# Phoenix: an Ultra-Low Power Processor for Cubic Millimeter Sensor Systems

Mingoo Seok, Scott Hanson, Yu-Shiang Lin, Zhiyoong Foo,
Daeyeon Kim, Yoonmyung Lee, Dennis Sylvester, David Blaauw
University of Michigan Ann Arbor
mgseok@umich.edu

## ABSTRACT

An integrated platform for sensor applications, called the Phoenix Processor, is implemented in a 0.18µm CMOS process with an area of 915x915µm$^2$, making on-die battery integration feasible. The Phoenix Processor uses a comprehensive standby strategy with a unique power gating approach, custom low leakage memory cells, low voltage ROM, CPU with compact ISA, data memory compression, and adaptive data memory leakage management. Measurements show that the Phoenix Processor consumes on average 29.6pW in standby mode and 2.8pJ/cycle in active mode.

**Categories and Subject Descriptors**
B.7 Hardware [**Integrated Circuits**]: General
**General Terms:** Design
**Keywords: s**ubthreshold circuits, sensor system, low power

## 1. INTRODUCTION

One cubic millimeter systems will deliver enormous value to a wide range of wireless sensor applications. While general environmental sensor systems will benefit from the lower costs associated with small volume, cubic millimeter systems will be particularly beneficial for medical implants such as intra-ocular pressure sensors [1][2] by making surgery less invasive.

Although circuit component size can be reduced due to advances in semiconductor technology, the required battery size for a target lifetime is the major bottleneck to achieving one cubic millimeter system volume. Given a 1mm$^2$ zinc/silver battery with a capacity of 100uAh/cm$^2$ and output voltage of 1.55V [2], system power consumption should be limited to 177pW to guarantee one year of battery life.

This power source limitation provides motivation for a system that consumes extremely little power. This paper extends upon the state-of-the-art by presenting an ultra-low power system, called the Phoenix Processor, which consumes only 29.6pW in standby mode. With such low power demands, a 1mm$^2$ battery could supply power for a multi-year lifetime [3]. The system includes data RAM (DMEM), instruction RAM (IMEM), instruction ROM (IROM), CPU, power management unit (PMU), watchdog timer, and temperature sensor.

The organization of this paper is as follows: we discuss a system overview in Section 2 and more details about each component in Section 3. In Section 4, design and testing methodology is described. After presenting measurement results in Section 5, we finally conclude in Section 6


*DAC 2009*, June 26–31, 2009, San Francisco, California, USA

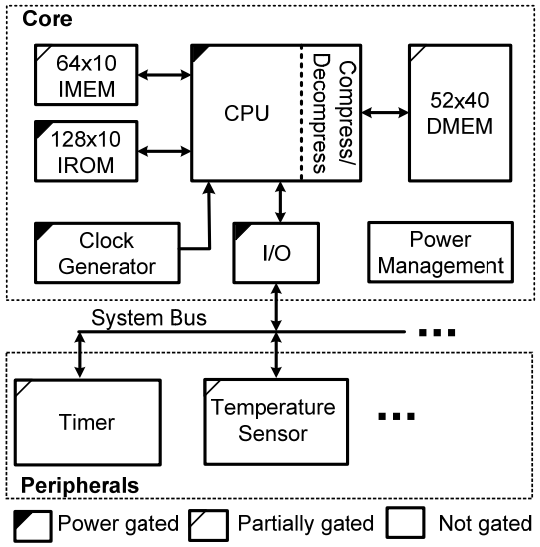




**Figure 1. The Phoenix Processor**

## 2. System Overview

As shown in Figure 1, the Phoenix Processor is divided into core and peripherals. The core which serves as a parent to the peripherals consists of an 8-bit CPU, a 52x40-bit DMEM, a 64x10-bit IMEM, a 64x10-bit IROM and PMU. The peripherals include a watchdog timer and a temperature sensor, but can be extended up to 8 devices for sensing system requiring additional functionality. The core and peripheral devices communicate over a system bus using a simple asynchronous protocol.

In typical operating conditions, the Phoenix Processor spends an extended period of time in standby mode (e.g., 10 minutes). The watchdog timer which is a 0.9pW current-starved oscillator wakes up the Phoenix Processor. Once awake, the Phoenix Processor accesses the temperature sensor and runs a short routine to process and store the measurement. After completing the data processing routine, the Phoenix Processor returns to standby mode and wait for future wakeup.

## 3. Ultra-low Standby Power Design Strategy

Recent work [4][11][12][13] has explored aggressive $V_{dd}$ scaling for reducing active energy but has overlooked the power consumed during standby periods, which can be >99% of the lifetime. Therefore, in addition to aggressive voltage scaling, the Phoenix Processor leverages a comprehensive standby strategy. In this section, various architectural and circuit techniques for minimizing standby power in each module of the Phoenix Processor will be discussed.

