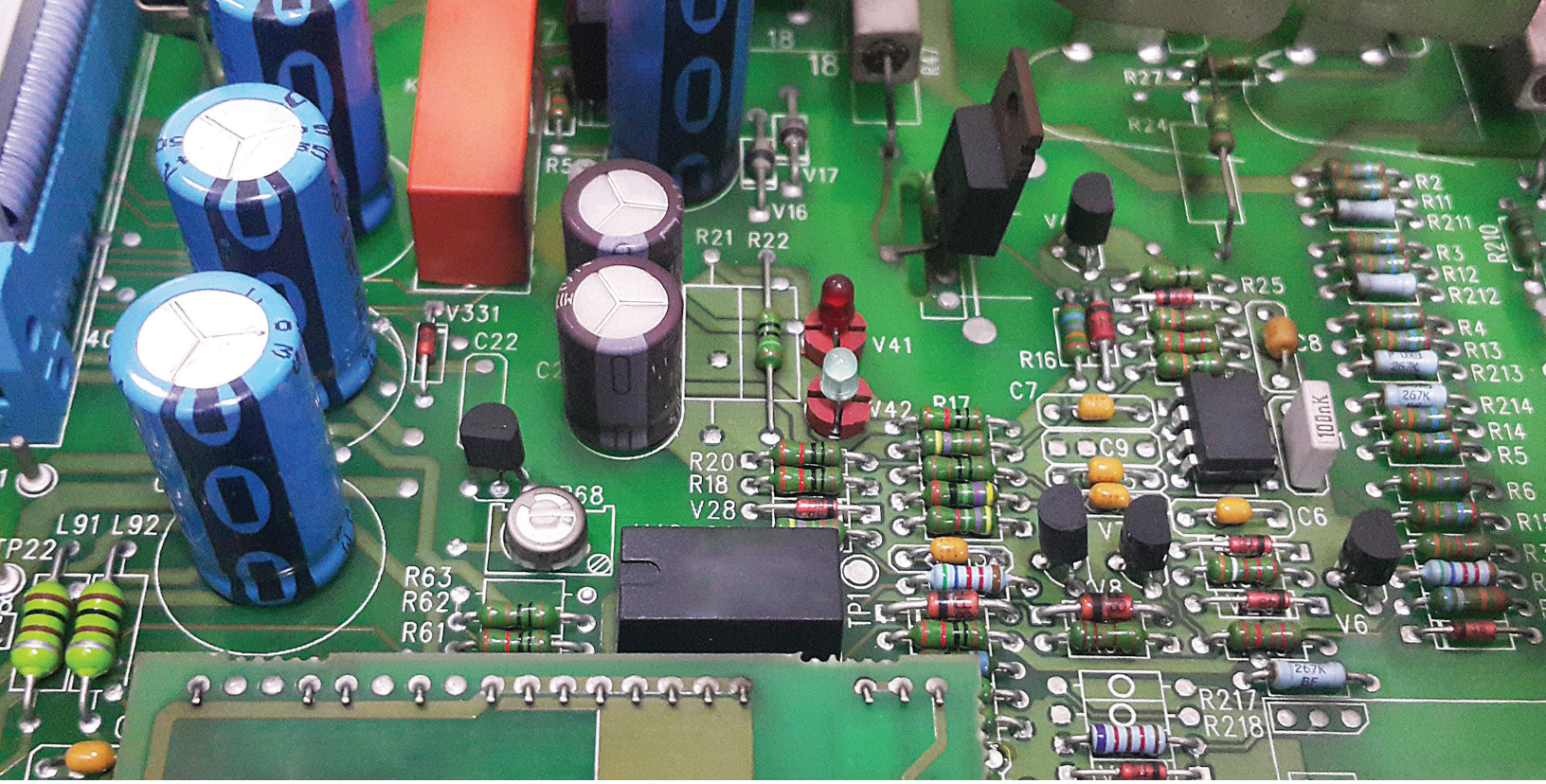

DIGITAL INDUSTRIES SOFTWARE

Reset domain crossing design verification closure using advanced data analytics techniques

Executive summary

Complex reset mechanisms are embedded in advanced SoCs to meet low-power and high-performance requirements. Multiple reset domains in a design can cause 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. The data generated by the RDC verification tools is very large, consisting of millions of RDC paths. The analysis of this data is a very time consuming and challenging task for design and verification engineers that often involves many iterations. In this paper we will highlight how to automate RDC results analysis using data processing and data analytics techniques to provide faster RDC verification closure.

