A *D&T* Roundtable

Challenges for Low-Power and High-Performance Chips

Microprocessor and other IC performance continues to improve at historic rates, with no visible end in sight for the next 10 years. However, we are starting to encounter a power wall. This is true for high-performance components as well as for low-power chips with a very limited energy budget offered by batteries. We need to find ways to manage power and energy consumption on all fronts—technology, design, and architecture—without compromising performance. Otherwise, we may face discontinuation of Moore's law for the semiconductor industry in the near future. This would be triggered not by any difficulty in the scaling of process technology but by formidable barriers posed by packaging and cooling, inefficacy of power delivery, and energy constraints dictated by battery technology, which is advancing at a very lukewarm pace.

IEEE Design & Test thanks roundtable participants Ching-Te Chuang (IBM), Shih-Lien Lu (Oregon State Univ.), Krishnamurthy Soumyanath (Intel), Hamid Partovi (AMD), and Takayasu Sakurai (Univ. of Tokyo). *D&T* gratefully acknowledges the help of Vivek De (Intel), our moderator; Kaushik Roy (Purdue Univ.), our Roundtable Editor who organized the event; and Yibin Ye (Intel), who acted as our photographer.

Special thanks go to the IEEE Computer Society's Test Technology Technical Committee (TTTC) for sponsoring this event and the VLSI Circuits Symposium for hosting it.

D&T: What are the most significant barriers to the evolution of low-power systems in the next five to ten years?

Partovi: ICs, among other requirements, have either of two primary specifications: low power or high performance. Some technology, circuit, and architectural choices benefit both; some don't. In the past, microprocessor designers have not worried about power except to ensure the integrity of the supply network. Since I have designed standalone SRAMs, I know that, in contrast, memories have stringent power budgets.

We can certainly reduce the power supply in processors and make a low-power machine from a high-performance machine. However, there has been one very successful attempt to design from scratch for low power: DEC's StrongArm. The DEC designers started with a clean slate and designed it specifically for low power.

D&T: What architectural, circuit, and technological choices most affect performance and power?

Partovi: Let me outline a designer's choices that affect power and performance. Architecturally, speculative execution leading to circuit duplication adversely influences power, while using pointers instead of shifting large data blocks saves power. Also, the designer must consider memory hierarchy, banking, and cache associativity. From a circuit perspective, the issues include dynamic versus static design, edge-triggered or level-sensitive latching, conditional clocking, and low-swing data

0740-7475/98/$10.00 © 1998 IEEE

transmission. And from a technology standpoint, scaling yields to higher leakage to attain higher ID sat. Probably, one of the most important things to consider is the increasing difficulty in controlling device parameters even on the same die: identically drawn devices will show different characteristics.

D&T: Hamid has touched on many of the issues. Let's focus on process now.

Soumyanath: Three components constitute low-power technology: process, circuits, and architecture. When you talk to a process person about low-power technology, you'll hear responses that indicate a key problem in achieving low-power technology from the process point of view: communication is fundamentally broken between the circuit and process engineers. We cannot ask process people to provide 600 or 700 microamps of drive current and provide one nanoamp of leakage at the same time. It really forces them into a corner. We have to decide what we really need.

D&T: Why is interaction between the circuit and the process people shaping up this way?

Sakurai: The circuit community may use a back gate to control the V_{TH} in a very optimized way. Then, we'd need a device that has a higher gamma, which is a back-gate bias effect coefficient. That's a different request from the conventional request. When circuit engineers come up with a new idea, it affects the optimization of the device level. If the pursuit is a joint effort, we can search a wider area.

Partovi: We've been studying devices for reliability for a long time. To ensure device lifetime, we've had to drop the power supply voltage. The demand for high drive currents while dropping the supply voltage leads to high leakage. To address this, designers have used dual-V_T technologies in which lower V_T devices are strictly used in critical paths.

Chuang: The leakage problem is always in the technology; we have to design for various process corners. The worst case for leakage is a burn-in condition. Different companies may have different burn-in strategies. At IBM, we burn-in at 140 degrees, 1.5 V_{DD}. However, we find that we still have extremely large leakage in this burn-in condition for scaled technology. It can even melt our chips. We'll be facing this dilemma with future technology scaling. The scenario has to change. Leakage can also do some other harm in memory—in certain structures, nodes may lose their data.