### 3.1 Power Gating Switch

One of the critical pieces of our standby strategy is the unique approach to power gating as shown in Figure 2. At nominal

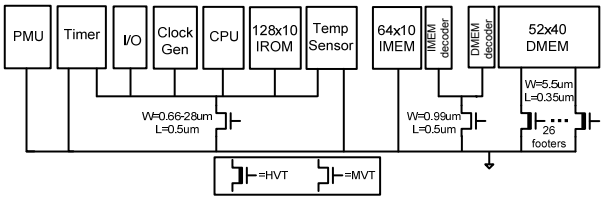



**Figure 2. Power gating switch allocation**

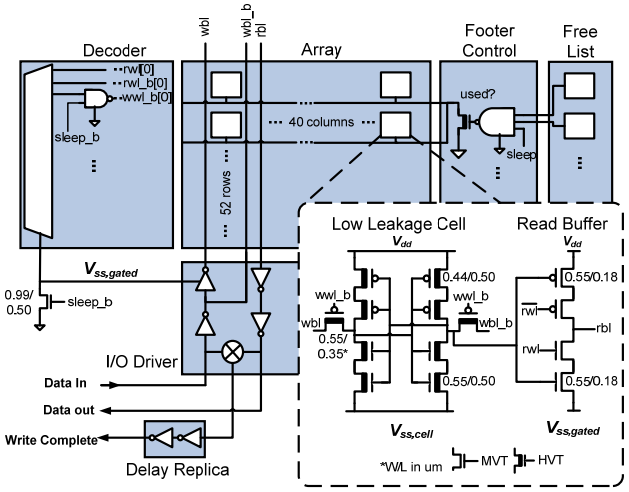



**Figure 3. DMEM Architecture with 7.1fW SRAM Cell**

supply voltage, high-Vt devices are typically used as power gating switches since they deliver on-current that is comparable to the nominal device while also delivering exponentially smaller off-current. Additionally, a wide power gating switch is typically used to minimize the performance penalty due to the voltage drop across the switch. Our approach for low $V_{dd}$ power gating switch differs in two ways. First a medium-Vt power gating switch is used instead of a typical high-Vt switch since high-Vt switches cannot meet the current demand in the low $V_{dd}$ regime due to the exponential dependency of on-current on Vt. For the same on-current, a high-Vt switch must be sized up exponentially as compared to a medium-Vt switch, which induces area overhead and a power overhead from charging/discharging the larger switch. Second we choose a very small power gating switch which is 0.66µm, 0.01% of total effective NFET width, since minimizing standby power is a more important goal for the Phoenix Processor than maintaining performance. The active energy penalty resulting from the voltage drop (maximum of ~100mV) over the narrow power gating switch is easily offset by a large saving in standby power, justifying the tradeoff with the power gating switch.

## 3.2 7.1fW Bitcell and Memory Architecture

While the CPU and other logic modules can be power gated, SRAM modules such as IMEM and DMEM cannot be gated due to their data retention requirement. Consequently these modules dominate standby power consumption. Therefore we place emphasis on low leakage SRAM design. Both IMEM and DMEM use the bitcell shown in Fig 3. The cross-coupled inverters and access transistors use high-Vt devices, while stack forcing and gate length biasing are used to further reduce leakage and improve subthreshold swing. Measurements show that a single bitcell consumes 7.1fW without peripherals (10.9fW with peripherals in DMEM) while retaining data. The bitcell area is 40um$^2$. Although active energy is penalized by the much larger bitcell (9X larger

