Scaling into Ambient Intelligence

Twan Basten[1]*, Luca Benini[2], Anantha Chandrakasan[3]†, Menno Lindwer[4]*, Jie Liu[5], Rex Min[3]†, Feng Zhao[5]

[1] *Eindhoven University of Technology, Eindhoven, the Netherlands, a.a.basten@tue.nl*

[2] *DEIS, University of Bologna, Bologna, Italy, lbenini@deis.unibo.it*

[3] *Microsystems Technology Laboratories, MIT, Cambridge, MA, USA, {anantha,rmin}@mtl.mit.edu*

[4] *Philips Research Laboratories, Eindhoven, the Netherlands, menno.lindwer@philips.com*

[5] *Palo Alto Research Center (PARC), Palo Alto, CA, USA,{jieliu,zhao}@parc.com*

Abstract

Envision the situation that high quality information and entertainment is easily accessible to anyone, anywhere, at any time, and on any device. How realistic is this vision? And what does it require from the underlying technology?

Ambient Intelligence (AmI) integrates concepts ranging from ubiquitous computing to autonomous and intelligent systems. An AmI environment will be highly dynamic in many aspects. Underlying technology must be very flexible to cope with this dynamism. Scalability of technology is only one crucial aspect. This paper explores scalability from the processing, the communication, and the software perspectives.

keywords*: multi-processor systems on chip, high-density wireless networks, scalable software infrastructure, scalable algorithms, low power and energy*

1 Introduction

Ambient Intelligence (AmI) [11] is the vision that technology will become invisible, embedded in our natural surroundings, present whenever we need it, enabled by simple and effortless interactions, attuned to all our senses, adaptive to users and context and autonomously acting. High quality information and content must be available to any user, anywhere, at any time, and on any device.

It is clear that the AmI vision requires the integration of many different and advanced technologies. Such technologies may range from energy-efficient, high-performance compute platforms and powerful media processing hard- and software to intelligent sensors and actuators, and advanced user-interface designs (vision, speech, gestures). The highly dynamic AmI environment plus tightening cost and time-to-market constraints for AmI products require that the enabling technologies for such products be highly scalable in almost every aspect. An interactive, multi-player gaming device must be able to seamlessly adapt to constantly changing networking conditions; a new generation of a high-end residential gateway must be introduced in the market without the necessity to redesign the processing infrastructure from scratch.

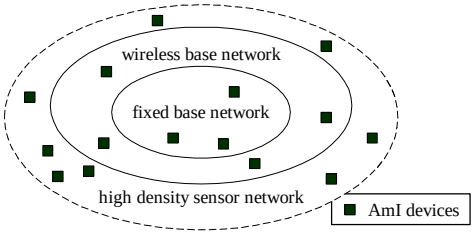

Figure 1. AmI processing and communication infrastructure.

This paper explores scalability of various AmI enabling technologies. The most interesting challenges lie in advanced techniques pushing the current state-of-the-art technologies to the limits. Figure 1 shows a conceptual picture of the AmI infrastructure. AmI will provide a very open environment with processing power present in almost any device. Thus, compute platforms are obviously a core AmI enabling technology. Many AmI devices, particularly server-like systems, will require extremely high processing power. Other, mainly mobile, AmI devices will need (high) processing power at a low energy cost. Section 2 explores scalable processing in an AmI environment in more detail.

A second crucial enabling technology for AmI is communication and network technology. AmI builds on Ubiquitous Computing, as envisioned by the late Mark Weiser [24] in the early 90's that computing technology would soon fade into and seamlessly blend with physical environments. To achieve pervasive computing, a communication infrastructure is needed that must be omnipresent, flexible, and reliable. The base infrastructure will consist of a very high bandwidth fixed base network, augmented with future generation high bandwidth wireless networks. To guarantee the 'anywhere at any time' aspect of communication, the base infrastructure can be complemented with sensor networks consisting of many inexpensive, low power, low bandwidth, densely placed communication nodes. Section 3 discusses issues in high-density wireless networks in more detail.

*Supported by the EU IST-2000-30026 project, Ozone.

