

Gene Frantz, Jörg Henkel,
Jan Rabaey, Todd Schneider,
Marilyn Wolf, and Umit Batur

Ultra-Low Power Signal Processing

This *IEEE Signal Processing Magazine (SPM)* forum discusses the latest advances and challenges in ultra-low power (ULP) signal processing (SP). The forum members bring their expert insights to issues such as design requirements and future applications of ULP SP systems. The invited forum members are Gene Frantz (Texas Instruments), Jörg Henkel (Karlsruhe Institute of Technology), Jan Rabaey (University of California at Berkeley), Todd Schneider (ON Semiconductor), and Marilyn Wolf (Georgia Institute of Technology). The moderator of the forum is Umit Batur (Texas Instruments). Our readers may agree or disagree with the ideas discussed next. In either case, we invite you to share your comments with us by e-mailing batur@ti.com or spm.columns.forums@gmail.com.

Moderator: Let's first start by defining our topic. What does "ULP SP" mean? What specific requirements should a signal processing system satisfy to be called an "ULP" system?

G. Frantz: This is a fun question as it means different things to different people. I wrote an internal white paper on this topic a couple of years ago so we could, as a company, minimize the confusion on what it means. Here is an excerpt from the paper:

The easy way to differentiate ULP from other power aware concepts is to create a chart on the priority order of the basic aspects: performance, price and power dissipation. Here is a description of my version of that chart.

- *High-performance* devices are those where performance is the

primary priority, if not the only priority.

- *Low-power* devices are those where, given no performance is sacrificed, the focus is on how much the power dissipation can be reduced.

- *ULP* devices are those where, given the absolute minimal power dissipation is achieved, how much performance is left? Then the focus is on performance.

So, I see ULP as a design philosophy rather than a specification.

J. Henkel: An ULP signal processing system should address low power consumption at all levels of abstraction i.e., it should apply the latest power efficient silicon technology to start with, it should utilize the most advanced power management techniques at OS-level, and, don't forget, the application itself should be trimmed to low power consumption. Especially algorithmic transformations at the application level can be very power efficient in signal processing systems. In summary, exploiting the potentials in power savings at each (or at least many) level makes to my mind a real ULP system. Therefore, designing an ULP signal processing system is a combined hardware/software problem.

G. Frantz: All aspects of a system should be involved: process, transistor, gate, hardware architecture, software, and system.

T. Schneider: We, at ON Semiconductor, design single-chip digital SP (DSP) systems that are almost always battery powered, so for us "ULP signal processing" means meeting demanding battery lifetime constraints while simultaneously accomplishing a demanding signal processing task. Thus, the prime metric is really one of efficiency: in a given, real world application, minimize the power

consumed by the DSP system for a given task. We use $mW/MIPS$, where MIPS are the actual instructions being executed for application and mW is the measured power consumption of the DSP system.

Specific applications have constraints driven by desired battery lifetimes and batteries that are used. As an example, a power-consumption constraint of 1 mW or less is common for hearing aids. Designers strive to get as much signal processing and the best overall audio quality they can within this power budget. Within this power budget, developers are expected to implement features like adaptive feedback reduction, dynamic range compression, noise reduction, and parameter selection based on environmental monitoring. For some smaller hearing aid applications, this power constraint can drop to as low as 500 μW .

The extreme end of the ULP signal processing we address is signal processing in implantable systems such as pacemakers, implantable defibrillators, cochlear implants, and neurostimulators. Typical functions included in an implantable device are low-pass filtering, level measurement, and threshold detection. Some implantable devices are designed for battery lifetimes of ten years with an average power consumption of less than 30 μW , including the therapy. This corresponds to a current of 10 μA from a typical battery. With the current required to deliver therapy, only a few μA average current remains for all of the electronics. Of course, static current is also key in implantable applications and minimizing this requires a tradeoff to be made between threshold voltage, supply voltage, and operating frequency.

