Power Management of Multi-Core Chips:

Challenges and Pitfalls

Pradip Bose, Alper Buyuktosunoglu, John A. Darringer, Meeta S. Gupta, Michael B. Healy, Hans Jacobson,
Indira Nair, Jude A. Rivers, Jeonghee Shin, Augusto Vega, Alan J. Weger
IBM T. J. Watson Research Center, Yorktown Heights, NY, USA

pbose@us.ibm.com

***Abstract*--Modern processor systems are equipped with on-chip or on-board power controllers. In this paper, we examine the challenges and pitfalls in architecting such dynamic power management control systems. A key question that we pose is: How to ensure that such managed systems are "energy-secure" and how to pursue pre-silicon modeling to ensure such security? In other words, we address the robustness and security issues of such systems. We discuss new advances in energy-secure power management, starting with an assessment of potential vulnerabilities in systems that do not address such issues up front.**

I. INTRODUCTION

The "power wall" [1] has forced chip and system architects to design with smaller margins between nominal and worst-case operating points. Smaller voltage margins make processors more vulnerable to inductive noise and single-event upsets induced by high energy particles. In fact, the power wall is forcing a trend towards "better than worst case" design. Lower power is achieved at the expense of tolerating occasional errors [2] that the processor is able to recover from. Alternatively, a slight performance hit is incurred in order to proactively prevent a circuit failure [3].

Dynamic power and thermal management control loops have already become an integral part of chip and system design [3-6]. Such management architectures allow the user and the system to work with changing demands for performance, while adjusting the power envelope accordingly, instead of operating always at the worst-case power consumption corner. In this paper, we examine the challenges and pitfalls in architecting such dynamic power management control systems at the chip or system level.

A key question that we pose is: how to ensure that such managed systems are "energy-secure" and not just energy-efficient on average? In other words:

- What are the challenges in verifying that the system will *always* meet the energy-related behavioral specifications?
- Can one identify corner-case scenarios where such management algorithms may be exploited to make the system unstable or unreliable by launching a malicious virus program?
- What intelligent safeguards must future dynamically managed systems possess to ensure that such reliability or security holes do not exist?

We discuss new advances in intelligent, energy-secure system architecture research. In section 2, we provide a summary overview of multi-core dynamic power management (DPM) research as published in prior work. We also describe the pre-silicon modeling infrastructure that is used in the definition of baseline power management architecture for multi-core processors. In section 3, we address the issues related to pre-silicon verification of a given multi-core power management protocol specification. In section 4, we focus on the problem of reliability-security "holes" in dynamic power management controllers – with a specific example. In section 5, we propose a particular solution approach (referred to as guarded power management) that can facilitate the progress towards designing dynamic power-thermal management controllers that are truly energy-secure. We conclude in section 6.

II. DYNAMIC POWER MANAGEMENT

In this section, we present a view of early-stage definition and modeling of baseline power management algorithms for multi-core chips. Such algorithms generally presume the existence of an on-chip or on-board power management controller, supported by a firmware-software system stack.

A. Dynamic Voltage-Frequency Scaling and Power Gating

In prior work [7-9], we describe some of the promising multi-core power management algorithms that yield significant benefit, when dealing with a specific control knob: that of dynamic voltage and frequency scaling (DVFS). Dynamic power gating (DPG) algorithms [10, 11] are targeted to reduce power by cutting off the power supply to unused resources. Depending on the accuracy of

978-3-9810801-8-6/DATE12/©2012 EDAA

the predictive control that drives such gating, the power savings could be very substantial, with minimal performance loss.

