Post-silicon Validation Challenges: How EDA and Academia Can Help

Jagannath Keshava
Intel Corp. M.S. SC12-214
2200 Mission College Blvd.
Santa Clara, CA 95052
(408) 653-8547
jagannath.keshava@intel.com

Nagib Hakim
Intel Corp. M.S. SC12-214
2200 Mission College Blvd.
Santa Clara, CA 95052
(408) 765-7902
nagib.hakim@intel.com

Chinna Prudvi
Intel Corp. M.S. JF5-254
2111 N.E. 25th Avenue
Hillsboro, OR 97124
(503) 712-3097
chinna.prudvi@intel.com

ABSTRACT

The challenges of post-silicon validation are continuously increasing, driven by higher levels of integration, increased circuit complexity, and platform performance requirements. The pressure of maintaining aggressive launch schedules and containing an increased cost of validation and debug, require a holistic approach to the entire design and validation process. Post-silicon validation is very diverse, and the work starts well before first silicon is available—for example, emulation, design-for-validation (DFV) features, specialized content development, etc. This will require enhancing pre-tape out validation to have healthier first silicon, developing more standard interfaces to our validation hooks, developing more predictive tools for circuit and platform simulation and post-silicon debug, adding more formal coverage methods, and improving survivability to mitigate in-the-field issues. We view the Electronic Design Automation (EDA) industry as a key enabler to help us bridge the gaps between pre-silicon and post-silicon validation, and extend the considerable intellectual wealth in pre-silicon tools to the post-silicon validation area.

Categories and Subject Descriptors

B.7.3 [**Integrated Circuits**]: Reliability and Testing – *Built-in tests, Error-checking, Redundant design, Test generation, Testability.*

General Terms

Measurement, Performance, Design, Reliability, Standardization, Verification.

Keywords

Design, Verification, Validation, Emulation, Test

DAC'10, June 13–18, 2010, Anaheim, CA, USA.

1. INTRODUCTION

Post-Silicon validation of large microprocessor designs entails testing of components in a system setting. The focus is on the components as well as the interactions between them and with the platform. While a significant amount of preparation readiness and influencing work is done prior to first silicon, the arrival of the first silicon samples typically heralds the post-silicon phase of validation. It involves multiple different aspects, such as logic validation and debug, electrical validation and debug, and debugging software and customer issues. The post-silicon validation domain has certain unique advantages and limitations over traditional pre-silicon simulation based validation. Post silicon validation is performed at target platform speeds—in the gigahertz range—and contains the real system components. In contrast, pre-silicon validation includes limited platform level interactions and runs in the hertz or tens of hertz range. However, visibility and ability to control the system are much reduced in post-silicon validation and failure analysis requires significantly more effort.

The disciplines of post-silicon validation include System Validation (SV), Compatibility Validation (CV), and Electrical Validation (EV) [1, 2]. System Validation is typically performed on custom validation boards using specially written tests. These boards are usually larger than commercial boards. They provide room for additional connections to probing, measurement, and debug tools, and allow interchangeability of components. The tests run the gamut of OS-level programs and random instruction tests, down to hardware-coded patterns and built-in self tests. The tool sets involved in SV are quite different from that in the pre-silicon world and rely heavily on in-target probes, logic analyzers, hardware accelerators, and tools using system management interrupts to bound the problem and reproduce it in a simulation environment [3]. Tests include focused tests to stress one aspect of the die, Random Instruction tests, or even OS-level applications. In CV, silicon is tested using commercially available operating systems and applications with a focus on ensuring backward compatibility with existing software, and running additional tests in an environment more resembling the end user's own platform and applications. EV focuses on validating the part from an electric perspective, analyzing its inherent performance and margin, and on its electrical interaction with other components on the board. Tests here range from commercial applications, to specific patterns that stress the various components and platform interconnections. Speedpaths are usually tackled by a separate step in the overall validation, using on both platform and high-speed testers [4]. While testers provide very good controllability of the voltage and temperature

environment, they may not duplicate the environment of a real system and may produce optimistic results. Platform-level tests enable to reproduce this environment better, and even provide patterns for future high-volume tests.

The whole domain of post-silicon validation is undergoing fundamental transformations, from radically changing environmental conditions [5, 6, and 7]. There has been recently a step up in new integration of functions into the processor, and interfaces and the primary visibility point of the past—the Front Side Bus (FSB) no longer comes out of the processor. The interfaces coming out of the processor are now a lot faster, and there are many more of them. Additionally, the increase in the number of cores and threads and the inclusion of advanced power management and virtualization drive additional complexity, boundary conditions, and bus requirements. The other key related issue has been that resources needed to validate a design have increased. On the capital expense side, logic analyzers have become a lot more expensive [6, 7, 8], while on the headcount side, the proportion of post-silicon validation resources as a percentage of design resources has gone up significantly over the last decade.

