International Conference on Advanced Computing, Security, and Intelligent Systems (ICACSIS-2025)
ISSN: 3107-9458| ICACSIS-2025 - Conference Proceeding
DOI: https://doi.org/10.64137/3107-9458/ICACSIS-113

*Original Article*

# CDC and RDC Analysis Methodologies for Timing-Critical FPGA Systems

**Venkatakrishna Brahmam Pogadadanda**
*Senior Silicon Design Engineer at AMD Xilinx, India.*

***Abstract:*** *Field Programmable Gate Arrays (FPGAs) are widely used in timing-sensitive applications such as aerospace, automotive systems, telecommunication, medical devices and high-speed data processing due to their versatility, parallel processing capabilities and real-time performance. Modern FPGA designs are becoming ever more complex and this poses serious verification challenges, notably in the areas of Clock Domain Crossing (CDC) and Reset Domain Crossing (RDC). CDC problems happen whenever signals are passed between asynchronous clock domains. They often lead to metastability , data corruption , synchronization failures and unpredictable system behavior . Asynchronous reset signals interacting across different reset domains can lead to RDC problems, such as functional mismatches, incorrect initializations or timing violations. These issues are especially relevant in safety- and mission-critical systems where reliability and deterministic behavior are important. In this paper, we present comprehensive CDC and RDC analysis methods to improve the resilience and reliability of timing-critical FPGA systems. The proposed solution combines architectural analysis, synchronization checking, timing constraint analysis and automatic detection methods to catch possible crossover violations early in the design cycle. Also discussed are state-of-the-art verification approaches such as synchronizer modeling, reset tree analysis and metastability mitigation schemes to improve design integrity. The paper describes practical implementation methodologies and common industry verification techniques used in modern FPGA development environments. A full case study is presented to show the effectiveness of the proposed methodologies for detecting and fixing CDC and RDC violations in a complex FPGA architecture. The expected effects include better system stability, fewer functional failures, more timing reliability and faster verification closure. This research contributes to the development of reliable verification procedures for next-generation timing-critical FPGA-based systems with strict performance and safety demands.*

***Keywords:*** *FPGA, Clock Domain Crossing (CDC), Reset Domain Crossing (RDC), Timing Analysis, Metastability, Synchronization, Static Verification, Timing-Critical Systems.*

## 1. INTRODUCTION

### *1.1. OVERVIEW OF TIMING-CRITICAL FPGA SYSTEMS*

Field Programmable Gate Arrays (FPGAs) are an important part of modern digital system architecture enabled by their flexibility, reconfigurability, and ability of high-speed parallel processing. FPGA technology has evolved from a simple prototyping tool to a key platform for implementing complex real-time systems in many application domains, such as aerospace, automotive, telecommunications, healthcare, and high-performance computing. FPGAs are widely used in aircraft applications i.e. radar systems, satellite communication, avionics control and navigation systems where high dependability and predictable timing are the key requirements. In the automotive sector, the use of FPGA-based designs is growing in advanced driver assistance systems (ADAS), autonomous vehicle controllers, and in-car communication networks for real-time analysis of large amounts of sensor data. FPGAs are used in medical technology such as imaging systems, robotic surgery platforms and patient monitoring equipment as they can conduct time-critical operations with minimum delay. FPGAs are widely adopted in telecommunication and high-speed network applications such as packet processing, signal modulation, and 5G infrastructure, as they can process many high-frequency data streams in parallel.

With the evolution of FPGA architectures, modern systems typically have several clock domains and independent reset networks to increase throughput and optimize resource utilization. These many clock and reset configurations improve performance but also

greatly increase the complexity of the system. For timing critical FPGA systems, it is important that different functional blocks are tightly synchronized for accurate and predictable performance. Any deviation in timing behavior causes functional errors, data corruption, or system instability. This has made ensuring reliable timing performance a major design and verification consideration for FPGAs. Designers should analyze carefully the interaction between asynchronous clock and reset domains for robustness of the system, particularly in applications where safety, dependability and real-time performance are critical.

### *1.2. CHALLENGES IN CDC AND RDC VERIFICATION*

Verification of Clock Domain Crossing (CDC) and Reset Domain Crossing (RDC) has become a major issue in the design of modern FPGA systems. When signals are sent across different clock domains that are running at different frequencies or are out of phase, this causes Clock Domain Crossing (CDC) problems. Since the clock domains are asynchronous, we expect the flip-flop to go metastable with some probability rather than some definite logic value. The metastability can spread through the system and result in random behavior , timing problems , or complete loss of functionality . In high-speed FPGA designs, a little synchronization error may have a significant impact on the overall performance and reliability of the system.

A big issue with CDC is that data might get corrupt if the synchronization procedures are not good. If the signals crossing clock domains are not synchronized correctly, the receiving domain could read incorrect or incomplete data. This may lead to loss of packets, false calculations or malfunction of the communication links. Verification becomes challenging in large FPGA systems with several clock regions that interact at different frequencies and timing constraints. These difficulties may not be found by conventional simulation tools, as metastability and asynchronous timing circumstances may happen seldom and unpredictably.

In many reset domains the reset behavior is asynchronous, which further complicates the testing of RDC. Reset recovery and removal violations can occur when the reset signal is deasserted close to an active clock edge, causing the registers to reach contradictory states. Furthermore, asynchronous resets can cause glitches or spurious transitions that can spread across the architecture and affect normal operation. As FPGA designs become larger and more heterogeneous, it becomes more difficult to detect RDC-related problems manually. High dependability is required for safety-critical systems, including aerospace control systems, automotive electronics and medical devices, and hence verification of CDC and RDC is vital to prevent catastrophic failures and ensure stable system operation.

