Coordinated Power, Energy, and Temperature Management for High-Performance Microprocessors

Heather Hanson[*] Stephen W. Keckler Doug Burger

Computer Architecture and Technology Laboratory
Department of Computer Sciences
The University of Texas at Austin
cart@cs.utexas.edu - www.cs.utexas.edu/users/cart
[*]Department of Electrical and Computer Engineering
IBM Technical Contact : Rob Bell

Abstract

High-performance desktop and server processor design is hampered by constraints on chip power, temperature, and energy. We have investigated dynamic power characteristics of high-performance pipelines and static energy management techniques for large on-chip caches in previous work. Our analysis indicates that a large portion of the power budget is devoted to overhead, spending power to achieve incremental performance improvements.

Existing approaches to managing both static and dynamic energy have proven useful but have room for improvement as the ratio of cost to benefit shifts with increased power liability per transistor in future processor generations. In response, we are currently developing a coordinated microprocessor manager that enables high performance throughout a wide range of throughput, power, energy, and temperature targets for the next generation of processor designs. In addition to budgeting dynamic power, the manager will coordinate leakage control mechanisms and balance resource activity levels to avoid thermal emergencies due to hot spots on die. This report summarizes the foundations of this work and our current status in this endeavor.

1 Introduction

A common saying in the VLSI design community is “transistors are free” due to Moore’s Law integration trends. Investing in large quantities of transistors to improve performance has proven effective in previous high-performance processor generations. As transistors and interconnect scaled to smaller dimensions, the processor die had room to fit on-chip caches, then larger caches, with multiple levels of memory hierarchy. Out-of-order and multiple issue architectures relied on many transistors devoted to support structures such as branch history tables, queues, and reorder buffers. Recent generations like the POWER4 have had the available transistor count to manufacture two high-performance cores with memory and communication support per die [1].

1.1 Power Liability

Boosting performance by using more transistors and interconnect for larger structures, wider pipelines, and sophisticated prediction mechanisms incurs a cost for both dynamic power and static leakage power. A transistor’s dynamic power depends upon the capacitance, voltage supply, and switching rate. Although capacitance per transistor and voltage supply scale down with successive generations of fabrication technology, an increase in switching rate and transistor density causes the overall dynamic power to escalate. For static power, fabrication trends indicate several impending problems. Subthreshold leakage and gate oxide tunneling create currents through nominally “off” devices through different paths within the transistor. Subthreshold leakage is exponentially dependent on temperature, so heat dissipated from a high-power device causes even more static power. Both types of leakage current grow as device dimensions shrink and adding more devices per die compounds the problem with more, leakier devices closely packed together. Another consideration is the variability of leakage current due to process variations. As Kim points out [13], a ten percent change in gate length can cause a factor of three difference in subthreshold leakage current. Semiconductor fabrication trends indicate that each transistor will bear a growing power liability in near-term future processor generations.

1.2 Power Management

Design engineers have taken advantage of semiconductor integration and packaging advances that allow more transistors and wires fabricated per processor die than contemporary power distribution and cooling technology could handle without intervention from power management techniques. Most of these techniques manage power, energy, and/or temperature by temporarily disabling features that were included on the chip to enhance performance, such as extra structure capacity or fast clock rate. Judicious use of this type of technique can allow performance features to be used as needed. Combined with advances in power distribution, manufacturing, and packaging technologies, these power management techniques have enabled the current generation of high-transistor density, high-power, high-performance processors. As fabrication trends push static power higher and limit reductions in supply voltage to temper dynamic power, the ratio of power cost to performance benefit shifts. Individual transistors' contribution to performance decreases as many transistors add incremental performance: for example, doubling cache capacity of a 4-way associative cache from 64KB to 128KB reduced the miss rate from 0.9% to 0.6% in one study [9] and doubling the number of execution units in a superscalar pipeline does not double performance [15]. Meanwhile, the power liability for each additional transistor grows. Effective power management is crucial to sustain future generations of high-performance chips as the power liability per transistor grows and Moore's Law integration trends offer more transistors per die each generation.

