metastability at Rx flop u_wrap_rx.u_f2. There is some kind of handshake protocol between the data_ctrl and the set1/rst1 reset signals. Hence these RDC paths are considered safe during non-UPF RTL analysis. With the insertion of the isolation cell, the isolation enable is driven by the new reset rst2 domain adds a new RDC path in the UPF RTL. There is no handshake protocol defined between the data_ctrl and the newly introduced rst2 reset signal. The Rx register u_wrap_rx.u_f2 can thus suffer metastability when rst2 reset of u_f1 flop asserts asynchronously disrupting the safe RDC paths which were controlled by data isolation strategy.

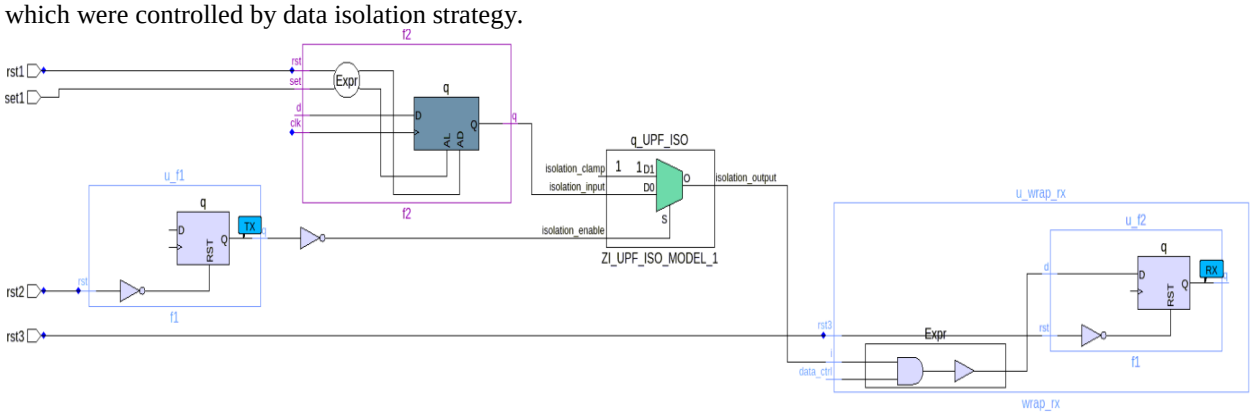


Figure 6. Isolation cell disrupts data isolation strategy

Case 4 – *Isolation cell disrupts reset assertion order sequence specified during non-UPF RTL analysis.*
Consider schematic of Figure.7. Without UPF instrumentation the crossing between Tx, u_f3 flop (rst2 domain) and Rx, f2 flop instance (rst1, set1 domains) is safe because the reset assertion order sequence defined between rst1 and rst2 was enough to avoid metastability propagation at Rx, f2 flop. Consider a scenario where the source power domain of the u_f1 instance is turned off in UPF instrumented design. This will result in clamp value 1'b1 disabling the reset rst1 of f2 SR flop. This breaks the existing reset ordering and thus creates a need to define a new set of reset assertion order sequences for reset groups rst2 and set1 to avoid metastability propagation at f2 Rx flop when rst2 asserts asynchronously.