### *1.3. PROBLEM STATEMENT*

Traditional timing verification in FPGA design is primarily concerned with synchronous timing analysis, where signals are synchronized to a single clock domain and display predictable timing characteristics. These techniques are applicable to synchronous circuits, but they often do not provide sufficient coverage for asynchronous interactions between different clock and reset domains. As the modern FPGA systems are progressively adopting multi-clock architectures to improve the performance, the complexity of asynchronous communication is still dramatically expanding.

Existing verification approaches have one major limitation: the difficulty of finding latent CDC and RDC problems during simulation. Many of the asynchronous timing problems, such as metastability and synchronization errors, may not always be observable in functional simulation because they are timing-dependent and occur seldom when signals change. Consequently, severe design flaws may not be discovered until the hardware implementation stage, which leads to costly debugging processes and even system failures.

Moreover, high-frequency FPGA systems require more comprehensive and scalable verification methodologies that can deal with complex clocking designs and asynchronous reset interactions. Existing verification approaches are often not sufficiently automated, and do not provide complete analytical coverage for time-critical applications. Therefore, there is a strong need of advanced CDC and RDC analysis techniques that may improve the accuracy of verification, improve the reliability of the system and enable robust FPGA design implementation.

### *1.4. MOTIVATION*

The fast evolution of high-speed FPGA systems has created a huge demand for more reliable and efficient timing verification methods. Modern FPGA devices can implement complex applications with a large number of processing units , high-bandwidth communication interfaces and real-time data collection . As the system performance requirement increases, the designers need to use

many clock and reset domains to increase the functionality and throughput. However, such architectural improvements also incur non-trivial verification problems for asynchronous temporal behavior.

One of the main motivations for this work is the increasing interest in functional safety and fault tolerance in critical applications. Industries such as aerospace, automotive, healthcare and defense want systems that perform reliably under strict safety rules. Failure can have serious results such as data loss, failure of equipment or safety hazards. Metastability, synchronization issues and improper reset handling can lead to these failures. Therefore, the need for advanced CDC and RDC verification procedures that can ensure continuous and dependable system functionality is growing.

The development of automated verification techniques has fueled research in this area. Manual inspection of CDC and RDC paths in large-scale FPGA designs is labor-intensive, mistake prone and often impractical. Automated verification techniques can reduce design complexity, speed up debugging and reduce hardware redesign requirements. In addition, enhanced CDC and RDC verification leads to faster timing closure, reduced development cost, and improved overall system resilience. Taken together, these problems imply the need for comprehensive techniques to examine asynchronous interactions in timing-critical FPGA systems.

## 2. LITERATURE REVIEW

### *2.1. FUNDAMENTALS OF CDC*

Clock domain crossover (CDC) is the transfer of signals between distinct clock domains that may have different frequencies, phases or timing relationships in a digital system. Clock domain crossing (CDC) is a major challenge in current FPGA systems. Multiple clock domains are typically used in high-performance systems to boost throughput and maximize resource use. To communicate reliably between these asynchronous domains, proper synchronization is required; otherwise, the system could behave erratically.

CDC is connected to the notion of metastability. Metastability occurs when a flip-flop samples an input signal near the active clock edge. The flip-flop output becomes an ill-defined intermediate state until it stabilizes to a definite logic value. However, metastability can never be abolished. Proper synchronization techniques can lessen its impacts. under case, the design is under meta-stable situations; data can be latched erroneously, timing can be violated or the system can fail. This is particularly the case for timing-critical FPGA applications.

To address CDC concerns, the literature proposes numerous synchronization strategies. Usually synchronizers are two-stage or multi-stage for single-bit signal transfers . This is because the synchronizer gives adequate time for metastability to settle down before the signal crosses over to the receiving domain.Handshake protocols and pulse synchronization techniques usually guarantee the reliable transfer of events. More advanced processes are needed to preserve data integrity and to avoid bit-skew problems for multi-bit signals . These include asynchronous FIFOs, Gray code counters and bus synchronization. Researchers say that the choice of synchronization approach is determined by aspects such as clock frequency, latency requirements and system complexity. Therefore, efficient CDC design strategies are important to increase the timing reliability and the overall robustness of FPGA systems.

### *2.2. FUNDAMENTALS OF RDC*

Reset Domain Crossing (RDC) problem concerns the interplay of signals between logic units working in various reset domains in a digital system. RDC difficulties are much like CDC problems in that they occur when reset signals are not properly synchronized throughout the various portions of an FPGA system. Modern FPGA systems usually employ several reset networks to achieve modular architecture, power efficiency, and independent startup of subsystems. However, bad reset management can cause timing issues and operational difficulties.

Reset synchronization procedures are used to safely assert and deassert reset signals with regard to the destination clock domain. Typically there are two major reset types in FPGA systems, asynchronous resets and synchronous resets. Async resets allow registers to be cleared instantaneously , regardless of the clock signal . They are important during power-up or in urgent situations. However, asynchronous resets may cause metastability and glitches related difficulties when they are deasserted near active clock edges. Synchronous resets are clocked ( ie happen at the same time as the clock ) , this reduces timing uncertainty , but may increase reset delay .

Another essential part of RDC verification is related to the review of reset recovery and removal timing. Recovery time is the minimum time between the deassertion of the reset and the next clock edge. Removal time is the minimum time following a clock edge to the assertion of the reset. If these timing constraints are breached, the sequential elements may be left in unanticipated states. Therefore, a correct RDC analysis is required to ensure the deterministic behavior and the continuous performance of the FPGA system.

