

Energy-Efficient DSPs for Wireless Sensor Networks

Alice Wang and Anantha Chandrakasan

Only in recent years, the challenge of ubiquitous sensing environments has become a technical possibility with growing research in wireless sensing networks. As the field of mobile computing and communication advances, so does the idea of a distributed, ad-hoc wireless network of hundreds to thousands of microsensors, which can be randomly scattered in the area of interest. A wireless communication network between sensors facilitates sensor collaboration, and a low-power digital signal processor (DSP) can do the analysis of sensor data locally. Networked microsensors enable a variety of new applications such as warehouse inventory tracking, location sensing, machine-mounted sensing, patient monitoring, and building climate control [1]-[3].

One prime example of a microsensor application is the use of acoustic sensors for environmental monitoring. Acoustic sensors are highly versatile and can be used in a variety of applications, such as speech recognition, traffic monitoring, and medical diagnosis. The sensor application that will be investigated in this article is source tracking and localization. Multiple sensors

can be used to pinpoint the location of an acoustic source (e.g., moving vehicle, speaker) by using a line of bearing estimation technique. Source localization can be useful for traffic monitoring, speech applications, and military exercises.

Large sensor arrays have long been used in civil and military applications to extend the field of view of the user. However, most current sensing systems consist of a few large macrosensors, which can be highly sensitive and expensive. Macrosensor systems are not fault tolerant; one faulty sensor can cause the entire system to fail. Networks of wireless microsensor nodes are becoming more popular for reasons such as lower cost, ease of deployment, and fault tolerance. Fig. 1 shows examples of various microsensor networks.

There are many new challenges to be faced in implementing signal processing algorithms and designing energy-efficient DSPs for microsensor networks. The first challenge is that all nodes are energy constrained. As the number of sensors increases, it becomes infeasible to recharge all of the batteries of the individual sensors. To prolong the lifetimes of the wireless sensors, all aspects of

the sensor system should be energy efficient, and design should focus on minimizing both computational and communication energy (see “Power Dissipation”). A second challenge is the high node densities, which results from a large number of microsensors within the network. The amount of sensing data will be tremendous, and it will be increasingly difficult to store and process the data. An efficient network protocol layer and signal processing application are needed to extract the important information from the sensor data (see “Storage”).

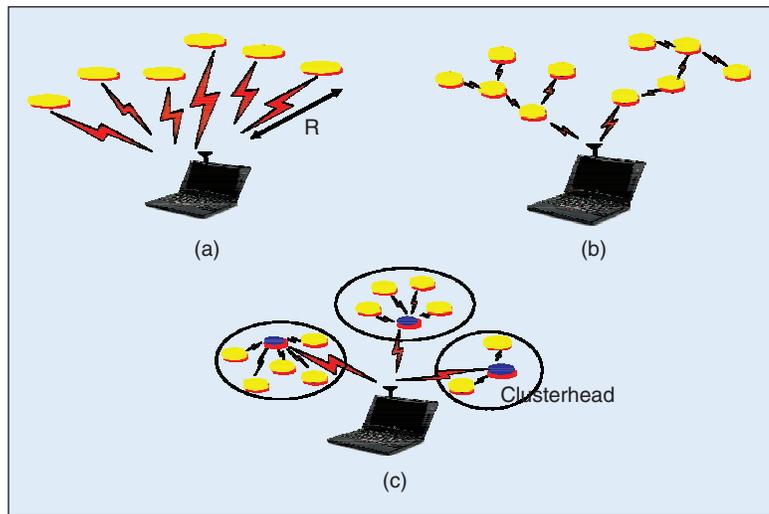
In this work, we will study system partitioning of computation to improve the energy efficiency of a wireless sensor networking application. We will explore system partitioning between the sensor cluster and the base station, employing computation-communication tradeoffs to reduce energy dissipation. Also we will show that system partitioning of computation within the cluster can also improve energy efficiency by using dynamic voltage scaling (DVS).

Wireless Sensor Node Architecture

Energy awareness and energy efficiency are two desirable characteristics of wireless sensor networks. The node architecture must be flexible so that the nodes can adapt to differing roles. For example, a node can act as a data gatherer, a relay, or a clusterhead. Also the node architecture must be adaptable so that nodes can be energy aware of varying conditions (e.g., changing signal statistics). An energy-aware node should be able to adapt energy consumption as energy resources of the system diminish or as performance requirements change. Energy awareness will lead to longer node lifetimes and more efficient sensor systems. Another desirable characteristic of wireless sensor networks is that all nodes are homogeneous and will have the same architecture.

At the heart of energy efficient sensor networks is the design of a low power wireless sensor node. An energy-efficient node architecture is being developed as part of the low power wireless sensor project at MIT (μ AMPS) [6]. Fig. 2 shows the architectural overview of a sensor node, which contains four basic modules: sensor, data and control processing, communication, and power. The sensor module contains the A/D converter and various sensors including acoustic and seismic sensors. The acoustic sensor is used for the application of source tracking and classification, and a 1 kHz A/D sampling rate is required. The data is continuously sampled and stored in on-board RAM to be processed by the data and control processing module.

The other three modules are adaptable depending on the role of the sensor within the network, and all three are controlled by the data and control processing module.



▲ 1. Three examples of microsensor networking protocols. In (a), the microsensors are using direct communication with end-user. In (b), the sensors transmit their data using multihop routing communication with base station. (c) The third protocol is a clustering algorithm. Sensors are grouped into clusters, and data is transmitted from sensors to clusterheads. Clusterheads perform data aggregation and transmit the result to the base station. The dark nodes represent clusterheads. (a) Direct communication, (b) multihop routing, and (c) clustering protocol.

For the source tracking application, we will assume that the sensor node can have one of three roles: data gatherer, clusterhead, and data relay. For example, if the node is a data gatherer, then the sensor data is passed from the sensor module to the communication module and then is transmitted. If the node is a clusterhead where signal processing is done, then the node will collect sensor data from neighboring sensors and perform the data aggrega-

Power Dissipation

Small sensor nodes imply limited physical space for batteries, and high density implies that periodic battery replacement will be a great inconvenience—and more likely, impossible. A state-of-the-art lithium primary battery offers an energy density of about 2 kJ per cm^3 [4]. Assuming that 1 cm^3 is available within the node for the battery and that the desired device lifetime is one year, the average power dissipation must be less than

$$(2000\text{J}) \left(\frac{1 \text{ year}}{365 \text{ days}} \right) \left(\frac{1 \text{ day}}{24 \text{ hrs}} \right) \left(\frac{1 \text{ hr}}{3600 \text{ s}} \right) = 63.4 \mu\text{W}.$$

As this value exceeds the standby power of most digital systems, energy dissipation is of paramount concern. Moreover, Moore’s law simply does not apply to batteries: the energy density of batteries has only doubled every five to 20 years, depending on the particular chemistry, and prolonged refinement of any chemistry yields diminishing returns [5]. Energy conservation strategies are therefore essential for achieving the lifetimes necessary for viable applications.