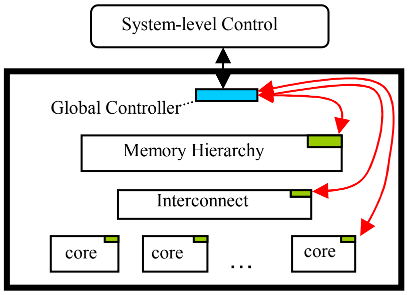


Figure 1. Hierarchical management and control

The generalized, high-level view of on-chip dynamic management is depicted in Figure 1. The design calls for a distributed monitor architecture that feeds sensed power/thermal, performance and reliability metrics to the on-chip global controller. Each resource may have built-in, local control mechanisms that allow actuation of mitigation knobs that are needed to react autonomously and at high speed, in response to localized problems that might demand immediate attention. Examples of such locally actuated knobs are: instruction fetch-gating [12], and dynamic frequency scaling in response to voltage dips [3]. The global controller coordinates across the set of localized actuation knobs and initiates chip-level mitigation actions, with directives from the system-level controller. For example, the system-level controller may react to a power emergency and direct the targeted chip to operate within a reduced power budget. In response, the on-chip global controller may decide to actuate the DVFS knob across the individual cores, in such a manner as to honor the system-specified power budget, at minimum performance cost. Experimental evaluation of the power-performance benefit of such multi-core DVFS algorithms (e.g. [7-9]) is essential, in order to design the power management architecture (and micro-architecture) for such systems.

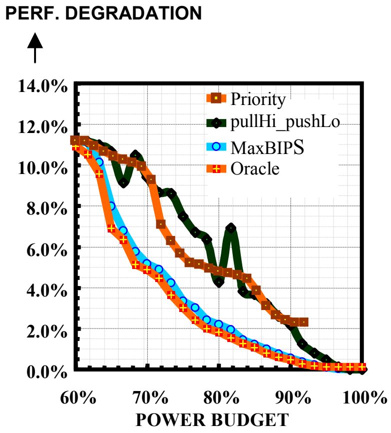


Figure 2. Power-performance trade-offs in multi-core DVFS

Figure 2 shows the power-performance characteristics of a multi-core processor with a global power controller. This is for a hypothetical 4-core system, with assumed per-core dynamic voltage and frequency scaling (DVFS) capability [7]. For each algorithm, the figure shows the chip-level throughput performance degradation as a function of chip-level power budget, expressed as a percentage of the nominal (or baseline) operational power. As described in [7], a particularly efficient heuristic control algorithm, called MaxBIPS that we devised, is able to approach the idealized power-performance characteristics of an oracular algorithm (labeled as "Oracle" in Figure 2) that has perfect, *a priori* knowledge about per-core workload characteristics and phase changes.

Figure 3 illustrates the power reduction potential, when unit-level predictive power gating is applied to a state-of-the-art super scalar microprocessor [10].

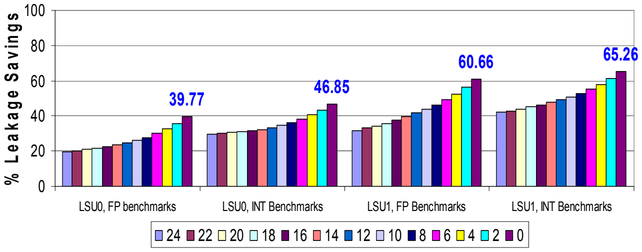


Figure 3. Power gating potential for LSU0 and LSU1 as a function of the breakeven point (varied from 0 to 24 cycles); for FP and INT benchmarks

It is seen that the leakage power savings may be up to ~40% for LSU0, while running SPECfp benchmarks and may be up to ~65% for LSU1, while running SPECint benchmarks. LSU0 and LSU1 refer to the load-store unit execution pipelines that support the address generation and load-store cache access logic within the super scalar processor.

B. Modeling Support for Power Management

In order to model the benefit of any particular power management algorithm, architects have to rely on pre-silicon power-performance simulators. Cycle-accurate performance simulators are augmented with energy models that are derived from circuit-level characterization of the underlying design macros [13, 14].

SLATE (System-Level Analysis Tool for Early Exploration) [8] was developed to enable system architects to quickly estimate performance and power dissipation at the early stages of design, before implementation. It provides a library of POWER system components, including cores, caches, controllers and a coherent bus-based interconnect for composing multi-core systems. Each component has a cycle-accurate transaction-level model written in SystemC. The models are trace-driven and produce utilization statistics for the major functional units. These statistics are also used to estimate power dissipation. SLATE was used to evaluate alternative high-level designs for meeting performance requirements for specific benchmarks, for evaluating alternative power management strategies and for testing a digital phase-locked loop (PLL) implementation.