Reetika and Sulabh Kumar Khare
Siemens EDA

SIEMENS

Introduction

As the complexity of designs is increasing every year, doubling every 20 months, verification of designs also becomes more difficult. In today's design landscape, the utilization of EDA tools for clock domain crossing (CDC) detection has evolved into a common practice for ensuring design quality. With the rise in the number of asynchronous resets in designs, a more recent addition to this field is the validation of RDCs. It has become essential to verify asynchronous data stability between different reset domains using various RDC static verification tools because data transfers between different asynchronous reset domains can lead to metastability issues and unpredictable behavior, much like the errors encountered in clock domain crossings.

Figure 1 illustrates a simple RDC problem between two flops having different asynchronous reset domains. The asynchronous assertion of the rst1 signal immediately changes the output of FF1 flop to its assertion value. Since the assertion is asynchronous to clock clk1, the output of the FF1 flop can change near the active clock edge of the FF2 flop, which can violate the set-up hold timing constraints for flop FF2 and, therefore, FF2 flop can go into a metastable state. This metastable state can cause unpredictable values to be propagated to down-stream logic and prevent a design from functioning normally.

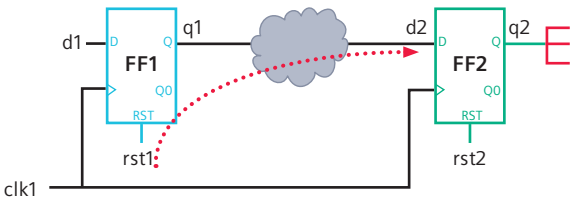

Figure 1. RDC between FF1 flop and FF2 flop from asynchronous reset rst1 to asynchronous reset rst2.

One of the biggest challenges for ASIC verification is the large volume of results generated by RDC verification analysis tools. The large volume of results data must be analyzed by design and verification engineers, so a large set up and review effort is invested, and the huge amount of data for review increases the likelihood that design bugs may be overlooked or missed, which can lead to silicon respins.

Currently RDC results analysis is done manually, which requires multiple design runs, occupies a significant amount of time, and is error prone. As newly written RTL for designs have no initial setup constraints, an RDC run reports lots of RDCs, and it takes a lot of effort and many iterations to analyze these crossings and understand the common root cause for all such paths and then apply some constraints and rerun the verification tools to resolve the violations, which ultimately affects the sign-off schedule.

Most of the RDC paths are due to some common design or setup issue, such as resets that were expected to be in a synchronous reset domain are defined in two different reset domains, resulting in a large number of RDC violations.

With the guidance of advanced data analysis techniques, we identify and suggest constraints that are observed due to a common set of design issues or setup problems. Since the RDC verification process is dependent on the quality of the design setup, we suggest the constraints like stable signal declarations, reset orderings, reset domains, specification directives, isolation signals, and constant declarations, etc. that can be used to narrow down the RDC analysis results. This eventually reduces the initial number of RDC violations and presents the actual issues for the verification engineers to analyze and fix, resulting in early verification closure.

RDC verification using data mining

The progressive application of supervised data processing and data analytics techniques helps in acceleration of RDC verification closure by analyzing RDC results to recognize patterns and suggesting setup related constraints.

Due to copyright concerns, we will not get into details of algorithms of data processing and analysis; however, we will examine the recommendations generated, and assess their impact on reducing violations, thereby evaluating the effectiveness of these recommendations. Some of the recommended suggestions are as follows.

