A Specification-Driven Methodology for the Design and Verification of Reset Domain Crossing Logic

Priya Viswanathan (priya_viswanathan@mentor.com)
Kurt Takara (kurt_takara@mentor.com)
Chris Kwok (chris_kwok@mentor.com)
Islam Ahmed (islam_ahmed@mentor.com)
Mentor, a Siemens Business
46871 Bayside Parkway
Fremont, CA 94538

***Abstract*- With the increasing complexity of today's System-on-a-Chip (SoC) designs, reset architectures have also increased in complexity. Traditional reset design and verification techniques have not evolved to address this increase in complexity. In order to avoid ad-hoc reset methods, this paper presents a specification-driven methodology to enable the design and verification of reset domain crossing (RDC) paths in large SoC designs. This methodology is a 3-step process that provides a requirements-based approach for RDC design and verification.**

I. INTRODUCTION

Modern System-on-a-Chip (SoC) designs incorporate hundreds of Semiconductor Intellectual Property (SIP) blocks, resulting in complex reset architectures, with many asynchronous reset domains [1]. The complex reset architectures enables transmission of data across sequential elements reset by different asynchronous reset domains thereby causing reset domain crossings (RDC) paths [1,2]. For RDC paths, an asynchronous reset at the source register may cause a destination register to sample an asynchronous event and become metastable. This metastability will cause unpredictable values to be propagated to down-stream logic and prevents a design to reset to a known good state which makes device power-up unreliable. In the worst case, reset issues may cause a device to consume too much power during assertion of reset, causing the device to be permanently damaged.

This paper presents a specification-driven methodology for the design and verification of RDC paths in real-world SoC designs. This methodology is a systematic and repeatable solution that includes:

- Step 1: RDC design requirements specification and verification plan
- Step 2: RDC design and verification
- Step 3: RDC results progress tracking and completion metrics

This RDC methodology is supported by an integrated requirements management capability to enable requirements traceability. We describe how the incorporation of RDC design requirements improves the design and verification process and creates a systematic approach for designing and verifying the reset architecture. Finally, the incorporation of coverage allows the correlation between the design requirements and verification efforts, and enables closure for the verification process.

II. SPECIFICATION-DRIVEN RDC METHODOLOGY

A. RDC Design Requirements Specification and Verification Plan

The objective of the reset architecture is to initialize every sequential element in the design directly or indirectly, so the RDC design requirements specification is a critical part of the reset architecture design. The design specification includes the definition of the asynchronous clock and reset domains, reset assertion ordering, and RDC synchronization methods. By clearly defining the reset architecture, design teams will facilitate proper RDC design, improve verification efficiency, and avoid RDC issues late in the design flow. The definition of the asynchronous reset domains will influence the RDC paths and RDC synchronization methods. The RDC specification will also include specifying reset ordering, RDC clock isolation enables, and RDC data isolation enables.

Advanced design methodologies also include the definition and documentation of the RDC verification tasks that allow a correlation between the design process, the verification procedures and the final results. To achieve this correlation, the RDC verification tasks should be defined, documented, and measured during the design process. In advanced RDC methodologies, coverage plays a fundamental role in enabling requirements specifications, activity definition, progress tracking, and completion reporting.

B. RDC Design and Verification

There is a number of design techniques used in reset architecture to mitigate RDC problems. In this paper, we will discuss a reset staging method and also a RDC synchronization method.

The staged reset synchronization structure enables sequenced removal of resets, minimizing the dangers of asynchronous and concurrent reset assertion. The staged resets avoid power surges by sequencing the assertion of resets to design blocks. Staged reset removal structures enforce implicit de-assertion ordering on data transmitted between adjacent reset domains. For example (Figure 1), if 'rsta' is the first staged synchronizer output and the 'rstb' is the second staged synchronizer output, all RDC crossings from the 'rsta' domain to the 'rstb' domain are ordered and follow the correct de-assertion sequence. Automatic verification of the function of such staged reset structures improves the RDC verification efficiency.