SLATE's power modeling uses a methodology described in [13, 14]. Since estimating clock power in clock-gated designs can be a major source of inaccuracy, the power model pays special attention for accurate estimation of clock-gating factors. The power model incorporates a set of microarchitectural event-based equations specifying when each latch-bank is clocked. Each unit designer associates each latch-bank, array or register file with an expression that specifies its clocking in terms of microarchitectural events used by the performance simulator [14]. This allows clock-gating factors to be estimated at a very fine spatial granularity and enables more accurate power estimation

SLATE's component-based structure made it straightforward to add a power management module to evaluate algorithms for managing the power dissipation of a multi-core system. SLATE was also used to test a digital PLL implementation that was proposed for more efficient power management. POWER7 provides the EnergyScale system-level performance-aware energy management system [6]. It has multiple sensors that provide performance, utilization and activity measurements. It also has critical path monitors (CPM) to detect timing issues and assist in choosing optimal frequency and voltage settings. Each POWER7 core can operate over a range of -50% to +10% of nominal frequency and uses autonomic circuit timing to reduce wasteful guard banding [3]. Conventional guard-banding is static and uses conservative voltage margins to guard against potential worst-case conditions. The CPM coupled with a digital PLL (DPLL) can detect potential concerns and correct them dynamically, yielding more energy-efficient operation [3]. In current work, we are studying the robustness of the CPM-DPLL control loop, in terms of stability and immunity to deliberately injected noise.

C. Evolution to an Integrated Modeling Framework

Point tools and analysis methods of the type alluded to before are valuable, but increasingly, there is the need for an integrated framework that supports such point tools within a common modeling environment supported by a central design database. Domain-specific server chips are complex Systems-on-a-Chip (SoC), with particularly tight power/thermal and resource budgets. To assist in the early stages of design, we are exploring the development of an Early Chip Planner [18] (see Figure 4). This tool brings together many diverse forms of analysis (e.g. power, performance, temperature and reliability). It provides a unified representation of the design at the early stages to drive the analysis tools. This helps automate the process of considering design tradeoffs. It also captures the design-decisions, assumptions and constraints used in reaching the design point and passes this information on to the next stage of design implementation. There is a "spreadsheet" interface for the system architect to input and analyze results, as well as prior chip design data conveniently viewable in a standard format.

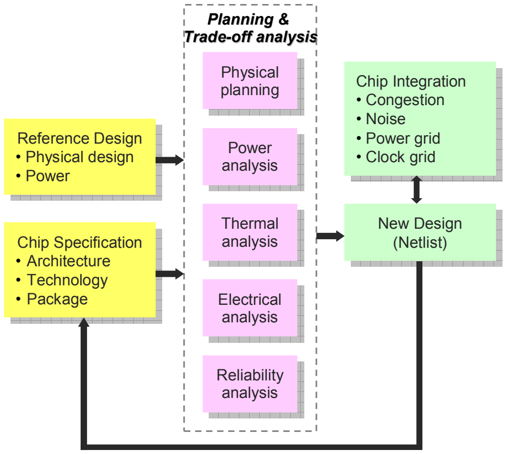


Figure 4. Early chip planner – functional overview

III. VERIFICATION COMPLEXITY OF POWER MANAGEMENT PROTOCOLS

In this section, we briefly examine the problem of pre-silicon verification of multi-core, dynamic power management algorithms, based on our team's prior work [15]. Let us consider the case of a global (on-chip or on-board) controller, which manages the total power allocation across a number of cores. Figure 6 depicts the case under consideration for n cores (n = 3 for illustration). We assume that the task of the global controller is to enforce a total power budget across the n cores, where the power budget value is provided by a higher level system manager.

The task of the global controller consists of monitoring power usage across each of the cores and actuating voltage (V_{dd}) and/or frequency (F) of cores as needed to maintain the system power within the specified budget.

In the graph (Figure 5), the power budget is indicated by a horizontal line; and, above that is the "Max Power" line that indicates the absolute maximum in power consumption, as dictated by the package (cooling) limits.