†Effort sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, under agreement number F33615-02-2-4005. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

A third essential technology is the embedded software infrastructure required for AmI systems; it must be flexible and fully decentralized. Media and signal-processing algorithms must be highly scalable in order to cope with the very heterogeneous and dynamic environment. Novel programming models are required as well. Scalability from the software point of view is the topic of Section 4.

2 The processing perspective

Ambient Intelligence devices are expected to provide scalable processing power at every level of the networked infrastructure. Flexibility, both at deployment time (i.e., programmability) and in-field (i.e., re-configurability) is required to support scalability: most AmI devices will be programmable and re-configurable, in order to provide adequate performance under widely varying conditions. Furthermore, truly scalable computation must be cheap, reliable and energy-efficient. Even assuming no slowdown in technology scaling, designing highly flexible devices that meet cost, reliability and energy-efficiency requirements is going to be extremely challenging. In fact, as technology scales down, it becomes increasingly unwieldy [12]: (i) logic gates get faster, but wires do not, and performance is increasinlgy interconnect-dominated; (ii) power density (in active state) and leakage power (in idle state) increase significantly even if supply voltage is down-scaled aggressively; (iii) signal-to-noise ratio and circuit reliability decrease; (iv) design complexity scales up.

To address the scalability challenge in view of technology and design complexity limitations, hardware architectures for AmI will have widely varying characteristics depending on the network tier where they are deployed. We briefly overview trends in devices for the three network types of Figure 1.

The fixed base network. Devices for the fixed network infrastructure are rapidly evolving toward parallel architectures. Single-chip multiprocessors are a natural evolution of current single-processor solutions in an effort to support truly scalable data processing. The ever-increasing ratio between wire delay (which remains constant, even in the most optimistic assumptions) and device switching speed (which scales with technology) has become a fundamental bottleneck in designing high-performance integrated systems [9]. The only viable answer to this challenge is to localize computation and emphasize parallelism. For this reason, even general-purpose single-chip processors for high-end servers are becoming increasingly distributed and highly parallel. Explicit parallelism is sought to avoid architectural bottlenecks (such as global register files and instruction fetch & decode logic) and to ensure long-term scalability across several technology generations.

Most newly-designed high-performance processors are highly parallel architectures, with multiple program counters [20]. Simultaneous multi-threading is emerging as the new architectural leit-motif [5]. Moreover, hardware designers routinely resort to distributed implementation styles: multiple clock domains, multiple execution units, distributed register files are representative instances of this trend [28]. One step further, next-generation architectures are focusing on parallelism not only at the micro-architectural level, but also across the entire memory hierarchy. Multi-processing becomes multi-streaming: streaming supercomputers are under development [13] where computation and storage are seen as successive stages of an information processing pipeline (a stream), and communication is made explicit as much as possible.

The wireless base network. When considering the wireless base network, the push toward highly parallel multi-processing architectures is even stronger, coupled with a trend toward single-chip integration. This convergence toward multi-processor system-on-chip (MPSoC) platforms is also motivated by the quest for scalability. Energy efficiency and cost constraints are much tighter than for high-performance servers, and the computational requirements can be matched only by resorting to heterogeneous architectures (in contrast with homogeneous general-purpose processors), which provide computational power tailored to a specific class of applications [27, 14]. Designing such heterogeneous, application-specific (AS) MPSoCs is a challenging task because their complexity is comparable to that of most aggressive general-purpose platforms, yet their time-to-market and profitability windows are much shorter and focused.

To tackle the design challenges of AS-MPSoCs, architectures and design flows are undergoing a profound revolution. Computational elements must be programmable (with application-specific programming paradigms, such as fine-grained bit-level, control-dominated word-level, dataflow, etc.) to provide the much needed post-fabrication programmability and in-field adaptation. Additionally, they should be standardized as much as possible to ensure software portability and to facilitate interfacing with other computational elements, on-chip and off-chip memories and input-output devices. The communication infrastructure that provides the connective fabric for the computational and storage elements on a single chip becomes the most critical architectural element. Current shared-medium (bus-based) standardized fabrics are not scalable, and scalable network-on-a-chip (NoC) communication architectures are under active exploration and development [2].

