Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 47 No. 06
June 2026

Pre-Silicon Validation of Clock Domains and Distribution System in Server SoC

Sahana M S, Dr.Srividya P,Tejinder Syan

Department of EC RV College of Engineering

Bangalore, India

Department of ECE RV College of Engineering

Bangalore, India

Silicon Validation Engineering Manager AI SoC Validation

Intel, Bangalore, India

***Abstract*—**Advancements in technology have enabled the in- tegration of complex multi-million gate designs into compact System-on-Chip (SoC) solutions. SoCs offer numerous advan- tages, including cost efficiency, reduced power consumption, enhanced performance, smaller form factors, and improved security. However, the growing complexity and shrinking geome- tries of SoCs have introduced significant challenges in design and validation. Traditional simulation methods are no longer adequate for comprehensive system-level verification, necessi- tating more robust approaches. This paper presents a detailed implementation of a SystemVerilog testbench for SoC verification, along with methodologies aimed at reducing verification time while ensuring accuracy.

Index Terms—System-on-Chip (SoC), Phase-Locked Loop (PLL), Subsystems, Clock Validation, Server SoC, Verification, RTL

INTRODUCTION

SoCs have become indispensable in embedded systems, addressing critical requirements such as cost, size, and power efficiency. However, their increasing complexity presents new challenges in design and validation. Unlike traditional ASIC designs, SoCs emphasize IP reuse, platform-based design, and embedded software integration. Early SoC models often faced connectivity issues during RTL development, leading to mismatches in integration. To mitigate these challenges, verification teams require models that facilitate rapid construc- tion, execution, and debugging. A key focus of this study is clock validation, a fundamental aspect of SoC design that influences IP integration and system performance. Different IPs within an SoC may demand varying clock frequencies, necessitating a standardized approach to clock distribution and control. This paper outlines a methodology to validate clock domains in server SoCs, aiming to reduce integration time and power consumption while ensuring synchronization across subsystems. Clocking architecture in SoCs involves group- ing IPs based on their frequency requirements. An external reference clock (e.g., crystal oscillator) drives a PLL, which generates the primary clock (”pll clk”). The clock generation block then produces multiple outputs with configurable phases, multiplexing, and gating features. Figure 1 illustrates this process, while Figure 2 depicts the server SoC model with subsystem connections verified using checkers.*PROBLEM STATEMENT*

The objective is to validate the clock behavior of various components within the server SoC. This ensures that all IP blocks receive properly synchronized and stable clocks. By performing this validation, we can identify and correct any inconsistencies in clock distribution. Ultimately, the goal is to standardize the clocking interface across the entire SoC for reliable integration.

A. SOLUTION

This paper presents a methodology for validating clock domains in server SoCs using an architectural

model provided by the design team. By completing and testing the clocking connections within this model, we were able to verify critical clocking features across the system. The proposed clock- ing architecture aims to streamline IP integration, accelerate SoC development cycles, and optimize power efficiency by establishing standardized clock distribution mechanisms and dynamic control methods. We believe this approach introduces novel verification techniques that distinguish it from conven- tional industry practices, offering a more efficient framework for clock domain validation in complex SoC designs. The methodology focuses on systematic validation of clock syn- chronization while addressing key challenges in modern SoC implementation.

CLOCK VALIDATION FOR SERVER SOC

Subsystem Validation and Clock Architecture in SoC De- sign The SoC architecture is composed of several subsystems, each designed and implemented as separate RTL components to support thorough and modular validation. Although the primary connections between major subsystems were suc- cessfully integrated and functionally verified, some of the internal linkages within smaller, grouped modules still needed additional testing.To streamline and enable effective valida- tion under these conditions, controlled reset sequences and specific data patterns were applied to the subsystems that were temporarily disconnected, ensuring all modules could be independently and reliably tested.