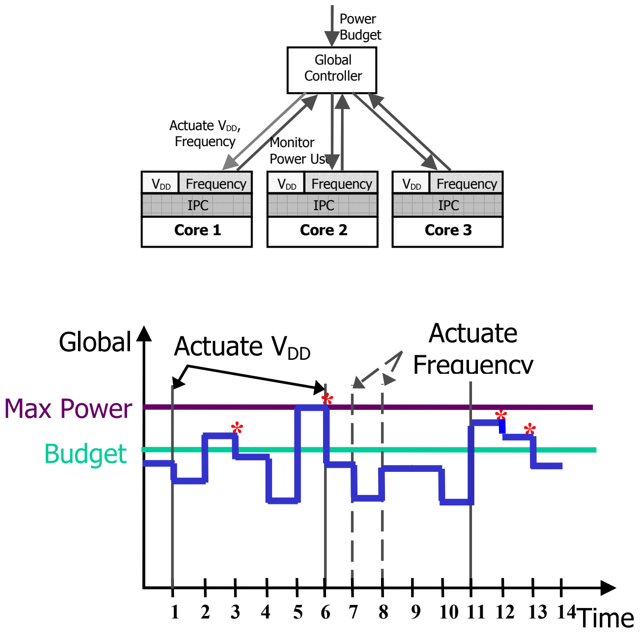


Figure 5. Formulation of global power control problem for analysis of verification complexity

The curve that shows periodic changes is the actual power consumption that results from the workload changes combined with the effect of periodic actuation of the V_{dd} and F knobs for each core. In the system above, V_{dd}-F joint actuations are assumed to be possible every 500 µsecs, with frequency-only (F) actuations possible every 100 µsecs. Occasionally, the estimated actuations may be wrong, in that the power exceeds the stipulated budget (as indicated by the * markings) temporarily. This causes the global controller to immediately decrease the V_{dd} and/or F to enforce adherence to the budget.

The power model used is in this case a very simple one, formulated as an analytical function of the observed instructions per cycle (IPC) and the current (Vdd, F) setting of the core. As described in [15], the state-space of the global controller operation can be specified in terms of the V_{dd}, F and IPC of each core. Both deterministic and probabilistic model checking based analysis can be pursued.

Figure 6 shows the feedback control loop that is implicit in the above problem formulation. The control algorithm tries to enforce the power to meet the specified budget (limit), after periodic monitoring of the utilization across individual cores. Figure 7 shows a snapshot of the results of the analysis associated with the control algorithm formulated (Figures 5 and 6). As indicated in Figure 7, the problem may be generalized by assuming a clustered control system, with m cores per controller (CPC). With the number of cores n = 3, m (=CPC) may be 1, 2 or 3. The CPC = 3 corresponds to a single centralized controller.

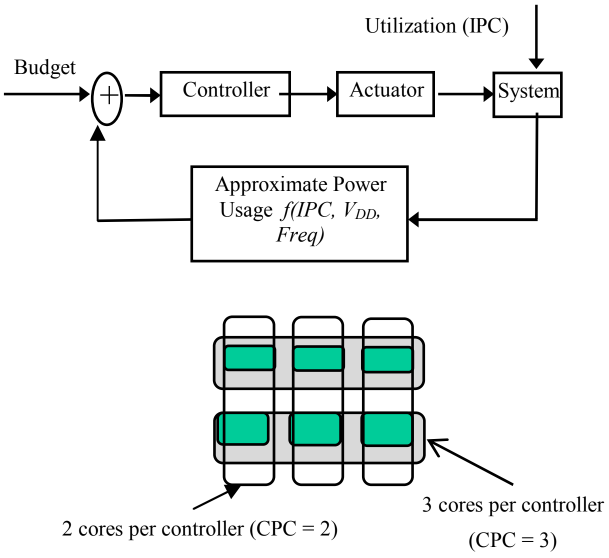

Utilization (IPC)
Budget
+
Controller
Actuator
System
Approximate Power Usage f(IPC, V_DD, Freq)
2 cores per controller (CPC = 2)
3 cores per controller (CPC = 3)

