

# Multi-Domain Verification: When Clock, Power and Reset Domains Collide

Ping Yeung, Erich Marschner  
Design & Verification Technology  
Mentor Graphics, Fremont, U.S.A.

Kaowen Liu  
Design Technology Division  
MediaTek Inc, Santa Clara, U.S.A.

*Abstract*- Integrated circuit design logic can be conceptually split into multiple types of partitions for inter-domain analysis. For example, a modern design typically has a power domain partition, a clock domain partition, and a reset domain partition. Historically, inter-domain analysis is confined to logic verification across boundaries of the same domain types (homogeneous domains): for example, power domain verification and clock domain crossing verification are performed as separate processes. As designs become hugely sophisticated and domains are inter-dependence of each other, this practice is no longer sufficient. Inter-domain logic between different types of domains (heterogeneous domains) is exceptionally complex and critical to the success of the device. Hence, new methodology is required to verify them effectively.

Multi-Domain Verification (MDV) is a comprehensive approach that analyzes and verifies both the logic that straddles homogeneous domains and the logic that straddles heterogeneous domains. It uses power domain information from the Unified Power Format (UPF) specifications, clock domain information from the clock tree models and reset domain information from the reset tree models. It employs specialized domain analyses and customizable verification methodologies to examine the complex interactions of logic that straddles boundaries—among both homogeneous domains and heterogeneous domains. Multi-domain verification is the only way to ensure that all inter-domain issues are explored and verified with complete confidence.

## I. INTRODUCTION

Many new SoC designs must integrate a lot of functionality and consume very low power. A design might have multiple processor cores to balance different load scenarios. It can have multiple complex logic clusters to handle system memory, system graphics, I/O interfaces, peripherals and other functions. To meet these requirements, a design typically operates in a spectrum of clock frequencies and uses a set of sophisticated power management strategies. It can incorporate advanced design technologies, such as asynchronous clocks, ratio-synchronous clocks, clock gating, multiple voltages, power switching, dynamic voltage and frequency scaling (DVFS) and so on. As an example, a design that we analyzed recently has close to 20 power domains, over 200 asynchronous clock domains and up to 32 reset domains. But before tackling the complex interactions of these domains at the top SoC level, the better approach is to verify the IPs first. We have analyzed a range of complex IPs. One interface IP has about 10 power domains, 20 clock domains and 3 reset domains. Another processor IP has 6 power domains, 7 ratio synchronous clock domains and 15 asynchronous reset domains[1].

Interaction of logical function spanning out in different power, clock and reset domains can potentially cause spectacular chip failures. Yet, design teams face several challenges to ensure these domains are working correctly with respect to each other. Although design teams usually do a good job partitioning a design into multiple power, clock and reset domains at the chip-level, there is challenging to understand how these domains interact with each other at the block or lower levels. The situation is made worse when design teams are integrating multiple IPs that they know little about together. In order to save power in today's designs, blocks are switched on and off continuously. Standard verification tools and methods offer no *reliable way* to analyze the different domains together and to verify their inter-operation comprehensively. Power-aware simulation is a good start, but there are too many combinations of power, clock and reset domains. It is impossible to achieve sufficient coverage with simulation to verify the inter-domain interaction dynamically. Hence, static domain analysis is essential to examine the design space thoroughly. Multi-Domain Verification (MDV) is developed to address the challenges in this area.

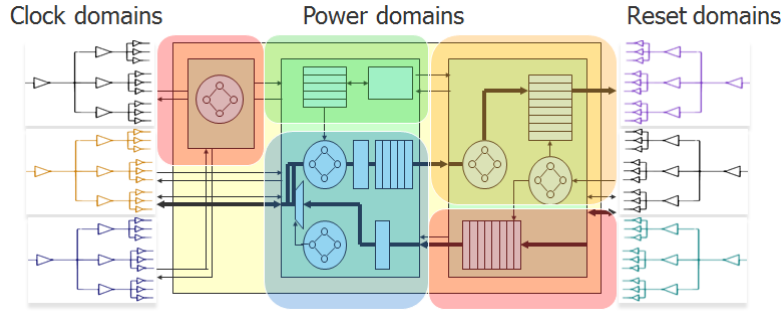


Figure 1. Multi-Domain Verification

This paper has three sections.

- In the first section, we summarize current power domain verification, clock domain crossing verification and reset domain crossing verification methods. They provide comprehensive verification of the homogenous domains, but at the same time, they fail to consider the impact of the other domains.
- In the second section, we explain the general concept of Multi-Domain Verification and its data model. Performing checks between heterogeneous domains first addresses two limitations in existing tools: (1) What are the effects of other heterogeneous domain (such as power domain) on domain-defining signals such as clocks and resets? (2) What happens to the domain-controlling signals (such as clock gating signals) when they across heterogeneous domains?
- Since Multi-Domain Verification is a new discipline, in the final section, we describe the findings from a few designs, and share some of the lessons we learned.

