Hibernets: Energy-Efficient Sensor Networks Using Analog Signal Processing

Brandon Rumberg, *Student Member, IEEE*, David W. Graham, *Member, IEEE*, Vinod Kulathumani, and Robert Fernandez, *Student Member, IEEE*

***Abstract*—Preprocessing of data before transmission is recommended for many sensor network applications to reduce communication and improve energy efficiency. However, constraints on memory, speed, and energy currently limit the processing capabilities within a sensor network. In this paper, we describe how ultra-low-power analog circuitry can be integrated with sensor nodes to create energy-efficient sensor networks. To demonstrate this concept, we present a custom analog front-end which performs spectral analysis at a fraction of the power used by a digital counterpart. Furthermore, we show that the front-end can be combined with existing sensor nodes to 1) selectively wake up the mote based upon spectral content of the signal, thus increasing battery life without missing interesting events, and to 2) achieve low-power signal analysis using an analog spectral decomposition block, freeing up digital computation resources for higher-level analysis. Experiments in the context of vehicle classification show improved performance for our ASP-interfaced mote over an all-digital implementation.**

***Index Terms*—Analog signal processing, energy-efficient, in-network processing, sensor networks.**

I. Introduction

WIRELESS sensor networks (WSNs) hold great promise for use in applications such as environmental monitoring, protection of borders/resources against intruders, and monitoring critical infrastructure like bridges and power grids [1]–[6]. However, wide-scale deployment of sensor networks for these applications has been inhibited primarily by the inability to last for long durations on small power sources, such as batteries and energy-harvesting systems.

One strategy to conserve energy locally is to perform minimal computation at each node while transmitting most of the data, thereby leaving a majority of the computation and any necessary decision-making to one or more centralized units. However, this strategy leads to increased communication overhead. Therefore, local processing and in-network aggregation are recommended for reducing power consumption due to the high cost of communication. However, the amount of processing that a node can perform is restricted by both its power budget and its limited processing resources. These constraints restrict the amount of signal processing that the node can perform and also limit the highest sampling frequency at which processing can be sustained. In order to perform more advanced signal processing and work with higher-frequency signals, often a sensor node, or "mote," with a faster processor such as the Intel Imote2 [7] or the Stargate platform is used, but this technique comes at the expense of higher power consumption and higher cost.

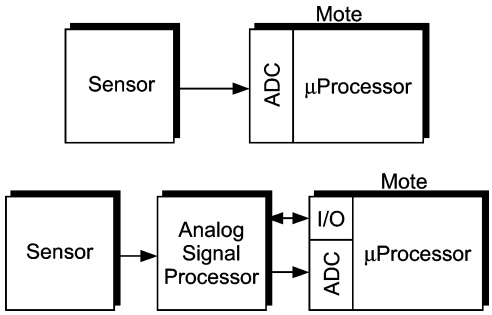

Fig. 1. In contrast to many WSN designs in which sensor data is directly converted into the digital domain by a mote, we introduce an intermediate stage composed of analog circuits for preprocessing of the sensor data. This analog preprocessor allows us to compress the sensor data into relevant characteristics, improve the performance of event detection (while letting the mote sleep), add processing capabilities, and reduce power consumption.

Thus, a fundamental tradeoff exists between the power required to communicate data and the power required to reduce communication using local processing. A variety of network-level design techniques have been developed that trade one for the other in order to increase the life-span of the system [8]–[24]. While these techniques have yielded useful improvements in life-span, available computational resources limit the degree of those improvements. Significant increases in life-span will require simultaneous consideration of both the hardware and the network-level algorithms.

In this paper, we suggest augmenting sensor nodes with an ultra-low-power analog signal processor (ASP). Since analog circuitry offers significant computational resources for minimal power consumption [25], [26], we are using analog signal processing within wireless sensor networks to increase the node-level computational resources while simultaneously reducing the power consumption of these nodes. One of the major objectives of this project has been to develop ways to perform analog preprocessing and classification *prior to* conversion to the digital domain [Fig. 1 (bottom)], as opposed to immediately converting analog data from a sensor into a digital signal via an analog-to-digital converter (ADC), as is typically done [Fig. 1 (top)]. Consequently, we are able to work with the sensor data

Manuscript received January 01, 2011; revised June 03, 2011; accepted July 20, 2011. Date of publication September 15, 2011; date of current version November 09, 2011. This work was supported in part by the West Virginia University Research Corporation and in part by the United States Army Research Laboratory under Contract W911NF-10-0109. This paper was recommended by Guest Editor V. Narayanan.

The authors are with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26506 USA (e-mail: david.graham@mail.wvu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JETCAS.2011.2164996

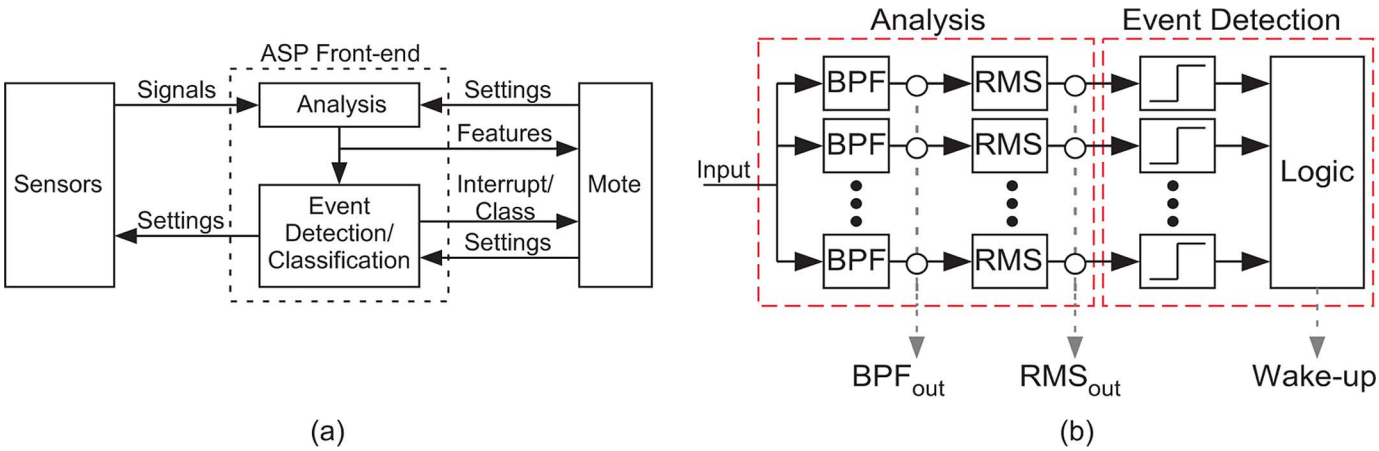

Fig. 2. (a) A generalized "hibernet" system. In such a system, an analog signal processor (ASP) continuously monitors sensor information while the subsequent digital system (i.e., data converters, microprocessor, and radio) are maintained in a low-power sleep mode. Hence, the higher power consuming portions of the signal-processing chain are allowed to hibernate until needed. The ASP consists of 1) an analysis stage that performs signal conditioning, preprocessing of data, and feature extraction and 2) an event-detection/classification stage that generates an interrupt signal when specific types of signals are encountered. This interrupt signal wakes up the digital portions for further processing and/or communication. The digital mote is used as the interface with the WSN designer such that all settings of the ASP can be modified via standard WSN techniques (e.g., TinyOS), even in the field. (b) The specific analog signal processor that we present in this work is capable of performing spectral decomposition of the incoming signal and generating interrupt signals to wake a sleeping mote. This integrated circuit is composed of a bank of bandpass filters (BPFs) for spectral decomposition, a magnitude/rms detection circuit for determining power levels within a sub-band, and comparators for generating interrupt signals.

in its native domain, avoiding unnecessary and power-wasting conversion. Only data that need to be converted are actually processed by the ADC, and only after first being processed/compressed/classified by the analog circuitry.

We demonstrated intial results of an ASP/WSN system in [27], and in this paper we have substantially expanded upon those results. Some of the significant additions are: 1) we elaborate on the circuit-level details of the ASP; 2) we describe our new circuit board which integrates the ASP, mote, and sensors; and 3) we demonstrate the performance (accuracy, system-level lifetime, and ease-of-use) of our ASP-augmented mote in the context of a realistic sensor network application (i.e., vehicle classification).

The outline for this paper is as follows. In Section II, we provide a description of our framework for using ASPs within WSNs to drastically reduce the overall power consumption of a sensor node, and we also provide an overview of how we can apply this technique to acoustic- and vibration-sensing systems. We describe related work and background material in Section III. In Section IV, we present our low-power analog circuitry that is used in our ASP, as well as provide demonstrations of analog event detection. In Section V, we discuss how to interface such ASP systems with standard commercially available motes, and then in Section VI, we apply this overall system to a vehicle-classification scenario. Finally, in Section VII, we discuss our results and summarize our work.

II. Analog Signal Processing in Sensor Networks

We present the framework shown in Fig. 2(a) as a way to use analog signal processing to simultaneously increase local computational resources while decreasing system-level power consumption. In such a system, the ultra-low-power, "always-on" analog circuitry constantly monitors incoming sensor data to determine if the information is relevant to the system's task. Meanwhile, the digital mote (including the ADC) is kept in a low-power state (e.g., sleep mode). Only when the incoming signal is relevant to the system's task does the ASP trigger the digital system and/or the radio to enter a higher-power state to further process and/or transmit the data.

To perform these wake-up and processing duties, the ASP consists of two parts: 1) signal analysis/preprocessing and 2) event detection/classification. The analysis portion serves two purposes: 1) generate features for use in event detection and 2) perform preprocessing to free up the mote's computing resources. The classifier wakes the mote when it detects events of interest and allows the mote to operate at a higher abstraction level, dealing with sensor data at the level of classes.

To demonstrate the potential of the ASP/WSN framework, we have designed and fabricated an analog integrated circuit [Fig. 2(b)] for use in wireless sensor networks. The analysis portion of the system performs spectral decomposition using a constant-relative-bandwidth filter bank with sub-band root mean square (rms) detection circuits. Event detection is performed using a comparator on the rms output of each sub-band, followed by digital logic which asserts a hardware interrupt when the signal spectrum matches a user-defined binary template. The core of this chip operates at an average power of 1–3 μW, which is less than the power consumed by a TelosB mote in its lowest-power sleep mode (>25 μW).

We note that spectral decomposition is a crucial first-step for many sensor network applications, such as acoustic/seismic object classification, event detection, and vibration monitoring [1], [4], [5]. By combining a spectral analyzer with a template-based classifier, our ASP can benefit any application where signal events can be distinguished from other events/noise based on instantaneous frequency content. Therefore, these analog circuits hold great promise for use in wireless sensor networks, and in this paper, we show different ways to utilize these circuits.

