

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/224252953>

# An Accelerator-Based Wireless Sensor Network Processor in 130 nm CMOS

Article in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* · July 2011

DOI: 10.1109/JETCAS.2011.2160751 · Source: IEEE Xplore

---

CITATIONS

29

---

READS

84

3 authors, including:



David C Brooks

Brigham and Women's Hospital

283 PUBLICATIONS 12,893 CITATIONS

[SEE PROFILE](#)



Gu-Yeon Wei

Harvard University

234 PUBLICATIONS 7,690 CITATIONS

[SEE PROFILE](#)

# An Accelerator-Based Wireless Sensor Network Processor in 130nm CMOS

Mark Hempstead<sup>\*</sup>  
ARM Ltd  
Cambridge, United Kingdom  
mhempste@eecs.harvard.edu

Gu-Yeon Wei, David Brooks  
School of Engineering and Applied Sciences  
Harvard University  
Cambridge, MA U.S.A  
{guyeon,dbrooks}@eecs.harvard.edu

## ABSTRACT

Networks of ultra-low-power nodes capable of sensing, computation, and wireless communication have applications in medicine, science, industrial automation, and security. Over the past few years, deployments of wireless sensor networks (WSNs) have utilized nodes based on off-the-shelf general purpose microcontrollers. Reducing power consumption requires the development of System-on-chip (SoC) implementations that must provide both energy efficiency and adequate performance to meet the demands of the long deployment lifetimes and bursts of computation that characterize WSN applications. This work takes a holistic approach and, thus, studies all layers of the design space, from the applications and architecture, to process technology and circuits.

This paper introduces the emerging application space of wireless sensor networks and describes the motivation and need for a custom system architecture. The proposed design fully embraces the accelerator-based computing paradigm, including acceleration for the network layer (routing) and application layer (data filtering). Moreover, the architecture can disable the accelerators via VDD-gating to minimize leakage current during the long idle times common to WSN applications. We have implemented a system architecture for wireless sensor network nodes in 130nm CMOS. It operates at 550 mV and 12.5 MHz. Our system uses 100x less power when idle than a traditional microcontroller, and 10-600x less energy when active.

## Categories and Subject Descriptors

C.1.3 [Processor Architectures]: Other Architecture Styles—*Heterogenous (hybrid) systems*; C.3 [Special-Purpose and Application-Based Systems]: Real-time and embedded systems

## General Terms

Design, Measurement

<sup>9</sup>\*This work was completed while the author worked toward his PhD at Harvard University. Currently at ARM Ltd., Dr. Hempstead will begin a faculty position at Drexel University in January 2010.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CASES'09, October 11–16, 2009, Grenoble, France.

Copyright 2009 ACM 978-1-60558-626-7/09/10 ...\$10.00.

## Keywords

Wireless Sensor Networks, Ultra-Low Power, Accelerator-based

## 1. INTRODUCTION

Networks of ultra-low-power nodes that include sensing, computation, and wireless communication have applications in medicine, science, industrial automation, and security. System-on-chip (SoC) implementations of such nodes can provide both energy efficiency and adequate performance to meet the long deployment lifetimes and bursts of computation that characterize wireless sensor network (WSN) applications.

Phenomena	Sample Rate (in Hz)
<i>Very Low Frequency</i>	
Atmospheric temperature	0.017 - 1
Barometric pressure	0.017 - 1
<i>Low Frequency</i>	
Heart rate	0.8 - 3.2
Volcanic infrasound	20 - 80
Natural seismic vibration	0.2 - 100
<i>Mid Frequency (100Hz - 1000 Hz)</i>	
Earthquake vibrations	100 - 160 Hz
ECG (heart electrical activity)	100 - 250
<i>High Frequency (&gt; 1kHz)</i>	
Breathing sounds	100 - 5k
Industrial vibrations	40k
Audio (human hearing range)	15 - 44k
Audio (muzzle shock-wave)	1M
Video (digital television)	10M

Table 1: Sensor Sampling Rates of Different Phenomena

Deployed sensor networks measure a wide range of phenomena including atmospheric temperature, heart rate, volcanic eruptions, and even the sound of a sniper rifle[19, 4, 20, 3]. The performance target (cycles of computation per second) for a WSN node is set by the sampling rate for the measured phenomena and the amount of on-node data filtering required. Table 1 lists the range of sampling rates for different physical phenomena. Environmental measurements—such as temperature and pressure—have time constants on the order of minutes. Consequently, nodes deployed to measure low-frequency phenomena will be idle most of the time. In contrast, nodes that measure higher-frequency phenomena—such as seismic vibrations and acoustic signals—will require higher performance processors.

