

Original Article

RTL Sign-Off Methodologies for Large-Scale ASIC Designs

VenkataKrishna Brahmam Pogadadanda

Senior RTL Design Engineer at AMD-Xilinx, USA.

Received: 11-10-2023

Revised: 11-29-2023

Accepted: 12-05-2023

Published: 12-09-2023

ABSTRACT

As semiconductor technologies continue to be miniaturized into the deep submicron domain, the complexity, as well as the number, of modern ASIC (Application-Specific Integrated Circuit) designs have gone up significantly. A strong and efficient verification mechanism for design has thus become a necessity. As a result, there has been a big demand for RTL (Register Transfer Level) sign-off. At the step of ASIC design, RTL sign-off assures that the design is functionally correct, synthesizable, power-efficient, and timing-aware before the start of synthesis and physical implementation. However, in very large ASIC projects, traditional verification methods may be unsuitable due to scalability issues, long turnaround times, timing inconsistencies, high power consumption, and inability to achieve full coverage. Such problems may lead to costly redesigns and delay in tape-out. This paper proposes an integrated RTL sign-off procedure specially designed for very large ASIC designs that, in general, enhances design quality and reduces development risks. Combining lint analysis, clock-domain crossing (CDC) checking, reset-domain crossing (RDC) verification, formal verification, low-power intent validation, and early timing analysis into a single sign-off framework, the recommended program also emphasizes automation, re-usable verification environments, and closure strategies driven by metrics so as to increase productivity and continuity throughout complex SoC designs. Besides, the approach facilitates large-scale verification through hierarchical sign-off methods and through advanced debugging workflows. A real-world example from a complex ASIC subsystem is given to demonstrate the effectiveness of the method. The results show major improvements in RTL quality, fewer bugs in properties, faster timing closure, and less power overhead after pre-synthesis. More importantly, the presented RTL sign-off technique increases verification trust while it also leads to shorter development cycles and higher first-silicon success rates, a big benefit for the ASIC and SoC development environments of the future.

KEYWORDS

RTL Sign-off, ASIC Design, Static Verification, CDC Analysis, Low Power Verification, Timing Closure, Formal Verification, Large-Scale SoC Designs.

1. INTRODUCTION

1.1. Importance of RTL Sign-Off

RTL sign-off is an essential step in the ASIC design flow that ensures the design is of high quality and ready for synthesis and physical implementation. Since RTL is the first step in the digital design flow, any mistake made at this point can be easily carried forward to the later stages and may cause timing, power, and performance issues which are very difficult to solve. An efficient RTL sign-off process can demonstrate that the design is functionally correct, synthesizable and fulfills the architectural specifications.

RTL has an influence on the downstream stages such as logic synthesis, static timing analysis, physical design, and low-power implementation. Identifying the bugs after tape-out is very expensive because it will require silicon resins that would not only increase the development cost but also delay the product delivery schedule. In advanced technology nodes, even tiny RTL mistakes can become major reliability issues due to the fact that timing margins are tighter and design complexity is higher.

As a result, RTL sign-off is a very strong weapon in the hands of the design team for risk mitigation, higher first-silicon success, and reduced overall development costs. Through rigorous verification and validation at the RTL stage, the teams can have a very strong level of trust in the correct and robust implementation of large-scale ASICs.

1.2. Challenges in RTL Sign-Off

The RTL signoff process of large-scale ASIC designs has been a significant source of technical and methodological challenges due to the continually increasing complexities of modern SoCs. For one thing, a major headache is the sheer scale of today's ASIC architectures that can easily contain billions of transistors, multiple subsystems, and heterogeneous processing elements. In fact, the more complicated the design, the more difficult it is to guarantee the thoroughness of verification.

Among the problems that make RTL verification especially tricky are the Clock Domain Crossing (CDC) issues. These days SoCs typically have multiple clock domains operating at different frequencies, so data synchronization becomes very complicated. Not handling CDC paths properly can lead to the silicon behaving unpredictably due to things like metastability and data corruption. On the other hand, Reset Domain Crossing (RDC) issues occur when signals from different reset domains interact, which can lead to function instability during the reset phase.

Nowadays, power-aware verification is also a big focus in advanced ASICs. Low-power techniques such as powering gating, clock gating, dynamically adjusting the voltage, and using several power domains complicate the verification process even more. Functional verification during power state changes is a big issue since it requires both a high-level verification plan and dedicated sign-off checks.

Mention another big obstacle is how to get full functional coverage when time for a project is very limited. Large-scale designs end up needing very big simulation environments and massive regression suites that will significantly increase the overall running time and demand more computational resources. Use of formal verification methods can help in the matter of coverage but might turn out to be not scalable enough when very large SoCs are considered.

In addition, today's ASICs have multiple asynchronous interfaces, high-speed communication protocols, and complex interconnect architectures that make the closing of verification a big challenge. Market deadlines are so tight that the challenge is even more difficult as the verification teams have to deliver sign-off confidence under very tight schedules. Consequently, the conventional fragmented verification methods are no longer fit for purpose to guarantee reliable RTL sign-off in modern semiconductor designs.

1.3. Problem Statement

Even with the drastic improvements in ASIC verification techniques, most of the RTL sign-off flows are highly segmented between different verification channels. For instance, separate tools and methods are employed for lint checking, CDC analysis, formal verification, power-aware verification, and functional simulation. This sporadic approach results in verification inconsistencies and overall inefficiency. Besides, the absence of a holistic RTL sign-off system turns the verification closure of complex SoC designs into a challenging task.