Figure 6. Depiction of the feedback control loop

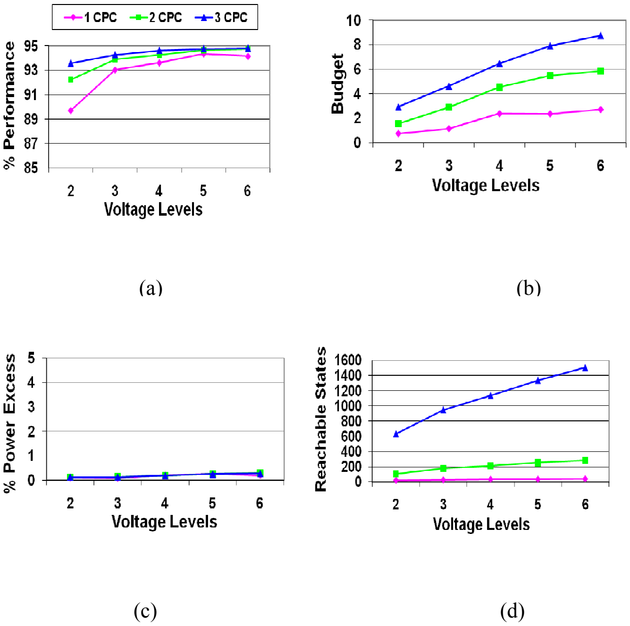
As depicted Figure 7(d), when the number of allowed voltage (V_{dd}) levels is varied from 2 to 6, the number of reachable states (which is an index of verification complexity) increases quite sharply for the centralized case (CPC = 3). The complexity growth is much more reasonable for clustered control (CPC = 1 or 2). Figure 7(a) show the percentage increase in multi-core performance (relative to a baseline design without DVFS), as the number of available V_{dd} levels is increased. We see that the performance gain is highest for the fully centralized case (CPC=3), although the difference across CPC settings becomes negligible, as the number of DVFS levels is increased to 6. Figure 7(b) plots the percentage of sampling (or monitoring) intervals over which the stipulated power budget is exceeded. Figure 7(c) depicts the average excess in power over the stipulated budget, as a function of the number of V_{dd} levels. We see that although the average power overrun value is small (regardless of the number of V_{dd} levels), the number of violations increases with the number of V_{dd} levels used and with increased values of CPC.

Thus, clustered control is better from the point of view of safety or robustness of the algorithm; and, smaller number of V_{dd} levels is preferred. The analysis shown in Figure 8 represents average data values obtained across selected SPEC2000 application workloads [15].

IV. Robustness And Security Issues In Power Management

Dynamic power management systems, when implemented as hierarchical, closed-loop feedback control systems, present robustness and security issues as a matter of course.

These security issues arise because feedback control systems are known to have regions of unstable (or unintended) behavior within the full range of their control parameters.



(a) (b)

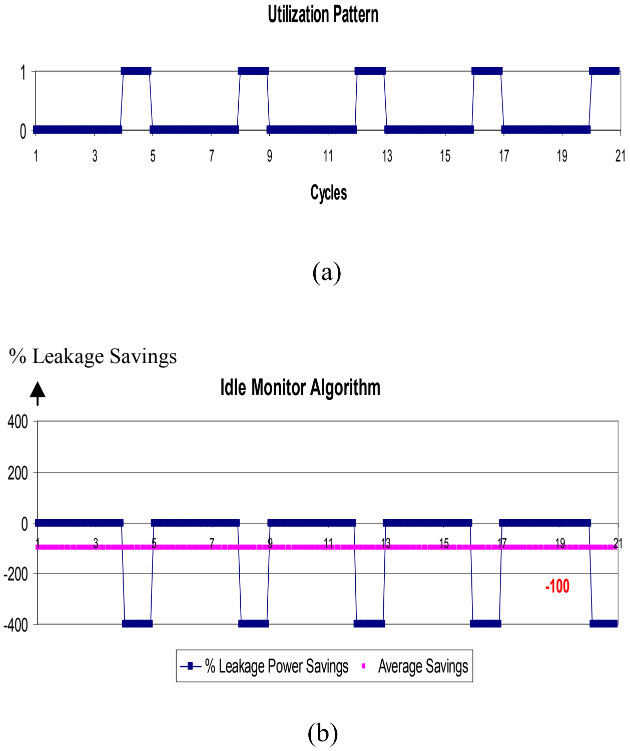
(c) (d)