The goal of this paper is to highlight some of the key challenges in the post-silicon validation space of large, highly integrated designs, and discuss some of the promising solution vectors on the horizon.

2. CHALLENGES OF POST-SILICON VALIDATION

The combined effects of increased product complexity, performance requirements and time-to-market commitments has put tremendous pressure on validation, which is usually the last step prior to volume manufacturing. This forces the validation team to constantly examine its methods and processes, to make validation faster, cheaper, and better.

2.1 Validation Time-to-Qualification

Increasing validation resources and the number of validation engineers can only go so far in speeding up the validation process. Oftentimes, other intrinsic factors determine the time required to qualify a product: complexity of the design, initial silicon quality, and validation and debug tools.

Qualification of the product is a judgment of product readiness at the end of validation, as illustrated in Figure 1 at the Product Release Qualification (PRQ) milestone. While several coverage metrics have been developed [9, 10], their application to qualify a product relies on heuristic formulae involving the number of features, events, and states exercised, with an emphasis on newly introduced features and risky designs. Other metrics rely on the rate of bugs being found, and usually indicate a product readiness when a period of time passes without finding new issues [2]. These traditional metrics are hard to obtain in post-silicon, as platform tests are often redundant and it is hard to observe the internal states of the machine. Furthermore, they require a historical perspective, subjective assessment, and a lot of expertise to convert them into a useful risk assessment which provides confidence in the goodness of a design. As a consequence, validation efforts tend to be overly conservative to minimize the risk of a bug escaping to the field.

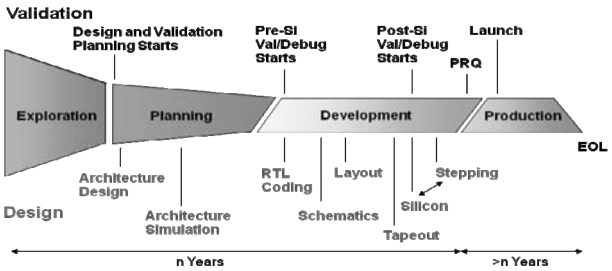

Figure 1. Design and Validation Product Lifecycle.

Validation time also depends on the quality of the initial design [2]. With the increase of design complexity, the simultaneous development of the product, the manufacturing technology, and the system boards, it is often the case that several product steppings are needed to eliminate all blocking bugs (bugs that prevent the continued validation and discovery of further issues) and achieve the level of performance required for the product. These steppings are quite costly to the validation as they imply dead time while bugs are fixed in the design and new masks are being manufactured. These can often be worked around, allowing some validation to proceed, but still limit the pace at which validation effectively proceeds.

The nature and impact of bugs is also changing with our designs [4, 5]. The integration of more chipset components on the microprocessor results in a larger number of clock domains, and clock domain crossings. Increasingly complex power management schemes, with multiple voltage domains and clock and power gating solutions, require more complex power grid integrity validation. The increased complexity of analog and mixed-signal circuits necessary to meet challenging performance and interconnection targets also requires special flows to validate these circuits in the context of their digital control and platform environment. The scope of silicon mixed-signal validation is greatly augmented by silicon process variation as well as voltage and temperature fluctuations. The complexity of verifying the electrical performance and its impact at the platform level in pre-silicon is a strong limiter, and renders the bugs that are found harder to debug due to the lack of system-level accurate electric simulation.

Historically, logic bugs have been relatively easy to reproduce by recording traces on the part and replaying them in a simulator [3]. However, with increased non-determinism from multi-core, multi-clock domain designs, a post-silicon trace ported to a simulator may not reproduce the bugs. Furthermore, simulators do not help debug transient electric effects, which are notoriously elusive and non-deterministic. With all these effects becoming more pronounced with every new generation of products, it is imperative that our validation tools and methodologies stays ahead to manage the added requirements. These tools and methodologies include validation infrastructure, test content, and debug tools and hooks.

