A Case for Opportunistic Embedded Sensing In Presence of Hardware Power Variability

Lucas Wanner, Charwak Apte, Rahul Balani, Puneet Gupta, and Mani Srivastava
{wanner@cs, {charwak, rahulb, puneet, mbs}@ee}.ucla.edu
University of California, Los Angeles

Abstract

The system lifetime gains provided by the various power management techniques in embedded sensing systems are a strong function of the active and sleep mode power consumption of the underlying hardware platform. However, power consumption characteristics of hardware platforms exhibit high variability across different instances of the platform, diverse ambient conditions, and over passage of time. The factors underlying this variability include increased manufacturing variations and aging effects due to shrinking transistor geometries, and deployment of embedded devices in extreme environments. Our experimental measurements show that large variability in sleep mode power is already present in commonly used embedded processors, and technology trends suggest that the variability will grow even more over time and affect active mode power as well. Such variability results in suboptimal lifetime and service quality. We therefore argue for energy management approaches that learn and model the power characteristics of the specific instance of the hardware platform, and adapt accordingly.

1 Introduction

Energy management methods in embedded systems rely on knowledge of power consumption of the underlying computing platform in various modes of operation. These power specifications are usually derived from the datasheets. Unfortunately, the microelectronic substrate is increasingly plagued by variability, especially in power consumption, both across multiple instances of a system and in time over its usage life. As a result the "datasheet power specifications" are heavily guardbanded (e.g., see [12]) leaving much of the energy potential or sensing quality untapped. The variability has few major sources:

- *Semiconductor manufacturing*. Scaling of physical dimensions faster than the optical wavelengths or equipment tolerances used in the semiconductor manufacturing line has led to increased process variability [2, 5] which makes integrated circuit designs unpredictable. Figure 1 [10] shows that the manufacturing variability in sleep (or static) power and total power is likely to grow over 500% and 100% respectively in the next decade[1].
- *Environment*. Ambient condition variability (e.g., voltage and temperature).
- *Battery*. Total energy capacity can vary for nominally identical batteries.
- *Aging*. Transistor aging (e.g., due to negative bias temperature instability [18]) can change system power/performance over time.
- *Vendor*. Multi-sourcing of parts with identical specification from different vendors is common and can cause significant variation.

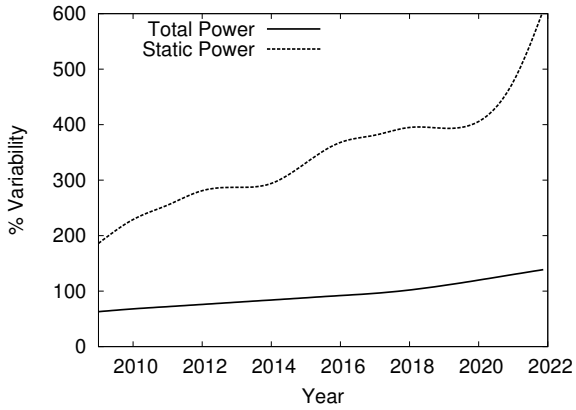

Figure 1: ITRS projections of power variability

Variability, thus far, has been largely addressed by process, device and circuit designers with software designers remaining isolated from it by a rigid hardware-software interface. Recently there have been some efforts at higher layers of abstraction. For instance, software fault tolerance schemes have been used to address voltage [17] or temperature variability [6]. Hardware "signatures" are used to guide adaptation in quality-sensitive multimedia applications in [14]. In the embedded sensing context, [13, 8] propose sensor node deployment methodology based on the variability in leakage power across different nodes.

[1]The fact that sleep power variability is large and growing stems from exponential dependence of leakage on most physical and environmental parameters.

