AN ARCHITECTURE FOR A POWER-AWARE DISTRIBUTED MICROSENSOR NODE

Rex Min, Manish Bhardwaj, Seong-Hwan Cho, Amit Sinha,
Eugene Shih, Alice Wang, and Anantha Chandrakasan

Department of Electrical Engineering
Massachusetts Institute of Technology
Cambridge, MA

Abstract - Networks of distributed microsensors are emerging as a compelling solution for a wide range of data gathering applications. Perhaps the most substantial challenge facing designers of small but long-lived microsensor nodes is the need for significant reductions in energy consumption. We propose a power-aware design methodology that emphasizes the graceful *scalability* of energy consumption with factors such as available resources, event frequency, and desired output quality, at all levels of the system hierarchy. Our architecture for a power-aware microsensor node highlights the collaboration between software that is capable of energy-quality tradeoffs and hardware with scalable energy consumption.

1. INTRODUCTION

The design of micropower wireless sensor systems has gained increasing importance for a variety of civil and military applications. Advances in MEMS technology and its associated interfaces, signal processing, and RF circuitry have enabled the development of wireless sensor nodes. The focus has shifted from limited macrosensors communicating with base stations to creating wireless networks of communicating microsensors, as illustrated in Figure 1. Such sensor networks aggregate complex data to provide rich, multi-dimensional pictures of the environment. While individual microsensor nodes are not as accurate as their expensive macrosensor counterparts, their size and cost will enable the networking of hundreds or thousands of nodes in order to achieve high quality, easily deployed, fault-tolerant sensing networks [1][2].

A key challenge in the design of a microsensor node is low energy dissipation. In this paper, we advocate *power-aware* system design, which calls for a system whose energy consumption adapts to constraints and variations in the environment, onboard resources, or user requests. Power-aware design methodologies offer scalable energy savings that are ideal for the high variabilities of the microsensor environment..

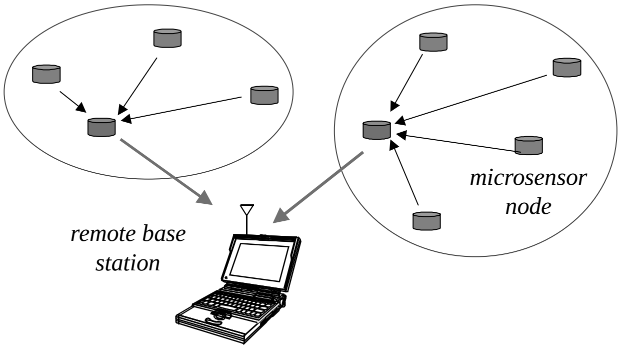


Figure 1. Microsensor networks for remote sensing.

2. POWER-AWARE SYSTEM DESIGN

Low-power system design assuming a worst-case power dissipation scenario is being supplanted by a more comprehensive philosophy variously termed *power-aware* or *energy-aware* or *energy-quality scalable* design [3]. The basic idea behind these essentially identical approaches is to allow the system's energy consumption to scale with changing conditions and quality requirements.

There are two main views motivating power-aware design and its emergence as an important paradigm. The first view is to explain the importance of power-awareness as a consequence of the increasing emphasis on making systems more scalable. In this context, making a system scalable refers to enabling the user to tradeoff system performance parameters as opposed to hard-wiring them. Scalability is an important figure-of-merit since it allows the end-user to implement operational policy, which often varies significantly over the lifetime of the system. At times, the user of a microsensor network might want extremely high performance (e.g., data with a high signal-to-noise ratio) at the cost of reduced battery lifetime. However at other times, the opposite might be true—the user may be willing to trade off quality in return for maximizing battery lifetime. Such trade-offs can only be optimally realized if the system is designed in a power-aware manner. A related motivation for power-awareness is that a well-designed system should gracefully degrade its quality and performance as available energy resources are depleted instead of exhibiting an "all-or-none" behavior of high-SNR data followed by a network failure [4].

While the view above argues for power-awareness from a user-centric and user-visible perspective, this paradigm can also be motivated in more fundamental, system-oriented terms. With burgeoning system complexity and the accompanying increase in integration, there is more diversity in operating scenarios than ever before. Hence, design philosophies that assume the system to be in the worst-case operating state most of the time are prone to yield globally sub-optimal results. This naturally leads to the concept of power-awareness. For instance, the embedded processor in a sensor node can display tremendous workload diversity depending on activity in the environment. Nodes themselves can also play a variety of roles in the network; a sensor networking protocol may call for the node to act as a data gatherer, aggregator, relay, or any combination of these [5][6]. Hence, even if the *user* does not explicitly change quality criteria, the processor can nevertheless exploit operational diversity by scaling its energy consumption as the workload changes.