Next-generation wireless base network devices will include tens to hundreds of application-specific processors (e.g., MAC accelerators, digital MODEMS, cryptoprocessors), as well as several general-purpose processing cores (e.g., FPGA fabrics, DSP cores, VLIW multimedia processors, FP coprocessors, RISC controllers, etc.), a significant

amount of on-chip storage (both volatile and non-volatile), various peripheral units (e.g., external DRAM controllers, interfaces to on-board buses, RF front-ends, BIST units, etc.). The connective fabric for this multitude of devices will be a NoC, most likely a packet-based network with QoS guarantees [8]. These platforms will heavily rely on technology support for aggressive power management. Multiple and variable clock-frequencies, supply voltages, transistor thresholds will be supported on the same chip [15].

The sensor network. The nodes of high-density sensor networks will be the most critically power and cost constrained. Scalability will most likely be achieved by compromising to some degree flexibility at the node level, focusing on maximum energy efficiency and minimum cost. Individual nodes will be extremely simple, but a sensor network will contain a large number of them, tightly connected to the wireless base network infrastructure. Ambient intelligence will therefore be obtained as a result of coordination and orchestration of a very large number of simple nodes through distributed, adaptive protocols for information transfer, processing and active energy management [7].

Integration in sensor nodes will be pushed to the limit to maximize power efficiency, and technology support for mixed digital-analog and MEMS will be required [23]. In the digital section, protocol processing and power management functions will be carried out by a simple programmable controller, while on-board sensor data processing and base-band communication will be performed by dedicated, highly-tuned hardware co-processors with limited or no flexibility and ultra-low supply voltage for minimum power consumption [21]. The node itself will not be highly scalable, but the sensor network as a whole will be, mostly thanks to its protocol-level scalability.

Trends. In summary, we see a common trend, across all the levels of the AmI network hierarchy, toward highly parallel architectures. As energy and cost constraints get tighter, however, heterogeneity and reduced flexibility set in, coupled with dedicated technology options.

3 The communication perspective

Emerging devices for AmI are characterized by shrinking size and increasing density. From the perspective of wireless communication, a high density of nodes implies competition over limited bandwidth, and the devices' small physical size suggests a limited battery capacity. It is therefore imperative that communication in AmI devices utilize these resources efficiently. This section focuses on the management and conservation of the latter resource –energy– as wireless devices scale into ambient intelligence.

We advocate two design philosophies for energy-efficient communication. First, it is crucial that energy-efficient communication software be based on sound models of the hardware on which it will operate. Incomplete or inaccurate energy models lead to surprising discrepancies between designers' expectations and system realities. Second, two power management techniques used widely by hardware designers hold great promise for protocols: application-specific design and energy-quality scalability.

Energy-scalable communication. Power-aware digital hardware frequently trades energy for quality. For instance, digital processing is more slowly and consumes less energy when the circuits' supply voltage is reduced. Dynamic voltage scaling therefore enables a graceful energy *vs.* latency trade off that responds to an application's changing needs. *Energy scalability* ensures that energy is not wasted by providing performance in excess of an application's needs.

Graceful energy *vs.* quality scalability for wireless communication can be achieved once the notion of communication "quality" is defined. We can define communication quality by four of its fundamental metrics: *range*, *reliability*, *latency* and *throughput*. Various combinations of these parameters can then be chosen by an application through a basic API. To bridge the gap between these performance parameters and the actual hardware "knobs" for energy scalability, we suggest a power-aware middleware layer in between the hardware and the communication software. The middleware manager must be empowered with accurate hardware energy models for the digital processing circuits and radio transceiver, allowing this layer to select the minimum-energy hardware settings for the performance level commanded through the API [18].