Wireless embedded sensing systems employ a variety of power management techniques to achieve system lifetime objectives [16]. A particularly common technique is duty cycling [7] where the system is by default in a sleep state but woken up periodically to attend to pending tasks and events. Judicious choice of duty cycling parameters depends also on the active and sleep mode power consumption of the platform. Often the duty cycle ratio (fraction of time the system is active) is extremely small ($<< 1\%$), so that the energy consumed by the platform during the sleep state accounts for almost all ($> 99\%$) of the energy consumption. In such cases, the achievable duty cycle ratio is primarily determined by the sleep state power consumption of the platform. Limitations on power management posed by sleep power, which is due to the leakage in circuits, has also been recognized and analyzed [11]. However, the impact of hardware variability on duty cycling or other power management techniques has not been studied. Perhaps most closely related work is by researchers who have studied adaptive duty cycling, although their focus has been on variations in workload [1] or energy availability (as would be the case in systems that rely on scavenging energy from the environment).

We discuss measured power variability for off-the-shelf embedded processors, and potential software approaches to handle variability in power management of embedded sensing systems. In particular, we focus on instance and temperature dependent sleep power variability that already manifests in contemporary embedded processors, and its impact on duty cycling.

2 Power Consumption Variability in Modern Embedded Processors

We focus our discussion on the sleep mode power P_{sleep}, which, as noted earlier, is a primary platform characteristic of interest in the low duty cycle ratio regime. Sleep or static power, occurs due to transistor leakage. During the active mode, additional switching power is consumed.

With shrinking geometries the ratio of sleep mode power to active mode power has been increasing (as high as 40% in chips fabricated using 65nm technology). This is due to the inability to turn the devices "off" effectively as device dimensions continue to shrink. Manufacturing spread in transistor parameters can cause up to 20x variation in sleep mode power [3] in addition to substantial variation with supply voltage, and temperature. Specifically in context of wireless sensor platforms, which often use unregulated battery output as supply, and could be deployed in extreme ambient conditions, the variation in leakage power during its lifetime, due to temperature and voltage variation may be substantial.

To understand sleep power variability in embedded processors, we characterized the sleep power for Atmel's SAM3U microcontroller, which is based on ARM's Cortex M3 processor core. While we have been unable to determine from available literature the precise technology node the chip is fabricated in, indirect evidence as well as the vintage suggests that it is most likely fabricated in a 130 nm process. Cortex M3 is a good representative of the current generation of low-end 32-bit embedded processors, and is incorporated in many emerging embedded platforms. Our characterization effort had two components: analytic modeling of sleep power as a function of temperature, and experimental measurement of sleep power as function of temperature across an ensemble of several identical SAM3U-EK boards incorporating SAM3U in a LQFP144 package.

2.1 Analytical Modeling of Sleep Power

Static power has four main sources: (i) Sub-threshold Leakage current that flows between source and drain of a MOSFET for gate-to-source voltages below the threshold, (ii) Gate Leakage current due to tunneling of carriers through the gate oxide to the substrate, (iii) Reverse Biased Junction Leakage current which flows from the source/drain regions to the substrate through the reverse biased p-n junctions due to band-to-band tunneling and diffusion, and (iv) Gate Induced Drain Leakage current due to band to band tunneling in the region of overlap between the gate and drain. At temperatures below 150°C, only the first two components are large enough, and of the two only sub-threshold leakage exhibits strong variability with temperature. Therefore, the sleep power can be modeled as the following function of temperature (derived from BSIM4 compact device model [4]):

$$P_{sleep} = V_{dd}(AT^2 e^{B/T} + I_{gl})$$

where A and B are technology-dependent constants, I_{gl} is the temperature-independent gate leakage current, and T is the core temperature. We combine the sleep power model with a model of the thermal dynamics of a packaged chip [9]:

$$RC\frac{dT(t)}{dt} + T(t) - RP(t) = T_{amb}$$

where $T(t)$ and $P(t)$ are the core temperature and power consumption of the chip at time t, R and C are the thermal resistance and capacitance of the chip package, and T_{amb} is the ambient temperature. At steady state $\frac{dT(t)}{dt} = 0$, so that $T_{steady-state} = T_{amb} + RP(t)$.

For the SAM3U in a LQFP144 package, the typical values of R and C are 50°C/W and 4–5 J/°C respectively. The nominal static power of SAM3U is 30 μW. Nominal

active power when operating at 48 MHz while performing a Dhrystone benchmark is 80 mW. This indicates that when sleep mode power measurements are performed, the self-heating of the chip due to static power consumption is negligible, and even in active mode, the temperature difference is $\sim$6.5°C.

