3D Integration for Introspection

In today's complex processors, specialized profiling and introspection hardware would be incredibly beneficial to software developers, but most proposals for its addition would increase the cost of every die manufactured. Modular, ''snap-on'' analysis hardware, stacked vertically with the processor die using a 3D interconnect, could be included with developer systems to assist in debugging and testing, and omitted from consumer systems to keep them economically competitive.

Shashidhar Mysore
Banit Agrawal
Navin Srivastava
Sheng-Chih Lin
Kaustav Banerjee
Timothy Sherwood
University of California,
Santa Barbara

Developing high-quality software for a modern computer system is no easy task. Performance-critical applications are likely to execute for quadrillions of instructions and operate in a complex environment with multiple runtime components. Moreover, they are increasingly responsible for managing various architectural resources, including power and hardware threads. To battle this complexity, application developers depend increasingly on sophisticated software analysis tools. Although developers can perform mixed static-dynamic analysis completely in software through binary instrumentation, the amount of analysis possible at test time is bounded by the tolerable performance impact. In long-running or interactive programs, this is especially critical. To enable runtime analysis with low overhead, many researchers have proposed the development of specialized on-chip hardware modules that can help software developers build applications that are more secure, bug free, and efficient. Unfortunately the cost of adding this extra hardware to every system is a major deterrent to adoption and, thus far, hardware support for developers has been limited to very simple hardware performance counters.

We propose a new, modular way to add analysis hardware to next-generation processors through the use of a 3D interconnect. Several 3D-interconnect technologies,[1] such as interdie vias, are currently being evaluated in industry as a way to stack multiple chips together. Potential applications include stacking DRAM or larger cache directly onto the processor die to alleviate memory pressure and designing stacked chips of multiple processors. We describe the details of this technology more fully elsewhere:[2] The main idea is to bond two pieces of silicon together to form a single chip, and connect the two active layers of the silicon through interdie vias (called posts), which run vertically between them.

The capability to interconnect multiple active layers means that we can consider adding to a processor an optional layer specifically for analysis, which would have easy access to most of the important system signals. Manufacturers could sell processors with this capability to software developers, and simply omit this extra analysis layer from commodity systems. Figure 1 compares the traditional and stacked approaches to integrating analysis hardware.

0272-1732/07/$25.00 © 2007 IEEE

Published by the IEEE Computer Society.

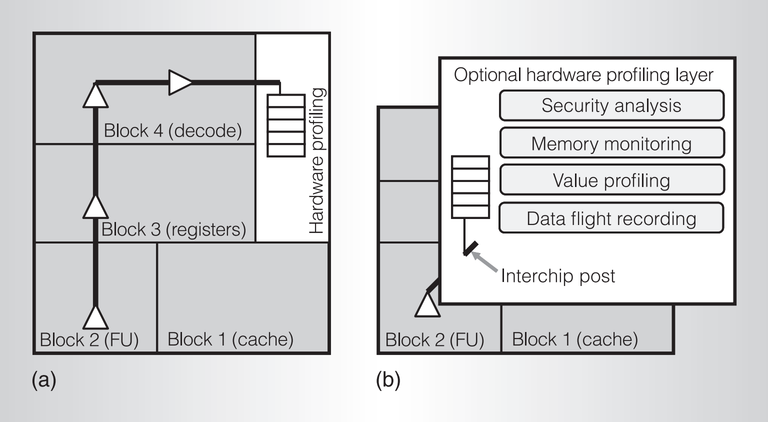

Figure 1. The traditional approach to attacking the hardware-profiling problem involves integrating specialized profiling functionality on the same die as the processor (a). Gathering information requires long global wires that necessarily cross multiple functional blocks. Achieving high performance requires buffers or pipeline latches, which in turn require access to silicon—all of which makes for a big mess. Alternatively, a stacked approach (b) requires only a single buffer to drive the post up to the analysis layer, which would be an optional feature for software developers.

3D introspection overview

The interchip 3D interconnect could take the form of any of several competing technologies, including chip bonding, multichip modules (MCMs),[3] chip-stacking with vias,[4] or even a wireless superconnect.[5] Although chip bonding and MCM technology are already used in a variety of embedded contexts,[6] several major industry consortia are heavily researching more aggressive interconnect technologies. Intel, for example, has been investigating 3D integration to include extra layers of cache. If a chip manufacturer includes this technology to add extra functionality for consumer machines, it would be an incremental step to add an additional optional analysis layer. Because chip stacking with vias is the most mature next-generation superconnect technology, and because major corporations such as Intel, IBM, and Inion have made major investments in its development, we use 3D integration as a starting point for our evaluation.