To achieve the lowest possible dynamic power consumption, we strive

to operate the signal processor(s) at the lowest possible operating voltage. This typically means minimizing the clock rate, which necessitates application driven design.

In our experience, the largest power savings can be derived from clever design of the signal processing algorithm(s) and from clever design of a system architecture onto which this algorithm can be efficiently mapped. In the late 1990s, we pioneered a novel approach that combines a flexible, software-programmable DSP with a more fixed function, micro-coded “accelerator.” This architecture recognizes that many DSP algorithms consist of a “vector number crunching” portion and a “control path/side-chain” portion. By partitioning an algorithm into these two portions with an efficient mapping, clock rates can be minimized. This allows the supply voltage to be minimized, which reduces power consumption. Properly executed, this approach provides an approximately linear reduction in power consumption. This approach makes a tradeoff between flexibility and power consumption, which strikes a compromise between “hard code everything in logic” (lowest possible power with zero flexibility) and “make everything programmable” (highest power and most flexibility). To fully realize the benefits of this approach, the two processing units must be run in parallel, which can increase programming complexity. Of course, there are applications (like implantables) where the power constraints are so stringent that there is no choice but to implement signal processing as fixed-function blocks.

In summary, for us, ULP signal processing means a focus on efficient algorithms and application-driven architectures both seeking to minimize the clock rate and operating voltage and thus realizing the lowest *mW/MIPs* in the intended application.

J. Rabaey: I tend to take a somewhat different tack, and tend to classify systems by their energy source or provision.

- *ULP systems:* self-contained—these systems either live off a single battery charge for the lifetime of the product, or scavenge energy

- *Low-power systems:* battery operated—need occasional battery replacement or recharge

- *Powered systems:* Performance is dominant. Energy-efficiency is important for either heat management or for cost reduction.

The actual boundaries between these different classes are variable depend upon the size of the node and the usage patterns. Observe that this definition is generic and extends beyond the signal processing label.

M. Wolf: I like Jan’s definition, but what about devices like radio-frequency identification (RFID) that receive energy from an antenna?

J. Rabaey: Devices like RFID that receive energy from an antenna fall clearly under the ULP class. They are “perpetual” and harvest electromagnetic or magnetic energy.

G. Frantz: I think there is a whole family of products of which I call “perpetual devices” that beg the same question. But I think Jan covers it in his definition. What I read his definition to say is the “power source lasts longer than the useful life of the product.”

J. Rabaey: I am perfectly in line with Gene on this one.

T. Schneider: We have an internal definition (more of a joke really) that we call “infinite battery life” that is along the same line as Gene’s comment above. If the device has replaceable batteries and the user cannot remember the last time they replaced the batteries then the device has, in effect, infinite battery life.

Reading these definitions, which are more general than mine, I realize that many of the applications we address have a power source that is predetermined by size, reliability, availability, or the fact that it has been historically used. This is a hard constraint and ULP also means cramming as much processing capability in as possible while living within this constraint.

G. Frantz: That is why I call it a philosophy rather than a specification. But, back to Jan’s “infinite battery life” rather than my “perpetual device,” the problem with mine is you can’t patent anything that starts with the word “perpetual.”

Moderator: Which signal processing applications are driving the need for ULP? What are the most important tradeoffs in these ULP applications?

J. Rabaey: This is a tough one to answer, as there are many options. My top choice would be medical applications. Various wireless monitoring and implanted devices are pushing the limits on what ULP design is all about. In the future, I can see various immersed media devices becoming crucial as well. How about virtual reality on a mobile?

J. Henkel: I agree with Jan’s points of medical and wireless but want to pick up his last point and extend it a bit to mobile multimedia in general. In a research project we are currently conducting, we are exploring the low-power (I hesitate to call it ULP) design space of a mobile multimedia device with respect to impacts on both the hardware architecture and the software architecture. So far, we ended up with a new hardware architecture that uses dynamic reconfiguration. On the other side, we had to heavily redesign the software architecture (in that case, an H.264) to exploit the hardware architecture’s low power capabilities by, for instance, exploiting a high inherent degree of instruction-level parallelism. Coming back to the question on the tradeoffs, in our case the price to pay is to completely redesign the software architecture and to develop a novel hardware architecture.

