Tutorial Submission – Industrial

Title: Static Signoff Best Practices – Learnings and experiences from industry use cases

Speaker: Vikas Sachdeva, Director of Product Strategy and Business Development at Real Intent. Phone: +91 8861374174, Email: vikas@realintent.com

Introduction

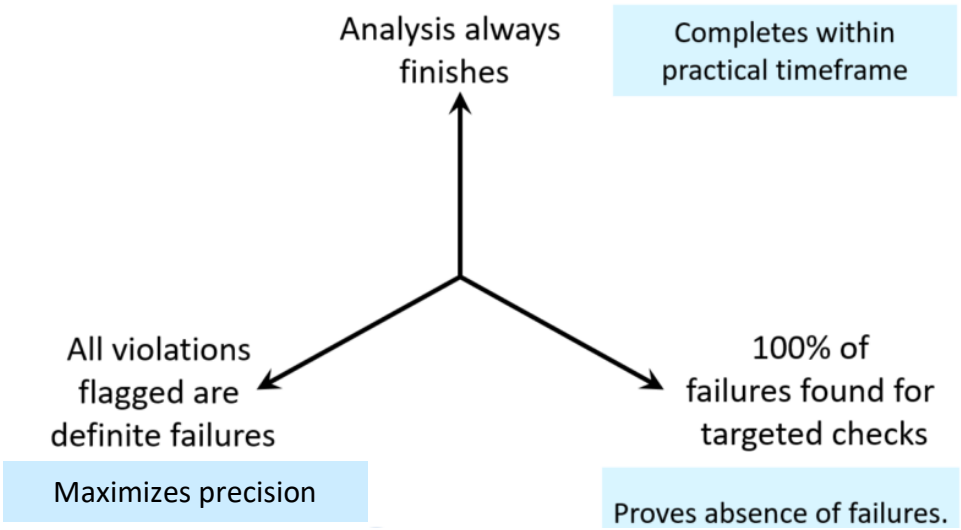
Verification flows seemingly have an infinite appetite to consume engineering resources. Per a recent case study by Wilson Research Group, design engineers typically spend 46% of their time doing verification. Verification engineers, of course, spend all their time doing verification. Although the most significant contributor to silicon re-spin is logic or functional bugs, clocking related bugs come a close second. With increased design sizes and ever-increasing complexity in clocking, power management and resets, formal and static methods have become crucial tools to close verification of complex designs.

Summary of the content of the tutorial

The tutorial covers the still-growing role of formal and static methods in various areas of front-end design: microarchitecture and functional correctness; clocks, resets, and metastability; and analog/mixed-signal verification. First, it contrasts Static verification with Dynamic verification. Next, it proposes when to use static methods based upon learnings from multiple industry use cases. It then details case studies and experiences and how to tame the problems that cause most of the design re-spins, namely logic and functional bugs, clocks and reset issues.

The tutorial teaches six fundamentals to maximize design and verification efficiency.

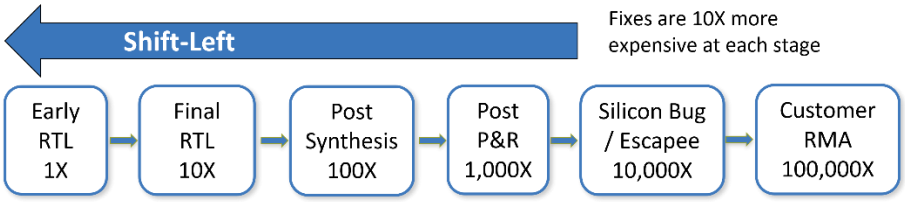
1. Use verification metrics to assess the efficiency of chosen approach - A simplified view of three key verification metrics to evaluate the merits of simulation, formal and static signoff are:
    a. The analysis always finishes — meaning that it completes in a reasonable timeframe.
    b. All the violations flagged by the analysis are definitely design failures, including ‘warnings’ of potential failures.
    c. 100% of the failures in the targeted checks are found by the analysis. Thus, the analysis proves the absence of any design errors.



An ideal verification methodology would rate a perfect score on each metric. But unfortunately, there is no perfect methodology available today: each approach has its strengths and weaknesses.

2. Deploy the bespoke mix of static, dynamic, & formal verification for a target domain. There are multiple target domains, like the clock and reset verification, where verification is essential for silicon success. Verification teams can efficiently apply various levels of dynamic and functional static signoff for these target domains.

3. Shift Left – start verification early because it pays to do it.



4. Continuous verification – is a very effective technique in minimizing the signoff burden. Any changes in RTL trigger a new cycle of static and dynamic checks in parallel.
5. Ensure your signoff methodology is complete, as one bug is enough to cause a respin.
6. Align static methodology with tools

The tutorial then dives deep into specific industry use cases and examples, demonstrating how the industry has begun to apply these six fundamentals in the real world.

- **Google's** case study on cloud-based signoff methodology
- **Nvidia's** case study on static signoff methodology and best practices, spanning RTL Linting, RDC, Single- & multi-mode CDC, RTL & gate-level CDC.
- **Samsung** case study on Samsung's RTL linting and clock domain crossing signoff methodology.

- Learnings from **Hailo's** static signoff methodology for edge AI processor
- **Palo Alto Networks** - advanced X-propagation methodology to identify X-initialization source errors and fix them to prevent the error from propagating.
- **Samsung** – Using the right mix of static and dynamic verification for CDC Sign-off
- **Fujitsu** – 30% reduction in logic simulation TAT using automatic formal techniques
- **SK Hynix** – Advanced reset design and verification methodology
- **Renesas** - Efficient functional sign off by automatic assertion generation for RTL building blocks
- **Real Intent** - the interrelated role of tools vs methodologies for static signoff.

The tutorial finally proposes a set of Static Signoff best practices based on industry learnings and experiences.

The tutorial is designed to benefit audience members and enable them to enhance the state of verification practice in their workplace.

Intended Audience:

RTL Designers and managers, Verification engineers and managers, SoC designers and verification engineers, Chip Architects, Clocks and Reset designers and architects