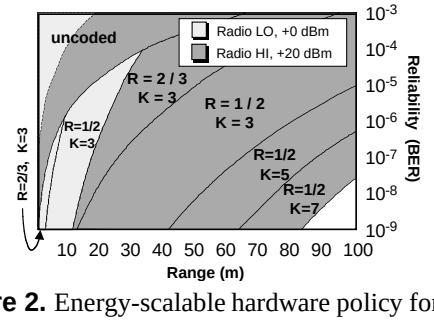

Figure 2. Energy-scalable hardware policy for communication on the μAMPS microsensor node [22].

Figure 2 illustrates a sample operational policy for a wireless microsensor node [22]. Reliability and range, two API-specified quality metrics, form the x- and y-axes. The shaded areas within the graph represent the minimum-energy hardware operational policies for radio transmission power (0 dBm *vs.* 20 dBm) and convolutional coding scheme (uncoded, or various combinations of code rate R and constraint length K) that result in minimum energy consumption. The range and reliability of communication increase as more power is radiated from the transmitter or stronger (lower R, higher K) error-correcting codes are used, resulting in longer transmit and receive times, and higher decoding energy.

Platform- and application-aware protocols. Many contemporary wireless communication protocols suffer from two related weaknesses: they are too general-purpose, and they are not designed with energy in mind. For example, the reliable transport and bandwidth fairness mechanisms of TCP are overhead for applications that do not require them. It is unreasonable to assume that one media access layer can efficiently support environmental microsensing, two-way voice, and on-demand video. Instead, protocols should be tuned to the target hardware and application, just as an ASIC is tuned for application-specific processing.

Consider the energy consumption of hardware. The radiated power necessary to transmit over a distance d is roughly d^n (with n typically between 2 and 4). Hence, *multihop* routing, the use of several shorter transmissions via intermediate relay devices, is commonly proposed to reduce this *path loss* from d^n to $h(d/h)^n$ for an h-hop relay. With the ubiquity of devices promised by AmI, multihop is especially appealing at first glance. However, d^n alone fails to consider the energy overheads of real-world hardware. A simple but more complete model for energy consumption per bit is $\alpha + \beta d^n$, where α is a distance-independent term that accounts for the overheads of transmitter and receiver electronics (PLLs, VCOs, bias currents, etc.) and digital processing, and β models power amplifier and antenna inefficiencies. Unfortunately, for virtually all of today's short-range radios, the term α substantially exceeds βd^n, even at the radio's maximum range [19]! Hence, while multihop remains useful for long-range radios, we must reconsider this technique for high-density AmI applications.

So an interesting question is what *application-specific protocols* can provide concise, energy-efficient solutions for specific AmI applications. One particular example is a protocol tuned for unidirectional data propagation in high-density microsensor networks. Microsensor nodes, which gather and relay observations about stimuli in the environment (people, vehicles, weather, etc.) to a central base station, have perhaps the lowest performance and highest lifetime requirements of all AmI devices. Thus, the only viable protocol is an application-specific one. In a microsensor network, all data is destined to predetermined sinks, and the entire notion of addressing a packet through a specific relay node is unnecessary. The only concern is that packets move progressively closer to a base station. This suggests a routing method that replaces addresses with measurements of distance to a base station [17]. A node wishing to send data to a base station simply broadcasts the packet including its own distance metric. Nodes that receive the packet compare their own distance metric to that of the packet, and the receiving node that is closest to the base station and farthest from the originating node relays the packet onward. This is implemented with a delay timer proportional to the difference between the packet's and relay node's distance metrics, plus a random delay for arbitration. The node with the lowest delay forwards the packet, and the others, hearing the forward, drop their respective copies. This surprisingly simple protocol achieves performance comparable to TCP-based *ad hoc* protocols while permitting greater flexibility in radio receiver shutdown.

Trends. We foresee the emergence of application-specific protocols for the many modalities supported by AmI devices, and increasing cooperation between hardware and protocols to enable energy- and quality-scalable communication. These trends, guided by accurate characterizations of the energy usage of hardware and applications, will ensure that AmI devices deliver the battery life required by the most demanding users.