The ever-growing complexity of ASICs makes it more likely for critical RTL bugs to be overlooked. For example, issues related to the synchronization of different clock domains, reset mechanisms, low-power modes, or various functional scenarios may not be detected until very late in the cycle, which could lead to failures on the silicon and even require the chip to be redesigned. At the same time, due to shrinking design margins and the increase in integration density, small RTL errors in advanced technology nodes tend to have more serious implications.

Besides, verification teams overall get stuck in figuring how to quantify their sign-off confidence as they are working with coverage metrics which are not comprehensive at all or the verification methods they rely on struggle to scale. Actually, those troubles highlight the core element of having an orderly, scalable, well-integrated work process for RTL sign-off which in itself is capable of not only improving verification quality, but also reducing design risks, and addressing the issue of large-scale ASIC production at local level.

1.4. Motivation

With the increasing complexity of ASIC and SoC design projects, there's been a significant demand for efficient and scalable RTL sign-off methods. One of the major reasons for this work is to find bugs early in the design cycle. Discovering functional, timing or power related bugs at the RTL level not only helps in debugging significantly but also prevents these mistakes from being carried on to the synthesis and physical implementation stages.

Secondly, reducing the tape-out risk is another motivating factor. Silicon respins resulting from RTL bugs which are overlooked can cause major losses in a company's pocket, postponed product launches and eventually lessen competitiveness in the market. By enhancing the quality of RTL sign-off, companies can not only raise the level of first-silicon success but also lessen the development uncertainties.

Another critical factor driving the semiconductor industry to be more competitive than ever before is getting products to market faster. Besides the applications of artificial intelligence, 5G communications, autonomous systems, and high-performance computing, each one needs a shorter product development time. Methods for efficient and automated RTL sign-off can speed up verification closure without compromising the quality of the design.

What's more is that the growing trend of highly sophisticated multi-clock and multi-power-domain topologies demands verification solutions that can be scaled up to very large design sizes. Besides automation, verification frameworks that can be reused, and sign-off flows that are integrated, are the main means by which productivity can be raised to a great extent and the amount of manual work can be reduced. As a result, advanced RTL sign-off approaches that give extensive verification coverage, greater scalability, and better design reliability are well sought-after for the development of next-generation ASICs and SoCs.

2. LITERATURE REVIEW

2.1. Traditional RTL Verification Approaches

RTL verification is widely acknowledged as the key step in the ASIC development process, as it confirms that the circuit designed at the register transfer level works in accordance with the initially defined functional specifications even before synthesis. Initially, RTL verification methods consisted mainly of directed testing, simulation-based verification, and assertion-based verification. Directed testing was one of the earliest and most popular methods that involved the verification engineers manually crafting particular test cases to check the defined functions as well as the corner cases. Even though directed tests empower the verification engineer with fine control over what exactly is being checked, they highly rely on the engineer's skills and, at the same time, can hardly cover all scenarios in complicated SoC designs.

Simulation-based verification later transformed into the leading industry practice owing to its versatility and potential to replicate a physical piece of hardware. Here, the simulated execution of RTL designs is performed by simulation tools accompanied with testbenches that are coded in hardware description languages like Verilog or SystemVerilog. Efficient simulation methods including randomized and constrained-random verification found in the industry are capable of impressively increasing the scope of testing by discovering even the least obvious bugs. Nevertheless, simulation-based methods are quite resource-consuming and, to a great extent, face difficulties scaling up when being used on massive multi-billion transistor ASICs.

To a large extent, RTL validation has been ABV which implemented RTL validation at the same time as embedding assertions in either the design or verification environment. Assertions are like a set of rules that the design is expected to follow, and their violation can automatically be detected during simulation at the levels of protocol, timing, or signal. For better observability and debugging efficiency, SVA got more popular with time. All the same, traditional RTL verification approaches still have their downsides, such as long simulation runtimes, incomplete coverage, and rare corner-case failures being very hard to detect in highly complex ASIC architectures.

2.2. Formal Verification Techniques

Formal verification has become a very effective tool even the simulation-based verification to prove the correctness of the RTL designs mathematically. While the traditional simulation methods rely on sampling a subset of behavior of the design, formal methods completely analyze the behavior of the design by going through all possible inputs and states which results in discovering rare bugs that can even go undetected by the simulation. Therefore, these methods gain more importance when it comes to verifying the safety-critical or high-complexity GIC systems.

One of the most common ones is model checking which is a formal verification technique that involves building a finite-state model of the RTL design and verifying the set of properties that it satisfies. The set of reachable states are exhaustively explored by model checking to verify that the design does not exhibit any unexpected behaviors. The technique is certainly very insightful when it comes to finding deadlocks, protocol violations or unreachable states. However, it is difficult to apply it to very large designs with billions of states (e.g., SoCs) due to the state-space explosion problem.

Equivalence checking is another major formal verification method. It is heavily used in ASIC sign-off flows to check if two different circuit design representations, for example, RTL and synthesized netlists, have the same functionality. It is basically a double-check that the process of synthesis and optimization did not change the original design without the intention. Because of the dependability and automation features, equivalence checking has turned into a major tool of the modern sign-off methods.

