Perspectives

On the cusp of a validation wall

Priyadarsan Patra
Intel

TRADITIONALLY, UNIVERSITIES TEACH how to make or build things but not so much how to "break" things or find, patch, or prevent breaks. However, much of industry validation hinges on the latter skills. Validation is something that does not get noticed when done well, but everyone notices when something goes wrong—such as the infamous Pentium floating-point division bug. Major semiconductor companies experience postsilicon validation turning into a very expensive, time-consuming proposition, yet very few college graduates are formally trained in the area. Validation is the activity of ensuring a product satisfies its reference specifications, runs with relevant software and hardware, and meets user expectations. Here, I discuss some of the key challenges to successful validation and show why a radical transformation is necessary if validation is to be effective in the near future.

Logic and low-DPM issues

The product development cycle often tends to be linear. The phases are roughly as follows:

- planning and architecture;
- RTL and schematic creation, combined with architectural and functional validation, leading up to tape out;
- circuit marginality, system, and compatibility validations coupled with silicon debugging;
- product release qualification; and
- high-volume manufacture and test.

There is usually insufficient closed-loop feedback, owing to a variety of reasons, including lack of a nimble automated system for tracking, data mining, and correlation analysis, and the subsequent driving of product improvement. However, as Figure 1 shows, processor bugs are growing with every new generation. Moreover, as Figure 2 shows, such bugs are increasingly becoming more complex and diverse. Bug prevention and detection must be driven upstream; it is common knowledge that testing shows only the presence of bugs, not their absence.

In addition to logic issues, low-DPM (defects per million) electrical and I/O (marginality) issues are fast-becoming a leading cause of bugs. Problems of observability, repeatability, and environment dependence complicate the timely arrest of bugs, thereby leading to time-to-market slips, costly screening, and extra silicon revisions. Why is low DPM problematic? The reason is that low DPM means that, say, one in 1,000 chips has an error in a multimillion product volume, but more detailed validation is practical only on a few hundred chips per million. A functional error would have made every chip defective and thus easier to identify, but difficult electrical and timing corners cause the "low DPM" problem.

Reinventing validation

The validation and test stage must reinvent itself amid a changing ecosystem marked by an increasing rate of architecture features, including usage models; instruction set complexity; safe interoperability of platform ingredients such as security, manageability, and virtualization; higher and diversified integration typified by SoC and terascale computing; IP reuse; electrical effects such as extreme process variation, power management, and the complexity of special circuits; soft errors and aging-related faults; and the rising costs of test and debugging equipment. Thus, in the area of product architecture and planning, effective validation requires innovations and tools to support resilience, testability, observability, and sur-

0740-7475/07/$25.00 © 2007 IEEE Copublished by the IEEE CS and the IEEE CASS

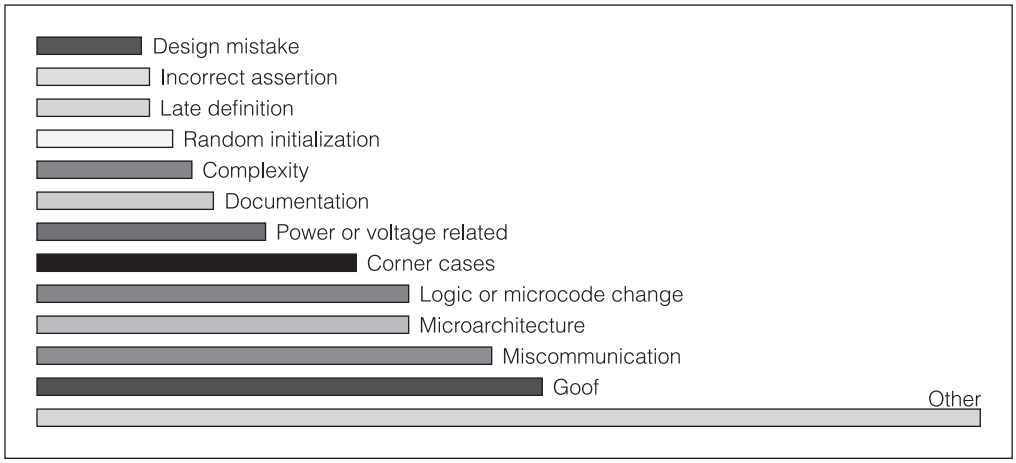

Figure 1. Sources of bugs in a recent processor's design life.

vivability features. In the area of design, effective validation requires modular validation; better and faster analog and mixed-signal simulation; generators of composable tests; coverage measurement; and assisted insertion of test and debugging features.

Virtual platforms and modular validation are important for addressing the scaling and heterogeneity problem in current systems. Mushrooming combinations of product stock-keeping units (SKUs) and runtime configurations, coupled with a matrix of power, performance, and reliability states, make validation increasingly unscalable without breakthroughs in modular validation. Marketing requires performance equivalence in order to sell different configurations as the same product. (In a multicore product, for example, different configurations result from choices made regarding die chops, fusings, component cores' locations, I/O links, and so on.) However, cycle-for-cycle identical behavior is needed for validation, test, and debugging. There are also various sources of asynchrony and nondeterminism, such as clock domain crossings, intrachip bit alignment and error correction, and the powering up of cores in *X*-ridden states, which further complicate SoC product validation.

