



Robust Verification of Clock Tree Network using “Clock Monitor” Integrated by ACRMG

Tejas Dipakkumar Dalal
Samsung Semiconductor
India Research
Bengaluru, India
(tejas.dalal@samsung.com)

Giridhar S
Samsung Semiconductor
India Research
Bengaluru, India
(giridhar.s@samsung.com)

Jeevan Nataraju
Samsung Semiconductor
India Research
Bengaluru, India
(j.nataraju@samsung.com)

Garima Srivastava
Samsung Semiconductor
India Research
Bengaluru, India
(s.garima@samsung.com)

Abstract-Clock Tree Network (CTN) is one of the complex design structures in SOC that controls clock supply for all the synchronized logic on the chip. It comprises numerous clock component instances enabled with dynamic control to establish Dynamic Voltage Frequency Scaling (DVFS) and Hardware Controlled Auto Clock Gating (HWACG) schemes, effectively contributing to overall performance and power optimization [1]. Due to increasing complexity and stringent performance requirements, the verification of the CTN has also become more intricate. This paper presents a robust verification strategy with Clock Monitor (CLKMON), a custom verification module developed to monitor, analyze and validate dynamic nature of CTN. “Accelerated Clock Reference Model Generator” (ACRMG), an in-house tool developed to automate CTN analysis [2], is used to integrate the CLKMON for all the IP clocks. The paper also presents how CLKMON is integrated with various SOCs having different CTN topology.

I. INTRODUCTION

The global development and the swift technological progression have led to a profound transformation in semiconductor industry, reshaping how chips are designed, manufactured and utilized. The industry is shifting towards new paradigms such as heterogeneous integration and chiplet based design allowing multiple and varied functionalities to be packed into a single package. With this increasing sophistication, the SOC design and development is undergoing rapid advancement to meet the Power, Performance and the Area (PPA) requirements.

The Clock Tree Network (CTN) is one such integral part of the SoC, comprising numerous clock components that drive all synchronous logic in modern digital designs while effectively optimizing power and performance. The CTN is known for its significant contribution in power dissipation on the clock nets, accounting for nearly 40% of the total power dissipation [1]. Additionally, the network is a complex structure, essentially spanning across the chip, with nearly 3000 plus clock component instances in a premium class mobile SOC [1]. The latest design trends enable the network with diverse functional features to integrate hardware driven control mechanism to incorporate auto clock gating and dynamic frequency scaling schemes. Each clock component is enabled with control to automatically avoid unnecessary transition on any clock net obtaining an overall power benefit of 17% to 50% [1]. Hence, the scale of the CTN along with its constituting components and diverse functional features, together classify it as one of the complex and critical designs.

This evolution in CTN design complexity and hardware schemes has made conventional verification techniques more intricate, necessitating advanced and scalable methodologies to ensure quality and reliability. The paper highlights the significance of one such custom model called Clock Monitor (CLKMON), developed to qualify SOC Clock Tree Network (CTN). The CLKMON is constructed with customized verification strategy to serve as a unified solution addressing the challenges associated with the legacy approach. It is integrated using “Accelerated Clock Reference model Generator” (ACRMG), an inhouse tool developed to automate the CTN analysis [2]. The paper presents:

- 1) Latest CTN design trends: HWACG and DVFS
- 2) Limitations with conventional verification methodology
- 3) Operational flow of the Clock Monitor (CLKMON)

It discusses the integration of CLKMON using ACRMG, its illustration, and further emphasizes the need for the same. The results indicated qualify the potential and scalability of the implementation.

II. ARCHITECTURE DESCRIPTION

A. CTN Architecture

A typical SOC constitutes multiple Sub-Systems (SS) based on the target application. Each of the SS includes dedicated Clock Control Unit (CCU) comprising clock source (PLL, OSCCLK), router (multiplexer, divider), Q-Channel Manager (QCH_MNGR) and clock gate components to supply desired clock frequency to all the IPs of the SS [2][3]. Such CCUs for each SS together constitute SOC CTN as shown in Fig. 1. A simplified representation of an example CTN for a single IP is shown in Fig. 2.