In the following sections we introduce prototype hardware and design methodologies for a power-aware microsensor node. Through these examples, we emphasize that power-awareness as a design driver does not necessarily devolve to traditional, worst-case-centric low-power/low-energy design.

3. SENSOR NODE ARCHITECTURE

A prototype of a sensor node that illustrates power-aware design methodologies is outlined in Figure 2. This system, the first prototype of our μAMPS (micro-Adaptive Multi-domain Power-aware Sensors) effort [7], is designed with commercial off-the-shelf components for rapid prototyping and modularity.

Power Supply. Power for the sensor node is supplied by a single 3.6V DC source, which can be provided by a single lithium-ion cell or three NiCD or NiMH cells. Regulators generate 5V, 3.3V, and adjustable 0.9-1.5V supplies from the battery. The 5V supply powers the analog sensor circuitry and A/D converter. The 3.3V supply powers all digital components on the sensor node with the exception of the processor core. The core is powered by a digitally adjustable switching regulator that can provide 0.9V to 1.6V in twenty discrete increments. The digitally adjustable voltage allows the SA-1100 to control its own core voltage, enabling dynamic

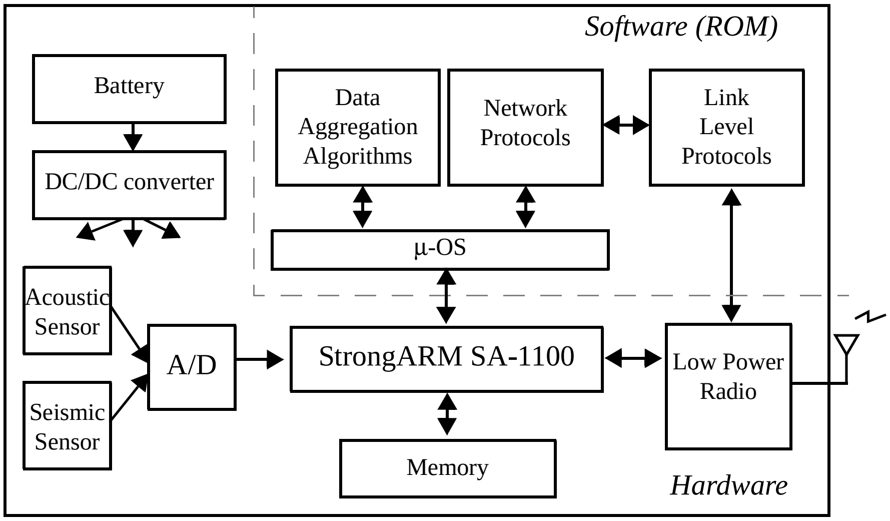


Figure 2. μAMPS sensor node hardware and software framework.

voltage scaling techniques. Section 4.2 elaborates on dynamic voltage scaling.

Sensors. The node includes seismic and acoustic sensors. The seismic sensor is a MEMS accelerometer capable of resolving 2 mg. The acoustic sensor is an electret microphone with low-noise bias and amplification. The analog signals from these sensors are conditioned with 8th-order analog filters and are sampled by a 12-bit A/D. The high-order filters eliminate the need for oversampling and additional digital filtering in the SA-1100. All components are carefully chosen for low power dissipation; a sensor, filter, and A/D typically requires only 5 mA at 5 Volts.

Microprocessor and Operating System. A StrongARM SA-1100 microprocessor is selected for its low power consumption, sufficient performance for signal processing algorithms, and static CMOS design. The memory map mimics the SA-1100 “Brutus” evaluation platform and thus supports up to 16 MB of RAM and 512 KB of ROM. The lightweight, multithreaded “μOS” running on the SA-1100 is an adaptation of the eCOS microkernel [8] that has been customized to support the power-aware methodologies discussed in Section 4. The μOS, data aggregation algorithms, and networking firmware are embedded into ROM.