2.2 Experimental Measurements

Based on the preceding analysis, for sleep mode measurements it is reasonable to assume that the static power follows similar dependence on ambient temperature as given by the previously derived expression for P_{sleep}. We verify this assumption from our measurements and characterize each instance of microcontroller based on the above model. We wait four RC time constants to make the measurements of leakage to ensure steady state.

Figure 2 shows that the variation in sleep power across ten instances of SAM3U at room temperature is more than $5x$. We also measured active mode power, but as expected for this processor, which is in an older technology, the variation is minimal ($< 10\%$).

Figure 3 shows the variation in sleep power across ten instances of SAM3U over a temperature range of 22–60°C, which is representative of the temperatures that embedded sensors deployed under unregulated and extreme ambient conditions often face (e.g. in factories, desert etc.). As expected, individual processor instances

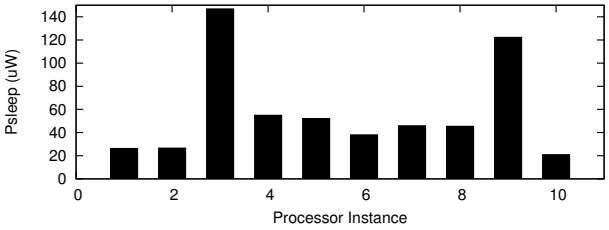

Figure 2: P_{sleep} at room temperature

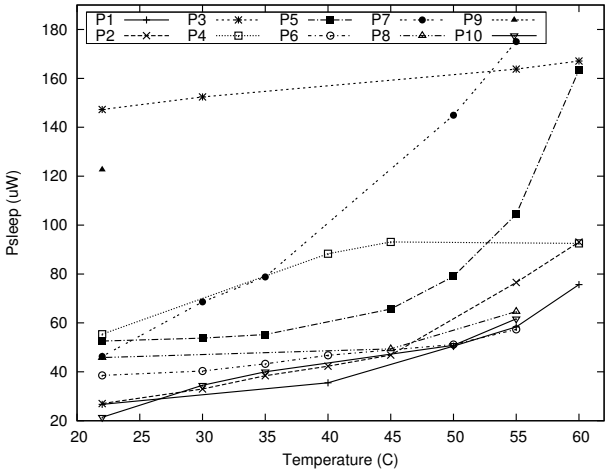

Figure 3: Variation in P_{sleep} with temperature

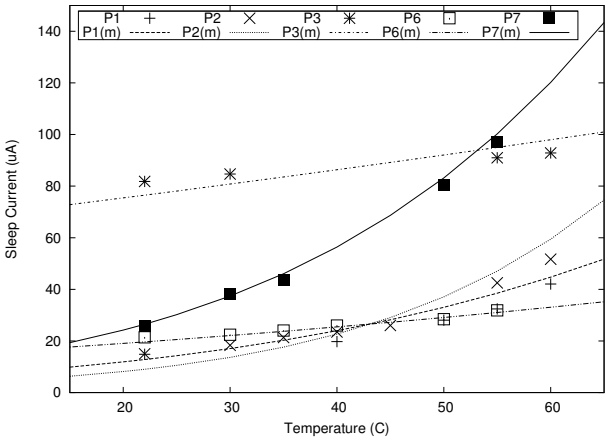

Figure 4: Comparing measured vs. modeled variability of sleep power with temperature. Only five of the ten boards are shown for clarity.

exhibit large sleep power variations over the temperature range. While change in sleep power for any individual processor is monotonic, the magnitudes of variations are different so that relative rankings of different processors change over temperature.

We also fitted the experimental measurements to the analytic model discussed earlier, using minimum mean square error criterion. Figure 4 shows that the simple model discussed earlier works well. We use these fitted models for the discussion in next section.

3 Towards an Opportunistic Software Stack for Sensing

As the discussion in the preceding section shows, significant variability in sleep power is already present in contemporary embedded processors. Duty cycling is particularly sensitive to variations in sleep power at low duty cycling ratios. Variability implies that any predeployment choice of duty cycle ratio that is selected to ensure desired lifetime needs to be overly conservative, and result in lower quality of sensing or lifetime.