Storage

The amount of storage needed for a sensor network that is continually on will grow exponentially with the resolution of sensing required. This problem can be solved with efficient networking protocols, and local signal processing is needed.

| | Temperature Sensor | Acoustic Sensor | Image Sensor |
|--|--------------------|-----------------|---------------------------------|
| Sampling frequency | 1 Hz | 1 kHz | 20 frames/s 25k pixels/frame |
| Bit rate | 7 b/sample | 12 b/sample | 8 b/pixel |
| Amount of sensing data from one hour per sensor | 25 kb | 43 Mb | 14 Gb |
| Amount of sensing data from one hour for entire network (100-1000 sensors) | 2.5-25 Mb | 4.3-43 Gb | 1.4-14 terebit |

tion algorithm. A data-aggregation algorithm combines multiple sensor signals into one signal. A data-relay node receives data from neighboring sensors and transmits the data to neighboring sensors or the end-user.

Examples of networking protocols that assign such roles to microsensors are direct communication, multihop routing, and clustering. In direct communication, as shown in Fig 1(a), all microsensors are data gatherers and transmit their results directly to the base station. In multihop routing, sensors act as routers for other sensor's data in addition to sensing the environment, as shown in Fig. 1(b). Multihop routing minimizes the distance an individual sensor must transmit its data and hence minimizes the dissipated energy for that sensor. However, this approach is not globally energy efficient. An energy-efficient networking protocol organizes sensors into local clusters, as shown in Fig. 1(c). Each cluster has a *clusterhead*, a sensor that receives data from all other sensors in the cluster, performs data fusion (e.g., beamforming), and transmits the results to the end-user.

This greatly reduces the amount of data that is sent to the end-user and thus achieves energy efficiency.

The central component of the data and control processing subsystem is the StrongARM SA-1100 microprocessor. The SA-1100 is selected for its low power consumption, sufficient performance for signal processing algorithms, and static CMOS design. In addition, the SA-1100 can be programmed to run at a range of clock speeds from 74-206 MHz and at voltage supplies from 0.85-1.44 V [7]. On-board ROM and RAM are included for storage of sampled and processed data, signal processing tasks, and the operating system.

A simple energy model can be used to model the active energy dissipation of the SA-1100 as a function of supply voltage

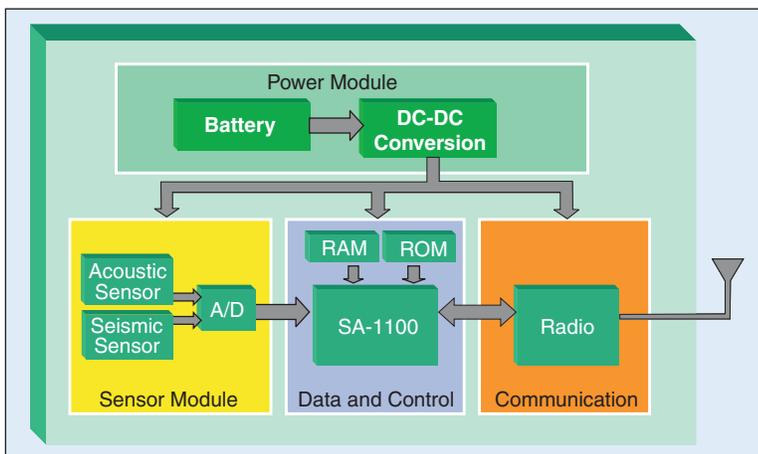
$$E_{\text{comp}} = NCV_{dd}^2 \quad (1)$$

where N is the number of clock cycles per task, C is the average capacitance switched per cycle, and V_{dd} is the supply voltage [8]. For the StrongARM SA-1100, experiments show that C is approximately 0.67 nF [9]. Another component of processor energy dissipated is the leakage energy, which is caused by subthreshold leakage currents between power and ground

$$E_{\text{leak}}(V_{dd}, f) = V_{dd} \left(I_o e^{\frac{V_{dd}}{nV_T}} \right) \left(\frac{N}{f} \right) \quad (2)$$

Most processor energy models only consider switching energy, but in systems that have low duty cycles, leakage energy dissipation can become large [9].

Also it is important to model the clock speed of the SA-1100 as a function of V_{dd}



▲ 2. Architectural overview of μ AMPS sensor node. The node can be broken down into four modules: sensor, power, data, and control and communication.

$$f \leq \frac{K(V_{dd} - V_{th})^a}{V_{dd}} \approx K(V_{dd} - c) \quad (3)$$

where a , K , c and V_{th} are processor dependent variables. Note that for a given processor, the maximum performance (f) of the processor is determined by the power supply voltage (V_{dd}), and vice versa. For minimal energy dissipation, a processor should operate at the lowest voltage for a given clock frequency. For the SA-1100 the frequency-voltage relation can be linearized to simplify the calculations, and experiments show that $K=239.28$ MHz/V and $c=0.5$ V. The frequency-voltage relation for the SA-1100 is shown in Fig. 4(b).

To collaborate with neighboring sensors and with the end-user, the data from the data and control module is passed to the radio or communication module. The primary component of the radio is a commercial transceiver optimized for ISM 2.45 GHz wireless systems. The radio module is capable of transmitting up to 1 Mb/s at a range of up to 10 meters [10].

An energy model for the communication module has also been developed to model the energy dissipated by a sensor node when transmitting and receiving data [11]. The radio module energy dissipation can be characterized into two types. The first is given by E_{elec} (J/b), the energy dissipated to run the transmit or receive electronics and the second is given by ϵ_{amp} (J/b/m²), the energy dissipated by the transmit power amplifier to achieve an acceptable E_b / N_o at the receiver. We assume an d^2 energy loss for transmission between sensors since the distances between sensors are relatively short [12]. To transmit a k -bit packet a distance, d , the energy dissipated is

$$E_{tx}(k, d) = E_{elec} \cdot k + \epsilon_{amp} \cdot k \cdot d^2 \quad (4)$$

and to receive the k -bit packet, the radio expends

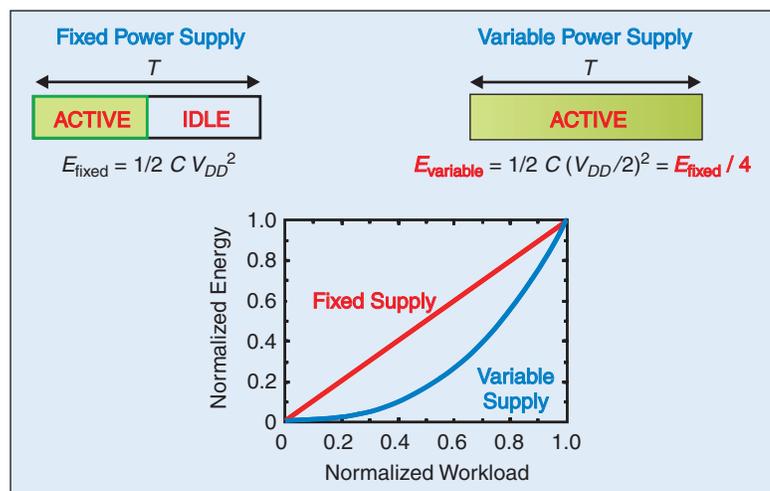
$$E_{rx}(k) = E_{elec} \cdot k. \quad (5)$$

For our radio, we use the parameters $E_{elec} = 50$ nJ/b and $\epsilon_{amp} = 100$ pJ/b/m².