Virtual platforms can help in three main ways. First, they can push postsilicon issues to presilicon, facilitating early bug finding. Second, they can foster content sharing between pre- and postsilicon test and validation. Third, they can enable early software development on the prototyped system. Of course, it is fairly well-known in industry that formal and dynamic methods are complementary: Formal verification is

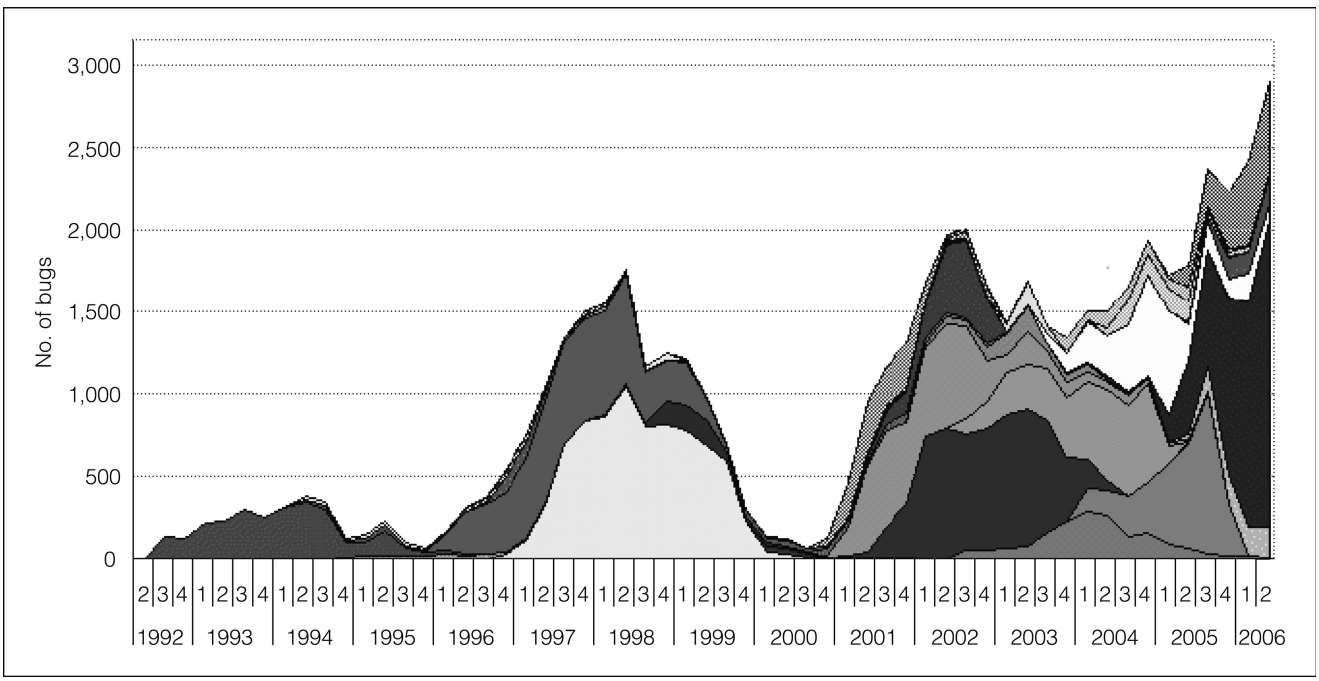

Figure 2. Quarterly bug count dynamics for several projects (overlaid in different shades).

suitable for verifying correctness of infinite traces but is limited in target block size. Dynamic testing is suitable for covering much of the target design but is limited to simpler and smaller traces. However, there are hardly any effective tools that integrate these two synergistic approaches seamlessly and robustly.

The work of presilicon validators does not end at the arrival of the first silicon. Rather, many of them enter the mixed world of postsilicon and RTL revision activities after the first *stepping* (silicon revision). Postsilicon validation involves a cornucopia of tools, ranging from array dumps, microbreakpoints, in-target probes, and logic analyzers, to kernel debuggers, hardware model accelerators, periodic system-management interrupts, and pattern and speed-path debuggers. There is very little automatic carry-over of validation content from the presilicon world to the postsilicon world. Rather, postsilicon validation is very effort intensive (Figure 3) and very dependent on team coordination and cross-team brainstorming. At the 2006 Electronic Design Processes Workshop (EDP 06), Intel reported an increase in head-count ratio between postsilicon validation and design from 1:5 in 2002 to 1:3 in 2005, with the risk of an upward trend. At the same time, capital costs for validation and test equipment were ballooning with every new generation.