D&T: How much gain should we expect if we change the burn-in conditions to the worst-case operating conditions?

Chuang: The effectiveness of the traditional burn-in is decreasing very rapidly. IBM is working to define the right burn-in strategy, but it is not an easy job to do.

I also have concerns about dynamic V_T control where the substrate is affected. Though we probably can do it in a small circuit, doing it globally in a large-scale design like a microprocessor becomes extremely difficult. You also need very complicated EDA tools to trace what's happening. It's very critical for microprocessors.

Partovi: One issue that's never been resolved is whether dynamic or static design is best for low power. With static design, multiple spurious transitions are possible per node. Dynamic circuits must be dual-rail if a designer is serious about them. So half of the nodes transition twice in a cycle. It is really not clear which style is best for low power.

D&T: Are the dynamic circuits that have always been used by circuit designers to get the best performance becoming the barrier to the tolerance of high leakage? Should we only use static CMOS for all our designs?

Partovi: This is interesting. If device leakage is too high, dynamic, and in particular, dynamic wired-OR structures such as RAM arrays are affected the worst. This is why we use longer channel devices or limit the number of pull-downs for these to bound leakage.

D&T: Dynamic circuits could get worse if hit by leakage.

Partovi: There is no question that the reason circuit designers are reluctant to lower the demands on leakage reduction is because they want to use dynamic circuits. They're reluctant to tell process engineers to allow more leakage. But on the other hand, once the design technique has become explicitly leakage resistant, we'll get braver, and we'll quit whining about the high leakage current. Then, maybe we'll let the process people give us high-transistor performance, perhaps at high leakage current. Circuit designers have a responsibility to deal with this problem.

D&T: If we want to lower the supply voltage and have the same drive current as in the past, it seems that no one can help us unless we accept high leakage. How do we address this problem?

Partovi: We need higher V_T or longer channels for particular circuits. Basically, we have used longer channels for some leaky circuits such as the RAM pass gate. Meanwhile, for critical paths we have used lower V_T transistors..

D&T: Interconnects are also becoming a problem.

Architects want communication over long distances in the definition of the architecture and microarchitecture. What are the possibilities of addressing interconnect problems from the architecture level?

Lu: Microprocessors started out in the late 1960s as a serial type of execution with each instruction taking several cycles. They evolved into pipeline structures, raising the instructions per cycle to close to 1. Now we use superscalars with multiple pipelines. This trend shows that the MIPS values have increased exponentially, and they will continue to grow. This growth is a product of the IPC speed (instructions per cycle) times the clock frequency.

IPC speeds are dependent on the microarchitecture. At this point with superscalar, we can get 1.5. In the next 10 years, we'll need to raise it to about 4 to match the trend line. As I look at this trend line and plot it, I see we would have to go to 12.5. Architects have a responsibility to address this question.

D&T: How should they approach it?

Lu: There are two ways to look at it: from a microarchitecture point of view and by looking at timing methodology. I have come up with terms or acronyms for timing: completely synchronous, which we're doing now; totally synchronous; (TS); locally synchronous (LS); and completely asynchronous. We should go for completely asynchronous. There are actually other ways of doing timing: globally, which I call GALA, locally asynchronous, GALS (globally asynchronous/locally synchronous), and LAGS (locally asynchronous/globally synchronous). Each have advantages and disadvantages. People from the asynchronous community have said that from the power point of view, asynchronous is the best because of the clock power. But that remains to be seen.

Partovi: Conditional clocking and handshaking are somewhat similar. Conditional clocking keeps certain processor blocks in the sleep mode when nothing is happening; we don't access them when they're not needed. That definitely reduces power, though it has some performance degradation.

D&T: What about the extra overhead to handle clock gating?

Sakurai: People have not addressed this question formally and have not come up with a definite answer as to which is better.

D&T: Asynchronous communication helps address the clock distribution problem in global communication. How does this affect the neighborhood issue, which is related to the interconnect problem?

Lu: We've been working on a microarchitecture called Compuflow. It was proposed by Sutherland's group, published in *Design & Test* in 1994, and came from the asynchronous community. The Compuflow idea is basically to have only local communications. Besides clocks, there are other signals in the pipeline that are global; for example, we need global signals to tell every stage to stall a pipeline. Removing those signals may help the clocking a little.