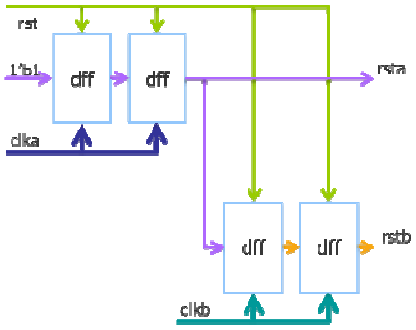

Figure 1. Staged Reset Generation.

Another technique for sequencing resets is to insert delays within an asynchronous reset domain. Understanding of the reset architecture also enables the inference of the reset assertion sequence from insertion of register delays within the same clock domain. In Figure 2, the assertion of RST will cause DFF2 to reset while DFF1 is still in functional state. The inconsistent reset delay between adjacent registers causes incorrect data at the downstream logic. The reset structure assumes that RST is asserted for more than 3 cycles. If RST is only asserted 2 cycles, then when RST deasserts, DFF2 will be corrupted by DFF1 before DFF1 gets the reset signal.

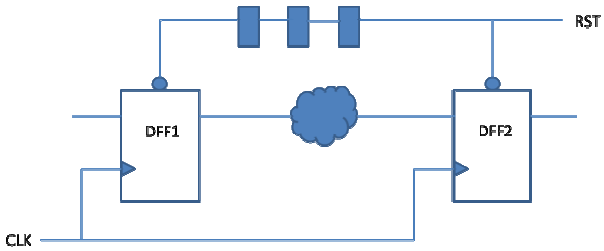

Figure 2. Reset Sequencing.

One common RDC synchronization method is to isolate the receiving domain register from the source domain register. This requires an enable signal to be generated in receiving register's clock and reset domain which isolates the receiving register from the transmitting register when the reset is asserted. Isolation by gating can happen through the data path or the clock path of the receiving register, as shown in Figure 3 and Figure 4. Specification of

the isolation mechanisms for RDC includes the method of implementation, the source of the isolation enables, the structural details and the possible mapping to the source and destination reset and clock domains.

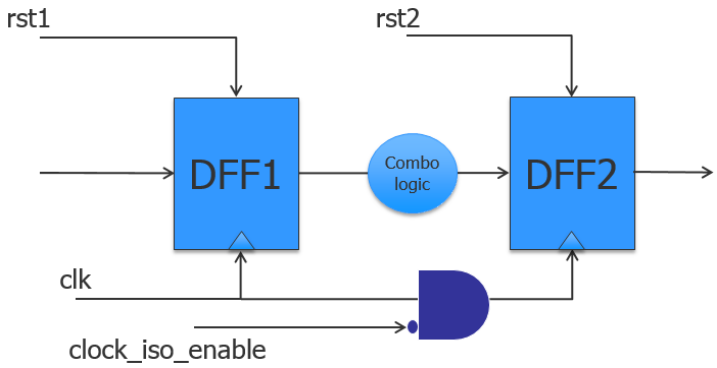

Figure 3. RDC Clock Isolation Method

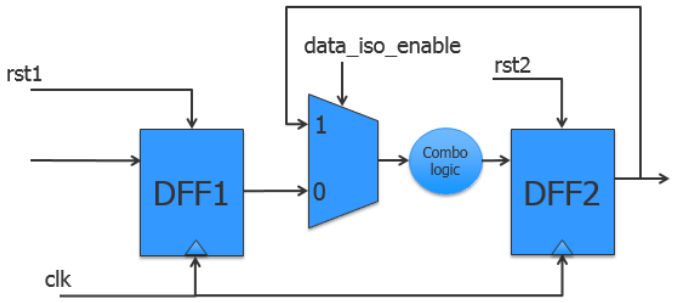

Figure 4. RDC Data Isolation Method

The RDC verification will verify the reset distribution and verify the RDC paths. The analysis will check that each RDC path is correctly synchronized and report any RDC path that is missing a synchronization structure or may be incorrectly synchronized. In addition, any correctly ordered RDC path will be reported as a correctly synchronized path.

