

DSPs for Energy Harvesting Sensors: Applications and Architectures

Energy harvesting from human or environmental sources shows promise as an alternative to battery power for embedded digital electronics. Digital signal processors that harvest power from ambient mechanical vibration are particularly promising for sensor networks.

Over the past decade, embedded digital electronics have proliferated in both number and variety. Applications such as cellular phones, portable multimedia devices, and sensor networks have kept pace with dramatic increases in computing power and functionality. Battery technology, however, has not. Batteries limit the operating lifetime of portable devices and add undesirable weight and volume. They can't store sufficient energy to support long-lifetime embedded applications such as monitoring civil infrastructure or studying the environment. Their replacement cost poses a major barrier to

scaling wireless sensor networks to hundreds or thousands of nodes.

Energy harvesting from human or environmental sources is a promising alternative to address these limitations and open new frontiers for integrating digital computation with sensing and actuation. Several alternative energy harvesting paradigms are possible, as either a substitute or a complement for batteries. The commercial sector has adopted mechanical energy harvesting as a redundant power source. Products already on the market include radios, flash-

lights, and cell-phone chargers powered by hand-cranked electrical generators or shake-to-recharge electronics. But these mechanisms aren't suitable for all applications: first, they have low energy and power densities; second, they require active user involvement.

Researchers have explored various passive energy-harvesting power sources for portable or wearable devices. These include gravity-driven and vibration-driven electromagnetic generators, piezoelectric shoe inserts, and thermocouples for harvesting energy from human body thermal gradients. Passive power sources for sensor networks, another target area for energy harvesting,¹ include ambient mechanical vibration. One project, which some of us worked on, developed a MEMS variable-capacitor transducer and accompanying chips that could harvest machine vibrations for sensor signal processing.²

Despite a promising start, energy harvesting is still in its infancy. For current sensor network nodes, vibration-based energy harvesting allows an RF transmit duty cycle of less than 3 percent, excluding any computation that occurs at the transmitter.¹ Communication typically dominates power consumption, so many applications must maximize the computation done at a particular node.³ Required off-chip power electronics can increase system cost and volume, and AC/DC conversion losses can limit energy-harvesting operation.

Rajeevan Amirtharajah,
Jamie Collier, and Jeff Siebert
University of California, Davis

Bicky Zhou
Intel

Anantha Chandrakasan
*Massachusetts Institute of
Technology*

TABLE 1
FFT specifications for machine vibration monitoring.

Computation	Sample rate	FFT size (n)	Number of n-point FFTs	Operations	Duty cycle (%)
Low-bandwidth FFT	1.8 kHz	512	13	542,464	2.5
High-bandwidth FFT	18 kHz	512	139	5,800,192	2.7
FFT total				10,682,368	5.2

Much current work addresses these problems from the power generation side by developing and optimizing energy harvesting transducers. However, the desire for smaller devices and higher integration levels poses constraints that fundamentally limit output power. Our work addresses the issues from the power consumption standpoint by developing digital signal processing (DSP) architectures and circuits that are energy efficient, energy scalable, and robust to transducer output-voltage variations.

Two applications

Sensor applications are particularly well suited to energy harvesting because they typically require low throughputs. We explored two such applications for energy harvesting: computing a fast Fourier transform (FFT) to monitor a shipboard gas turbine's vibrations and using data from a wearable acoustic biomedical sensor to analyze a user's exertion state.

Monitoring gas turbine vibration

Large machinery generates vibrations during operation. These vibrations offer a possible energy source, but the vibration signature can also indicate changes in the machine's performance or even impending failure. The vibration spectrum supports analysis of sudden shifts in vibration. An FFT is a computationally elegant means of computing this spectrum.

Each FFT butterfly operation must read four operands and write back two results. Assuming a flat memory hierarchy (no caching) and fully serial computation, every operation requires one read-evaluate-write cycle. The computation thus requires a large number of cycles. This is

acceptable, however, because it also has a low duty cycle, so an application can spread the computation out in time.

This particular application must compute one FFT every five minutes. Of every 10 FFTs, nine are low-bandwidth computations (1.8 kHz) and one is high-bandwidth (18 kHz). Despite the high cycle count, the processor idles with its clock gated off for significant periods thanks to the low throughput requirement. This implies that leakage power becomes significant for low-threshold-voltage complementary metal-oxide semiconductor (CMOS) transistors. Table 1 shows the specifications for both FFTs, including spectral averaging.