## II. INDIVIDUAL DOMAIN VERIFICATION

### A. Clock Domain Crossing

Clock domain crossing (CDC) logic must follow strict design principles for reliable design function [5][6]. But because of transistor-level analog effects in circuits, verifying CDC logic is not possible using standard simulation and static timing analysis techniques.

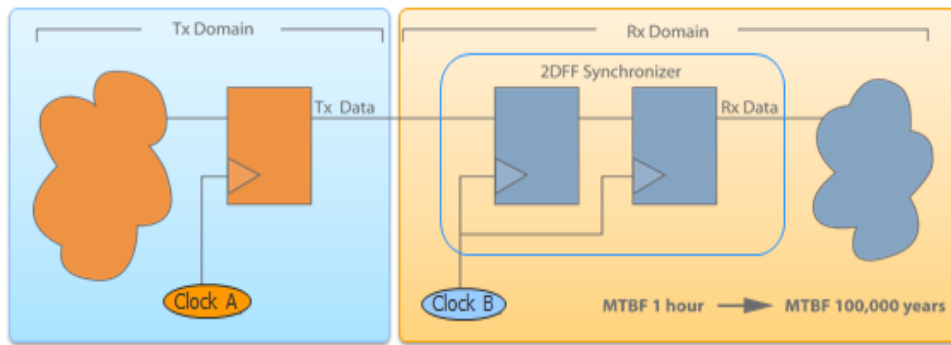


Figure 2. Clock Domain Crossing

To handle this dilemma, CDC verification is divided into three phases [7]:

1. Identify CDC signals and statically validate the intended CDC schemes.
2. Dynamically and/or formally verify CDC protocols. This process ensures the CDC control and data signals are stable after crossing domain boundaries and possible synchronization.
3. Simulate the design with metastability injection and/or formally verify reconvergent structures. This ensures the chip is tolerant of unpredictable delays from the synchronizers embedded in the CDC paths.

CDC structures and schemes are quite well understood. But when a design with 100s asynchronous clocks and 1000s clock gating signals operate in multiple power domains, their interactions are difficult to predict.

### B. Reset Domain Crossing

Today's large SoC designs integrate internal IPs and external IPs from multiple sources. This presents a significant challenge for design teams since there is no standard for the block level reset circuitry. System level reset strategy also becomes increasingly complex as they combine various sources of reset [8]. For example, reset signals are needed for each power domain and they also must be synchronized to their target clock domains before use.

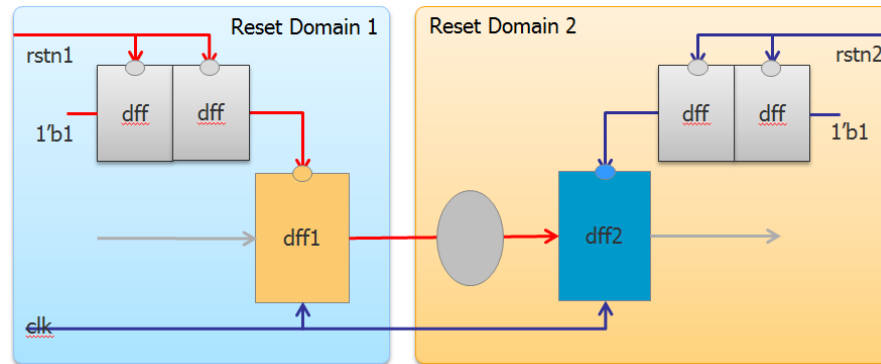


Figure 3. Reset Domain Crossing

When a set of data dependent registers in the same clock domain, are driven by different reset signals, Figure 3, the asynchronous reset signal on the source register will cause an asynchronous event to happen at the receiving register. If *rstn1* is asserted while *rstn2* is not asserted, the asynchronous output data from *dff1* can cause metastability on *dff2*. This is the reset domain crossing we are concerned about.

Reset domain crossing verification can be divided into 3 parts [9]: 1. Static analysis is used to find and build the tree structure of all the reset signals in the design. Resets can be classified as either synchronous or asynchronous resets, as well as active high or active low. 2. Rather than rely on gate level simulation, RTL simulation with silicon-accurate X-propagation semantics can be used to find issues related to uninitialized design elements. 3. Formal verification can be used to verify many aspects of the reset tree, including: connectivity of all sources of resets to their intended destinations and detecting corruption of correctly reset storage elements with unknown logic values.