Typical low-power sensor mote platforms, such as the TelosB mote [28], are unable to process incoming data at frequencies higher than a few kilohertz, and are also unable to perform significant signal processing operations such as the FFT [29]. This limitation typically warrants the use of higher-processing-ability platforms, such as the Stargate [30], to perform these signal-processing operations, thus increasing the overall system power requirements. By using the ASP to perform spectral decomposition, we offload major computational tasks away from the mote, thus allowing us to use even a low-power platform such as the TelosB mote. Also, by simply sampling the rms energy of individual frequency sub-bands (which can be done at a much lower sampling frequency than the original signal), a mote is able to obtain a complete spectral analysis of the signal. This allows us to operate the system on signals with much higher frequencies than would be possible with a mote alone.

In order for an ASP/WSN system to be practical, the use of the ASP must be as straight-forward as writing programming code in a high-level programming language. Also, there should be some flexibility in controlling the parameters of the circuit at run-time and after deployment. With these requirements in mind, we have interfaced the ASP to a TelosB mote [28]. All signal analysis outputs are multiplexed to a single analog-to-digital converter (ADC) pin on the mote, allowing the mote to sample these outputs using standard TinyOS [31] sensor interfaces. The event detector can be set to generate an interrupt when activity has been detected in a user-selected combination of channels. Additionally, the frequency range and spacing of the filter bank can be varied using the mote's built-in digital-to-analog converters (DACs).

We demonstrate the effectiveness of our cooperative analog-digital mote architecture by using it to implement a vehicle classification system similar to [32] and by comparing the system performance (accuracy, latency, and energy) with an all-digital implementation. Our chosen application scenario is representative of typical WSN applications for monitoring that involve detection and classification of rare, short-lived events and that demand high accuracy and energy-efficiency. By using an ASP to perform computations and by using the digital mote to refine the classification decisions, we are able to achieve classification accuracies of 90%, while extending the battery lifespan from four months for a mote-only implementation to nine years for our ASP-mote implementation.

III. Background

Many WSN applications require some form of spectral analysis for detection and classification of events [1], [4], [29], [32]–[34]. All of these applications have discussed the need for processing within the network in order to decrease communication requirements. Our analog front-end would complement all such systems by providing low-power processing capabilities. Additionally, our ASP can complement the low-power digital processors that are being developed for sensor networks [35]–[38].

Hardware-based event detection, in contrast to sensor polling, has been suggested for reducing power consumption in sensor networks. Jevtic *et al.* [39] reported a crack monitoring device which uses a comparator to trigger a wake-up signal based on the amplitude of the signal. Their complete wake-up circuit consumes 16.5 μW, and they describe the use of both a passive sensor for event detection and a high-precision sensor for event recording. Malinowski *et al.* [40] developed a cargo-monitoring tag with a total quiescent current of approximately 5 μA. In their event-detection circuits, they prepend peak detectors to the comparators, triggering interrupts based on the envelope of the signal. They also describe a dynamically adjustable threshold scheme to achieve a post-event refractory period. Goldberg *et al.* [41] presented an acoustic surveillance system, which uses a digital VLSI periodicity detector (with a core power consumption of 835 nW) to wake up the system. In this paper, we present an analog event detector which goes beyond amplitude-based event detection. We also show how the signal analysis performed for event detection can supplement the mote's processing capability.

As power constraints on various types of systems are becoming more stringent, analog circuits are being reinvestigated for use in low-power systems, such as hearing prostheses [42], implantable electronics [43], and high-level signal-processing algorithms [44] which are normally implemented in digital, such as support vector machines [45], cepstral transforms [46], vector quantizers [47], bidirectional associative memories [48], and belief propagation [49]. With such a portfolio of operations, analog circuits can take the place of digital circuits in many signal processing tasks, such as acoustic event detection as we discuss in this paper. While digital circuits have been used in most settings because of their flexibility, ease of use through programming, noise robustness, benefits of aggressive technology scaling, and scalable dynamic range, analog circuits are able to operate in real-time and perform many computations inherently that would require significant overhead in the digital domain (e.g., multiplication) [26]. Additionally, analog circuitry provides significant power savings over digital, even with the benefits of CMOS scaling for digital systems. For example, it has been observed that ASP performance-per-power represents a 20-year leap over DSP scaling [50], meaning that analog circuitry will continue to provide more efficient signal processing over digital, even though digital processing is progressively becoming more power efficient. This also means that analog has the added benefit of not needing to use the most recent, and often prohibitively expensive, CMOS processes to achieve very low power levels. Instead, analog circuitry can use older and far less-expensive processes and still provide significant power savings. Furthermore, increased leakage current in smaller processes (i.e., current that flows even when a gate is not switching) requires extra attention to keep power low in newer digital systems and can limit the lowest power state of the digital system [51]; this same leakage current rarely affects continuous-time analog circuits to the same degree since they are typically biased to the exact amount of current required for the application.

While general-purpose microcontrollers dominate most WSN systems due to their flexibility, application-specific (and less flexible) digital circuitry could also be used to perform preprocessing for wake-up tasks (i.e., wake up a more powerful digital system). However, the infrastructure required to support such digital systems can still be quite costly in terms of power consumption. One major advantage of using an ASP as opposed

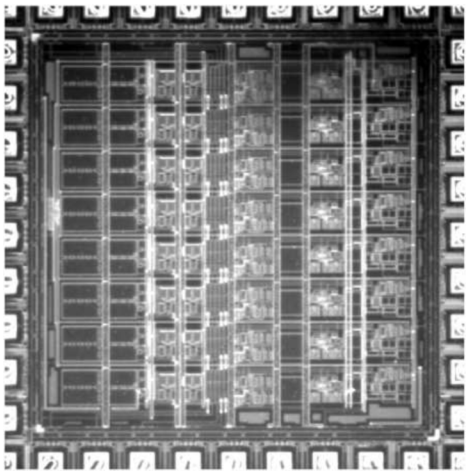

Fig. 3. Die photograph of our ASP.

to a digital ASIC for untethered sensing applications is that the sensed signal will inherently be an analog signal. As a result, an ASP can work directly with the signal in its native format. Additionally, a digital system requires data conversion at the full speed of the signal of interest, whereas an ASP approach can reduce/compress the signal content, thereby allowing a further reduction in required power of the ADC. Beyond the necessary ADC, digital systems also require other infrastructure such as a clock, whereas a continuous-time ASP does not, and generating the clock signal will require even further power consumption.

IV. Design of Analog Computational Elements

Our analog signal processor, shown in Fig. 3, is fabricated on a standard 0.5 μm CMOS process available through MOSIS. This integrated circuit is 2.25 mm^2 and consumes only 3 μW when biased for speech frequencies. The intent is to make a low-power, but discriminating, event detector which can call attention to compelling characterisitics of a signal. The detection approach is to identify when the signal matches a binary spectral template. This integrated circuit has two stages: a spectral analysis stage and an event detection stage that is formed by combining an array of comparators with external logic.

The spectral decomposition front-end is composed of a filter bank with sub-band rms detection circuits. This spectral analysis system is used for frequency-based event detection and for offloading some of the signal processing which would otherwise be performed by the mote. Since the outputs of all of the filters and rms circuits are multiplexed to a single pin, a mote can select the filter output or rms output of any frequency band in order to acquire a frequency-domain representation of the signal.

A. Bandpass Filters

An array of eight bandpass filters (BPFs) is used to create the constant-relative-bandwidth filter bank. The BPFs are based on the compact filters we presented in [52], [53] [shown in Fig. 4(b)], and have been designed to increase the linear range of operation. Using the design principles of [52], we designed these BPFs to obtain an input linear range of 200 mV. These are electronically-tunable continuous-time filters which are suitable for low-frequency, low-power applications.

The corner frequencies are orthogonally tunable through voltages $V_{\tau l}$ and $V_{\tau h}$, which control transconductances g_{m2} and g_{m3}, respectively. The transfer function for the filter is

$$\frac{V_{\text{out}}}{V_{\text{in}}} = -\frac{C_1}{C_2}\frac{s\tau_l\left(1-s\tau_f\right)}{1+s\left(\tau_l+\tau_f\left(\frac{C_O}{C_2}-1\right)\right)+s^2\tau_h\tau_l} \tag{1}$$

where the time constants are given by

$$\tau_h = \frac{C_T C_O - C_2^2}{C_2 g_{m3}} \quad \tau_l = \frac{C_2}{g_{m2}} \quad \tau_f = \frac{C_2}{g_{m3}} \tag{2}$$

and where $C_T = C_1 + C_2 + C_W$ and $C_O = C_2 + C_L$. The time constant τ_f is typically at a sufficiently high frequency that its effect can be safely ignored for most operating conditions, so that the transfer function takes the familiar form of a bandpass filter.

B. Resistive Biasing

Since these circuits are operated in weak inversion, the transconductance values (g_{m2}, g_{m3}) vary exponentially with bias voltages $V_{\tau l}$ and $V_{\tau h}$. This exponential relationship between voltage and frequency allows us to achieve the desired log-frequency spacing across the whole filter bank using a simple resistive divider internal to the chip. The configuration that is used to bias the filter bank is shown in Fig. 4(a), where two large resistive lines are used to generate linearly spaced bias voltages for each channel's $V_{\tau l}$ and $V_{\tau h}$, respectively. The voltages on either end of the resistive dividers can be tuned to cover different frequency ranges and spacings, similar to the procedure that was done with early silicon cochlear models (e.g., [54]–[56]). We use the 1/N octave spacing convention [57], which is common in vibrational and acoustical analyses. In fractional-octave spacing, there are N filters per octave, and the filters cross at their -3 dB frequencies. Fig. 4(c) demonstrates the ability to set the filter bank for one, two, or three filters per octave. These data, and all subsequent data (unless otherwise specified), were obtained from our 0.5 μm standard-CMOS integrated circuit, shown previously in Fig. 3.

One significant benefit to using resistive lines for biasing is the ease of use when incorporated into the larger system with the digital mote. In-the-field reconfiguration, which is a highly desirable attribute of WSNs, is easily obtained by connecting the ends of the resistive lines to digitally programmed voltage supplies (e.g., DACs or digital potentiometers). Only a small number of biases must be changed to alter the frequency range and bandwidths of the filters.