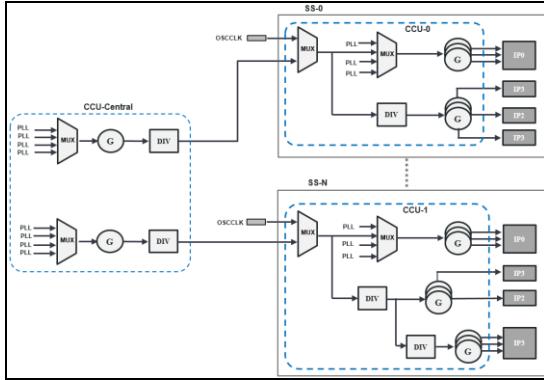


Figure 1. Example Clock Tree Network

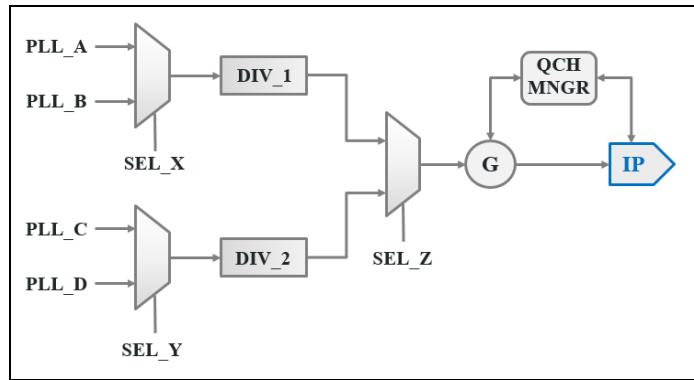


Figure 2. Example Clock Tree of an IP

B. Hardware Auto Clock Gate (HWACG)

The CTN design includes hardware-controlled mechanism to incorporate auto clock gating based on the IP activity, effectively contributing to dynamic power reduction. As indicated in Fig. 2, the QCH_MNGR associated with the target IP monitors the IP activity and gates the clock supply by controlling the CLKGATE component when the IP is idle. The recent design involves hardware driven fully automatic clock gating scheme that adds similar control mechanism on every connecting arc, enabling all the CTN components (MUX, DIVIDER, PLL) in the network to cut off its output clock when there is no activity [1].

C. Dynamic Voltage Frequency Scaling (DVFS)

The CTN design incorporates frequency scaling based on the performance requirement as part DVFS scheme, effectively contributing to reduction of dynamic power dissipation. The dynamic power dissipation (P) is proportional to the square of the supply voltage (V), and the operating frequency (f) as in (1) [4].

$$P \propto V^2 f \quad (1)$$

By scaling frequency with varying workload, the power dissipation reduces linearly with the workload. Further, maximum possible energy is saved by adapting both the supply voltage and the operating frequency to match the workload such that the performance loss is minimized. A simplified example of CTN with DVFS is shown in Fig. 2, where the operating frequency of the IP can be dynamically changed by configuring the Multiplexers (SEL_X, SEL_Y and SEL_Z), Divider ratios (DIV_1 and DIV_2) and the PLL source (PLL_A, PLL_B, PLL_C and PLL_D), for different DVFS corners.

III. METHODOLOGY

A. Limitations with conventional flow

The conventional or legacy verification flow was determined to be an unreliable approach limiting comprehensive CTN verification and significantly hindering verification cycle due to the following drawbacks that compromises the verification quality:

- 1) Unguaranteed network integration, due to restricted connectivity checks.
- 2) Directed checks to qualify IP clock frequency missing out on dynamic nature of the network.
- 3) Extended manual review to realize test bench for such a vast network, further prone to human error.
- 4) Verification scope limited to targeted test features only.
- 5) Limited functional coverage due to design complexity.

B. CLKMON (Clock Monitor)

CLKMON is a custom verification module developed to monitor the target IP clock frequency and its associated component configurations. It is integrated with network information that enables it to sample and process real time data, which makes it more suitable for HWACG and DVFS scheme verification.

The Clock Reference Model (CRM) includes CLKMON for every IP in the SOC as shown in Fig. 3:

- 1) To monitor and ensure the IP clock frequency is in-accordance with the associated component configuration.
- 2) To ensure that the clock is gated/ungated based on the IP activity.
- 3) To ensure sign-off frequency is in accordance with the selected DVFS corner.