Consider an application running on ATSAM3U that periodically wakes up, samples a sensor, sends the result to a forwarding module (e.g. for storing, sending to the network), and goes into sleep mode. We assume that the sampling task will complete within 10 ms with a 48 MHz clock in active mode. For the sake of clarity, we also assume that the system and the forwarding module take negligible time and power to complete their operations. The performance requirement is a desired lifetime of 20000 hours using two AA batteries (5400 mAh) operating at 1.8V. As shown in Figure 5, across the five hardware samples and over temperature variation, the worst case duty-cycle to achieve the desired lifetime is approximately 0.43%, which results in a sleep du-

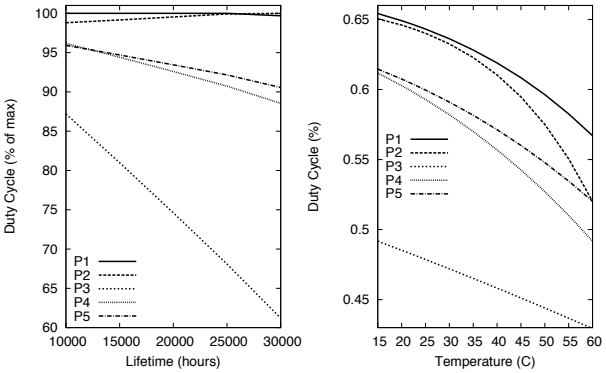

Figure 5: Implication of P_{sleep} variability on duty cycling

ration of 2300 ms for every active period of 10 ms. The best case duty-cycle is approximately 0.65%, which results in a sleep duration of 1500 ms and approximately 51% more sensor data being collected. It should be noted that these results take only instance and temperature dependent variation into account. In long running systems, variation due to aging would also need to be modeled and/or learned to find allowable duty cycling rates.

In order to maximize the sensing quality in the presence of such variation, an opportunistic sensing software stack can help discover and adapt the application duty cycle ratio to the sleep mode power variations across parts and over time. The run-time system for the opportunistic stack will have to keep track of changes in hardware characteristics, and provide this information through interfaces accessible to either the system or the applications. Figure 6 shows several different ways such an opportunistic stack may be organized; the scenarios shown differ in how the sense-and-adapt functionality is split between applications and the operating system. Scenario

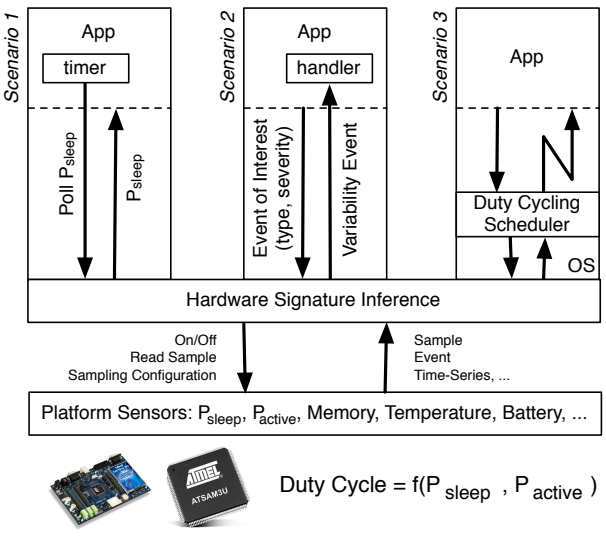

Figure 6: Designing a software stack for variability-aware duty cycling

1 relies on application polling the hardware for its current "signature" while in the second scenario, application handles variability events generated by the operating system. In the last scenario, handling of variability is largely offloaded to the operating system.