While using a resistive divider to bias the filter bank makes the ASP easy to use, there are a few drawbacks. First, the accuracy of the filter parameters depends on the matching of the resistors, which is generally poor. The effects of this mismatch can be observed by looking at the variation in gain across the AC sweeps in Fig. 4(c). Second, if using the mote's DAC to permit run-time modification of the biases, resistive biasing will require the mote's DAC to remain turned-on all the time, adding to the quiescent power draw. Both of these issues can be solved by using floating-gate transistors for parameter biasing, as we showed in [58]. Floating-gate transistors allow precise programming of each parameter; also, since floating-gate transistors are

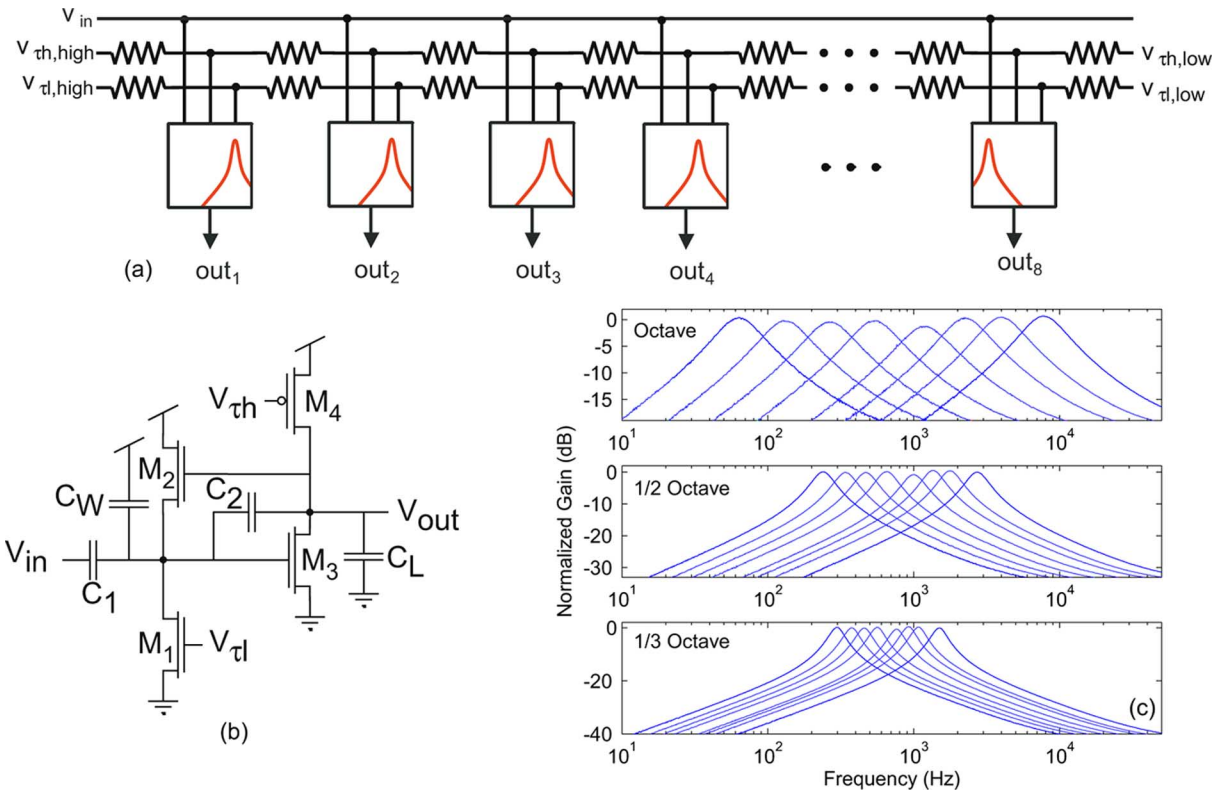

Fig. 4. (a) Schematic of our filter array and biasing structure. Each of the eight filters receive the input signal in parallel. Two resistive lines are used to bias the corner frequencies of all of the filters. Since the filters are operated in the subthreshold regime, linear spacing of the bias voltages translates into exponentially spaced center frequencies. (b) Schematic of our bandpass filter. The corner frequencies are electronically tunable and are independent of each other; they are established by biasing $V_{\tau l}$ and $V_{\tau h}$, respectively. (c) Frequency response of the filter bank for octave spacing, 1/2 octave spacing, and 1/3 octave spacing.

nonvolatile, they do not require any external biasing once they have been programmed. Consequently, in our future work, we are looking to floating-gate transistors to provide further accuracy and control to our ASP/WSN systems.

C. RMS Estimation

To generate an estimate of the signal's power in each subband, we approximate an instantaneous rms by rectifying the output of the BPF and then applying a lowpass filter to the result. To do so, we use the circuit shown in Fig. 5(a). The first stage is based upon the peak detector of [59], but it has been modified such that the output (V_{PD}) is slowly discharged via a constant current drain through M_2. V_D is chosen such that this leakage current through M_2 (i.e., the decay) is much less than the current supplied by the operational transconductance amplifier (OTA), which is illustrated in Fig. 5(b). This peak detector performs a rectification operation and is followed by the second stage, which is a G_m-C lowpass filter.

The rms circuit has three adjustable parameters: attack rate (set by G_{m1}), decay rate (set by V_D), and filtering time constant (set by G_{m2}). The attack and decay rates can be adjusted to track either the rms or the envelope of the signal, and in Fig. 5(b), we illustrate the circuit's ability to track the envelope of the signal. The filtering time constant is adjusted to obtain the best compromise between responsiveness (phase lag) and accuracy (rejecting ripple from the peak detector). Fig. 5(c) demonstrates the combination of the filter bank and rms detector. In Fig. 5(c), our spectral decomposition system is set for 1/2-octave spacing, starting at 250 Hz. The input to the filter bank is a logarithmic chirp signal. Shown below the input are the responses of the second, fourth, sixth, and eigth bands of the decomposition system. As the chirp sweeps to higher frequencies, the response of the higher-frequency sub-bands increases, and the response of the lower-frequency sub-bands decreases. Note that the output of the rms circuit is the signal content in that band.

Voltage biases for the rms circuits are set by a resistive divider in the same way as for the filters, allowing the frequency range of the entire spectral analysis system to be controlled by two bias voltages.

D. Event Detection

By combining the spectral analysis system with comparators and digital logic, we form a simple yet selective event-detection system, with flexibility to define what constitutes an event. Fig. 6 provides a simple example in which an event is defined as occuring when signal content is present in one of two channels, but not both. The two bands being compared are 500 Hz and 1.4 kHz. The input consists of a 500 Hz sine wave and a 1.4 kHz sine wave which overlap for 10 ms. The wakeup signal is generated by combining the comparator outputs for those two bands using an exclusive-or (XOR) operation, so that the interrupt is asserted only when one band exceeds the threshold. In [27], we also illustrated an example in which we detected harmonically related content, which is a scenario that is straightforward to

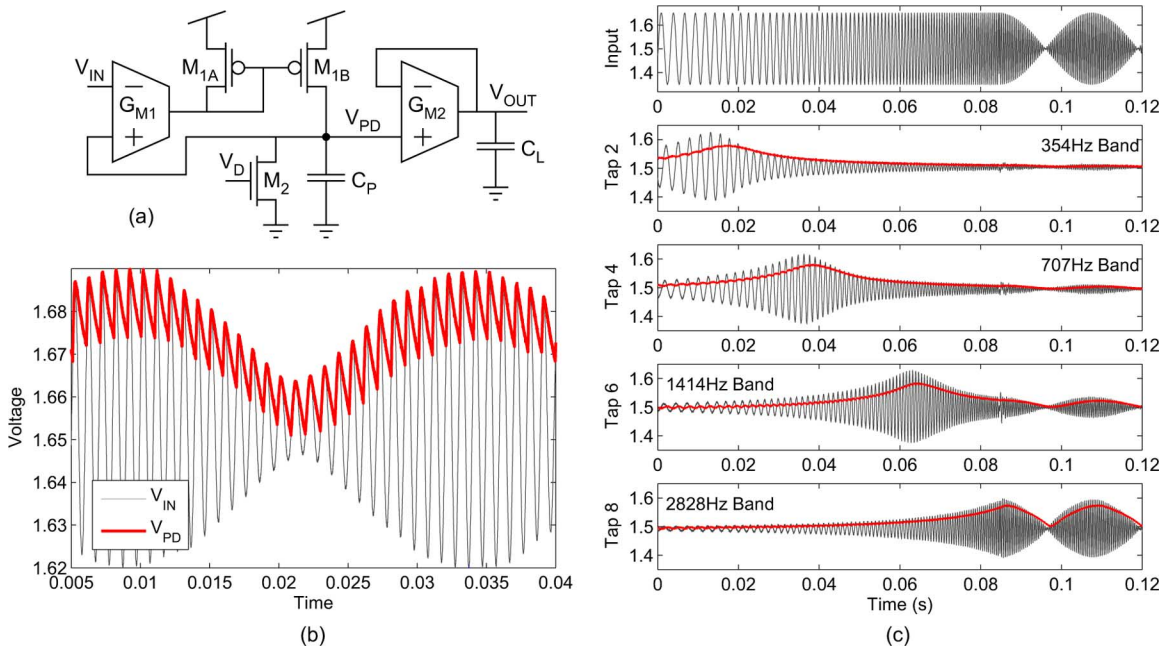

Fig. 5. (a) Schematic of the circuit used for magnitude/rms detection. The current mirror half wave rectifies the output current of G_{M1} onto C_P. The bias for G_{M1} is high enough that the peak detector output (V_{PD}) follows the input signal when $V_{\text{in}} \geq V_{PD}$. Current sink M_2 is tuned to obtain the desired linear decay of V_{PD} when $V_{\text{in}} < V_{PD}$. This tuning results in the sawtooth-like peak detector output seen in (b), which is then filtered with the follower-integrator to obtain the smooth rms output seen in (c). (b) Peak detection waveforms. (c) Demonstration of the spectral decomposition front end. The input (top plot) is a logarithmic chirp. The rest of the plots show the response of four of the sub-bands to obtain an estimate of the rms of the signal. Note that each channel's response is frequency dependent, and that the rms outputs represent spectral characteristics of the signal.