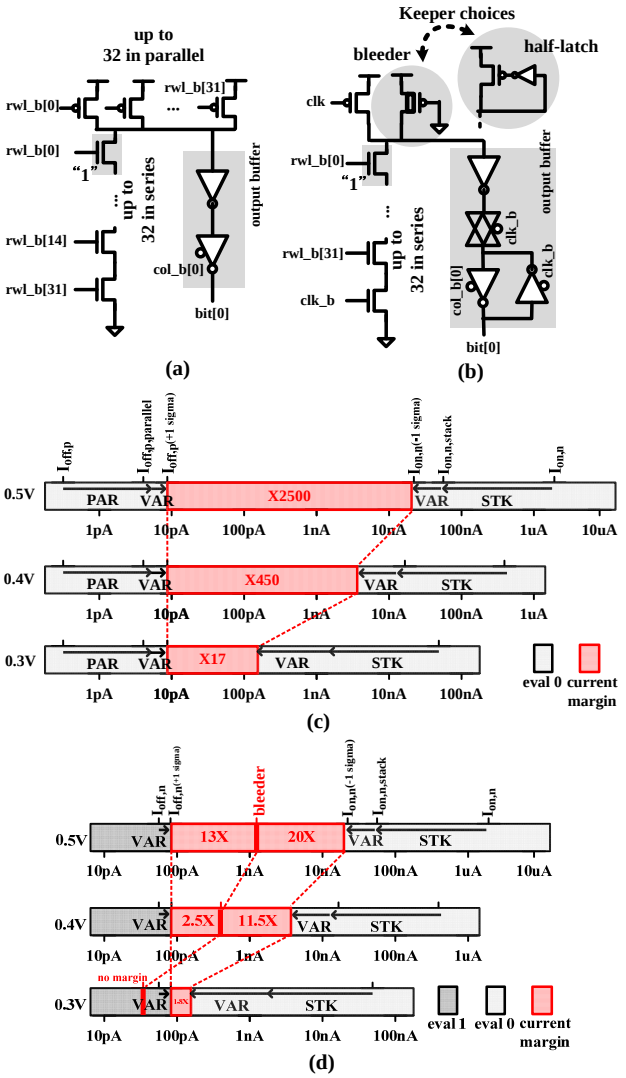



**Figure 4. (a)Static and (b) Dynamic NAND ROM, (c) Current Margin Meter for (c) 32 Stack Static NAND ROM and (d) 32 Stack Dynamic NAND ROM**

compared to the typical bitcell in this technology), it is beneficial from a total energy perspective since the active energy that IMEM and DMEM take is small (~8%) while the standby energy is large (~90%). Furthermore, many sensing applications tend to be standby power dominated due to the low duty cycle.

To enable robust low $V_{dd}$ operation, the proposed cell includes a medium-Vt read buffer similar to [5]. The MVT read buffer also enables single-cycle read-out despite the aggressive use of high-Vt devices elsewhere. This is useful for the IMEM, where a read occurs every cycle. Since the write operation in DMEM is slow relative to the CPU (implemented using medium-Vt devices), write operations are asynchronously performed. Write completion is determined by reading the contents of the row being written and comparing to the write data. Read is single-ended, so a replica delays the write completion signal to guarantee that both sides of the cell have been written correctly, as shown in Figure 3.

To further reduce standby power, the DMEM uses a leakage reduction scheme based on the free-list. The free-list, managed by

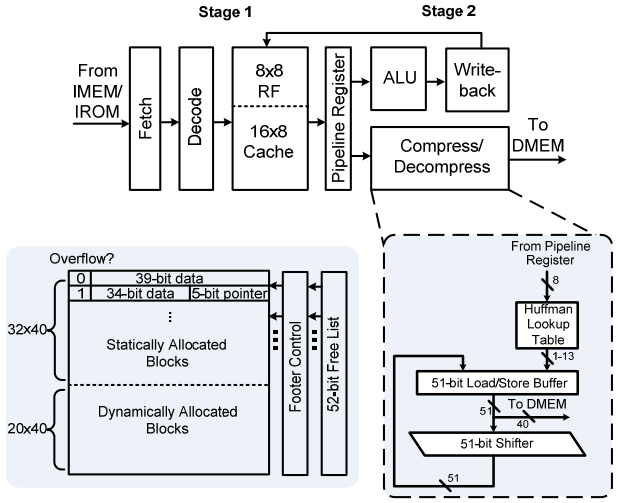



**Figure 5. CPU with Hardware Compression Support**

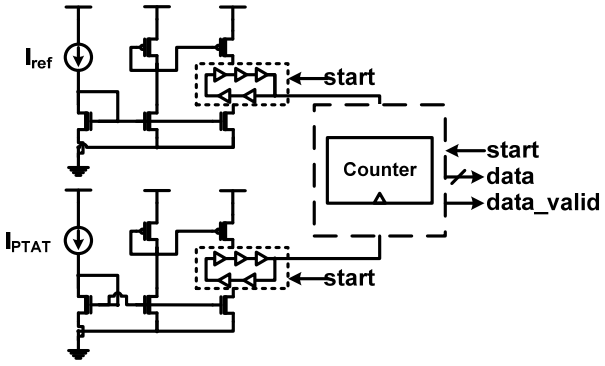



**Figure 6. Schematics of Low Power Temperature Sensor**

CPU contains information about whether a particular row in DMEM is used or not. The DMEM has 26 footers (power gating switches), with each connected to 2 rows (Figure 3). The choice of 2 rows per footer offers a good trade-off between high granularity in power gating and footer overhead. These footers are selectively turned off during standby mode based on the contents of the free-list, reducing DMEM standby power. Additionally, decoders, read/write buffers, and other peripherals are power gated during standby time.