To lower power consumption, design teams are adding more complex power domains into the design. The activity of the reset signals (as part of the power sequence) has increased significantly. It is difficult to foresee the possible interactions of the reset signals on the connected modules.

### C. Power Domain Crossing

With the Unified Power Format (UPF) standard [2], project teams are able to capture the power intent of the design. The power intent specification consists of power domains, supply networks, power structures, and power states. The verification goal is to check the functionality of the power management elements to ensure they are working properly. Each subsystem can be turned-on, transitioned into its power-modes and shut-down independently.

A power-aware verification tool can check the placement of isolation cells and level shifters on the power domain crossing signals. The isolation cells will be controlled by the power controller with respect to the power-up and power-down sequences. If retention registers are used, the save and restore operations will also need to be verified. To help guide verification teams on this new challenge, a power verification checklist is presented in [3]. It focuses on various aspects of power domain toggling, isolation cells, memories, and retention of registers.

As the number of power domain continues to increase in designs, they have created a heavy burden for both design and verification. By supporting the UPF standard, we enable design teams to verify their implementation early at the RT level.

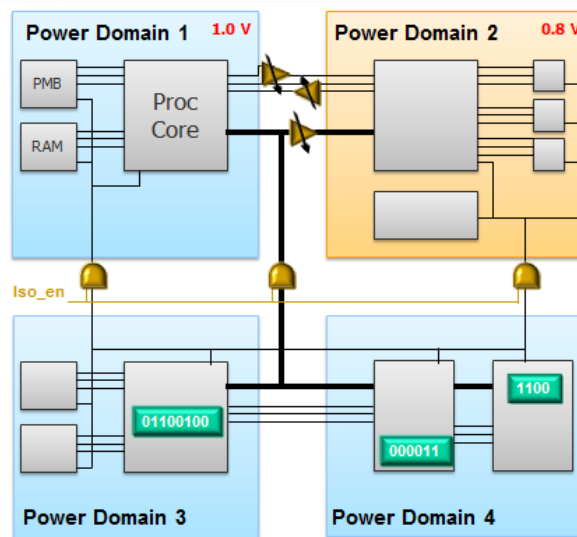


Figure 4. Power Domain Crossing

### III. MULTI-DOMAIN VERIFICATION

As designs are getting more complex and heterogeneous domains are interacting more with each other, it is insufficient to verify one domain at a time. To account for the introduction of power domains, some CDC verification tools [10][11] are starting to leverage the power domain information from the UPF specification. It helps identify and verify power control signals that are crossing the clock domains. It is a step in the right direction to verify the interactions between domains. However, it still lacks a complete domain specific view of the design. Hence, we want to take a major step forward to architect a new verification environment. It will have all the domain information extracted and represented persistently. This will allow the environment to verify all the domain interactions comprehensively. At the same time, extra information from a different domain can be pulled in on-demand to refine and understand the severity of an identified situation.

We named this approach Multi-Domain Verification (MDV). It extracts the power domain information from the Unified Power Format (UPF) specification, the clock domain information from the clock trees and the reset domain information from the reset trees. It is equally important to represent the logic functions controlling the different domains: power control signals, clock gating signals and reset enabling signals. To truly understand the interaction between domains, we have to examine the functionality of the control signals. For instance, if the clock to a register is gated off or the register is in retention mode, the register is going to be very stable. Hence, clock domain crossing or reset domain crossing from the register will not be a problem. Hence, for each domain, there are 3 types of information as shown in Figure 5: domain structure information, domain control logics and domain crossing signals.

	Domain Structure	Domain Control	Domain Crossing
Domain Verification	Structure Verification	Control Verification	Crossing Verification
Power Domain	Power supply nets	Power switching, isolation and retention control	Power domain crossing (PDC) Isolation, Retention, Level shift
Reset Domain	Async and sync reset, tree structure, connectivity	Reset synchronization, polarity, control and gating conditions	Reset domain crossing (RDC)
Clock Domain	Async and sync clocks, tree structure, connectivity	Clock division, control and gating conditions	Clock domain crossing (CDC) Synchronizers, 2DFFs, FIFO and handshake structures

Figure 5. Domain control and crossing signals

As shown in Figure 6, the multi-domain verification process is divided vertically into 3 phases: domain structure verification, domain control verification and domain crossing verification. Instead of verifying one homogeneous domain at a time, this vertical phase by phase (heterogeneous) approach is unique. It allows users to focus on the big picture first. For instance, if the power domain specification is wrong, or the clock tree is not well defined yet, it is not going to be productive examining the detail violations generated from the power or clock domain crossing signals. Users should focus on refining the UPF specification, the clock trees and the reset trees first. Once the early phase has been finalized, users can move onto the latter phase. Consequentially, the latter phases can take into account and leverage all the information, augmentation, and refinement done in the previous phases.