Reset ordering: There could be cases in which the receiver (Rx) flop's reset is asserted before the reset of transmitter (Tx) flop. If the receiver's domain flop is reset before the asynchronous event in the transmitter's domain, metastability cannot propagate in the design. Figure 2 illustrates a RDC problem between two flops having different asynchronous reset domains. If reset RST2 is asserted before reset RST1, then the path from TX to RX would be safe (an RDC path through the transmitter to the receiver register in which metastability cannot propagate is deemed to be a safe path). So, if there is no ordering constraint defined between the resets during tool set up by design and verification engineers, then it could lead to reporting several RDC violations for a common cause.

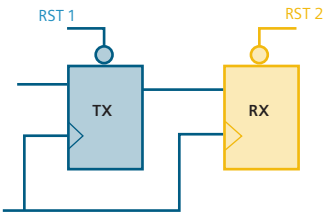

Figure 2. RDC between TX flop and RX flop from asynchronous reset rst1 to asynchronous reset rst2.

Synchronous reset domains: A RDC problem is caused when the reset of the source register and the reset of the destination register are asynchronous. Due to the asynchronous nature of resets, Tx register can generate an asynchronous signal that can violate the setup and hold time requirements for the Rx register, which can cause the Rx to go to a metastable state. While setting up the tool, if the user does not group the reset signals in the synchronous reset domain, then many unsynchronized crossings will be reported between them. Although contemporary RDC static verification tools have the capability to detect and analyze the resets within a design, the initial unconstrained design is often the starting point, the RDC analysis of which can result in a substantial number of RDC violations. Analyzing the possible cases and doing intelligent evaluation based on multiple factors, including equivalence checking, on the reset expression can significantly reduce the RDC violation count and speed up the whole analysis.

Directive specifications: Directive specification helps to reduce noise and avoid unnecessary waiver usage. For example, there is a chance that the clock of a receiving register is off when the reset of a Tx register is asserted. Since the Rx clock is already off, the Tx reset cannot violate the setup and hold time requirements of the Rx flop, and hence there will not be any occurrence of metastability in the design. If the user neglected to specify this clock-off constraint while setting up the tool, it may lead to unwanted RDC crossings. Also, it is likely that the Tx register output is already at its reset value before its asynchronous reset is asserted. In such cases, there will be no metastability on its Rx register. So it becomes a cause of noisy RDCs

getting reported if the user has not specified valid constraints to not flag RDC violations starting from such flops.

Stable signals: The existence of some signals within the design that are part of two asynchronous reset domains, but which have no chance of creating metastability, can be marked as possible stable signals. If the user did not specify those signals as stable, then the tool infers them as a potential candidate for RDC violations.

Isolation signals: There are ways to isolate the data transfer from one reset domain to another reset domain to block metastability due to an asynchronous reset assertion at Tx in the setup and hold window of the Rx clock domain. Any suggestions around this helps to fine-tune the safe RDCs and flag the real issues where metastability may occur. Figure 3 shows a simple example of this strategy with an isolation signal Iso that anticipates a Tx asynchronous reset activation and preemptively 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.

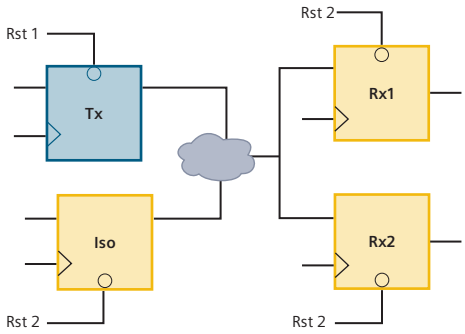

Figure 3. The RDCs from the Tx flop to the Rx1 and Rx2 flops are safe, as the Iso flop preemptively isolates the Rx domain logic.

Safe RDCs with Rx as non-resettable register: The RDCs in which the receiver flop is a non-resettable register (NRR) sometimes does not have a metastability issue if a flop exists at some depth after the NRR flops, having a reset of the same domain as the Tx reset, provided the reset signal is long enough to suppress the metastability. Figure 4 illustrates that the path from flop TX to flop NRR is safe as the flop OUT has the same reset domain as flop TX. If the user does not specify the depth at which the flop with the same reset domain exists, several false RDCs will be reported.

