Delivering Functional Veri cation Engagements

A Proven, Systematic and Assured Approach

July 2015

Author

Subhranil Deb
Sr. Design Consultant,
Synopsys India Pvt. Ltd.

Introduction

With the advent of smarter and higher performing devices, there has been a tremendous increase in design complexity. Driven by new high-end hardware feature and intelligent software requirements, these devices are comprised of multi-core processors and a multitude of interface IP, memory and other analog circuitry, communicating via many different interface protocols. This poses a huge challenge in terms of scoping, resource planning and delivering a bug-free design. To address these technical challenges (as depicted in Figure 1) within budget and schedule, the emphasis on rst-pass silicon success is paramount – which calls for a thoroughly veri ed design. To transform an optimal speci cation to a fully veri ed system, you need a rich package of industry-proven tools and reliable methodologies combined with an experienced team of veri cation specialists.

This white paper provides insight into the challenges and need for a robust functional veri cation infrastructure, plus a means to achieve closure by leveraging Synopsys' suite of high-end tools and applications. Additionally, best practices followed by Synopsys Professional Services consultants to deliver advanced SoC/ASIC veri cation solutions to leading customers are explained.

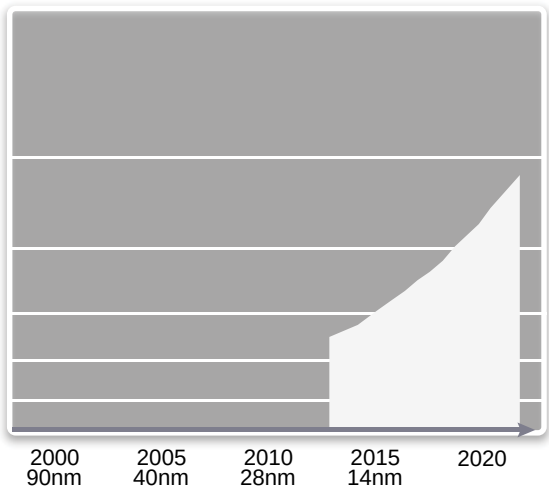

Figure 1: Veri cation challenge as process technology advances and design complexity increases. Courtesy: Synopsys

Increasing Need for Functional Veri cation

Consumer demand for new features and functionality, higher performance and lower power are key factors in driving the evolution of devices. As demand increases, so does competition, putting immense pressure on companies to be rst to market. First-silicon success has become imperative, but achieving it is extremely dif cult. Failure can be disastrous, resulting in the need for design re-spin. The impact of design re-spin is increased cost and risk of missing the market window versus competition.

According to IBS, at 28-nm the price of each mask set is between two and three million dollars and ve and eight million dollars for 22-nm/20-nm. Studies also suggest that sixty percent of ASIC/IC respins are attributed to “logical/functional failures” and “speci cation changes”. Finding and xing these issues earlier in the design process is less costly as shown in Figure 2.

So how do you ensure the quality and compliance of the ASIC to the speci cation? Though architecting and developing a design is fundamental, validating correctness by testing the design against the speci cation is critical. The earlier you identify a problem, the easier it is to x. The use of pre-veri ed design blocks and veri cation IP coupled with industry proven veri cation tools and techniques can be a game-changer. In the following sections, we outline some of the key functional veri cation challenges design teams face, explain the phases of veri cation and share Synopsys’ veri cation consultant recommendations and best practices, followed by two case studies.

Figure 2: Cost impact of bugs at different phases of execution

Functional Veri cation – Challenges

The key to delivering a successful ASIC is to ensure that the intent of the product (speci cation or requirement) and the nal deliverable (design) converge. This effort to verify design convergence is often signi cant, therefore it is imperative to develop a strategy that is both scalable and reusable. Important elements of this strategy that must be addressed are:

Domain Expertise

Veri cation is a perfect example where expertise is inversely proportional to the time required to validate a design, hence there is no substitute for experience. The work is manual and how a testbench is con gured varies with schedule, technical and system requirements. Understanding design priorities are essential. Most designs use multiple interfaces and interleaved bus interconnects. Experts are needed to understand and look at the design from the system point of view to tackle various protocol and performance nuances during the project. As we learn later, development and debug are two areas that need specialist attention to identify the issues, determine whether they are design or testbench related and then resolve them. This has been a challenge for engineering teams to nd the right balance of specialists and tools versus cost. Apart from protocol and veri cation techniques, there are other areas including low power and hardware/software simulation that need domain expertise, but these are not addressed herein.