C. Implementation

ACRMG tool automates the CRM generation by analyzing every branch, node and associated feature of CTN [2]. As shown in Fig. 3, it parses the CTN specification and processes the data using custom algorithm, to integrate the CLKMON for all the IP clocks. It also generates DVFS and HWACG cover-groups for the corresponding IP Clocks. The CTN specification encapsulates the network component details, connectivity, supported features and configurations in a pre-defined format. The tool is validated to generate the CRM on different target SOCs (Mobile and Automotive) and multi-die/chiplet based architecture, signifying its scalable nature [2].

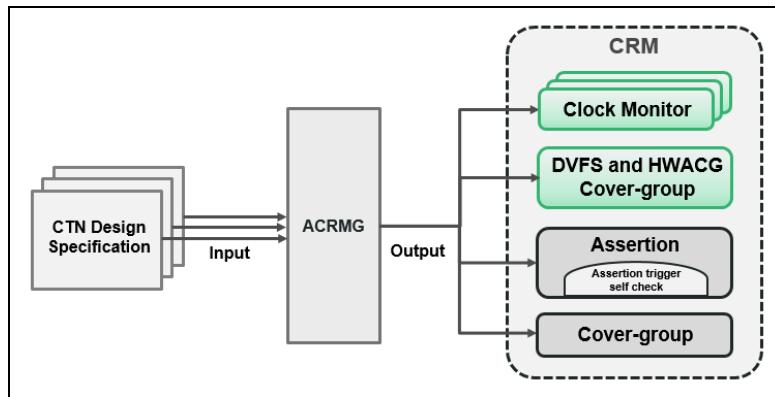


Figure 3. ACRMG Execution Flow

D. Operational Flow

Fig. 4 represents the operational flow of the CLKMON module. The module is sensitive to clock component configuration associated with the IP clock under supervision. Any change in this component configuration will initiate the processing. The module fetches its immediate parent component details driving the IP clock, and further traces the active branch of the CTN to find the clock source component. It samples the RTL configurations of all the parent components in the active branch and uses it to calculate the expected frequency, and additionally fetch DVFS frequency from the specification. The sampled data enables it to check if there is any active clock request allowing it to compare the calculated and DVFS frequency, with the actual frequency. If there is no active clock request, it checks whether the clock is gated or not. Any mismatch in the check or comparison is logged as an error converging the test to fail.

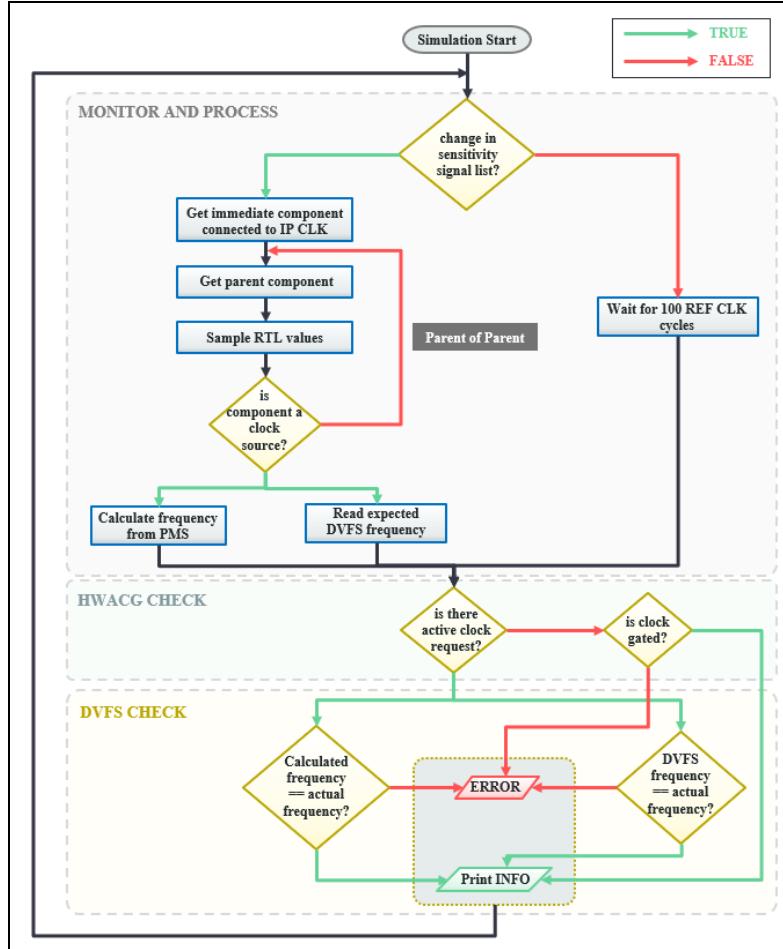


Figure 4. Clock Monitor Operational Flow

IV. ILLUSTRATION

A. CLKMON Structure and Execution

The CLKMON module is structured with network details into 3 primary sections as follows, to facilitate real time processing and dynamic checks:

- 1) Parameter details: It captures the basic design parameters such as the IP clock name, clock group information, supported frequencies and the number of different clock components in the active branch of the CTN associated with the IP.
- 2) Sensitivity and control signals: It captures the list of signals responsible for different clock configurations, along with RTL hierarchies for the IP clock sensitivity list.
- 3) Network connectivity details: The component characteristics and their connectivity details are classified based on the component category, to include its output clock port, its immediate parent component port, the control signals and the configuration values for all the supported DVFS levels.

A small subset of an example CTN for an SOC SS is shown in Fig. 5. All the IPs of this SS (IP_1 to IP_5) receive clock supply from a dedicated branch in the CTN, routed from the main PLL sources (PLL_1 to PLL_5). The network spans across the Local CCU that is part of the sub-system, all the way to the Central CCU.

The CLKMON bind generated by ACRMG for IP_2 of this CTN subset would follow the standard 3 section structure. Section 1 includes the parameter details for IP_2 clock port. Section 2 captures the configuration and sensitivity signal list. Section 1 and Section 2, together are responsible for CLKMON activation and monitoring.

The component characteristics and interconnectivity belong to Section 3, corresponding to the order represented in Fig. 5 from C1 to C8 (red colored dotted line) indicating the connectivity from IP_2 leaf node to its clock source. The component port and the parent port information, part of Section 3 aids the back-tracing mechanism of the CLKMON, starting with the Gate Component G4 (C1) back till the PLL source (C8). While the configuration signals part of every component enables real time operating frequency and state sampling, the specification details bound with the module provides the expected results, and also aids theoretical calculation. Collectively, they ensure all the necessary data availability for the dynamic comparison checks.

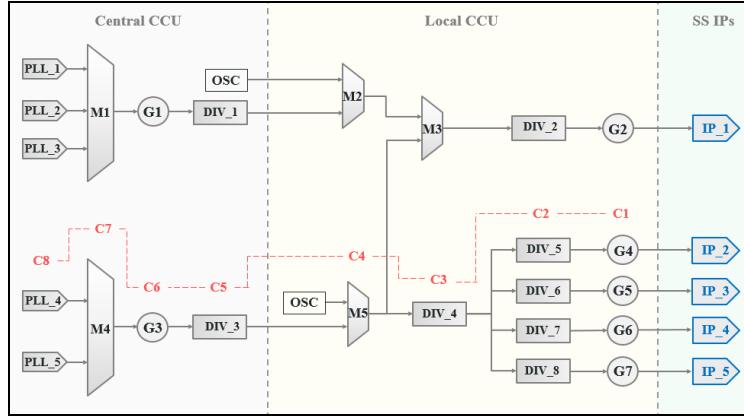


Figure 5. Example CTN of a Sub-System

B. Functional Coverage

ACRMG tool generates functional cover-groups for HWACG and DVFS feature coverage:

- 1) **HWACG Coverage:** The component configuration signals associated with the QCH_MNGR and the CLKGATE components are used to define HWACG functional cover-groups for every IP. As the controllability over these components enable dynamic switching between auto clock gating and manual mode, the cover-groups are targeted to cover if all the IP clocks are exercised for their activity in all the supported configurations.
- 2) **DVFS Coverage:** Though the CTN component configuration aids dynamic frequency change of any IP clock, the frequency scaling as part of DVFS is not targeted for an individual IP. From SOC perspective, different SSs are integrated together to operate in accordance with other SSs, to realize overall SOC functionality. Hence, the frequency scaling is achieved considering real time applicability and design factors such as the supported voltage rails and DVFS levels of different SSs. This analysis would be concised to a set of scenarios involving different voltage rails in different DVFS levels, that are adequate for the network to achieve all valid combinations. The DVFS coverage targets the same to check if all the IP clock frequencies in different corners are validated, and if they are exercised for all the valid combinations.