1.3 Next Generation

Existing solutions for power management techniques have been sufficient for the current generation of power-constrained products but do not provide the comprehensive management necessary for future designs for several reasons. First, a greater percentage of transistors will be tightly controlled to meet projected power budgets [11] in the near-term future. New control mechanisms will be required for structures not currently managed. Second, power, energy and thermal considerations will require management techniques with distinct, possibly conflicting objectives. A collection of individual techniques may be enabled in destructive or ineffective combinations. Third, existing open-loop control techniques do not guarantee effective operation throughout the wide range of process variability, application space, and operating conditions. The policy-driven approach of enabling a power (or other metric) saving technique based on a pre-defined set of events such as "after 1000 cycles of inactivity, transition to sleep mode" does not take into account the effectiveness of the action at run-time.

Our next step is to find a solution that enables high-performance processor design by intelligently coordinating power, energy, and temperature management. Circuit, fabrication, packaging, and architecture decisions determine in large part the total power, energy, and temperature limits months or years before a processor is in use. Our research focus is the microarchitectural components that can provide run-time flexibility while leveraging those early decisions.

This document outlines the research and development of a resource manager designed for future generations of high-performance microprocessors. Section 2 describes our research investigations into dynamic and static power characteristics and summarizes related work in processor power management. Section 3 discusses limitations of open-loop management techniques for future generations of processors and proposes a solution for coordinating power, energy, temperature, and performance management. Section 4 describes our evaluation methodology and Section 5 concludes this status report with a summary of work to date and overview of ongoing research.

2 Power Analysis

Several research studies address problems of power, energy, and temperature management, providing insight for resolving conflicts between high performance and reasonable resource use. In recent work, we investigated the dynamic power characteristics of a super-scalar pipeline [16]. In a previous study, we compared the three types of static energy management [8].

2.1 Dynamic Power

In [16] we tracked dynamic power consumption throughout a pipeline model of the Alpha 21264 processor, noting the power tax of mis-prediction and over-provisioning. The pipeline uses several predictive mechanisms to perform potentially useful work faster, which could improve throughput and reduce energy use. However, prediction also provides an opportunity to waste power on processing speculative instructions that are not committed. Furthermore, in the event of a mis-prediction, the pipeline wastes power correcting the error and re-executing instructions. In our study, we found that mis-prediction accounted for approximately 6% of pipeline energy use. Over-provisioned structures that are designed for maximum throughput but not fully used by typical programs accounted for about 17% of the pipeline energy. In this study, energy used by the global clock network is tabulated separately and not included in either mis-speculation or over-provisioning. Figure 1 shows the distribution of energy throughout the pipeline for the clock network, committed