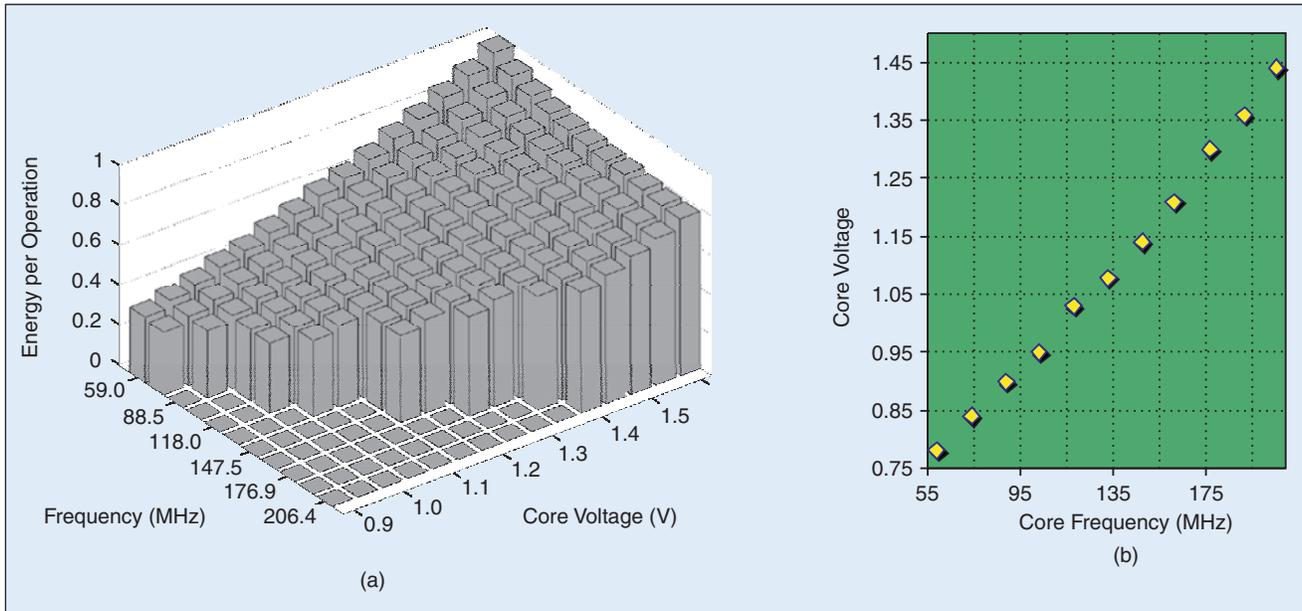
The fourth module is the battery or power module that supplies the variable power for the node. The power for the node is supplied by a single 3.6 V dc source, which can be provided by a single lithium-ion cell or three NiCD or NiMH cells. Regulators are used to generate 5 V, 3.3 V, and an adjustable 0.9-1.6 V supplies from the battery. The 5 V supply powers the analog sensor circuitry and A/D converter. The 3.3 V supply powers all digital components on the sensor node with the exception of the processor core. A digitally adjustable switching regulator that can provide 0.9 V to 1.6 V in 20 discrete increments powers the StrongARM SA-1100 core. Having a digitally adjustable voltage supply allows the SA-1100 to control its own core voltage and enables DVS techniques [13].

DVS is one technique that can be used to create an energy-scalable sensor system, which is able to adapt energy dissipation with changing operating conditions. In an energy-scalable system, where there is a variable computational load, the sensor system should be able to scale the amount of energy dissipated as the computational load changes. In typical systems, design is done for the worst-case scenario. For energy-scalable systems, this type of design may not be globally optimal for energy dissipation. For example, assume in a fixed throughput system (T), E_{fixed} is the processor computational energy for task A . Notice that the task is finished in half the allotted time, $T/2$, as shown in Fig. 3. This is nonoptimal since after the processor completes the task, it will idle for $T/2$ seconds. This task can be completed with less energy dissipated by reducing the clock frequency by half, so that the processor is active for the entire T seconds, and consequently we can reduce the voltage supply by half. According to (1), energy is related to V_{dd} squared, so that by reducing the voltage supply by one-half causes the energy dissipated for a variable voltage supply (E_{var}) to drop down to one-fourth that of the fixed voltage supply case. Fig. 3 also shows a graph comparing energy for a fixed voltage supply and energy for a variable voltage supply as the computational workload changes. This graph shows the quadratic relationship between energy and computation when using a variable voltage scheme. Having a variable voltage supply will enable energy-scalable implementation of signal processing algorithms.

Fig. 4(a) depicts the measured energy consumption of an SA-1100 processor running at full utilization. Energy consumed per operation is plotted with respect to the processor frequency and voltage. This figure shows the quadratic dependence of switching energy on supply voltage. Also for a fixed voltage, the leakage energy per operation increases processor energy dissipation, since



▲ 3. As computational workload varies, variable power supply gives quadratic energy savings. For a fixed latency requirement of T , if the computational load is one-half of the worst case load, then the processor will be idle for $T/2$. If a variable power supply is used, the frequency and voltage supply can be halved dropping the energy dissipated down to a quarter of the fixed power supply case.



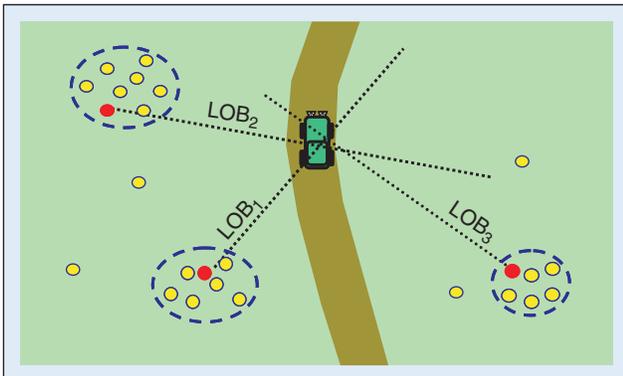
▲ 4. (a) This data shows the measured energy consumption characteristics of SA-1100, as V_{dd} is varied between 0.85-1.44V and as clock frequency is varied between 74-206 MHz. For minimal energy dissipation for a given clock frequency, the SA-1100 is operated at the minimum voltage supply. These operating voltage and frequency pairs are given in (b).

operations occur over a longer clock period. This graph shows that the optimal V_{dd} and f points of operation is to operate at the lowest possible voltage supply level for a given frequency. Fig. 4(b) shows all 11 frequency-voltage pairs for the StrongARM SA-1100 for energy-efficient operation.

Sensor Application Case Study

One application for wireless acoustic sensors is vehicle tracking and localization. In this section, we will introduce algorithms that would be running locally at the acoustic sensor cluster and show how system partitioning can yield a more energy-efficient sensor system.

Suppose a vehicle is moving over a region where a network of acoustic sensing nodes has been deployed. To determine the location of the vehicle, we first need to find the line of bearing (LOB) or direction from which sound is being detected. Fig. 5 shows the scenario for vehicle tracking using LOB estimation. Multiple clusters of sen-



▲ 5. Line of bearing estimates from multiple sensor clusters can be triangulated to determine the vehicle's location.

sors determine the source's LOB to be the direction with maximum sound energy from their perspective. The intersection point of multiple LOBs will determine the source's location.