Application Domain	Desired Lifetimes	Computation Re-requirements (sample rates)	Example
<b>Scientific Applications</b>			
Habitat/Weather Monitoring	Months/decades	very low	Great Duck Island[19]
Volcanic Eruption Detection	Months/decades	mid	Volcano WSN[20]
<b>Military and Security Applications</b>			
Building/Border Intrusion Detection	Years/decades	low	
Structural and Earthquake monitoring	Years/decade	low/mid	
Active Battlefield Sensing	Months	mid/high	Sniper Detection/Localization[3]
<b>Medical Applications</b>			
Long-Term Health Monitoring (pulse)	Days	low	
Untethered Medical Instruments (ECG)	Days	med	EKG mote[4]
<b>Business Applications</b>			
Supply Chain Management	Months	low	
Expired/Damaged Goods Tracking	Months	low	
Factory/fab monitoring	Months/years	med/high	Industrial WSN[10]

Table 2: Example WSN application domains.

Sensor nodes are sometimes deployed in hard to reach places, which make it difficult and expensive to change batteries regularly. In this work, we classify node lifetime based on the availability of wired power sources or battery replacements. In some domains, such as military and security applications, nodes embedded deeply in the structure of a building would be difficult to manually maintain and would consequently require node lifetimes of several years on one battery. In medical domains (not including bio-implants) a patient or health care professional would be able to replace batteries daily. Table 2 lists a few example application domains with an estimate of their deployment lifetimes and computation requirements.

In this work, we target a class of habitat monitoring WSN applications that aim for long deployment lifetimes and that incorporate data filtering and multihop routing on the nodes. Specifically, this architecture was informed by the volcano monitoring system deployed by Werner-Allen et al. [5]. In that system, nodes sampled both seismic and infrasound signals and use an exponentially weighted moving average (EWMA) filter to detect interesting events and transmit data back to a team of vulcanologists.

Proposed SoC implementations for WSNs typically rely on general-purpose microcontrollers as the main compute engine and often run in subthreshold to minimize energy [15]. Unfortunately, subthreshold operation increases susceptibility to on-die parameter variations, limits the performance needed for real-time applications, and requires custom SRAM design [9]. In order to accommodate the wide variety of computing needs in WSNs while minimizing energy consumption, we propose an accelerator-based system architecture.

Our design fully embraces the accelerator-based computing paradigm, including acceleration for the network layer (routing) and application layer (data filtering). Moreover, our architecture can disable the accelerators via VDD-gating to minimize leakage current during the long idle times common to WSN applications. We show that the accelerator-based system architecture, implemented in 130nm CMOS, significantly improves energy efficiency and performance of computations when compared to a general-purpose microcontroller for a variety of WSN benchmarks.

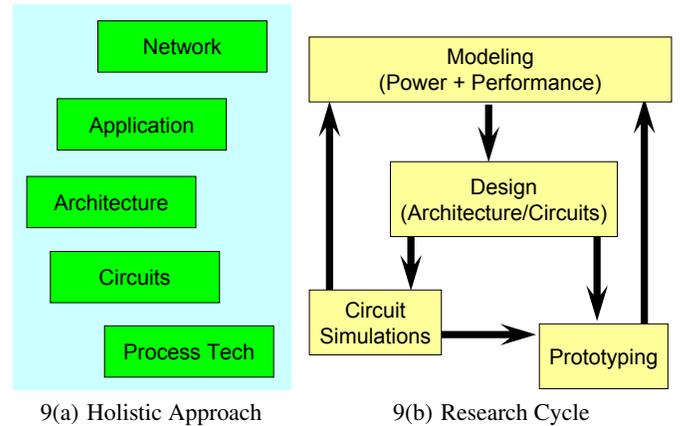


Figure 1: Research Approach. We take a holistic approach to research — understanding and addressing power consumption at all layers of the design space. Architecture innovations are informed by modeling and prototyping.

## 2. HOLISTIC APPROACH TO LOW POWER DESIGN

During the course of our research, we have taken the view that all layers of the design space influence power consumption, from the application and network to the architecture and circuits. Figure 1 provides a graphical description of the research approach we employed. Our research efforts follow an iterative approach through modeling, design and prototyping and our models incorporate inputs from a variety of design layers. For example, the PowerTOSSIM model accepts inputs from the network and application layers and physical power measurements of nodes[18].