### *2.3. EXISTING CDC ANALYSIS TECHNIQUES*

There are several CDC analysis approaches developed for detection and alleviation of asynchronous timing problems in FPGA devices. A popular approach is static CDC analysis which evaluates the structural relationship of clock domains without the need of dynamic simulation. Static analysis tools automatically detect crossovers between asynchronous clock domains and identify potential synchronization violations such as unsynchronized transfer of signals, absence of synchronizers and incorrect handshake implementation. Static analysis, which is independent of simulation vectors, provides high coverage and is well suited for large scale FPGA systems.

In addition, CDC verification is complemented by structural analysis methods of synchronization structures and signal transmission paths in the design netlist. These techniques classify clock crossings by synchronization protocols and detect potentially dangerous communication paths that might lead to metastability or data corruption. Structural analysis enables designers to verify correct implementation of common synchronization structures, such as two-flop synchronizers and asynchronous FIFOs.

The formal verification methods have received a lot of attention due to their importance because they can mathematically verify the correctness of CDC properties under all possible operational scenarios. Formal methods use assertion-based verification and state space exploration techniques to prove synchronization accuracy and to detect hidden corner-case errors that traditional simulation may miss. Formal verification gives excellent accuracy but may be computationally intensive for very large FPGA systems.

Simulation-based CDC verification remains a major complementary approach. Functional simulations allow designers to investigate dynamic signal behavior and test synchronization logic in realistic operational situations. Advanced simulation methodology can use metastability modeling, stochastic clock generation and timing-aware simulation techniques to improve failure detection. However, simulation alone often gives limited coverage. Asynchronous failures may occur only under rare temporal conditions. Therefore, for a complete CDC validation, modern verification approaches typically involve static analysis, structural verification, formal verification and simulation-based methods.

### *2.4. EXISTING RDC VERIFICATION TECHNIQUES*

The RDC verification techniques are based on the behavior analysis of reset signals across many reset domains to verify the reliable system initialization. Reset tree analysis is one of the main methods used for RDC verification. Reset tree analysis assesses the reset signal distribution network of the reset in the FPGA design and detects mismatches such as insufficient fan-out configurations, non-synchronized reset paths, and conflicting reset dependencies. This strategy allows the designers to ensure that all the later elements get stable and well timed reset signals.

Recent work has studied formal reset verification mechanisms in great detail. The formal procedures are based on the mathematical models and the assertion-based qualities to verify reset sequencing, reset propagation and reset release conditions. These approaches are quite useful for the detection of hidden RDC flaws that may not be surfaced in normal simulation. Formal verification can determine whether all the logic blocks can successfully recover from reset conditions for a variety of operational scenarios.

RDC structural verification is a common methodology in the context of industrial FPGA verification flows. RDC approaches for structure analyze relationships between reset domains and find dangerous reset crossings that might cause metastability, inconsistent initialization, or functional instability. The tools automatically detect missing reset synchronizers, asynchronous reset release issues and incorrect reset interactions. The combination of structural checking and formal and temporal analysis techniques significantly improves the reliability of RDC verification for timing-driven FPGA designs.

**Table 1. Literature Review on CDC and RDC Verification Techniques in FPGA Systems**

| Author(s) & Year | Title | Methodology / Technique | Key Findings | Limitations |
|---|---|---|---|---|
| Vatandoost et al. (2019) | Comparison of CDC Bottle Bioassay with WHO Standard Method | Comparative CDC methodology | Demonstrated effectiveness of CDC-based assessment methods | Not related to FPGA CDC verification directly |
| Kasim, Gupta & Jebin (2020) | Methodology for Detecting Glitch on Clock, Reset and CDC Path | Glitch detection and CDC path verification | Improved detection of clock, reset and CDC-related glitches | Limited focus on integrated RDC verification |
| Viswanathan et al. (2018) | A Specification-Driven Methodology for the Design and Verification of Reset Domain Crossing Logic | RDC verification using specification-driven approach | Enhanced reset-domain verification accuracy | Focused mainly on RDC logic |
| Park et al. (2007) | A Decision Processing Algorithm for CDC Location Under Minimum Cost SCM Network | CDC decision-processing algorithm | Optimized CDC placement and processing decisions | Application-specific methodology |
| Mehrasa et al. (2016) | Dynamic Model, Control and Stability Analysis of MMC in HVDC Transmission Systems | Stability and control analysis | Demonstrated importance of synchronization in complex systems | Not specifically targeted at FPGA CDC/RDC |
| Wu, Zhu & Wu (2020) | A Novel Rotor Initial Position Detection Method Utilizing DC-Link Voltage Sensor | Signal synchronization methodology | Improved detection reliability in real-time systems | Limited relevance to FPGA verification |
| Existing Static CDC Analysis Methods | Structural CDC Verification | Static analysis of clock-domain crossings | High coverage and automatic detection of synchronization issues | Cannot fully capture dynamic behavior |
| Formal CDC Verification Approaches | Assertion-Based CDC Verification | Formal verification and state-space exploration | Detects hidden corner-case synchronization failures | Computationally expensive for large designs |
| Simulation-Based CDC Verification | Functional Simulation Techniques | Timing-aware simulation and metastability modeling | Evaluates real-time dynamic behavior | Limited coverage of rare asynchronous failures |
| Reset Tree Analysis Techniques | RDC Structural Verification | Reset network and dependency analysis | Detects reset distribution and synchronization issues | Requires comprehensive reset modeling |
| Formal RDC Verification Methods | Mathematical Verification of Reset Behavior | Formal property checking for reset propagation and sequencing | Identifies hidden reset-domain crossing violations | High computational complexity |
| Structural RDC Verification Approaches | Automated RDC Structural Analysis | Detection of missing reset synchronizers and reset-release issues | Improves reset reliability in FPGA systems | Requires accurate structural modeling |