The throughput requirement also sets the clock rate. Five minutes is a very long time to perform the computations required, even with the serial approach.

We assume a clock rate for each computation equal to the sample rate. Each

FFT has 512 frequency points computed with 12-bit fixed-point data.

This FFT application is just one example of computationally intensive signal preprocessing. Further computation processes this spectrum into a machine-state diagnosis.

Signal processing for a wearable physiological monitor

We explored a physiological monitoring application that uses a wearable microphone as a biomedical sensor to determine the wearer's physical condition (exertion state). We estimated that 400 μ W of power would be available from a wearable AA battery-sized electromagnetic generator,⁴ sufficient for recently demonstrated biomedical devices. Figure 1 shows the demonstration system.⁵ The system algorithm first detects heartbeats, then uses them to determine heart rate as the basis for a physiological assess-



Figure 1. Wearable biomedical acoustic sensor demonstration system.

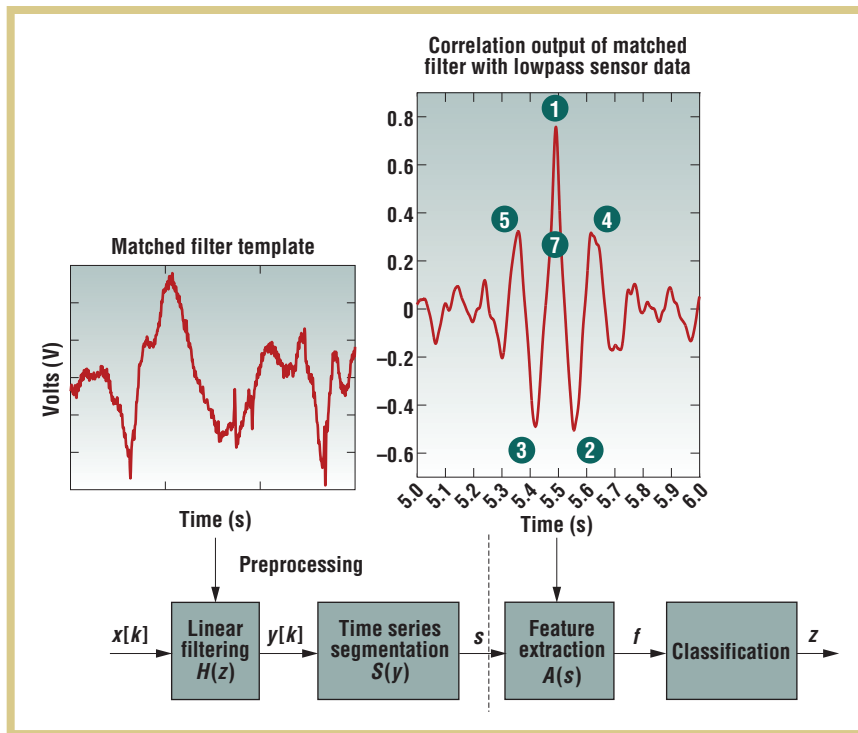


Figure 2. Heartbeat detection and classification algorithm.

ment. We used a similar system to examine the possibility of determining breathing rate as a basis for assessment.

Heartbeat detection. The basic algorithm has three phases. First is *preprocessing*, which also has three phases:

- *Low-pass filtering.* The data is band-limited to below 200 Hz to eliminate as much of the voice and breath energy as possible.
- *Matched filtering.* The low-pass filter's output goes through a matched filter to determine the candidate heartbeat signals.
- *Segmentation.* The sensor output is divided into overlapping segments at least long enough to contain a full heartbeat but short enough not to contain more than one.

The algorithm's second phase is *feature extraction*, which computes a vector of seven features from the segmented, matched filter output. The features are scalar quantities helpful in recognizing heartbeats—for example, filter output peak values.

Finally, the *classification* phase uses a parametric Gaussian multivariate classifier to classify each feature vector into a heartbeat or nonheartbeat.

Figure 2 summarizes the algorithm. Assuming that antialiasing band-limits the data, the first significant computation is the matched filtering. The matched filter impulse response, or *filter template*, is a denoised version of the acoustic heartbeat signature. When convolved with the input data, the filter produces large correlation peaks at the input heartbeats' time locations. Figure 2 includes a template and an example correlation peak.

The segmentation phase localizes the regions of this time series that have correlation peaks. The algorithm then extracts features from these regions, labeled 1, 2, 3, 4, 5, and 7 in the figure, and classifies them. (Feature 6 represents the total energy in the segment, so its label doesn't appear on the graph.) The preprocessing steps, in particular the matched filtering, require the most operations and consume the most time, as the algorithm specifications in table 2 show.