In this article, we describe the potential of 3D interconnect technology to enable new forms of introspective chips. In the original version of this article, we precisely quantify both the chip bandwidth requirements for full introspection and the relevant characteristics of 3D interconnect technology.[2] We further quantify the increase in area, the interconnect overhead, and both the power and thermal impacts of such a design. Here, we elaborate some of the advantages of 3D introspection over traditional hardware integration.

Reducing introspection routing problems

A significant advantage of building monitoring hardware on top of the main processor is that this arrangement drastically reduces interconnect congestion. As practitioners agreed during a recent discussion of the challenges facing performance-monitoring hardware (2005 Workshop on Hardware Performance Monitor Design and Functionality), gathering data from all over the chip for centralized analysis creates a global interconnect that causes serious headaches. Such an interconnect would have to cross almost every design block and would consume a good deal of the top metal layers. Not only would the interconnect have to join these different regions of the processor, it would also have to run very fast. For example, capturing the address of every load instruction would require a bandwidth of about 64 Gbps. This data rate, coupled with the long distances required, necessitates wire buffering and even pipeline latches. This, in turn, requires silicon to be reserved in many different blocks to give the wires access to the needed transistors; Figure 2 more clearly illustrates the problem, showing the different blocks in a Pentium 4 processor floorplan and our estimate of where an analysis engine would have to gather profile data. In a full-custom design, on-die monitoring hardware requires a significant amount of engineering effort spread across every level of the physical and architectural design. Many companies are reluctant to add the complexity of these additional global nets to their designs.

Instead of routing performance data through other blocks, we can have interdie vias move data out of plane to a layer specially constructed for gathering and analyzing runtime information. Of course, this does

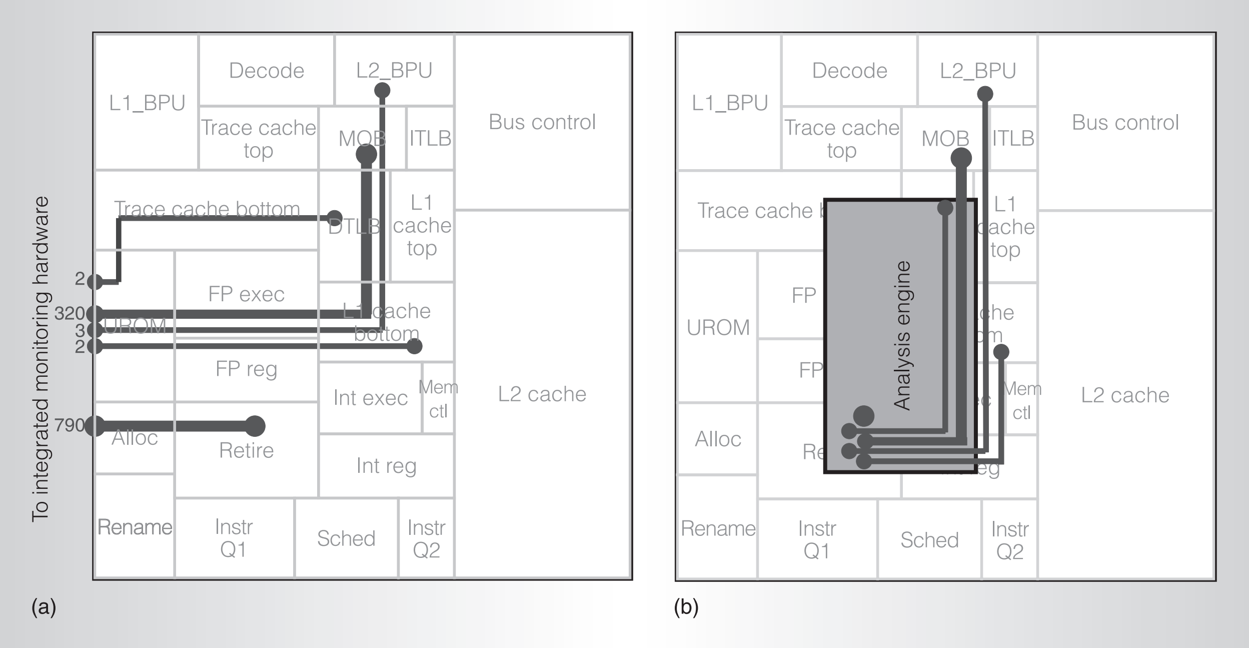

Figure 2. Locations for gathering chip profile data on the P4. If the analysis engine is located on the periphery of the chip (a), significant signals must be global, even if engine placement minimizes wire length (as in this example). Values at the ends of the lines indicate numbers of bits. If the analysis engine is stacked on top, connected with posts (b), there is no new global routing on the processor layer. Global routing on the analysis layer is minimal because the analysis can be centrally located.

