Abstract. In the concluding chapter of this book we present our outlook at the future of processor verification. As the complexity of microprocessors grows and volumes of production increase, hardware vendors are subjected to greater risks of showstopper bugs. Yet, when it comes to verification, the microprocessor industry is often reactive, developing new solutions after critical escapes occur. Here, we advocate a proactive approach to verification, where the chip’s architecture is analyzed early in the design phase and pre-silicon, post-silicon and runtime techniques are integrated and developed according to a design’s needs. In this chapter we also overview upcoming trends in microprocessor architectures, such as heterogeneous multi-core structures and complex on-chip interconnects, and discuss how these trends will affect the nature of processor validation.

9.1 Advances and Trends in Processor Validation

Over the last four decades, advances in processor validation followed the evolution of the designs under verification themselves, rapidly increasing in capabilities and scope. Yet, as we discussed throughout this book, our ability to design digital circuits is constantly outpacing our ability to verify them, widening the verification gap with every new generation of microprocessors. The response of chip designing companies to this trend has been to employ mostly extensive, rather than intensive approaches. In other words, verification teams and capital investments have grown, while methods of validation have shown little change. Quantum improvements in verification technology and new techniques have been developed most often in response to the discovery of critical escape errors that had a major impact on a processor product. For example, the FDIV bug that escaped in the Intel Pentium processor, and the subsequent recall of defective parts have led to significant profit losses and caused a wave of negative publicity for Intel. This highlighted to all hardware ven-
dors the value of employing formal pre-silicon verification practices, which could have caught this error, in their design flows. As a response, Intel staffed its teams with several additional verification engineers that focused exclusively on formal verification aspects, particularly of datapath units, leading to effective contributions (in terms of discovering additional bugs early in the development phase) in subsequent product generations.

Similarly, escaped control logic errors showed that without microcode patching techniques, the only viable options left to manufacturers upon discovery of a bug were to recall the faulty components or to completely disable the affected on-die features, often leading to severe performance loss. The 2007 escaped bug in AMD’s newest Phenom processor, which was related to the memory subsystem, is undoubtedly pushing the industry to step up verification efforts in cache-hierarchy and on-chip interconnect, as well as to consider solutions that can ensure the runtime correctness of memory operations. This bug and the trends towards increasing the number of cores in a single chip, which in turn leads to extremely complex memory subsystems, has already stimulated much research in academic environments and has advocated the development of several of the techniques presented in this book.

9.2 A Proactive Approach to Verification

We believe, however, that in general a reactive approach to verification, where actions are taken only after a major slip-up, is fundamentally flawed: the financial blow of a nasty escape to a manufacturing company may be too much to recover with subsequent products. Production volumes today are much higher than, for instance, in the days of the first Pentium generation, thus recalling millions of defected microprocessors would cost a company today significantly more than the $420 millions reported by Intel in 1994. Delays in the release of a new processor have similar effects in today’s competitive markets. Therefore, hardware validation must be proactive: researchers in industry must analyze new designs for error vulnerabilities, anticipate the types of errors that may occur in them and design verification techniques to fulfill the project needs. To some degree, companies like Intel have already started to deploy this early validation analysis, however, more proactive steps must be taken for future generations, when designs become even more complex.

In addition to complexity and device density growth, recently developed processors are also incorporating new features, never tackled by verification before. As with all cutting-edge, untested technologies, these features present greater exposure to escaped bugs and, therefore, require the development of novel validation techniques, geared specifically for them. For example, in this book we had several discussions on validation solutions specific to multi-core designs. Before the appearance of this processor architecture in the early 2000s, much of vendors’ verification resources were spent on a device’s computational engine. In multi-cores, however, the focus has shifted, because datapaths are simpler and thus are easier to validate, while the communication between cores through the memory subsystem