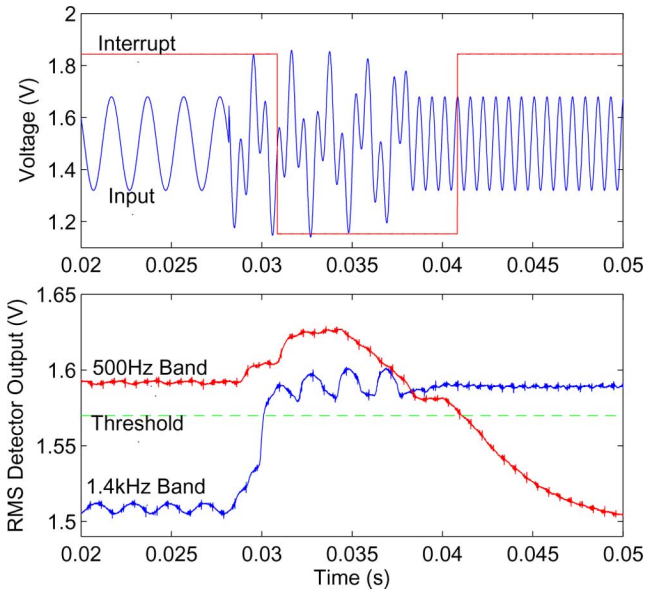

Fig. 6. Demonstration of multiband detection using an exclusive-or template. The input consists of two overlapping sine waves. The bottom plot shows the outputs of the rms circuits for the 500 Hz and 1.4 kHz bands. The comparator outputs of those two bands are combined via an exclusive-or to generate an interrupt when only one band exceeds the threshold.

establish using a filter array with $1/N$ octave spacing, such as ours. For example, we defined an event to contain spectral activity in multiple harmonically related bands with the simultaneous absence of spectral activity in nonharmonically related bands.

In these examples, we observe that there is some lag-time between when the event occurs and when the interrupt signal is asserted. The lag-time is caused by the rms circuit, and is a result of filtering the peak-detected signal. By adjusting the parameters of the rms circuit, the phase-lag can be reduced at the expense of reduced rms tracking accuracy. This lag time is related to the frequency, f, of the sub-band, and is approximately $4/f$ for the rms circuit biasing used in this paper. For an application where the mote should record the event, this phase-lag could cause the onset of the event to be overlooked. Regardless of how small the phase lag is, we will miss the prelude to the event. This problem will be present in all systems that wake up based on event detection. To solve the phase-lag problem, the designer can include a memory buffer. This buffer may take the form of an analog delay line (continuous-time continuous-value), an array of sample-and-holds (discrete-time continuous-value), or low-power ADC and RAM (discrete-time discrete-value). This memory can also have a second use of adding memory to the event detection algorithm.

E. Power Consumption

The power consumed by our analog integrated circuit is dominated by the bandpass filters, and to a lesser extent, the rms-estimation circuits. As we presented in [52], which describes the circuit that this paper's BPF is based upon, the power consumed by the BPF is linearly proportional to its center frequency. This relationship is shown in Fig. 7 for a filter tuned to a 1/2-octave bandwidth. This relationship enables the system designer to determine the maximum frequency of operation available at a given power budget.

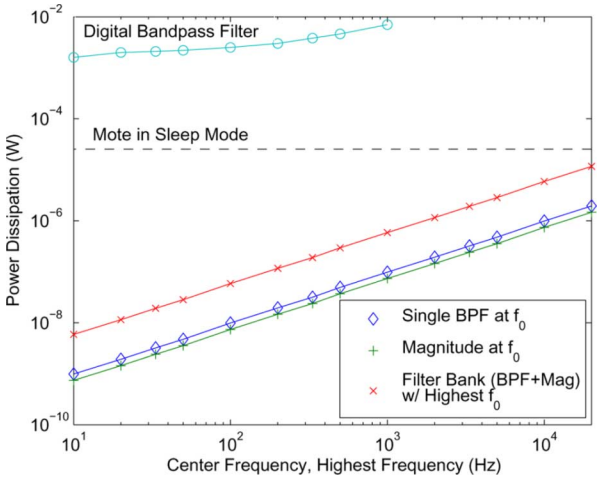

Fig. 7. The amount of power consumed by our analog spectral-decomposition block depends primarily on the center frequency of the bandpass filter of the highest-frequency sub-band. The x-axis shows the center frequency for the filter and rms circuit, and also shows the center frequency of the highest-frequency sub-band for an array that performs spectral analysis. The power numbers are extrapolated from circuit simulations. Also included are power measurements from the digital mote including the minimum measured power consumption in sleep mode and also the power consumption of the mote performing a simple, single bandpass filtering operation (at multiple frequency locations). Note that this mote was unable to simultaneously sample and filter data at frequencies above approximately 1 kHz.

As described for the BPF, the power consumption of the rms circuit also scales with frequency. Additionally, the rms circuit can be tuned in various fashions within a given frequency band f_0; for example, this circuit can follow either the envelope or the rms of a signal. Therefore, this circuit has a range of power-consumption values for a given f_0. Fig. 7 shows the worst-case scenario (i.e., highest power consumption) for operation within a given frequency band, f_0.

The overall power consumption of our analog spectral-decomposition block is set by the center frequency of the highest filter tap. The power consumption of the entire spectral-decomposition system is described by a geometric series, resulting in a total power consumption of

$$P_{\text{tot}} = \frac{P_{\text{BPF,high}} + P_{\text{rms,high}}}{1 - 2^{-1/N}} \tag{3}$$

where $P_{\text{BPF,high}}$ and $P_{\text{rms,high}}$ represent the power consumed by the BPF and rms circuits in the highest-frequency sub-band, and N indicates the number of filters per octave. The total amount of power consumed by the analog block is shown in Fig. 7 for the case of 1/2-octave spacing. Included in Fig. 7 is the measured power consumption of the TelosB mote in sleep mode (25.4 μW, which is within the specified bounds of 15–60μW). For the entire audio frequency band, our spectral-decomposition block consumes less power than a sleeping mote.

V. Interfacing With the Telos Mote

To evaluate the potential of the ASP/WSN framework, we interfaced the integrated circuit described in Section IV with the TelosB mote. The TelosB mote was chosen for its low-power sleep mode (25 μW) and fast wakeup time (6 μs), which make it suitable for a hardware-based wake-up system. A printed circuit board [Fig. 8(b)] was built to combine all components of the system [Fig. 8(a)]. A summary of the power consumption of the circuit board is shown in Table I.

Two acoustic sensors, including both an electret microphone and a MEMS microphone, are incorporated on the circuit board. Additionally, there is an auxiliary interface for connecting different sensors, such as the passive piezoelectric microphone which we use as the main microphone. Included is a low-power microphone amplifier based around the MCP6141 operational amplifier. The sensor output is available as an input to both the ASP and the mote's ADC.

As mentioned in Section IV, the filter bank and sub-band processing elements are biased with a resistive divider. On the circuit board, the voltages at the ends of these resistive lines are provided by a network of digital potentiometers (AD5263) and two low-power voltage references (the ISL60002 and the REF3318). In addition, the comparator trigger point is also set via the digital potentiometers. A resolution of approximately 2.5 mV is available across the nominal range of bias voltages for each circuit. The mote applies new settings to the potentiometers after receiving updates over the radio.

A complex programmable logic device (CPLD) was used for the hardware-based pattern classifier. The XC2C32A was chosen due to its low power consumption. The CPLD arbitrates all digital connections between the Mote and ASP and serves two roles: 1) it implements the detection rules that operate on the comparator outputs, i.e., it performs the role of the "Logic" in the event detector [Fig. 2(b)], and 2) it serves as a serial-to-parallel converter, allowing us to use just three of the mote's digital I/O pins to select which of the ASP's analog outputs are connected to the mote's ADC, and also to choose between different sets of detection rules which are preloaded into the CPLD. For event detection, the CPLD receives the comparator outputs from all frequency bands, and performs template matching to detect/classify events. Upon detecting an event, one of the CPLD output pins wakes up the mote via a hardware interrupt, and the other output pins indicate the classification of the event.

To provide the mote with access to the ASP's signal analysis, the outputs from all bandpass filters and rms circuits are multiplexed to the mote's ADC. The mote communicates through the CPLD to specify which sub-band is connected to the ADC. To acquire the entire spectral representation, the mote selects a new sub-band between each sample. Due to the low-frequency nature of the rms outputs, the mote is able to cycle through all channels without experiencing aliasing.

VI. Performance Evaluation

In this section, we describe two modes in which we can exploit the computational capabilities of the analog integrated circuit for WSN applications, namely 1) selective wake-up mode and 2) selective sample mode. We then quantify the performance gained in both cases. Finally, we demonstrate the use of our ASP-interfaced mote in a vehicle classification application and highlight the energy efficiency gained in comparison to an all-digital implementation.