We use modeling to guide design decisions which are verified by circuit simulations and prototyping. Section 3 describes a design motivated by the modeling of application behavior and addresses leakage current, which is increasing due to technology scaling. Be-

cause our power consumption targets are so low, we developed a prototype in 130nm CMOS to verify that our design achieves ultra low power operation. Taking a holistic approach to design allowed us to include features (such as hardware acceleration and VDD-gating) that required coordination across layers.

### 3. ARCHITECTURE

The system architecture combines the energy efficiency found in application specific integrated circuits (ASICs) with the flexibility and programmability of a general purpose processor. As power consumption is the main design constraint, the proposed event-driven system for WSNs uses three techniques to reduce power consumption.

- *Lightweight event handling in hardware* - Initial responsibility for handling incoming interrupts is given to a specialized Event Processor, removing the software overhead that would be required to provide event handling on a general-purpose processor.
- *Hardware acceleration for typical WSN tasks* - Modular hardware accelerators are included to complete regular application tasks such as data filtering and message routing.
- *Application-controlled fine-grained VDD-gating* - Addressing leakage current with architecture support for VDD-gating enables accelerator blocks to be powered off when unused.

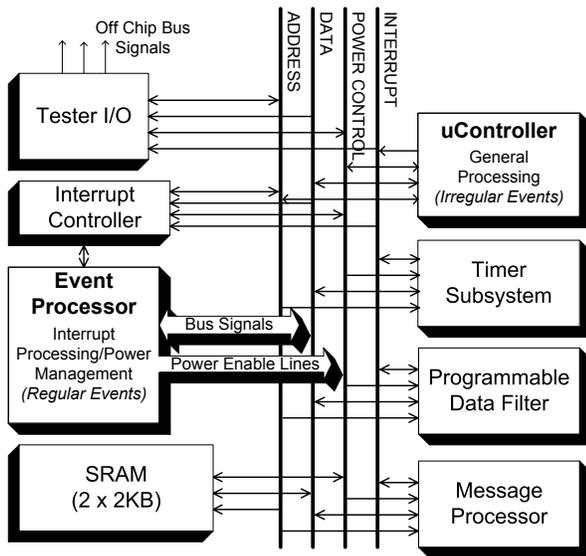


Figure 2: System Block Diagram.

Fig. 2 presents a block diagram of the prototype chip. The Event Processor (EP) is a small programmable state machine that runs interrupt service routines (ISRs) to control the flow of data between the on-chip memory and multiple accelerators, such as the message processor, programmable data filter, and timer, which are memory mapped and connected via the system bus [7]. The EP also acts as a power manager, turning accelerators on and off as needed by the running application. While the system also includes an 8-bit general-purpose microcontroller to handle infrequent and irregular tasks, it can usually be disabled. During long idle times, only the

EP—and perhaps select blocks such as the timer—must be powered. The tester I/O block facilitates testing to verify functionality.

A key benefit of the modular design of the architecture is its ability to employ fine-grained power management of individual components (both masters and accelerators). Selectively turning off components and using VDD-gating enables the system to minimize leakage power. For example, the general-purpose microcontroller core could be relatively complex and power-hungry when active, but can be VDD-gated most of the time when idle. The event processor handles all interrupts, distributes tasks to accelerator devices, and wakes up the microcontroller only rarely, when necessary.