not come free: Each interlayer via (or post) is around 5 microns on a side. Space must be reserved for the gates that drive the posts, and switching these large pieces of metal will require some amount of power. While there is some overhead, the area needed for the posts is localized to the position of the tap (where the profile data is gathered), and no extra coordination is required between the designers of the different blocks. Because the wires can be much shorter, the power overhead is actually less than that for on-die routing. (In our original article, we quantify the wires in terms of the number of wire buffers necessary, the area they occupy, and the power they consume compared to on-die routing.)

Decoupling developer needs from user systems

In addition to significantly reducing interconnect congestion, 3D integration can also reduce the total cost to the user. Every user must pay the cost for an integrated hardware monitor, although most will neither use nor need such functionality. In the US, there are an estimated 225 million PCs in use—more than three computers for every four people—but only 700,000 programmers (Computer Industry Almanac; http://www.c-i-a.com). Even if every programmer demanded a system with hardware support for debugging, the market for such devices would still be several orders of magnitude less than that for commodity PCs. By allowing fabrication of an analysis model with steps that are complementary to (but separate from) the main processor, stacked interconnect offers the potential to add monitors on just a small subset of devices without impacting the overall cost of the main processor. Just to be clear, we envision that the processor would always be fabricated with connections for hardware monitoring. The difference between the system for the consumer and that for the developer would be only whether the hardware monitor devices are actually stacked on top.

Opening the door to heavyweight analysis

The final major advantage of stacking a hardware monitor on the main processor is the potential this arrangement has to open new avenues of research in heavyweight dynamic program analysis. Current runtime systems are heavily constrained by both the overhead of analysis and the very limited monitoring bandwidth available. A full analysis of the potential of a stacked system to enable new types of dynamic analysis is beyond the scope of this article; however, many examples of such analysis already exist. For example, Mondrian memory protection extends the idea of memory protection to include protection on arbitrarily small ranges of memory with permissions for read, write, and execute, and has been shown to be effective at identifying many types of software bugs through emulation in software.[7] Unsafe pointer dereference analysis (such as fat pointers[8]) or unsafe memory region tracking[9] can identify the code that is most likely to be exploited by worms and other network-based attacks. Tracking dataflow tags through the architecture can point to the suspicious use of data so that worms can be identified in the wild,[10,11] and data flight recording can allow the playback of architectural state when bugs or attacks are identified.[12]

3D technology

Now let's look at the interdie through-via 3D technology and an example hardware analysis engine that we use to evaluate our proposal for a 3D introspection engine.

Manufacturing posts between two dies

One popular method of fabricating 3D integrated chips is to bond together two fully processed wafers on which transistors and wires have been fabricated, so that the wafers overlap completely. The manufacturer first thins the top wafer to approximately 10 to 50 microns. Optically adjusted bonding is then used to stick this layer to the bottom wafer, using a 2-micron organic adhesive layer of polyimide. After metallization of both layers and prior to the bonding process, electrical connections must be created between the two wafers. The connections are made by interdie vias (posts), which are etched through the intermetal-layer dielectric on the top wafer, the thinned top silicon wafer itself, and the cured adhesive layer. The interdie vias are then formed in these etched holes using chemical-vapor-deposited (CVD) tungsten, which can withstand the high temperatures (400°C) of the wafer-bonding process. In a modern process, these vertical interconnects typically have cross sections of 5 microns × 5 microns and heights of 30 to 40 microns, whereas a normal metal wire's cross section is on the order of 1 micron × 1 micron (2001 *International Technology Roadmap for Semiconductors*, http://www.itrs.net).