C. RDC Results Progress Tracking and Completion Metrics

Advanced RDC methodologies utilize a structured flow for verifying designs for RDC issues. In order to efficiently manage RDC verification efforts, RDC solutions must report verification results and generate coverage metrics. By quantifying the verification tasks, design teams have explicit criteria for completing each step in RDC verification methodology. RDC verification coverage metrics will quantify the verification methodology including the setup, analysis, review, and debug tasks [3] and these coverage metrics will be correlated to the initial RDC verification plan (Figure 5). The correlation between the requirements and coverage will allow the project team to determine the completed RDC verification tasks and the incomplete RDC verification tasks. This correlation will also enable the design team to demonstrate RDC verification completion and design tape-out readiness.

#	Section	Description	Link	Type	Weight	Goal
1.5	Reset Synchronization	Coverage section for the reset synchronizers in the design under test			0	0
1.5.1	Custom Synchronizers	Custom Sychronizers Section			0	0
1.5.1.1	No redundant Custom Synchronizers	Ensure that there are no redundant custom synchronizers in the design.	/top/STATIC_RC_RESULTS/SYNC/NO_SERIES_REDUNDANT	Bin	2	100
1.5.1.2	Custom Synchronizers are properly connected	Ensure that all the custom synchronizers in the design are properly connected.	/top/STATIC_RC_RESULTS/SYNC/NO_CS_MISMATCH	Bin	2	100
1.5.1.3	No Combo-logic before Custom-synchronizer	Ensure that there are no combo-logic driving custom sync ports.	/top/STATIC_RC_RESULTS/SYNC/NO_CS_COMBO_LOGIC	Bin	2	100
1.5.2	Reset Synchronization Issues	Reset Synchronization Issues Section			0	0
1.5.2.1	No Combo-logic before RDC synchronizer		/top/STATIC_RC_RESULTS/SYNC/NO_COMBO_LOGIC_BERFORE_RDC_SYNC	Bin	2	100

Figure 5. RDC Testplan and Coverage

II. Real World Results

This methodology was applied on both an IP block and a digital networking SoC design. Both designs used clock gate isolation methods in the RDC structures. These designs also show how reset domain crossing paths may overlap with clock domain crossings (CDC) paths, so traditional CDC synchronization structures may address both RDC and CDC issues.

In the specification-driven RDC flow (Figure 6), the specification determines the RTL RDC synchronization implementation and also guides the RDC analysis. The XML testplan and UCDB coverage database are generated from the RDC analysis results. In this flow, the RDC structural checks are mapped to functional coverage covergroup bins in the UCDB and a functional coverage viewer is used to view the UCDB coverage database.

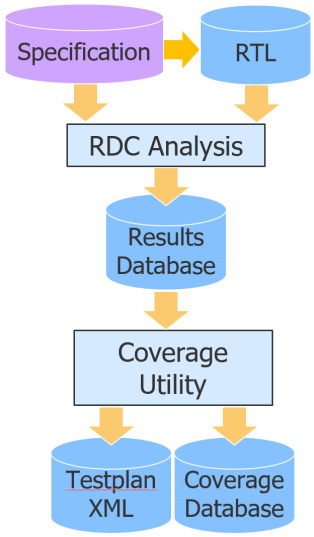

Figure 6. Specification-driven RDC flow.

A. RDC Design Requirements Specification and Verification Plan

During step 1, designers specify the design properties and RDC synchronization methods. Here are some examples of the property types and commands required. In this design, both reset ordering and clock isolation methods were used to synchronize RDC paths.

i. Reset Specification

netlist reset hrst –async –active_low
netlist reset srst –async –active_high

This command specifies the types of reset sources in the design. For example, 'hrst' is an asynchronous reset which is asserted when it is active low and srst is an asynchronous reset which is asserted active high.

ii. Clock Specification

netlist clock clock clk1

This command specifies the clock inputs to the design. For example, 'clk1' is an input clock pin to the design under test.

iii. Reset Ordering Specification

resetcheck order assert –from srst –to hrst

This command specifies the assertion sequence of the asynchronous reset sources in the design. For example, asynchronous reset domain 'srst' is asserted before asynchronous reset domain 'hrst'. This implies that any RDC crossing from source reset domain 'hrst' to receiving reset domain 'srst' will not have any metastability issue due to RDC because the receiving reset domain is asserted before source reset domain. Such crossings are considered ordered RDC crossings and not reported as issues.

