Underdesigned and Opportunistic Computing

Puneet Gupta
Electrical Engineering, UC Los Angeles
puneet@ee.ucla.edu

Rajesh K. Gupta
Computer Science & Engineering, UC San Diego
gupta@cs.ucsd.edu

***Abstract*— Variation in the specifications of microelectronic chips across parts and over time has been a great source of concern for the integrated circuit chip designers because of the ever-increasing guard-bands that the circuit and system designers must rely upon to ensure working parts and systems. This prompts us to look for solutions that can mitigate the effect of performance and power variability through innovations in system software. In this paper, we outline a novel, flexible hardware-software stack and interface that use adaptation in software to relax variation-induced guard-bands in hardware design.** [1]

Keywords: Variability, DFM, Hardware/Software Co-design.

I. INTRODUCTION

Increasing performance and power variations in the manufactured semiconductor parts are affecting cost and reliability of electronic systems (see Figure 1). This variability stems from the semiconductor manufacturing process itself that is increasingly building features at atomic scale, the operating environment, as well as variations across manufacturers and aging of the microelectronic parts. These variations can be permanent (e.g., due to manufacturing) or transient (e.g., due to ambient). They can be parametric in nature (e.g., changes in power consumption) or cause functional failures (e.g., bit flips in cache due to low noise margin at increased temperatures).

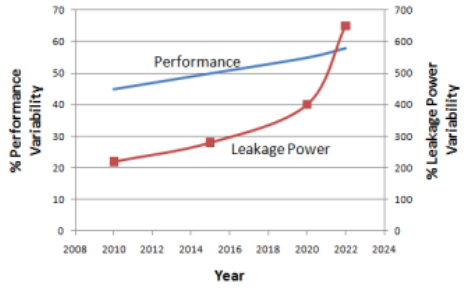

Figure 1. ITRS projections of variability.

While the process variability is increasing, the basic approach to designing and operating electronic systems has remained unchanged. That is, the software assumes the hardware to deliver a certain specified functions and level of performance, and that all manufactured parts are exactly the same as seen by the software. The component designers work hard to meet the specifications, relying primarily on conservative guard-bands resulting in significant overdesign to ensure reasonable hardware manufacturing yields. A recent study puts the cost of overdesign to be 40% larger chips that consume 35% more active power and 60% more sleep power than what a nominal design would require [9]. These costs directly translate into system-level inefficiencies from expensive cell phones, heat-generating servers to costly sensor nodes with big batteries.

In fact, manufacturing variations should be viewed in the same light as other sources of variations, such as operating conditions and variation due to aging of the parts. This prompts us to envision a new paradigm for computer systems, one where nominally designed (and hence *underdesigned*) hardware parts work within a software stack that *opportunistically* adapts to variations (see Figure 2). The resulting Underdesigned and Opportunistic (UNO) computing machines can be classified along following two axes: (a) *Type of Underdesign:* use parametrically under-provisioned circuits (e.g., voltage over-scaling as in [4],[5]) or be implemented with explicitly altered functional description (e.g., [3], [6]); and (b) *Type of Operation:* rely upon application's level of tolerance to limited errors (as in [7], [8]) to ensure continued operations. By contrast, error-free UNO machines correct all errors (e.g., [4]) or rely on hardware to correct-operation limits (e.g., [1], [2]).

Thus UNO machines seek to expose difficult-to-predict spatio-temporal variations in hardware, instead of hiding these behind conservative specifications. The IC design flow will use software adaptability and error resilience for relaxed implementation and manufacturing constraints. Instead of crash-and-recover from errors, the UNO machines make proactive measurements and predict parametric and functional deviations to ensure continued operations and availability. This will preempt impact on software applications, rather than just reacting to failures (as is the case in fault-tolerant computing) or under-delivering along power/performance/reliability axes.

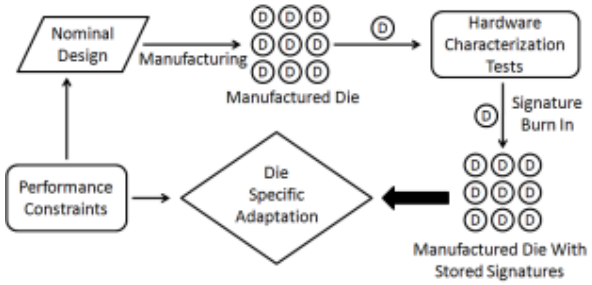

Figure 2. The UNO adaptation model.

II. EXAMPLE UNO COMPUTING MACHINES

We briefly describe two examples of parametrically underdesigned, error-free UNO computing machines.

[1] The work is supported in part by NSF Variability Expedition (http://variability.org)

A. *Variation-Aware Duty-Cycling for Embedded Sensing [2]*

Sensor node design makes a tradeoff between quality of sensing data (through increased duty-cycle of the node for increasing sampling) and the longevity of the on-board battery. Variance in manufacturing and temperature can have a dramatic effect on the quality of sensing and longevity of the sensor node (see Figure 3). Our measurements for the Atmel SAM3U show a variation of 40% in active power across 200 degC, and 14x in leakage power across 40 degC. Ignoring this variation has significant cost and sensor node availability effects.