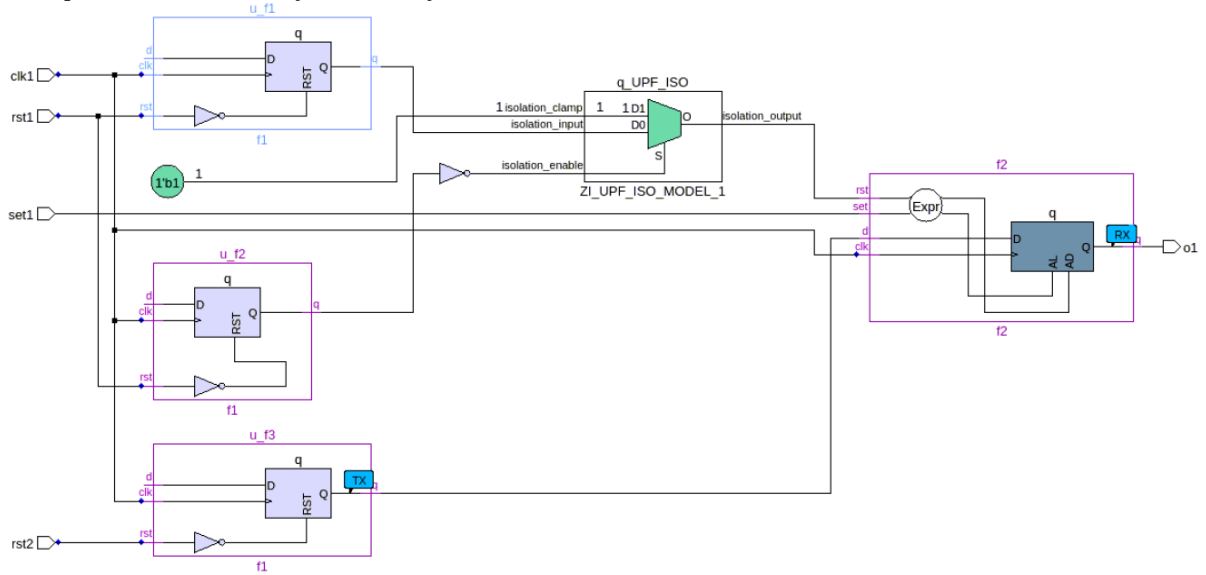


Figure 7. Isolation cell disrupts reset assertion order sequence

Case 5 – *Isolation enable signal driven by internally generated resets affecting non-RDC paths.*

In Figure.8 schematic, without UPF instrumentation it looks like there is no RDC crossing between Tx, u_f2 (set1, rst1 reset domains) and Rx, u_f4 (set1, rst1 reset domains). Even with UPF instrumentation of isolation cell, it doesn't seem that it will affect the RDC path as the isolation enable signal u_f5.q seems to be driven by rst1 reset domain from reset synchronizer sources u_f3_sync1 and u_f3_sync2. One key observation to be noted here is that rst1's de-assertion is synchronized from the clk1 clock domain to the clk2 clock domain. Also note that the output of NAND gate will make deassertion of rst1 as assertion for u_f5. Since u_f5 is in clk1 domain, output of NAND gate will act as a new reset as the assertion of output signal of NAND gate is asynchronous to u_f5 clock domain. We will treat this signal as new reset and will term it "int_rst" as it is internally generated. The new reset will fall into a new reset domain and will create an additional RDC path between u_f5.q (int_rst reset domain) to u_f4.q (set1, rst1 reset domain).