Planning

Determining the extent of veri cation required and the time it takes to verify a design per schedule stipulated is

to improve reusability. Agents should be made active or passive depending on whether they are used to drive stimulus or only monitor them. Try to restrict randomization in a single con guration object during the build phase to help scenario re-creation based on seeds. When there are multiple agents in a system, use virtual sequencer to streamline data and make user testcases easier to code. Handle resets at a later time but keep provisions for resets in the components to avoid re-tuning of otherwise stable components. To avoid confusion, a veri cation engineer should follow the speci cation and not the design. Interpretation of speci cation grey areas need to be discussed with the team and plan of action mutually agreed to.

Often testbench and design development coincide. In such situations, verifying the testbench ow with an “Empty DUV” setup can expose testbench errors early. “Empty DUV” setup is like having the full-testbench wired or connected directly without the DUV. A driver connects to its peer (or receive) monitor and likewise for all interfaces. When tests are simulated, it can pipe-clean the components and also help develop the automated scoreboard. Figure 4 highlights an example Empty DUV setup for early testbench ow veri cation, thus enabling actual veri cation of the design to start immediately once it becomes available.

Figure 4: Example Empty DUV setup

One of the common scoreboard challenges is when it tries to perform cycle accurate predictions to match the output. The data is sometimes early, sometimes late, but in both cases seen as a testbench error. In such cases,7(s a)4is sc10.8(c(c10.8(c(c10.

Once regressions are mostly passing, it is time to check the coverage numbers. Planned coverage goals and cover groups are implemented during the development phase. Tracking coverage numbers early won't yield much if substantial number tests or features are not completed. Analyzing the coverage numbers and nding the holes in coverage is necessary to determine if some corner scenarios have been missed to ascertain if more tests are needed. Sometimes it is easy to create a directed test to hit a particular corner. Complete functional coverage is a necessity, while structural coverage exclusions are an optional consideration post design review. One-hundred percent functional coverage and passing regression tests should be the absolute minimum sign-off criterion.

Review

Periodic reviews are essential to track progress. It is important to involve the right set of people in reviews to be most constructive. From the initial estimations, to veri cation planning, to test extraction, to coverage closure, every deliverable and dependency should be reviewed. Reviews can potentially expose holes in veri cation strategy, missing features and helping to prioritize work. Decision makers and other key stakeholders gain useful insight into project status and issues that require escalation to x as early as possible during veri cation to keep the project on schedule.

Case Study 1

This design was based on an image sensor application comprised of two back-to-back modules that supported three incoming sensor protocols mapped into a periodic data stream which were controlled by con guration registers. The con guration space was relatively large with interdependent variables. The prime objective was to completely architect and verify the ASIC with one-hundred percent functional coverage.

The veri cation team ramped up on the different protocols and created the initial estimates and schedules for discussion and nalization after the team understood protocol complexity and initial design architecture. Following this, the team worked on the design speci cation and sensor standards with all stakeholders to create the feature list. The feature list was a living document on which the testbench was architected. The idea was to stimulate the DUV with different sensor information packed into transactions from the active agents (AGENT1 and AGENT2) and monitor the data at different interfaces based on the con guration done through AGENT3. The testbench architecture was scalable, enabling top-level reuse. The same testbench was reused for veri cation of both design (DUV1 and DUV2) blocks. Figure 6 shows the testbench architecture with data ow across the testbench. The

The major veri cation challenges were:

Evolving Speci cation

The design was based on the standard protocols which were suggested by the veri cation team as well, so there was overlap in the development phase. Because of this, testbench components were aligned to the actual standards and not to the design speci cation when it was released. This caused substantial rework and patches as the project progressed. Some important lessons learned were:

- Special care needs to be taken about feature ambiguity. All features should be reviewed with the system architect to ensure no inconsistent, unintended, overlapping or unclear features. This may not be a simple task. For example, you may have feature A plus feature B that somehow violates or invalidates feature Z, as described in a different chapter or speci cation.
- When a standard protocol is designed, the stimulus should follow the standard even if the design is a subset of the standard. However, the monitor should align with the design requirements.
- Sometimes it helps to overhaul logic early in the projecepo3.2(w)2.3(ep14E s)-12.3(h)-9.3(o)6.1(o)-10.8(e s)-7.1(ti)-8(m)-9ee(d)-11.9(e)
  e quem ier.1(ti)-8(m)-.6(i)-10.5((i)-10.4 a)-12.5(l)-7.6la,y(a)-11.4(sthg)-9.5(i)-7.9(s(c)-14.8(a)-p14E s h)-10.3(a)4.7(v)3.3(e(i)-8(m)-10.8(

Figure 7: Distribution of effort during the veri cation phase

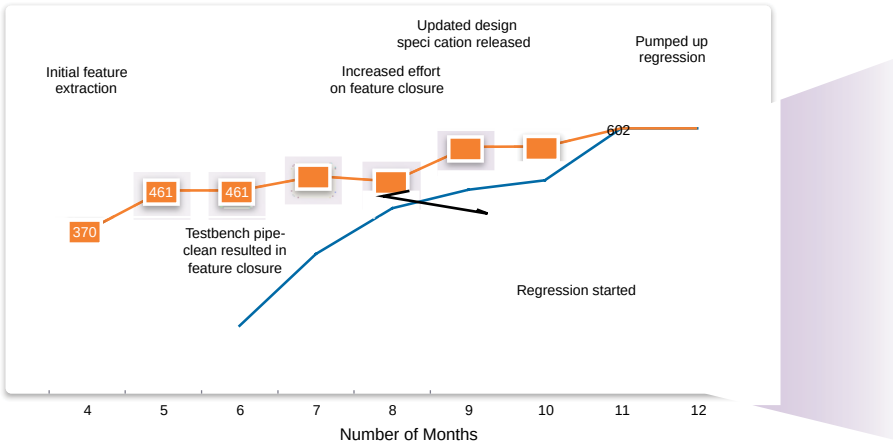

Figure 8: Rate of feature closure compared to total feature extraction and fail percentages versus pass seeds

The lower right insert in Figure 8 depicts the fails percentages of regressions run. During the initial eight to nine weeks of regression, the fail percentage varied between 0.06% and 4.06% with approximately nine thousand seeds. During the nal four weeks, when the number of seeds were increased signi cantly, fail percentages decreased dramatically going down from 0.167% to 0.002% for one-hundred-eighty-thousand seeds.

Deliveries included:

- A fully UVM compliant testbench with approximately sixty random test cases
- One-hundred percent functional and assertion coverage and ninety-one percent line coverage with no exclusions
- Six-hundred and two features closed with 182,225 seeds passing
- Exhaustive veri cation plan including the testbench, test cases and explanation how to execute

Case Study 2

In this case the design was a custom interconnect with industry standard protocols P1, P2, P3 (names withdrawn for con dentiality) and Ethernet. It was a router with 6x9 bidirectional interface and con gurable protocol support. The input protocol was mapped and routed to output protocol based on con guration, which led to a very large con guration space with multiple options including number of lanes, speeds, and data types, while maintaining bandwidth and throughput. The prime objective was to completely verify the ASIC with one-hundred percent coverage of all features, datapath and con guration combinations. A set of system scenario-like use cases was

Deliverables included:

- Complete veri cation solution with more than ve-hundred tests cases in four different testbenches
- Regression suite with more than fteen-hundred differenffery demarks is igr.41p.(f9(u)x1o)0.6(m)-4.5(50003>9(h)7/ActualText<FEFF200A>>>R)0.6(e)-6.2w 0 -1.33ivegr.41pAusen setig.4(r)-1(ef(

Synopsys, Inc. • 690 East Middle eld Road • Mountain View, CA 94043 • www.sy nopsys.com

07/15.CE.CS6130.