2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE
MUNICH, GERMANY
OCTOBER 15-16, 2024

Functional Coverage Sign-off assisted by Formal Connectivity

Asheque Mohammad Zaidi, Infineon Technologies AG, Neubiberg, Germany
(asheque.m.zaidi@infineon.com)

Muhammad Ul Haque Khan, Cadence Design Systems GmbH, Feldkirchen, Germany
(muhammad@cadence.com)

***Abstract*—With increasing complexity of Automotive Microcontrollers (MC) and the vigorous decreasing trend of time to market (TTM) the verification of MCs becomes much more challenging. As parallelization cannot help in signing off the verification faster, the only possibility is to optimize the verification strategies and methodologies. In such cases, reducing verification effort in coverage sign-off using automated formal connectivity checks can greatly decrease the sign-off time.**

Keywords—Microcontroller; Complexity increase; Functional verification; Formal verification; Connectivity checks; Automation; Coverage Sign-off

I. INTRODUCTION

Verification has always been the bottleneck of the semiconductor industry with about 70% of overall project development time spent only for verification [1]. More complex product requirements and fast TTM has made it important to search for faster and robust methodology to speed up the verification phase.

The increase in complex product requirements in digital electronics and hence exponential increase in design complexity [2] makes it even more difficult for verification to cope with the challenging timeline for chip tape-out. Hence the need for immediate improvement of verification methodology and strategy cannot be overlooked.

With the rise in usage of automated formal checks and connectivity verification [3], in this paper we investigate the usage of automated formal connectivity verification to verify internal connectivity subject to complex conditions in our project.

II. DESIGN

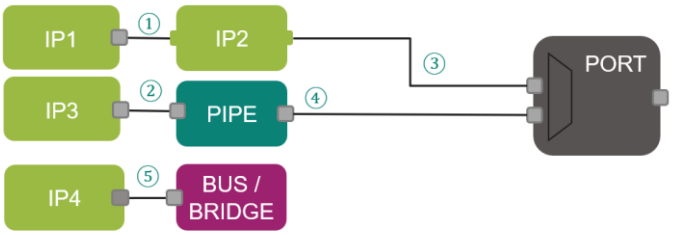

Fig. 1: Types of connections in SB

The design is a sub-system (SB) of MC System on Chip (SoC) with about 13000 connections between IP to IP(s), IP to pipeline stages, IP to bus or bridges and finally to ports. A stripped-down version of where formal connectivity has been applied within the MC SoC has been presented in Fig. 1. Since the focus is only on connectivity checks at the SB level, complete internal workings of the MC have not been described in detail.

III. DESIGN VERIFICATION

For verification sign off, toggle coverage is a primary target in SB level verification. As the IPs integrated within the SB comes pre-verified at IP level, the main task of verification at SB level is to make sure the connections during the integration have been done properly and that the features that encapsulates multiple IPs are working seamlessly. Since the target is a subset of all features of the individual IPs, it is quite evident that not all ports of each IP, bridge or port will toggle. Usually, an experienced verification engineer together with their team would need to look at the uncovered toggle coverage and then cross-refer with the scenarios stimulated at the SB level and IP verification toggle report to waive for coverage, which with the proposed methodology can be done automatically with formal connectivity checks with much less effort.

A. Formal Connectivity

Connectivity checking plays a vital role in verifying the correctness of interconnections between SBs. It involves ensuring that all signals are properly routed and connected according to the design specifications. Connectivity errors, such as missing or incorrect connections, can result in malfunctioning of the SoC and are often challenging to debug once the chip is fabricated. There are connections specific to SBs for example, pads and port feedthrough paths which require directed testing. This could amount to ~10000 signal connections each with directed test cases in some cases. Therefore, thorough connectivity checking during the integration phase helps identify and rectify such issues early in the design cycle, saving time and resources.

Moreover, connectivity checking complements other verification goals, such as simulation based functional verification and timing analysis. While simulation based functional verification focuses on ensuring that each SB performs its intended operations correctly and collects the corresponding structural toggle coverage, connectivity checking verifies that the interactions between SBs occur as expected. By incorporating connectivity checking into the overall verification strategy, SB verification engineers can achieve a more holistic verification approach, mitigating risks associated with integration.