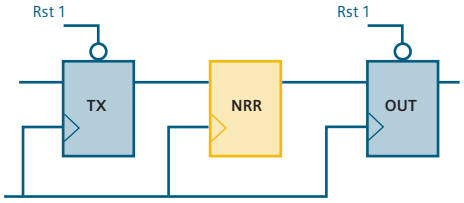

Figure 4. Safe RDC from TX flop to NRR flop.

In addition to the above-mentioned suggestions, there could be many other potential recommendations that the data analysis model could predict and thereby help solve critical RDCs involving non-resettable registers and provide automatic waiver suggestions. By employing these initial suggestions/constraints in the design run, we can eventually resolve the related and correlated violations for a common design issue and thus increase the quality of RDC results. As there is now a significant reduction in the total number of RDC violations, the user is now exposed to the real scenario RDC bugs.

Another important aspect is to provide flexibility for users to analyze those suggestions first that have a higher impact on the RDC results. Generating a report or displaying results in a GUI showing a clear correlation between each suggestion and the associated RDC violations it resolves can be very helpful in qualifying the suggestions before applying them for RDC analysis. This way the invested effort for the analysis and debug would always be in the high impact areas and will converge as the user progresses with analysis.

Refer to figure 5 for a detailed overview of RDC verification using advanced data analysis methods. The four steps are illustrated below.

I. Run the RDC analysis with the design RTL and design constraints.

II. Run the design using the data analysis techniques on the reset and RDC results to generate constraints.

III. Run RDC analysis with the design RTL, design constraints and the generated setup constraints. The data mining and data analytics techniques will fine tune results and prune out redundancies.

IV. Users to review the results.

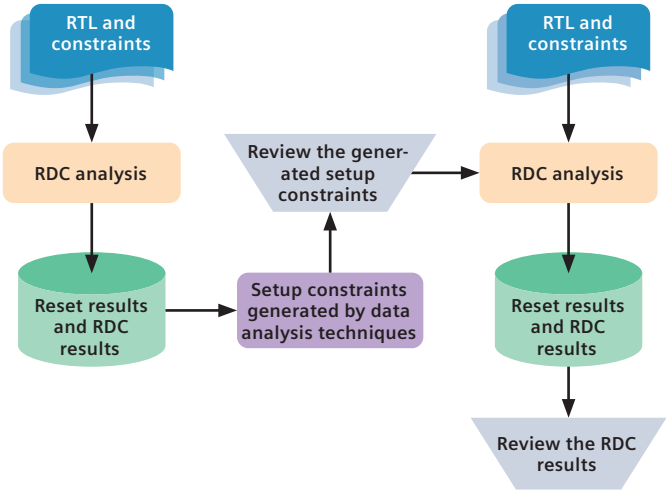

Figure 5. RDC verification using data analysis techniques.

Case study

An experimental trial was conducted on a design to assess how the data analytic techniques enhanced the speed of debugging and the reporting of critical root causes of failures. The design consisted of 263,657 register bits, four latch bits, 40 RAMs, five clock domains, nine reset domains, and 67,000 RDCs that required data isolation, specification constraints, reset ordering, data isolation, and other setup constraints. The reference design had previously undergone RDC analysis with manually crafted constraints, rendering it a perfect candidate for assessing the precision and efficiency of data mining and analytics support.

In a run using data analytics method, with a threshold set at a value of 200 (the term "threshold" indicates that a specific constraint will be recommended only if it affects at least the minimum number of specified paths), and with minimal setup constraints (i.e., only some constraints related to reset and clock domains were defined), the RDC verification tool reported around 8000 RDC paths between different asynchronous reset domains. Using data mining and data analytics techniques, a consolidated report was generated, recommending several constraints, along with showing the number of RDC violations impacted by the suggested constraints.

After applying the suggested constraints, the count of RDC violations was reduced from 8000 to 2732. The count of filtered paths, for a particular root case, with a run using manually applied constraints and using the constraints suggested in the previous run was nearly same (see table 1). The minor difference in the results could be due to the threshold path limit being set to 200, but overall, the results were good as more than 60% of RDC violations were resolved by applying the suggested constraints.