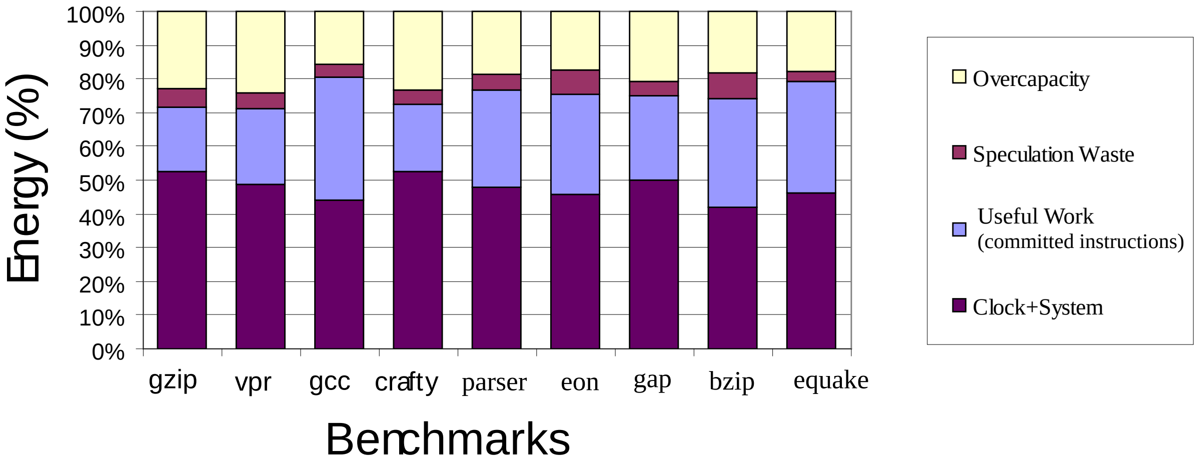

Figure 1. Energy Expenditure

(useful) instructions, over-provisioned structures, and misspeculation. As illustrated in the figure, a smaller portion of energy is devoted directly to the useful work of processing instructions than to combined power overhead. We predict that more aggressive out-of-order processors introduced since the Alpha 21264 will have more power overhead from over-provisioning and mis-speculation.

With the large power overhead of multiple-issue, out-of-order pipelines, management techniques achieve significant power savings by turning off idle or non-critical components, such as the techniques of clock gating [7] or controlling the power overhead with techniques like pipeline gating [14]. Other approaches include dynamic voltage and frequency scaling [5], [6] that reduce both the switching rate and the power cost per switch.

2.2 Static Power

We evaluated static power management techniques for on-chip caches in [8], noting the control techniques' effectiveness in energy reduction and effect on processor performance. We compared three techniques against a high-performance cache without leakage control. One technique named *dual-*V_t uses a combination of low-leakage transistors in SRAM cells and faster, higher-leakage transistors in the control circuitry. A second technique, *gated-*V_{dd}, provides variable leakage control for each cache line and destroys the cell contents during the low-leakage mode. A third technique, *MTCMOS* dynamically adjusts the effective threshold voltage of SRAM cell transistors, which perserves cell contents in standby mode but incurs additional time to wake up cells prior to read and write accesses. Without leakage control, a large secondary cache composed of fast, leaky transistors consumed an excessive amount of energy, and even smaller primary caches lost substantial current through subthreshold leakage. Each leakage-control technique in the study effectively curbed leakage current with varying degrees of performance degradation. With our estimates of leakage current and read access time, the *MTCMOS* cache was the most effective type of the level-1 caches and dual-V_T was the best unified secondary cache.

Most static power management to date targets on-chip caches. Large memory arrays account for nearly half of the total transistor count and generally contain redundant state that can be retrieved from alternate locations if necessary, which allow aggressive energy-saving techniques. Instruction cache resizing [18], cache decay leakage control [12], drowsy caches [4], and the techniques we studied all manage static leakage by directing portions of the cache into a low-leakage mode. Future generations are likely to require leakage control throughout the chip, not only in cache structures. One technique applicable to other components or the full chip is supply voltage reduction in dynamic voltage and frequency scaling.

2.3 Temperature

Temperature management has also emerged as a critical issue [17]. Without temperature management, chip and system failure rates will increase to dire levels. One example of temperature management in a commercial product is the Thermal Control Circuit in the Pentium4 processor that intermittently stops processor clocks when the chip reaches high temperatures, reducing chip activity and allowing the device to cool [10].

3 Power, Temperature, and Energy Management

Simply extending the existing class of microarchitectural management techniques to encompass power, energy, and temperature constraints falls short of a robust management system. Contemporary power-saving or energy-saving techniques usually sacrifice performance or rely on schedule slack or idle components, which may be inappropriate for high-throughput pipelines with multiple threads or minimal slack. With transistor count out-pacing power distribution and cooling technology, new mechanisms will be required to control power, energy, and temperature as structures grow larger in capacity and new features emerge on the processor chip.