B. Methodology improvements

A flow is required which would reduce or remove manual directed test cases writing or manual waiving of structural toggle coverage when verified at IP or SB level. This is quite a challenging task for verification team since every connection which is not already covered with functional tests must be categorised either as not verified at the SB level as it is already verified at IP level and then manually waiving them, or essential gap is identified, and a new functional test should now cover the structural toggle coverage item. Therefore, a general verification management cockpit that can integrate results from multiple verification engines is a prerequisite. Depending on the size and complexity of the SB, this manual waiving or writing test-cases can be weeks of effort from the verification team and this effort grows exponentially with the size and complexity of the SB. This is exactly where Formal Verification flow comes into the picture. Formal verification techniques offer a powerful means to address verification challenges related to integration, particularly in assessing toggle on connection points and ensuring completeness of connections. Toggle analysis is a critical aspect of verification coverage and requires a basic understanding of overlapping functionalities among IPs. Formal-based connectivity checking methods enable a debugging mechanism which is quite straight-forward, bypassing the complexity of the functionality of the system and allowing for minimal traces to be analyzed.

A further benefit of formal methods is in verifying the completeness of connections in the SoC design as there is a complete structural synthesized design. The completeness of formal is unlike simulation-based approaches, which rely on stimulus-driven testing; formal verification exhaustively explores all possible input combinations and system states to ensure that every connection is exercised and behaves as specified. This exhaustive analysis helps uncover corner-case scenarios and edge conditions that may be missed during simulation-based verification, thus enhancing the overall robustness and reliability of the integrated system. Furthermore, the formal tools can generate the connection maps directly by analyzing the design and hence scalable, which is one of the biggest advantages of using formal flow for such a use-case model. The generated maps which are then converted

automatically by the tool to assertion and toggle properties which are human readable and can be reviewed considerably faster than manually waiving the item as described above.

C. Proposed solution

By leveraging formal verification techniques alongside traditional verification methods, and the ability to easily incorporate this coverage information in a verification management cockpit can greatly improve DV productivity and provide higher degree of confidence in the functional correctness, performance, and reliability of the integrated system. The flow used to achieve faster sign-off of toggle coverage in SB level is described in Fig. 2.