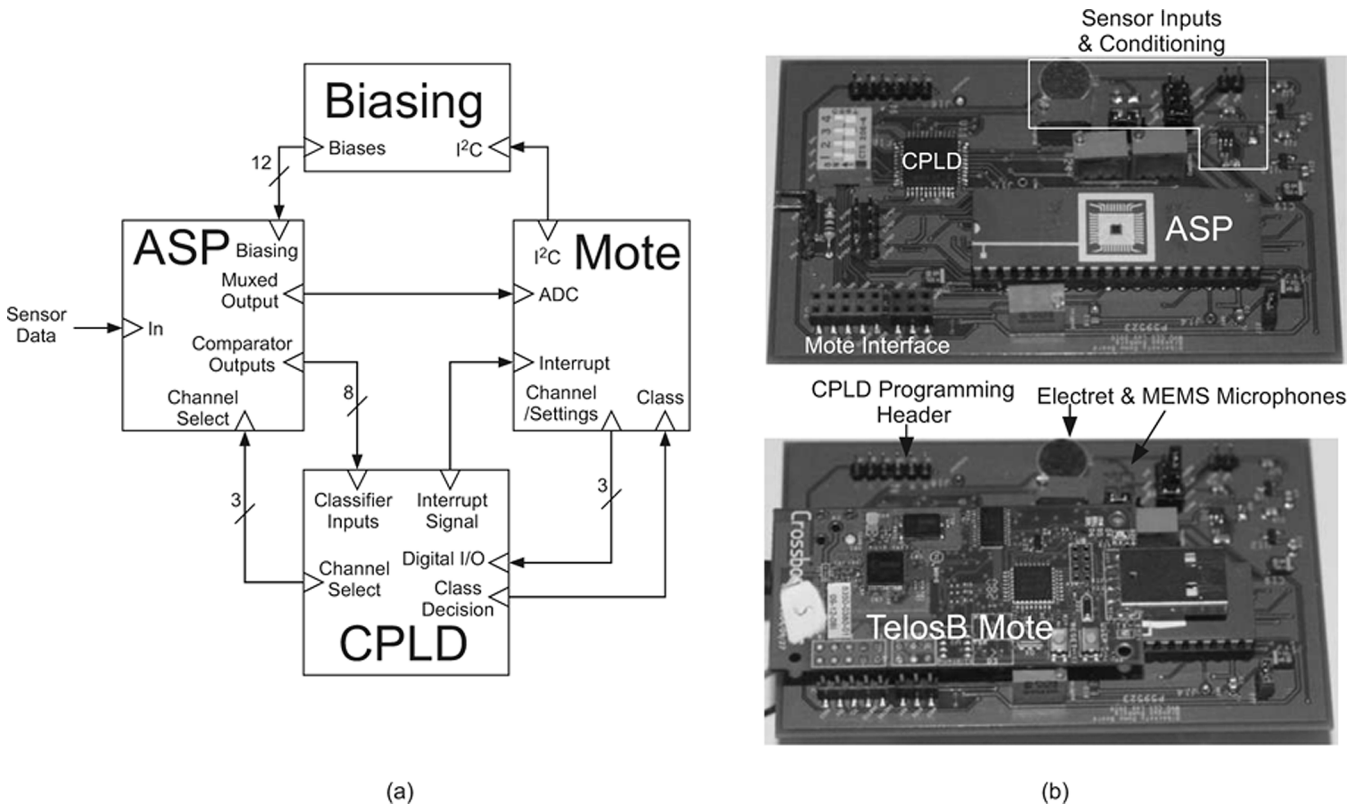

Fig. 8. (a) Block diagram of the entire system. (b) The complete system shown with and without the mote connected to it. A significant portion of our printed circuit board is consumed by the large packaging for our integrated circuit, which was used for ease of testing. However, such large packaging as this would not be needed in future versions for an integrated circuit that is only 2.25 mm^2. Additionally, future versions of this integrated-circuit design could incorporate most of the external support circuitry that is on the printed circuit board. An updated version of this integrated circuit will include analog memory elements (floating-gate transistors) to replace the digital potentiometers, and a programmable logic array to replace the external CPLD. Consequently, a future version is expected to be significantly smaller and consume even less power.

TABLE I
POWER CONSUMPTION

	Device	Power	Power (Projected)
ASP-mote event-monitoring	ASP	3μW	3μW
	CPLD	48μW	5μW
	Biasing	135μW	20μW
	Sensor (w/ interfacing)	3μW	3μW
	Sleeping mote	25μW	25μW
	Total	214μW	56μW
ASP-mote sampling/ processing	ASP board	189μW	31μW
	I/O Buffers	30μW	30μW
	Awake mote	1.5mW	1.5mW
	Total	1.72mW	1.56mW
Transmitting	ASP board	189μW	31μW
	Transmitting mote	60mW	60mW
	Total	60.19mW	60.03mW

A. *Selective Wake-Up Mode*

In the selective wake-up mode, we take advantage of the low-power processing capability of analog circuits by placing the mote into long periods of hibernation and then selectively waking the mote when a user-specified combination of frequency components are present in the signal. Fig. 9(a) demonstrates single-band event detection. The band of interest is 1 kHz and the filter has a quality factor of 2.8. Signal content appears in the band at 2.6 s but has noise added to it. This noise is a combination of white noise and tones at 100, 600, and 10 kHz. The bandpass filter focuses on the frequency of interest and the comparator triggers once the rms reaches the threshold. Note that the sub-band event is detected despite having much lower amplitude than the noise and other frequency components. In this mode, the mote samples the raw sensor signal when it wakes up and then transmits it to a basestation. The signal received by the basestation is shown in the bottom trace.

In order to compare the power consumed by the ASP-interfaced mote with a mote-only implementation, we implement a second-order Butterworth bandpass filter on a TelosB mote running TinyOS and measure the power consumed. The measurements are taken with a stock TelosB mote, without any of the components added for ASP-interfacing. The digital filter is implemented by buffering 100 samples at a time and then computing the filter outputs after every 100 ms. The power consumed for this operation is measured for sampling frequencies ranging from 10 Hz to 1 kHz. In Fig. 7, we compare the average power consumed by the digital filter with the power drawn by an analog bandpass filter for different sampling frequencies and center frequencies, respectively. No data points could be obtained for mote power at frequencies above 1 kHz since that is the highest sampling frequency that the TelosB can simultaneously sample and filter data. We point out that the energy consumed by our entire spectral analysis system is over 1000

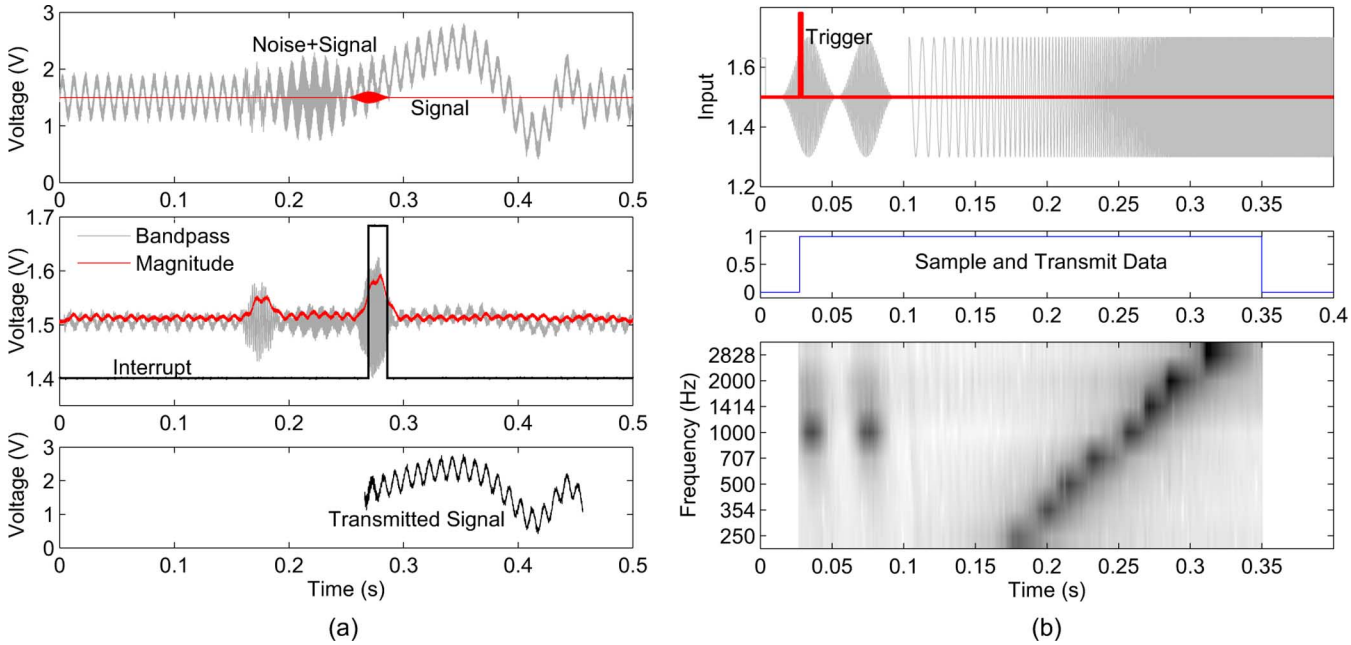

Fig. 9. (a) Single-band event detection. The frequency of interest is 1 kHz and is the "Signal" trace in the top plot. Broadband noise and tones at 100, 600, and 10 kHz are added to "Signal," generating "Noise+Signal," which is the input to the analog signal processor. The middle plot shows the response of the three stages of the processor within the 1 kHz sub-band. The bandpass filter cuts the undesired frequencies, while the rms circuit tracks the magnitude. Once the magnitude exceeds the threshold, an interrupt is generated to wake the mote. The mote then samples the output of the sensor and transmits it to the basestation. The received signal is plotted in the bottom plot. (b) Sampling of preprocessed sub-bands and spectral analysis performed by the analog IC. (Top) The input signal is composed of two 1 kHz pulses followed by a logarithmic chirp signal. (Middle) Once the trigger goes high, the ADC samples all eight channels for a user-specified amount of time (e.g., 300 ms). (Bottom) Spectrogram of the transmitted frequency-dependent magnitude data as received by the base station.

times lower than the power consumed by a single digital filter, thus signifying the energy savings compared to keeping a mote always turned on.

B. Selective Sample Mode

In the selective wake-up mode described in the previous subsection, once the mote is awake, it samples the raw signal for processing or transmission. The drawback with this approach is that a low-power processing platform such as the TelosB mote is unable to sample and process signals of high frequencies and is also limited in the kind of signal processing operations that can be performed (an FFT, for example, is infeasible on a TelosB mote [29]). This often warrants the use of a platform with greater processing capabilities, such as the IMote2 or Stargate [30], for performing these signal processing operations, which increases the overall power requirements of the system. In this subsection, we highlight the selective-sample mode of operation in which we take advantage of the ASP's ability to perform pre-ADC signal analysis. The ASP is used to perform a full spectral analysis of the input signal, and the mote only samples the rms energy of each sub-band. Thus, we are able to reduce the computational resources required at the mote, allowing for lower power operation.

In the experiment of Fig. 9(b), the input signal consists of two 1-kHz pulses followed by a chirp signal. The 1-kHz pulse is used to trigger the mote into sampling the rms energy of each sub-band in succession, for a specified period of time. The mote scans through sub-bands by writing to the general-purpose input/output (GPIO) ports output register between each sampling operation. The frequency-decomposed rms data obtained by the mote is transmitted to a base station and is displayed in the bottom plot. We note that by scanning through the energy of all the sub-band channels in succession, a complete spectral decomposition can be obtained at the mote in real time using the analog circuit. By doing so, we are also able to operate the system on signals with much higher frequencies (since we sample only the rms amplitude of sub-bands) than would be possible with a mote alone.

C. Evaluation in the Context of a Vehicle Classification Application

In order to evaluate the accuracy and energy-efficiency of our ASP-interfaced mote in the context of an actual sensor network application, we have used the system in a recreation of the all-digital acoustic-sensor-based vehicle classification experiment that we described in [32]. The vehicle classification system is intended for unattended monitoring of secure facilities. The objective of the system is to accurately identify an approaching vehicle as belonging to one of multiple categories, such as small, medium, and large vehicles, and then accurately raise an alert when a vehicle of a particular type has been detected. The vehicles are assumed to appear in isolation and not concurrently with other vehicles. The system is required to have a long lifespan on battery sources, while at the same time retaining high accuracy and low latency in classification. Arrival of any vehicle is expected to be a rare event, therefore rendering duty cycling of resources essential for energy-efficiency—but at the same time it is critical that no vehicles are missed. We note that the chosen application is representative of typical wireless sensor network applications for monitoring, such as detection of anomalies in bridges [60], unattended ground sensing by military personnel in combat situations, classification of objects

for asset protection [1], classification of animal sounds [61], and monitoring of seismic activity. All of these applications involve detection and classification of rare, short-lived events and demand high accuracy and high energy-efficiency.