3.1 Independent Techniques

Extensive power management through individual policy-driven techniques poses two main concerns. First, management policies are determined with incomplete knowledge of physical environment, operating conditions, and application characteristics. If code profiling runs and processor simulations do not accurately match actual run-time conditions, the mismatch can lead to ineffective management. For example, changing the frequency and voltage settings based on recent program behavior via a performance monitor may provide excellent control for the test benchmark suite yet result in a pathological case for a customer's proprietary software.

Second, run-time events could repeatedly trigger conflicts between management policies. For example, an energy-saving policy sets the frequency at a fast rate for a program that can complete quickly, then turns off all units to conserve static energy. A separate temperature policy sets a lower frequency to cool the chip in the event of excessive heat dissipation. During program execution, the chip could breach a temperature threshold, causing oscillations between management mechanisms that trigger a slower frequency for cooling and faster frequency to optimize leakage. Avoiding such conflicts requires testing each combination of techniques, adding to the cost and complexity of processor verification.

3.2 Open-Loop Techniques

The basic problem with open-loop control policies is that they are unable to adapt to run-time conditions. They cannot provide a guarantee that the policy will, in fact, lower the chip temperature or curb leakage power, or save dynamic energy. Even with accurate estimates, parameters may vary between production lots or packaging styles. Conservative safety margins lead to missed opportunities at design time for enhanced performance and energy-efficiency throughout the chip [3].

3.3 Overview of Coordinated Approach

Our solution to the problem of providing high-throughput performance within the constraints of limited power, energy, and temperature levels is a coordinated, goal-oriented approach. Our manager chooses a goal of desired performance and operating conditions, then coordinates the available set of management mechanisms to reach the goal.

Unlike policy-driven decisions that react to specific events with pre-determined responses, such as the Pentium4 thermal control policy paraphrased as "if temperature exceeds the threshold, then enable intermittent clock gating", the goal-seeking approach is flexible. For example, our manager could enable clock gating in a thermal emergency like the Pentium 4 or migrate an active thread to another core, or simply reduce the frequency and voltage levels. A goal-driven management approach is able to adapt to a wider range of operating conditions and resource use, allowing the processor to run closer to the edge of power, temperature, and energy limits. A coordinated system can also ensure safe operating conditions for run-time environments and configurations not expected during design and validation phases.

3.4 Design Criteria

In order to be effective, our proposed manager must meet the following criteria:

- effective interaction over a range of environmental conditions and applications, with different program input for designing and testing mechanisms
- consistently meet energy, power, temperature, and performance targets
- guarantee safe operation

To be worth the design effort of creating the coordinated manager, the manager must also outperform a set of independently triggered techniques.

3.5 Implementation

The preliminary design of the coordinated manager is a small processor that shares the die with the primary core(s), similar to the service processor found on many IBM systems for extracurricular processing in support of system reliability and robustness. The manager could be built from a small, low-power commodity core with the addition of a

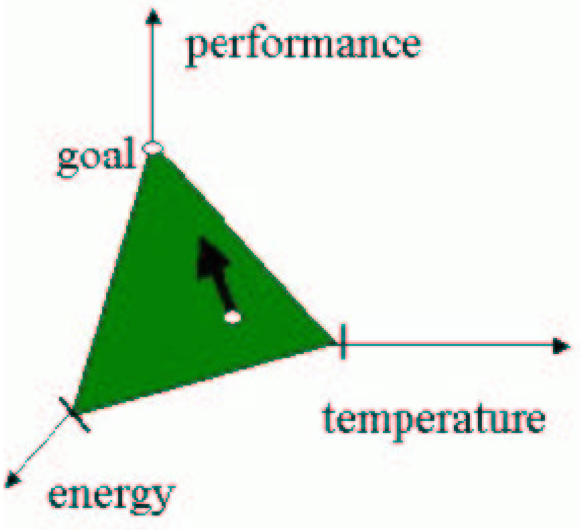

Figure 2. Goal-Driven Management

specialized communication interface for collecting physical sensor information such as temperature readings and event indicators such as cache miss rates and performance counters from the primary processor. An advanced implementation could be distributed through multiple chips or boards in a multi-processor system.