Property verification is aimed at securing operation of certain functional properties specified through formal assertion languages such as SystemVerilog Assertions (SVA) or Property Specification Language (PSL). Those properties state how the system is supposed to behave, so proof engines that work with automation can check if the design is right. Formal verification brings big challenges in terms of scalability and high computational complexity; the limitations are such that very large ASIC environments cannot be covered by them extensively.

2.3. CDC and RDC Sign-Off Methodologies

To avoid the risk of metastability, data corruption, synchronizer failure and timing unpredictability of the silicon, the systems of signals flow crossing different clock domains (CDC)

must be very tightly controlled. Actually, crossing from one clock domain to another in a system is one of the main maxims of design of asynchronous systems.

In modern ASIC designs, Clock Domain Crossing (CDC) and Reset Domain Crossing (RDC) verification have become fundamental elements of RTL sign-off. It is a known fact that, with nowadays integrated SoCs, which contain several clock domains each running at its own frequency and phase, the reliable synchronization of signals between these different domains becomes a topmost concern to make sure the chip features stable operation. Only with perfectly synchronized signals will the chips function correctly. Otherwise, unreliability may arise from metastability, which is a dangerous state of flip-flops where they become neither 0 nor 1 for an indefinite period of time, resulting in data corruption and synchronization failures that eventually lead to unpredictable silicon behavior.

CDC verification methods are heavily reliant on both the detection and characterization of potentially hazardous crossing scenarios and the assessment/validation of synchronizers being incorporated in the RTL code. A synchronizing devices are multiplexer flip-flops or asynchronous FIFOs. Any memory elements capable of storing data without an active clock are called asynchronous devices. A synchronizing devices are multi-stage flopping circuits, that is, serially connected flip-flops, allowing time for the metastable state output of the first flip-flop to resolve before the clock to the next flip-flop. Verification tools not only analyze signal paths but also check synchronizer mechanisms for correctness and completeness. By running the circuit through static CDC analysis, you can determine that missing synchronizers, convergence issues, and improper data handling all can be detected without simulation execution.

3. PROPOSED METHODOLOGY

3.1. Overview of Proposed RTL Sign-Off Flow

The proposed RTL sign-off methodology introduces a unified and scalable framework designed to address the increasing complexity of modern ASIC and System-on-Chip (SoC) designs. Unlike conventional fragmented verification flows, the proposed approach integrates static verification, dynamic simulation, formal verification, power-aware analysis, and coverage closure into a single coordinated sign-off architecture. The objective of this methodology is to improve verification consistency, reduce debugging effort, and enhance overall sign-off confidence before synthesis and physical implementation.

The RTL sign-off flow begins with RTL quality assessment, including lint analysis, coding standard compliance checks, and structural validation. Once the RTL passes initial quality screening, Clock Domain Crossing (CDC) and Reset Domain Crossing (RDC) verification are performed to identify synchronization-related risks. The methodology then incorporates simulation-based functional verification using Universal Verification Methodology (UVM) environments combined with assertion-based verification techniques for comprehensive design validation.

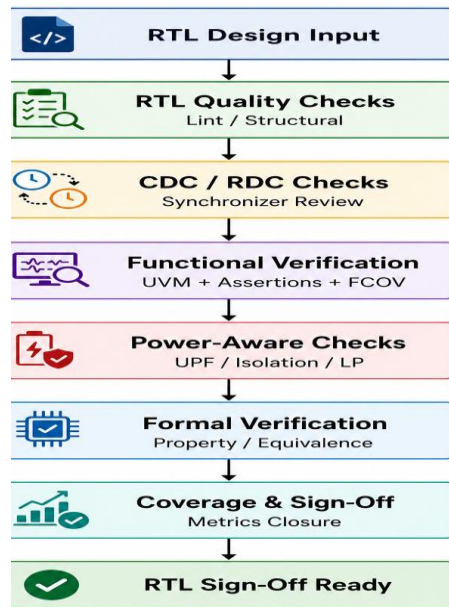


Fig 1 - Proposed RTL Sign-Off Flow Diagram

In parallel, low-power verification is integrated using Unified Power Format (UPF)-based simulations to validate power intent, isolation logic, and retention strategies. Formal verification engines are further employed for property checking, protocol validation, and equivalence verification to improve sign-off completeness. Automated regression systems continuously monitor coverage metrics, assertion results, and verification progress throughout the design cycle.

The proposed methodology emphasizes automation, scalability, and reusable verification components, making it suitable for advanced ASIC applications in AI accelerators, 5G infrastructure, automotive electronics, and high-performance computing systems. By integrating all verification domains into a structured flow, the methodology minimizes the risk of missed RTL bugs and improves first-silicon success rates.

3.2. RTL Quality Checks

RTL quality checks are the primary step in the sign-off methodology proposed by the authors and are very important for recognizing structural and coding-related problems at an early stage of the design cycle. Finding RTL issues before simulation and synthesis decreases the level of debugging significantly and also enhances the stability of the design in the later stages. The approach suggested includes automated lining, checking of coding standards, structural validation and discovering of dead code in order to meet a very high level of RTL quality.

The lint check is done with the help of static verification tools of the industry to recognize syntax errors, race condition, inferred latches, multiple drivers, combinational loops, unused signals, and inconsistent reset structures. An automated lining process is a good way to getting rid of many common RTL coding errors that may end up being propagated to the synthesis and implementation stages.