In this subsection, we describe the implementation of the vehicle classification system described above using our ASP-mote architecture and compare the system performance of our cooperative analog-digital implementation with that of an all-digital implementation. We specifically consider classification into two vehicle categories: car and truck.

1) Data Collection: The dataset collected for the experiments described in [32] was used for performance evaluation in this paper. The acoustic sensor used for data collection was a Samsung C01U—USB Studio Condenser Microphone. The directional microphone was placed 10–12 ft from the road, mounted one foot off the ground, and combined with Samson windshields to filter out wind noise. A mid-sized car and a pickup truck were considered as the two vehicle classes. Multiple observations were collected for both vehicles, which were driven at speeds between 10 mi/h and 30 mi/h. Ambient data was also collected using the microphone without any vehicle being present in the scene.

2) Training: The dataset was first normalized so that the peak amplitude of the signal across vehicle classes was uniform. The dataset was then divided into two sets, one for training and the other for testing, and regions of the data corresponding to when the vehicle was and was not present were manually identified. Based on the short-time FFT spectra of the data, the ASP's filter bank parameters were chosen to be half-octave spacing from 100 to 1131 Hz. Using these filter bank settings, analysis was performed on all of the training samples by streaming them through the ASP using a DAC and recording the rms output of each sub-band. After obtaining the rms data, the objective was to determine the combination of comparator trigger-point and codeword assignments (where a codeword was defined as the 8-bit output from the eight comparators) which achieves the desired classification performance. During training, each of the possible 256 codewords were associated with a class (i.e., car, truck, and no vehicle).

The training procedure, which was performed offline, was to iterate through comparator threshold values (20 steps of 10 mV), performing the following steps for each threshold: 1) thresholding was applied to the rms data to obtain an 8-bit codeword for each time step, 2) the distribution of each class (i.e., car, truck, and no vehicle; combined across all observations of the class) across all codewords was computed, 3) each codeword was assigned to the class that was most likely to result in observing that codeword (i.e., the class that caused that codeword for the largest percentage of time), and then 4) the combination of comparator threshold and codewords was evaluated by finding the percentage of time-samples which were associated with the correct class. After iterating through the threshold values, the threshold that resulted in the largest percentage of correct decisions in step 4 was chosen as the final threshold, and the codeword assignments found in step 3 for that threshold were used as the final codeword assignments.

Once the comparator threshold and codeword assignments were determined, the system was configured by transmitting the threshold value to the mote and programming the codeword assignments into the CPLD via the JTAG header on the circuit board. The CPLD was programmed such that the interrupt pin went high whenever a codeword associated with either a car or truck was encountered, and the classification pin went high whenever a truck was encountered.

Note that the instantaneous categorization generated by the ASP is susceptible to false decisions due to noise or differences in the "approaching" versus "present" sounds of the vehicle. Hence, it is possible for an interrupt pin to be reset despite the presence of a vehicle, causing the GPIO pins to provide a false classification. In order to compensate for these false decisions, we use the mote to generate the final classification output based on inputs from the ASP over a length of time. Once an interrupt has been generated by the ASP, the mote stays on and records the state of the interrupt pin and the GPIO pin until the interrupt stays low continuously for a duration of 100 ms, confirming that the vehicle is outside of the sensing range. The mote then generates the final classification result as the most frequent decision from the ASP over the duration of the event. This simple decision-accumulation scheme provides good classification results; however, the scheme increases latency since it waits until the vehicle has left the sensing range before making a decision. Alternative schemes may be used to make the decision sooner, and future versions of the ASP will include decision-accumulation capabilities to avoid waking the mote prematurely.

3) Testing: All testing was performed by streaming the samples into the ASP using a 16-bit DAC at a sampling frequency of 4 kHz. The operation and the power consumption of the ASP-interfaced mote is shown in Fig. 10 in the form of a timing diagram for one 10-s test sample of a truck [Fig. 10(i)] being classified. The truck is closest to the sensor between seconds 4–6 of the test sample [shown in Fig. 10(ii)]. The spectral analysis output of the event detector front-end is shown in Fig. 10(iii) in the form of a spectrogram. The comparator outputs of the eight filter bands are shown in Fig. 10(iv), and the CPLD outputs are shown in Fig. 10(v)–(vi). The CPLD interrupt pin goes high when either a car or truck is detected, and the CPLD class pin specifies the classification (high for truck, low for car), which is only valid when the interrupt pin is high. Once the interrupt is generated, the mote is awake and starts accumulating the classifications from the CPLD (consuming about 1.5 mW of power). When a final decision is made, the output is transmitted via radio (if it was determined that an event occurred), which consumes 60 mW [Fig. 10(vii)]. The detailed power consumption of the ASP-interfaced mote for the various operations being performed are shown in Table I. The accuracy of classification is highlighted in Table II. An overall accuracy of 90% is achieved with an average false alarm rate of one false positive every 50 s in the presence of amplified ambient wind noise.

4) Comparison With All-Digital Implementation: Low-power computing platforms such as the TelosB mote are unable to perform spectral analysis on-board, and therefore processing platforms such as the Stargate have to be used to perform signal processing. Since these devices consume significantly higher power, they are typically used in a layered architecture in conjunction with mote platforms that act as wakeup devices to trigger the detection of an event. In [32], we presented an

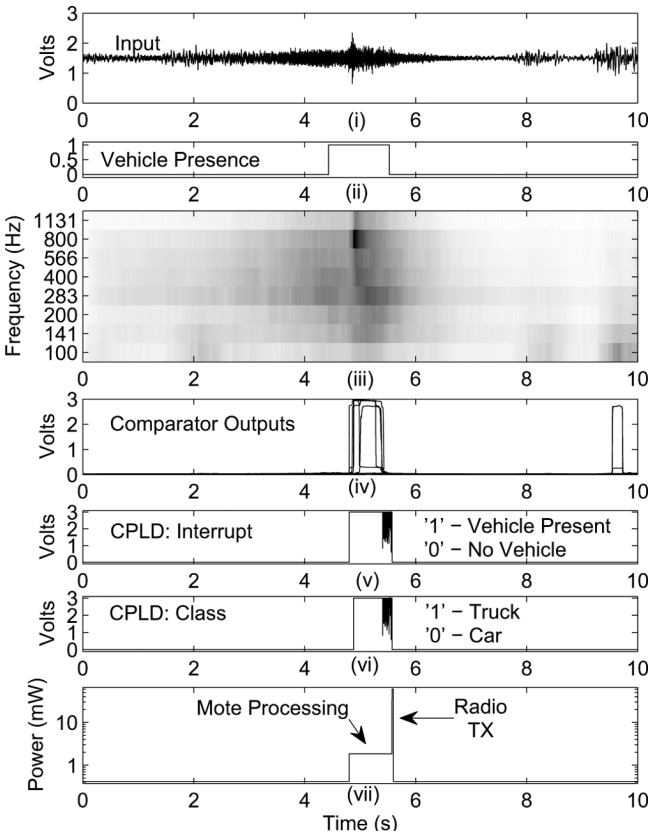

Fig. 10. Demonstration of the stages of the detection system for a 10-s test sample of a truck being classified. The truck is closest to the sensor between seconds 4–6 of the test sample [shown in (ii)]. The comparator outputs of the eight filter-bands are shown in (iv) and the CPLD outputs are shown in (v) and (vi). The CPLD interrupt pin goes high when a car or truck is detected. The CPLD class pin specifies the classification (high for truck, low for car) and is only valid when the interrupt pin is high. Once the interrupt is generated, the mote is awakened and starts recording and accumulating the CPLD classifications (consuming about 1.5 mW of power). When a final decision is made, the output is transmitted via radio, which consumes 60 mW [shown in (vii)].

TABLE II
VEHICLE CLASSIFICATION RESULTS

	Ground Truth		
	NULL (200 seconds)	Car (10 Samples)	Truck (10 Samples)
NULL		20%	0%
Car	2 false alarms	80%	0%
Truck	2 false alarms	0%	100%

all-digital implementation using such a layered architecture for the vehicle classification system described above. In that all-digital implementaion, a low-power Mica2 mote attached to a seismic sensor stays on all the time to detect the arrival of a vehicle. Upon detection of a vehicle, the mote triggers a signal to wake up a Linux-based Stargate platform that performs spectral analysis for vehicle classification. The Mica2 mote stays on all the time and consumes 24 mW of power when processing and 60 mW when transmitting. The Stargate running off of a 4.2 V battery consumes 420–470 mA when processing for a duration of 8–10 s per vehicle detection. In comparison, the cooperative analog-digital implementation described in this paper consumes only 214 μW of power when idle and 1.5 mW when an event is detected.

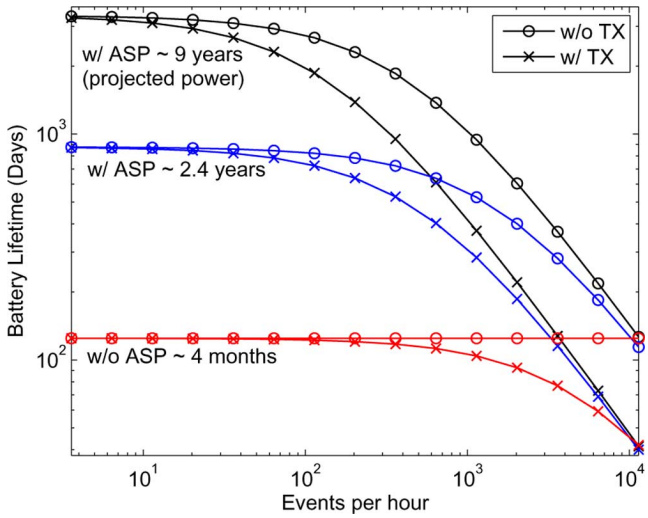

Fig. 11. System lifetime as a function of event frequency.

Now we analyze the power savings afforded by using the ASP in the vehicle classification scenario. Table I details the contribution of each component to the system's power budget, showing the power breakdown of the ASP-augmented mote for three operating states: event-monitoring (mote asleep while ASP performs event detection), sampling/processing (mote awake, e.g., the decision-accumulation scheme discussed in Section VI-CII), and data transmission. Since this initial IC did not have optimized biasing and did not integrate the logic that is needed for the pattern matching portion of the event detector, we provide two sets of power numbers: the measured values for this system which are shown under "Power," and the expected power values (assuming integrated event-detection logic and floating-gate biasing) which are shown under "Power (Projected)." The projected power numbers are based on previous floating-gate-biased filter banks and programmable logic arrays that we have made. The "Sensor (w/interfacing)" number is for a passive piezoelectric microphone with a low-power amplifier.