Using architecture similar to that of scenario 3 in Figure 6, we have implemented a prototype variability-aware duty cycling framework in TinyOS. Application modules specify to the scheduler a range of acceptable duty cycling ratios, and the scheduler selects the actual duty cycling ratio based on run-time temperature measurements and a stored temperature vs. sleep power map that is learnt off-line for the specific processor instance. While this approach is potentially less flexible than the ones presented by scenarios 1 and 2, it simplifies application development by abstracting the underlying complexities of the variability signature model. Measurements with ATSAM3U suggest an average of 1.8x improvement in duty cycle ratio (and thus in quality of sensing) for a given lifetime target. Furthermore, projections presented in Figure 7, using sleep-to-active mode power in future technologies available from IBM [15], indicate as much as two orders of magnitude gains with a variability-aware duty cycling approach as embedded processors move to advanced technologies.

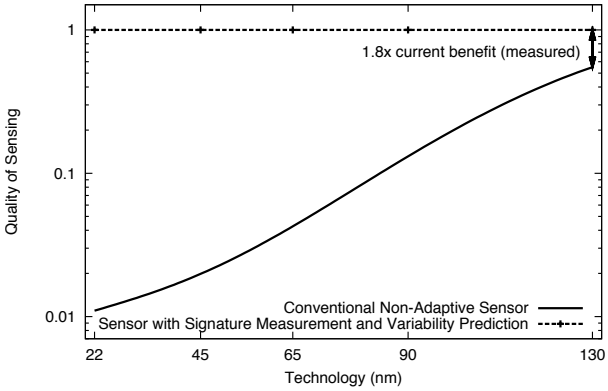

Figure 7: Current and projected gains from variability-aware duty-cycling management

As embedded components in more advanced technologies (65 nm and smaller) become commonplace and operating voltages get reduced to near-threshold and sub-threshold regimes for energy efficiency, the variations will increase significantly [2] and also manifest in active mode power and achievable speeds. The elasticity inherent in sensor information processing algorithms may be used to gracefully move in the fidelity-lifetime space as platform speed, sleep power, and active power characteristics vary across different instances and over time. For instance, sensing applications such as video surveillance lend themselves to adaptive media processing. From our earlier work in [14], Figure 8 shows that quality-power tradeoff (with dynamic voltage scaling) can differ sig-

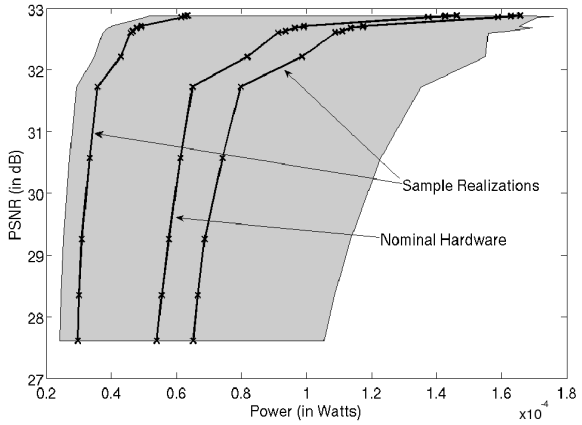

Figure 8: PSNR vs. active power for 1000 simulated samples of H.264 encoder in 45nm technology.

nificantly for different hardware instances of a video encoder. This can be leveraged opportunistically for instance dependent encoding configuration choice. Same work shows that such instance or hardware signature based adaptation can improve average PSNR by 3dB in presence of performance variability.

4 Conclusions

In this paper, we have argued for an adaptable software stack for embedded sensing that will opportunistically adjust to variability while making use of the elasticity inherent in sensor information processing, instead of designing the software for a rigid and conservative platform specification. Such software will make use of measurements from variability monitors embedded in the platform coupled with offline testing, characterization, software-based inference and statistical modeling to learn and predict the actual power-performance characteristics of the hardware.

While we motivated the problem of power variability and outlined a software-based solution approach in the context of sleep power and duty cycling, both the problem and the solution approach have broader implications. Static power will become more comparable in magnitude to active power [15], which would require variability to be handled not just in ultra-low duty cycle regime, but also during regular operation. Moreover, using appropriate hardware monitoring mechanisms, battery as well as aging induced variability can also be exposed to the software layers.

Acknowledgements

This material is based upon work supported by the NSF under award # CNS-0905580 and CCF-1029030. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF. Lucas Wanner is supported in part by CAPES/Fulbright grant #1892/07-0.