V. RESULTS

A. Real time network analysis

Fig. 6 and Fig. 7 represent an example scenario for clock activity monitoring and failure detection. The “clock_active” and the “clock_stable” signals are the processed output of CLKMON module for IP clock “CLK” under supervision, indicating active clock request and stable configuration status respectively. At timestamp M1, there is no active clock request processed by the module, but the IP “CLK” remains to be active indicating a network bug. This inconsistency is detected and reported as shown at timestamp M2.

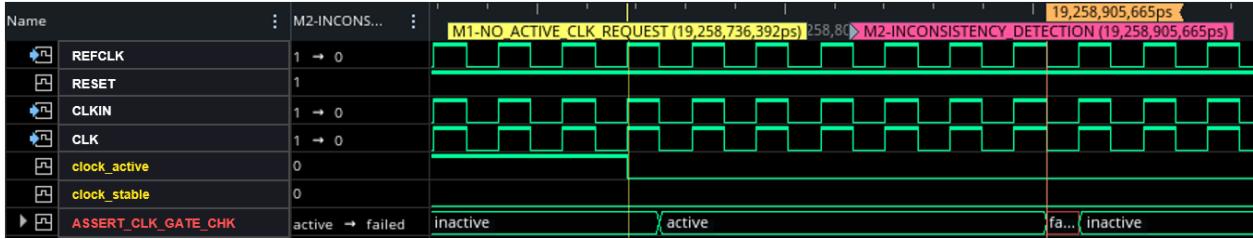


Figure 6. CLKMON clock inconsistency detection

```
xmsim: *E,ASRTST (clk_mon_bind,249):
(time 19258905665 PS) Assertion incr_top.clk_mon_bind.SS_1.IP_1.CLK ASSERT_CLK_GATE_CHK has failed
[ASSERT_CLK_GATE_CHK] : Clock :[CCU_SS_1_IP_1_CLK] should be Gated
```

Figure 7. CLKMON clock inconsistency report

B. Comprehensive check and report mechanism

Fig. 8 represents an example scenario for IP clock frequency mismatch detection and detailed reporting mechanism. The incorrect network configuration or a design bug sets the IP to DVFS corner DVFS_LVL_6, while the expected corner is DVFS_LVL_5, which is captured by the CLKMON as an error. The CLKMON failure report for a frequency mismatch includes:

- 1) Supported DVFS corners for the IP clock under check
- 2) Expected DVFS corner vs the actual corner
- 3) Expected frequency range vs the actual frequency

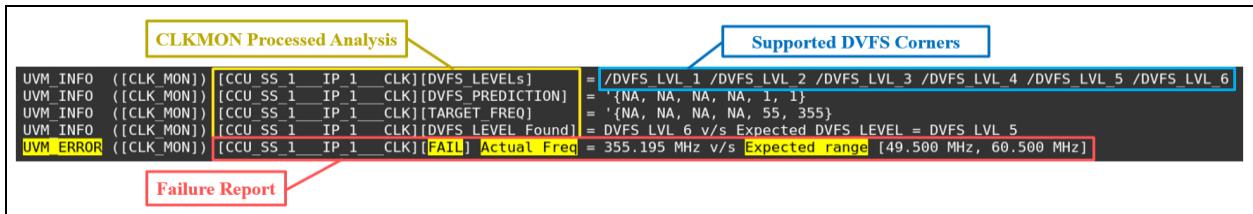


Figure 8. CLKMON frequency mismatch detection report

C. Holistic Coverage

Fig. 9 and Fig. 10 show the initial coverage report generated for a mid-range mobile SOC CTN, with nearly 780+ DVFS and 3.4K+ HWACG cover bins for 1.8K+ CLKMON instances. The functional cover-bins cover all the supported DVFS corners and the HWACG feature for all the IPs in the SOC. The CLKMON instance number corresponds to the number of target IP clocks in the SOC, bound for monitoring.

The uncovered bins indicate the feature coverage miss for a set of IP clocks. Fig. 11 and Fig. 12 represent the coverage closure data retrieved from follow-up regressions to exercise IP clocks for the uncovered features.