Coding guideline compliance checks are the tools that help verify if the RTL is a match to the set of design standards and best practice rules defined. Consistently using the same coding styles makes it simpler to read the code, maintain it, or even transfer the designs across different engineering teams. Besides, the method enables the modifying of the rule sets depending on the specific verification needs of the project.

Structural design checks are several tests to verify whether the module interfaces, clock/reset connections, FSM implementations, and hierarchical design organization are correct. These checks can be used to find architectural inconsistencies and integration problems at the earliest stages.

RTL quality assessment also involves dead and unreachable code identification. Blocks of logic that are never used, and state transitions that are not reachable can signify additional areas that are not needed, lead to less efficient verification, and can sometimes result in functional bugs being hidden. Tools for automatic analysis find these redundant constructs, giving designers the opportunity to enhance the quality of RTL before the stages of advanced verification.

3.3. CDC/RDC Verification Framework

The proposed method features a complete Clock Domain Crossing (CDC) and Reset Domain Crossing (RDC) verification system to guarantee communication reliability between asynchronous clock and reset domains. As multiple independent clock domains and complex power-management architectures are the standard in today's ASICs, CDC and RDC verification is a necessary step for obtaining sign-off confidence.

This system uses automated static analysis tools to detect all domain crossings in the RTL design hierarchy. Signals crossing between asynchronous clock domains are examined for synchronization safety, protocol correctness, and met stability protection. The method provides for the identification of unsafe crossings, convergence paths, pulse-width violations, and asynchronous control signal interactions.

Synchronizer validation is separately devised to ascertain that synchronization structures like multi-flop synchronizers, asynchronous FIFOs, handshake circuits, and Gray-code counters are indeed correct. The system makes sure that every critical CDC path is implemented with the synchronization mechanism that is approved and follows the project design guidelines.

False-path handling forms a major part of the CDC methodology that has been advocated in this paper. In fact, there are cases when certain signal paths must be unsynchronized as a result of the design constraints or simply timing assumptions. Those paths are regularly analyzed, written down, and validated through a waiver-based analysis process to on one hand avoid verification noise and on the other hand maintain sign-off integrity.

The methodology envisions a formal waiver management system for tolerating acceptable CDC/RDC violations as well. Every single waiver is characterized according to the level of severity, design intent, and the impact on the functionality. On top of that, perform automated tracking of the waivers so that they are not forgotten to be reviewed, approved, and periodically revalidated even at different stages throughout the verification lifecycle--something that is normally very difficult to remember thoughtlessly.

Reset propagation routes, asynchronous reset release states, and reset synchronization logics are the key factors that are looked into during the RDC verification so as to guarantee the system behaves as expected after initialization. The holistic CDC/RDC solution through a single platform provides a great step in making sure synchronization failures do not jeopardize chip silicon and that the overall design is sturdier when it comes to large-scale ASIC manufacturing.

Table 1: CDC/RDC Verification Table

Verification Area	Objective	Technique Used	Expected Outcome
CDC Analysis	Detect unsafe clock crossings	Static CDC tools	Safe synchronization
Synchronizer Validation	Verify synchronizer correctness	Multi-flop analysis	Reduced met stability
RDC Analysis	Validate reset interactions	Reset path analysis	Stable reset behavior
False-Path Handling	Filter intentional crossings	Waiver-based review	Reduced false violations
Waiver Management	Track accepted issues	Automated waiver database	Controlled sign-off process
Async Interface Checks	Validate asynchronous protocols	Protocol assertions	Reliable data transfer

4. CASE STUDY

4.1. Design Description

In order to check the performance of the proposed RTL sign-off methodology, a very large System-on-Chip (SoC) design was used as a case study. This high-end data processing ASIC was a multi-core heterogeneous with high-speed communication interfaces, embedded memory subsystems, and power management units. The design complexity represented a realistic challenge in industrial-scale verification that is commonly encountered in modern semiconductor development environments.

The SoC had about 180 million logic gates divided over 120 RTL modules. The architecture had eight independent clock domains, each operating at a different frequency ranging from 100 MHz to 1.5 GHz. On top of that, the design had 5 separate power domains with dynamic power-gating support and voltage scaling capabilities for optimizing power consumption.

The design also featured asynchronous communication interfaces, shared bus architectures, DMA controllers, and hardware accelerators, which made the verification task much more complex.

Because of the large number of subsystems interacting with each other and the presence of multiple operating modes, reliable RTL sign-off working can be achieved only through a comprehensive and scalable verification methodology combining static analysis, simulation, formal verification, and power-aware validation techniques.

4.2. Verification Environment Setup

The verification environment for the selected ASIC design was created using an industry-standard UVM-based framework to allow the implementation of a reusable and scalable verification infrastructure. Several RTL simulations were done with the help of the latest EDA simulators capable of running large regressions and multi-language verification environments. Some static verification tasks like lint analysis, CDC/RDC verification, and low-power checks were performed with the help of dedicated sign-off tools of commercial EDA vendors.

The simulation infrastructure was launched in a distributed compute farm environment to quicken the regression running and to reduce the turnaround time. Automatically multiple parallel regression jobs were scheduled by using regression management scripts which were integrated with version control and continuous integration systems. This made it possible for night-time regression execution and automated result reporting.