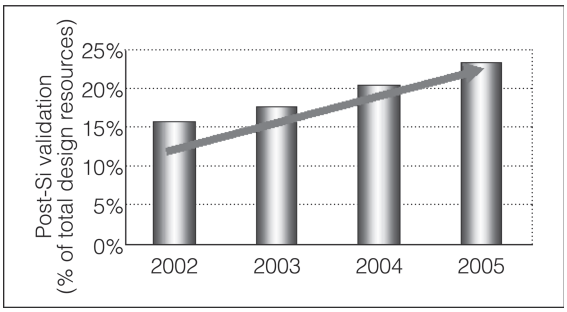

Figure 3. Postsilicon validation as a percentage of total design resources, in recent years. (This figure was derived from data presented by Intel's John Barton at the Gigascale Research Center March 2006 Workshop in a talk entitled "Overview and Challenges of Post Silicon Validation.")

Mitigating the risk of SoC silicon respins

The number of SoC silicon respins (reportedly 71% from logic bugs) in industry is increasing. An already

Interdependent landscape of validation technologies

Many validation disciplines are intertwined and codependent. The following is a partial list of some of these key disciplines:

- *Protocol validation*. This discipline includes formal verification, simulation, and hybrids.
- *Architecture validation*. This includes instructions and instruction combinations, operand data spaces, and all programmer-visible features.
- *Microcode validation*. This discipline involves micro-operations and path interaction of microcode with microarchitecture hardware (formal methods and simulation).
- *Unit (or cluster)*. This discipline focuses on internal pipelines and interfaces, as well as boundary conditions.
- *Full chip*. Interfaces between clusters and global conditions (such as reset or init) interacting with the microarchitecture need to be validated.
- *Power validation*. This concerns the different voltage islands and different power-performance states of the CPU.
- *Debugging validation*. This involves debugging or testing hooks for DFT features.
- *Multiprocessor validation*. This discipline requires a memory consistency model.
- *Analog, system, and compatibility validation*. These areas concern the processor working correctly with different boards, software drivers, and analog components.
- *Cosimulation*. Chip set and CPU RTL models run in one simulation environment.

Electrical validation must precede logic validation because many electrical issues can mask logic problems, yet electrical validation requires correct logic. Therefore, validation involves controlled iterations. Once logic and electrical validation are almost complete, software testing can begin.

expensive postsilicon bug is many times more costly at postdeployment. (One unofficial estimate puts the recall cost of the Pentium 4 Processor at $60 billion!) The following scenarios illustrate a few root causes of this general problem and how they were mitigated:

- Temperature and voltage-sensitive signal race due to interdomain clock skew between core and I/O phase-locked loops led to flip-flop metastability and system hang-up. Mitigation: Apply dynamic clock skew and jitter simulation.
- Write-setup-time failure occurred on various word lines because of victim-attacker noise coupling. Mitigation: Integrate noise into the timing simulation.
- Too many clocks turning on simultaneously because of excessive clock gating caused voltage droops, which limited the processor speed to 90% of the target. Mitigation: Apply mixed-signal simulation and dynamic clock skewing.
- Chipset received short interrupt assertions on Peripheral Component Interconnect Express (PCIe), leading to system deadlock. Mitigation: Employ better usage model capture.
- Logic bug in speculative page fault became manifest in corner cases. Mitigation: Perform model-derived pattern generation.

Research in effective defect-based testing is necessary because burn-in is fast-becoming impractical. High-level design language and verification should merge to facilitate bug prevention. Efficient alternatives to large, generic farms of computers that run simulators must emerge, as hardware and electricity costs skyrocket. Moreover, runtime errors are growing due to circuit and process marginalities or aggressive deployment, device aging, multibit logic or memory soft errors, and ambient-induced variations—and vulnerability to side-channel attacks is growing as well. In addition, worst-case design or validation that could ensure complete freedom from errors is prohibitive or impossible. Hence, better-than-worst-case design and validation implies a certain resilience in the design. Enabling such resilience calls for effective tools and methodologies in runtime self-validation, software-assisted self-test, self-reconfiguration, and recovery. It also necessitates migration of traditional design automation tools upstream, operating partly in the architecture and validation space. ■

Acknowledgements

I am thankful for my conversations with several Intel colleagues—especially Raj Yavatkar, John Barton, Brian Moore, and Milind Karnik—as well as validation forum discussions.

Priyadarsan Patra is a senior staff scientist at the Validation Research Lab of Intel's Microprocessor Technology Lab. His research interests include runtime, on-die, in-field validation applied to shared-memory multiprocessors; architectures and compilers for rapid and accurate emulation-based presilicon validation of many-core processors; high-level, rapid design exploration; and validation of on-chip communication fabrics. Patra has a BE in electronics and telecommunications engineering from the Indian Institute of Science, Bangalore, India, an MS in computer and information science from the University of Massachusetts, Amherst, and a PhD in computer sciences from the University of Texas at Austin. He is a senior member of the IEEE and a member of the ACM. He also leads a nonprofit organization to bring computer and communication literacy to remote and disadvantaged communities in Southeast Asia.

■ Direct questions and comments about this article to Priyadarsan Patra, 12688 NW Naomi Ln., Portland, OR 97229; priyadarsan.patra@intel.com.