Breath detection. Breath and speech acoustic energy is concentrated in the high-frequency (> 200 Hz) portion of the sensor data spectrum. Peaks in the moving average of the energy for narrow-bandpass-filtered versions of the original signal indicate fairly well when breaths are occurring. Peak width indicates breath duration. A “popping” noise—indicated by sharp, narrow spikes in the energy time series and probably representing a sensor artifact—contributes extra energy in these bands that might lead to a misclassification. As with heartbeats, we use a classifier-based approach for breath detection.

The algorithm divides the time series into short-duration nonoverlapping segments. Each segment is labeled according to whether breathing (class 1), “popping” noise or speech (class 3), or background noise (class 2) is occurring during the segment. The extracted features are basically the signal energy in three different high-frequency bands, normalized by the energy in the highest band to eliminate misclassification of broadband noise. This normalization factor and the total high-frequency energy form the last two candidate features.

These five features are

- normalized energy in the 200 to 600 Hz band,
- normalized energy in the 600 Hz to 1 kHz band,
- normalized energy in the 1 to 1.4 kHz band,
- normalization factor for energy in the 1.4 to 1.8 kHz band, and
- total energy in the 200 Hz to 1.8 kHz band.

The 4 kHz sampling rate sets the upper bound on frequency at 2 kHz, but the signal doesn't have much energy at the edge of the antialiasing passband. The feature probability distributions indicate that

TABLE 2
Heartbeat detection algorithm specifications.

Computation	Sample rate	Clock rate	Operations	Duty cycle (%)
Heartbeat preprocessing	160 Hz	1.2 kHz	18,170,000	99.8
Feature extraction and classification	Variable	250 kHz	110,000	0.2
Heartbeat total			18,280,000	100.0

they are basically Gaussian. Therefore, we again use the multivariate Gaussian parametric classifier as the breath detection engine. For simulation purposes, we eliminated the speech and “pop” noise classes, since a more sophisticated algorithm will probably be needed to handle these high-energy signals.

The classifier recognition performance for these candidate features is generally poor (< 70 percent accurate), particularly because the transition times when a breath starts and stops are difficult to classify accurately solely on the basis of energy. However, if we can classify consecutive breath samples consistently, we can construct a binary sequence that determines when a breath is occurring. Each breath would consist of several 1s in a row, and each pause in breathing by a string of 0s. Using the 0 to 1 and 1 to 0 transitions, we could estimate the duration of the breath and pauses. Counting the pulses of 1s would give a good estimate of the breathing rate. This would constitute a first step toward estimating the wearer’s exertion state using the breath signals. We could process this stream of 1-bit samples serially.

Voice stress analysis. Voice stress might also be a good indicator of physical exertion state. Further research on low-power speech algorithms might show the feasibility of processing voice stress for low- to medium-throughput low-power DSPs. However, this analysis requires complex algorithms that we have yet to explore and might require unreasonably high performance from the signal-processing engine, given the limited available power.

Implications for architecture

Two factors contribute to CMOS power dissipation P . One is *dynamic power* spent in switching capacitances and the other is *static power* dissipated during constant current flow:

$$P = \alpha CV_{dd}^2 f + I_{stat} V_{dd} \quad (1)$$

where α represents the probability of a particular node switching, C is the node capacitance, V_{dd} is the supply voltage, f is the clock frequency, and I_{stat} accounts for static current flowing from the supply to the ground, including analog bias circuits and device leakage.

A second equation

$$P = E_{diss}/\Delta t \quad (2)$$

expresses power as dissipated energy E_{diss} divided by time Δt .

As tables 1 and 2 show, the FFT and heartbeat estimation applications represent two extremes of energy harvesting sensor operation. The FFT application requires a very low duty cycle, so static power due to leakage will increase as CMOS technology scales. We can decrease leakage at the cost of increased dynamic power by employing serial computation.

In contrast, the heartbeat detection algorithm runs continuously and is dominated by preprocessing steps, so a low-power DSP architecture must optimize its frequent computations.

Sensor DSP architecture

Because of the unknown and time-varying nature of the power available from energy harvesting, energy scalability is a critical feature for energy harvesting sensor DSPs: they must be able to trade energy dissipation for some quality metric of DSP output. Energy-scalable hardware includes techniques for *approximate processing*, which treats power and arithmetic precision as system parameters that can trade off each other.