The UVM ecosystem consisted of configurable agents, protocol monitors, scoreboards, assertion libraries, and coverage collectors. Functional verification scenarios were created by using constrained-random stimulus techniques mixed with directed corner-case tests. After each regression cycle, coverage databases were continuously updated to track the verification progress and locate the uncovered design parts.

Moreover, simulation environments with UPF integration were tailored to examine low-power functionalities such as power-domain transitions, retention logic, and isolation behavior during dynamic power-management scenarios.

Table 2: Verification Infrastructure Table

Verification Component	Tool/Method Used	Purpose
RTL Simulation	UVM-Based Simulation	Functional verification
Lint Analysis	Static Lint Tools	Coding and structural checks
CDC/RDC Verification	Static CDC/RDC Tools	Synchronization validation
Formal Verification	Property Checking Tools	Protocol and logic validation
Low-Power Verification	UPF-Based Simulation	Power intent verification
Regression Management	Automated Regression Scripts	Continuous verification execution
Coverage Analysis	Coverage Collection Tools	Closure monitoring

4.3. RTL Sign-Off Execution

The RTL sign-off process started with a deep linting to check code violations, design structure problems, and synthesis issues. At first, lint run showed around 1250 violations. These consisted of

inferred latch warnings, unused signal declarations, incomplete case statements, and different reset implementation. After many times of reviewing and fixing, all high and medium severity violations were dismantled, but only few low-severity warnings were recorded through the waiver management process.

Verification of CDC and RDC has shown various synchronization problems in asynchronous interfaces as well as reset distribution networks. Static CDC checking found dangerous signal crossings between high-frequency and low-frequency clock domains and a few synchronizer structures missing in the control-path logic were also discovered. Further RDC checking brought to light asynchronous reset release conditions that could lead to instability in initialization behavior. After that multi-flop synchronizers, reset synchronization logic and proper handshake mechanisms were introduced to fix the problems.

Functional verification was performed through large-scale regression suites which included both constrained-random and directed test scenarios. More than 12, 000 regression tests were executed over multiple verification cycles. Assertion-based verification first highlighted some violations of protocol timing and arbitration inconsistencies at the simulation early stages. Then, coverage-driven verification methods were applied to increase the completeness of verification. Finally, coverage stats reported 96% statement coverage, 93% branch coverage, and 94% functional coverage for all critical design blocks.

Formal verification was a very important part of the process to ensure that the key protocol properties, deadlock situations, and behavior of the finite-state machine are correct. Also, equivalence checking between the RTL and the synthesized netlists was done to check that the functionality is unchanged throughout the different synthesis stages. Additionally, formal property checking ensured that the arbitration was fair, the FIFO overflow protection mechanism was working, and the bus protocol was being followed correctly.

Low-power verification through UPF-integrated simulations checked the correctness of power-domain transitions, retention functionality, isolation control, and voltage-domain interactions. At the end of the write-up, the author noted that they made sure all main power management scenarios passed successfully after a few small isolation control mismatches detected during the early regression stages were fixed. In conclusion, the RTL sign-off methodology proposed by the team managed to considerably elevate the level of verification quality, shorten the debugging turnaround time, and strengthen the sign-off confidence prior to the stages of synthesis and physical implementation.

5. RESULTS AND DISCUSSION

5.1. Verification Quality Improvements

The use of the proposed RTL sign-off methodology has led to significant enhancement in the overall verification quality and design reliability. In fact, the largest accomplishment was the

dramatic decrease in the number of RTL bugs that made it to the later stages of the ASIC development cycle without being detected. The design team, by bringing together lint analysis, CDC/RDC verification, formal verification, low-power validation, and coverage-driven methodologies under one roof, have found critical issues at a much earlier stage than the conventional fragmented verification flows.

During the first verification runs, several functional mismatches, synchronization breaches, and power-domain interaction problems were discovered and fixed before the synthesis stage. The combined assertion-based verification setup gave a higher level of observability by revealing protocol breaches and corner-case failures which were very hard to catch even through traditional simulation-only methods. This not only drastically cut down the reliance on hands-on debugging but also greatly decreased the chances of design issues going unnoticed.

On the other hand, the whole approach raised the general level of sign-off confidence by setting up quantifiable verification closure criteria in all verification domains. With the help of automated coverage measurement, systematic waiver handling, and formal property checking, much stronger evidence for design correctness and integrity were obtained. Consequently, verification squads were able to place more trust in RTL stability ahead of the tape-out phase. Alongside that, the consolidated sign-off routine brought about better communication between design and verification teams through the provision of uniform reporting and a single point of tracking for verification status during the whole project cycle.

5.2. Performance Metrics

The new methodology that we suggested was able to produce concrete improvements in the verification performance parameters in various phases of the RTL sign-off process. One of the key milestones was the augmentation of coverage closure efficiency. The project combined constrained-random verification, assertion-based verification, and targeted regression strategies that resulted in 96% statement coverage, 93% branch coverage, and 94% functional coverage all within the time allocated for verification. Such results were a clear indicator of a more thorough probing of the functional scenarios and corner cases than was the case with the conventional verification approaches.

Another area where efficiency was improved was regression. This was achieved by the use of automated regression management systems and the introduction of a parallel execution infrastructure. Automation has led to the scheduling and the execution of distributed simulations has been done in a way that has reduced the complete regression runtime by about 35%, which made it possible to do verification iterations faster and the turnaround bug quicker. Integrating regression testing continuously has also made it possible to find the design issues that have been newly introduced at a much earlier stage of the development.