To perform LOB estimation, often beamforming algorithms are used. A beamformer is a spatial filter that operates on multiple sensor data to enhance the amplitude of a desired coherent waveform and to diminish the effects of background noise on the desired signal. By pointing the beam in the direction of the source, the signal in the desired direction is amplified while ambient noise from all other directions are diminished. Delay-and-sum beamforming is a conventional beamforming algorithm, which applies delays on the multiple sensor data before summing over all sensors [14]. Beamforming of sensor data is beneficial in two ways. First, by scanning over multiple directions, the direction of arrival of the signal with the most signal energy relative to the orientation of the sensor cluster can be found. This means that the direction of arrival of sound is correlated to the LOB of the source signal. Second, the beamformer output in the direction of arrival will have better SNR since the effect of the ambient noise sources have been reduced and the desired signal is coherently added. Other beamforming algorithms include the maximum power beamforming algorithm [15] and the least mean square algorithm [16].

Traditionally, delay-and-sum beamforming has been implemented in the time domain, frequency domain, or a hybrid of the two. In the time domain, for a given direction of arrival, first the time delays between multiple sensor data are estimated. Then the sensor data is shifted so that they are aligned in time, before summing over all sensors. One drawback of implementing the time-domain delay-and-sum beamforming in the digital domain is that

it is necessary to oversample the signal or interpolate the sensor data to get the fine-grained delays necessary for good beamformer performance. An acoustic sensor output, which is nominally sampled at the Nyquist frequency of 1 kHz, may have to be oversampled at 5-10 kHz. This can lead to high computational demand depending on the length of the interpolation filters and desired performance [14]. A more energy-efficient method is to apply interpolated sinc functions in an FIR filter beamformer approach [17].

Delay-and-sum beamforming can also be performed in the frequency domain, where delays in the time domain correspond to phase shifts in the Fourier or frequency domain. First the sensor data is transformed into the frequency domain. In the digital domain, the discrete Fourier transform or the fast Fourier transform (FFT) can be used [19]. Then phase shifts are applied before summing over all sensors. The main advantage of a frequency-domain implementation is that there is no need for oversampling the sensor output, as compared to the time-domain approach. For this application a 1024-pt. FFT is used.

The output of the delay-and-sum beamformer is then fed into an LOB estimator. A simple LOB estimator takes the beamformer output and calculates the signal energy for each direction. The maximum weighted average of signal energy over all directions is the LOB estimate for the signal.

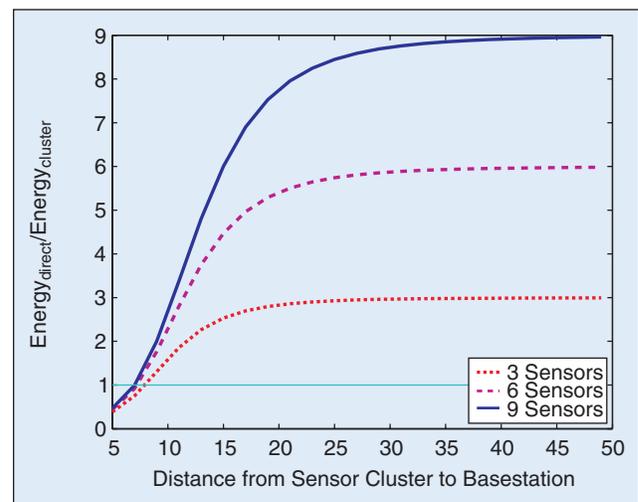
Energy-Efficient System Partitioning Between End-User and Sensors

In current wired sensor systems, all sensor nodes take on the role of data gatherer and the data is sent to the end-user to be processed because there is no energy constraint on wired sensor systems. For a wireless sensor network, direct communication between sensors and the end-user is both cumbersome and highly energy intensive for a variety of reasons. One reason is that the end-user is usually far away from the sensing area, and therefore communication of raw sensor data to the end-user can be quite costly. Another reason is that as the number of sensors in a network grows larger and larger, it becomes difficult to manage the large amount of data collected from the sensors. Also, with increased node densities in one location, multiple sensors may view the same event so there is a lot of redundant sensor data. Fig 1(a) shows a wireless sensor network that uses a direct communication networking protocol.

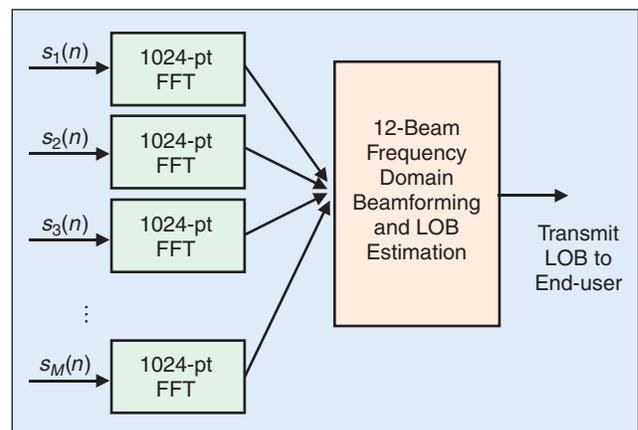
Research into energy-efficient wireless networking protocols for sensor networks has shown that energy dissipation can be reduced through both sensor collaboration and intelligent communication and computation energy tradeoff between the end-user and the sensors [21]. Sensor collaboration means that the sensors can communicate locally and share information. Since closely located sensors tend to have highly correlated data, sensor collaboration allows for signal processing of the sensor data (e.g., beamforming) to reduce redundant information. Also it is important to compare communication and

computation energy dissipated for a given application. Commercial radios typically dissipate ~ 150 nJ/bit [22]-[24] versus the StrongARM, which dissipates 1 nJ/bit [25]. Therefore communication is typically more expensive than computation done on a general purpose processor, allowing one to perform 150 instructions per bit communicated. In comparison, application-specific hardware can achieve up to three orders of magnitude lower energy than a general purpose processor. By using application specific hardware, millions of bits of operations can be performed per bit communicated.

An energy-efficient network protocol, such as clustering protocols, can take advantage of this asymmetry between communication and computation energy to greatly reduce the energy dissipation in wireless sensor systems. Using low energy adaptive clustering hierarchy (LEACH) [21], an energy-efficient clustering protocol, the sensors are organized into local clusters, as shown in Fig. 1(c). Each cluster has a clusterhead, a sensor that receives data



▲ 6. For a remote base station ($d > 10$ m) communication energy dominates over computation energy, and it becomes more energy efficient to perform data aggregation (beamforming) locally at the sensor cluster, rather than direct communication of the raw sensor data.



▲ 7. Signal processing blocks of the LOB estimation algorithm includes a 1024-pt. FFT of the sensor data, frequency domain beamforming, and a LOB estimator.

from all other sensors in the cluster, performs data fusion (e.g., beamforming) and transmits the results to the end-user. This greatly reduces the amount of data that is sent to the end-user and thus achieves energy efficiency.

