

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

Priya Viswanathan, Kurt Takara, Chris Kwok, Islam Ahmed  
Mentor, a Siemens Business, Fremont, CA

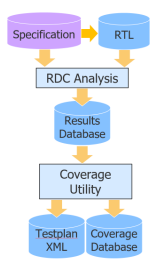
## Abstract

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.

## Specification Driven Methodology

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

- Benefits:**
- **Integrated requirements management capability to enable requirements traceability**
  - **Systematic approach for designing and verifying the reset architecture**
  - **Correlation between the design requirements and verification efforts to enable closure for the verification process**



## Step 1: RDC Design Requirements Specification and Plan

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.

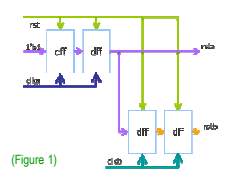
Reset Specification Example:  
netlist reset hrst -async -active\_low  
netlist reset srst -async -active\_high  
Clock Specification Example:  
netlist clock clock clk1  
Reset Ordering Specification Example:  
resetcheck order assert -from srst -to hrst

## Step 2: RDC Design and Verification

Specify the design techniques for RDC synchronization and reset distribution. Some common techniques include reset staging methods and RDC synchronization.

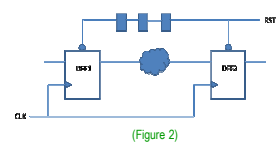
### Staged Reset Generation

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.

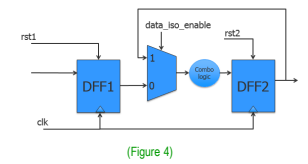
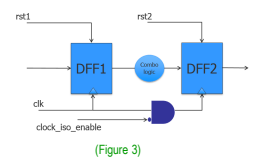


### Reset Sequencing

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.



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.



## Step 3: RDC Results Tracking & 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 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 complete and incomplete RDC verification tasks and 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			100	0
1.5.1	Custom Synchronizers	Custom Synchronizers Definition			100	0
1.5.1.1	No redundant Custom Synchronizers	Ensures that there are no redundant custom synchronizers in the design	Report:ATC_RDC_RESULTS/SYNCHRONIZERS_REDUNDANT	Bin	2	100
1.5.1.2	Custom Synchronizers (Timing Connected)	Ensures that all the custom synchronizers in the design are properly connected	Report:ATC_RDC_RESULTS/SYNCHRONIZERS_TIMING_CONNECTED	Bin	2	100
1.5.1.3	No Combino Logic before Custom Synchronizers	Ensures that there are no combino logic driving custom sync nodes	Report:ATC_RDC_RESULTS/SYNCHRONIZERS_COMBINO_LOGIC	Bin	2	100
1.5.2	Reset Synchronization Issues	Reset Synchronization Issues Section			100	0
1.5.2.1	No combino logic before RDC synchronizer	Ensures that there are no combino logic driving custom sync nodes	Report:ATC_RDC_RESULTS/SYNCHRONIZERS_COMBINO_LOGIC_BEFORE_RDC_SYNC	Bin	2	100

(Figure 5)

## Real World Results

This methodology was applied on both an IP block and a digital networking SoC design (Figure 6). Both designs used clock gate isolation methods in the RDC structures. These designs also show how reset domain crossing paths may overlap with CDC paths, so traditional CDC synchronization structures may address both RDC and CDC issues.

In the specification-driven RDC flow, 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 coverage bins in the UCDB and a functional coverage viewer is used to view the UCDB coverage database.

	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

(Figure 6)

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. 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.

(Figure 7)

## Summary

This specification-driven methodology 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.