Figure 7. Performance, power and verification complexity characteristics of global controller for selected SPEC2000 applications

Consider for example, the management of unit-level power gating within a single processor. Figure 8(a) shows a periodic utilization profile of a monitored resource within a processsor, in response to a large-iteration tight loop workload. A classical predictive power-gating algorithm [10] would direct the controller to turning off the resource, after observing it to be idle for a specified number of cycles.

Depending on the threshold value of the so-called "idle-detect" parameter [10] and the workload periodicity profile, the controller may turn off the resource just before the resource is again needed by the application program. Thus, because of the repeated invocation of the ill-timed power gate command, the power savings seen is actually *negative*, as illustrated in Figure 8(b).

In general, as discussed in [16], corner-case workloads (both real and synthetic) can be made to deceive (or disrupt) a power gating controller to the point where there is a significant power overrun and/or a system performance shortfall. Similarly, in [17], we show that per-core power gating (PCPG) algorithms, if designed without protection can pose robustness and security problems as well.



(a)

(b)

Figure 8. Illustration of pitfall in power gating control algorithm

V. Guarded Power Management

Having described the inherent verification, safety and reliability (or security) issues in closed-loop feedback control based power management algorithms, we propose a "guarded" control mechanism as a viable strategy to boost up the overall robustness attributes.

A. Energy Security via Guarding

Figure 9 shows the high-level concept architecture of a guarded power gating controller, used to orchestrate a PCPG mechanism for a multi-core processor. The baseline power gating manager implements a simple gating algorithm – for example, one based on monitoring of idle period duration. The guard mechanism constantly monitors the effectiveness of the baseline manager, in terms of power, performance and safety related metrics associated with the managed system. When anomalies are detected, the power gating manager is turned off by the guard mechanism; and, in extreme cases, the system administrator may be notified if a deliberate (virus) attack is suspected. This is our basic research strategy in pursuing the goal of energy-secure computing.

The detailed description of how a simple "guard" may be implemented within the system software (management firmware/OS/hypervisor) hierarchy is described in our most recent power-gating related papers [16, 17]. The first paper [16] describes and evaluates the benefit of guarding for unit-level power gating within each core; and the second one [17] makes a case for guarded PCPG algorithms.

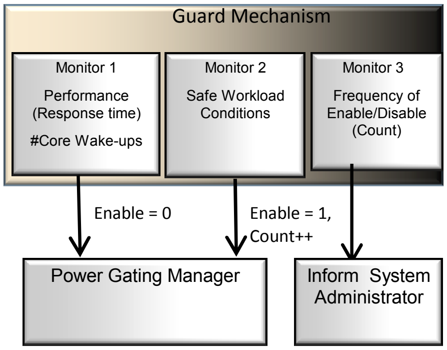


Figure 9. The concept of guarded power management (applied to PCPG)

It can be argued that instead of trying to strengthen the baseline management algorithm (which increases the verification complexity considerably), it is better to have a *simple* baseline algorithm protected by a *simple* guard mechanism. Verification complexity analysis (see section III) would show that the guarded management algorithm scales better with the number of cores than a highly complicated baseline algorithm that is designed with an attempt to make it fully secure against corner-case workloads (e.g. virus attacks).

B. Results Summary for Guarded Power Gating

For fine-grain unit-level power gating we have developed guard mechanisms [16] that generally prevent negative energy savings. In a few cases where it is possible, the marginal energy penalty is less than 1%. Worst-case performance degradation margin is limited to 2%. Similar energy-secure management benefits are reported in our coarse-grain PCPG research as well [17]. A detailed discussion of our latest ideas and results in energy-secure computing is omitted here for brevity.