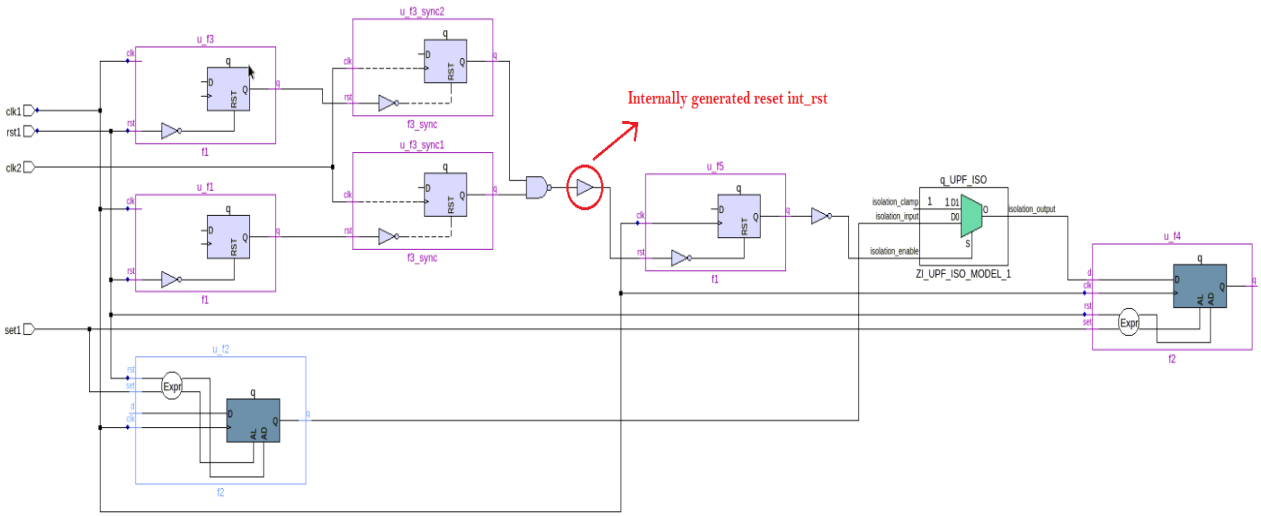


Figure 8. Isolation enables driven by internally generated resets

Case 6 – *Isolation enable signal driven by SR flop creating a new RDC path.*

In Figure.9 schematic, we can encounter two problems based on design implementations. First, the isolation cell enable signal is driven by SR f2 flop (set, rst domains). This creates a new RDC path between f2.q (set, rst reset domains) to u_f2.q (rst2 domain). Further, the reset and set pins of the f2 flop are driven by different internal soft/configurable synchronous reset registers u_f3 and u_f4. In a SoC, there could be instances where some part of the design logic is synchronous that gets no active clock during power-up or chip initialization [8]. This could be an intentional part by the designers for power reduction. In such cases when the system powers up, the chances are the set and reset pin values of f2 SR flop driven by the synchronous reset registers u_f3 and u_f4 can get corrupted and affects the isolation enable path.

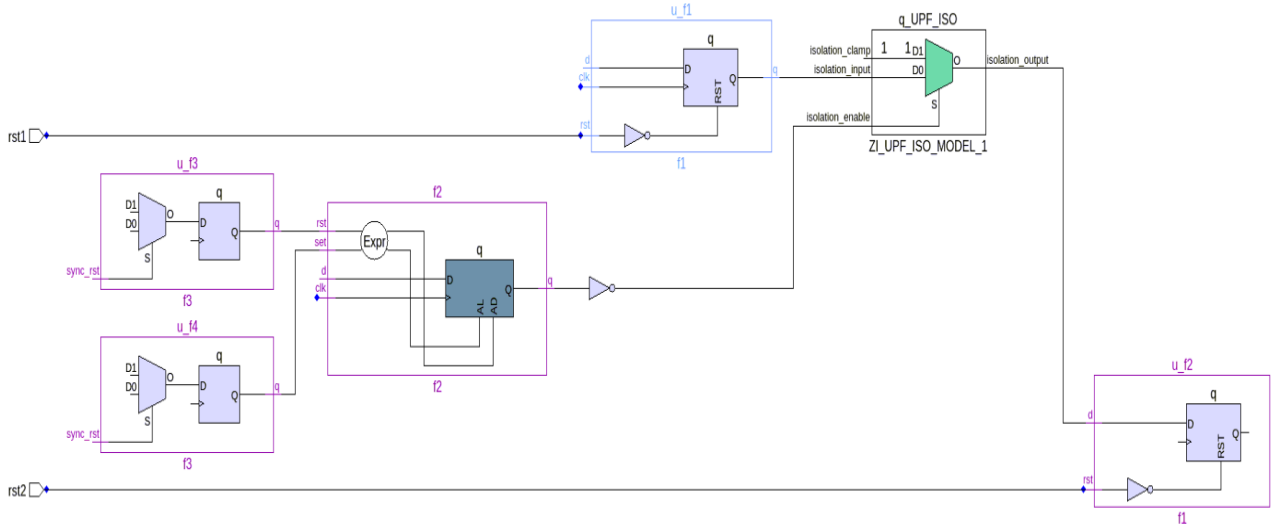


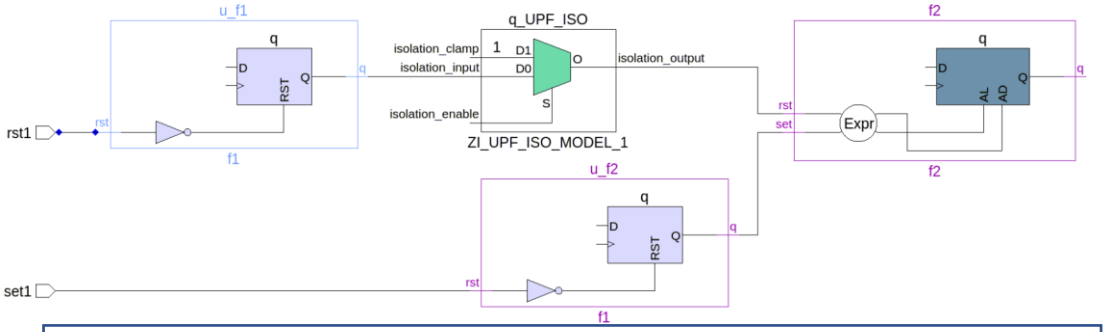
Figure 9. Isolation enable signal driven by SR flop creates a new RDC path

IV. PROPOSED NEW RULE CHECK SETS

When a design is implemented with a notion of power intent specification through the UPF artifacts, we need to ensure the original RTL functional behavior and its architecture is not impacted or disturbed by the additional power control logic. As a part of this, we have observed it is important to define some new rule checks and some of them are derived from our previous DVCON US 2021 paper on simple flip flops [1]. These rule checks help designers verify statically the reset propagation and integrity. Following are the scenarios which depict the need for these new rule checks.