A hierarchy of intelligence gathering and processing components in the manager distributes decisions according to required response time: quick response for phenomena with shorter time constants, such as current spikes in the power distribution network, and longer intervals between decisions for slow-moving trends like gradual chip warming. The central component is a multi-criteria optimization algorithm that sorts priorities and balances conflicting goals for performance, power, energy, and temperature. The manager selects a goal, such as “maximum performance within set temperature and energy limits” and enables a coherent set of management mechanisms to achieve the goal. The closed-loop feedback system between the manager and sensors and performance counters provides continual updates on chip and system status, allowing the manager to intelligently tune its directives to achieve the desired response. The manager tracks system behavior and shifts goal objectives in synchrony with changing application demands and energy resources.

Figure 2 illustrates the goal-driven decision process for an example where the multi-criteria optimization algorithm maximizes performance while minimizing temperature and energy. The shaded area represents the operating space within energy and temperature limits and realistic performance expectations. The arrow delineates movement from the current state toward the goal as the manager adjusts the current set of power/energy/temperature management techniques to reduce energy and temperature while increasing performance.

A brief example of the management process follows. The coordinated manager’s goal is high throughput within limits of a strict upper bound on temperature and moderate power and energy thresholds. The processor is currently operating with a mid-range voltage level; sensor data indicate that the temperature is within an acceptable range and that the performance is less than the goal. The manager directs the voltage regulator to step up the supply voltage and monitors the temperature rise and performance counters, and continues to raise the frequency and voltage until achieving the desired performance target. If software application behavior causes a thermal spike, the manager takes immediate action to coordinate a response between the voltage, frequency, and activity migration controls, while postponing a cache leakage policy that would have created a temporary increase in write-back traffic at an inopportune moment. With coordinated information from multiple sources and a goal-driven algorithm, a hierarchical power/energy/temperature manager can adapt to the system environment and push the operating conditions to the edge of acceptable limits.

4 Evaluation

Our hypothesis is that a coordinated, goal-driven manager will provide better control than a collection of independent policy-driven mechanisms. Although meeting energy, power, or temperature targets—even moving targets—with a collection of unrelated management techniques may be possible, we hypothesize that an intelligent goal-driven resource manager can do a better job for future processors because it can accommodate variations between design and run-time environment and unexpected situations in both physical conditions and software applications. This section outlines the experimental methodology and reports the current status of the evaluation.

We are currently developing architectural simulation infrastructure to quantify the effect of power, temperature, and energy management decisions. In our pipeline study, we added a power model to a validated microarchitectural simulator that models the Alpha 21264 processor [16], [2]. In our current work, we have added the HotSpot [17] temperature estimator and plan to incorporate the HotLeakage static power tool. We are extending the simulator capabilities to model the behavior of power, energy, and temperature control techniques. The techniques included thus far are dynamic frequency and voltage scaling and cache leakage control. Next, we will incorporate dynamic pipeline resizing. Our initial benchmarks will include programs from the Spec2000 suite, but we will also examine high performance and embedded applications.

Initial experiments will establish a baseline for a collection of individual techniques. First, we will monitor the power, temperature, energy, and performance for each management mechanism separately using the best-known policy settings. Then, we will combine pairs of techniques, three techniques together, etc. and monitor the effects of each permutation. Data collected from the initial phase will form a database of independent techniques. In the second experimental phase, we will use our coordinated manager with the same collection of techniques to evaluate the effect of intelligent oversight. We will compare the performance, energy, power, and temperature behavior and any incidents of safety threshold violation, and use the experimental data to refine our algorithms.

5 Conclusion

High-performance desktop and server processor design is hampered by constraints on chip power, temperature, and energy, and contemporary power-saving or energy-saving techniques will be inadequate to support future high-throughput microprocessor designs. A grab-bag of interacting, ad-hoc optimizations is undesirable for a robust management system; instead, we are developing a coordinated microprocessor manager that enables high performance throughout a wide range of throughput, power, energy, and temperature targets.

