The challenge of predicting and tuning a microprocessor’s net quality well before its tape out and eventual production cycle has never been more difficult than what we are witnessing in the late CMOS design era. I use the term “quality” to include metrics such as performance, which has been and continues to be the primary yardstick of evaluation in many markets; power and thermal characteristics; robustness, in terms of reliable and secure functionality; and so on.

Of course, other important end-user metrics that determine the ultimate success of the systems built around these microprocessors are things such as price and ease of use (software programmability).

Why is it inherently more challenging to predict these quality metrics today than it was a decade ago? One reason is that technology scaling has allowed a continued exponential increase in the number of transistors on a die, per Moore’s Law. More importantly, designers have deployed increasingly complex on-chip architectural paradigms, attempting to keep per-chip performance on its historical growth curve despite the slowdown in individual device speeds and overall chip frequencies.

During the 1990s, designers escalated the complexity of single-core chips to extract more instruction-level parallelism (ILP) while also increasing frequency. Owing to power and thermal constraints, the trend in the new millennium has shifted toward multicore chips with significantly lower frequency growth projections. Thus, high-end, general-purpose processor chips are beginning to look like systems-on-a-chip, often with complex on-chip buses and fabric (interconnect) structures.

Performance modeling needs have quickly evolved from trace- or execution-driven simulation of single-core (uniprocessor) chips to full-system simulations of multicore (multithreaded, multiprocessor) chips. And, to combat the tightening speed-of-simulation bottleneck, the use of hardware (FPGA) emulation technology to bring up complex simulation environments has emerged as a new trend. Of course, power, temperature, and reliability evaluation modules integrated into such pre-silicon evaluation platforms are almost mandatory in the late CMOS design era.

What is the effect of pre-silicon, integrated modeling and analysis on real design? Projections of metrics such as performance and power are what determine the survival of a product development plan. Because the accuracy of these projections is under constant challenge, modelers often make rather conservative assumptions about parameter values that are driven by uncertain technology nodes of the future. This conservatism often results in less-than-spectacular projections of quality metrics; if these don’t meet competitive expectations, project cancellations or major redirections often follow.

As pre-silicon models escalate in complexity, their validation has become much more difficult. The actual design team must invest a major fraction of its resources to pre-silicon verification and validation. Early-stage power-performance-reliability models (traditional, cycle-accurate software simulators) used in high-level chip microarchitecture definition often do not receive a large investment of resources during a chip-specific development project. Yet, the early-stage design decisions made with such models have a great impact on the end product’s actual quality. In addition, late-stage discovery of power or performance problems (based on more accurate design information and RT-level models) often cause major disruptions or even project cancellations. This is one reason that experienced pre-silicon modelers tend to be overly conservative (rather than optimistic) in modeling and projection for current-generation development projects that use integrated power-performance models.

Given these circumstances, a one-time large investment in developing accurate, fast, and rapidly configurable hardware (FPGA) emulators for pre-final-silicon modeling is indeed the way to go for most chip R&D activities of the future. This approach has the promise of very fast evaluation speeds, allowing full benchmarks to be run without sampling. The challenge, however, will be in designing such hardware simulators in a manner that allows easy model debugging and parameterization (perhaps through a software user interface).

This issue of IEEE Micro is devoted to computer architecture simulation and modeling. The guest editors—Timothy Sherwood and Joshua J. Yi—have done a tremendous job of organizing this very important theme issue. I hope you enjoy and benefit from these very timely articles.