We implemented an energy-scalable serial computation technique as part of a DSP chip. We called the chip Sensor-DSP and measured its performance for heartbeat detection.

Bit-serial computation and distributed arithmetic

Leakage currents are expected to contribute an ever larger percentage of total power as CMOS technologies scale.⁶ We can address this by using serial arithmetic techniques. Older CMOS processes used bit-serial techniques to reduce the area of large arithmetic structures such as multipliers. These techniques use registers to decrease the amount of combinational logic needed to perform a computation. To maintain a fixed throughput, we must clock a bit-serial implementation at N times the specified frequency, where N is the data bitwidth. This increases the dynamic power consumption (equation 1). However, the serial implementation’s reduced area and transistor count also decrease the static power consumption due to leakage currents.

Figure 3 shows the estimated total power dissipation for serial and parallel multipliers as frequency and technology scale. At high throughputs, the serial implementation’s dynamic power dominates the total power because of higher required clock frequencies. At low throughputs, the implementation’s low static power consumption presents a significant advantage. So, it can be seen that below a certain throughput threshold serial computations have lower total power, and this threshold increases as technology scales due to increased leakage currents in deep-submicron CMOS. At 130 nanometers, the throughput

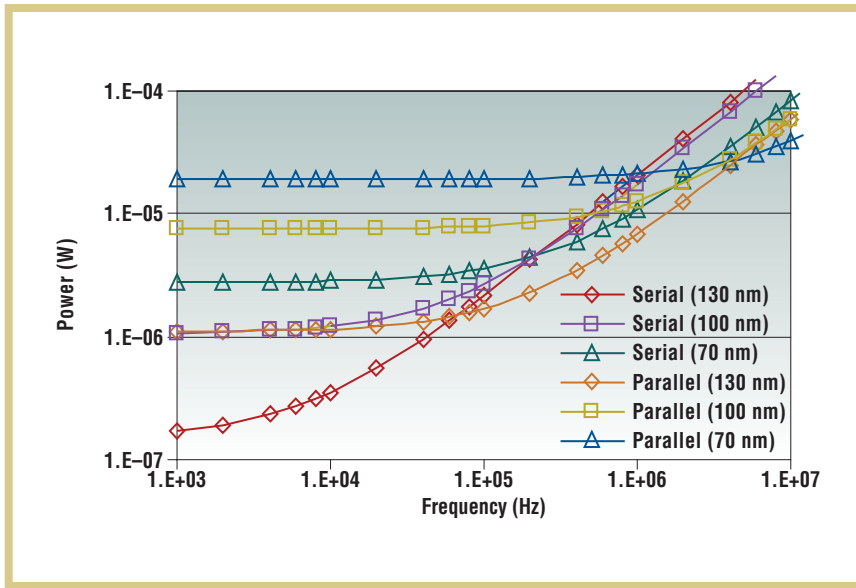


Figure 3. Estimated power dissipation for serial and parallel multipliers as technology scales.

threshold is just below 100 kHz, about four times faster than the maximum throughput requirement for the FFT application. At 18 kHz, a 130-nm serial multiplier implementation for the FFT will have three times less power than a traditional array multiplier.

Serial computation is also an elegant approximate-processing technique. By reducing the number of bits shifted in during the computation (truncating the input data), the dynamic power decreases linearly with the input bitwidth. The resulting increased truncation error also increases the quantization noise that degrades the computation's output. *Distributed arithmetic*, a method of computing vector dot products (equivalent to finite-length impulse response filters) without multipliers, offers an energy-efficient serial DSP hardware implementation.⁷ DA reorders the computation by considering a bit slice through all input samples rather than

each sample individually. Each bit slice is an M -bit binary number corresponding to a unique linear combination of filter coefficients. The programmer precomputes all 2^M possible combinations and stores them in a *lookup table*.

The computation addresses the lookup table with successive bit slices, then shifts and accumulates the table's read data until the DA unit consumes all input data bits. By truncating the computation before reaching this condition, DA features successive approximation properties. If a single table implemented a typical filter, the lookup table would grow exponentially and would be unrealizable. Instead, adders can accumulate multiple smaller DA units' outputs into the final result.