## 3.3 Low Voltage ROM for Robust Operation

Although DMEM must be both read and written, IMEM can be re-optimized to take advantage of its read-only nature. For example, by storing common procedures in ROM with a power gating switch, both standby power and area can be reduced.

There are four key challenges for designing robust ROM at low voltages: 1) The reduced on-current to off-current ratio from large number of stacked or parallel-connected devices causes robustness problems, 2) there is potentially a large skew in beta ratio (relative strength between NFET and PFET) at low voltage, 3) for dynamic ROM styles, conventional keepers (half-latches) are likely to lose state and 4) significant variability further complicates each of the previous three issues.

We compare static NAND ROM and conventional dynamic NAND ROM, shown in Figure 4(a) and 4(b), respectively, using Monte-Carlo iteration as well as a back-of-the-envelope method, referred to as a current margin plot, which estimates the theoretical functionality of ROM at ultra-low voltages and provides guidelines for design decisions. The plot for the 32 stack static and dynamic NAND ROM is shown in Figure 4(c) and 4(d). As the supply voltage is scaled down to 0.3V, the on-to-off current ratio for the static ROM is still larger than 0 while that for dynamic ROM is diminished to 0, implying more robust operation of the static ROM in the low $V_{dd}$ regime. Test circuit measurements show that the static NAND ROM improves performance by 26X, energy by 3.8X, and minimum functional supply voltage by 100mV over a conventional dynamic NAND ROM.[6]

In the Phoenix Processor, 16 stack NAND-style ROM, instead of 32 stack ROM, is implemented. System standby power is reduced by 43% and area reduced by 10.7% by replacing 128 out of 192 SRAM words with power-gated ROM.

## 3.4 ISA Optimization and Data Compression

While the standby power consumption of the CPU itself is minimized by the power gating switch as shown in the Section 3.1, the CPU still needs to be optimized for reducing IMEM and DMEM footprint. IMEM is minimized by optimizing ISA. A minimum group of basic instructions including support for compression interrupt and standby functionalities are used. Additionally, to limit instruction width to only 10 bits, common instructions use flexible addressing modes while less common instructions use implicit operands.

Hardware support for compression is included to minimize the DMEM footprint in standby mode and to maximize memory capacity, as shown in Figure 5. A virtual data memory space of 512B is mapped to the 256B DMEM using Huffman encoding with a fixed dictionary. Maximum compression ratio is 50%. DMEM is divided into statically and dynamically allocated partitions. Each group of 16B (a block) in virtual memory is given one line in the statically allocated partition. If a write to memory causes a block to overflow its statically allocated entry, overflow data is written to an entry in dynamically allocated memory whose location is noted by a pointer in the statically allocated entry. A 52b free-list, which is visible to the CPU, monitors the usage of entries in both memory partitions.

## 3.5 Low Power Temperature Sensor

A low power temperature sensor is designed as shown in Figure 6. A temperature insensitive current source ($I_{ref}$) and proportional to absolute temperature (PTAT) current source ($I_{ptat}$) are generated separately. Each current source is mirrored and fed into the current-starved ring oscillator to translate the temperature information into frequency. Afterwards, the clock signals are fed into an up-counter to generate digitalized output. Since the temperature sensor has no data retention requirement, it is power gated when not being used to eliminate standby power. [7]

# 4. Design Methodologies

In this section, design methodologies are described in three phases. First we will discuss on system-level optimization which provides energy optimal technology, supply voltage and power gating switch size with given application requirements. Second, the circuit design and implementation will be described. Finally testing setup is addressed.

## 4.1 Matlab-Based System-Level Optimization

The system-level optimization is performed to minimize total energy consumption. We identify 6 high-level factors that affect the total energy of the system: technology, the size of power