Name	Overall Average Grade	Overall Covered
Overall	94.29%	781 / 787 (99.24%)
Code	n/a	0 / 0 (n/a)
FSM	n/a	0 / 0 (n/a)
Functional	94.29%	781 / 787 (99.24%)
Assertion	n/a	0 / 0 (n/a)
CoverGroup	94.29%	781 / 787 (99.24%)
FaultNode	n/a	0 / 0 (n/a)

Figure 9. DVFS coverage (with first regression)

Name	Overall Average Grade	Overall Covered
Overall	89.74%	3409 / 3819 (89.26%)
Code	n/a	0 / 0 (n/a)
FSM	n/a	0 / 0 (n/a)
Functional	89.74%	3409 / 3819 (89.26%)
Assertion	n/a	0 / 0 (n/a)
CoverGroup	89.74%	3409 / 3819 (89.26%)
FaultNode	n/a	0 / 0 (n/a)

Figure 10. HWACG coverage (with first regression)

Name	Overall Average Grade	Overall Covered
Overall	100%	787 / 787 (100%)
Code	n/a	0 / 0 (n/a)
FSM	n/a	0 / 0 (n/a)
Functional	100%	787 / 787 (100%)
Assertion	n/a	0 / 0 (n/a)
CoverGroup	100%	787 / 787 (100%)
FaultNode	n/a	0 / 0 (n/a)

Figure 11. DVFS coverage (with final regression)

Name	Overall Average Grade	Overall Covered
Overall	100%	3819 / 3819 (100%)
Code	n/a	0 / 0 (n/a)
FSM	n/a	0 / 0 (n/a)
Functional	100%	3819 / 3819 (100%)
Assertion	n/a	0 / 0 (n/a)
CoverGroup	100%	3819 / 3819 (100%)
FaultNode	n/a	0 / 0 (n/a)

Figure 12. HWACG coverage (with final regression)

D. Bug Findings

CLKMON integration contributed towards catching both architectural and configuration bugs as listed below:

- 1) Network relation mismatch: Fig. 13 represents an example illustration of critical network relation bug, where the Multiplexer output of M1 and M2 are erroneously cross connected to different branches of the CTN as indicated in red colored dotted lines. The target IPs, IP_1 and IP_2 are considered to be powered with two different voltage rails VDD_1 and VDD_2 respectively. In normal operating conditions, both these IPs would work on a common DVFS corner and matching target frequencies if the component configurations are same, resulting in this bug to go unnoticed. However, due to the independent voltage rails, both the IPs are likely to be configured to different DVFS corners DVFS_LVL_1 and DVFS_LVL_2 simultaneously, driving unintended frequencies to the destination IPs due to incorrect PLL source frequency. Additionally, when either of the voltage-rails is locally powered-down, the cross connected MUX and PLL fail to drive the clock to the active IP as they are potentially turned off. Such mismatches were identified very early with the CLKMONs network back-tracing ability, reporting an inconsistent clock activity based on parent component state.

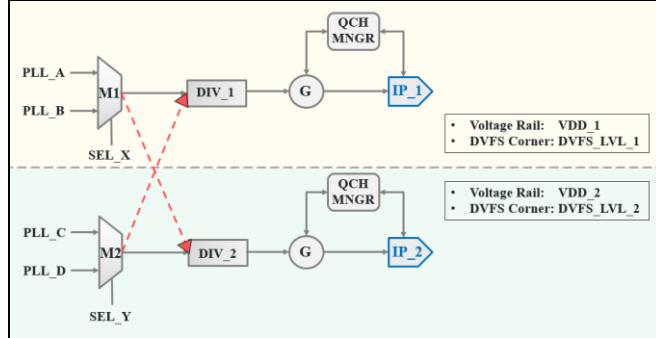


Figure 13. Example CTN relation bug

- 2) CTN Component Configuration Signoff: The dynamic frequency checks were able to identify incorrect component configurations in different DVFS levels very early in verification cycle. The detailed real time analysis, along with its reporting mechanism facilitated easy root cause of the malfunctioned component and aided early configuration signoff.