## 3. PROPOSED METHODOLOGY

### *3.1. PROPOSED CDC ANALYSIS FRAMEWORK*

The proposed Clock Domain Crossing (CDC) analysis paradigm provides a comprehensive and systematic approach to detect, analyze and mitigate asynchronous timing issues in timing critical FPGA systems. The framework stresses the identification of critical signal crossings across clock domains with the assurance of reliable synchronization and the maintenance of timing integrity in high-frequency designs.

The primary step of the proposed system is to identify the clock domains. Modern FPGA systems sometimes include multiple different clock domains for speed and parallel processing operations. Hence, it is important to identify all the clock sources, derived clocks and asynchronous clock interactions accurately. The system extracts clock data from design constraints, netlists and timing reports to identify synchronous and asynchronous clock domains. This phase also identifies generated clocks, phase-shifted clocks and gated clock setups that can affect CDC behavior.

The second stage is devoted to the recognition of synchronizers. Synchronization structures, like two-flop synchronizers, multi-stage synchronizers, handshake circuits and asynchronous FIFOs, are automatically detected by structural pattern recognition algorithms. The methodology evaluates the installation of proper synchronization mechanisms for each crossing path and detects risky or insufficient synchronization structures. Such automatic analysis can decrease manual debugging and boost verification efficiency.

The framework then classifies signal crossings into single-bit control paths, multi-bit data channels, pulse transfers and bus interfaces through data path classification. Synchronization requirements are implemented according to the nature and time characteristics of the signal. For single bit control signals, synchronizer chains are usually employed for validation, whereas for multi-bit data transfer techniques such as Gray coding or FIFO-based communication are used to allow synchronization at the receiver side.

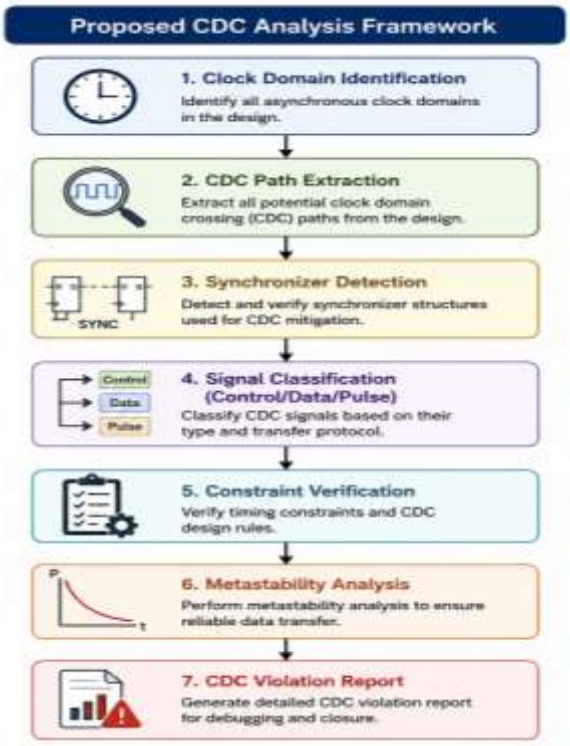


**Figure 1. Proposed CDC Analysis Framework for Timing-Critical FPGA Systems**

### *3.2. PROPOSED RDC ANALYSIS FRAMEWORK*

The proposed Reset Domain Crossing (RDC) analysis framework aims at providing reliable reset behavior and predictable initialization for timing-critical FPGA systems with multiple reset domains. With the advent of distributed reset topologies in modern FPGA designs for better modularity and power optimization, it becomes imperative to ensure competent RDC verification to avoid metastability, inconsistent initialization and functional failures.

Reset domain extraction is the first stage of RDC architecture. The reset signals and the corresponding sequential elements are derived from the FPGA netlist and grouped into corresponding reset domains. The framework considers reset propagation channels, fan-out configurations, and reset dependencies to determine how reset signals interact with different parts of the design. Special attention is paid to asynchronous reset networks, which are more vulnerable to timing-related problems during the deassertion of the reset signal.

The second level is about the analysis of reset synchronization. Asynchronous resets can be turned off independently of the target clock domain, requiring reset synchronization circuitry to avoid metastability during reset recovery. The framework independently recognizes reset synchronizers and examines the sufficiency of the synchronization mechanisms used for each reset crossing. Multi-stage reset synchronizers are evaluated to offer adequate metastability resolution time before the reset signals propagate to the functional circuitry.

The methodology identifies unsafe reset crossings by analyzing the relationships between reset domains operating under different clock or reset conditions. Structural analysis tools uncover dangerous reset paths, missing synchronizers, improper reset ordering, and conflicting reset dependencies. These verification checks prevent initialization mismatches, and operational instability in complex FPGA architectures.

Finally, a verification of the recovery and removal time is made to determine proper temporal correlations between the reset signals and clock edges. Recovery time violations happen when the deassertion of reset is close to an active clock edge, while removal time violations occur when the assertion of reset overlaps with current clock activity. The framework assesses the time constraints and finds possible violations that could lead to the sequential elements to reach unexpected states. The suggested RDC technique integrates the study of reset structure with the timing verification, which can increase the initialization reliability and the entire system robustness in the timing-critical FPGA applications.