Sakurai: To reduce bus power, we can lower the signal voltage swing to 200 mV or less, using a differential pair line. Since the power is proportional to the signal swing, it's an important technique. We can also use the small swing clock to reduce the clock power, using a special flip-flop.

Partovi: We have copper and low-K interconnects coming up. Should we design them for low capacitance or low resistance? Obviously, we'd like low resistance for high performance and low capacitance for lower power. Either can be achieved. There might be a common ground where we can use both, or at different layers we can have lower capacitance or lower resistance.

Sakurai: We can reduce interconnection power by using a system-on-a-chip approach. This approach embeds several modules on one chip, reducing the interconnection capacitance among modules. If we use external DRAMs, we use about 1 W for processor-DRAM communication for 1-Gbyte/s memory bandwidth. With embedded DRAMs, we get the same bandwidth by two orders of magnitude less interconnection power. Since PCs are dependent on using caches, main memory, and mass storage, we can also reduce communication for distant modules to use less interconnection power. In the longer term, a globally asynchronous/locally synchronous system is preferable.

Partovi: Using low-swing buses inside a chip could enhance speed and power. However, the logical partitioning should be such that the buses are received at the end of the cycle so they can be sampled by the sense-amplifier flip-flops. Also, it appears to me that open-drain I/Os have the best features for the lowest power.

D&T: How does conditional clocking apply to a desktop server/processor scenario? Will it provide any help?

Partovi: Power is basically a thermal issue for high performance that we can't get around. DEC designers believe they should power down wherever or whenever power isn't needed, but doing so creates problems with race. Whenever we have boundaries on clocks, we have to worry about extra design effort. Time to market may increase because we have

to solve many race problems.

Also when a chip comes out of quiet mode, many things suddenly turn on, causing a power supply droop that could result in loss of state in memory nodes or system slowdown.

Sakurai: Since some chip parts don't require very high performance at a time, we can reduce V_{DD} to slow some parts when we don't need full speed. Only software programmers know when it's possible, so why not add some power control instruction set to existing instructions and give some of the duties for achieving low power to the software engineer?

Soumyanath: It's not even clear that clock gating saves power. If done incorrectly, we can actually burn power mainly because we are turning more capacitors on line. If the clock-gating is at a fine-enough level already, we may lose rather than win.

To amplify on Hamid's comment, the high power demand coming out of sleep mode will put a huge pressure on the power delivery people. Think of just trying to supply, say, 70 amps at 700 mV. We also want to deliver it at 900 mV, so this is pretty close to an impossible problem. We ought to think about both level solutions to this problem. The power delivery is less stringent, and we have other manufacturers who are motivated to help us.

D&T: How much thermal design power does it relax for the desktop? Someone could write a virus that would keep everything active all the time and cause the microprocessor to heat up. Then clock gating does not help. The power lost by turning the clock on and off must be much less than the amount saved by keeping it off for a reasonable period of time. How do we predict that some unit on the chip would not be required for a number of cycles?

Partovi: You're right; there could be a particular application in which we don't know when everything will be running, although, we do know in other modes like stop clock or sleep mode. I have to stress that designers may be able to guarantee operation exclusivity of some functional blocks.

Lu: A compiler or software can do it. From the architectural point of view, there's no way to know when the unit will turn off; it's best to leave it on.

Soumyanath: Some type of powerful, statistically weighted set of benchmarks like the common performance benchmarks would really help. Architects agree with benchmarks, and, more importantly, the sales force agrees with them. Then we could use the benchmarks to determine when or when not to do the clock gating and which unit. I don't know how we would know in real time, but the statistics would help us zero in on what units we want to really focus on.

Partovi: There's a possibility of using an on-chip temperature sensor to handle thermal issues. In other words, if the chip heats up, a circuit, or the operating system, or whatever will say "shut off." So we stop the clock and wait until the chip has cooled down. We need to have a grasp of the die's temperature in case things start running away.

D&T: We'd need a thermal device of some kind, something similar in software, a sensor, or a signal. Let's turn now to your thoughts on the technology evolution.

Chuang: Most people believe bulk CMOS is running out of steam, and we should move to SOI design. But SOI is probably not as straightforward as people think. Also, the design issues for low power and high performance are completely different in SOI. For example, if we use SOI for the low power and portable types of applications, we'd have much less demand on performance and scalability. We wouldn't use the most aggressive technology and wouldn't worry about scalability because power is the issue.