### 4. IMPLEMENTATION

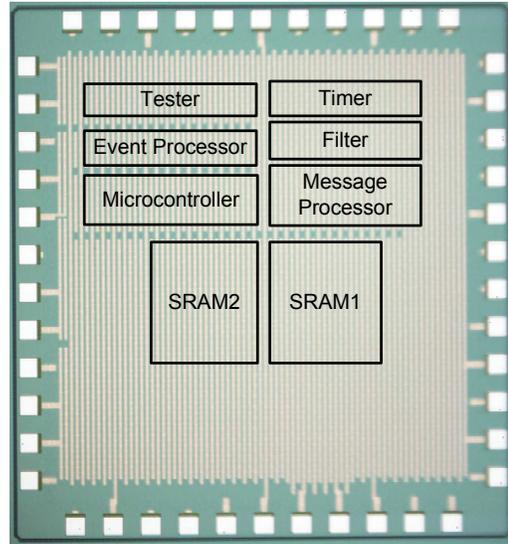


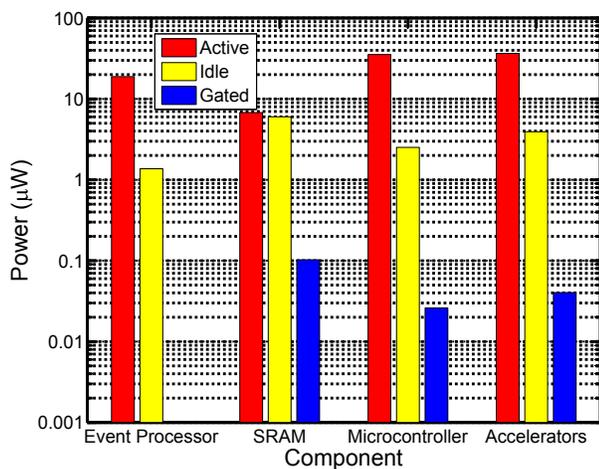
Figure 3: Die Photograph of the Prototype. System includes an event processor and several accelerators for regular operation. The system has been realized in 130nm CMOS on a 2mm x 2mm die.

Because the architecture is new and the power consumption targets are aggressive, physical measurements are necessary to verify that the architecture meets our goals. The chip was manufactured in a 130nm bulk CMOS process with eight layers of metal. A die photograph is shown in Figure 3. The system contains 444,982 transistors including 4KB of foundry-supplied SRAM. The chip area is pad limited due to the large number of pins purposely added for testing and the fine-grain power measurements of nine different power domains. Decoupling-capacitors were included on all of the top level power domains as well as the virtual power domains separated by VDD-gating transistors.

### 5. MEASUREMENTS

Our first experimental measurements have verified reliable operation across a range of lower clock frequencies—25 kHz to 12.5 MHz—that are suited to the low power needs of WSN applications. SRAM reliability limits the minimum operating voltage to 450mV.

Fig. 4 plots the per-block power consumption of the system, running custom microbenchmarks written to exercise each block in three operating modes - active (12.5MHz @550mV), idle (0MHz @550mV), and powered off (VDD-gated). VDD-gating reduces the power consumption of individual blocks by 50-100x, which



**Figure 4: Per Block Power Consumption 550 mV and 12.5 MHz.**

helps to minimize power consumption during long periods of inactivity. The event processor block cannot be VDD-gated since it must always be available to handle interrupts. The accelerator blocks consume more power when fully active than the microcontrollers but, as shown in Section 7, the more computationally efficient accelerators lead to energy savings.

In Section 6, we compare our prototype to nine processors for WSNs in the existing literature. Because the commonly used metric of energy-per-instruction cannot be easily applied to accelerator-based systems, we introduce the concept of *energy-per-task*. We defined a task as a collection of dependent computations that are executed periodically. We present measurements of a task similar in nature to the volcano monitoring application. This task takes 131 cycles to execute and ultimately consumes 678.9 pJ at 550 mV and 12.5 MHz. An equivalent routine written for the Mica2 mote requires 1532 instructions. Using this information, we compute the *energy per equivalent instruction* as 0.44 pJ, which is significantly lower than systems in the literature: the lowest energy systems—general purpose cores operating in subthreshold—consume 2-3 pJ per instruction.

## 6. COMPARISON TO RELATED WORK

Several research groups have recognized the need for ultra-low power systems designed specifically for wireless sensor networks. The systems differ significantly because of the architecture decisions and circuit techniques used to implement the system. For example, several systems are based around a traditional general purpose core but the circuits are designed to operate in subthreshold – trading-off performance for reduced power consumption. Our work operates above threshold but uses hardware acceleration to increase energy efficiency. First, we categorized systems based on the circuit techniques employed to improve energy consumption.

- *Subthreshold operation* - By using a power supply less than the threshold voltage, systems such as the Subliminal and Phoenix processors from the University of Michigan and a subthreshold MSP430 from MIT are able to trade off performance for reduced active power consumption[11, 14, 13, 16, 6, 21].
- *Asynchronous Circuits* - Processors such as SNAP from Cor-

nell University eliminate clock power by relying on asynchronous circuits[1, 2].

- *Power Supply Gating* - To address increasing leakage current, the Charm processor, from the University of California at Berkeley, and our work employ transistors that switch the power supplies of unused blocks[17, 7].

Along with different circuit techniques, designers of WSN processors differ in their approach to architecture support for applications.

- *General Purpose Computation* - Off-the-shelf and custom designed systems employ load-store or accumulator based processors as the core processing engine of the system.
- *Application Acceleration* - our work and the Charm processor from University of California at Berkeley provide hardware acceleration for common tasks to reduce active energy consumption and increase system performance.