This structure enables another power performance trade-off. Enabling various DA units allows the number of filter taps to vary independently of the input bitwidth. The SensorDSP chip has demon-

strated this trade-off in energy harvesting applications.⁸

SensorDSP implementation and results

We developed the SensorDSP chip to demonstrate low-power and energy-scalable signal processing for wearable biomedical sensors. Figure 4 shows the chip's architecture, which follows the algorithm described earlier. We used energy-scalable DA to implement the matched filter. Its output feeds a nonlinear/short linear filtering unit, which calculates quantities used in segmentation. The microcontroller performs the segmentation, feature extraction, and classification. The buffer provides synchronization between the front-end filtering and back-end processing. The filtering front end must run continuously to process input samples, which arrive at a fixed rate. However, the back-end classification must be performed only for each segment, not each sample.

The system first filters the input and writes results to the buffer. The microcontroller continuously executes a small loop, checking to see if the preprocessing logic has written a full segment to the buffer. When the microcontroller detects a segment, it executes the feature extraction and classification code on the buffered data.

Cycle-level chip simulations show that the algorithm spends 99.8 percent of its time executing the matched filtering and other preprocessing functions. These computations dominate both time and power consumption and are optimized by the specialized functional units. Multiple clock modes meet performance constraints by running the front-end filtering

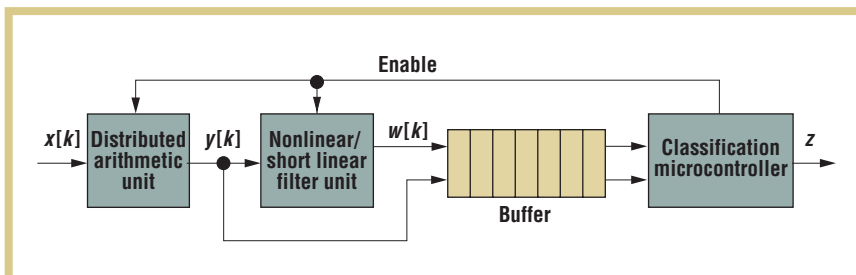


Figure 4. SensorDSP chip architecture.

Figure 5. SensorDSP heartbeat recognition performance and power scaling with data quantization.

at 1.2 kHz and back-end processing at 250 kHz. Because device leakage is not significant at the process node for this chip, we used parallel arithmetic structures in the microcontroller to minimize dynamic power.

Figure 5 shows the power versus heartbeat recognition performance trade-off from measurements of the SensorDSP chip. As the serial input data to the DA unit scales from 8 to 4 to 2 bits, the increased noise degrades recognition accuracy from about 96 to 86 percent. Power consumption when only the DA and the microcontroller are on decreases from 176 nanoWatts for 8 bits to 133 nW for 4 bits and finally to 121 nW for 2 bits. When all functional units are on, the power decreases from 299 nW for 8 bits to 239 nW for 4 bits, and lastly to 229 nW for 2 bits.

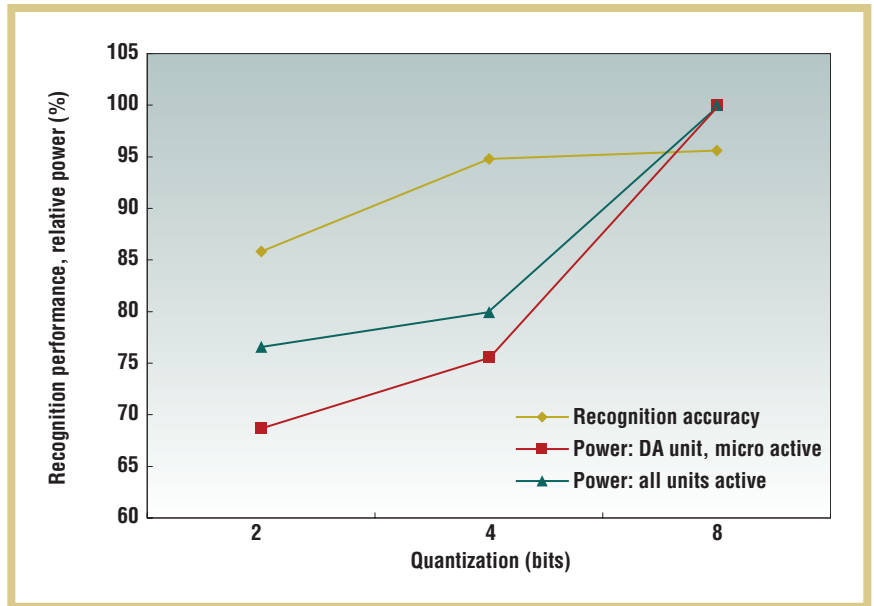
The figure shows the change in relative power as quantization is scaled and demonstrates that the fixed power of some functional units limits the system's total energy scalability. The heart rate estimation algorithm consumes 560 nW total power when running on SensorDSP. This corresponds to 26.6 picoJoules of energy dissipated per input data sample. The same algorithm running on a StrongARM SA-1100 consumes 11 μ J of energy, about six orders of magnitude more.⁸

These results demonstrate the dramatic energy efficiency gains enabled by targeted DSP architectures.