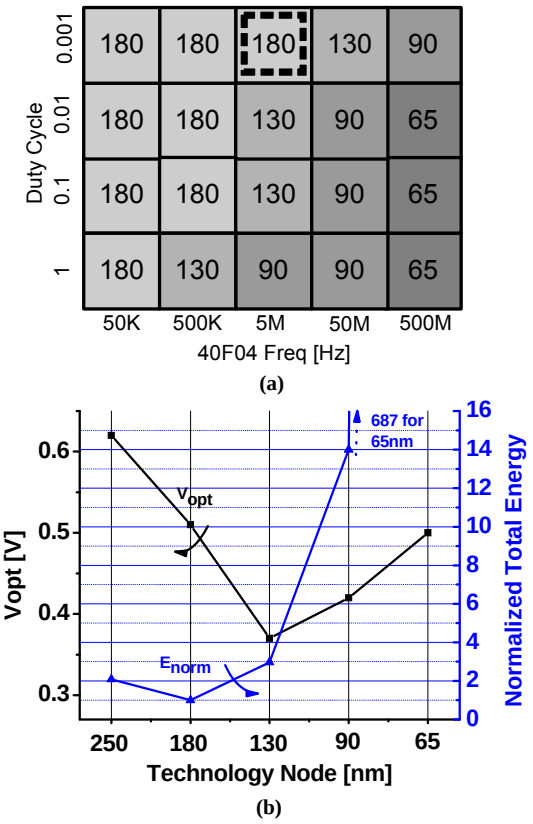



**Figure 7. (a) Energy Optimal Technology Matrix (b) Optimal Vdd and Energy over Technologies**

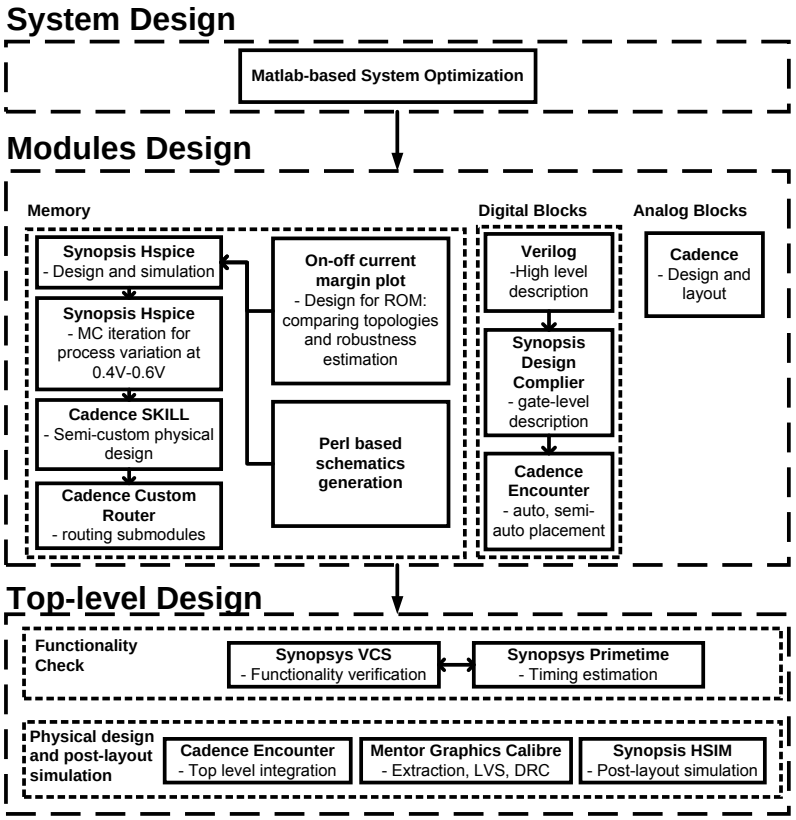



**Figure 8. Design Flow**

gating switch, supply voltage, duty cycle (defined as active time to total time ratio), and the ratio of memory to logic area. The memory to logic area ratio and duty cycle are determined primarily by application requirements while technology, the size of power gating switch, and supply voltage are chosen to minimize total energy consumption.

In order to find the ratio of memory to logic area, we need to estimate DMEM and IMEM size for applications. We examine periodic sensor data logging as a typical operation. We choose 512 words of DMEM since it will take approximately one year to fill the 512 words if one word of data is stored on daily basis. The IMEM size is set relatively smaller at 64B since it is supplemented with 128B IROM. Adding more IROM can be done with negligible increase in standby power. However the size of DMEM and IMEM needs to be changed for different applications.

Based on the DMEM and IMEM size, we estimate the area ratio of memory to logic as 1, which is backed by the actual die photo in Figure 10(c). Duty cycle is assumed to be 0.001.

With the chosen area ratio of memory to logic and the duty cycle, we optimize the total energy consumption by sweeping technology, the size of power gating switch, and supply voltage based on the optimization framework in [8]. Five industrial CMOS technologies from 65nm to 0.25µm are considered. Supply voltage and the size of the power gating switch are selected within the range allowed by each technology. The Phoenix Processor is abstracted as a large collection of inverter chains with different activity ratios. The activity ratio for SRAM modules (DMEM and IMEM) is assumed to be 0.02 while that for CMOS logic (CPU, IROM) is set 10X larger at 0.2. We include high-Vt devices in the optimization process since they can be used in the SRAM bitcell for leakage reduction. High-Vt IO devices are also considered as a viable option for building bitcells.