M. Wolf: I think that sensor networks/cyber-physical systems are an important category. For example, people don’t want to have to crawl through a building or bridge every year to replace the batteries in the sensors, which is what they have to do right now.

G. Frantz: So, to follow the thread above, Jan’s definition (or my rewording) of ULP describes what markets it will be used. Those markets for which the batteries should last longer than the useful life of the product.

I was talking to a friend of mine in the world of implants. I suggested that implantables needed to be perpetual as they needed to last longer than the life of the individual. He said that it wasn’t true, implants only needed to last ten years; not for the life of the human but

to outlast the obsolescence of the technology. His point was that, independent of the battery life, technology needed to be replaced at least every ten years, if not sooner.

Marilyn's example of changing batteries in a bridge monitoring application gives a different perspective of the market. I argue there is another market where the product has been buried or placed in an unreachable location. For example, load cells at the bottom of structural columns, under highways, or in nuclear plants.

But to keep it simple, when I am asked the question about what market/product can use ULP, I explain that I don't know. It's kind of like fishing in a new hole: you don't know what bait to use and you don't know what you're going to catch, but it's exciting. I believe ULP will address new markets we have never thought of and that is what makes it exciting.

Now to discuss compromises: To guarantee the lowest possible power dissipation, one must tradeoff price and performance. Which begs the questions "what value does ULP bring to the party?" "Will people pay for it?" "How much performance will I need to eliminate?"

J. Henkel: I disagree that ULP is necessarily a question of the price one is willing to pay. You can get ULP devices quite cheap. An example: a few years back I talked to the developers of Texas Instruments' MSP430. Embedded in a system it can run for ten to 15 years with one battery i.e., without any recharge! This, of course, depends on the application and assumes that the processor only wakes up from time to time to, for instance, gather some data and to do some simple calculations and finally storing some of the data. A simple system built with this (or a comparable) processor is quite cheap, but according to some previous definitions in this discussion, it is an ULP design. Anyway, I agree that one needs to tradeoff performance when ULP is required.

G. Frantz: I understand questioning price. But my experience has been if you choose to make one variable fixed you need to tradeoff the other two. In this

case, we choose to not compromise power dissipation, we will find we either need to reduce performance at the same price or increase price to increase performance. I get this from the idea that what Moore's law will continue to give us is more transistors. We can either increase performance or decrease power dissipation by adding more transistors.

Now having said that, Jörg's prime example of where that isn't the case is a good one. And there are many more. So, I'll be glad to not push it any further but ask a more interesting question: does ULP have perceived value in a product? Can a premium be placed on a product with a longer battery life?

T. Schneider: I agree with Jan's point about medical, especially implantable and body worn (e.g., hearing aids), applications. However, in applications that use wireless, the power savings gained

**FUTURE ULP SYSTEMS
NEED TO BE ABLE TO
FLEXIBLY ADAPT AT RUN
TIME TO ALWAYS
(OR MOST OF THE TIME)
RUN AT MINIMUM
POSSIBLE POWER.**

through ULP signal processing can sometimes be overshadowed by the power consumption of the wireless subsystem. Still, short range wireless applications like body sensor networks are likely to be a future application where ULP signal processing will make a difference. This could evolve into a more general category of wearable or (semi) implantable electronics for both medical and recreational purposes.

In the multimedia area, some portable, body-worn audio designs (e.g., headsets and mobile phones) require ULP signal processing to make the design viable. In these cases, significant DSP resources are devoted to echo cancellation and noise reduction to compensate for the mechanical aspects of the device (e.g., a small form factor that locates the speaker close to a microphone or a design that relies on DSP to compensate for other mechanical or acoustic

features). Pushing more and more functionality into smaller and smaller devices and having them used in a wide range of environments is likely to place higher demands on ULP signal processing in these types of applications.