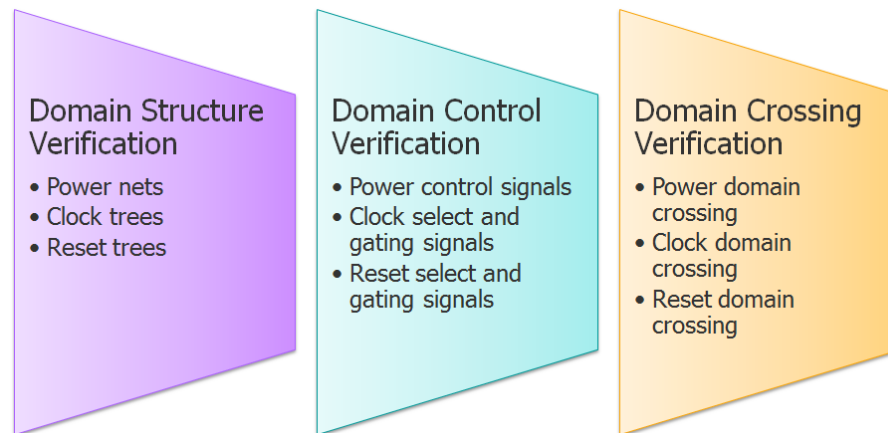


Figure 6. The 3 phases of Multi-Domain Verification

#### A. Domain Structure Verification

Instead of verifying the structure of a domain one at a time, multi-domain verification extracts, represents and verifies the structures of the power domain, clock domain and reset domain at the same time. It allows us to understand how the domain structures, such as clock trees and reset trees, is affected by the presence of the other domains.

	Power Supply	Reset Tree	Clock Tree
Power Domain		Reset tree in multiple power domains	Clock tree in multiple power domains
Reset Domain	Power in multiple reset domains		Clock tree in multiple reset domains
Clock Domain	Power in multiple clock domains	Reset in multiple clock domains	

Figure 7. Structures in different domains

For instance, as depicted in Figure 7, there are a number of challenging scenarios:

1. a reset tree in multiple power domains – it is desirable if the reset signal is generated from an always-on domain and consumed in various other power domains. However, it is potential risky, if the reset signal is generated from in one power domain and fanned out to another.
2. a clock tree in multiple power domains – similar to the reset tree, it is good if the clock signal is generated from an always-on domain and consumed in various other power domains. It will be risky, if the clock signal is generated from in one power domain and fanned out to another.
3. a clock tree in multiple reset domains – there will be potential asynchronous reset domain crossing issues. In addition, one part of the design could be reset while the other part is not. It could be challenging to initialize such a design correctly.

There are also some scenarios that are common in designs.

1. a power supply in multiple reset domains – it is normal for the power supply net to cover multiple reset domains.
2. a power supply in multiple clock domains – it is normal for the power supply net to cover multiple clock domains.
3. a reset tree in multiple clock domains - it is normal for the reset tree to be used in multiple clock domains. However, the reset signal has to be synchronized into each of the clock domain before use [12]. The reset signal is asserted *asynchronously*, but has to be removed *synchronously* with respect to the target clock domain.

The objective of structure verification is to ensure that the power supply nets, clock signals and reset signals are distributed to different parts of the design correctly. During the process, domains are colored, the domain boundaries are carved, and the number of domains is calculated. Then, for each design element in the design (such as register, memory array, and module), we will know the power, clock and reset domains it is associated with.

### B. Domain Control Verification

As power switching, frequency switching and clock gating are common practices for low-power designs, it is important to verify the domain control signals as well. From Figure 5, domain control signals include power switch, isolation and retention control signals; clock select, control and gating signals; reset control and gating signals. The power control signals are generated by the power controller. The clock control signals can be from a number of sources. For dynamic voltage and frequency switching, the clock frequency selection signals will be from the power controller. To design for test, the scan and functional clock selection signals will be from the test controller. Other internal clocks are determined by the mode of operation and are programmable by software. They are controlled directly or indirectly by the configuration registers. Locally, clocks driving the data path elements can be gated by enable signals to save power. After examining all these control logics fanning into a clock tree, it is very likely that the clock is controlled by a network of logic more complex than the clock tree itself.

For domain control signals, there are two important aspects that need to be verified:

1. Domain dependence – it is important to make sure that the control signals are:
  - a. From an always-on power domain or in the same power domain
  - b. From some stable registers or from the same synchronous clock and reset domain
2. Functionality – to power up or power down a domain correctly, the power supply, the clock and the reset signals have to be asserted and removed following a precise sequence. To do so, the corresponding control signals, the power switch, clock and reset gating logic have to have the right values at the right time.