It is useful to compare the energy dissipated for the two network protocols mentioned, direct communication and LEACH. Fig. 6 compares the amount of energy required to communicate information from an M sensor cluster ($M=3,6,9$) and to transmit the result to the end-user. The ratio of energy for direct communication (E_{direct}) versus energy for a clustering approach with delay-and-sum beamforming (E_{cluster}) is plotted as a function of distance from end-user to sensor cluster. When the distance to the end-user is large ($d > 10$ m), the communication energy dissipated transmitting data to the end-user dominates, and there is a large advantage to performing local signal processing. When performing LOB estimation, there is even more communication energy savings because the only data transmitted to the end-user is the LOB estimate.

Another advantage of doing local signal processing is that the computation and communication energy can be further reduced if the signal of interest has a limited band-

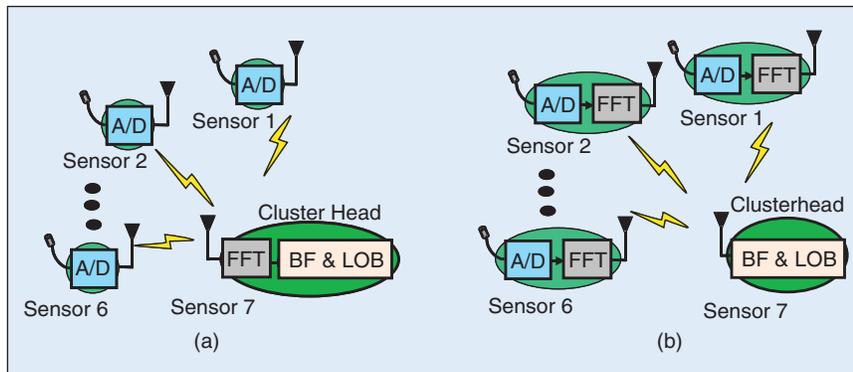
width. For our acoustic sensor application, the signal of interest has a bandwidth of $20 \text{ Hz} \leq f_{\text{source}} \leq 250 \text{ Hz}$. Therefore only approximately one-fourth of the computation is needed. However, this is highly application dependent.

Energy-Efficient System Partitioning Between Sensors and Cluster-Head

Within a sensor cluster, each sensor node has its own processor. We can take advantage of a network of multiple microprocessors to distribute the computation among the sensors. In this section we will show that intelligent system partitioning with dynamic voltage scaling can make our sensor network even more energy efficient.

We will demonstrate the effectiveness of system partitioning on the energy dissipation of the LOB estimation application. Fig. 7 is a block diagram that breaks down the computation involved in the LOB algorithm. The first part is to transform collected acoustic sensor data from each sensor into the frequency domain using a 1024-pt. FFT. Then, the FFT data is beamformed into 12 uniform directions. The direction of the signal with the most energy is the LOB of the source.

The LOB estimation algorithm can be implemented in two different ways. In the direct technique, each sensor has a set of acoustic data, $s_i(n)$. This data is transmitted to the clusterhead where all LOB estimation computation is done. This technique is demonstrated in Fig. 8(a). Alternatively, we can first perform the FFTs at each sensor and then send the FFT results to the clusterhead. This can be done since we assume a homogeneous sensor network where all sensors have the same architecture (shown in Fig. 2). This is the distributed method and is shown in Fig. 8(b). If we assume the processor models discussed previously, then performing the FFTs with



▲ 8. There are two ways to partition the computation for LOB estimation. (a) The first is the direct technique, where all of the computation (FFT and beamforming) is done at the clusterhead. (b) The second system partitioning scheme is the distributed technique. Here, the FFT is distributed among all sensors, and the FFT coefficients are transmitted to the clusterhead where beamforming is done.

| | | Direct | Distributed |
|-------------|----------|---------|-------------|
| Nodes | V_{dd} | — | 0.85 V |
| | f | — | 74 MHz |
| Clusterhead | V_{dd} | 1.44 V | 1.17 V |
| | f | 206 MHz | 162 MHz |
| | Latency | 19.2 ms | 18.4 ms |
| | Energy | 6.2 mJ | 3.4 mJ |

the distributed technique has no computational energy savings over the direct technique, because the same total amount of computation is being done. However, by having a DVS-enabled sensor node, the node can take advantage of the parallelized computational load by allowing voltage and frequency to be scaled while still meeting latency constraints.

In the DVS-enabled sensor node, there is a large advantage in having the computation distributed among the sensor nodes, since the voltage supply can be reduced. Table 1 shows the computation energy for a seven-sensor cluster. In the direct technique, with a computation latency constraint of 20 ms, all of the computation is performed at the clusterhead at the fastest clock speed, $f=206$ MHz at 1.44 V. The energy dissipated by the clusterhead processor is measured to be 6.2 mJ and the latency is 19.2 ms. In the

distributed technique, the FFT is parallelized to the sensor nodes. In this scheme, the sensor nodes sense data and perform the 1024-pt. FFTs on the data before transmitting the FFT data to the clusterhead. At the clusterhead, the beamforming and LOB estimation is done. Since the FFTs are parallelized, the clock speed and voltage of both the FFTs and the beamforming can be lowered. For example, if the FFTs at the sensor nodes are run at 0.85V voltage supply and 74 MHz clock speed while the beamforming algorithm is run at 1.17V and 162 MHz clock speed then with a latency of 18.4 ms, only 3.4 mJ energy is dissipated by all of the processors combined. This is a 45.2% improvement in energy dissipation. This example shows that energy-efficient system partitioning by parallelism in system design can yield large energy savings.

Increasing parallelism is a common technique used in circuit design to reduce energy dissipation when there is a fixed latency. For example, if computation C can be computed using two parallel functional units instead of one, then the throughput is increased by two. For a fixed throughput, however, the clock frequency is reduced to $f/2$, and voltage supply of $V_{dd}/2$, then the energy is reduced by four times over the nonparallel case. However, this does come at an increase of area and overhead control hardware. For a simple adder-comparator 8-b datapath, it was shown that the power was reduced by approximately 2.5 times [8].