B. RDC Design and Verification

During the design process, the clock isolation enable structures were implemented per the specification. Once the clock isolation structures were implemented, the designers are able to specify the reset isolation enable signals:

resetcheck isolation clockgate enable –rx_clock clk1 –rx_reset hrst

This command allows the designer to specify the clock-gate isolation enable signals and the associated destination clock domain and/or reset domain for which they can be used. In the above command example, 'enable' is an isolation signal in the design which is expected to isolate RDC crossings only in the destination domains – clock 'clk1' and reset 'hrst'.

An industry tool was used to perform RDC verification and the results are shown in Table 1. The clock isolation structures were detected during the RDC analysis. This design also shows how reset domain crossing paths may overlap with clock domain crossings (CDC), so traditional CDC synchronization structures may address both RDC and CDC issues.

TABLE I
SUMMARY OF RDC RESULTS

	Design 1	Design 2
Number of Asynchronous RDC issues in same clock domain	185	15519
Number of Asynchronous RDC issues in multiple clock domains	318	584
Number of Reset domain crossings with Synchronizers	0	258
Number of Isolated Reset domain crossings through clock gating	503	7293

C. RDC Results Progress Tracking and Completion Metrics

The last step in this methodology is to make sure that all requirements have been verified in order to achieve verification closure. A coverage utility was used to generate a XML testplan and an UCDB coverage database from the RDC results database. The coverage results were annotated to the testplan and the results were viewed in a coverage viewer as shown in Figure 7 and Figure 8. A 100% coverage on any testplan item indicates the requirement has been completed, but any requirement without 100% coverage indicates that additional work is required by the project team.

It is critical that the testplan tracks requirements for both the setup and RDC analysis. Both design, clock, and reset setup requirements must be complete to generate accurate RDC results. When setup requirements are not 100% complete (Figure 7), designers must complete the setup requirements before analyzing the RDC results. In a similar manner, the testplan will also demonstrate RDC analysis completeness (Figure 8) and tape-out readiness for the design.