E. Deployment Study – Bug Detection and Simulation Overhead

Fig. 14 and Fig. 15 represent the bug detection and simulation overhead aspects along with the scalability of CLKMON integrated to different SOCs with increasing design size and complexity. The bar graphs on the plot represent the total number of CLKMON instances (Blue) and overall HWACG and DVFS cover bins (Green). The line graphs correspond to bug detection and simulation time in Fig. 14 and Fig. 15 respectively, for both with CLKMON (solid line) and without CLKMON (dotted line).

The gradual increase in bug detection in Fig. 14 indicates the CLKMON’s efficiency to scale with design complexity. Following reduction in bug detection deviating from predicted bug rate signifies the improvement in design quality with CLKMON integrated for derivative projects, and also the ability of the CLKMON to detect and report multiple issues in a single simulation. The real-time network back-tracing ability of the CLKMON allows it

to root-cause and report string of inconsistencies on every branch of the CTN at once (represented with solid line), preventing delayed and progressive identification of connected issues and duplicate issue reports without CLKMON (represented with dotted line). This significantly accelerates bug resolution turn-around time, in turn contributing to reduced design and verification execution cycles. These gains are achieved with less than 25% of increased simulation time for real time CLKMON processing as shown in Fig. 15.

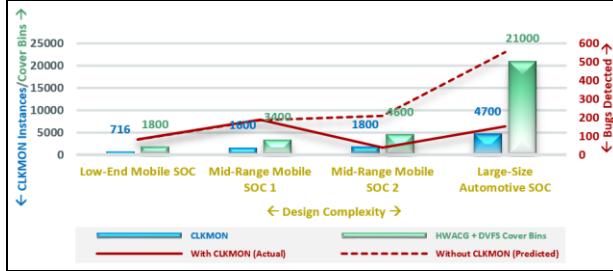


Figure 14. Design Complexity vs Bug Detection

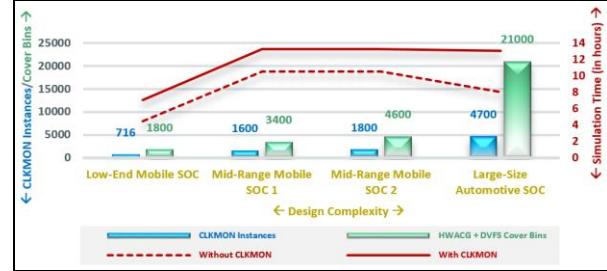


Figure 15. Design Complexity vs Simulation Overhead

VI. CONCLUSION

In the rapidly evolving semiconductor industry and fast-paced progression in design methodologies for diverse applications driving mobile, automotive and networking sectors, it is critical to ensure the performance and quality of the product without impacting the time to market. The proposed CLKMON solution adapts to this swift transition that demands innovative verification techniques and methodologies. It illustrates the importance and efficiency of advanced strategy in CTN verification, addressing the limitations of the legacy approach.

In this paper, we showed the complexity and sophistication of the CTN in SOC architecture. With the CLKMON integration, every IP clock of the SOC along with associated branch of the CTN is monitored throughout the simulation validating the network integrity for DVFS and HWACG schemes. Additionally, the functional coverage generated for the same ensures the feature inclusion of all the CTN components and the network as a whole, boosting verification sign-off. CLKMON was integrated by default across all the functional scenarios, and was validated for both mobile and automotive SOCs. The deployment results presented indicate the efficiency of the comprehensive strategy and the scalability of the solution. The rigorous dynamic checks, holistic coverage and scalable integration makes it one of the practical and robust solutions for the latest CTN design trends.

REFERENCES

- [1] J. -G. Lee, Y. Choi, H. Jeon, J. -J. Lee and D. Shin, "Fully Automated Hardware-Driven Clock-Gating Architecture With Complete Clock Coverage for 4 nm Exynos Mobile SOC", in IEEE Journal of Solid-State Circuits, vol. 58, no. 1, pp. 90-101, Jan. 2023
- [2] Tejas Dipakkumar Dalal, Giridhar S, Jeevan Nataraju, Garima Srivastava, "Quality Driven Analysis of Clock Tree Network using Accelerated Clock Reference Model Generator", in DVCN Japan, Aug. 2024, in press.
- [3] AMBA Low Power Interface Specification, ARM Standard IHI 0068C (ID091216)
- [4] Ajit Pal, *Low-Power VLSI Circuits and Systems*, Springer New Delhi, 2014