2.2 Validation Costs

Historically, logic analyzers have represented the single largest capital equipment expenditure in post-silicon validation as they are necessary to observe traffic on the bus and I/O pins [8]. However, with the proliferation of multi-core designs, as well as SOC designs, and the integration of additional components previously implemented in the chipset, many of the internal

interfaces have become increasingly hard to observe. Traditional probes have been modified to accommodate the higher bandwidth, the increase in bus traffic, and the hidden interfaces, resulting in much higher cost. In addition, JTAG ports, used to control many of the DFV hooks on the die, are too slow for real-time control and synchronization of events. This has put tremendous pressure to implement many of these capabilities directly on the die, increasing the overall complexity and cost of the design. One example is the Intel® Interconnect BIST (Intel® IBIST) infrastructure [11] used to configure and validate high-speed interconnects. It provides a set of configurable registers to configure and observe the various components on the die. Another example leverages the System Management Interrupt (SMI) to take a snapshot of the state of the machine at periodic times for debug applications [18]. These hooks and others not described here, are invaluable in providing visibility and controllability during the validation process, and can serve a dual purpose by also serving as survivability features. A big challenge in this area is designing these hooks with low area and power overhead but with maximum benefits and reuse.

With incentive to do more validation in pre-silicon and the proliferation of emulation hardware, there is more widespread reliance of designs on emulation to verify the design before tape-out and also to assist in post-silicon debug. Even with decreasing emulation platform costs, the cost of using these systems in a widespread fashion will quickly consume a significant portion of the overall validation budget. In addition to that, circuit and speedpath debug requires a wide variety of different sophisticated tools, such as infrared emission microscope (IREM), laser assisted debug, focused ion beam (FIB) editing, and others, which may require special handling of the die and special custom-made validation boards. Additionally, these tools often lack the automation of more classical methods and require special board infrastructure.

Post-silicon validation tools do provide invaluable help to validators, but are not keeping up with the increase in complexity and reduced design cycle times, and in general lag behind pre-silicon tools in terms of productivity, ease of use, and overall maturity. Thus, despite the significant investments, post-silicon debug remains to some extent an art, where a validator's experience plays a huge role.

3. SOLUTION VECTORS

The solutions to the above validation problems span the product lifecycle shown in Figure 1. Naturally, they are very diverse and cover a wide range of technologies, to address the differing requirements at each stage of the design. In this section, we outline some of the potential breakthroughs that would enable faster, better, and less expensive post-silicon validation.

3.1 Pre-silicon Engagement

One of the big factors in determining the post-silicon validation time is the incoming silicon health. It takes a long time to debug issues in post-silicon due to limited visibility into the design. Bug fixing in silicon is also a lengthy process since it involves design convergence, tape out, and fab throughput. It is very advantageous to catch as many issues as possible in the architecture, design, and pre-silicon validation phases. During the architecture phase, development and refinement of specs using high-level abstraction models will result in introducing fewer bugs during the design development process (RTL, schematics, layout, etc.). There are several opportunities to improve quality during the design phase. Some of these include extensive use of formal methods during the pre-silicon phase, emulation for finding bugs that require long cycle count, and analog mixed signal validation to find bugs in the logic/circuit interaction. Bugs and other issues found in the architecture and design phases take less time to root cause and result in significant savings downstream.

Design complexity, as illustrated in Figure 2, and the number of new architectural features are also key contributors to the number of bugs and rising cost of validation (as quantified here by the number of Person-Years (PY) to validate a complex product). While pre-silicon validation pays special attention to these new features, they still pose a risk to post-silicon validation. One of the key challenges of product planning and design is to manage this complexity by being able to accurately predict their risks to the validation time and overall product development cycle.

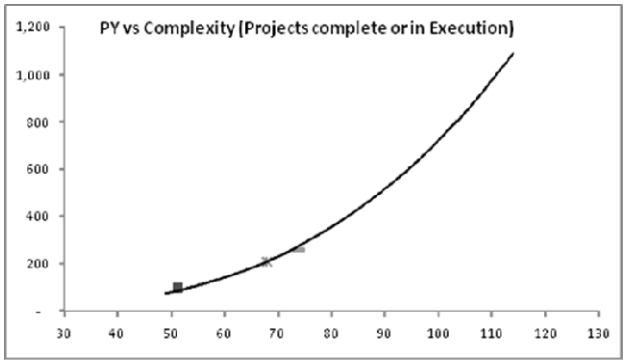

Figure 2: Increasing complexity drives increased time-to-market and cost (Person-Year).

Another significant component to the quality of the design lies in the electrical performance of the part. Mixed-signal validation has become the de-facto standard for verifying the analog portion of the design, but needs fundamental improvements to deal with the increased complexity of these circuits and the increased variability of analog circuits. Key components of making mixed-signal validation more effective include fast (and accurate) SPICE and mixed-signal simulation, automatic model generation and verification (including process variations and voltage/temperature effects), the inclusion of a larger portion of the analog circuits in the simulation flow, and possible inclusion of more of the platform I/O links. On the platform electrical validation side, it is critical to include more of the on-die effects, such as power-grid noise, process variations, and switching patterns.

