Originally posted at EETimes.com on 06/04/2007 at:
http://www.eetimes.com/showArticle.jhtml;jsessionid=0JABUMEC53ZI4QSNDLRCKHSCJUNN2JVN?articleID=199703678
and reprinted with permission of EE Times.

Delivering on formal verification's original promise

Peter Feist, President and CEO, OneSpin Solutions

Functional verification is a severe bottleneck in SoC projects, with a significant impact on time-to-market, competitiveness and profitability. It involves two major tasks: first, ensuring that the SoC's modules and IP operate as expected and, second, ensuring that the interaction of these building blocks delivers the required chip-level functionality.

Advanced simulation-based verification is and will remain the primary chip-level verification workhorse for the foreseeable future. But what about module-level verification, which accounts for up to 70% of the total verification effort? Using established verification approaches, can we answer the fundamental question: can I be absolutely sure that my module/IP behaves as expected for all possible input scenarios?

The short answer is, no. None of the established approaches can detect all unexpected, unintended and unspecified behaviors – not even for the subset of input scenarios selected to reduce verification effort. Nor can these approaches identify remaining verification gaps. Rather, verification is terminated in the full knowledge that only a subset of the possible behaviors has been verified. The EDA industry has attempted to address this high-risk situation with a plethora of incremental advances to simulation-based verification, such as adding assertions, piggy-backing formal verification for corner case bug-detection, and hardware acceleration.

Do these advances answer the fundamental question? No. To the contrary, the trends are all in the wrong direction. Functional verification complexity, effort and costs are growing; tool productivity gains are failing to keep pace with increasing design complexity; and the error escape rate shows no sign of abating. The result? Redesign and respin rates remain stubbornly high. Continuing to rely on simulation for module/IP verification – no matter how advanced, no matter how beefed up with additional tools – cannot reverse this trend.

Is this a reason for despair? Not really. Look at what happened to synthesis verification a decade ago. Gate-level simulation was replaced by equivalence checking – at that time, a new formal verification approach. Though a paradigm shift requiring new tools, methodologies and skills, equivalence checking was adopted because it proved to deliver far superior results. It is a lean approach that does not need a lot of software licenses and expensive hardware platforms; its inherent parallelism delivers considerably higher productivity than sequential gate-level simulation; and it leaves no verification gaps. Equivalence checking eliminated the synthesis verification bottleneck.

Is a corresponding paradigm shift possible in functional verification? Yes. The new paradigm is *complete* formal functional verification. Completeness is the crucial point. Verification is complete when all output signals have been verified to have their expected values at any point in time for any possible input scenario. This is the way – and the only way – to eliminate all of the bugs. And you can't achieve better verification quality than that.

What equivalence checking did for synthesis verification a decade ago, complete formal functional verification can do for digital modules and IP today. It is a one-tool, one-methodology, one-workstation solution; it delivers 2x to 5x higher productivity than a thorough simulation-based verification; and its completeness ensures 100% input scenario coverage and 100% output behavior coverage. It leaves no doubt that the module behaves as expected – always. It delivers a true functional sign-off. And this was the original promise of formal verification.

Do all formal functional verification approaches deliver these results? No. Historically, formal functional verification has failed to deliver on its promise. Engineering teams have continued to invest in existing verification environments, using formal verification only as an ancillary aid.

However, there are now mature, complete formal verification solutions that have transformed formal functional verification into a robust engineering practice. And they have been successfully applied to a broad range of digital modules and IP, such as large processors, memory controllers, bus interfaces, peripherals, as well as communication, error correction, and multi-media components.

Is complete formal verification a silver bullet? No, it is not, just as equivalence checking is not the silver bullet for *all* possible comparisons of design representations. But for a broad range of digital modules and IP, complete formal verification is known to deliver far superior results in terms of quality, effort and costs. That is why it has the potential to replace simulation-based verification for diverse types of digital modules and IP.

Today, huge investments are made in technology that is known not to deliver the required results. With a fraction of this investment, a verification team can assess the productivity and capabilities of complete formal verification. The recent Aberdeen Group Report "Electronics - Correct by Design" reports that best-in-class companies expend on average 42% of their effort on verification, compared to the industry average of more than 60%. And the "industry average" achieves worse results. The report tells us why. The best-in-class companies use best-in-class verification technologies to meet aggressive deadlines and stringent quality targets. So the way forward is clear.

The means to efficiently achieve error-free modules and IP is not a vision, but a reality today. IP companies, SoC companies and the EDA industry can drive the paradigm shift. They can adopt complete formal functional verification as a standard engineering practice – and reverse the trends of recent years. The time has come to make that shift.

(For a detailed explanation of complete verification see the EETimes article: **http://www.eetimes.com/showArticle.jhtml;jsessionid=IGZ2FP35N3ADWQSNDLPSKH0CJUNN2JVN?articleID=197005268).**