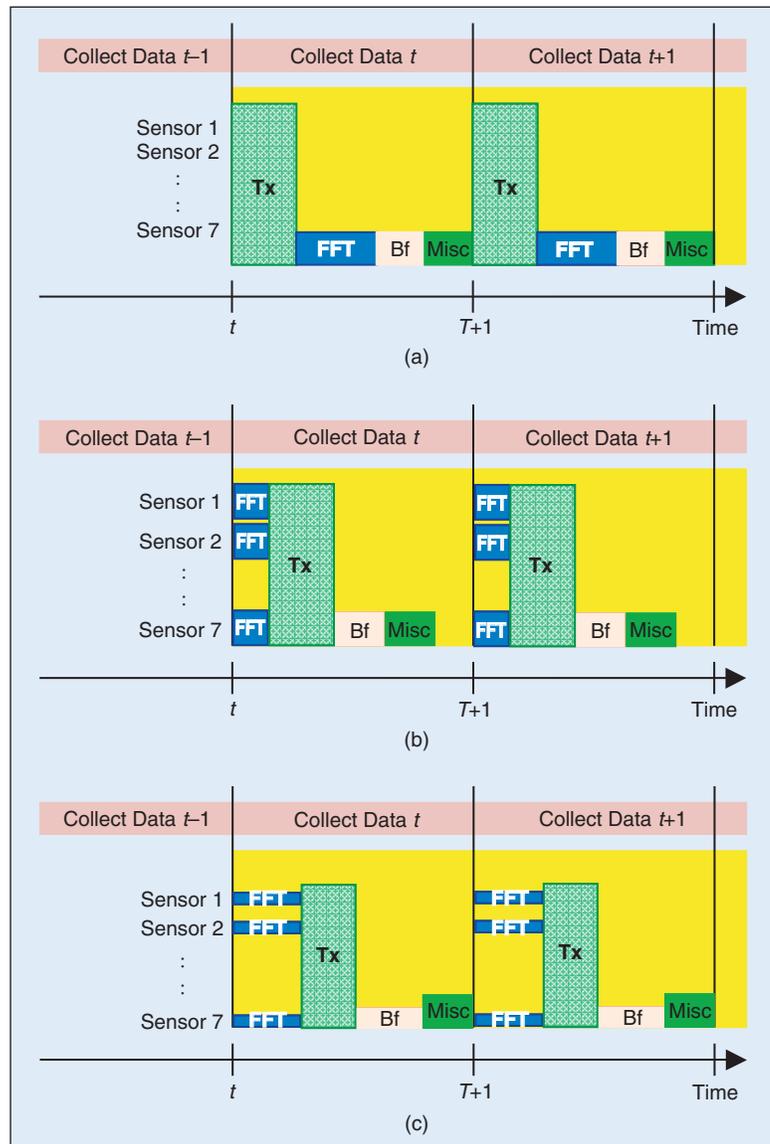
For the sensor cluster, the energy savings comes by distributing the computation to six-node processors that run in parallel versus one clusterhead processor running all tasks serially. By running the FFT task in parallel, *all processors* are able to operate at lower frequencies and consequently switch the capacitance at lower voltages. This leads to lowered energy dissipation. The computation energy in this example is measured using an experimental setup using the StrongARM evaluation board [9].

Fig. 9 shows the timing diagram for the three cases: direct technique, distributed technique without DVS, and distributed technique with DVS. For all three scenarios, at time t , the sensors begin sensing acoustic data from the microphones. After 1024 samples are collected, then the nodes can begin processing the data. Note that while the computation is being done, at the same time, new data is being collected from the microphones. To minimize buffer size, the LOB estimation computation and intersensor communication should be completed before the next 1024 samples have been collected.

In the direct technique timing diagram, the clusterhead sensor (S7) does all of the signal processing for LOB estimation. In the distrib-

uted technique without DVS, distributing the FFT to the sensors would help the throughput of the LOB estimate, but does not decrease the total computation energy dissipated. Since this is a fixed latency system, the processors would be idle for a long period. Therefore, by using DVS to reduce clock frequency and voltage supply, the processors would be active the entire period and energy dissipation is also reduced.

An additional bonus in distributing the FFT is the reduction of communication energy between sensors. Due to the nature of the sound source, the signal of interest has a bandwidth between 20 and 250 Hz. This means that after doing the 1024-pt. FFT, only 230 Fourier coefficients



▲ 9. Timing diagrams of three techniques: (a) direct technique, (b) distributed technique without DVS, and (c) distributed technique with DVS. In the direct technique, all of the computation is performed at the clusterhead serially. In the distributed technique without DVS, the FFT is distributed and performed in parallel, thereby improving throughput. In a fixed latency scheme, it is better to use the distributed technique in conjunction with DVS. By lowering the clock frequency and voltage supply, the throughput is fixed, but energy dissipated is reduced.

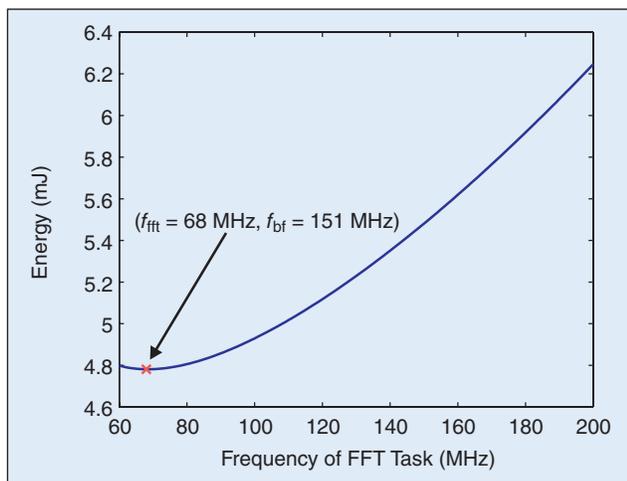
needs to be transmitted. This means that the communication energy for the distributed technique can be reduced by 50% over the direct technique case, where all 1024 samples are transmitted.

Optimal V_{dd} and Clock Frequency Scheduling

By distributing the computation across sensors and using DVS, we have shown that we can reduce the energy dissipated. The next step is to calculate the optimal V_{dd} and clock frequency for minimal computation energy dissipation. This is important because the system should adjust operating voltages and frequencies of the sensor nodes to changes in system parameters (e.g., number of sensors, number of samples, etc.). Also, the operating system has the ability to run different tasks (FFT, beamforming, etc.) at different V_{dd} and clock frequencies. However, the optimal V_{dd} and clock frequency should be chosen to minimize energy while making sure the computation is done within the latency constraint.

Finding the optimal operating points is not easy because the energy curve as a function of frequency and voltage is highly nonlinear. Fig. 10 shows the computational energy dissipated for the seven-sensor LOB estimation application for a fixed latency requirement of 20 ms given our energy and frequency models. The curve is plotted as a function of all possible $\langle f_{\text{fft}}, f_{\text{bf}} \rangle$ pairs. There is a minimum energy operating point at $f_{\text{fft}} = 68$ MHz and $f_{\text{bf}} = 151$ MHz, due to the tradeoff between energy and performance between the two tasks and also remaining within the latency requirement. To find a closed solution for the optimal operating voltage and frequency, we want to minimize the total computation energy for an M sensor cluster

$$E_{\text{comp}} = MN_{\text{fft}}CV_{\text{fft}}^2 + N_{\text{bf}}CV_{\text{bf}}^2 \quad (6)$$



▲ 10. Energy dissipated by a seven-sensor cluster for different $\langle f_{\text{fft}}, f_{\text{bf}} \rangle$ pairs for a latency constraint of 20 msec. The “x” indicates the optimal frequency pair for the FFT and BF tasks, which gives minimum energy dissipated.

with the latency constraint that

$$T_{\text{comp}} \geq \tau_{\text{fft}} + \tau_{\text{bf}} = \frac{N_{\text{fft}}}{f_{\text{fft}}} + \frac{N_{\text{bf}}}{f_{\text{bf}}}. \quad (7)$$

Here, V_{fft} and V_{bf} are the operating voltages of the two tasks, FFT and beamforming, and N_{fft} and N_{bf} are the cycle counts, respectively. In general, to minimize energy, we will want to reduce both voltage and clock frequency, but the frequency must be large enough to satisfy the latency constraint in (7).