Modern SoCs integrate numerous IP blocks operating at varying clock frequencies. To optimize clock distribution, these IPs can be categorized into distinct groups based on their

frequency requirements.IPs with similar clocking needs are clustered together, resulting in approximately five to six logical groups within the SoC, depending on the design's complexity and the number of integrated IPs.

The clock generation architecture begins with an external reference clock source, such as a crystal oscillator, which drives the input of a phase-locked loop (PLL). Once configured to the target frequency, the PLL produces a stable output clock pll clk.

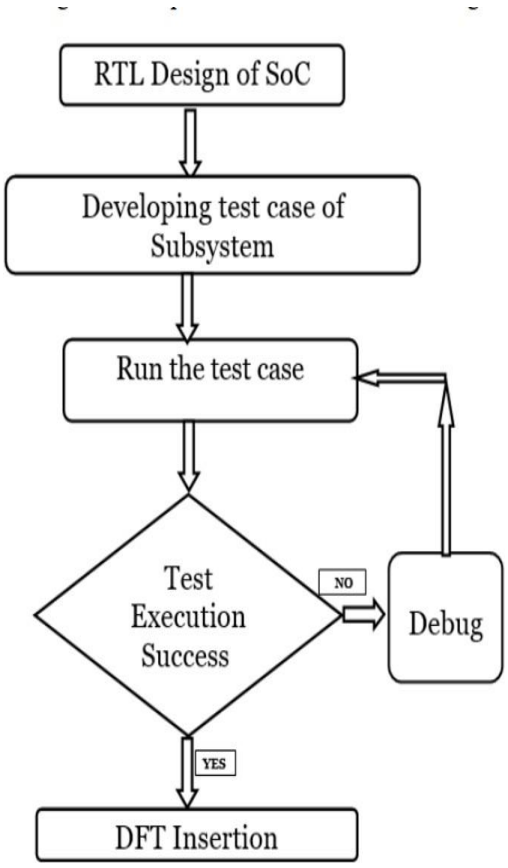


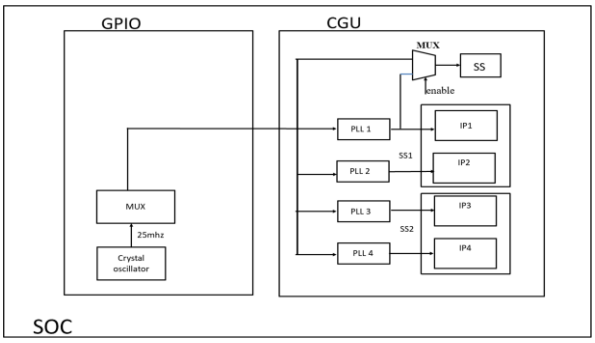
Fig. 1. Functional Verification Flow for RTL Subsystem

Subsequent clock generation circuitry then derives multiple clock signals with configurable properties, including phase alignment, multiplexing, and gating capabilities. These features may be shared or independently controlled across dif- ferent clock domains, as illustrated in Fig. 1. The architecture ensures flexible and efficient clock distribution while meeting the diverse timing requirements of various IP blocks within the SoC.

This structured approach to subsystem validation and clock management enhances both design reliability and performance, addressing critical challenges in complex SoC implementa- tions.

The Server SoC model is shown in Fig. 2 below, where subsystems connections are done with the help of checkers.

Fig. 2. Generalized Clock Architecture in SoC



The proposed verification methodology employs a register- based configuration approach to validate clock domains within the SoC architecture. This technique involves two primarytest modes: targeted subsystem verification and system-wide broadcast testing. During verification, the Device Under Test DUT is exercised through carefully designed register write operations that follow a comprehensive test plan. These op- erations specifically validate the functionality of individual RTL subsystem components while maintaining system-level integrity.

For subsystem-specific validation, we implemented focused test sequences that exclusively addressed particular RTL mod- ules. This granular approach allowed for thorough examination of clock domain interactions at the subsystem level. The methodology proved effective, as demonstrated by successful register configuration and subsequent reception of expected response patterns from all targeted RTL subsystems. These responses confirmed proper clock synchronization and data integrity across the verification framework.