The most significant tradeoff in my experience is flexibility (and therefore programmability). Reduced programmability generally means less memory is used, which means smaller die. This drives lower cost at volume.

Jörg's point about dynamically reconfiguring hardware is an interesting one. This is one way to gain some flexibility in a power-efficient way without resorting to a fully software programming system.

In many ULP signal processing applications, performance tradeoffs are also made. These applications will often have "just enough performance." There is little or no "headroom" for new features or product expansion.

J. Rabaey: Just to address some of Todd's comments: It is indeed true that the wireless communication overshadows the signal processing budget. However, signal processing can help substantially to reduce the wireless communication power either by reducing the amount of data to be transmitted, or by making the RX/TX more efficient (in short distance wireless, the RX/TX power is way larger than the communication power).

With respect to the programmability aspect, I fully agree. Fully programmable solutions and ULP do not go together (unless you want to run extremely slow, such as the Michigan subthreshold processor). A combination of accelerators, reconfigurable/parameterizable modules and a rarely used simple core is the best option.

T. Schneider: Jan's point regarding the power reduction via ULP signal processing for wireless links is good and I completely agree. I was thinking more of standards based approaches like LE Bluetooth and Zigbee, but these can hardly be classified as ULP.

G. Frantz: We took a chart from Berkeley several years ago and added one point. The chart showed the amount of energy it took to transmit 1-b of information through various ways. We added

a point (as I remember) to the chart. That point was how much energy it took to execute one instruction. The thought behind it was that it should be lower power to compress a bit then to transmit a bit. As I remember, the cost of compressing a bit was several orders of magnitude less than to transmit a bit. We did this in 2002, so the data is old. But I think the concept is still there. What I have tried to do is to change the idea from analog to digital (A/D) to analog to information (A/I). As we compress (convert) data into information, it will always be cheaper to transmit the information rather than the data. For example, in a security system at my home, the system doesn't need to transmit a picture of me as I walk through the house, it just needs to send a minimum set of information saying what I did.

Moderator: What are the most important ULP signal processing design techniques used today in semiconductor device fabrication, and processor, system, and software design?

M. Wolf: If one is designing an ULP system, one arguably doesn't want any software on the ULP device itself.

G. Frantz: On the contrary, the only way it will work in the IC world we are creating is for it to be programmable, or at least configurable, to be viable. I would agree, given today's memory technology, you are correct. So I would change your words to "how do we make memory ULP and nonvolatile." In fact, there are some technologies that might make this possible in the research labs now. One example is the FERAM.

So, let me go back to my first comment on the IC world. At each node we are making it more difficult to design and tool a new device. Once done, it is virtually free to manufacture devices. Jeff Bier has predicted that companies like Texas Instruments would only be able to create a handful of digital chips per year due to the prohibitive cost of design and tooling. That will force us to a few devices that can serve thousands of markets. My conclusion is that this results in programmable platforms on which innovation takes place.

I will make a bold statement and say that "ULP devices are not the innovations. They are the platforms on which innovation will happen to create a whole new world of products."

With that behind, the areas that most need to be dealt with for success in the product are: zero power memories, ULP wireless communications, energy scavenging, and analog.

M. Wolf: Memory isn't the only power hog in a programmable system—there's the I-box and the datapath. Of course, programmability doesn't necessarily mean Turing machine...

G. Frantz: Very good points. I also swept by, far too quickly, the concept of programmability. There is a spectrum of programmability from fixed function to infinitely programmable. Somewhere between these two extremes we find concepts like configurable and accelerator. Even a fixed function processor has two instructions (on and off), or should have. A lot of fun to be had by all.