The domain control logic, especially the clock control logic, can be very complex. It is impossible to simulate all the possible functional scenarios and corner cases. Hence, formal connectivity verification is a natural fit to verify the conditional connectivity of the clock tree structure.

### C. Domain Crossing Verification

After domain structure verification, we will know the power, clock and reset domains associate with each design element in the design (such as register, memory array, and module). Domain crossing analysis will examine all the data dependent elements in the design.

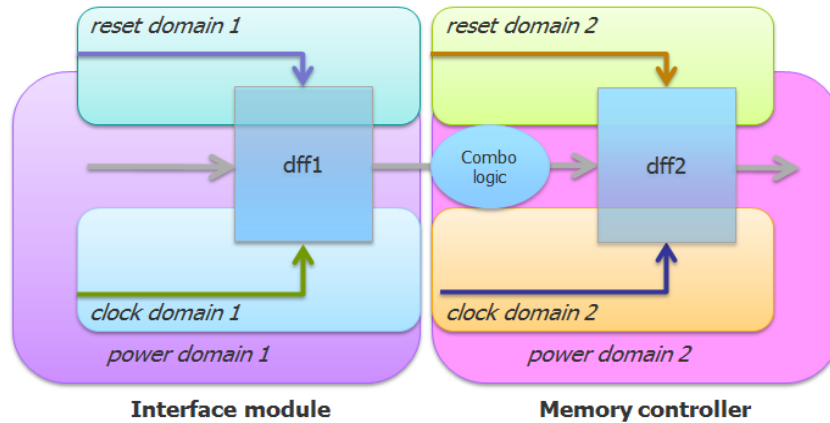


Figure 8. Power, Clock and Reset Domain Crossing

Domain control verification will validate the power, clock and reset domains of the design elements. If there is domain crossing of any type, power, clock or reset, traditional domain crossing verification can be performed to ensure the crossing is implemented correctly. It is very beneficial to leverage the existing homogeneous domain verification technology as described in Section 2. Clock domain crossing and other domain verification tools are mature and come with a comprehensive debug environment. Once the domain crossing verification requirements have been established, they can be deployed locally to verify all the crossings between modules.

For instance, in Figure 8, an interface module is connected to the memory controller. In certain mode of operation, if the interface module is powered off completely, only power domain crossing analysis needs to be performed between the interface module and the memory controller. Clock domain crossing and reset domain crossing analyses are not required. On the other hand, if the interface module and memory controller are in the same power domain, their clock and reset domains will be examined to determine whether clock or reset domain crossing verification is required.

## IV. RESULTS

A prototype multi-domain verification environment has been stitched together. We have applied multi-domain verification methodology on a few customer designs. The verification was performed at the RT level with UPF specification. One major benefit of performing verification with UPF is that the power structures are not casted in stone yet. Any modification (as it did happen) can be done without significant impact on the implementation schedule. Results from 3 of the blocks are summarized in table1 below. With multi-domain verification, the tool was able to identify all the power, clock and reset domains correctly. Users can amend the supply net groupings, clock and reset tree groupings within each domain by assigning explicit values to the control signals, or by grouping the clock and reset signals explicitly with directives.

Number of domains	Block 1	Block 2	Block 3
Power domains	2	3	4
Asynchronous Clock domains	3	4	16
Asynchronous Reset domains	2	11	9

Number of trees crossing domains	Block 1	Block 2	Block 3
Clock trees crossing power domains	2	2	6
Clock trees crossing reset domains	2	3	5
Reset trees crossing power domains	1	2	7
Reset trees crossing clock domains	2	5	16

Number of control signals crossing domains	Block 1	Block 2	Block 3
Clock control signals from different power domain	0	4	5
Reset control signals from different power domain	0	2	13
Power control signals from different clock domain	2	4	6

Number of data signals crossing domains	Block 1	Block 2	Block 3
Power domain crossings	5/0	142/0	441/0
Clock domain crossings	20/6	5845/4047	10956/3074
Reset domain crossings	3/0	25768/129	225/15

m/n: evaluations/violations

Table 1 Summary of Multi-domain verification

Block1 is a peripheral interface controller with 2 power domains. The clock and reset trees cross the power domains a few times. They are mainly going from the always-on power domain to other switchable power domains. When a clock tree is crossing reset domains, it means that different reset signals are used to initialize registers within the same clock domain. As some of the registers are depending on each other, this leads to the 3 reset domain crossings reported latter in the table. Reset trees crossing power and clock domains are normally done by design. It means that the same reset tree is being used in different clock or power domains. This will be a problem however if the power domain needs to be reset independently with respect to other domains in the design. As the clock and reset signals are generated externally for this block, there is no control signal for them.