Radio. The radio module interfaces directly to the SA-1100. The radio is based on a commercial single-chip transceiver optimized for ISM 2.45 GHz wireless systems. The PLL, transmitter chain, and receiver chain are capable of being shut-off under software or hardware control for energy savings. To transmit data, an external voltage-controlled oscillator (VCO) is directly modulated, providing simplicity at the circuit level and reduced power consumption at the expense of limits on the amount of data that can be transmitted continuously. The radio module is capable of transmitting up to 1 Mbps at a range of up to 15 meters.

4. POWER-AWARE METHODOLOGIES

In this section, we present energy-scalable design methodologies geared specifically toward our microsensor application. At the hardware level, we note the unusual energy consumption characteristics effected by the low duty cycle operation of a sensor node, and adapt to varying active workload conditions with dynamic voltage scaling. At the software level, energy-agile algorithms for sensor networks such as adaptive beamforming provide energy-quality tradeoffs that are accessible to the user. Power-aware system design encompasses the entire system hierarchy, coupling software that understands the energy-quality tradeoff with hardware that scales its own energy consumption accordingly.

4.1 Low Duty Cycle Issues

The energy consumption characteristics of the components in a microsensor node provide a context for the power-aware software to make energy-quality tradeoffs. We discuss the energy consumption of the processor and radio in this section and expand our discussion of the processor in the following section.

Energy consumption in a static CMOS-based processor can be classified into

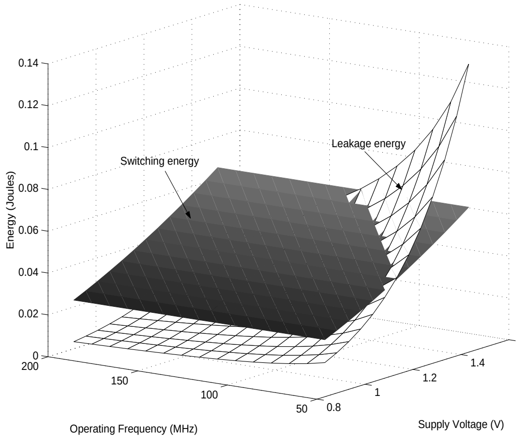


Figure 3. Comparison of leakage and switching energy in SA-1100

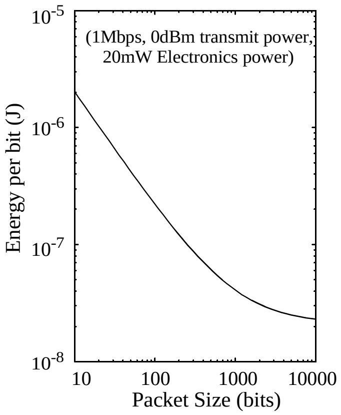


Figure 4. Modeled energy consumption per bit for a radio transmission of 1000 bits

switching and leakage components. The switching energy is expressed as $E_{switch} = C_{tot}V_{dd}^2$ where C_{tot} is the total capacitance switched by the computation and V_{dd} is the supply voltage. Energy lost due to leakage currents is modeled with an exponential relation to the supply voltage [9]:

$$E_{leak} = (V_{dd}t)I_O e^{\frac{V_{DD}}{(nV_T)}} \tag{1}$$

While switching energy is usually the more dominant of the two components [10], the low duty cycle operation of a sensor node can induce precisely the opposite behavior. Figure 3 demonstrates that, for sufficiently low duty cycles or high supply voltages, leakage energy can exceed switching energy. For example, when the duty cycle of the StrongARM SA-1100 is 10%, the leakage energy is more than 50% of the total energy consumed. Techniques such as dynamic voltage scaling and the progressive shutdown of idle components in the sensor node mitigate the energy consumption penalties of low duty cycle processor operation.

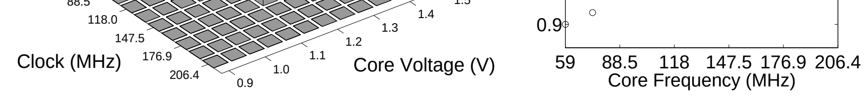
Low duty cycle characteristics are also observable in the radio. Figure 4 illustrates the energy required to power up a radio and transmit a packet of varying length. Ideally, the energy consumed per bit would be independent of packet length. At lower data rates, however, the start-up overhead of the radio's electronics begins to dominate the radio's energy consumption. Due to its slow feedback loop, a typical PLL-based frequency synthesizer has a settling time on the order of milliseconds, which may be much higher than the transmission time for short packets. Particular effort is required to reduce transient response time in low power frequency synthesizers for low data rate sensor systems.