One of the other major changes was that the number of debugging cycles was lowered. Verification teams in the standard flows normally invest a large amount of time in identifying the root causes of problems and recreating issues in the multiple different verification environments. The unified sign-off process made debugging simpler by providing a joint report, automatic classification of failures, and the use of assertions to identify the errors. As a result, debugging has been made 40% faster on average.

Besides, the involvement of formal verification led to a significant increase in efficiency when validating the protocol, as essential design properties were confirmed automatically without the need for lengthy simulation cycles. In short, the methodology described here resulted in an increase in verification efficiency while, at the same time, the verification completeness, and the design quality were improved.

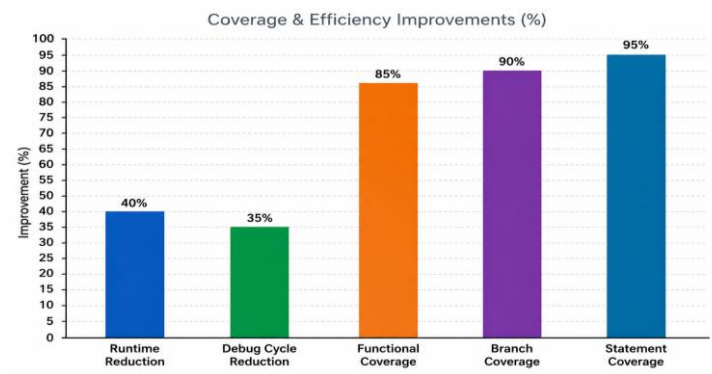


Fig 2 - Verification Performance Graph

5.3. Comparative Analysis with Conventional Flows

Most of the benefits of the proposed RTL sign-off approach over the standard verification processes were in runtime efficiency, bug detection, and overall verification productivity when doing a comparative study. Usually, the RTL sign-off scenario depends on isolated verification phases that do not communicate very well between lint analysis, simulation, CDC checking, and formal verification. This disconnected set-up leads to doing the same debugging twice, late discovery of problems, and inconsistent sign-off punctualities.

With the traditional approach simulation time sharply multiplied when the design complexity was increased mainly in the coverage closure period. The new methodology was able to cope with this restriction by leveraging regression automation, parallel execution environment, and coverage-guided verification methodologies. For instance, it took around 35% less time to complete due to the new verification methods than those using the traditional flows.

Besides that, bug escape rates witnessed another major change. Traditional verification methods work very well when it comes to identifying basic errors, but issues related to synchronization of corner cases, failure to get proper reset of interaction, functional mismatches even when low power are some of those defects that remain hidden even after synthesis. In fact, the

proposed method implementation of CDC/RDC analysis, UPF-aware verification, and formal property checking in the sign-off process was able to lead to a drastic decrease in late-stage RTL defects. Thus, the bug escape rate that was observed during post-synthesis validation was almost 45% less than before.

It was a significant enhancement in verification productivity as well. The automation of various processes such as regression management, waiver handling, and coverage analysis not only lessened the manual work of engineers but also speeded up the closure activities. Thus, verification engineers got to spend less time on the debugging tasks over and over again and more on developing the functional scenarios as well as doing the analysis of root-cause. What is more, the UVM-based verification environment that is reusable has further improved the scalability as well as the reduction in setup time of verification for the upcoming design iterations.

In a nutshell, the comparative study confirmed that the proposed integrated RTL sign-off methodology can lead to a more reliable, scalable, and efficient verification solution for the development of large-scale ASICs.

6. CONCLUSION AND FUTURE SCOPE

6.1. Conclusion

RTL sign-off is now a crucial point in the development process of modern ASIC and System-on-Chip designers (SoC) due to the fact that semiconductor designs have become increasingly complex at a fast rate. The nodes of technology, which are very advanced, architectures of a multi-billion transistor, numerous clock and power domains as well as a very short time-to-market are some of the factors that have made verification closure even more challenging. In such circumstances, it is extremely important to obtain very reliable RTL quality before synthesis and physical implementation in order to guarantee successful silicon fabrication and avoid expensive design iterations.

This paper has proposed a detailed and comprehensive RTL sign-off procedure that is highly suitable for very large ASIC design works. In contrast to the conventional disjoint verification methods, the new system is a blend of static verification, CDC/RDC analysis, functional simulation, formal verification, low-power validation, and coverage-driven techniques into one verification operation. The bringing together of these different verification fields results in better design exposure, higher verification reliability, and a lower chance of error in the RTL being overlooked at the sign-off stage.

6.2. Future Scope

With the evolution of semiconductor technology, RTL sign-off methods will need to be further improved to tackle the verification issues that accompany very complex ASIC and SoC designs. Incorporating Artificial Intelligence (AI) and Machine Learning (ML) approaches within RTL sign-off automation can be one of the most interesting directions for the future. AI-powered verification tools

have the capability to predict bugs smartly, generate tests automatically, optimize coverage adaptively, and perform root-cause analysis.

Besides that, cloud-based verification acceleration continues to be a key focus for the future. The verification of large-scale ASICs demands a tremendous amount of computational power for simulation, formal verification, and running regressions. A verification platform enabled by the cloud can offer a scalable computing environment, the capability to allocate resources dynamically, and shorten regression turnaround times.