### *3.3. INTEGRATED CDC-RDC VERIFICATION FLOW*

The suggested technique provides an integrated CDC-RDC verification flow, which merges the clock domain and reset domain analysis into one verification flow. Modern FPGA systems tightly couple clock and reset interactions. Separating the two encounters can let asynchronous errors slip through without notice. The integrated verification methodology aims at increasing the analysis coverage, reducing the verification complexity and improving the overall design reliability.

The first feature of the integrated flow is the coupling of structural and functional verification. Structural verification is the analytical checking of clock and reset crossing pathways, synchronization circuits and timing correlations on the netlist. The structural analysis is complemented by a functional verification that checks the behavior of the signals in a simulation realistic of the real-world operational environment. The combination of these approaches enhances the detection accuracy and allows to detect both static design problems and dynamic errors during execution .

The suggested flow consists of static timing analysis to capture asynchronous timing restrictions and synchronization behavior. Timing analysis tools evaluate false routes, asynchronous clock groups, multicycle paths and reset timing relationships to achieve precise timing closure. The framework validates that timing restrictions are correct to simulate asynchronous interactions and finds mismatches that may cause metastability or synchronization difficulties.

The proposed solution is further advanced by the application of formal verification tools to mathematically show the correctness of synchronization and the stability of reset features. Formal verification techniques employ assertions and comprehensive state-space exploration to uncover corner-case faults that may not be detected by conventional simulation. This enhances the verification confidence notably for safety essential FPGA systems under tight reliability constraints.

The last step of the integrated verification process includes automatic reporting and troubleshooting. The system creates reports that classify and identify hazardous clock crossings, reset violations, metastability concerns, missing synchronizers and timing constraint problems. Visualization tools and hierarchical debugging features allow designers to easily identify trouble locations in large FPGA designs. Automated severity classification helps to prioritize significant issues and cut the verification time. The joint CDC-RDC checking cycle provides a scalable and fast approach to analyze asynchronous timing behavior in complicated FPGA devices.

**Table 2. Summary of Proposed CDC-RDC Verification Framework**

| Verification Stage | Analysis Method | Purpose |
|---|---|---|
| Clock Domain Identification | Structural Analysis | Identify asynchronous clock domains |
| CDC Path Detection | Netlist Analysis | Detect clock crossing signals |
| Synchronizer Verification | Pattern Recognition | Validate synchronization circuits |
| Reset Domain Extraction | Reset Analysis | Identify reset dependencies |
| RDC Verification | Structural Checking | Detect unsafe reset crossings |
| Timing Verification | Static Timing Analysis | Check recovery/removal violations |
| Formal Verification | Assertion-Based Verification | Detect hidden corner-case failures |
| Reporting & Debugging | Automated Analysis | Generate violation reports |

# 4. CASE STUDY

## 4.1. FPGA SYSTEM DESCRIPTION

As a case study, a practically timing-critical system based on FPGA was selected to evaluate the effectiveness of the proposed Clock Domain Crossing (CDC) and Reset Domain Crossing (RDC) verification techniques. The platform used is a Xilinx Zynq-7000 FPGA architecture, which integrates programmable logic resources and embedded processing capabilities. This family of FPGAs is widely used in aerospace, industrial automation, telecommunications and real-time embedded systems for high-speed processing, flexible clocking resources and support of complex communication protocols.

The system has numerous functional subsystems, each in a different clock domain. It has a high-speed data acquisition module, a communication controller, a memory interface subsystem and a processor unit. The architecture includes many asynchronous clock sources, such as a 100 MHz system clock, a 200 MHz communication clock and a 50 MHz peripheral control clock. The multi-clock architecture offers parallel data processing and speed improvement, but it also introduces significant synchronization problems.

The system has several communication interfaces, including UART, SPI, Ethernet and DDR memory, each of which has different time limitations. The design comprises a hierarchical reset architecture with global resets, subsystem resets and local synchronized reset signals . Asynchronous resets are originally generated under power-up conditions and are then synchronized to certain clock zones to provide robust system initialization. The complexity of the multi-clock and multi-reset architecture makes the FPGA system very suitable for evaluating the suggested CDC and RDC analysis techniques in a real timing-critical scenario.