4.2 Dynamic Voltage Scaling

Dynamic voltage scaling (DVS) exploits variabilities in processor workload and latency constraints and realizes this energy-quality tradeoff at the circuit level [11][12]. As discussed above, the switching energy of any particular computation is $E_{switch} = C_{tot}V_{dd}^{2}$, a quantity that is independent of time. Reducing V_{dd} offers a quadratic savings in switching energy at the expense of additional propagation delay through static logic. Hence, if the workload on the processor is light, or the latency tolerable by the computation is high, we can reduce V_{dd} and the processor clock frequency together to trade off latency for energy savings. Both switching *and* leakage energy are reduced by DVS; as Equation 1 indicates, leakage energy varies more than exponentially with V_{dd}.

Figure 5a depicts the measured energy consumption of a SA-1100 processor running at full utilization. Energy consumed per operation is plotted with respect to the processor frequency and voltage. As discussed above, a reduction in clock frequency allows the processor to run at lower voltage. The quadratic dependence of switching energy on supply voltage is evident, and for a fixed voltage, the leakage energy per operation increases as the operations occur over a longer clock period. Our selections of voltages corresponding to each frequency, plotted in Figure 5b, provide energy savings at reduced clock frequencies with a reasonable margin of safety.

Figure 6 illustrates the regulation scheme on our sensor node for DVS support. The μOS running on the SA-1100 selects one of the above eleven frequency-voltage pairs in response to the current and predicted workload. A five-bit value corresponding to the desired voltage is sent to the regulator controller, and logic external to the SA-1100 protects the core from a voltage that exceeds its maximum rating. The regulator controller typically drives the new voltage on the buck regulator in under 100 μs. At the same time, the new clock frequency is programmed into the



(a)

(b)

Figure 5. (a) Measured energy consumption characteristics of SA-1100.
(b) Selection of (*frequency, voltage*) pairs for DVS

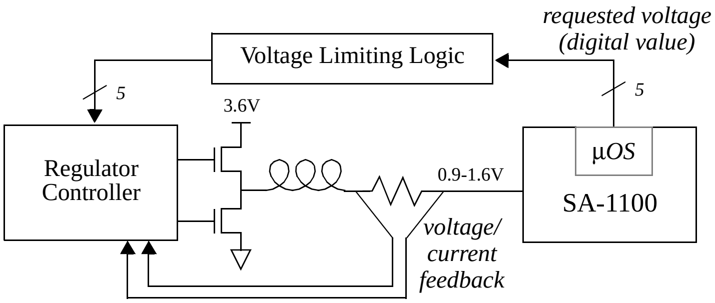
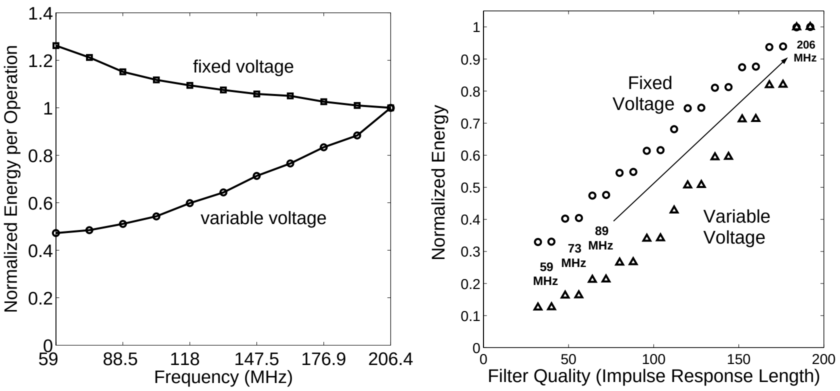


Figure 6. Feedback for dynamic voltage scaling.



(a)

(b)

Figure 7. (a) Measured energy savings in an energy *vs.* latency trade-off.
(b) Energy *vs.* filter performance

SA-1100, causing the on-board PLL to lock to the new frequency. Relocking the PLL requires 150 μs, and computation stops during this period.