References

[1] G. Anastasi, M. Conti, and M. Di Francesco. Extending the lifetime of wireless sensor networks through adaptive sleep. *IEEE Trans. on Industrial Informatics*, 5(3):351 –365, 2009.

[2] K. Bernstein, D. J. Frank, A. E. Gattiker, W. Haensch, B. L. Ji, S. R. Nassif, E. J. Nowak, D. J. Pearson, and N. J. Rohrer. High-performance CMOS variability in the 65-nm regime and beyond. *IBM J. Res. Dev.*, 50(4):433–449, 2006.

[3] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De. Parameter variations and impact on circuits and microarchitecture. In *Proc. Design Automation Conf. (DAC)*, pages 338–342, 2003.

[4] BSIM. http://www-device.eecs.berkeley.edu/∼bsim3/.

[5] Y. Cao, P. Gupta, A.B. Kahng, D. Sylvester, and J. Yang. Design sensitivities to variability: extrapolations and assessments in nanometer VLSI. In *Proc. IEEE Intl. ASIC/SOC Conf.*, pages 411–415, 2002.

[6] J. Choi, C.-Y. Cher, H. Franke, H. Hamann, A. Weger, and P. Bose. Thermal-aware task scheduling at the system software level. In *Proc. Intl. Sym. on Low power electronics and design (ISLPED)*, pages 213–218, 2007.

[7] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler. Design of a wireless sensor network platform for detecting rare, random, and ephemeral events. In *Proc. Sym. on Information processing in sensor networks (IPSN)*, 2005.

[8] S. Garg and D. Marculescu. On the impact of manufacturing process variations on the lifetime of sensor networks. In *Proc. Intl. Conf. on Hardware/software codesign and system synthesis (CODES/ISSS)*, pages 203–208, 2007.

[9] H. Huang, G. Quan, and J. Fan. Leakage temperature dependency modeling in system level analysis. In *Proc. Intl. Sym. on Quality Electronic Design (ISQED)*, pages 447–452, 2010.

[10] ITRS. The international technology roadmap for semiconductors. http://public.itrs.net/.

[11] R. Jejurikar, C. Pereira, and R. Gupta. Leakage aware dynamic voltage scaling for real-time embedded systems. In *Proc. Design Automation Conf. (DAC)*, pages 275–280, 2004.

[12] K. Jeong, A.B. Kahng, and K. Samadi. Impact of guardband reduction on design outcomes: A quantitative approach. *IEEE Trans. on Semiconductor Manufacturing*, 22(4):552–565, 2009.

[13] T. Matsuda, T. Takeuchi, H. Yoshino, M. Ichien, S. Mikami, H. Kawaguchi, C. Ohta, and M. Yoshimoto. A power-variation model for sensor node and the impact against life time of wireless sensor networks. In *Intl. Conf. on Communications and Electronics (ICCE)*, pages 106–111, 2006.

[14] A. Pant, P. Gupta, and M. van der Schaar. Software adaptation in quality sensitive applications to deal with hardware variability. In *Proc. Great lakes Sym. on VLSI (GLSVLSI)*, pages 85–90, 2010.

[15] R. Puri, L. Stok, and S. Bhattacharya. Keeping hot chips cool. In *Proc. Design Automation Conf. (DAC)*, pages 285–288, 2005.

[16] V. Raghunathan, S. Ganeriwal, and M. Srivastava. Emerging techniques for long lived wireless sensor networks. *IEEE Communications Magazine*, 44(4):108–114, 2006.

[17] V. J. Reddi, M. S. Gupta, M. D. Smith, G. Wei, D. Brooks, and S. Campanoni. Software-assisted hardware reliability: abstracting circuit-level challenges to the software stack. In *Proc. Design Automation Conf. (DAC)*, pages 788–793, 2009.

[18] R. Zheng, J. Velamala, V. Reddy, V. Balakrishnan, E. Mintarno, S. Mitra, S. Krishnan, and Yu Cao. Circuit aging prediction for low-power operation. In *IEEE Custom Integrated Circuits Conf. (CICC)*, pages 427–430, 2009.