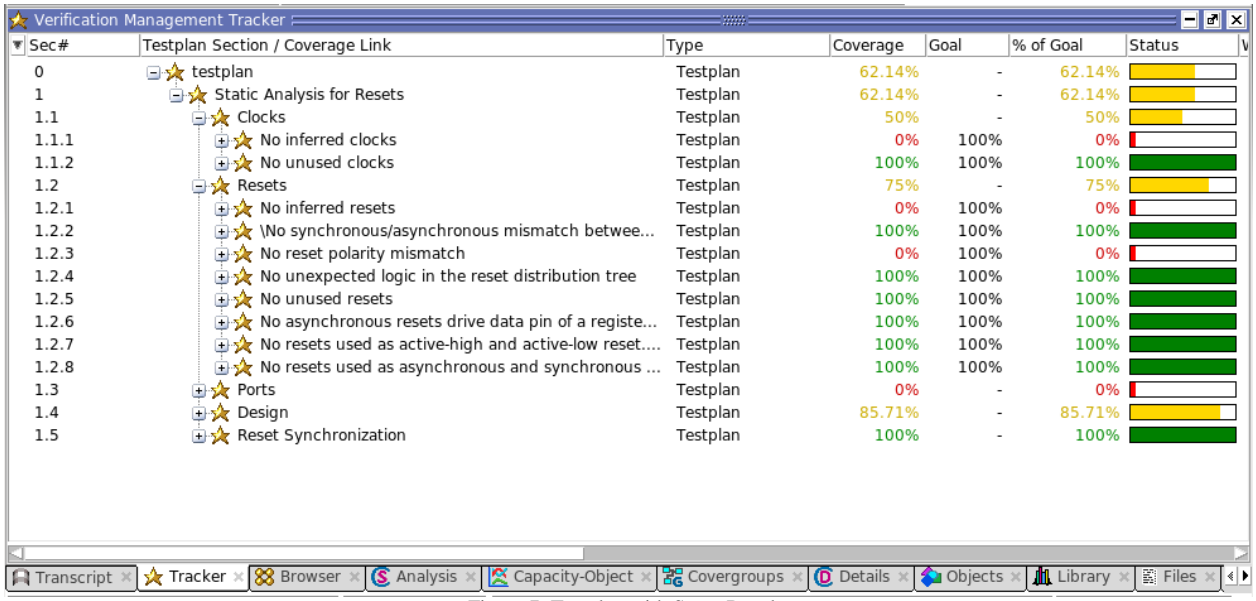

Figure 7. Testplan with Setup Results

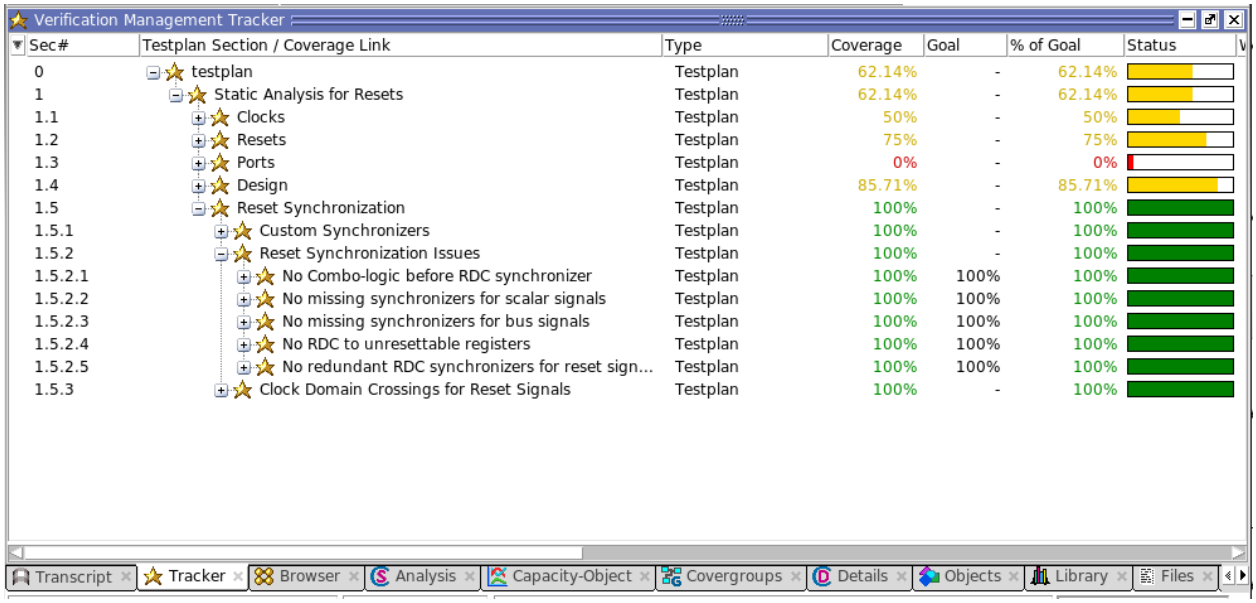

Figure 8. Testplan with RDC Results

IV. SUMMARY

The specification-driven methodology described in this paper is a systematic and repeatable solution for the design and verification of RDC paths. The incorporation of RDC design requirements into this methodology improves the design and verification process and creates a systematic approach for designing and verifying the reset architecture. Finally, the incorporation of coverage metrics with an integrated requirements management capability allows the correlation between the design requirements and verification results and enables closure for the verification process. The tracking of design and verification requirements and verification closure metrics are important for safety-critical applications such as the DO-254 flight safety specification and the ISO 26262 automotive safety standard.

REFERENCES

[1] Chris Kwok, Priya Viswanathan, Ping Yeung, "Addressing the Challenges of Reset Verification in SoC Designs", DVCon US, 2015.

[2] Yossi Mirsky, "Comprehensive and Automated Static Tool Based Strategies for the Detection and Resolution of Reset Do-main Crossings", DVCon US, 2016.

[3] Kurt Takara, Chris Kwok, Dominic Lucido, "Coverage-driven Clock Domain Crossing (CDC) Verification Enables DO-254 Safety-critical Designs", MAPLD, May 24, 2017.

[4] Ping Yeung, Erich Marschner, "Multi-Domain Verification: When Clock, Power and Reset Domains Collide", DVCon US, 2015.