Check 1: Asynchronous reset/set signal of SR flop driven by Power OFF domain.

In Figure.10 schematic, the asynchronous active low reset signal f2.rst is driven by source power domain u_f1 (rst1 domain) flop. When the design is operating in sleep/standby mode the power control configuration could turn off the power domain for the u_f1 instance. With power states defined for the stand-by mode the reset pin of f2 flop reads the clamp value of 1'b1 which deactivates the reset f2.rst as it is active_low reset. Hence the SR flop f2.q output in power saving/standby mode can be associated with set1 asynchronous active_low reset domain




```
create_power_domain PD_VDD1 -elements {u_f2 f2} -supply {primary SS_VDD1}
create_power_domain PD_VDD2 -elements {u_f1} -supply {primary SS_VDD2}
add_port_state VDD1 -state {ON_10 1.0}
add_port_state VDD2 -state {OFF off}
add_port_state VSS -state {ON_00 0.0}
create_pst PST1 -supplies {VDD1 VDD2 VSS}
add_pst_state NORMAL -pst PST1 -state {ON_10 OFF ON_00}
```

Figure 10. Asynchronous reset/set signal of SR flop driven by Power OFF domain

This check helps designers to ensure that all the registers are reset and no register is left out because of resets originating from the power off domains. This ensures the correct state of the design during reset.

Check 2: Asynchronous reset/set signal driven by isolation logic is always active due to clamp value.

In Figure.11 schematic, similar to Check1 as discussed above when the source power domain of flop u_f1 (rst1 domain) driving the asynchronous reset pin of f2 flop is isolated or power-down, the asynchronous active low reset signal f2.rst is driven by isolation cell logic whose clamp value is a constant 1'b0 that prevents the flop f2 from getting out of its reset state. Since f2.rst is a priority reset the set1 reset domain will no longer have any impact on the f2 flop during the power-down state.

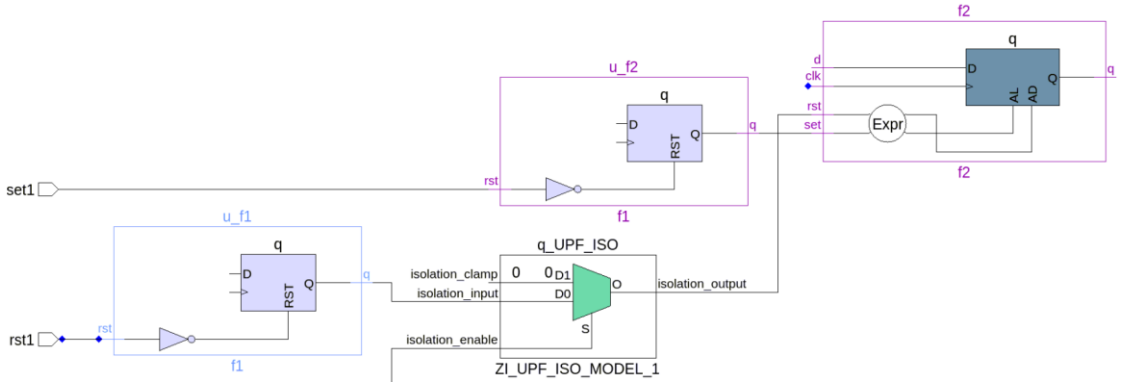


Figure 11. Isolation cell driving constant 0 making flop remain in the reset state

This check ensures users to verify whether the asynchronous reset signal, which is driven by a logic that is a constant value, prevents the resettable element from being released from the always active reset state or not.

Check 3: Isolation Enable is driven by a different reset group.

In Figure.12 schematic, we can see that the low power elements like isolation cell enable signal is driven by a different reset groups (rst1, set1 domains). Power control logic (isolation enable) can be driven by a simple resettable flop or a priority flop. In the case of a priority flop driving isolation enable can lead to additional RDC paths in the design as explained in the **Case 1** scenario. To resolve this we need to define the correct reset groups for the isolation signal such that it belongs to Tx or Rx reset domains or specify the correct reset assertion orderings between the reset pairs of isolation enable's reset and destination register's reset domain.