A second approach relies on thermocompression bonding between metal pads in each wafer. In this case, Cu-Ta (copper-tantalum) pads on both wafers serve as the electrical contacts between the interchip vias on the top thinned silicon wafer and the uppermost interconnects on the bottom silicon wafer. Banerjee et al. describe these and other processes for 3D integration of VLSI chips.[6]

Figure 3 shows the cross section of the stacked processor and analysis layers—including the active layer (where CMOS logic is designed), metal layers (for routing), vias (connecting metal layers), buffers (driving vias), and vertical posts (3D interconnect). We will use these terms to explain the advantages and overhead of introspective 3D chips.

Details of the interconnect

Now let's look at some important aspects of the interconnects and associated buffers that are involved in our evaluation (our earlier article includes a detailed analysis).[2] To compare the overhead of the 3D interconnect and buffers with a 2D system, for each we calculate the active layer area overhead (area of active layers), metallization area overhead (area of metal layers), and power consumption. The analyses for 3D and 2D are completely different because the two approaches differ in number of buffers and complexity of interconnect. The 2D design must drive the interconnect across the chip, so it requires more buffers; in the 3D design, the interconnect goes vertically up to the layer above, which requires just one buffer in the processor layer. However, we need to route the

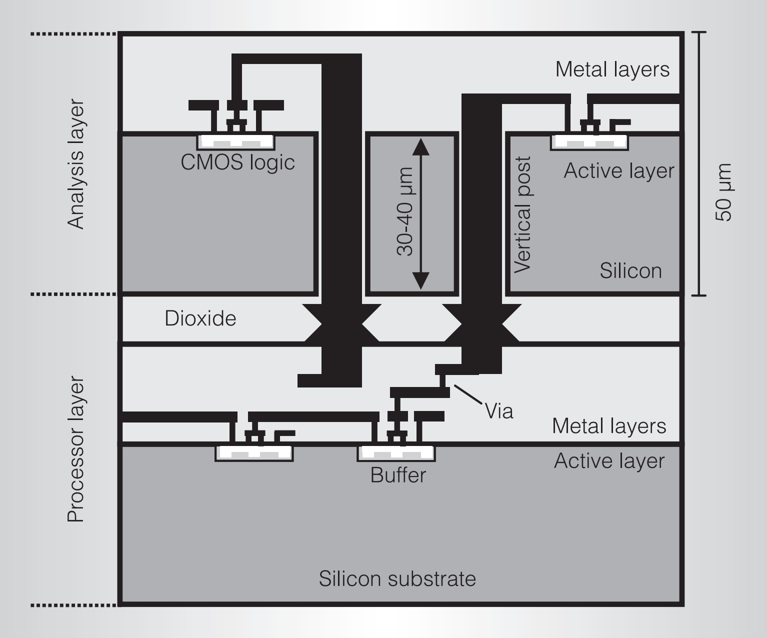

Figure 3. Cross section of the introspective 3D chip, with processor and analysis layers separated by the dioxide layer. The vertical posts are 50 microns in length, with a 5-micron × 5-micron cross section.

interconnects in the analysis layer of the 3D chip from the vertical posts, which are driven from the processor layer. For the 3D interconnect, we use a post 50 microns in height. Figure 4 shows the orientation of these posts.

Example hardware monitor

Designing an analysis engine capable of performing a variety of online program analyses is no trivial task. At one extreme, a counter is probably the simplest mechanism to aid program analysis; at the other extreme, complex analysis processors could run tens of profiling algorithms, enable multiple optimizations, and perform analysis over different profile data of the running program—all at the same time. Rather than explore this massive design space, our purpose here is to examine a concrete example to argue the feasibility of an introspective 3D chip. So, we begin by analyzing a simple example engine based on prior work, which we use as the layer-2 analysis engine in the proposed 3D architecture. For all analysis and design processes we present in this article, we assume a 2-GHz clock rate and 0.13-micron technology at 1.1 V.

Looking at operations that a programmable analysis engine might most frequently perform, we find that the core functions are an associative lookup followed by counter increments or simple manipulations on a set of counters, and a periodic sequence of complex computations. With the goal of providing at least these features on our layer-2 analysis engine, we began with the architecture of an analysis engine based on the profiling coprocessor proposed by Zilles and Sohi.[13] We modified the coprocessor architecture to suit the 3D interconnect architecture, made room for more complex analysis than earlier by adding features to the microprocessor, and increased the amount of memory.