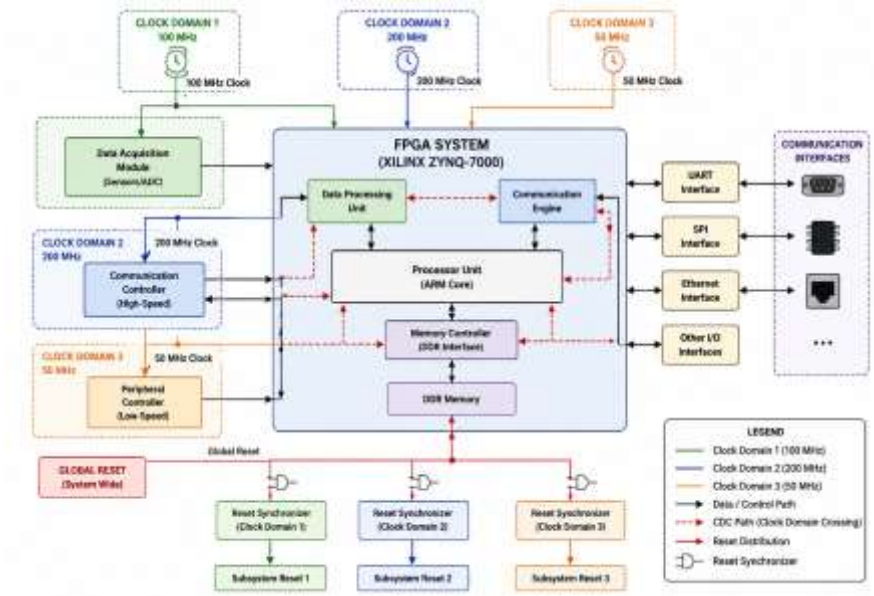


**Figure 2. Multi-Clock and Multi-Reset FPGA Architecture Used for CDC/RDC Verification**

## 4.2. EXPERIMENTAL SETUP

The experimental setup was built using industry standard FPGA design and verification tools to evaluate the proposed verification methods. RTL synthesis, implementation, static timing analysis and CDC verification was done using Xilinx Vivado Design Suite. Functional simulations were conducted out using Siemens QuestaSim, while formal verification and advanced CDC/RDC structure analysis were performed using Synopsys SpyGlass CDC/RDC tools.

The FPGA architecture was implemented with various asynchronous clock domains running at different frequencies. The primary system clock was 100 MHz. The communication and the peripheral subsystems were 200 MHz and 50 MHz, respectively. Additional clocks were derived from Phase-Locked Loop (PLL) resources to emulate real high-speed FPGA operation scenarios.

Simulation and verification involved multiple reset conditions, including asynchronous global reset assertion, delayed subsystem reset release and distinct peripheral resets. The verification setup included metastability analysis, false path constraint validation,

recovery and removal timing checks and synchronization structure identification. The experimental environment was set up to simulate time-critical operational scenarios and to evaluate the durability of the proposed CDC and RDC verification procedures under genuine FPGA implementation conditions.

### *4.3. CDC ANALYSIS IMPLEMENTATION*

The first step towards implementation of the CDC analysis was automatic recognition of clock domain crossover routes inside the architecture of an FPGA. The resultant netlist was structurally examined to derive clock relations and to classify signal transfers between asynchronous clock domains. The verification approach categorized these crossings into single-bit control transfers, multi bit data transfers and FIFO based communication interfaces. We examined each category with synchronization approaches adjusted to the communication parameters.

Several hazardous CDC pathways were initially found in the course of the study. The asynchronous control signals passing between the 100 MHz system clock domain and the 200 MHz communication clock domain demonstrated the metastability threat and possible timing violation. Single-bit control routes used two-flop synchronizers to allow enough time for metastability resolution to overcome this problem. High-speed multi-bit data transmission across clock domains was achieved by using asynchronous FIFO structures with Gray-coded read and write pointers for reliable data transfer.

Constraint validation was a key aspect of the CDC implementation procedure . Timing constraints were verified to ensure correct modeling of asynchronous timing relationships, including asynchronous clock group definitions, false path constraints and multicycle path declarations. During the investigation, various bogus path constraints were detected, lacking which could lead to wrong time closure during implementation. Static timing analysis techniques were used to re-verify and update the constraints.

The CDC verification approach also suffered from synchronization problems and reconvergence problems. Sporadic glitches due to loss of synchronization of a pulse signal crossing asynchronously from one clock domain to another. The problem was solved by developing a pulse synchronizer circuit together with handshake-based communication. Verification of the proposed synchronization methods after implementation indicated stable operation, removal of dangerous CDC violations and enhanced timing reliability on all asynchronous communication interfaces.

### *4.4. RDC ANALYSIS IMPLEMENTATION*

The implementation of the RDC study was largely focused in the assessment of the reset behavior in some reset zones inside the FPGA design. The initial step was to extract the reset domains. This approach involved identification and classification of all reset sources and their associated sequential elements by their reset dependencies. The architecture had a global asynchronous reset signal and various sub-system-specific reset domains on distinct clock domains.

The structural check discovered many reset crossings which de-assert asynchronous reset signals into active clock domains without synchronisation. These hazardous reset interactions had metastability, non-uniform initiation and reset recovery time violations as dangers. To solve these challenges, two-stage reset synchronizers inside each destination clock domain were created to reset the synchronization circuits. This enabled secure local clock edge reset deassertion.

The RDC verification approach revealed violations of recovery and removal timings at reset transitions. One situation was when a communications controller reset was asserted too close to an active clock edge, causing the registers to become undefined in simulation. The answer was to tweak the reset release mechanism and to add synchronized reset distribution logic.

Another RDC issue involved reset anomalies in combinational reset gating circuits. These anomalies propagated into portions of the design and briefly precluded subsystem start. The problem was resolved by redesigning the reset generating circuitry using registered reset control signals and synchronized reset sequencing operations.

The requested modifications were implemented and the RDC was tested several times. Stable reset behavior was observed, subsystems booted correctly and unsafe reset crossings were eliminated. The experimental results validated that the proposed RDC approach significantly enhanced the reset reliability and decreased the asynchronous reset-related timing challenges in the FPGA system.

# 5. RESULTS AND DISCUSSION

## 5.1. CDC VERIFICATION RESULTS

The proposed Clock Domain Crossing (CDC) verification method is implemented with considerable improvements in identification and mitigation of asynchronous timing issues in the FPGA system. In the initial step of the structural research, it was observed that numerous clock domains had 47 CDC violations. The breaches included incorrectly ordered control signal transfers, illegal multi-bit data crossings, reconvergence difficulties, and a lack of synchronization measures. Most of the breaches found were related to the communication interfaces operating between the 100 MHz system clock and the 200 MHz communication clock domain.