To visualize the power savings of the event detector, we plot the system lifespan as a function of the frequency of events (Fig. 11), assuming a nominal battery capacity of 1500 mAh. Using the power numbers given in Table I, the lifespan is calculated for the following platforms: an ASP-augmented mote (w/measured power numbers), an ASP-augmented mote (w/projected power numbers), and a digital-only mote. For our comparisons, the digital-only mote is the TelosB, which is one of the lowest-power commercially-available mote platforms. Comparing against the TelosB mote provides overly optimistic projections for all-digital platforms since, as we discussed in Section VI-A, the TelosB is unable to perform even a single bandpass filter in real-time for signal bandwidths exceeding 1 kHz; a more power-consuming mote would actually need to be used for an actual all-digital implementation, resulting in further lifespan reductions. Each platform is considered with and without the cost of transmitting the classification decision (which requires the radio to be turned on for 16 ms). The digital-only scenario with no transmission has a constant lifetime since it is always awake and processing, while the digital-only scenario with transmission shows a decreasing lifetime with

increasing event frequency since the radio is turned on more frequently. In the ASP scenario, the mote enters a low-power state between events. When events are infrequent, the average power consumption of the system approaches the sum of the ASP and sleeping-mote power levels. As events become more frequent, the mote spends a larger percentage of time awake, and the system's average power consumption approaches the sum of the ASP and awake-mote power levels. When events occur so rapidly that the mote never turns off, the lifespan of the ASP-augmented system drops slighly below the lifespan of the mote-only system due to the additional power of the ASP.

5) Discussion: We note that in our implementation, we used the ASP to output binary decision bits, which are read with the GPIO pins on the mote and used to make the final classification output. Alternatively, the mote can be used to sample the rms energy of each sub-band, as described in Section VI-B, while the interrupt pin stays high, and the mote can then use the sampled spectrogram of the signal to make a decision. Such an approach is likely to be beneficial in a more general classification scenario with much more than two classes.

D. Other Applications and Potential Extensions

We chose to implement spectral decomposition as our computational block in this version of our ASP because spectral analysis is often the first step in a majority of WSN event detection/classification applications (e.g., vibration monitoring for anomaly detection in buildings, bridges, etc., [5], vehicle classification [4], [33], habitat monitoring, perimeter monitoring [34]). Combining a filter bank with a template-based classifier allows the system to be used for any scenario where events can be distinguished from other events/noise based on the instantaneous frequency content. Further improvements on this acoustic processor could be gained by using more sophisticated classifiers (e.g., [62]–[64]) that have memory, and also by using signal features other than the spectrum, such as the cepstrum, which has been shown to provide better separation between acoustic classes for both speech and vehicle applications, and has been previously implemented in analog ICs [46]. If the application is changed to something that is not suitable for frequency analysis (e.g., imaging or chemical sensing), or if the domain is changed to something other than event detection (e.g., object localization/tracking), then a different set of operations will need to be implemented in the ASP.

Digital processing is likely to provide a simpler solution for systems in which a particular sensor node demands very high resolution processing, branching (e.g., state machines), long-term data storage of sensor information, or a high degree of flexibility for in-the-field reconfiguration. However, ASPs can still be used to complement the digital processors in such scenarios. For example, a system which requires high-resolution processing (e.g., SNR $>$ 16 bits) can use a lower-resolution ASP (where ASPs are always more efficient than digital circuitry [25]) to act as an energy-management tool to wake up a high-resolution digital system. Also, operations which require branching can be accomplished by incorporating state machines into the "Logic" portion of the ASP [see Fig. 2(b)]. Additionally, recent developments in programmable/reconfigurable analog systems enable analog integrated circuits to be general-purpose and easy to use, thereby providing significant flexibility and reducing the design time [65], [66].

VII. Conclusion

In this paper, we have described how ultra-low-power analog circuitry can be integrated with sensor nodes to reduce the node-level power consumption. We have shown the ability to interface these circuits with existing sensor platforms and have presented demonstrations to illustrate how analog hardware can reduce node resource usage and increase performance. We have implemented a vehicle classification system using our ASP-interfaced mote and have shown that it significantly improves the energy-efficiency over that of an all-digital implementation while retaining high classification accuracies.

We have utilized the strong points of both analog and digital such that each computational domain compensates for the limitations of the other. Specifically, by combining the ASP and the mote's microcontroller, we retain the flexibility of configuring system parameters at run-time and of implementing additional high-level decision making on the motes. At the same time, the use of the ASP enables ultra-low-power operation by reducing the amount of time that the mote is powered on and by reducing the required computational resources implemented by the mote. By using both the analog and digital systems together, we have increased the lifetime of a wireless sensor network system from a few months to several years.

Acknowledgment

The authors would like to thank A. Singireddy for his help with the layout of the analog integrated circuit.

References

[1] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A line in the sand: A wireless sensor network for target detection, classification, and tracking," *Comput. Netw.*, vol. 46, pp. 605–634, Dec. 2004.

[2] A. Arora, R. Ramnath, E. Ertin, P. Sinha, S. Bapat, V. Naik, V. Kulathumani, H. Zhang, H. Cao, M. Sridharan, S. Kumar, N. Seddon, C. Anderson, T. Herman, N. Trivedi, C. Zhang, R. Shah, S. Kulkarni, M. Aramugam, and L. Wang, "Exscal: Elements of an extreme scale wireless sensor network," in *Proc. IEEE Int. Conf. Embedded Real-Time Computing Syst. Appl.*, Hong Kong, 2005, pp. 102–108.

[3] T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, G. Zhou, J. Hui, and B. Krogh, "Vigilnet: An integrated sensor network system for energy-efficient surveillance," *ACM Trans. Sensor Netw.*, vol. 2, pp. 1–38, Feb. 2006.

[4] M. Duarte and Y. Hu, "Vehicle classification in distributed sensor networks," *J. Parallel Distributed Comput.*, vol. 64, pp. 826–838, Jul. 2004.

[5] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Proc. Int. Conf. Embedded Networked Sensor Syst.*, Baltimore, MD, 2004, pp. 13–24.

[6] V. Raghunathan, C. Schurgers, S. Park, M. Srivastava, and B. Shaw, "Energy-aware wireless microsensor networks," *IEEE Signal Process. Mag.*, vol. 19, pp. 40–50, Mar. 2002.

[7] L. Nachman, J. Huang, J. Shahabdeen, R. Adler, and R. Kling, "Imote2: Serious computation at the edge," in *Proc. Int. Wireless Commun. Mobile Comput. Conf.*, 2008, pp. 1118–1123.

[8] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "Networked Slepian-Wolf: Theory, algorithms and scaling laws," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4057–4073, Dec. 2005.

[9] S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 51–60, Mar. 2002.

[10] M. Gastpar, P. L. Dragotti, and M. Vetterli, "The distributed Karhunen-Love transform," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5177–5196, Dec. 2006.

[11] D. Chu, A. Deshpande, J. Hellerstein, and W. Hong, "Approximate data collection in sensor networks using probabilistic models," in *Proc. Int. Conf. Data Eng.*, Apr. 2006, pp. 3–7.

[12] R. Wagner, R. Baraniuk, S. Du, D. Johnson, and A. Cohen, "An architecture for distributed wavelet analysis and processing in sensor networks," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, Nashville, TN, 2006, pp. 243–250.

[13] J. Acimovic, B. Beferull-Lozano, and R. Cristescu, "Adaptive distributed algorithms for power-efficient data gathering in sensor networks," in *Proc. Int. Conf. Wireless Netw., Commun. Mobile Comput.*, 2005, vol. 2, pp. 946–951.

[14] A. Ciancio, S. Pattem, A. Ortega, and B. Krishnamachari, "Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, Nashville, TN, 2006, pp. 309–316.

[15] G. Shen and A. Ortega, "Joint routing and 2D transform optimization for irregular sensor network grids using wavelet lifting," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, St. Louis, MO, 2008, pp. 183–194.

[16] S. Pattem, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 4, no. 4, pp. 60–66, Sep. 2008.

[17] S. Pattem, G. Shen, Y. Chen, B. Krishnamachari, and A. Ortega, "SenZip: An architecture for distributed en-route compression in wireless sensor networks," presented at the Workshop Sensor Netw. Earth Space Sci. Appl., San Francisco, CA, Apr. 2009.

[18] P. von Rickenbach and R. Wattenhofer, "Gathering correlated data in sensor networks," in *Proc. Joint Workshop Foundations Mobile Comput.*, Philadelphia, PA, 2004, pp. 60–66.

[19] G. Shen, S. Pattem, and A. Ortega, "Energy-efficient graph-based wavelets for distributed coding in wireless sensor networks," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 2009, pp. 2253–2256.

[20] R. Cristescu, B. Beferull-Lozano, M. Vetterli, and R. Wattenhofer, "Network correlated data gathering with explicit communication: NP-completeness and algorithms," *IEEE/ACM Trans. Networking*, vol. 14, pp. 41–54, Feb. 2006.

[21] S. Servetto, "Sensing LENA—Massively distributed compression of sensor images," in *Proc. IEEE Int. Conf. Image Process.*, 2003, vol. 1, pp. 613–616.

[22] J. Gao, L. Guibas, N. Milosavljivec, and J. Hershberger, "Sparse data aggregation in sensor networks," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, Cambridge, MA, 2007, pp. 430–439.

[23] R. Sarkar, X. Zhu, and J. Gao, "Hierarchical spatial gossip for multi-resolution representations in sensor networks," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, Cambridge, MA, 2007, pp. 420–429.

[24] V. Kulathumani and A. Arora, "Distance sensitive snapshots in wireless sensor networks," in *Proceedings of the International Conference on Principles of Distributed Systems*, Guadeloupe, French West Indies, 2007, pp. 143–158.

[25] R. Sarpeshkar, "Analog versus digital: Extrapolating from electronics to neurobiology," *Neural Computat.*, vol. 10, pp. 1601–1608, Oct. 1998.

[26] P. Hasler, P. Smith, D. Graham, R. Ellis, and D. Anderson, "Analog floating-gate, on-chip auditory sensing system interfaces," *IEEE Sensors J.*, vol. 5, no. 5, pp. 1027–1034, Oct. 2005.

[27] B. Rumberg, D. Graham, and V. Kulathumani, "Hibernets: Energy-efficient sensor networks using analog signal processing," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, Stockholm, Sweden, 2010, pp. 129–139.

[28] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw.*, Los Angeles, CA, 2005, pp. 364–369.

[29] B. Greenstein, C. Mar, A. Pesterev, S. Farshchi, E. Kohler, J. Judy, and D. Estrin, "Capturing high-frequency phenomena using a bandwidth-limited sensor network," in *Proc. Int. Conf. Embedded Networked Sensor Syst.*, Boulder, CO, 2006, pp. 279–292.