We'd choose fully depleted SOI because it is easier to design. Since the transistor count would also be low, manufacturability would not be a concern. There would be much less demand on design methodology and resources. Typically, in a low-voltage application, reliability isn't an issue. It's also much easier and very effective to put in a body contact because the frequency is low, but using SOI for high performance in a microprocessor is completely different. There is a very strong demand on performance and scalability. There are tens of millions of transistors. We'd have no choice but to use partially depleted SOI because it provides much better scalability, much better manufacturability.

D&T: What trade-offs are involved?

Chuang: In partially depleted SOI, designers have to be very selective about placement of the body contact. They also have to worry about operation frequency, and the body contact better have lower RC to be effective. It's really important that designers understand the device's behavior and the circuit topology.

D&T: There is a lot of difficulty in designing in SOI. What are the advantages compared to a bulk design?

Chuang: Everyone in the industry keeps asking that question. The main issues that determine the leverage over bulk technology actually spread across the entire spectrum. At the lowest level, we have to start with the right process and device design. The device must have a reasonably high V_T

while containing the leakage. If the V_T is too high, all the SOI advantage will be lost.

There are many other factors. The design strategy also determines the leverage. Suppose you are a microprocessor program manager and you tell your team that you want a design that works on both SOI and the bulk product. With this kind of design strategy, you are going to hurt the performance of both.

In the bulk SOI version, designers must reserve the area for selective body contact. But degradation to the bulk version is at most a small percentage of area. The SOI version is going to hurt tremendously. That version won't have an adequate amount of decoupling capacitors. In today's microprocessor, we typically want on-chip, dedicated decoupling capacitors in the range of 200 nF or above. SOI gets rid of bad and also good capacitance because about 30 to 40 percent of the decoupling capacitors are supplied by the nonswitching, built-in capacitance coming from the diffusion-to-well and well-to-substrate capacitance.

Partovi: That is an excellent point.

Chuang: The SOI version is going to see a tremendous power supply bounce that degrades performance. Timing is another factor. Some circuits will run much faster when you go to SOI. If designers optimize timing for bulk, the long path in the bulk version is not necessarily the long path in SOI. So the design isn't optimized.

D&T: Are you saying that we cannot design for both bulk and SOI?

Chuang: Yes. If anyone thinks they can get the best from both, they're wrong. The timing issue is even more serious. You have nonuniform speedup. In microprocessor design, people worry about the short path much more than the long path. With a short path, designers have to pad very carefully. In SOI, the short path can be substantially shorter because of the floating bodies. Another consideration is the wire delay—it will cost you in the dual-design strategy. In the SOI version, designers cannot play with a dynamic V_T control driver. Then the wire delay is going to kill performance.

D&T: How can we get more out of SOI?

Chuang: If you're a design coach, most likely you're not going to try for SOI-only design, though if you go for it, the performance will likely increase. You're going to have high density, your SRAM is going to shrink, and perhaps you can put in more decoupling capacitors right at the beginning of the design. Your timing is going to be optimized for SOI. You can choose the circuits that have particularly high leverage on SOI, and selectively use dynamic V_T control in very long line drivers (not globally in the logic; that's very difficult to do). This cuts down the wire delay. So when you do this, you are going to get much more out of SOI.

D&T: What are the pros and cons for integrating a DRAM and microprocessor on one chip?

Partovi: It may be prohibitive from a technology standpoint. People seem to agree that DRAM technology doesn't fit well with microprocessor technology.

D&T: If we assume it's feasible technologically, is there a good reason to put DRAM and logic on the same chip for microprocessors?

Partovi: For the general desktop PC, an embedded DRAM is less important, but there are applications in which it makes sense. Using embedded DRAMs in mobile systems and CPUs that are suitable for palmtops or PDAs is reasonable.

Lu: From an architectural point of view, embedded DRAMs help us avoid hitting the well-known memory wall. Eventually there will be a point that, because the gap between DRAM access time and processor sequencing has been growing every year, no matter how fast a processor is, an application is bounded by memory. For every four or five instructions, there's a memory reference. I agree that the embedded DRAM's usefulness depends on the application; for general purposes, it will not be that useful.