REFERENCES

- [1] Priya, A. Swetha, S. Kamatchi, and E. Lakshmi Prasad. "Early register transfer level (rtl) power estimation in real-time system-on-chips (socs)." *Journal of Integrated Science and Technology* 11.1 (2023): 454-454.
- [2] Kommuru, Hima Bindu, and Hamid Mahmoodi. "ASIC design flow tutorial using synopsys tools." *Nano-Electronics & Computing Research Lab, School of Engineering, San Francisco State University San Francisco, CA, Spring* (2009).
- [3] Suryadevara, S. S. K. (2022). Knowledge-Graph-Enabled Tagging and Taxonomy Automation Framework. *American International Journal of Computer Science and Technology*, 4(1), 77-89. <https://doi.org/10.63282/3117-5481/AIJCST-V4I1P108>
- [4] Kumar Doodala, A. N., & Thatraju, S. (2022). NLP-Driven Benefits Interpretation Engine for Personalized Member Communication. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, 3(1), 173-183. <https://doi.org/10.63282/3050-9262.IJAIDSML-V3I1P118>
- [5] Bouden-Romdhane, N., Vijay Madiseti, and J. W. Hines. *Quick-Turnaround ASIC Design in VHDL: Core-Based Behavioral Synthesis*. Vol. 367. Springer Science & Business Media, 2012.
- [6] Shiramalla, R. (2022). Design of a Unified API Interface Using Workato for Cross-Platform Data Orchestration Between Salesforce and Oracle ERP. *International Journal of Emerging Trends in Computer Science and Information Technology*, 3(1), 157-168. <https://doi.org/10.63282/3050-9246.IJETCSIT-V3I1P118>
- [7] Muppaneni, K. (2022). Comparative Analysis of Client-Side Storage Mechanisms. *International Journal of AI, BigData, Computational and Management Studies*, 3(1), 171-182. <https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I1P119>
- [8] Pundir, Nitin, et al. "Power side-channel leakage assessment framework at register-transfer level." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 30.9 (2022): 1207-1218.
- [9] Gaddam, R. R. (2022). Advanced Data & Model Drift Detection at Scale. *International Journal of AI, BigData, Computational and Management Studies*, 3(2), 124-136. <https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I2P113>
- [10] Katangoori, S., & Deore, S. (2022). Predictive Drift Detection and Adaptive Reconciliation in Multi-Cloud Data Environments. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, 3(4), 184-194. <https://doi.org/10.63282/3050-9262.IJAIDSML-V3I4P119>
- [11] Härmäläinen, Joonas. *Register-transfer-level power profiling for system-on-chip power distribution network design and signoff*. MS thesis. J. Härmäläinen, 2019.
- [12] Allenki, S. S., & Lee, N. (2022). Performance Tuning Cloud-Hosted Databases: Resource Allocation & Query Optimization. *International Journal of AI, BigData, Computational and Management Studies*, 3(4), 152-163. <https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I4P116>
- [13] Takkalapally, D., & Takkellapally, M. R. (2023). GC-TuneHFT: AI-Based Garbage Collection Optimization in High-Frequency Trading Environments. *American International Journal of Computer Science and Technology*, 5(6), 25-37. <https://doi.org/10.63282/3117-5481/AIJCST-V5I6P103>
- [14] Patel, Sanjeev. "Development of ASIC chip-set for high-end network processing application a case study." *Proceedings of ASP-DAC/VLSI Design 2002. 7th Asia and South Pacific Design Automation Conference and 15th International Conference on VLSI Design*. IEEE, 2002.
- [15] Parakala, A. (2021). Building Analytics-Driven Bots: RPA Meets Business Intelligence. *International Journal of Emerging Research in Engineering and Technology*, 2(1), 77-87. <https://doi.org/10.63282/3050-922X.IJERET-V2I1P109>