The verification process incorporates compliant practices for clock domain crossing CDC validation, ensuring robust timing closure and metastability protection. By employing this systematic approach, we achieved comprehensive coverage of clock-related functionality while maintaining the rigorous standards required for complex SoC designs. The method- ology's effectiveness is particularly evident in its ability to isolate and verify subsystem-specific clock behavior while preserving overall system synchronization requirements..

I. EXPERIMENTS ON SOC MODEL

We ran a few clock test cases on the SoC model to verify the clocks synchronization in the SoC. Also, we ran the tests in the subsystems of the SoC as shown in Fig 3. Subsystem 1, Subsystem 2, subsysetm3 like this till the nth Subsystem are verified depending upon their requirements in the SoC model. Each subsystem requires different sets of frequencies , this can be generated from the PLL Phase Locked-Loop. Capturing waveforms is crucial for verification in order to analyse test failures and verify the functionality of various subsystem components. Generally, capturing waveforms comes after the completion of the run.

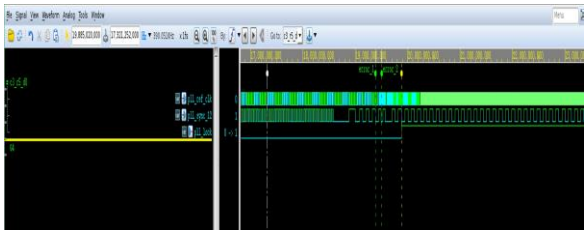


Fig. 3. Phased Locked Loop in Subsystems

The presented SoC clocking architecture is designed to ensure reliable, flexible, and synchronized clock distribution to various Intellectual Property (IP) blocks within the system. As illustrated in the block diagram, the system begins with a crystal oscillator that serves as the primary reference clock. This clock is routed through a multiplexer MUX within the General-Purpose Input/Output GPIO block, which selects the

appropriate source for feeding into the Clock Generation Unit CGU.

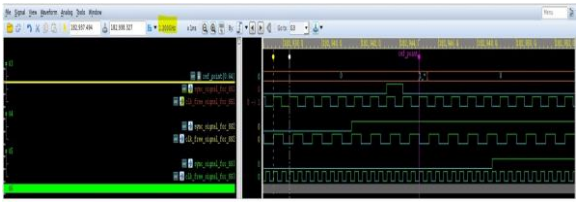


Fig. 4. sync signal for Subsystems

Within the CGU, the reference clock is distributed to four distinct Phase-Locked Loops PLLs, each configured to generate different frequencies or

phase-shifted outputs suitable for various IP blocks. PLL1 and PLL2 feed Subsystem 1 SS1, which in turn distributes clock signals to IP1 and IP2, while PLL3 and PLL4 support Subsystem 2 SS2, delivering clocks to IP3 and IP4. A control MUX selects the active clock path based on the enable signal, and it interfaces with a supervisory control block SS, which ensures proper clock gating and switching as required. The first waveform diagram provides insight into synchronization logic among signals. It shows that the sync signal operates in a repetitive cycle, and clock control signals toggle in coordination with the sync pattern, indicating the proper switching or enabling of clocks to subsystems or IPs based on functional requirements.