3.2 Post-silicon Opportunities

There are several opportunities to reduce cost and improve debug throughput that can have a big impact in the time-to-market of products. On the cost side, there are opportunities to develop content in the pre-silicon phase that can be leveraged in post-silicon and tools that analyze post-silicon coverage and help target content in areas which are more complex or newly developed and are thus more risky, or areas which are not easily predicted in pre-silicon, such as high-speed I/O circuits in an off-chip interconnect. A fundamental question in post-silicon validation is when does a validator know when they are done validating? This may require additional observability hooks to capture internal states and count

specific events, which can then give a better estimate of the coverage of various features.

Most of the traditional post-silicon debug techniques still rely on the use of increasingly expensive logic analyzers for observing external interfaces of the Device Under Test (DUT). The alternative to external logic analyzers is to provide observability inside the DUT and a way to expose this information to the system debuggers [12, 13, 14]. Post-silicon bug disposition goes through the following phases:

1) Triage (Isolating the failure to a specific component or area)

2) Root causing of the failure

3) Validation of the fix or workaround.

As mentioned earlier logic analyzers are predominantly used in post-silicon debug, including triage and root-cause analysis phases. These can be effectively replaced with well thought out on-die debug hooks. Efficient triage will require debug hooks that help isolate problems to specific components on the platform/DUT (e.g. software, hardware, firmware, etc). Efficient debug will also require a way to observe the relevant state internally and good triggering hooks to focus on the relevant sections of the traces/state for debug and root cause analysis.

Together with these hardware hooks, analysis tools are required to help guide the validation process. Some of these approaches include trace buffer analysis [15, 16], trace reduction [17] and formal methods [18]. These techniques take on an increased importance in the context of a fast turnaround time requirement or to support automatic bug localization.

Formal methods are also a good way to ensure that fixes are robust and don't lead to additional rework in the future.

3.3 Survivability (Post-silicon Debug and In-field Repair)

Despite the best efforts in the pre-silicon phase, and due to the increasing complexity of the devices, some issues will have to be found and debugged in the post-silicon phase. A smaller number of these issues may also escape into the field. To assist with debug of the issues in the post-silicon phase and to survive issues in the field, designs need to implement comprehensive survivability features. These features can take the form of microcode patches or be implemented as programmable hardware (configuration registers, fuses, etc.) [11]. During the post-silicon validation phase, survivability features can help in isolating issues to smaller design segments and can also assist in working around any blocking bugs. This can also help post-silicon validation in onion peeling of the bugs (oftentimes bugs hide behind other bugs), without recourse to expensive additional silicon steppings. If the workaround meets the product goals, the product can be shipped with the workaround. Once the product is shipped, survivability features can help in working around bugs found in the field. In this usage model, the additional requirement is that the fix should be deployable in the field.

In all of the above three areas, the EDA industry and academia can play a big role in exploring and developing tools and capabilities that solve issues facing post-silicon validation.

3.4 EDA and Research Opportunities

Based on the problems highlighted above, there is a dire need for innovations in tools, methods, and design features to address the validation challenges. Some solutions were outlined in the previous paragraphs, and additional EDA topics are singled out here, but much of the needed research topics still remain to be identified. Extending current pre-silicon tools for post-silicon applications is a first key step. One example of current EDA efforts includes system-level test generation, fault grading, and coverage. Most of the development of this area has happened in the pre-silicon domain or in high-volume manufacturing test, but its extension to platform-level validation has been quite limited. Similarly, recent developments in statistical inference may provide a breakthrough for data mining observed signals and guide the debug and root-cause analysis. While post-silicon tests offer a huge speedup over pre-silicon tools and avoid simulation and modeling errors, these tools may still play a role in helping guide and speed-up post-silicon validation and debug.

These topics may provide a transition of current EDA capabilities to post-silicon, and in this field that is characterized by very practical and physical constraints, injecting more formalism and thorough analysis would enable a more analytical and systematic approach to the art of bug detection, confirmation, and debug. This may also provide more continuity between pre-silicon and post-silicon validation efforts, and more solid footings for making decisions that relate to validation strategy and the product's readiness to release to market.

While a path from pre-silicon tool extension to post-silicon validation may provide immediate benefits, it would also create the momentum and critical mass needed for more radical innovation, and would lead to establishing post-silicon validation as its own discipline in design automation. This transition is essential to keep the EDA industry focused on the most challenging problems facing the electronic design world. Post-silicon validation is emerging as one of the biggest issues now facing us. The EDA industry's involvement and related academic research's innovation are key to providing the breakthrough and scalability needed in future product generations.