J. Rabaey: I tend to agree with Gene that for ULP signal processing to be successful, a "reusable" platform needs to be created. The ad hoc approach does not fly. This means that we need to start thinking design methodology, libraries, reuse strategies, and common concepts. Innovation is cool for a while, but does not lead to mass impact. This is where the true challenge is. Gene's list is very good—I would add some: mechanical computing, passive computing, and bio-inspired processing (just to make it more exciting).

M. Wolf: Reusability can come from a combination of the host and sensor sides. Fancy signal processing, for example, can be performed at the host after basic data reduction is done at the sensor.

T. Schneider: Very interesting answers so far... the scope has broadened to include economic considerations as well as methodology and system aspects.

I agree with Jan that the design approach is key. I also agree with the points made by Gene regarding the economic considerations. These together imply that one of the most important aspects is the partitioning and code-

sign of the system. What portions of the device get "hard coded" permanently in hardware (for the lowest possible power)? Which portions of the device will retain some flexibility, through free programmability on a processor or via lower-power options like microcoding an accelerator in a ROM? Wise choices will lead to a system that will find broad application (that as Gene says "will serve thousands of markets"); whereas poor choices will result in a system that will not become a widely used platform.

An additional area that I see as important is software/configuration tools that provide an efficient and effective means of programming deeply embedded parallel systems in the absence of an operating system (in my opinion, an OS and ULP just don't go together). Ideally, these tools must work with a multiprocessor, heterogeneous system, and provide close to the same efficiency as one could achieve with hand coding. This is a challenge because both partitioning and scheduling across multiple compute units must be addressed. Without tools like this, building complex ULP systems will remain a challenge.

Moderator: It seems that hardware/software partitioning and codesign present important tradeoffs when building ULP systems. How do you expect ULP design techniques to evolve in the near future to address these trade-offs more effectively?

J. Henkel: I agree that HW/SW partitioning is a crucial decision to make when it comes to ULP. This decision is done at design time. I want to go a step further and claim that future ULP systems need to be able to flexibly adapt at run time to always (or most of the time) run at minimum possible power. There are scenarios that simply cannot be predicted at design time/compile time and if the system would not be able to adapt appropriately, it would give away some of the potentials in power savings. So, my statement is: future design techniques for ULP systems need (more than nowadays) to provide the ULP system with the capability of more run-time flexibility (even though classic design paradigms teach us that flexibility costs power consumption).

G. Frantz: Good point, a significant tradeoff will be flexibility versus power consumption. But let me replace flexibility with a different concept: time to market. I tell people that my customer wants four things from a vendor; a solution that has good enough performance, low enough cost, low enough power dissipation, and the ability to get them first to market.

In fact, forget the first three, “help me get to market first and give me a roadmap to cheap.” Note that I said “roadmap to cheap” rather than “roadmap to ULP.” At the end of the day, the system designer need not have ULP, but low enough power to lead the market. Flexibility is the key to leading a market.

T. Schneider: Perhaps I am being too optimistic, but I believe that with all of the activity we see in mainstream multiprocessor systems, it is only a matter of time before ULP design techniques evolve to support high-level assessment of different multiprocessor/computational unit/reconfigurable accelerator design concepts. These tools will allow assessment of the tradeoffs that are made between communications/data transfer overhead and the power consumption reductions realized through parallelization.

To extend Gene’s point, ULP done properly is lower cost because an efficient design will typically have fewer gates, use less memory, and is therefore likely to consume less power and less silicon area. Flexibility is good, but optimizing the approach for very high volume ULP applications is the key for realizing the cost targets that are typically required.

Moderator: How mature are the ULP design tools that are needed to efficiently and effectively explore all tradeoffs we mentioned so far?

G. Frantz: I think this is a simple answer: They don’t yet exist. But in case I am wrong, can we list the ones we need and the ones we already have?

J. Rabaey: Dismal is the right answer. There is progress in modeling and simulation (including statistical analysis). Logic synthesis tools can be adapted to serve for instance subthreshold logic

without to much of a challenge. The rest is mostly spreadsheet. We really need good exploration tools.