We tabulated key parameters for each of the discussed systems including circuit techniques, architecture style, datapath width, throughput and energy per instruction. Figure 5 presents the results of the tabulation. The processors at the top (Atmel ATmega128L, TI-MSP430) are off-the-shelf microcontrollers included in commercially available WSN nodes such as the Mica2. The remaining processors are prototype systems designed specifically for WSN applications.

From Figure 5, we observe a relationship between the use of subthreshold operation and the performance and energy consumption of the system. All of the systems that operate in subthreshold are limited to clock frequencies less than 1 MHz. However, the low supply voltage results in a low energy per instruction between 2 and 4 pJ. Our system uses transistor switches more efficiently through hardware acceleration. Consequently, our system has the lowest measurement of energy per equivalent instruction when the accelerators are used (0.44 pJ). For irregular tasks that employ the general purpose microcontroller, our system has a higher energy per instruction than the systems operating in subthreshold (3.4 pJ). As our per-block power measurements show, SRAMs can consume a dominant fraction of total energy consumption. Consequently, systems that contain larger memories (greater than a few KB) consume larger amounts of energy compared to similar systems at the same voltage, frequency and architecture.

Unfortunately, standard benchmark suites do not exist for the WSN space, though a few research groups have proposed some possibilities [8, 12]. Without running the same application on each system, it is not possible to judge the programmability, energy efficiency, and performance of the different systems fairly. The efficacy of the energy per instruction metric to compare different systems has been questioned before, but in this case, it could actually lead to completely misleading conclusions. The notion of an *instruction* is lost on both the Charm processor and our system because most of the processing is handled by custom hardware accelerators. Even among the general-purpose architectures, selecting the most energy efficient architecture is not an easy decision due to the differences in instruction set architectures (ISAs), process technologies, memory sizes, and clock frequencies. Also, WSN applications often experience long periods of inactivity. Consequently, we must consider the power consumption of the system while idle — which is not captured by the energy per instruction metric. In an effort to make a more fair and accurate comparison, we compare our system to a general purpose architecture in the next section.

System	Arch Style	DP width (bits)	Circuit Techniques	Memory (KB)	Process (nm)	VDD (V)	Clk Freq (MHz)	Energy (pJ/ins)
Atmel ATmega128L	General Purpose (GP)	8	NA	4	350	3.0	7.3	3200
TI MSP430	GP	16	NA	10	NA	3.0	8.0	750
S. Hanson, JSSC'08	GP	8	Subthreshold	0.3125	130	0.35	0.354	3.5
B. Zhai, VLSI'06	GP	8	Subthreshold	0.25	130	0.36	0.833	2.6
Phoenix, VLSI'08	GP	8	Near-Threshold	0.41	180	0.5	0.106	2.8
J. Kwong, ISSCC'08	GP	16	Subthreshold	128	65	0.5	0.434	27.3
Charm, VLSI'06	GP + protocol	NA	Two power domains	68	130	1,0.3	8.0	96
SNAP, ASPLOS'04	GP	16	Asynchronous	8	180	1.8 0.6	200 M 23	218 24
SmartDust, ISSCC'04	GP	8	Two Clocks	3.125	250	1.0	0.500	12
<b>THIS WORK</b>	Accelerator	8	VDD-Gating	4	130	0.55	12.5	0.44
	GP							3.4

Figure 5: Comparison to Other Systems Designed for WSN.

## 7. COMPARISON TO GENERAL PURPOSE

As stated in the previous section, the metric of energy per instruction does not isolate the benefits of an accelerator-based architecture from the process technology, circuit implementation, and amount of SRAM. Thus, we compare the cycle count and energy of full applications running on accelerators to applications running on the on-die general-purpose microcontroller. These applications combine data filtering, outgoing message preparation, and flood-based message routing, which are prototypical WSN routines. We analyze routines for data filtering (EWMA and threshold); network routing using a CAM structure; recording an outgoing message; detecting an incoming irregular message; and automatically relaying a regular message. The on-die Z80 microcontroller closely resembles 8-bit architectures employed in other WSN SoCs. For fairness, all routines were written in assembly and hand-tuned for accelerator- and microcontroller-based operation, respectively. Fig. 6 presents the cycle count of each routine for both scenarios. Accelerators process data in parallel and include simplified decode logic, enabling the speedups. Multiple points for a particular routine reflect different inputs that yield different performances. Accelerator implementations see cycle speedups from 15 to 635x, which directly translate in to energy savings. Through measurements of energy consumption for each of the routines, it was found that hardware accelerators consume 1/10th to 1/600th the energy consumed by software-based routines running on the microcontroller. Energy savings is greater at higher frequencies because VDD-gating can reduce leakage during the longer idle times afforded by hardware acceleration.