To find the optimal voltage and frequency operating points, first relate (7) in terms of voltage supply by using the processor frequency equation (3) and solve a Lagrangian minimization problem to get the relation between V_{bf} and V_{fft}

$$(V_{\text{bf}} + c) = \sqrt[3]{M}(V_{\text{fft}} + c). \quad (8)$$

Equation (8) is substituted back into (7) and then we solve for V_{fft} , V_{bf} , f_{fft} , and f_{bf}

$$V_{\text{fft}} \geq \frac{1}{T_{\text{comp}}K} \left[N_{\text{fft}} + \frac{N_{\text{bf}}}{\sqrt[3]{M}} \right] + c \quad (9)$$

$$V_{\text{bf}} \geq \frac{\sqrt[3]{M}}{T_{\text{comp}}K} \left[N_{\text{fft}} + \frac{N_{\text{bf}}}{\sqrt[3]{M}} \right] + c \quad (10)$$

$$f_{\text{fft}} = \frac{(\sqrt[3]{MN_{\text{fft}}} + N_{\text{bf}})}{\sqrt[3]{MT_{\text{comp}}}} \quad (11)$$

$$f_{\text{bf}} = \frac{(\sqrt[3]{MN_{\text{fft}}} + N_{\text{bf}})}{T_{\text{comp}}}. \quad (12)$$

These equations suggest that in general it is desirable to run the parallelized task (FFT) at lower voltage and frequencies than that of the nonparallelized task (beamforming).

To verify this on the SA-1100, the FFT and beamforming algorithms were run on the SA-1100:

$$N_{\text{fft}} = 200.73 \text{ kcycles} \quad (13)$$

$$N_{\text{bf}} = 319.3M + 341.6 \text{ kcycles}. \quad (14)$$

This analysis assumes that the processor can run at continuous voltage and frequency levels. However, implementing this on processors that do not have continuous voltage and frequency levels, such as the SA-1100, is also possible. First, there is a constraint that T_{comp} fall between the range of

$$\frac{N_{\text{fft}} + N_{\text{bf}}}{f_{\text{max}}} \leq T_{\text{comp}} \leq \frac{N_{\text{fft}} + N_{\text{bf}}}{f_{\text{min}}} \quad (15)$$

where f_{\max} and f_{\min} are the maximum and minimum frequencies possible for the processor. If T_{comp} falls below the lower limit, there is never enough time to complete the computation, and if T_{comp} is above the upper limit, minimizing energy dissipation means always operating at the lowest frequency and lowest voltage levels.

Fig. 11 shows energy dissipated for the direct technique versus distributed technique with optimal voltage scheduling as M is increased from 3-10. Between 30-65% energy reduction can be achieved with the distributed techniques with optimal voltage scheduling.

This partitioning scheme can be generalized to any application where parallelism can be exploited to distribute computation among multiple processors. Let us assume there are two tasks, A and B , each of which can be characterized by their processor specific cycle counts, N_A and N_B , respectively. A is the task to be parallelized to the M processors and B is the nonparallelized task. Also assume that the bandwidth between the processor does not change for the two schemes. In the direct scheme, the frequency of the processor for each task is set to

$$f_A = f_B = \frac{MN_A + N_B}{T_{\text{comp}}} \quad (16)$$

For the distributed scheme, the voltages and frequencies are given in (12)-(15)

The ratio of E_{direct} , the energy of the direct technique to $E_{\text{distributed}}$, the energy of the optimal distributed technique, is calculated to approximately

$$\frac{E_{\text{direct}}}{E_{\text{distributed}}} = \frac{(M \cdot N_A/N_B + 1)^3 + D(M \cdot N_A/N_B + 1)^2}{M(N_A/N_B + M^{-1/2})^3 + D \cdot M(N_A/N_B + M^{-1/2})^3 (N_A/N_B + M^{-2/2})} \quad (17)$$

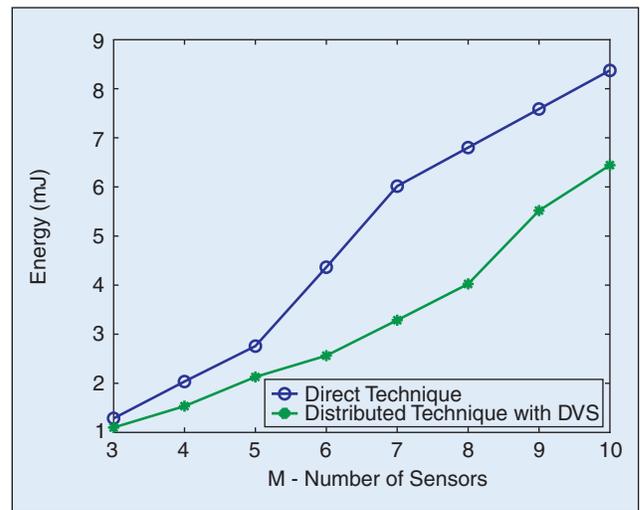
where $D=2cKT_{\text{comp}}/N_B$. This ratio shows that as the number of processors (M) increases, there is a large opportunity for energy savings because there is more parallelism. Fig. 12 shows a plot of the energy ratio, $E_{\text{direct}}/E_{\text{distributed}}$, as a function of cycle ratio, N_A/N_B , for $M=(5,7,9)$. This shows that when the computation for task A (parallelized task) is relatively large compared to task B (e.g., $N_A/N_B=10$), then there is a great deal of energy savings. Even when the computation in task A is small compared to that for task B (e.g., $N_A/N_B=0.1$), however, there is still a possibility for energy savings of $2x$. The upper limit of energy savings is when we take the limit of (17) as N_A goes to infinity, then the limit on energy savings is M^2 . This shows that for the general case, there is a potential for a great deal of energy savings.

Summary

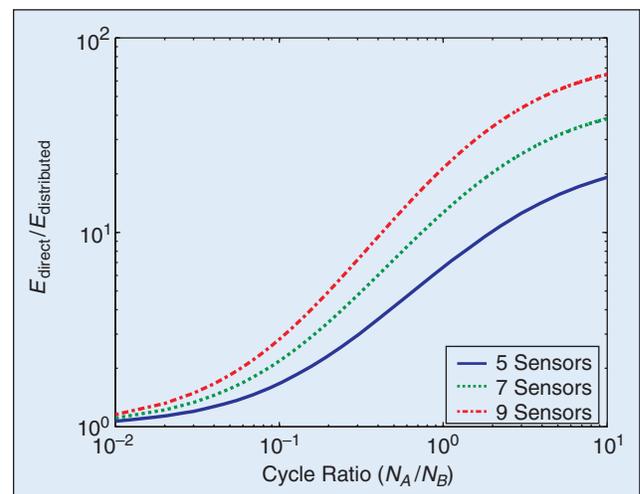
In this article we introduced two techniques for implementation of energy-efficient signal processing algorithms for

wireless sensor networks. One technique exploits communication versus computation energy tradeoffs. Clustering the sensors and performing signal processing on the multiple sensor data reduce communication of redundant information reduced at the expense of processor energy. We show that as the distance to the end-user increases and as processor energy is much cheaper than communication energy, it becomes more energy efficient to perform signal processing locally at the sensor node.