T. Schneider: I agree on all points. I’ve seen some tools (and been involved in a research project) that studied exploration tools. They worked, but the models used for power consumption were too simplistic to make them useful in real applications. For now, experienced systems/chip designers, a good methodology for exploring design concepts, and spreadsheets are the best we’ve got.

G. Frantz: Let me give a real example of the state of our power evaluation tools in IC design. We just introduced (in the last month or so) a new device that was designed specifically for low power. Once we got silicon and tested it we found that the silicon’s power dissipation was off by a factor of two from the design tool estimate. Fortunately it was off by two in the

ULP DONE PROPERLY IS LOWER COST BECAUSE AN EFFICIENT DESIGN WILL TYPICALLY HAVE FEWER GATES, USE LESS MEMORY, AND IS THEREFORE LIKELY TO CONSUME LESS POWER AND LESS SILICON AREA.

right direction. That means we have a lower power part than we expected. Although this sounds like great news, had we known that earlier we might have changed how we marketed it and given our system customers a greater head start on taking advantage of the lower power.

Moderator: What are the implications of analog versus digital implementations of SP algorithms in ULP designs?

G. Frantz: I don’t know that analog versus digital changes much at a high level. But it does once under the hood. ULP digital will get a great portion of its advancing by way of lowering the operating voltage, running slower, and using parallel concepts. Analog will not have as much of an advantage with lower voltage but will get its advantage by using analog

concepts to do the intensive math operations. But the old thought that we would be able to virtually eliminate analog and do everything in digital is dying away and we now have the advantage of making tradeoffs between digital and analog implementation for SP.

As an aside, we have seen digital concepts used to enhance the analog to digital conversion accuracy. In the same way we will begin to see analog concepts used to enhance the performance of digital circuits. This will be relatively independent of whether the task is to increase performance or lower power dissipation. Of course, this leads to the question of what and how do we prepare students for this new world of ULP SP.

J. Rabaey: I essentially agree with Gene. It seems that for a number of SP functions (such as spectrum analysis), analog, and “mechanical” implementations can be more efficient than digital as long as the required accuracy is low (the number that I have often seen is around 6-b of accuracy as the transition point). When you need higher resolution/accuracy, you need to go digital. We have gradually migrated towards believing that digital is always the better way, but that is simply not true. Analog can often be very efficient, but don’t ask accuracy.

Passive solutions don’t take any energy at all!

Now here is where the opportunity lays, as Gene has perfectly pointed out: a combination of low-accuracy analog with higher resolution providing digital. The latter can still scale for a bit in terms of energy efficiency. The former benefits from the complex functionality that can be realized with few devices. And yes, this begs the next question.

T. Schneider: I agree with the points made above. In my experience, if you can “tolerate” an analog solution (pun intended) it will generally be lower power. Analog circuits also have the advantage of offering lower input-output delay and this can make them preferred (even required) in some application. Analog solutions may also bring in an additional degree of freedom. In some applications you can tradeoff size for reduced noise.

J. Henkel: I agree with most that has been said about analog. At the same time I want to point out that analog techniques for low power are by far not as advanced as digital ones. For example, transistor sizing in digital is highly optimized for low power. Techniques for optimizing sizing for analog transistors is often rather ad hoc and hence less power efficient. However, the whole picture in a DSP system might be more complex. Let's assume that an analog signal is processed without the need to convert it to digital. This saves the power the ADC/DAC consume and might compensate for the effect I pointed out above.

Moderator: Let's now look at ULP from an algorithm designer's point of view. What should SP algorithm designers focus on to enable ULP implementations of their algorithms?

J. Henkel: One important decision an algorithmic designer has to make is to determine the numerical representation of data in a DSP application i.e., to choose fixed point or floating point representation. This decision will heavily impact the power consumption since not only the fixed point and floating point calculations differ widely in power consumption but also the power related to storing/moving the respective data in/to/from memory can differ by many x. The ULP algorithmic designer should therefore consciously and carefully weigh the use of floating point versus fixed point.