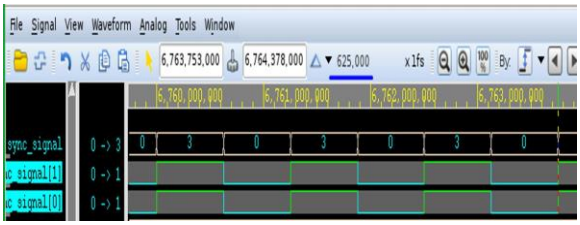


Fig. 5. Duty Cycle of Clock Signal Validation

Together, these diagrams validate the operational behav- ior of clock distribution in the SoC—from oscillator sourc- ing, PLL-based frequency synthesis, subsystem clock rout- ing, and synchronization—to error detection and correction during clock handovers. This systematic validation ensures the robustness of clocking architecture, which is vital for the correct operation of all IP modules within a server-grade SoC. The subsystem_1 requires the two clock cycles early with respect to reference point this can be analyzed in through the waveform as shown in Fig.4. Similarly the Subsystem2 and subsystem_3 requires the four clock cycles early and eight clock cycles delayed respectively as shown in Table 1. And also subsystem_1 requires 1GHz frequncy, subsystem_2 requires 1.2 GHz frequency and subsystem_3 requires 2.4 GHz frequency as shown in Table 1.

TABLE ISUBSYSTEM CLOCK REQUIREMENTS

Subsystem	Frequency Requirement	Clock Cycle Requirement
Subsystem 1	1 GHz	2 cycles early
Subsystem 2	1.2 GHz	4 cycles early
Subsystem 3	2.4 GHz	8 cycles delayed

CONCLUSION

SoC technology has emerged as a critical solution for addressing key requirements including cost optimization, form factor reduction, and power efficiency. The increasing scale and complexity of modern SoCs present both significant opportunities and substantial challenges in both design imple- mentation and verification processes. A fundamental distinc- tion between SoC and traditional ASIC design methodologies lies in three critical aspects: intellectual property (IP) reuse strategies, platform-based design approaches, and embedded software integration challenges.Modern SoC architectures typ- ically incorporate numerous embedded processor cores, shift- ing the design focus toward comprehensive IP verification and platform-specific software debugging. The evolution of SoC design methodology began with early implementations that initiated the product development cycle at the RTL design phase [1]–[12].

REFERENCES

[1] I. Consortium, “Server soc clock validation strategy,” *Journal of Semi- conductor Validation*, vol. 12, no. 3, pp. 45–52, 2017.

[2] P. Mohandoss and A. Rengaraj, “Pre-silicon dft verification on soc slim model,” in *2018 19th International Workshop on Microprocessor and SOC Test and Verification (MTV)*, 2018, pp. 76–78.

[3] M. Jiao, D. Wang, and J. Qiu, “A gru rnn based momentum optimized algorithm for soc estimation,” *Journal of Power Sources*, vol. 459, p. 228051, 2020.

[4] F. S. Gorostiza and F. M. Gonzalez Longatt,

"Deep reinforcement learning-based controller for soc management of multi-electrical energy storage system," *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 5039–5050, 2020.

[5] M. Bansal and R. R. Maiya, "Phototransistor: The story so far," *Journal of Electronics*, vol. 2, no. 04, pp. 202–210, 2020.

[6] D. Nirmal, "High performance flexible nanoparticles based organic electronics," *Journal of Electronics and Informatics*, vol. 1, no. 1, pp. 99–110, 2019.

[7] J. Smith and A. Brown, "Advanced clock domain crossing verification techniques," in *2016 IEEE International Test Conference (ITC)*. IEEE, 2016, pp. 1–10.

[8] R. Jones and E. Wilson, "Low-power clock distribution networks for soc designs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 4, pp. 712–725, 2018.

[9] D. Lee and W. Chen, "Machine learning approaches for soc verification," in *2019 56th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2019, pp. 1–6.

[10] X. Chen and L. Zhang, "Formal methods for clock domain verification in complex socs," *ACM Transactions on Design Automation of Electronic Systems*, vol. 26, no. 2, pp. 1–25, 2021.

[11] R. Patel and S. Kumar, "Automated clock gating verification for low- power socs," in *2020 IEEE 38th VLSI Test Symposium (VTS)*. IEEE, 2020, pp. 1–6.

[12] S. Wilson and M. Taylor, "Challenges in pre-silicon validation of modern processor designs," *IEEE Design & Test*, vol. 34, no. 5, pp. 68–77, 2017.