The result of the optimization is shown in Figure 7. The matrix shows the energy optimal technology at the given performance and duty cycle requirement of each application. With the target duty cycle of 0.001, which is typical for sensor applications, we have three different optimal technologies for different performance requirements. Since the Phoenix Processor has a relaxed performance requirement and higher performance causes more energy consumption, 0.18µm is selected as the energy optimal technology.

For the highlighted block whose duty cycle is 0.001 and performance is 5MHz (40F04 frequency), the optimal supply voltage and energy is plotted in the Figure 7. Old technologies are favored since they have lower leakage energy. The reason why 0.18µm technology gives lower energy consumption than 0.25µm is that the particular 0.18µm that we investigated has a higher-Vt IO device than the 0.25µm technology. If the 0.25µm technology offered a comparable high-Vt device, it could be the optimum choice unless the area overhead associated is intolerable.

The optimum supply voltage estimated is 0.5V. The size of the power gating switch is determined proportionally to the relative size of actual modules to the inverter chains used in the optimization process.

## 4.2 Circuit Simulation and Implementation

As shown in Figure 8, the entire design is performed in three phases: system design, module design, and top-level design. The system design provides energy-optimum technology, supply voltage and the size of the power gating switch given the area ratio of memory to logic, and duty cycle for application requirements. The module design is divided up into memory, digital, and analog design. Memory design including DMEM, IMEM, and IROM, is performed using Synopsys HSPICE for simulations and Monte-Carlo iterations. Due to its regular structure, Cadence SKILL script and Custom Router are used for semi-custom physical design in a Cadence environment. For IROM design the current margin plot is used for comparing different ROM topologies and estimating robustness at low $V_{dd}$.

Digital design including CPU and PMU generally follows three

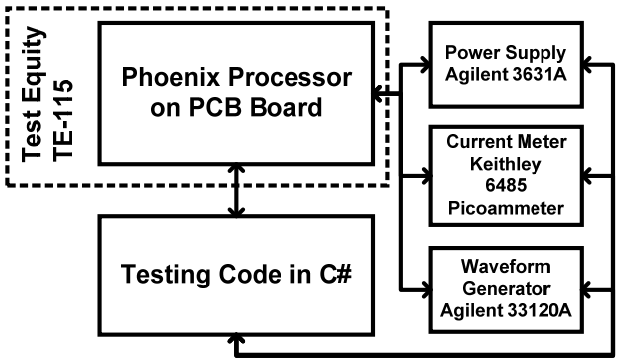



**Figure 9. Testing Setup**

steps: RTL-description in Verilog, synthesis by Synopsys Design Compiler, and APR (Automatic Placement and Routing) by Cadence Encounter. For synthesis and APR, the Artisan Standard Cell Library for 0.18µm technology is used. In the case of the PMU, the gate-level description is written manually and is implemented using a custom high-Vt library.

Analog design including the watchdog timer and the temperature sensor is carried out in the Cadence environment with Spectre simulation and Virtuoso. Monte-Carlo iterations are performed to guarantee tolerance to process variation.

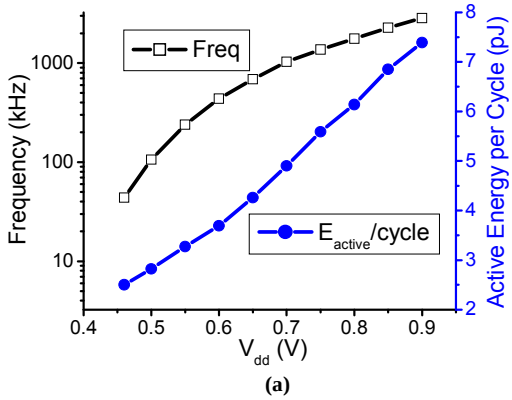
After the module design, top-level design is performed. Synopsys VCS and Primetime are used for functional and timing verification. For physical design and post-layout simulation, Cadence Encounter, Mentor Graphics Calibre, and Synopsys HSIM are used. Since the wire resistance is negligible compared to the on-resistance of the FET in low voltage, minimum width lower metal layers are used for clock and power distribution for minimizing capacitance.

For testing purposes, we have several knobs in the design. A flip-flop-based scan chain is used. Body pins are separated from source pins to enable future post-silicon body biasing tuning. In addition, since very small currents need to be measured, explicit ESD diodes are removed from power pads. However we ensure that these pads drive a sufficient source/drain area for protection against ESD. Finally we have a configurable power gating switch whose width can be modulated from 0.66µm to 28µm using jumpers in the testing printed circuit board.