T. Schneider: Algorithm designers should focus on clever optimizations that retain (or alternatively tradeoff as little as possible) algorithm performance, while minimizing the computation load (and hence the clock speed). In our experience, the biggest gains are realized by efficient SP algorithms. Doing this allows operation at reduced voltage that will deliver the reduced power. In many situations, partitioning an algorithm into elements that can be run in parallel across multiple computational units is an effective way to reduce power consumption, provided communication overhead does not consume the power

reduction that can be gained via this approach. If the parallel computation units can be made reconfigurable or hard coded (in hardware), additional efficiencies can also be gained through more efficient utilization of hardware resources. I would also argue that in the vast majority of applications, floating-point and ULP don't mix. If you want the lowest possible power, you require a deep understanding of the algorithm and the minimum precision required to realize the required levels of performance.

PANELISTS

Gene Frantz (genf@ti.com) is the principal fellow at Texas Instruments and has been with Texas Instruments for more than 30 years. He holds 40 patents in memories, speech, consumer products, and DSP. He has written more than 50 papers and articles. He is an IEEE Fellow.

Jörg Henkel (henkel@kit.edu) is the chair for Embedded Systems at Karlsruhe Institute of Technology, Germany. He was previously with NEC Laboratories in Princeton, New Jersey. His current research is focused on design and architectures for embedded systems with focus on low power and reliability. He received the 2008 DATE Best Paper Award and the 2009 IEEE/ACM William J. McCalla ICCAD Best Paper Award. He is the chair of the IEEE Computer Society, Germany Section, and the editor-in-chief of *ACM Transactions on Embedded Computing Systems (ACM TECS)*. He is the coordinator and a founder of the German national research focal program "Design and Architecture for Dependable Embedded Systems." He holds ten U.S. patents.

Jan Rabaey (jan@eecs.berkeley.edu) is the Donald O. Pederson Distinguished Professor in the Electrical Engineering and Computer Science Department at the University of California, Berkeley. He is the scientific codirector of the Berkeley Wireless Research Center, as well as the director of the GigaScale Systems Research Center. He received numerous scientific awards, including

the 1985 IEEE Transactions on Computer Aided Design Best Paper Award (Circuits and Systems Society), the 1989 Presidential Young Investigator Award, the 1994 Signal Processing Society Senior Award, and the 2002 ISSCC Jack Raper Award. He serves on the technical advisory board of a wide range of companies. He is an IEEE Fellow.

Todd Schneider (Todd.Schneider@onsemi.com) is the vice president of the Medical Diagnostics, Monitoring and Therapy product line at ON Semiconductor in Waterloo, Ontario, Canada. He holds BSc. and a MSc. degrees from the University of Waterloo, with a specialization in DSP. In 1998, he cofounded Dspfactory Ltd. He has published numerous refereed papers and conference proceedings. He also holds a number of U.S. and international patents in the field of DSP.

Marilyn Wolf (marilyn.wolf@ece.gatech.edu) is the Rhesa "Ray" S. Farmer, Jr. distinguished chair in embedded computing systems, and the Georgia Research Alliance Eminent Scholar at the School of Electrical and Computer Engineering at the Georgia Institute of Technology. She was with AT&T Bell Laboratories in Murray Hill, New Jersey from 1984 to 1989. She was with Princeton University from 1989 until 2007. She is a cofounder of Verificon Corporation. She helped to start several technical conferences, including CODES and MPSoC. She has written four textbooks.

MODERATOR

Umit Batur (batur@ti.com) received his B.S. degree in electrical engineering from Bilkent University, Turkey in 1998, and the M.S. and Ph.D. degrees in electrical engineering from Georgia Institute of Technology, Atlanta in 2000 and 2003, respectively. In 2003, he joined the Digital Signal Processing R&D Center of Texas Instruments in Dallas, where he is currently the manager of the imaging R&D branch. **[SP]**