	Number of paths using constraints generated manually	Number of paths using constraints generated by data analytics
Ordered RDC paths	2764	1563
RDC paths with clock off specifications	58	62
Data isolated RDC paths	101	79
RDC paths with safe fanout	5	5
Clock isolated RDC paths	2095	1973
RDC paths with Tx at reset value specifications	1978	1594

Table 1. Comparison of RDC paths with the manually generated constraints and using the constraints generated by data analytics technics.

Results

In RDC design verification, design and verification engineers face a major challenge in fixing the most common RDC problems related to incorrect or missing constraints for reset ordering and reset grouping. Typically, there can be hundreds of RDC paths that may have a common root cause, and if we are able to get some initial information about possible common causes of a number of RDC violations, this data will help users to quickly solve a lot of issues and hence save a lot of time and effort.

The application of advanced data analytic techniques results in a major reduction of unsynchronized RDC crossings detected in a design. We observed that from the RDC results obtained after applying the potential constraints and suggestions, we were able to reduce the RDC verification closure time for the IP/subsystem design from around ten days down to less than four days, as up to 60% of violations on average were getting resolved (refer to table 1 and 2). The reduction in RDC violations demonstrates the efficacy of the data analysis features in analyzing the data and providing setup guidance without designer intervention.

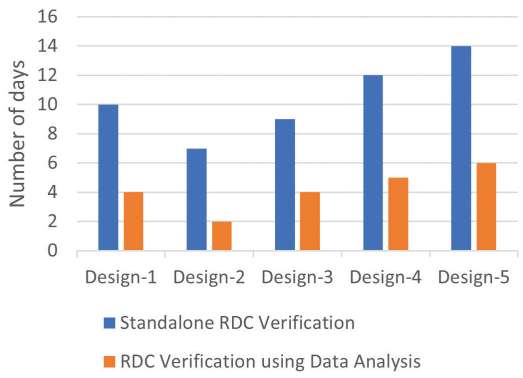

Figure 6. Reduction in RDC verification turnaround time due to RDC data analysis.

Design	Count of RDC violations without applying potential root cause	Count of potential constraint suggestions	RDC violations count after applying constraints	Percentage decrease
Design 1	23045	73	9392	59.2
Design 2	44002	35	17258	60.7 %
Design 3	34678	43	12892	62.8 %
Design 4	14760	28	4473	69.6 %
Design 5	72307	54	27999	61.2 %

Table 2. Reduction in RDC violations due to RDC data analysis.

Conclusion

The manual verification of RDC results takes a lot of time and effort and there is a high probability that design bugs may be overlooked. The proposed solution of RDC verification by applying constraints suggested by advanced data analytic techniques is intended to reduce the manual set up and review effort by design and verification engineers, improve the quality of results, and avoid the possibility of neglecting to specify constraints that lead to design bugs.

References

1. Akanksha Gupta, Ashish Hari, Anwesha Choudhary. “Systematic methodology to solve reset challenges in automotive SoCs”, DVCon Europe, Dec 2019.

2. Yossi Mirsky. “Comprehensive and automated static tool based strategies for the detection and resolution of reset domain crossings”, Intel Corporation.

3. Mark Handover, Siemens EDA. “Reset your reset domain crossing (RDC) verification with machine learning”, DVCon Europe, Dec 2022.

Siemens Digital Industries Software

Americas: 1 800 498 5351

EMEA: 00 800 70002222

Asia-Pacific: 001 800 03061910

For additional numbers, click here.

Siemens Digital Industries Software helps organizations of all sizes digitally transform using software, hardware and services from the Siemens Xcelerator business platform. Siemens' software and the comprehensive digital twin enable companies to optimize their design, engineering and manufacturing processes to turn today's ideas into the sustainable products of the future. From chips to entire systems, from product to process, across all industries, Siemens Digital Industries Software – Accelerating transformation.

siemens.com/software

86184-D2 9/24 H