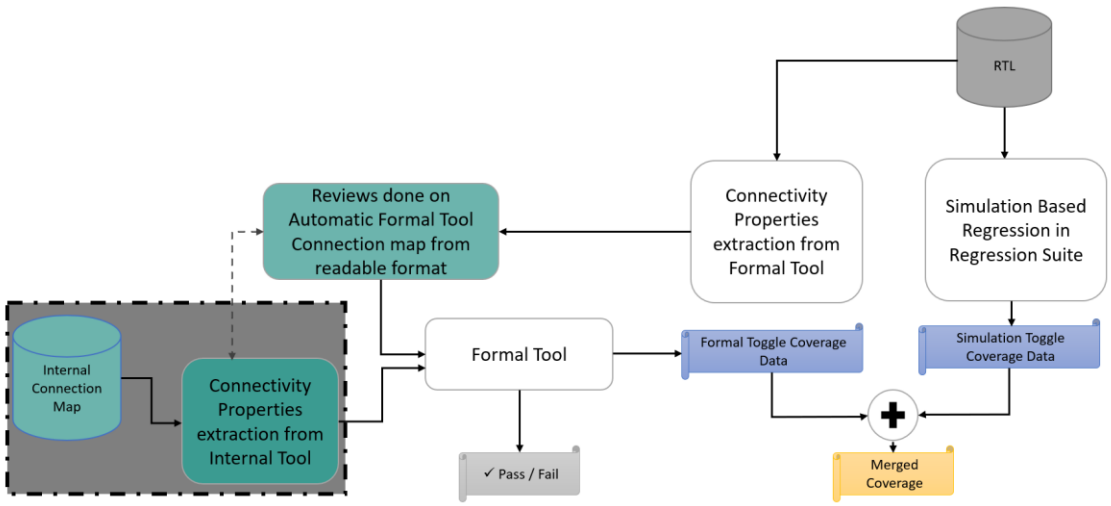

Fig. 2: Formal and Simulation based Verification Flow for Coverage sign-off

The proposed solution focuses on toggle coverage and it being a primary goal for verification sign-off at SB level, we propose an easier method to achieve 100% closure albeit the toggle coverage achieved using this method only shows that a signal transitioned between logic values in the context of connectivity checks. This method does not close toggle coverage gaps using any functional behaviors-oriented tests for the design. Holes in toggle coverage are targets of this approach which have not transitioned in all previous tests in the test suite targeting the design. Using this proposed approach, we are primarily checking connectivity and using this limited context to reduce efforts compared to more traditional coverage closure approaches. These include writing directed testcases focusing only on connectivity, and manually waiving toggle cover items which are not interesting for SB level and are functionally already covered at the IP level.

Using the proposed flow, where the toggle coverage generated from the formal tool and the simulation tool is of the same format as depicted in Fig. 3 and can be easily merged in verification management cockpit for review. As mentioned above, the manual waiving can be drastically reduced with formal checks for connections which are already covered at the IP level since these connections are covered again by the formal flow and the toggle coverage is shown as covered by formal toggle property checks. On top of that, the coverage hole from functional simulation can be covered with the assertion and toggle properties generated with formal tool which also makes writing the directed test case redundant, since the formal connection is derived from the design and should be correct by construction after the properties have been reviewed. To guarantee that certain structural toggle coverage which is identified as should have been covered with existing functional verification flow and still shows as not covered in simulation, is not missed, a semi-automated review process of comparing the generated

connectivity map from Formal tool and internally generated connectivity maps from golden source where connections between IPs to IP, bus, bridges and ports is defined has been put in place to make the flow robust.

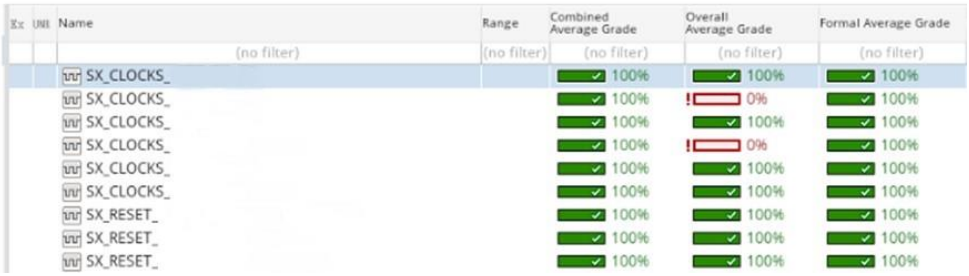

Fig. 3: Merged Simulation and Formal Toggle Coverage in Regression suite

Nevertheless, not all toggle coverage holes in functional simulation can be replaced by formal connectivity toggle checks since there are integration verification requirements which states and expects certain parts of RTL to toggle and this has been review both by the verification, design, and the concept teams, where necessary.

IV. RESULTS

The approach showcases an automated process for establishing formal connectivity checks at the SB level, eliminating the need for manual intervention, while providing coverage metrics necessary for verification sign-off. By process optimization, the overall turnaround time was reduced significantly, and the verification cycle was completed within a few days instead of months. This flow was deployed on an automotive product it was successfully taped-out with significant reduction in overall SB level verification time. The speedup was observed to be up to 8x in certain parts of the flow. The design had 13000 Top-level and Inter-IP connections that require verification and including initial iteration of adopting this flow had an overall 4x faster sign-off as summarized in Table-1.

Table I. Results

Comparison Criteria	***Simulation***	***Formal Connectivity***	***Speed up with proposed flow***
Sign-Off time	2 MM	2-3 MW	3-4x
Debugging time	1 MM	0.5 – 1 MW	4-8x
Run time	1 MM	2 MW	2x

[a.] *MM is Man-Months and MW is Man-Weeks of Efforts Spent*

REFERENCES

[1] Foster, Harry, “Trends in functional verification: a 2014 industry study”, Design Automation Conference 2015 (DAC’15).

[2] Hind Benbya, “Complexity and Information Systems Research in the Emerging Digital World”, MIS Quarterly, 44(1), pp. 1-17.

[3] Harry Foster. “Guidelines for creating a formal verification testplan”. In: 2006.