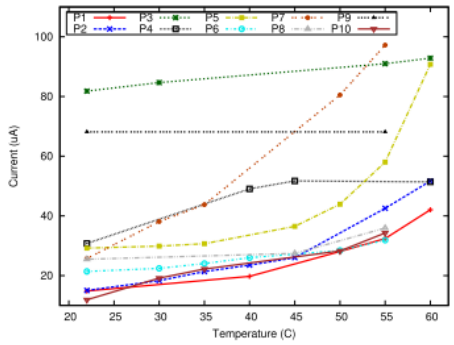

Figure 3. Sleep power variation across temperature for ten instances of nominally identical ARM Cortex M3 processors.

To utilize this variation, we introduce a duty cycle abstraction for operating system (TinyOS) that allows applications to explicitly specify lifetime and minimum duty cycle requirements for individual tasks. The software dynamically adjusts duty cycle rates so that overall quality of service is maximized in the presence of power variability. Results show that variability-aware duty cycling yields a 3–22x improvement in total active time over conventional worst-case designs based on data sheets that do not even meet the required lifetime targets, with an average improvement of 6.4x across a wide variety of deployment scenarios based on collected temperature traces. Using a target localization application, our results show that a variability-aware duty cycle yields a 50% improvement in the sensor data over the one based on worst-case estimations.

B. *Application Adaptation of Hardware Variations [1]*

Applications are often required to be reconfigurable and adaptive, e.g. video encoding and decoding, multimedia stream mining, gaming, embedded sensing, etc. They are capable of operating in various configurations by adapting to certain input or environmental conditions in turn producing similar or different quality of service. This allows us to use variation-affected hardware to drive application adaptation. From IC design and manufacturing point of view, "hardware-signature" based application adaptation is an easy and inexpensive (to implement) means that can better use application requirements and manage yield-cost-quality tradeoffs in current design flows. Such adaptation attempts to find the optimal software operating configuration that maximizes output application quality within application execution time constraints. In this case, the hardware signature corresponds to measured frequency of one or more independent components (motion estimation, entropy encoding, transform, etc) of a H.264 video encoder. The quality-complexity tradeoff depicted in Figure 4 shows various software configurations that UNO adaptation can pick from using the actual hardware signature. Adaptation can result in significant yield improvements (as much as 30% points at 0% hardware overdesign), a reduction in hardware overdesign (by as much as 8% points at 80% yield) as well as application quality improvements (about 2.0 dB increase in average peak-signal-to-noise ratio at 70% yield point).

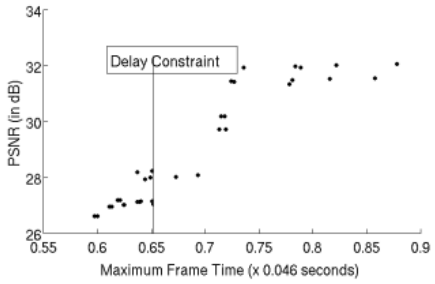

Figure 4. Different operating configurations of the H.264 encoder.

III. CONCLUSIONS

Hardware-enabled sensing and adaptation of the computing machines holds significant promise for continued reduction in the cost of microelectronic system designs and improvements in their reliable operations. Our early results specifically point to advantages in performance-constrained multimedia applications as well as power-constrained embedded applications. The UNO computing paradigms opens up several research challenges ranging from inexpensive monitoring methods for hardware signatures to variation-aware operating system adaptation mechanisms.

IV. REFERENCES

[1] A. Pant, P. Gupta, M. van der Schaar, "Appadapt: Opportunistic application adaptation in presence of hardware variation," *IEEE Transactions on VLSI*, 2011 (to appear).

[2] L. Wanner, R. Balani, S. Zahedi, C. Apte, P. Gupta, M. Srivastava, "Variability Aware Duty Cycle Scheduling in Long Running Embedded Sensing Systems," *DATE*, 2011.

[3] P. Kulkarni, P. Gupta, and M. Ercegovac, "Trading accuracy for power in a multiplier architecture," *Journal of Low Power Electronics*, 2011.

[4] Todd Austin, David Blaauw, Trevor Mudge, and Krisztián Flautner, "Making Typical Silicon Matter with Razor," *IEEE Computer*, 2004.

[5] Vinay K. Chippa, Debabrata Mohapatra, Anand Raghunathan, Kaushik Roy, Srimat T. Chakradhar, "Scalable effort hardware design: exploiting algorithmic resilience for energy efficiency", *DAC*, 2010.

[6] Doochul Shin and Sandeep K. Gupta. "Approximate logic synthesis for error tolerant applications", *DATE*, 2010.

[7] L. Leem, H. Cho, J. Bau, Q. Jacobson, S. Mitra, "Error-resilient system architecture for probabilistic applications," *DATE*, 2010.

[8] A.B. Kahng, S. Kang, R. Kumar, J. Sartori, "Designing processors from the ground up to allow voltage/reliability tradeoffs," *HPCA*, 2010.

[9] K. Jeong, A. B. Kahng and K. Samadi, "Impacts of Guardband Reduction on Design Process Outcomes: A Quantitative Approach", *IEEE Transactions on Semiconductor Manufacturing* 22(4), 2009