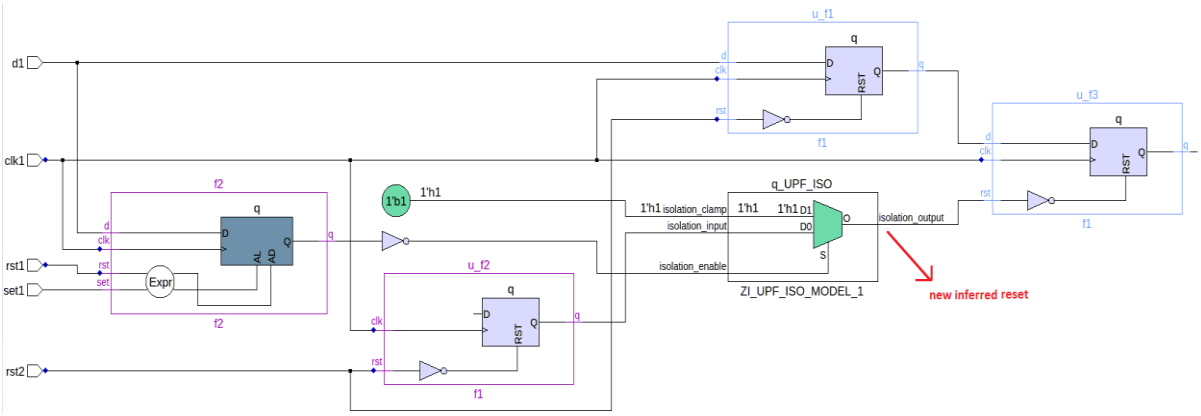


Figure 12. Isolation cell driving constant 0 making flop remain in the reset state

The check ensures that the designers resolve the increase in the new inferred resets (inferred due to combinational expr of {rst1, set1, rst2} domains) due to isolation cells.

V. CONCLUSION

This paper is part of our effort to bring out the new trending technology that deals with low-power designs to understand how reset functionality impacts the design. We have analysed the multi-domain representation and the crossing of power, clock, and reset domains. We have highlighted some of the new evading functional metastability design reset issues when the RTL is extended with UPF power management information. We have shown in this paper that each domain (power/reset/clock) cannot be verified independently and interactions between these domains need to be concurrent. The Set/Reset Priority flop cases in this paper indicate how the complexity of RDC structure increases many folds when compared to simple resettable flops in power-aware design. The static RDC low power-aware verification helps verify the new metastability issues due to complex Set/Reset flop structures that are undetected during simulations. We have also proposed to enhance the rule check sets for the designs having Priority flops. These rule check sets helps determine the correct reset architecture and connectivity to the blocks that are operating in different power domains. We have shown how low-power UPF

strategies like the insertion of isolation cells, retention logic, etc. can have a huge impact on the reset strategy as the number of new resets increase dramatically which in turn increase RDC paths.

VI. REFERENCES

[1] Inayat Ali, Abdul Moyeen, Manish Bhati, Manjunatha Srinivas, “Bringing Reset Domains and Power Domains together – Confronting issues due to UPF Instrumentation”. DVCon 2021

[2] Abdul Moyeen, Inayat Ali, “Reset Domain Crossing for designs with set-reset flops”. DVCon 2021

[3] Mark Handover, Jonathan Lovett, Kurt Takara, “Power Aware CDC Verification of Dynamic Frequency and Voltage Scaling (DVFS) Artifacts”. DVCon 2015.

[4] Kurt Takara, Chris Kwok, Naman Jain, Ashish Hari, “Next-generation Power Aware CDC Verification - What have we learned?”, DVCon, 2015

[5] P. Yeung, “Multi-Domain Verification: When Clock, Power and Reset Domains Collide”, DVCon, 2015.

[6] Anindya Chakraborty, Naman Jain, Saumitra Goel, “Power-Aware CDC Verification for Faster SoC Verification”, DVCon India 2014.

[7] Deepak Mahajan, Saloni Raina, Khushboo Gupta, “Reset connectivity checks in complex low power architectures”, https://www.design-reuse.com/articles/37798/reset-connectivity-checks-in-complex-low-power-architectures.html

[8] Rostislav (Reuven) Dobkin, “Asynchronous reset synchronization and distribution – challenges and solutions”, https://www.embedded.com/asynchronous-reset-synchronization-and-distribution-challenges-and-solutions/