Next-generation DSPs for energy harvesting

We are exploring new circuits for next-generation energy harvesting DSPs.

Figure 6. Self-timed data path currently under development with power-on reset circuitry.

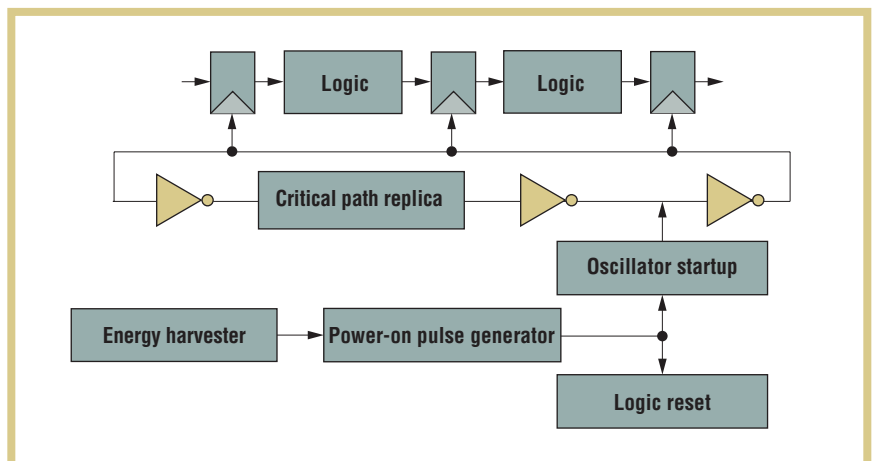


Some applications require a DC supply voltage, while others can use the AC energy harvester output voltage directly. This eliminates the need for power electronics and boosting the power available for computation. Self-timed circuits offer a way to improve performance for high-speed pipelined data paths and for low-power applications because they eliminate the need for power-hungry clock buffers and clock distribution. Self-timed circuit operation is also robust to parameter variations, including supply voltage. This robustness is what makes self-timed circuitry a promising design style for data paths in energy harvesting applications.

Self-timed circuit design

Figure 6 shows a self-timed pipeline that uses a replica critical path ring oscillator to provide the clock. The inverter chain represents the clock buffer.

In the SensorDSP chip, the critical path consists of a tree of carry chains for summing the DA outputs. To scale the approach to more complicated data paths, we can use multiplexors to add more inverter stages to the oscillator. This design style resembles traditional synchronous design in that the worst-case performance of the slowest pipe stage dictates the operational frequency. However, the ring oscillator



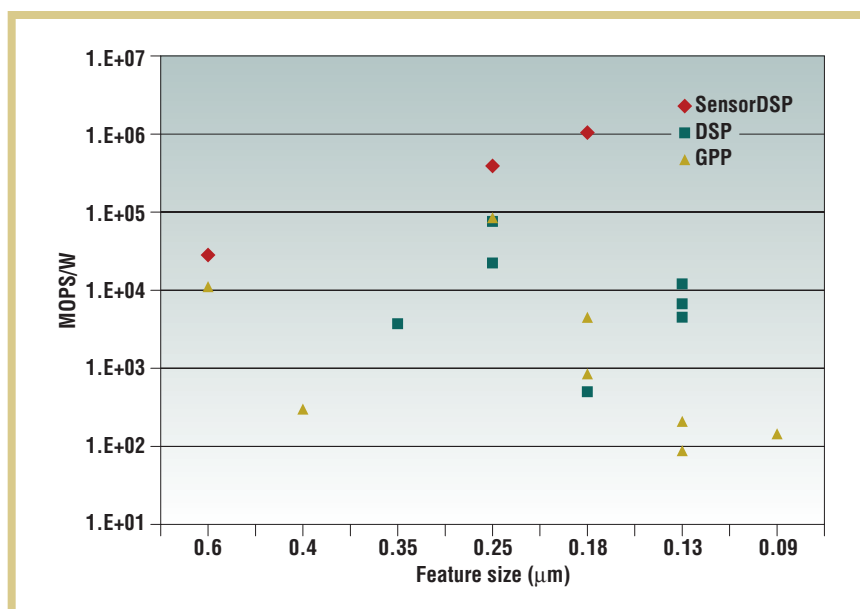


Figure 7. Energy efficiency of various processors. Specialized or application-specific architectures, such as the SensorDSP chip, exploit low peak performance requirements to achieve better energy efficiency with decreasing feature size.