The recommended synchronization mechanisms like two-flop synchronizers, handshake protocols, and asynchronous FIFO systems, were implemented and the number of hazardous CDC pathways was greatly decreased. In the verification process, about 92 % of the faults initially detected were correctly corrected. The other pathways were categorized as planned asynchronous crossings and validated by applicable time constraints and formal verification techniques.

The synchronization efficiency was estimated in terms of the signal stability, the probability of the metastability propagation and the rates of successful data transmission in the asynchronous domains. The use of synchronizers has improved the steady capture of signals and the ambiguity of time in asynchronous communication. Gray-coded FIFO interfaces have been shown for reliable multi-bit data transfers with randomized clock phase settings.

The proposed methodology has been able to reduce considerably the metastability difficulties. The simulation results and the timing analysis showed that the implementation of the synchronizer considerably reduced the probability of metastability propagation. Extensive simulation cycles and formal verification examination did not reveal any functional problems owing to metastability. The results reveal that the proposed CDC architecture improves the synchronization reliability and stabilizes the operation significantly in timing-critical FPGA systems.

## 5.2. RDC VERIFICATION RESULTS

The Reset Domain Crossing (RDC) test process was able to successfully identify a large collection of timing and synchronization issues associated with resets in the FPGA architecture. Preliminary structural analysis revealed 29 violations of the requirements for asynchronous reset release, reset synchronization, and hazardous interactions among various reset domains. A number of violations were caused by subsystem reset controllers in different clock domains.

The analysis of recovery and removal times showed that some reset signals were removed close to active clock edges, which increased the chance of an unstable initialization of registers. The timing violations were more critical in the communication subsystem and memory controller interface. All major recovery and removal timing violations were eliminated with the adoption of synchronized reset release mechanisms and multi-stage reset synchronizers.

The performance evaluation of reset synchronization included the investigation of reset propagation stability, initialization consistency and the accuracy of reset release time. Use of logic for synchronized reset distribution greatly improved deterministic initialization performance of all subsystems. Simulation results show that all functional blocks are brought to a stable operational condition after reset deassertion without any glitches or failures due to metastability.

The registered reset control logic minimized signal glitches from resets and prevented unwanted transitions during initialization routines. This proposed RDC verification method has been found to be very effective in improving reset reliability and achieving stable reset behavior in multi-reset FPGA devices.

## 5.3. PERFORMANCE EVALUATION

The CDC and RDC verification procedures proposed had noticeable improvements on the overall system's performance and timing reliability of the FPGA. One intriguing result was the enhancement of the efficiency of timing closure. Before the proposed verification system, FPGA implementation has encountered various asynchronous timing violations that impeded the achievement of correct timing closure. All operating frequencies with consistent timing closure are achieved by using synchronization structures, optimal timing constraints and reset synchronization techniques.

Static timing studies showed improvements on setup, hold, recovery and removal timing margins across the whole FPGA system. False-path constraints were applied and asynchronous clock groups were accurately classified, resulting in a reduction of the needless workload of timing analysis and the amount of false timing violations reported by the synthesis tools. This enhanced accuracy of implementation results and accelerated the complete design validation process.

The impact of the proposed solutions on the use of the FPGA resources was analyzed. Some additional synchronizer circuits, asynchronous FIFOs, and reset synchronization logic enhanced a little the logic resource utilization. Synchronization phases somewhat enhanced the use of flip-flops, but FIFO systems required more memory and routing resources. The increase, however, was within the permissible design limitations and had little impact on the overall FPGA use efficiency.

### *5.4. COMPARATIVE ANALYSIS*

The effectiveness, scalability and accuracy of the proposed CDC and RDC verification methodology were evaluated in comparison to the traditional verification methods. Traditional timing verification methods are primarily focused on synchronous timing analysis and often provide limited support for asynchronous clock and reset interactions. The proposal is to combine structural analysis, temporal verification, formal verification and automatic detection of synchronizations into a single framework.

The proposed system has shown much better detection coverage for asynchronous timing problems than the independent simulation-based verification methods. Traditional simulations sometimes do not show metastability and rare failures of synchronization, because these problems happen only under specified timing conditions. The combination of structural and formal verification methodologies in the proposed methodology increased the capability of finding hidden corner-case failures and synchronization inconsistencies.

The scalability analysis indicated an improvement in comparison to traditional manual verification methods. Existing approaches tend to need a lot of manual tweaking of constraints and synchronization analysis, which gets progressively more difficult as FPGAs get bigger and contain several clock and reset zones. The automated reporting and hierarchical analysis in the suggested framework reduced the verification complexity and improved the debugging efficiency.

**Table 3. Performance Comparison of Traditional and Proposed CDC-RDC Verification Methods**

| Parameter | Traditional Method | Proposed Method |
|---|---|---|
| CDC Violation Detection | Moderate | High |
| RDC Violation Detection | Moderate | High |
| Metastability Analysis | Limited | Comprehensive |
| Automation Level | Low | High |
| Debugging Efficiency | Moderate | High |
| Verification Coverage | Partial | Comprehensive |
| Scalability | Limited | Excellent |
| Timing Closure Support | Moderate | Strong |

## 6. CONCLUSION AND FUTURE SCOPE

### *6.1. CONCLUSION*

This research offered comprehensive techniques to analyze Clock Domain Crossing (CDC) and Reset Domain Crossing (RDC) in timing-critical FPGA systems. Modern FPGA systems are highly reliant on many clock domains and distributed reset networks to achieve higher processing performance, optimize resource use and improve system flexibility. However, these architectural changes provide significant verification challenges for metastability, synchronization failures, reset instability, and asynchronous timing violations. The proposed procedures were developed to tackle these problems via integrated verification methodologies that combine structural analysis, static timing analysis, formal verification and functional simulation.