Enabling programmers and software developers to more easily track down bugs, identify performance bottlenecks, and secure their code against attacks must be one of the primary concerns of system designers at all levels, including computer architects. Even today, software bugs are so damaging and widespread that they cost the US economy an estimated $59.5 billion annually (more than half a percent of the gross national product.[14] Although it is certainly not possible to remove all errors, the National Institute of Standards and Technology estimates that an improved testing and analysis infrastructure could

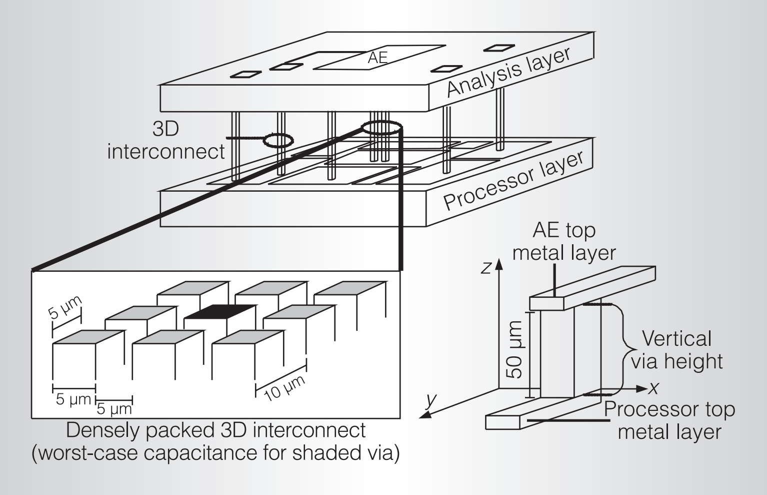

Figure 4. 3D interconnects and their dimensions. The enlarged cross section of posts gives an idea of the worst possible coupling capacitance.

eliminate more than a third of the cost associated with bugs.[14] The rapidly increasing hardware complexity exposed to programmers on desktop and server machines in the form of threading, parallelism, and complex application middleware will do nothing to help the problem of inefficient and buggy software. To cope with this complexity, and to ensure the quality of software infrastructures, an increased reliance on sophisticated software analysis and testing tools seems inevitable. The challenge is enabling these techniques with a minimum of impact on typical user systems.

One of the biggest advantages of our approach is that it decouples the cost of specialized analysis hardware from the highly cost-sensitive consumer market. It lets users continue to buy cheap, high-performance machines because the only extra hardware they pay for are stubs. The hardware stubs left in the consumer machines increase area by no more than 0.021 mm^2 and power by no more than 0.9 percent—numbers that careful design might reduce. At the same time, developers and users both benefit from the increased analysis power of dynamic monitoring tools. Although our argument here is economic in nature, elsewhere we have used the metrics of area, power, routability, and temperature to quantify one possible design.[2] While the thermal impact of stacking two hot cores together is always a concern in 3D design, we show that the effect is manageable for both our sample system and for a system eight times as powerful. Given that developers would need to pay more for this additional hardware anyway, the incremental cost of additional cooling should be minor. MICRO

References

1. B. Black et al., ''Die Stacking (3D) Microarchitecture,'' *Proc. IEEE Int'l Symp. Microarchitecture* (Micro 06), IEEE CS Press, 2006, pp. 469-479.
2. S. Mysore et al., ''Introspective 3D Chips,'' *Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems* (ASPLOS XII), ACM Press, 2006, pp. 264-273.
3. C. Massit and N. Gerard, *Three-Dimensional Multichip Module*, US patent 5,373,189, Patent and Trademark Office, 1994.
4. W.R. Davis et al., ''Demystifying 3D ICs: The Pros and Cons of Going Vertical,'' *IEEE Design & Test*, vol. 22, no. 6, Nov.–Dec 2005, pp. 498-510.
5. N. Miura et al., ''A 195Gb/s 1.2W 3D-Stacked Inductive Inter-Chip Wireless Superconnect with Transmit Power Control Scheme,'' *Proc. IEEE Int'l Solid-State Circuits Conf.* (ISSCC 05), IEEE Press, 2005, pp. 264-265.
6. K. Banerjee et al., ''3-D ICs: A Novel Chip Design for Improving Deep Submicron Interconnect Performance and Systems-on-Chip Integration,'' *Proc. IEEE*, vol. 89, no. 5, May 2001, pp. 602-633.
7. E. Witchel, J. Rhee, and K. Asanovic, ''Mondrix: Memory Isolation for Linux Using Mondriaan Memory Protection,'' *Proc. ACM Symp. Operating Systems Principles* (SOSP 05), ACM Press, 2005, pp. 31-44.
8. G.C. Necula, S. McPeak, and W. Weimer, ''CCured: Type-Safe Retrofitting of Legacy Code,'' *Proc. ACM SIGPLAN-SIGACT Symp. Principles of Programming Languages* (POPL 02), ACM Press, 2002, pp. 128-139.
9. S.H. Yong and S. Horwitz, ''Protecting C Programs from Attacks via Invalid Pointer Dereferences,'' *ACM SIGSOFT Software Engineering Notes*, vol. 28, no. 5, pp. 307-316.
10. G.E. Suh et al., ''Secure Program Execution via Dynamic Information Flow Tracking,'' *Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems* (ASPLOS XI), ACM Press, 2004, pp. 85-96.
11. J.R. Crandall and F.T. Chong, ''Minos: Control Data Attack Prevention Orthogonal to Memory Model,'' *Proc. IEEE/ACM Int'l Symp. Microarchitecture* (Micro 37), IEEE CS Press, 2004, pp. 221-232.
12. S. Narayanasamy, G. Pokam, and B. Calder, ''BugNet: Continuously Recording Program Execution for Deterministic Replay Debugging,'' *Proc. Int'l Symp. Computer Architecture* (ISCA 05), IEEE CS Press, 2005, pp. 284-295.