On the other hand, there is an issue with the power control signal. From the schematic in Figure 9, an isolation cell is used to isolate the switchable and the always-on power domains. The two power domains belong to the same clock domain. However, the isolation signal is from a different clock domain (such as the power controller clock domain). As a result, the isolation signal introduced a clock domain crossing signal into the design. To design this correctly, the isolation signal should be synchronized to the RX clock before used. Hence, using a single isolation signal for all the output ports can easily lead to this problem.



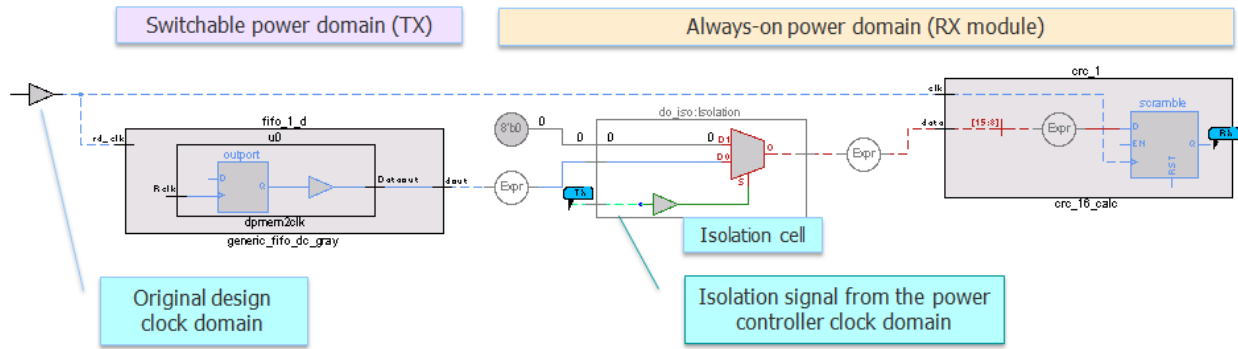


Figure 9. Clock domain crossing power isolation signal

Another interesting situation we have observed is reconvergence of CDC paths from different power domains, Figure 10. This happens when data from different modules are re-converging at the controller. We know that reconvergence is an issue for CDC verification. Power domains add another dimension to the problem. If the switchable power domain is off, there will be no reconvergence of CDC paths. When it is on, we will need to consider the reconvergence problem seriously.

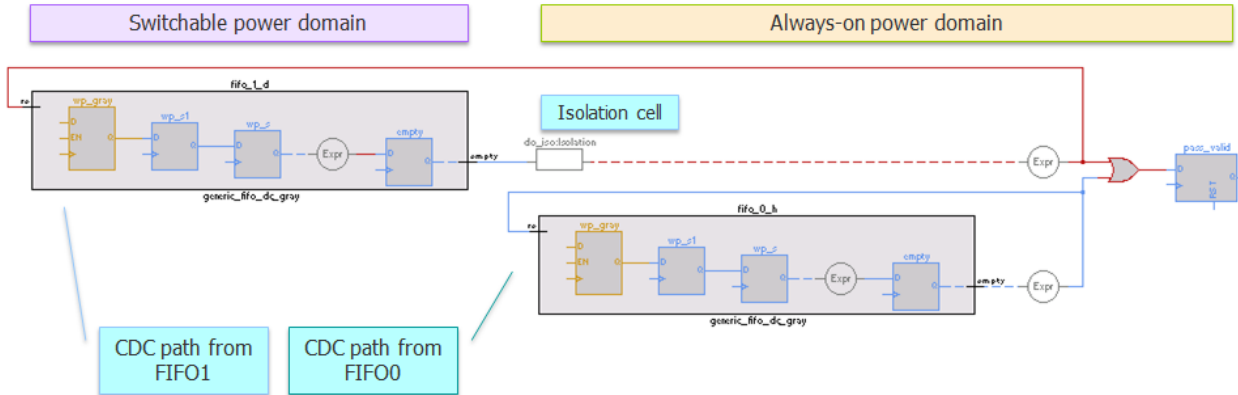


Figure 10. Re-converging CDC paths from different power domains

Block2 is a functional controller for a design. It has a few clock and power domains, but it generates a lot of reset signals to control structures locally and to other blocks of the design. For this block, we were focusing on verifying the reset structures. Power domains and reset signals are closely related. Each switchable power domain should have a reset signal to initialize the storage elements every time when it is powered on. However, when data is flowing from one power domain to another, it is not a good idea to use different reset signals to initialize the registers when they are residing within a same clock domain. In Figure 11, a reset signal is needed for the RX module in the switchable power domain. Instead of using reset1 from the TX module, the designer synchronized an external reset, reset2, for RX module. As a result, different reset signals are used for the data path from TX module to RX module. An undesirable reset domain crossing path is introduced.