VI. CONCLUSION

In this paper, we provided an overview of the multi-core power management problem, and pointed to the inherent pitfalls in terms of targeted robustness and verification complexity of closed loop feedback control based managers. We sketched the idea of guarded power management as a solution approach to yield energy-secure system architectures. In future work, we will provide details of our simulation-based projection of power management vulnerabilities, backed by direct hardware-based experiments. We will also keep addressing mitigation solutions to the identified problems.

ACKNOWLEDGMENT

This research is supported by the Defense Advanced Research Projects Agency under SPAWAR Pacific Award No. N66001-11-C-4027. The views, opinions, and/or findings contained in this paper are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of DARPA or the Department of Defense.

REFERENCES

[1] P. Bose, "The Power Wall," in Encyclopedia of Parallel Computing, David Padua ed., Springer, 2011.

[2] D. Ernst et al., "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation," *Proc. 36th Symp. on Microarchitecture, MICRO-36,* Dec. 2003.

[3] C. Lefurgy et al., "Active management of timing guardband to save energy in POWER7," *Proc. 44th Symp. on Microarchitecture, MICRO-44,* Dec. 2011.

[4] D. Albonesi et al., "Dynamically tuning processor resources with adaptive processing," *IEEE Computer*, 36, 12, (2003), 43–51.

[5] R. Singhal, "Inside Intel^{R} Core^{TM} Microarchitecture (Nehalem)," *Digest of the Hot Chips conference*, Aug.2008.

[6] M. Floyd et al., "Introducing the adaptive energy management features of the POWER7 chip," *IEEE Micro,* vol. 31, no. 2, March/April, 2011.

[7] C. Isci et al. "An analysis of efficient multi-core global power management policies: maximizing performance for a given power budget," *Proc. 39th Symp. on Microarchitecture, MICRO-39,* Dec. 2006.

[8] R. Bergamaschi et al., "Exploring power management in multi-core systems," *Proc. Asia-Pacific Des. Autom. Conf.,* ASP-DAC*,* Jan. 2008.

[9] J. Sharkey, A. Buyuktosunoglu, P. Bose, "Evaluating design tradeoffs in on-chip power management for CMPs," *Proc. Int'l. Symp. on Low Power Electronics and Design, ISLPED,* Aug. 2007.

[10] Z. Hu et al., "Microarchitectural techniques for power-gating of execution units," *Proc. Int'l. Symp. on Low Power Electronics and Design, ISLPED,* Aug. 2004.

[11] D. Meisner et al., "PowerNap: eliminating server idle power," *Proc. Arch. Support for Prog. Langs. & Operating Sys (ASPLOS),* March 2009.

[12] A. Buyuktosunoglu, T. Karkhanis, D. Albonesi, P. Bose, "Energy efficient co-adaptive instruction fetch and issue," *Proc. the Int'l. Symp. on Comp. Arch. ISCA*, June 2003.

[13] D. Brooks et al., "New Methodology for Early-Stage Microarchitecture-Level Power-Performance Analysis of Microprocessors," *IBM Journal of Research & Development*, Vol.47, No.5/6, September/November, 2003.

[14] H. Jacobson et al., "Abstraction and microarchitecture scaling in early-stage power modeling", *Proc. Int'l. Symp. on High-Performance Computer Architecture (HPCA),* Feb. 2011.

[15] A. Lungu et al., "Multicore power management: ensuring robustness via early-stage formal verification." *Proc. 7th IEEE Int'l. Conf. on Formal Methods & Models for Codesign (MEMOCODE)*, July 2009.

[16] A. Lungu, P. Bose, A. Buyuktosunoglu, D. Sorin, "Dynamic power gating with quality guarantees," *Proc. Int'l. Symp. on Low Power Electronics and Design, ISLPED*, Aug. 2009.

[17] N. Madan, A. Buyuktosunoglu, P. Bose, M. Annavaram, "A case for guarded power gating in multi-core processors," *Proc. 17th Ann. Int'l. Symp. on High Performance Computer Architecture (HPCA),* Feb. 2011.

[18] J. Shin et al., "Early chip planning cockpit," *Proc. Design Automation and Test in Europe (DATE),* March 2011.