4 The software perspective

As wirelessly networked intelligent sensors become the mass majority of AmI devices that deeply embed into the physical world, the amount of gathered information, the complexity of software development, and the cost of system testing will soon surpass what current technologies can support. As illustrated in Figure 1, these devices must be seamlessly integrated with mobile computing devices and fixed backbone infrastructures. The vision of AmI requires a fundamental paradigm shift on system architectures, programming models, and algorithm designs, for example, so as to integrate client/server frameworks with peer-to-peer frameworks, address-based routing with name/data-centric routing, location-transparent computing with location- and resource-aware computing, fixed, centralized processing with adaptive, in-network processing.

Scalable software infrastructure. AmI systems need drastically different software infrastructure support than the current Internet or desktop systems, especially for deeply embedded sensor networks. Infrastructural services, such as synchronized time, directories of all nodes and their configurations, global routing tables, global operation modes and sequencing of events, may be extremely hard to set up in AmI systems, not mentioning the resources required for maintaining them. Some devices in an AmI system are mobile, and cannot be assumed to connect to the fixed base network all the time. Some devices may have various duty cycles to preserve energy. Many AmI systems will be formed in ad hoc ways or re-configured from existing systems. Interactions among devices may be established because of their location proximity. Capabilities to form ad hoc networks, discover relevant services, and interoperate across heterogeneous devices are crucial. Recent advances, such as data-centric routing [10] and recombinant computing [6], although still in development, have shown promising trends on ad hoc routing and transportable computation.

A scalable AmI software infrastructure also needs effective ways to manage tasks and resources. Having sens-

ing and computing deeply embedded into our natural surroundings greatly improves the way in which raw data is collected, but also introduces a large amount of unpredictability in task requirements and resource availability. An AmI system must prioritize among multiple competing tasks using a limited amount of resources, and coordinate tasks across multiple embedded, mobile and fixed networks. Cross-node task management in deeply embedded systems is still an open issue.

Scalable algorithms. On wireless devices, the ratio of the energy cost for transmission to processing can vary greatly, highlighting the need for judicious tradeoffs between processing and communication. If communication is the dominant factor in the energy consumption, one way to expand the life span of an AmI system is to selectively switch nodes to low-duty-cycle mode, either by reducing the voltage supply or by putting them into sleep. For example, in one embedded sensor platform known as the Berkeley MICA mote [4], one second of sleeping can save enough energy to transmit about 16K bits. Many centralized and distributed algorithms have to be re-examined under these energy characteristics.

Traditional signal processing approaches focus on optimizing estimation quality for a fixed set of resources. However, for power-limited and multi-user decentralized systems, it becomes critical to carefully select the embedded sensor nodes participating in a sensor collaboration, balancing the information contribution of each against its resource consumption or potential utility for other users. We term this style of computation in distributed embedded systems as *collaborative signal and information processing* (CSIP).

Scalable AmI systems need resource-aware CSIP algorithms and novel ways of resource management. Resource-aware algorithms adapt to the available CPU cycles, bandwidth, sensing modalities, and battery power and may provide a spectrum of answers, quantitative ones or qualitative ones, with different fidelity. The resource management scheme, depending on application-specific characteristics such as the dynamics of the physical phenomena and/or user requests, may select proper algorithms to provide satisfactory answers under global constraints.

CSIP algorithms also need to be scalable with the number of nodes under limited infrastructure support. Thus:

1) Individual AmI devices should rely on as little global coordinations as possible. Algorithms must have some degree of autonomy and operate asynchronously.

2) The "capability" of an AmI system should increase with the number of nodes. As the size of the system increases, more tasks should be enabled.

A key to achieve this scalability is the formation of *local collaboration groups*. Each device in an AmI system may only have limited resources and limited information of the world. To produce useful information without transmitting raw data to central locations, a node must collaborate with some of its neighbors. Take distributed target tracking as an example, where a set of wirelessly networked microsensors collaboratively track a moving target using, say, acoustics sensing [26]. At a given time, each sensor node only has a limited "view" of the target. It should selectively collaborate with some of its neighbors, which may have a different "view" of the target, so that the overall estimation of the target position can be maximally improved, an algorithm we call *information-driven sensor query* (IDSQ). This collaboration is completely local. As the target moves, the collaboration group follows the target, and leaves other nodes in the network free to perform other tasks.