## 8. WORKLOAD ANALYSIS AND DVFS

By incorporating the concept of workload in our analysis, we bring together all features of the architecture (speedup, energy efficiency, VDD-gating) and calculate total energy consumption. As detailed in Section 1, the amount of computation required to execute an application varies by orders of magnitude depending on the phenomena being sensed, the amount of computation required, and the complexity of the operation. System clock frequency and

supply voltage also depend on the computation requirements of the workload. Idle leakage power consumption between tasks was not captured in the analysis of speedup but can be a large contributor of total power consumption.

Building on individual characterizations in Section 5, we compare compute-block power consumption for different workload requirements and include idle power in our analysis. In order to clarify the comparison, these results exclude additional system power overheads (e.g., EP and SRAM) common to both types of systems. WSN workload intensity varies significantly depending on the observed phenomena—from 1 task/minute for weather observations to  $> 10^5$  tasks/second for high-frequency data collection. While most workload requirements are low, sometimes short bursts of high-performance, time-sensitive activity are followed by long idle times (e.g., bursty seismic activity preceding a volcanic eruption).

Figure 7 plots the average power consumption of routines run on either the accelerators or the microcontroller while varying workload intensity. For each datapoint, the lowest power voltage/frequency operating point was chosen. For light workloads ( $< 10$  tasks/sec), the system can operate at the lowest voltage and frequency (450mV, 25 KHz) and power consumption is dominated by leakage current. For medium-intensity workloads ( $10^4$  tasks/sec), using accelerators provides 1000x power savings due to a 635x speedup in cycle counts and a 50% lower supply voltage. As workload increases, active power dominates until the clock frequency required by the microcontroller reaches the performance limit of the system at the maximum supply voltage of 1.2V. Routines run on the accelerator can operate up to  $10^7$  tasks per second with a voltage less than 1.1 V. The plot also demonstrates that, VDD-gating lowers the power consumption for both scenarios under light loads, but the accelerators' higher inherent performance enables VDD-gating for longer periods of time that translate to additional power savings.

## 9. CONCLUSION

To explore some of the microarchitecture challenges of accelerator-based designs, we developed a system architecture and prototype processor for wireless sensor network (WSN) applications. Our

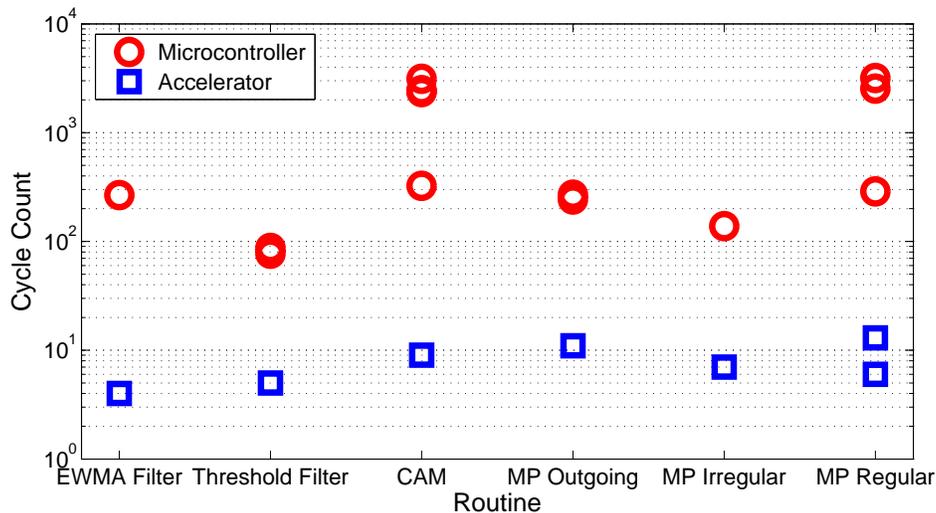


Figure 6: Cycle Count Comparison

system architecture includes a set of hardware accelerators for typical WSN tasks and an event processor to facilitate communication and power management among the accelerator devices. Application support for VDD-gating was included because leakage current can dominate the total power consumption of some WSN applications with long idle times. We constructed a prototype in 130nm CMOS and measured the power consumption of each major functional block.