The project is currently in a phase of building infrastructure for experimental simulation. We are extending a performance and power model of the Alpha 21264 processor to estimate temperature, and adding run-time management mechanisms such as dynamic frequency and voltage scaling, cache leakage abatement, and structure resizing. We have established design criteria and preliminary specifications for the coordinated management module. Our work continues with developing both simulation infrastructure and the coordinated manager, then comparing our design with existing approaches in microarchitectural simulation and applying classical linear and non-linear control theory to further develop the management algorithm.

References

[1] C. Anderson, J. Petrovich, J. Keaty, and G. Nusbaum. Physical design of a fourth-generation power ghz microprocessor. In *International Solid-State Circuits Conference*, pages 232 – 233, 451, 2001.

[2] R. Desikan, D. Burger, and S. W. Keckler. Measuring experimental error in microprocessor simulation. In *Proceedings of the 28th Annual Symposium on Computer Architecture*, pages 266–277, 2001.

[3] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge. Razor: A low-power pipeline based on circuit-level timing speculation. In *Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 7–18, 2003.

[4] K. Flautner, N. Kim, S. Martin, D. Blaauw, and T. Mudge. Drowsy caches: Simple techniques for reducing leakage power. *International Symposium on Computer Architecture*, pages 148 – 157, June 2002.

[5] K. Flautner and T. Mudge. Vertigo: Automatic performance-setting for linux. In *OSDI2002*, pages 105–116, 2001.

[6] K. Flautner, T. Mudge, and S. Reinhardt. Automatic performance setting for dynamic voltage scaling. In *7th Annual Int. Conf. Mobile Computing and Networking 2001*, pages 260–271, 2001.

[7] M. Gowan, L. Biro, and D. Jackson. Power considerations in the design of the alpha 21264 microprocessor. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 726–731. ACM/IEEE, June 1998.

[8] H. Hanson, M. Hrishikesh, V. Agarwal, S. W. Keckler, , and D. Burger. Static energy reduction techniques for microprocessor caches. *IEEE Transactions on VLSI Systems*, 11(3):303–313, June 2003.

[9] J. L. Hennessy and D. A. Patterson. Computer architecture: A quantitative approach, page 391, 1996.

[10] Intel Pentium4 processor datasheet. http://developer.intel.com/design/pentium4/datashts/298643.htm.

[11] ITRS 2000 update, overall technology roadmap characteristics.

[12] S. Kaxiras, Z. Hu, G. Narlikar, and R. McLellan. Cache-line decay: A mechanism to reduce cache leakage power. *Lecture Notes in Computer Science*, 2001.

[13] N. S. Kim, T. Austin, D. Blaauw, T. Mudge, K. Flautner, J. S. Hu, M. J. Irwin, M. Kandemir, and V. Narayanan. Leakage current: Moore's law meets static power. *IEEE Computer*, 36(12):68–74, December 2003.

[14] S. Manne, A. Klauser, and D. Grunwald. Pipeline gating: Speculation control for energy reduction. In *Proceedings of the 25th Annual Symposium on Computer Architecture (ISCA)*, pages 132–141, 1998.

[15] D. T. Marr, F. Binns, D. L. Hill, G. Hinton, D. A. Koufaty, J. A. Miller, and M. Upton. Hyper-threading technology architecture and microarchitecture: A hypertext history. *Intel Technology Journal*, 6(1), February 2002.

[16] K. Natarajan, H. Hanson, S. W. Keckler, C. R. Moore, and D. Burger. Microprocessor pipeline energy analysis. In *Proceedings of ISLPED 2003*, 2003.

[17] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. In *Proceedings of the 30th International Symposium on Computer Architecture*, pages 2–13, 2003.

[18] S.-H. Yang, M. D. Powell, B. Falsafi, K. Roy, and T. Vijaykumar. An integrated circuit/architecture approach to reducing leakage in deep-submicron high-performance caches. In *International Symposium on High-Performance Computer Architecture*, pages 147–157, 2001.