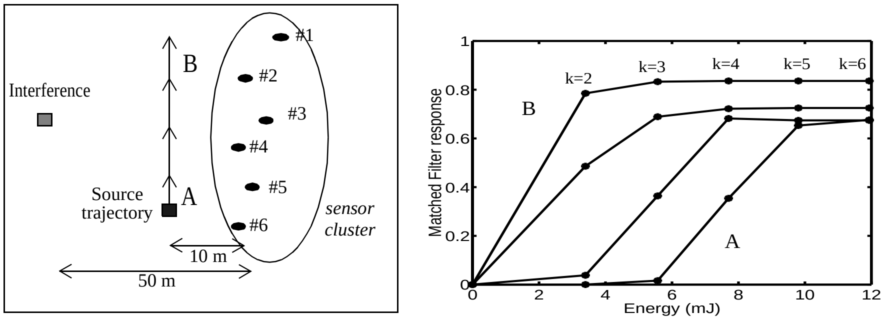
Our implementation of the above system demonstrates energy-quality tradeoffs with DVS. In Figure 7a, for a fixed computational workload, the latency (the inverse of quality) of the computation increases as the energy decreases. In Figure 7b, the quality of a FIR filtering algorithm is varied by scaling the number of filter taps. As we sacrifice filter quality, the processor can run at a lower clock speed and thus operate at a lower voltage. In each example, our DVS-based implementation of energy-quality tradeoffs consumes up to 60% less energy than a fixed-voltage processor.

4.3 Energy-Agile Algorithms

As the node's processor is capable of scaling energy consumption gracefully with computational workload, we can exploit this scalability at the algorithm level with *energy-agile* algorithms of scalable computational complexity. Scalability at the algorithm level is highly desirable because a large range of both energy and quality can be achieved. As the energy-quality characteristics of DSP algorithms may not be optimal due to data dependencies, it is important to use algorithmic transforms to achieve desirable energy-quality (E-Q) characteristics and accurately model the energy-quality relationship through benchmarking.

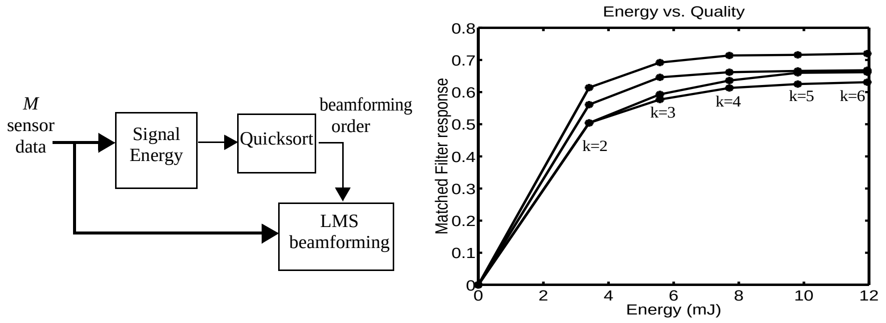
Algorithmic transformations such as the *most significant first transform* can improve the E-Q characteristics of a particular algorithm by reducing data dependencies. Figure 8a shows our testbed of sensors for beamforming [13][14], a class of algorithms often used in sensor arrays to make inferences about the environment. In our testbed, an array of six sensors is spaced roughly linearly at intervals of approximately 10 meters, a source moves parallel to the sensor cluster at a distance of 10 meters, and interference exists at a distance of 50 meters. We perform beamforming on the sensor data as we vary the number of sensors *k*, and we measure the energy dissipated on the StrongARM SA-1100 in relation to *k*. We calculate the matched filter output (quality) and derive a reliable model of the *E-Q* relationship as we vary the number of sensors in beamforming.

We compare the E-Q characteristics for two scenarios, the first being traditional beamforming, and the second using a most significant first transform. In the first scenario, beamforming is simply done in a preset order <$^{\#}1,^{\#}2,^{\#}3,^{\#}4,^{\#}5,^{\#}6$>. As the



(a) (b)

Figure 8. (a) Sensor testbed. (b) E-Q curve for Scenario 1.



(a) (b)

Figure 9. (a) Most significance first transform used to improve E-Q characteristics of LMS beamforming. (b) E-Q curves for Scenario 2.

source moves from location A to B, the E-Q curves changes dramatically as seen in Figure 8b. With the source in location A, the beamforming quality is close to maximum when *k=5,6* because the source is closest to sensors $^{\#}$5 and $^{\#}$6. However, with the source at B, quality is close to maximum after beamforming only 2 sensors, thus showing the dependency of the E-Q graph on the relative source location.