We found that evaluating the efficacy of our architecture and comparing it to related systems required the creation of several new metrics and analysis methodologies. Compared to similar microprocessors proposed for WSNs, our system has the lowest energy per equivalent instruction (0.44 pJ). Many designers of related systems based their design around a general purpose computing engine. We compared routines implemented on the accelerators with routines implemented in software on the same chip and showed a 15x-635x performance improvement and a 10x-600x energy savings. Because the performance requirements of WSN nodes vary widely, we conducted an analysis of our system while sweeping workload and scaling voltage and frequency. The results of this analysis show that our system architecture sees a reduction of 100x for low intensity workloads due to VDD-gating. Due to a combination of hardware acceleration and voltage scaling, our system sees 10x-1000x power reduction over general purpose based designs for medium-intensity workloads. At high-intensity workloads when the general purpose microcontroller does not have the performance to meet user demand, the accelerators are able to execute the higher performance application. The system provides efficient computation through hardware acceleration for habitat monitoring applications. The modular architecture and event processor enable the management of idle power through VDD-gating.

Because wireless sensor networks nodes are often untethered from power grids, the WSN application class has stricter power constraints than desktop or mobile environments. Taking a holistic approach to design, this work addressed power consumption at the application, architecture and circuit design layers. We have shown that an architecture based around specialized accelerators can reduce the amount of energy per routine by several orders of magnitude and that leakage power can be managed through the VDD-gating of the modular accelerators. As technology continues to

scale, mobile and desktop processors will need to incorporate an increasing amount of specialization to maintain growth in microprocessor performance. In the future, we aim to adapt our holistic approach and lessons learned from building this prototype to the mobile, and server computing markets.

## 10. REFERENCES

- [1] V. Ekanayake, C. Kelly, and R. Manohar. An ultra low-power processor for sensor networks. In *Proceedings of the 11th international conference on Architectural support for programming languages and operating systems*, Boston, MA, Oct 2004.
- [2] V. Ekanayake, C. Kelly, and R. Manohar. BitSNAP: Dynamic Significance Compression For a Low-Energy Sensor Network Asynchronous Processor. In *Proceedings of the 11th International Symposium on Asynchronous Circuits and Systems*, March 2005.
- [3] G. S. et al. Sensor network-based countersniper system. In *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys'04)*, Baltimore, MD, Nov 2004.
- [4] T. R. F. Fulford-Jones, G.-Y. Wei, and M. Welsh. A Portable, Low-Power, Wireless Two-Lead EKG System. In *Proceedings of the 26th IEEE EMBS Annual International Conference*, San Francisco, CA, Sept 2004.
- [5] G. Werner-Allen et al. Fidelity and yield in a volcano monitoring sensor network. In *Symposium on Operating Systems Design and Implementation (OSDI)*, November 2006.
- [6] S. Hanson, B. Zhai, M. Seok, B. Cline, K. Zhou, M. Singhal, M. Minuth, J. Olson, L. Nazhandali, T. Austin, D. Sylvester, and D. Blaauw. The Phoenix Processor: A 30pW Platform for Sensor Applications. In *IEEE Symposium on VLSI Circuits (VLSI-Symp)*, June 2007.
- [7] M. Hempstead, N. Tripathi, P. Mauro, G.-Y. Wei, and D. Brooks. An ultra low power system architecture for sensor network applications. In *The 32nd Annual International Symposium on Computer Architecture (ISCA)*, June 2005.
- [8] M. Hempstead, M. Welsh, and D. Brooks. TinyBench: The