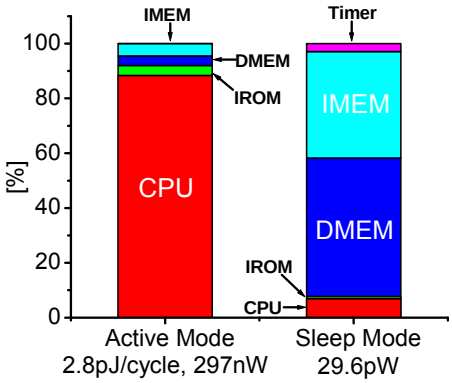
## 4.3 Testing Setup

The testing setup is shown in Figure 9. For environmental temperature control, a Test Equity TE-115 temperature chamber is used. A 4 layer PCB board is designed and connected to Agilent 3631A power supplies, a Keithley 6485 Picoammeter, and an Agilent 33120A waveform generator. Although we have clock generation circuitry inside the Phoenix Processor, the pulse generator is sometimes used to facilitate the testing process. Testing code is written in Microsoft Visual Studio C# and run on Microsoft Windows XP.

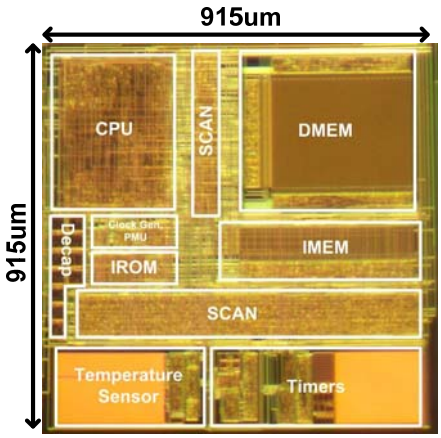
Several guidelines from Keithley [9] help to measure ultra-low current. First the plus polarity of the ammeter is recommended to connect with higher impedance nodes for low current measurement. The Phoenix Processor in standby mode can be treated as a ~TΩ impedance block due to the ultra-low current draw. Therefore we reverse the direction of the ammeter connection. Second all the wires among testing modules are




(a)




(b)




(c)

**Figure 10. (a) Shmoo Plot in Active Mode, (b) Energy Breakdown in Active and Standby Mode, (c) Die Photo**

prevented from moving or bending to avoid static charge buildup and unstable readings in the current meter.

## 5. Measurement Results

A prototype circuit for the Phoenix Processor, including the temperature sensors and timers, was fabricated in a commercial 0.18µm process. The total area is 915x915µm$^2$ as shown in Figure 10(c). In Figure 10(a), active energy and working frequency over supply voltage are plotted, showing 2.8pJ/cycle and 106kHz at $V_{dd}$=0.5V. Among the components in the Phoenix Processor, CPU dominates the total active energy.

The Phoenix Processor consumes 29.6pW in standby mode with half the DMEM being filled as shown in Figure 10(b). The standby power for temperature sensor is simulated as 3.5pW. The 0.18μm technology chosen from the system-level optimization helps to reduce the standby power by 647X compared to the 65nm technology, as shown in Figure 7(b).

Due to the extremely small power gating switch optimized for standby power consumption, the CPU and IROM consume only ~2pW. If conventional power gating switch sizing, which is ~10% of total effective NFET size [10], CPU standby power would increase by 1000X and dominate the total power consumption, confirming the importance of the new approach of sizing the power gating switch for minimizing standby power.

The reduction of standby power in the CPU causes the DMEM and IMEM to dominate the total standby power. On top of optimizing the bitcell, dynamic power gating in DMEM saves standby energy by turning off unused rows. Measurement shows it can modulate the standby power of the DMEM from 7.5pW to 22.5pW, depending on the number of rows in the DMEM being used. In addition, the hybrid use of ROM and SRAM for instruction memory gives 43% saving in standby power and 10.7% reduction in area compared to use SRAM-only instruction memory. The hardware compression support is expected to give a 50% compression ratio, playing as another important contributor to reduce the standby power in the DMEM.

| | Tech | E/cycle | $P_{standby}$ |
|---|---|---|---|
| Zhai, et al.[11] | 0.18um | 2.6pJ | 238nW |
| Hanson, et al.[12] | 0.13um | 3.5pJ | 154nW |
| This work | 0.18um | 2.8pJ | 34pW |

**Table 1. Comparisons of Recently Published Sub-Vt Designs**

We compared the Phoenix Processor with two published Sub-Vt microprocessor designs in Table 1, showing that the Phoenix Processor reduces the standby power by more than 7000X and 4000X over [11] and [12], respectively. This power reduction can benefit either reducing the size of battery at iso-lifetime or increasing the lifetime of system with same power source.