Because leakage will dominate total power dissipation in the future, next-generation sensor DSPs might require reduced memory size, perhaps by increasing computation. Examples include decoding complex instruction streams and compressing data before storing it to memory.

frequency varies with supply voltage, temperature, and other variables automatically to ensure correct operation.

We've demonstrated the data path's robustness to voltage variation in simulation using a ripple-carry adder pipelined into two stages. The self-timed data path's performance will degrade as supply voltage decreases, but it will nevertheless operate correctly. In traditional synchronous designs, the designer must eliminate all types of timing violations because the clock frequency is fixed; otherwise, variations that exceed the designed tolerances will cause the data path to fail.

The self-timing structure in figure 6 includes power-on reset blocks. Because the vibration's frequency will periodically lower a vibration-based power supply's output to zero, the circuit must handle frequent power-up/power-down cycles correctly. Conventional power-on reset circuits use a resistor-capacitor network to create the delay period.⁹ We're currently designing a power-on pulse generator that does not depend on capacitance and enables functionality over a wider frequency range. We're also exploring the use of integrated dynamic RAM for storing data between energy harvester output voltage cycles.

Architectural support for energy variability


Self-timed operation introduces jitter in input sample processing. For many sensor applications, this jitter will be small compared to the system's desired sample rate, and overall signal processing performance will not suffer. The heartbeat example relies on a sample rate of 160 Hz, which is likely to be much slower than variations in self-timed circuit operation. Larger jitters lead to nonuniform sampling and require more sophisticated processing to compensate for the timing uncertainty.¹⁰ This might require more flexible functional units in the DSP, including energy-efficient implementation of adaptive filters.

Figure 5 shows that fixed power overhead limits the DSP's total energy scalability. Converting more functional units to energy-scalable processing (for example, bit-serial computation) allows a larger fraction of the DSP power to be adjusted through software. The SensorDSP chip relied on special registers to configure the input data quantization, number of matched filter taps, buffer depth, and clock frequency. These registers will proliferate as the chip design exposes more energy scalability options to the programmer.

The past decade has seen tremendous advances in low-power and energy-efficient design techniques for various processors, including DSPs. As process technology scales, power constraints will limit achievable performance and put a premium on energy-efficient architectures.¹¹ This premium will be even higher for processors in energy harvesting applications.

In general, the lower the processor's peak performance, the higher its energy efficiency. Specialized architectures are typically more efficient than general-purpose architectures. Figure 7 shows summary energy-efficiency data (expressed as millions of operations per second per watt (MOPS/W), equivalent to μJ per operation) from a number of general-purpose processors, DSPs, and the SensorDSP specialized processor. It shows SensorDSP chip data scaled from the original 0.6 μm CMOS implementation to 0.25 μm and 0.18 μm . The chip's low peak performance ensures better energy efficiency than chips targeting higher performance.

The SensorDSP has greater energy efficiency because of its specialized functional units, such as the DA module. This efficiency motivates our work on a next-generation DSP architecture that combines a reconfigurable DA array with a

low-energy microcontroller. We are targeting an energy efficiency of 10^7 MOPS/W, or 10 teraOPS/W (0.1 pJ/Op). Achieving this unprecedented level of energy efficiency will be challenging as leakage power becomes significant beyond the 130-nm process node and requires new approaches such as self-timed circuits and bit-serial arithmetic. By exploring such innovative concepts in circuits and DSP architecture, we hope to enable the next generation of energy harvesting sensors. 

ACKNOWLEDGMENTS

The US Army Research Laboratory supported this work under the Advanced Sensors Federated Lab program, contract no. DAAL01-96-2-0001. Additional funding came from a New Faculty Research Grant from the University of California, Davis.