Sakurai: Since the external memory reference is two or even three orders of magnitude higher than multiplication and other basic data processing, reducing that reference is effective for low power. Adopting reconfigurable elements such as programmed FPGAs might eliminate the need for supplying instructions in each clock and cut power.

Partovi: Would you suggest, then, that for low-power systems, caching should not be required? Is it possible to go directly to an embedded DRAM? We can use caches and access parts of it. That would save power with a slight performance degradation. To enhance performance, we can use set associativity along with banking, though it might reduce clock frequency.

Sakurai: The main objective of introducing caches or using local memory is to enhance performance, so that caching is required for processors.

Partovi: Caching is effective, of course, to reduce communication, as it exploits temporal and spatial locality of data.

Chuang: The IEEE has held workshops on this topic; one of the issues is whether to use either logic- or DRAM-based technology. There is no argument; we have to use logic-based technology. The DRAM typically has thicker oxide in the peripheral circuits due to the boosted wordline voltage. In a logic-based technology, the peripheral circuit is going to be three to five times faster, and there's a chance we may be able to use it as cache.

Soumyanath: I have to question whether we actually save power by putting a DRAM on a chip. We'd have I/O drivers that drive the thing on the board. My guess is that this would be much less than 5% of the chip power.

D&T: Can we live with leakage, or should we stop oxide scaling?

Partovi: We cannot. As long as we continue device scaling, as long as we make channels shorter, we must scale the oxide or we lose control over the channel altogether.

Sakurai: Oxide tunneling leakage and subthreshold leakage are similar in that they drain all the time. For circuits, we might control oxide leakage by applying some of the selective cutoff mechanisms that have been effective in controlling subthreshold leakage.

Chuang: For the foreseeable future, let's reduce demands on leakage and allow more leakage current.

D&T: Yesterday I heard that we should just build design around leakage! What is the effect on soft error when we lower the supply voltage?

Partovi: Some people have been designing radiation-hard memories for a long time. The basic concept is that we build delay in the keeper feedback path, hence allowing for state recovery of a node disturbed by alpha particles.

Soumyanath: Right now, we take the node, invert it, and put it in a different channel that keeps it. So we have to use radiation-hard techniques, and we get delay. We are going to do all kinds of different designs.

Sakurai: The amount of charge that's being held on the nodes is going to be extremely small. We're not paying enough attention to this problem. It's easier to use memories to implement soft-error resilience; flip-flops are harder. They are distributed over a chip, and the area overhead of the rad-hard flip-flop is big. Some system-level countermeasures such as data correction will be more meaningful.

About the participants

Ching-Te Chuang is a manager of High-Performance Circuit Design at the IBM T.J. Watson Research Center, Yorktown Heights, New York. He is a Fellow of the IEEE.

Vivek De, our moderator, is a principal engineer and manager of Low Power Circuit Technology at Intel's Micro-Computer Research Labs in Hillsboro, Oregon.

Shih-Lien Lu is an associate professor of electrical and computer engineering at Oregon State University in Corvallis.

Krishnamurthy Soumyanath manages high-performance circuit research at Intel's Circuit Design Research Lab in Hillsboro, Oregon.

Hamid Partovi is an AMD Fellow at AMD's California Microprocessor Division in Sunnyvale, where he is currently the cocircuit design lead of the K7 microprocessor.

Takayasu Sakurai, a professor at the University of Tokyo's Center for Collaborative Research, works on all aspects of low-power, high-performance VLSI designs.

Soumyanath: The real problem is in mobile applications; people on an airplane don't want their laptop to crash. We can't always blame it on the software vendors for writing bad codes. We'll get a bad reputation as an industry if we have unreliable things that don't work very well at high altitudes.

Chuang: You are starting to sound as if we need to have structures like stack capacitors, just like DRAMs, so that we have higher capacitance at certain nodes.

Soumyanath: Indeed, this one might happen because we are running out of capacitance. Circuit engineers and process people have to work together to solve this.

Partovi: One of the things I should bring up again is CD control of very short channel devices. This is going to be a very serious problem, because now we have to do statistical analysis on our designs.

D&T: Thanks to each of you for participating so fully in this discussion. We still have many challenges to look forward to.

[Don't miss the deep-submicron noise roundtable in the October-December 1998 issue of IEEE Design & Test.*—Ed.]*