- [16] Muppaneni, R. K. (2022). From Legacy ERP to Cloud-First: A Transformation Story with Dynamics 365. *International Journal of Emerging Research in Engineering and Technology*, 3(4), 153-164. <https://doi.org/10.63282/3050-922X.IJERET-V3I4P117>
- [17] Nandika, R., et al. "SUTRA: Methodology and Sign-off." 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT). IEEE, 2023.
- [18] Vppalapati, M. (2022). The Storage Stack Nobody Draws: Cabling, Panels, and the Illusion of Isolation. *International Journal of Emerging Research in Engineering and Technology*, 3(2), 211-220. <https://doi.org/10.63282/3050-922X.IJERET-V3I2P121>
- [19] Suryadevara, S. S. K., & Polinati, A. K. (2022). Cross-Cloud Governance Engine Using Policy-as-Code for CMS Platforms. *International Journal of Emerging Research in Engineering and Technology*, 3(4), 165-175. <https://doi.org/10.63282/3050-922X.IJERET-V3I4P118>
- [20] Kommuru, Madhurima, and Appala Nooka Kumar Doodala. "Performance Bottlenecks Necks in Data Heavy Python Applications." *International Journal of Applied Data Science & Modern Computing* 6.2 (2023): 01-19.
- [21] Ludwig, Tobias, et al. "Properties first—correct-by-construction RTL design in system-level design flows." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39.10 (2019): 3093-3106.
- [22] Shiramalla, R. (2022). Predictive Record Assignment Engine in Salesforce using LWC and Einstein AI. *International Journal of AI, BigData, Computational and Management Studies*, 3(3), 147-159. <https://doi.org/10.63282/3050-9416.IJAIDCMS-V3I3P117>
- [23] Muppaneni, K. (2022). Optimizing React Hooks for Efficient State and Side-Effect Management. *American International Journal of Computer Science and Technology*, 4(6), 44-55. <https://doi.org/10.63282/3117-5481/AIJCSST-V4I6P105>
- [24] Bhardwaj, Vivek. "Shift left trends for design convergence in soc: An eda perspective." Vivek Bhardwaj. *Shift Left Trends for Design Convergence in SOC: An EDA Perspective*. *International Journal of Computer Applications* 174.16 (2021): 22-27.
- [25] Vanapalli, Satyendra Kumar. "AI-Driven Fraud Detection in Financial Systems: A CRM-Centric Approach." *International Journal of Machine Learning and Predictive Analytics* 5.2 (2022): 01-14.
- [26] Kumar Doodala, A. N. (2022). Strategic Migration for JBoss to IIBM WAS: A Framework for Enterprise-Grade Modernization. *International Journal of Emerging Research in Engineering and Technology*, 3(2), 161-170. <https://doi.org/10.63282/3050-922X.IJERET-V3I2P117>
- [27] Muppaneni, R. K. (2022). Data Privacy in the Age of AI: How Dynamics 365 Handles Regulatory Challenges. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, 3(4), 159-170. <https://doi.org/10.63282/3050-9262.IJAIDSML-V3I4P117>
- [28] Mohan, Prashanth. *Soft Embedded FPGA Fabrics: Top-down Physical Design & Applications*. Diss. Carnegie Mellon University, 2022.
- [29] Kommuru, Madhurima. "Concurrency is hard: Java vs Python pitfalls you can't ignore." *International Journal of Modern Research in Science & Engineering* 5.2 (2022): 01-18.
- [30] Katangoori, S., & Katangoori, A. (2021). AI-Augmented Data Governance: Enabling Intelligent Access, Lineage, and Compliance across Hybrid Clouds. *American International Journal of Computer Science and Technology*, 3(6), 36-45. <https://doi.org/10.63282/3117-5481/AIJCSST-V3I6P104>
- [31] Fadiheh, Mohammad Rahmani, et al. "An exhaustive approach to detecting transient execution side channels in RTL designs of processors." *IEEE Transactions on Computers* 72.1 (2022): 222-235.
- [32] Srigadde, B. R. (2021). Future Methods, Most Underrated Apex Features. *American International Journal of Computer Science and Technology*, 3(1), 35-45. <https://doi.org/10.63282/3117-5481/AIJCSST-V3I1P104>
- [33] Zamiri Azar, Kimia, et al. "Basics of VLSI Design." *Understanding Logic Locking*. Cham: Springer International Publishing, 2023. 1-23.
- [34] Gaddam, R. R. (2022). Cost-Aware Autoscaling for Batch vs. Online Inference. *International Journal of Emerging Trends in Computer Science and Information Technology*, 3(4), 134-143. <https://doi.org/10.63282/3050-9246.IJETCSIT-V3I4P113>
- [35] Vppalapati, M., & Talasila, P. K. (2022). Correlated Independence: Why Redundant Storage Systems Share the Same Fate. *International Journal of Emerging Trends in Computer Science and Information Technology*, 3(1), 169-179. <https://doi.org/10.63282/3050-9246.IJETCSIT-V3I1P119>
- [36] anish Donthi, Raj. *Low Power ASIC Design of RISC Core Using 90NM Technology*. Diss. CALIFORNIA STATE UNIVERSITY NORTHRIDGE, 2008.

-
- [37] Parakala, A. (2022). Integrating Salesforce and UiPath: Cross-System Intelligent Automation. *International Journal of Emerging Trends in Computer Science and Information Technology*, 3(4), 88-99. <https://doi.org/10.63282/3050-9246.IJETCSIT-V3I4P109>
 - [38] Allenki, S. S. (2022). Securing Databases in the Cloud with RBAC and Encryption Best Practices. *International Journal of Emerging Research in Engineering and Technology*, 3(3), 173-182. <https://doi.org/10.63282/3050-922X.IJERET-V3I3P117>
 - [39] Meng, Xingyu. *Ensuring hardware robustness via security verification*. The University of Texas at Dallas, 2023.
 - [40] Vanapalli, Satyendra Kumar. "Bridging Business and Technology: The Role of CRM in Digital Transformation." *International Journal of Artificial Intelligence & Digital Transformation* 5.2 (2022): 01-17.
 - [41] Srigadde, B. R. (2021). When Rounding Up Matters: Working with Decimals in Apex. *International Journal of AI, BigData, Computational and Management Studies*, 2(1), 122-131. <https://doi.org/10.63282/3050-9416.IJAIBDCMS-V2I1P113>
 - [42] Pulli, Adithya, and Matteo Lupi. "A simulation methodology for verification of transient fault tolerance of ASICs designed for high-energy physics experiments." *Journal of Instrumentation* 18.01 (2023): C01038.
 - [43] Taluri, R. (2022). Cloud Data Engineering Strategies for Large-Scale Financial Data Integration and Intelligent Corporate Performance Reporting. *International Journal of Emerging Research in Engineering and Technology*, 3(4), 176-188. <https://doi.org/10.63282/3050-922X.IJERET-V3I4P119>