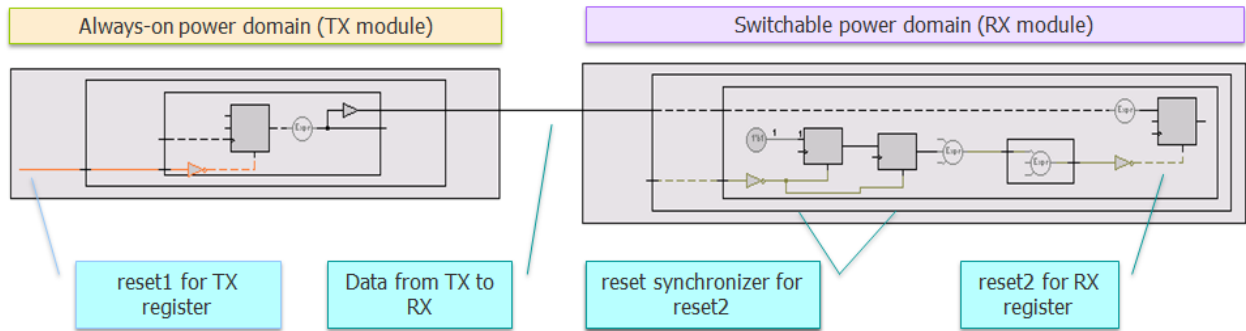


Figure 11. Reset domain crossing between two power domains

Besides the clock and reset signals, their corresponding control and gating signals are also very important. With more and more clock gating and reset control signals in today's design, they should be represented and verified in conjunction with the clock and reset trees. In Figure 12, an external reset signal is synchronized and is used in the RX module. This reset signal can be disabled by a control signal from the TX module in a different power domain. In normal operation, the reset in RX module will be disabled once RX module has been initialized correctly and has entered functional mode. However, when the RX module is powered off and on again, the disabled reset signal cannot be used to initialize the design. This is a big concern for the design team. As a result, extensive connectivity verification has been performed on the reset trees and the associated control signals. The goal is to ensure all the generated reset signals can be asserted with the desired conditions. Besides using formal method to verify the connectivity, it is also very useful to generate assertions for simulation. With power-aware simulation, the reset tree and control signals can be verified when different modules are being powered on and off as in the targeted system environment.

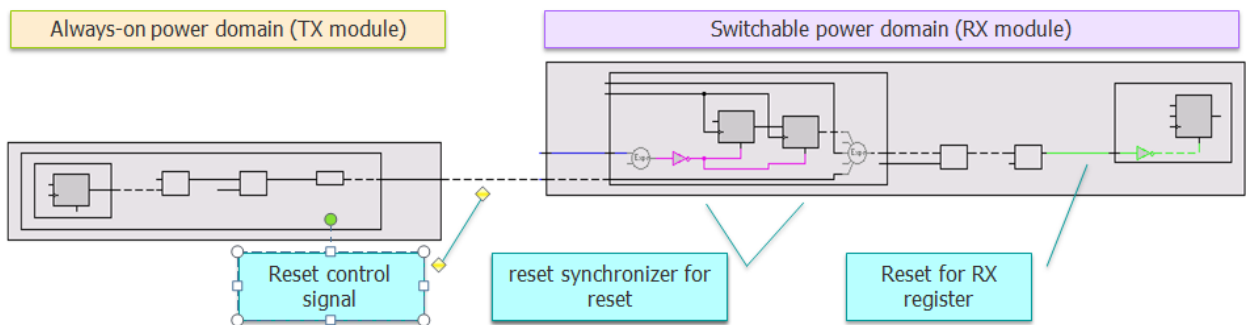


Figure 12 Reset control signal from a different power domain

Block3 is part of an interconnect controller for multiple processor cores. The design has 10 primary clock signals. Some of the clock signals are interface specific and they only go to the specific interface modules. Other system level clock signals are distributed widely. Since the top-level of the design is in an always-on power domain, most of the clock trees going from the top to a switch-able power domain are not a concern. We were particularly worrying about clock trees going from one switch-able power domain to another switch-able power domain. After reviewing the results carefully, we were happy that this scenario had not happened.

However, that is not the case with the reset signals. The design has 3 primary reset signals. From them, internal reset signals, warm reset signals, and synchronized reset signals were derived internally. From the top-level, the reset signals are driven into each of the power domains and at the same time, synchronized into each of the clock domains respectively. After reviewing the issues reported by the tool, we have found a few cases that the reset signals are crossing power domains. As shown in Figure 13, the two reset signals for the RX module are passing through a switchable power domain. The IP module is a data computational unit. It gets its reset signal from the top-level module and the input reset signal is also used to drive two output reset signals for the receiving datapath. The potential problem is: if the switchable power domain is powered down, it will be impossible to reset the RX module any more. From the schematic, it is quite clear that the RX module does not need to get the reset signals from the IP module, instead, it can get the reset signals directly from the top-level. When we reviewed this situation with the

design team, they told us that they do not know the functionality of the IP module. They used the reset signals from the IP module to ensure that the receiving datapath will be in-sync with the computation in the IP module.