In a typical sensing application that requires 2000 instructions to be run once every 10 minutes, the average power is only 39pW. As long as the Phoenix Processor spends more time in standby mode, the total power approaches to average 29.6pW. After integrating with a $1mm^2$ sized micro battery, the entire system whose volume is only $1mm^3$ can last more than year, greatly benefitting biomedical and environmental sensor systems.

## 6. Conclusions

In this work, we describe a sensor processor that operates at $V_{dd}$=0.5V to minimize active mode energy and uses device-, circuit-, and architecture-level techniques to minimize standby mode energy. Measurements show that Phoenix Processor consumes 297nW in active mode and only 29.6pW in standby mode. A thin film or micro-sized battery with the same form factor as Phoenix Processor could provide a lifetime on the order of years, making Phoenix Processor an attractive candidate for future sensor type computing systems.


## Acknowledgement

The authors appreciate Dr. Bo Zhai, Dr. Carlos Tokunaga, Jae-sun Seo, and Gregory Chen for the valuable discussions during the project. We also appreciate the help of MOSIS staff during chip fabrication.


## 7. REFERENCES


[1] U. Schnakenberg, P. Walter, G. vom Bogel, C. Kruger, H.C. Ludtke-Handjery, H.A. Richter, W. Specht, P. Ruokonen, W. Mokwa, "Initial investigations on systems for measuring intraocular pressure," Sensors and Actuators 85, No. 1-3, pp. 287-291, 2000.

[2] Y.-S. Lin, S. Hanson, F. Albano, C. Tokunaga, R. Haque, K. Wise, A. Sastry, D. Blaauw, D. Sylvester, "Low-Voltage Circuit Design for Widespread Sensing Applications," International Symposium on Circuits and Systems, pp. 2558-2561, 2008.

[3] M. Seok, S. Hanson, Y.-S. Lin, Z. Foo, Y. Lee, N. Liu, D. Sylvester, D. Blaauw, "The Phoenix Processor: A 30pW Platform for Sensor Applications," Symposium on VLSI circuits, pp. 188-189, 2008.

[4] A. Wang, A. Chandrakasan, "A 180mV FFT Processor Using Subthreshold Circuit Techniques," International Solid-State Circuits Conference, pp. 292-293, 2004.

[5] L. Chang, D. Fried, J. Hergenrother, J. Sleight, R. Dennard, R. Montoye, L. Sekaric, S. McNab, A. Topol, C. Adams, K. Guarini, W. Haensch, "Stable SRAM Cell Design for the 32 nm Node and Beyond", Symposium on VLSI Technology, pp.128-129, 2005

[6] M. Seok, S. Hanson, J. Seo, D. Sylvester, D. Blaauw, "Ultra-Low Voltage ROM Design," Custom Integrated Circuits Conference, pp.423-426, 2008.

[7] Y.-S. Lin, D. Sylvester, D. Blaauw, "An Ultra Low Power 1V, 220nW Temperature Sensor for Passive Wireless Applications", Custom Integrated Circuit Conference, pp.507-510, 2008

[8] M. Seok, D. Sylvester, D. Blaauw, "Optimal Technology Selection for Minimizing Energy and Variability in Low Voltage Applications", International Symposium on Low Power Electronics and Design, pp.9-14, 2008

[9] Keithley, "Low Current Measurements", www.keithley.nl

[10] J. Kao, S. Narendra, A. Chandrakasan, "MTCMOS hierarchical sizing based on mutual exclusive discharge patterns", Design Automation Conference, pp.495-500, 1998

[11] B. Zhai, L. Nazhandali, J. Olson, A. Reeves, M. Minuth, R. Helfand, S. Pant, D. Blaauw, T. Austin, "A 2.60pJ/Inst Subthreshold Sensor Processor for Optimal Energy Efficiency", Symposium on VLSI circuits, 2006

[12] S. Hanson, B. Zhai, M. Seok, B. Cline, K. Zhou, M. Singhal, M. Minuth, J. Olson, Leyla Nazhandali, T. Austin, D. Sylvester, D. Blaauw, "Performance and variation optimization strategies in a sub-200mV, 3.pJ/inst, 11nW subthreshold processor", Symposium on VLSI circuits, pp-152-153, 2007

[13] M.-E.Hwang, A. Raychowdhury, K. Kim, K. Roy, "A 80mV 40nW Process-Tolerant Subthreshold 8x8 FIR Filter in 130nm Technology, Symposium on VLSI circuits, pp.154-155, 2007