Programming models. One of the most challenging, and probably least understood, issues about AmI systems is how to program them. Traditional embedded system programming technologies, rooted in writing device drivers and optimizing at assembly level for small footprint and fast response, do not scale. AmI systems deeply embed into our surroundings, inevitably inheriting physical properties such as space and time. However, location and time have been systematically removed from programming languages in particular, and computer sciences in general, to raise the level of abstraction. How to re-introduce physical properties into programming without suffering in abstraction is a key challenge.

We advocate the development of formal programming models at a collaborative behavioral level, and the use of software synthesis technologies to automatically generate the interactions of algorithm components, a methodology depicted in Figure 3.

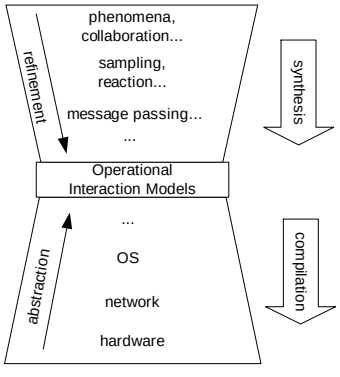

Figure 3. A programming methodology for deeply embedded systems.

At the top level, the programming models should be expressive enough to describe physical phenomena, user interaction, and collaborative algorithms, without introducing node-by-node instructions. These models will be domain specific. For example, the language SAL [25] allows reasoning about physical geometries in distributed sensing and control systems; various biologically inspired computational models [1, 3] study collaborative complex behav-

iors built from simple components. The programming models should also be structural enough to allow synthesis algorithms to exploit patterns and generate effective code. We believe that automated software synthesis is a critical step toward the scalability of programming AmI systems. Hardware-oriented concerns such as timing and location will be introduced gradually by the refinement process. A set of operational models would serve as the common substrates for code generation. These common substrates will abstract away hardware idiosyncrasy across different platforms, but still expose enough information for applications to take advantage of low-level features. These models have to manage concurrencies well as the hardware architectures are becoming more and more parallel. No single model may be universal for all applications, thus understanding the semantics of the set of models and their interactions is essential [16]. At the end, platform dependent compilers can compile the generated application into executables.

5 Conclusions

The emergence of Ambient Intelligence (AmI) is a trend that is unstoppable. It will have a huge impact on everyday life. AmI systems will radically differ from today's systems. We will need to rethink everything we thought to know about embedded-system and hardware design. The AmI vision requires scalability of processing, communication, and software infrastructure in very many aspects. It is clear that the design community faces a lot of interesting challenges. It should be fun to be an engineer for the next few years.

References

[1] H. Abelson et al, "Amorphous Computing." *Communications of the ACM* 43(5):74-82, May 2001.

[2] L. Benini, G. De Micheli, "Networks on Chip: A New SOC Paradigm." *IEEE Computer* 35(1):70–78, Jan. 2002.

[3] C. Calude, M.J. Dinneen, and F. Peper (Eds.), *Unconventional Models of Computation*. LNCS 2509, Springer, 2002.

[4] Crossbow, *MICA data sheet*, http://www.xbow.com/Products/Product_pdf_files/MICA%20data%20sheet.pdf

[5] D. Deleganes, J. Douglas, B. Kommandur, M. Patyra, "Designing a 3 GHz, 130 nm, Intel Pentium 4 processor." *IEEE Symp. on VLSI Circuits*, pp. 130–133, Jun. 2002.

[6] W.K. Edwards, M.W. Newman, J. Sedivy, T. Smith, and S. Izadi, "Challenge: Recombinant Computing and the Speakeasy Approach." *Proc. MobiCom'02*, 8th International Conference, Sep. 2002, Atlanta, GA, pp. 279-286.

[7] D. Estrin, D. Culler, K. Pister, G. Sukhatme, "Connecting the Physical World with Pervasive Networks." *IEEE Pervasive Computing* 1(1): 59–69, Jan.-Mar. 2002.