The second technique is to do efficient system partitioning of computation among sensor nodes. By parallelizing the computation, energy reductions of up to 65% can be achieved in a source localization application. Also introduced is the closed-form expression for the optimal voltage supply and clock frequency that minimizes the processor energy dissipation. Using measurements from the StrongARM SA-1100 this technique is verified for a LOB estimation algorithm. Also the general case is discussed.



▲ 11. The measured computational energy dissipated for the direct technique is higher than that for the distributed technique with DVS.



▲ 12. As the cycle ratio increases, energy savings increase for an $M=5,7,9$ sensor cluster.

Acknowledgments

We would like to thank R. Min, M. Bhardwaj, A. Sinha, and W. Heinzelman for their valuable contributions, discussions, and insight. We also would like to acknowledge N. Srour, T. Pham, K. Tran, and A. Nguyen from the Acoustic Signal Processing Branch at ARL for providing us with the algorithms for LOB estimation. We would also like to thank the reviewers for their helpful comments and suggestions. A. Wang is supported by the Lucent Fellowship.

This research is sponsored by the Defense Advanced Research Project Agency (DARPA) Power Aware Computing/Communication Program and the Air Force Materiel Command, USAF, under agreement F30602-00-2-0551; and by the Army Research Laboratory (ARL) Collaborative Technology Alliance, through BAE systems, Inc. subcontract RK7854.

Alice Wang received her S.B. and M.Eng. degrees in electrical engineering from the Massachusetts Institute of Technology (MIT) in 1997 and 1998, respectively. She is currently working toward her Ph.D. in the design of low power signal processors for wireless sensors at MIT. Her research interests include energy-efficient signal processing algorithms and low power DSPs for wireless systems. She is a Lucent Technologies Fellow.

Anantha P. Chandrakasan received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1989, 1990, and 1994, respectively. Since 1994, he has been with the Massachusetts Institute of Technology, Cambridge, and is currently an Associate Professor of Electrical Engineering and Computer Science. He held the Analog Devices Career Development Chair from 1994 to 1997. He received the NSF Career Development award in 1995, the IBM Faculty Development award in 1995, the National Semiconductor Faculty Development award in 1996 and 1997 and several best paper awards including the 1999 Design Automation Conference Design Contest Award. He is a co-author of several books. He has served on the technical program committee of various conferences. Currently, he is the Technical Program Chair for ISSCC'03 and an Associate Editor for the *IEEE Journal of Solid-State Circuits*. He is also a member of the Design and Implementation of Signal Processing Systems (DISPS) Technical Committee of the Signal Processing Society.

References

[1] J. Karn, R. Katz, and K. Pister, "Next century challenges: Mobile networking for smart dust," in *Proc. ACM MobiCom'99*, Aug. 1999, pp. 271-28.

[2] G. Asada, M. Dong, T.S. Lin, F. Newberg, G. Pottie, and W.J. Kaiser, "Wireless integrated network sensors: Low power systems on a chip," in *Proc. ESSCIRC'98*, 1998, pp. 9-16.

[3] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proc. ACM MobiCom'99*, Aug. 1999, pp. 263-270.

[4] R. Powers, "Advances and trends in primary and small secondary batteries," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 9, pp. 32-36, Apr. 1994.

[5] R. Hahn and H. Reichl, "Batteries and power supplies for wearable and ubiquitous computing," in *Proc. 3rd Int. Symp. Wearable Computers*, 1999, pp. 168-169.

[6] R. Min, M. Bhardwaj, et al., "An architecture for a power-aware distributed microsensor node," in *Proc. IEEE Workshop Signal Processing Systems (SiPS '00)*, Oct. 2000, pp. 581-590.

[7] <http://developer.intel.com/design/strong/>

[8] A. Chandrakasan and R. Brodersen, *Low Power Digital CMOS Design*. Norwell, MA: Kluwer, 1995.

[9] A. Sinha and A. Chandrakasan, "Energy aware software," in *Proc. VLSI Design 2000*, Calcutta, India, Jan. 2000.

[10] National Semiconductor Corporation, LMX 3162 Evaluation Notes and Datasheet, Apr. 1999.

[11] A. Wang, W. R. Heinzelman, A. Sinha, and A. Chandrakasan, "Energy-scalable protocols for battery-operated microsensor networks," *J. VLSI Signal Processing*, pp. 223-237, Nov. 2001.

[12] M. Ettus, "System capacity, latency and power consumption in multihop-routed SS-CDMA wireless networks," in *Proc. Radio and Wireless Conference (RAWCON '98)*, Aug. 1998, pp. 55-58.

[13] V. Gutnik and A. Chandrakasan, "Embedded power supply for low-power DSP," *IEEE Trans. VLSI Syst.*, vol. 12, pp. 425-435, Dec. 1997.

[14] R.A. Mucci, "A comparison of efficient beamforming algorithms," *IEEE Trans Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 548-558, June 1984.

[15] K. Yao, R.E. Hudson, C.W. Reed, C. Daching, and F. Lorenzelli, "Blind Beamforming on a Randomly Distributed Sensor Array System," *IEEE J. Selected Areas Commun.*, vol. 16, no. 8, Oct. 1998, pp. 1555-1567.

[16] S. Haykin, J. Litva, and T.J. Shepherd, *Radar Array Processing*. New York: Springer-Verlag, 1993.

[17] R.A. Gabel and R.R. Kurth, "Synthesis of efficient digital beamformers," *J. Acoust. Soc. Amer.*, vol. 70, no. S1, p. 517, 1981.

[18] P. Rudnick, "Digital beamforming in the frequency domain," *J. Acoust. Soc. Amer.*, vol. 46, no. 5, pp. 1089-1090, 1969.

[19] A. Oppenheim and R. Schaffer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[20] A. Sinha and A. Chandrakasan, "JouleTrack - A Web based tool for software energy profiling," in *Proc. 38th Design Automation Conf.*, June 2001, pp. 220-225.

[21] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. HICSS 2000*, Jan. 2000, pp. 3005-3014.

[22] L. Nord and J. Haartsen, *The Bluetooth Radio Specification and the Bluetooth Baseband Specification*, Bluetooth, 1999-2000.

[23] H. Darabi, S. Khorram, H.M Chien, M.A. Pan, S. Wu, S. Moloudi, J.C. Leete, J.J. Rael, M. Syed, B. Ibrahim, M. Rofougaran, and A. Rofougaran, "A 2.4 GHz CMOS Transceiver for Bluetooth," *IEEE J. Solid-State Circuits*, vol. 36, pp. 2016-2024, Dec. 2001.

[24] F.O. Eynde, J.J. Schmit, V. Charlier, R. Alexandre, C. Sturman, K. Coffin, B. Mollekens, J. Craninckx, S. Terrijn, A. Monterastelli, S. Beerens, P. Goetschalckx, M. Ingels, D. Joos, S. Guncer, and A. Pontoglou, "A fully integrated single chip SOC for Bluetooth," in *Dig. Tech. Papers 2001 IEEE Solid-State Circuits Conf.*, 2001, pp. 196-197.

[25] R. Amirtharajah, S. Meninger, J.O. Mur-Miranda, A. Chandrakasan, and J. Lang., "A micropower programmable DSP powered using a MEMS based vibratio-to-electric energy converter, in *Dig. Tech. Papers 2000 IEEE Solid-State Circuits Conf.*, 2000, pp. 362-363.