The proposed CDC architecture was focused on the discovery of asynchronous clock interactions, detection of hazardous synchronization structures, categorization of data transfer channels and validation of timing constraints across several clock domains. The RDC framework also provided for reset domain extraction, reset synchronization verification, recovery and removal timing analysis, and identification of risky reset crossings. The proposed methodology combines CDC and RDC analysis into a single

verification flow to improve coverage and increase the accuracy of detecting asynchronous timing errors that may be missed by traditional verification techniques.

The actual case study with FPGA demonstrated the efficiency of the proposed verification methodologies in a realistic multi-clock and multi-reset environment. Experimental results showed significant reductions in CDC and RDC violations due to the use of synchronizer circuits, asynchronous FIFOs, gray-coded interfaces and synchronized reset release techniques. The solutions strengthened temporal closure, reduced metastability hazards, improved reset stability and improved overall system robustness. Structural analysis and formal verification techniques were successfully applied to find hidden synchronization difficulties and reset time violations that were hard to find using simulation alone.

### *6.2. FUTURE SCOPE*

The presented approaches are quite effective in verification of CDC and RDC, although there are plenty of potential for additional research and improvement. An interesting avenue is the development of AI assisted verification techniques for CDC and RDC. Artificial intelligence and data-driven analysis can automate the classification of synchronization, the prediction of violations, and root-cause analysis, leading to reduced human verification and increased debugging efficiency.

Machine learning is a vast field of study. They attempt to keep false positives to a minimum. Modern structural analysis methods result in a huge number of warnings, which makes verification difficult and wastes important engineering resources. Applying machine learning model with historical verification data can be an efficient way to distinguish serious violations from non-critical alarms, thereby providing higher accuracy and efficiency to the analysis process.

In the future, FPGA systems can benefit from the development of adaptive synchronization algorithms that can be dynamically tuned to vary the synchronization behavior based on the operational variables such as clock frequency, voltage variations and workload requirements. Such adaptive approaches can improve the metastability reduction and time reliability in highly dynamic computing environments.

One key future research goal is to develop the heterogeneous SoC-FPGA designs, which consist of a CPU, accelerator, AI engine and programmable logic on a single platform. These systems feature more intricate asynchronous connections with more advanced verification techniques than the usual approaches of FPGA analysis.

## REFERENCES


[1] Vatandoost, Hassan, et al. "Comparison of CDC bottle bioassay with WHO standard method for assessment susceptibility level of malaria vector, Anopheles stephensi to three imagicides." *Journal of Arthropod-Borne Diseases* 13.1 (2019): 17.

[2] Kasim, Mohammad, Vrinda Gupta, and Mohandas Jebin. "Methodology for detecting glitch on clock, reset and CDC path." *2020 5th International Conference on Communication and Electronics Systems (ICCES)*. IEEE, 2020.

[3] Viswanathan, Priya, et al. "A Specification-Driven Methodology for the Design and Verification of Reset Domain Crossing Logic." *The Design and Verification Conference in US*. 2018.

[4] Borrud, Lori G., et al. "National health and nutrition examination survey: national youth fitness survey plan, operations and analysis, 2012." (2014).

[5] Guo, Jinquan, et al. "Urban global driving cycle construction method and global optimal energy management in plug-in hybrid electric vehicle." *Energy Procedia* 152 (2018): 593-598.

[6] Huang, Kuo-Hua, et al. "A virtual reality system to analyze neural activity and behavior in adult zebrafish." *Nature methods* 17.3 (2020): 343-351.

[7] Sirken, Monroe G., et al. "Changing methods of nchs surveys: 1960–2010 and beyond." *MMWR Suppl* 60.4 (2011): 42-48.

[8] Zhu, Motao, et al. "Implications of conducting trend analyses of emergency department visits using publicly released masked design variables." *Annals of emergency medicine* 57.6 (2011): 683-687.

[9] Ostchega, Yechiam, et al. "Waist circumference measurement methodology study: national health and nutrition examination survey, 2016." (2019).

[10] Pupo, Yasmine Mendes, et al. "Diagnostic validity of clinical protocols to assess temporomandibular disk displacement disorders: a meta-analysis." *Oral surgery, oral medicine, oral pathology and oral radiology* 122.5 (2016): 572-586.

[11] Hanchette, Carol, Charlie H. Zhang, and Gary G. Schwartz. "Ovarian cancer incidence in the US and toxic emissions from pulp and paper plants: a geospatial analysis." *International journal of environmental research and public health* 15.8 (2018): 1619.

[12] Mehrasa, Majid, et al. "Dynamic model, control and stability analysis of MMC in HVDC transmission systems." *IEEE Transactions on Power Delivery* 32.3 (2016): 1471-1482.

[13] Krenzke, Tom, et al. "Addressing disclosure concerns and analysis demands in a real-time online analytic system." *Journal of Official Statistics* 29.1 (2013): 99-124.

[14] Wu, Ximeng, Zi-Qiang Zhu, and Zhanyuan Wu. "A novel rotor initial position detection method utilizing DC-link voltage sensor." *IEEE Transactions on Industry Applications* 56.6 (2020): 6486-6495.

[15] Park, N. K., et al. "A Decision Processing Algorithm for CDC Location Under Minimum Cost SCM Network." *Asian Simulation Conference*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.