REFERENCES

1. S. Roundy et al., "A 1.9-GHz RF Transmit Beacon Using Environmentally Scavenged Energy," paper presented at 2003 Int'l Symp. Low-Power Electronics and Design (ISLPED 03); <http://engnet.anu.edu.au/DEpeople/Shad.Roundy>.
2. R. Amirtharajah et al., "A Micropower Programmable DSP Powered Using a MEMS-Based Vibration-to-Electric Energy Converter," *Int'l Solid State Circuits Conf. (ISSCC 2000) Digest of Tech. Papers*, IEEE Press, 2000, pp. 362–363, 469.
3. A. Wang and A.P. Chandrakasan, "Energy-Efficient DSPs for Wireless Sensor Networks," *IEEE Signal Processing Magazine*, July 2002, pp. 68–78.
4. R. Amirtharajah and A. Chandrakasan, "Self-Powered Signal Processing Using Vibration-Based Power Generation," *IEEE J. Solid-State Circuits*, vol. 33, no. 5, 1998, pp. 687–695.
5. D. Rowe, "Demonstration System for a Low Power Classification Processor," master's thesis, Mass. Inst. of Technology, Feb. 2000.
6. K. Roy, S. Mukhopadhyay, and H. Mahmood-Meimand, "Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits," *Proc. IEEE*, vol. 91, no. 2, 2003, pp. 305–327.
7. S.A. White, "Applications of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review," *IEEE ASSP Magazine*, July 1989, pp. 4–19.
8. R. Amirtharajah and A. Chandrakasan, "A Micropower Programmable DSP Using Approximate Signal Processing Based on Distributed Arithmetic," *IEEE J. Solid-State Circuits*, vol. 39, no. 2, 2004, pp. 337–347.
9. T. Yasuda, M. Yamamoto, and T. Nishi, "A Power-On Reset Pulse Generator for Low Voltage Applications," *Proc. Int'l Symp. Circuits and Systems (ISCAS 01)*, vol. 4, IEEE Press, May 2001, pp. 599–601.
10. F.A. Marvasti, *A Unified Approach to Zero-Crossings and Nonuniform Sampling*, Nonuniform Pub., 1987.
11. M. Horowitz and W. Dally, "How Scaling Will Change Processor Architecture," *Int'l Solid State Circuits Conf. (ISSCC 2004) Digest of Tech. Papers*, IEEE Press, 2004, pp. 132–133.

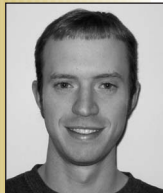
For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.



Rajeevan Amirtharajah is an assistant professor in the Electrical and Computer Engineering Department at the University of California, Davis. His research interests include low-power VLSI design for sensor applications, powering systems from ambient energy sources, and high-performance circuit and interconnect design. He received his PhD in electrical engineering from the Massachusetts Institute of Technology. He's a member of the IEEE, AAAS, and Sigma Xi. Contact him at the Dept. of Electrical and Computer Eng., Univ. of California, One Shields Ave., Davis, CA 95616-5294; ramirtha@ece.ucdavis.edu.



Jamie Collier is a graduate student in electrical engineering at the University of California, Davis. Her graduate research is in the area of low-power memory and interconnect for energy harvesting power supplies. She received her BS in electrical engineering from Case Western Reserve University. Contact her at the Dept. of Electrical and Computer Eng., Univ. of California, One Shields Ave., Davis, CA 95616-5294; jscollier@ucdavis.edu.



Jeff Siebert is a graduate student in electrical engineering at the University of California, Davis. His graduate research is in circuits for energy harvesting with a focus on pipelined data paths using AC power supplies. He received his BS in electrical engineering from Case Western Reserve University. Contact him at the Dept. of Electrical and Computer Eng., Univ. of California, One Shields Ave., Davis, CA 95616-5294; jdsiebert@ucdavis.edu.



Bicky Zhou is a post-silicon validation engineer at Intel. Her research interests focus on custom hardware design for energy scalability and power trade-offs between serial and parallel computations as CMOS technology scales. She received her MS in electrical and computer engineering from the University of California, Davis. Contact her at 900 Jackson St., Apt.105, Oakland, CA 94607; bicky.b.zhou@intel.com.



Anantha P. Chandrakasan is a professor of electrical engineering and computer science at the Massachusetts Institute of Technology and a cofounder and chief scientist at Engim, a company focused on high-performance wireless communications. His research interests include the ultra-low power implementation of custom and programmable digital signal processors, distributed wireless sensors, ultra-wideband radios, and emerging technologies. He received his PhD in electrical engineering and computer sciences from the University of California, Berkeley. Contact him at MIT, 50 Vassar St., 38-107, Cambridge, MA 02139; anantha@mtl.mit.edu.