The Last Byte

Design for Verification?

Todd Austin
University of Michigan, austin@umich.edu

The pecking order in most design teams is clear: Marketing people dream up products, designers implement them, and verification engineers are stuck trying to get them to work. When the work is complete, management often credits designers for product successes and blames verification engineers for the failures.

Consider, if you dare, a fifth dimension beyond that known to engineers. The middle ground between performance and correctness is the dimension of *imagination*. It is an area that we call the design-for-verification zone (imagine harp music playing now).

For example, marketing ordered engineers at my company to build an implementation of the new Rijndael cipher that could run at a speed of at least 1 Gbps. Those marketing types obviously haven't studied the Rijndael algorithm—it's computationally intensive and has no parallelism; 1 Gbps is unlikely.

The formal verification engineer in charge of the project made my design team hold off on detailed design until we could come up with a design that provided a "separation of concerns" between correctness and performance.

We worked for two weeks with the formal verification team and finally thought up a clever design. We would build two processors, a core cryptoprocessor and a checker processor. The core cryptoprocessor was my original vision of the design: an aggressive, over-the-top Rijndael implementation. The checker is a simple microcontroller that checks the cyclical redundancy code (CRC) of packets processed by the cryptoprocessor. Computing a CRC check is much simpler than the Rijndael algorithm, so we wouldn't need much muscle for the task.

During normal operation, the cryptoprocessor encrypts and decrypts communication packets, and the checker processor verifies the integrity of the plain-text output of decryption, using the fast CRC check. If a packet ever fails this check, the packet is once again encrypted or decrypted, but this time using the microcontroller and a reference Rijndael implementation in software.

Do you see the separation of concerns? The cryptoprocessor is concerned with performance. The checker processor is concerned with the integrity of the encryption/decryption computation. If the CRC fails, the computation is reprocessed using the slower, simpler, and more reliable microcontroller. Since the checker processor would fix any bug in the cryptoprocessor, the checker is the only component that need be completely correct.

In the end, the separation of concerns turned out to be a huge win. We delivered an aggressive core design to the verification team, which they coupled with their checker design. The formal verification team fully verified the checker; they also applied formal verification techniques to the checker software. However, verifying the checker software was less of a concern because we could upgrade it in the field. We could even push the cryptoprocessor clock harder than originally planned because we knew that the

continued on p. 77

0740-7475/01/$10.00 © 2001 IEEE

continued from p. 80

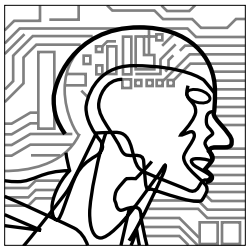
checker processor would detect and fix occasional core processor timing errors. We simplified the cryptoprocessor design by moving the entire cipher setup functionality to the checker processor. We delivered the design early and beat the performance target. Best of all, we beat Acme CryptoWidgets to market by three months (and more harp music plays).

Sure, this is fiction, but must it be? These are trying times for the verification engineer. Verification teams and time to market grow as engineers struggle to verify increasingly complex designs. Incorporating verification as a first-class consideration of the design process is one way to buck the trend.

At the University of Michigan, my colleagues and I are exploring ways to ease the verification burden of complex designs. The DIVA (Dynamic Implementation Verification Architecture) project (http://www.eecs.umich.edu/diva) has developed a clever microprocessor design that provides a near complete separation of concerns for performance and correctness. The design employs two processors: a sophisticated core processor that quickly executes the program, and a checker processor that verifies the program by reexecuting all instructions in the wake of the complex core processor. By leveraging address and branch predictions from the core processor, the checker design is both simple and fast. Karem Sakallah, his students, and I have successfully demonstrated that the checker processor lends itself to formal verification.

I challenge formal verification engineers to find ways to make their designs more readily verifiable, and to push these techniques into the design cycle. By separating correctness and performance concerns, you minimize the portion of the design that must be completely correct, reducing your verification burden. At the same time, you will create new opportunities for optimization, resulting in faster, more energy-efficient, and less-complex designs. ■

Todd Austin is an assistant professor of electrical engineering and computer science at the University of Michigan. Contact him at taustin@eecs.umich.edu.



IEEE COMPUTER SOCIETY
design automation
TECHNICAL COMMITTEE
NEWSLETTER

A MESSAGE FROM THE CHAIR

The 38th Design Automation Conference (DAC) was held 18-22 June 2001 in Las Vegas, Nev. I attended, as did several other executive committee members. Mark your calendars now for the 2002 meeting in New Orleans, La., 10-14 June.

We had an executive committee meeting, and minutes of it will be available on the DATC Web site, as well as in a condensed version here in the next issue. The agenda included a treasurer's report, discussion regarding this column in *IEEE Design & Test of Computers*, the DATC Web pages, and a welcome to Steve Grout, Chair Elect. Also discussed was the William J. McCalla ICCAD Best Paper Award, the TAB meeting in Seattle, the VHDL RF/MW Initiative, high-performance simulation, the SIG-DL interface, the EDPS report, the *IEEE Design & Test of Computers* EIC selection, and DATE surplus and DATC.

Charles Rosenthal, past chair of DATC, attended a TAB meeting in Seattle and minutes of it will be available on our Web pages as well as in next issue's column.

Joe Damore

IEEE CIRCUITS AND SYSTEMS SOCIETY PIONEER AWARD

This award honors a person or person with outstanding and pioneering contributions in developing academic- and industrial-research results into industrial applications and/or commercial products.

This year's award was given to Aart J. de Geus, a technologist, scientist, intellectual, and blues guitarist. De Geus serves as chair and chief executive of Synopsys, and has been the driving force behind the company since its inception in 1986.

He has grown the company from a start-up focused on synthesis—a breakthrough technology that he spearheaded at General Electric—to an electronic design automation (EDA) industry leader focused on system-on-a-chip technology. One of the world's leading experts in EDA, de Geus was recently elected chair of the board for the Electronic Design Automation Consortium (EDAC), the international association of companies engaged in the development, manufacture, and sale of design tools to the electronics engineering community.

He is also a former chair of the Computer and Network Design (CANDE) group and has served on the board of governors of the IEEE Circuits and Systems Society. IEEE made him a Fellow in January 1999.

CONTRIBUTIONS TO THIS NEWSLETTER: Please send any contributions concerning electronic design automation that you would like included in this DATC Newsletter to Joe Damore, 36 Hagan Drive, Poughkeepsie, NY 12603; phone +1 845 462 1364; fax +1 845 463 4311; e-mail JDamore@prodigy.net.