[30] Stargate Gateway (SPB400) C. T. Inc. [Online]. Available: http://www.willow.co.uk/

[31] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," in *Architectural Support Programm. Languages Operat. Syst.*, Nov. 2000, vol. 35, pp. 93–104.

[32] J. Frigo, V. Kulathumani, S. Brennan, and E. Raby, "Sensor network based vehicle classification and license plate identification system," in *Proc. Int. Conf. Networked Sensing Syst.*, Pittsburgh, PA, 2009, pp. 224–227.

[33] B. Malhotra, I. Nikolaidis, and J. Harms, "A simple vehicle classification framework for wireless audio-sensor networks," *J. Telecommun. Inf. Technol.*, pp. 43–50, Jan. 2008.

[34] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler, "Design of a wireless sensor network platform for detecting rare, random, and ephemeral events," in *Proc. Int. Symp. Inf. Process. Sensor Netw.*, Los Angeles, CA, 2006, pp. 497–502.

[35] V. Ekanayake, C. Kelly, IV, and R. Manohar, "An ultra low-power processor for sensor networks," in *Proc. Int. Conf. Architectural Support Programm. Languages Operat. Syst.*, Boston, MA, 2004, pp. 27–36.

[36] B. Zhai, S. Pant, L. Nazhandali, S. Hanson, J. Olson, A. Reeves, M. Minuth, R. Helfand, T. Austin, D. Sylvester, and D. Blaauw, "Energy-efficient subthreshold processor design," *IEEE Trans. Very Large Scale (VLSI) Syst.*, vol. 17, pp. 1127–1137, Aug. 2009.

[37] M. Sheets, F. Burghardt, T. Karalar, J. Ammer, Y. Chee, and J. Rabaey, "A power-managed protocol processor for wireless sensor networks," in *Proc. IEEE Symp. VLSI Circuits*, 2006, pp. 262–263.

[38] M. Hempstead, N. Tripathi, P. Mauro, G.-Y. Wei, and D. Brooks, "An ultra low power system architecture for sensor network applications," in *Proc. Int. Symp. Comput. Architecture*, 2005, pp. 208–219.

[39] S. Jevtic, M. Kotowsky, R. Dick, P. Dinda, and C. Dowding, "Lucid dreaming: Reliable analog event detection for energy-constrained applications," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, Cambridge, MA, 2007, pp. 350–359.

[40] M. Malinowski, M. Moskwa, M. Feldmeier, M. Laibowitz, and J. Paradiso, "Cargonet: A low-cost micropower sensor node exploiting quasi-passive wakeup for adaptive asynchronous monioring of exceptional events," in *Proc. ACM Conf. Embedded Networked Sensor Syst.*, Sydney, Australia, 2007, pp. 145–159.

[41] D. Goldberg, A. Andreou, P. Julian, P. Pouliquen, L. Riddle, and R. Rosasco, "A wake-up detector for an acoustic surveillance sensor network: Algorithm and VLSI implementation," in *Proc. Int. Symp. Inf. Process. Sensor Netw.*, Berkeley, CA, 2004, pp. 134–141.

[42] R. Sarpeshkar, C. Salthouse, J.-J. Sit, M. Baker, S. Zhak, T.-T. Lu, L. Turicchia, and S. Balster, "An ultra-low-power programmable analog bionic ear processor," *IEEE Trans. Biomed. Eng.*, vol. 52, no. 4, pp. 711–727, Apr. 2005.

[43] R. Harrison and C. Charles, "A low-power low-noise CMOS amplifier for neural recording applications," *IEEE J. Solid-State Circuits*, vol. 38, no. 6, pp. 958–965, Jun. 2003.

[44] P. Hasler and D. Anderson, "Cooperative analog-digital signal processing," in *Proc. IEEE Conf. Acoustics, Speech, Signal Process.*, Orlando, FL, May 2002, vol. 4, pp. 3972–3975.

[45] R. Genov and G. Cauwenberghs, "Kerneltron: Support vector machine in silicon," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1426–1434, Sep. 2003.

[46] P. Smith, M. Kucic, R. Ellis, P. Hasler, and D. Anderson, "Mel-frequency cepstrum encoding in analog floating-gate circuitry," in *Proc. IEEE Int. Symp. Circuits Syst.*, Scottsdale, AZ, 2002, vol. 4, pp. 671–674.

[47] G. Cauwenberghs and V. Pedroni, "A low-power CMOS analog vector quantizer," *IEEE J. Solid-State Circuits*, vol. 32, no. 8, pp. 1278–1283, Aug. 1997.

[48] B. Linares-Barranco, E. Sanchez-Sinencio, A. Rodriguez-Vazquez, and J. Huertas, "A CMOS analog adaptive BAM with on-chip learning and weight refreshing," *IEEE Trans. Neural Netw.*, vol. 4, no. 3, pp. 445–455, May 1993.

[49] H. Loeliger, F. Lustenberger, M. Helfenstein, and F. Tarkoy, "Probability propagation and decoding in analog VLSI," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 837–843, Feb. 2001.

[50] T. Hall, C. Twigg, P. Hasler, and D. Anderson, "Application performance of elements in a floating-gate FPAA," in *Proc. Int. Symp. Circuits Syst.*, Vancouver, Canada, 2004, vol. 2, pp. 589–592.

[51] Y. Taur, "CMOS design near the limit of scaling," *IBM J. Res. Develop.*, vol. 46, pp. 213–222, 2002.

[52] D. Graham, P. Hasler, R. Chawla, and P. Smith, "A low-power, programmable bandpass filter section for higher-order filter applications," *IEEE Trans. Circuits Syst. I*, vol. 54, no. 6, pp. 1165–1176, Jun. 2007.

[53] D. Graham, "A biologically inspired front end for audio signal processing using programmable analog circuitry," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, GA, 2006.

[54] A. van Schaik, E. Fragnière, and E. Vittoz, "Improved silicon cochlea using compatible lateral bipolar transistors," in *Advances in Neural Information Processing Systems 8*. Cambridge, MA: MIT Press, 1996, pp. 671–677.

[55] R. Sarpeshkar, R. Lyon, and C. Mead, "An analog VLSI cochlea with new transconductance amplifiers and nonlinear gain control," in *Proc. IEEE Int. Symp. Circuits Syst.*, Atlanta, GA, 1996, vol. 3, pp. 292–296.
[56] D. Graham and P. Hasler, "Capacitively-coupled current conveyer second-order sections for continuous-time bandpass filtering and cochlea modeling," in *Proc. IEEE Int. Symp. Circuits Syst.*, Scottsdale, AZ, May 2002, vol. 5, pp. 485–488.
[57] *American National Standard Specification for Octave-Band and Fractional-Octave-Band Analog and Digital Filter*, ANSI S1.11-1986.
[58] D. Graham, P. Smith, R. Chawla, and P. Hasler, "A programmable bandpass array using floating-gate elements," in *Proc. IEEE Int. Symp. Circuits Syst.*, Vancouver, BC, May 2004, vol. 1, pp. 97–100.
[59] M. Kruiskamp and D. Leenaerts, "A CMOS peak detect sample and hold circuit," *IEEE Trans. Nucl. Sci.*, vol. 41, no. 1, pp. 295–298, Feb. 1994.
[60] J. Rice, K. Mechitov, F. Spencer Jr., and G. Agha, "Autonomous smart sensor network for full-scale structural health monitoring," in *Proc. SPIE Conf. Sensors Smart Structures Technol. Civil, Mechan., Aerospace Syst.*, 2010, vol. 7647.
[61] W. Hu, V. Tran, N. Bulusu, C. Chou, S. Jha, and A. Taylor, "The design and evaluation of a hybrid sensor network for cane-toad monitoring," in *Proc. Int. Symp. Inf. Process. Sensor Netw.*, Los Angeles, CA, 2005, pp. 382–387.
[62] R. Edwards and G. Cauwenberghs, "Mixed-mode correlator for micropower acoustic transient classification," *IEEE J. Solid-State Circuits*, vol. 34, no. 10, pp. 1367–1372, Oct. 1999.
[63] T. Yamasaki and T. Shibata, "Analog soft-pattern-matching classifier using floating-gate MOS technology," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1257–1265, Sep. 2003.
[64] S. Peng, P. Hasler, and D. Anderson, "An analog programmable multidimensional radial basis function based classifier," *IEEE Trans. Circuits Syst. I*, vol. 54, no. 10, pp. 2148–2158, Oct. 2007.
[65] M. Kucic, J. Dugger, P. Hasler, and D. Anderson, "Programmable and adaptive analog filters using arrays of floating-gate circuits," in *Proc. Conf. Adv. Res. VLSI*, Atlanta, GA, Mar. 2001, pp. 148–162.
[66] T. Hall, C. Twigg, J. Gray, P. Hasler, and D. Anderson, "Large-scale field-programmable analog arrays for analog signal processing," *IEEE Trans. Circuits Syst. I*, vol. 52, no. 11, pp. 2298–2307, Nov. 2005.

Brandon Rumberg (S'08) received the B.S. degrees in electrical engineering and computer engineering and the M.S. degree in electrical engineering from West Virginia University, Morgantown, in 2007 and 2009, respectively, where he is currently working toward the Ph.D. degree in electrical engineering.

His research interests include low-power sensing systems, mixed-signal integrated circuits, and extending circuit-level innovations to benefit system-level applications.

David W. Graham (S'00–M'07) received the B.A. degree in natural science from Covenant College, Lookout Mountain, GA, in 2001, and the B.S. degree in electrical engineering, and the M.S. and Ph.D. degrees in electrical and computer engineering all from Georgia Institue of Technology, Atlanta, in 2001, 2003, and 2006, respectively.

He is an Assistant Professor in the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown. His research interests are in developing biologically inspired electronics, cooperative analog and digital signal-processing systems, programmable analog devices, and low-power electronics.

Vinod Kulathumani received the B.E. degree in computer engineering from the University of Mumbai, Mumbai, India, in 1999, and the M.S. and Ph.D. degrees in computer science from The Ohio State University, Columbus, in 2001 and 2008, respectively.

He is an Assistant Professor in the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown. His research interests are in designing energy-efficient and robust middleware services for wireless sensor actuator networks and developing low-power computing platforms for embedded systems.

Robert Fernandez (S'10) received the B.S. degrees in electrical engineering and computer engineering and the M.S. degree in electrical engineering from West Virginia University, Morgantown, in 2009 and 2011, respectively.

His research interests include analog circuit design and low-power sensing systems.