4. CONCLUSION

Post-silicon validation is fast becoming a potential bottleneck in productivity and cost. The EDA industry and academic researchers can play a big role in exploring solutions and developing tools and capabilities that solve issues facing post-silicon validation. Many of the pre-silicon methods, formalisms and tools may be extended to solve post-silicon problems, and bridge the gap between pre- and post-silicon. Others need to be developed from the ground up to target issues specifically seen in post-silicon and break the fundamental limitations in post-silicon controllability and observability. Automation and analysis tools should also help speed up the validation and debugging processes, quantify coverage and risk, and achieve faster time to market.

5. ACKNOWLEDGMENTS

The authors would like to thank Stuart Goldstein for his editorial contributions to this paper, and Rand Gray, Keshavan Tiruvallur,

Priyadarsan Patra, and Narendar Sahgal for their valuable input and technical guidance.

6. REFERENCES

[1] Intel Corp. 2003. Intel Platform and Component Validation, http://download.intel.com/design/chipsets/labtour/PVPT_WhitePaper.pdf

[2] Bentley B. and Gray R. 2001. Validating the Pentium®4 Processor. *Proceedings of the 38th annual Design Automation Conference*, Las Vegas, Nevada, United States: ACM, 2001, pp. 244-248.

[3] Silas, I., Frumkin, I., Hazan, E., Mor, E., and Zobin, G., System-Level Validation of the Intel® Pentium® M Processor , *Intel Technology Journal, Vol. 7, Issue 2, May 2003*
URL: http://developer.intel.com/technology/itj/index.htm

[4] Gray, R. 2008 Post-Silicon Validation Experience: History, Trends, and Challenges. *GSRC Workshop on Post-Si Validation*, Anaheim, June 9, 2008.

[5] Patra, P 2007. On the Cusp of a Validation Wall. *Design & Test of Computers, IEEE*, 24(2), 193-196.

[6] Keshava, J. 2009 Post-Silicon Validation Challenges. *International Test Conference 2009*, ITC'09, Nov. 1, 2009

[7] Tiruvallur K. 2009. Beyond Design…Challenges of IA Platform Productization. *International Conference on Computer-Aided Design 2009*, Nov. 2009.

[8] Yerramilli, S, 2006. On the Need for Convergence Between Design Validation and Test. *In International Test Conference, 2006*, ITC'06, 2006, p 14.

[9] S. Tasiran and K. Keutzer. Coverage metrics for functional validation of hardware designs. *IEEE Design & Test of Computers*, 18(7):36–45, Jul/Aug 2001.

[10] Bojan, T., Aguilar Arreola, M., Shlomo, E., and Shachar, T. 2007. Functional coverage measurements and results in post-Silicon validation of Core™ 2 Duo family. In *Proceedings of the 2007 IEEE international High Level Design Validation and Test Workshop* (November 07 - 09, 2007). HLDVT. IEEE Computer Society, Washington, DC, 145-150.

[11] Nejedlo, J.; Khanna, R.; "Intel® IBIST, the full vision realized," *Test Conference, 2009. ITC 2009. International*, vol., no., pp.1-11, 1-6 Nov. 2009
doi: 10.1109/TEST.2009.5355667

[12] Park, S. and Mitra, S. 2008. IFRA: instruction footprint recording and analysis for post-silicon bug localization in processors," *Proceedings of the 45th annual conference on Design automation*, Anaheim, California: ACM, 2008, pp. 373-378.

[13] Abramovici, M., "A silicon validation and debug solution with great benefits and low costs," International *Test Conference, 2007. ITC 2007. IEEE International*, 2007, p. 1.

[14] E. Anis Daoud and N. Nicolici, "Real-Time Lossless Compression for Silicon Debug," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 28, no. 9, pp. 1387–1400, Sep 2009.

[15] Park, S. et al. 2009. Post-Silicon Bug Localization in Processors Using Instruction Footprint Recording and Analysis (IFRA). *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 28(10), 1545-1558

[16] H. F. Ko and N. Nicolici, "Algorithms for State Restoration and Trace Signals Selection for Data Acquisition in Silicon Debug," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 28, no. 2, pp. 285–297, Feb 2009.

[17] Chang K., Bertacco, V. Markov, I, 2005. "Simulation-based bug trace minimization with BMC-based refinement," *International Conference on Computer-Aided Design*, pp. 1045-1051, Nov 6-10, 2005

[18] Ho, R. et al. 2009. Post-silicon debug using formal verification waypoints. *Design and Validation Conference, 2009, DV Con 2009*