Intelligent data processing can circumvent this dependency. Intuitively, we wish to beamform the data from sensors which have higher signal energy to interference energy, or process the *most significant first*. Figure 9a shows a block diagram for applying a most significant first transform to beamforming. To find the desired beamforming order, each sensor's data energy is estimated. The energies are then sorted using quicksort. The quicksort output determines the desired beamforming order. We plot the E-Q curves for the new scenario in Figure 9b. By finding the desired beamforming order, we achieve similar E-Q plots even as the source moves with respect to the sensors. The energy cost required to gain this additional scalability is low compared to the energy cost of LMS beamforming itself: on the SA-1100, the additional computational cost was 8.8mJ, which is only 0.44% of the total energy for LMS beamforming (for the 2 sensor case). The incremental refinement characteristics of a sensor's beamforming algorithm are improved, leading to more uniform and predictably scalable E-Q curves in the presence of data dependencies.

The energy scalable framework proposed in this paper enables the development and implementation of energy-agile applications. It will be important that all processing in the sensor node be energy scalable, including link level protocols, sensor network protocols, data aggregation algorithms, and sensor signal processing.

5. CONCLUSION

Our initial μ-AMPS sensor node prototype demonstrates the effectiveness of *power-aware* system design methodologies. Inefficiencies of low duty cycle operation are countered with a focus on leakage current and start-up time reduction, and variations in processor workload are exploited by dynamic voltage scaling. Variations in incoming data rate and volume are exploited by energy-agile algorithms whose computational complexity scales with the arrival statistics of the data, allowing switching energy savings in the hardware. Close collaboration between the hardware and software of a microsensor node result in dramatic energy savings.

But the power-awareness philosophy captures more than just energy savings. Inherent to power-awareness is an adaptability to changing environmental conditions and resources, as well as the versatility to prioritize either system lifetime or output quality at the user's request. Such flexibility and adaptability are essential characteristics of a microsensor node, a system that will be subjected to far more resource, workload, and input variability than most electronic devices. As continuing developments in VLSI technology reduce the size and increase the functionality of microsensor nodes, we foresee the power-aware design methodology as the dominant enabler for a practical, energy-efficient microsensor node.

ACKNOWLEDGMENTS

The authors would like to thank Travis Furrer for his dedicated efforts in adapting eCOS to run on the StrongARM SA-1100. R. Min is supported by a NDSEG fellowship. M. Bhardwaj is supported by an IBM Fellowship. A. Wang is supported by a Lucent Fellowship.

This effort is sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-00-2-0551.[1] The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

REFERENCES

[1] A. Chandrakasan et al., "Design Considerations for Distributed Microsensor Systems," *Proc. CICC 1999*, pp. 279-286.

[2] K. Bult et al., "Low Power Systems for Wireless Microsensors," *Proc. ISLPED 1996*, pp. 17-21.

[3] A. Sinha, A. Wang, and A. Chandrakasan, "Algorithmic Transforms for Efficient Energy Scalable Computation," Accepted for presentation at the *International Symposium on Low Power Electronics and Design (ISLPED)*, 2000.

[4] S. H. Nawab et al., "Approximate Signal Processing," *J. of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, Vol. 15, No. 1/2, Jan. 1997, pp. 177-200.

[5] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proc. HICSS 2000*, Jan. 2000.

[6] D. Estrin et al., "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proc. Mobicom 1999*, Aug. 1999, pp. 263-270.

[7] http://www-mtl.mit.edu/research/icsystems/uamps

[8] "eCos Documentation," *http://sourceware.cygnus.com/ecos/docs.html*

[9] A. Sinha and A. Chandrakasan, "Energy Aware Software," *Proc. Thirteenth International Conference on VLSI Design*, Jan. 2000, pp. 50-55.

[10] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design, A Systems Perspective*, 2nd edition, Reading, Mass.: Addison-Wesley, 1993, p. 236.

[11] G. Wei and M. Horowitz, "A Low Power Switching Supply for Self-Clocked Systems," *Proc. ISLPED 1996*, pp. 313-317.

[12] J. Goodman, A. Dancy, and A. Chandrakasan, "An Energy/Security Scalable Encryption Processor using an Embedded Variable Voltage DC/DC Converter," *Journal of Solid State Circuits*, Vol. 33, No. 11, Nov. 1998, pp. 1799-1809.

[13] K. Yao et. al., "Blind Beamforming on a Randomly Distributed Sensor Array System," *IEEE J. on Selected Topics in Communication*, Vol. 16, No. 8, Oct. 1998, pp. 1555-1567.

[14] S. Haykin, J. Litva, T.J. Shepherd, *Radar Array Processing*, Springer-Verlag, 1993.

[1] The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), the Air Force Research Laboratory, or the U.S. Government.