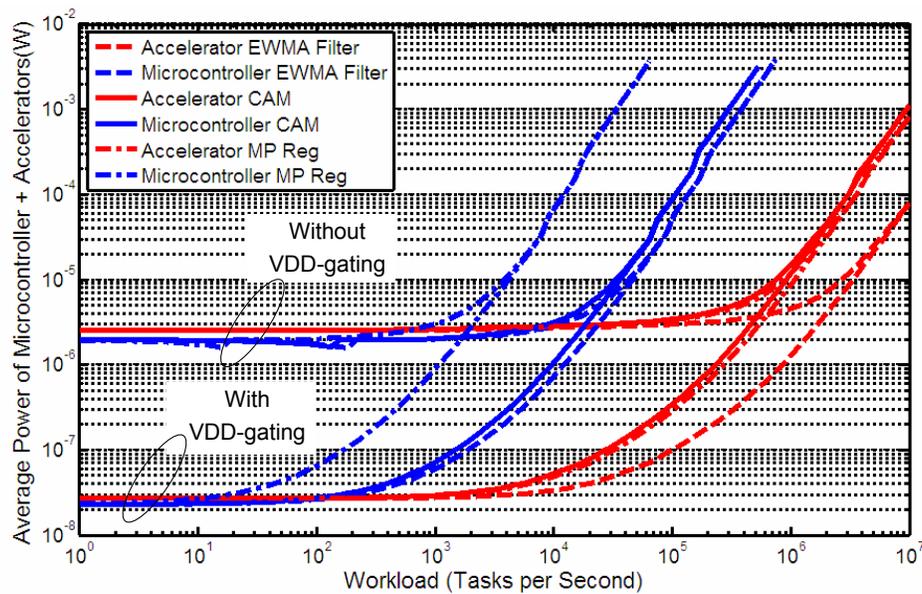


Figure 7: Power vs. Workload

Case For A Standardized Benchmark Suite for TinyOS Based Wireless Sensor Network Devices. In *Proceedings of the First IEEE Workshop on Embedded Networked Sensors (EmNets'04)*, Tampa, FL, Nov 2004.

- [9] J. Kwong et al. A 65nm Sub- $V_t$ , Microcontroller with Integrated SRAM and Switched-Capacitor DC-DC Converter. In *IEEE International Solid-State Circuits Conference (ISSCC)*, February 2008.
- [10] L. Krishnamurthy, R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, N. Kushalnagar, L. Nachman, and M. Yarvis. Design and deployment of industrial sensor networks: Experiences from a semiconductor plant and the north sea. In *Proceedings of the Third ACM Conference on Embedded Networked Sensor Systems (SenSys'04)*, San Diego, CA, Nov 2005.
- [11] J. Kwong, Y. Ramadass, N. Verma, M. Koesler, K. Huber, H. Moormann, and A. Chandrakasan. A 65nm Sub- $V_t$ , Microcontroller with Integrated SRAM and Switched-Capacitor DC-DC Converter. In *IEEE International Solid-State Circuits Conference (ISSCC)*, February 2008.
- [12] L. Nazhandali, M. Minuth, and T. Austin. Sensebench: toward an accurate evaluation of sensor network processors. In *IEEE International Workload Characterization Symposium*, October 2005.
- [13] L. Nazhandali, M. Minuth, B. Zhai, J. Olson, S. Hanson, T. Austin, and D. Blaauw. A second-generation sensor network processor with application-driven memory optimizations and out-of-order execution. In *ACM/IEEE International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES)*, September 2005.
- [14] L. Nazhandali, B. Zhai, J. Olson, A. Reeves, M. Minuth, R. Helfand, S. Pant, T. Austin, and D. Blaauw. Energy optimization of subthreshold-voltage sensor network processors. In *The 32nd Annual International Symposium on Computer Architecture (ISCA)*, June 2005.
- [15] S. Hanson et al. Exploring variability and performance in a sub-200-mv processor. *IEEE Journal of Solid-State Circuits*, 43(4):881–891, April 2008.
- [16] M. Seok, S. Hanson, Y.-S. Lin, Z. Foo, D. Kim, Y. Lee, N. Liu, D. Sylvester, and D. Blaauw. Performance and Variability Optimization Strategies in a Sub-200mV, 3.5pJ/inst, 11nW Subthreshold Processor. In *IEEE Symposium on VLSI Circuits (VLSI-Symp)*, June 2008.
- [17] M. Sheets, F. Burghardt, T. Karalar, J. Ammer, Y. Chee, and J. Rabaey. A power-managed protocol processor for wireless sensor networks. In *Proc. VLSI*, June 2006.
- [18] V. Shnyder, M. Hempstead, B.-R. Chen, G. W. Allen, and M. Welsh. Simulating the Power Consumption of LargeScale Sensor Network Applications. In *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys'04)*, Baltimore, MD, Nov 2004.
- [19] R. Szweczyk, J. Polastre, A. Mainwaring, and D. Culler. Lessons from a sensor network expedition. In *Proc. the First European Workshop on Wireless Sensor Networks (EWSN)*, January 2004.
- [20] G. Werner-Allen, K. Lorincz, J. Johnson, J. Less, and M. Welsh. Fidelity and yield in a volcano monitoring sensor network. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, November 2006.
- [21] B. Zhai, L. Nazhandali, J. Olson, A. Reeves, M. Minuth, R. Helfand, S. Pant, D. Blaauw, and T. Austin. "a 2.60pJ/inst subthreshold sensor processor for optimal energy efficiency. In *IEEE Symposium on VLSI Circuits (VLSI-Symp)*, June 2006.