[8] K. Goossens, J. van Meerbergen, P. Wielage, A. Peeters, "Networks on Silicon: Combining Best-Effort and Guaranteed Services." *Proc. DATE'02*, Mar. 2002, pp. 423–425.

[9] R. Ho, K. Mai, M. Horowitz, "The Future of Wires." *Proc. IEEE* 89(4):490–504, Apr. 2001.

[10] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks." *Proc. MobiCom'00*, 6th International Conference, Aug. 2000, Boston, MA, pp. 56-67.

[11] ISTAG, *Scenarios for Ambient Intelligence in 2010*. Feb. 2001. http://www.cordis.lu/ist/istag.htm.

[12] T. Karnik, S. Borkar, V. De, "Sub-90nm Technologies – Challenges and Opportunities for CAD." *Proc. IEEE/ACM ICCAD'02*, Nov. 2002, pp. 203-206.

[13] B. Khailany, et al., "Imagine: Media Processing with Stream." *IEEE Micro* 21(2), Mar.-Apr. 2001.

[14] T. Koyama, K. Inoue, H. Hanaki, M. Yasue, E. Iwata, "A 250-MHz Single-chip Multiprocessor for Audio and Video Signal Processing." *IEEE J of Solid-State Circuits* 36(11): 1768–1774, Nov. 2001.

[15] D. Lackey, et al., "Managing Power and Performance for Systems-on-chip Designs using Voltage Islands." *Proc. IEEE/ACM ICCAD'02*, Nov. 2002, pp. 195-202.

[16] E.A. Lee, "What's Ahead for Embedded Software?" *IEEE Computer*, Sep. 2000, pp. 18-26.

[17] R. Min and A. Chandrakasan. "Energy-Efficient Communication for Ad-Hoc Wireless Sensor Networks." *Proc. 35th Asilomar Conf. on Signals, Systems, and Computers*, vol. 1, Nov. 2001, pp. 139-143.

[18] R. Min and A. Chandrakasan, "A Framework for Energy-Scalable Communication in High-Density Wireless Networks." *Proc. ISLPED 2002*, Aug. 2002.

[19] R. Min and A. Chandrakasan, "Top Five Myths about the Energy Consumption of Wireless Communication". *ACM Sigmobile Mobile Communication and Communications Review (MC2R)* 6(4), to be published.

[20] A. Moshovos, G. Sohi, "Microarchitectural Innovations: Boosting Microprocessor Performance beyond Semiconductor Technology Scaling." *Proc. IEEE* 89(11): 1560–1575, Nov. 2001.

[21] J. Rabaey et al., "PicoRadios for Wireless Sensor Networks: The Next Challenge in Ultra-low-power Design." *IEEE Int. Solid-State Circuits Conf.*, pp. 200–201, Feb. 2002.

[22] E. Shih, et al., "Physical Layer Driven Algorithm and Protocol Design for Energy-efficient Wireless Sensor Networks." *Proc. MobiCom'01*, Jul. 2001.

[23] B. Warneke, M. Last, B. Liebowitz, K. Pister, "Smart Dust: Communicating with a Cubic-millimeter Computer." *IEEE Computer* 34(1): 44-51, Jan. 2001.

[24] M. Weiser, "The Computer for the Twenty-First Century." *Scientific American*, Sep. 1991, pp. 94–104.

[25] F. Zhao, C. Bailey-Kellogg, and M. Fromherz, "Physics-Based Encapsulation in Embedded Software for Distributed Sensing and Control Applications." *Proc. IEEE*. To appear.

[26] F. Zhao, J. Shin, J. Reich, "Information-Driven Dynamic Sensor Collaboration for Tracking Applications." *IEEE Signal Processing Magazine* 19(2), Mar. 2002.

[27] H. Zhang, et al., "A 1-V Heterogeneous Reconfigurable DSP IC for Wireless Baseband Digital Signal Processing." *IEEE J of Solid-State Circuits* 35(11): 1697–1704, Nov. 2000.

[28] V. Zyuban, P. Kogge, "Inherently Lower-power High-performance Superscalar Architectures." *IEEE Transactions on Computers* (50)3: 268–285, Mar. 2001.