13. C.B. Zilles and G.S. Sohi, ''A Programmable Co-processor for Profiling,'' *Proc. Int'l Symp. High-Performance Computer Architecture* (HPCA 01), IEEE CS Press, 2001, p. 241.
14. *The Economic Impacts of Inadequate Infrastructure for Software Testing,* NIST Planning Report 02-3, Nat'l Inst. of Standards and Technology, 2002.

Shashidhar Mysore is a PhD candidate in the Department of Computer Science at University of California, Santa Barbara (UCSB). He previously worked as a research assistant at the Indian Institute of Science, Bangalore. His research interests are in computer architecture, specifically in the design of algorithms and hardware accelerators to aid runtime program and system analysis. His work on Range Adaptive Profiling received the Best Paper Award at the Code Generation and Optimization Conference in 2006. Mysore received his BTech from the BMS College of Engineering, Bangalore. He is a student member of the IEEE.

Banit Agrawal is a PhD student in the Department of Computer Science at UCSB. His research interests include network and security processors, three-dimensional ICs, memory design and modeling, and power-aware architectures. He has a master's degree in computer science from University of California, Riverside, and a BTech(H) from the Indian Institute of Technology, Kharagpur. He is a member of the IEEE.

Navin Srivastava is a PhD candidate at UCSB studying nanometer-scale VLSI interconnect issues. Previously, he spent three years working in the electronic design automation industry. Srivastava received a BTech with honors from the Indian Institute of Technology, Kharagpur, India, and an MS in electrical and computer engineering from University of California, Santa Barbara. He was a corecipient of the Outstanding Graduate Student Paper Award at the VLSI Multilevel Interconnect Conference, 2005. He is a student member of the IEEE.

Sheng-Chih Lin is a PhD candidate in the Department of Electrical and Computer Engineering at UCSB. His research interests include electrothermal modeling and analysis, variation-aware circuit design optimization, and thermal management for nanoscale CMOS and high-performance ICs. He received a BS in electrical engineering from National Taiwan University, Taiwan, and is a student member of the IEEE.

Kaustav Banerjee is an associate professor of electrical and computer engineering at UCSB. Before joining the UCSB faculty in 2002, he was a research associate at Stanford University's Center for Integrated Systems. His present research focuses on nanometer-scale issues in high-performance and low-power VLSI, and on circuits and systems issues in emerging nanoelectronics. Banerjee has a PhD in electrical engineering and computer sciences from University of California, Berkeley. He is a senior member of the IEEE.

Timothy Sherwood is an assistant professor in the Department of Computer Science at UCSB. His research interests are in computer architecture, specifically in the development of novel high-throughput methods by which systems can be monitored and analyzed. For example, his program phase analysis tool, SimPoint, is used by industry partners and many academics to guide the design of their largest microprocessors. Sherwood received his BS from University of California, Davis, and his MS and PhD from University of California, San Diego. He is a member of the IEEE.

Direct questions and comments about this article to Shashidhar Mysore, Univ. of California, Santa Barbara, Computer Science Dept., Engineering I, Room 2104, Santa Barbara, CA 93106-5110; shashimc@cs.ucsb.edu.