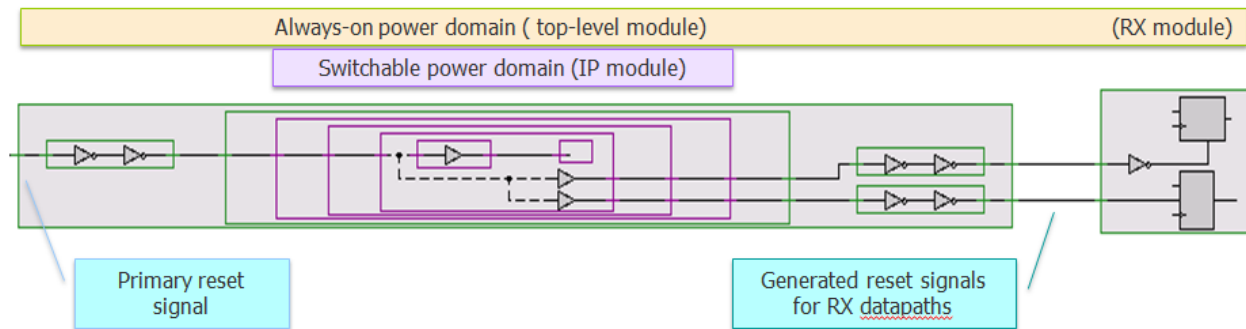


Figure 13 Reset signals crossing different power domains

## V. CONCLUSION

This paper presents the Multi-Domain Verification approach that verifies the power, clock and reset domains with UPF specification concurrently at the RT level. It analyzes both the logic that straddles homogeneous domains and the logic that straddles heterogeneous domains. By representing and verifying the 3 domains together, it is more intuitive to understand the interaction between them and hence, anticipate any domain issue as early as possible in the design cycle. The multi-domain verification process is divided into 3 phases: domain structure verification, domain control verification and domain crossing verification. This divide-and-conquer approach allows us to focus on the different aspects of the multi-domain verification problem. While we have to develop new algorithms for multi-domain structure and control verification, we can leverage the established homogeneous tools to perform normal clock, power and reset domain crossing verification. With multi-domain verification, all inter-domain issues can now be verified with complete confidence.

## VI. FUTURE WORK

The same multi-domain verification methodology can be expanded to include other types of partitions for inter-domain analysis. For instance, it can include test and scan domains, security domains, etc.

## VII. REFERENCES

- [1] "ARM® Cortex®-A17 MPCore Processor Technical Reference Manual Revision: r1p0", <http://infocenter.arm.com>.
- [2] Unified Power Format, "1801-2013 IEEE Standard for Design and Verification of Low-Power Integrated Circuits", IEEE.
- [3] Freddy Bembaron, Sachin Kakkar, Rudra Mukhejee, Amit Srivastava, "Low Power Verification Methodology Using UPF", DVCon 2011.
- [4] Amit Srivastava, Madhur Bhargava, "Stepping into UPF 2.1 world: Easy solution to complex Power Aware Verification", DVCon 2014.
- [5] Ran Ginosar, "Metastability and Synchronizers, A Tutorial", IEEE Design and Test 2011.
- [6] Cliff Cummings, "Clock Domain Crossing (CDC) Design & Verification Techniques Using SystemVerilog", Synopsys User Group Meeting (SNUG) 2008.
- [7] Chris Kwok, Vijay Gupta & Tai Ly, "Using Assertion-Based Verification to Verify Clock Domain Crossing Signals", DVCon 2003
- [8] Kaowen Liu, Penny Yang, Jeremy Levitt, Matt Berman, Mark Eslinger, "Using Formal Techniques to Verify System on Chip Reset Schemes", DVCon 2013.
- [9] Chris Kwok, Priya Viswanathan, Ping Yeung, "Addressing the Challenges of Reset Verification in SoC Designs", DVCon 2015.
- [10] Anindya Chakraborty, Naman Jain, Saumitra Goel, "Power Aware CDC Verification at RTL for Faster SoC Verification Closure", DVCon India, 2014.
- [11] Kurt Takara, "Next-generation Power Aware CDC Verification – What have we learned", DVCon 2015
- [12] Cliff Cummings, Don Mills, Steve Golson, "Asynchronous & Synchronous Reset Design Techniques", SNUG 2003 Boston.