Retrospective on "Power-Sensitive Multithreaded Architecture"

John S. Seng
Computer Science Department
California Polytechnic State University, San Luis Obispo
San Luis Obispo, CA

Dean M. Tullsen
Dept. of Computer Science and Engineering
University of California, San Diego
La Jolla, CA

George Z.N. Cai
Intel Corporation
Hillsboro, OR

***Abstract*—This article provides a retrospective look at the research that went into the 2000 ICCD paper "Power-Sensitive Multithreaded Architecture". At the time, simultaneous multithreading processors were soon to be commercially available and power consumption was proving to be a challenging design constraint. That research introduced optimizations that increased power and energy efficiency through multithreading, while maintaining performance. This article discusses the optimizations in the paper and discusses how processor designs have changed since its publication.**

***Index Terms*—simultaneous multithreading, hardware multithreading, power consumption, energy efficient processors**

I. Context of the Paper

In 2000, two important technological trends were poised to change the field of microprocessor design and architecture. First, multithreaded processors were soon to appear. Intel introduced their first simultaneous multithreading processor in 2002. Second, power and energy consumption were starting to become a primary design constraint, not just in processors designed for embedded or mobile applications, but also high-performance processors. The typical power of Intel processors had grown from well under 5 Watts a decade before to over 20 Watts [1], and would continue to grow. The paper [2] was the first to look at the confluence of these two trends. In particular, this work demonstrated that the former could provide solutions for the latter.

This paper would utilize and describe one of the early architecture-level research power models. It would demonstrate the inherent energy efficiency of multithreaded designs, despite their potential to exacerbate power. Last, it would introduce three important mechanisms that exploit multithreading to address the power problem. Below, we highlight some specific contributions of the paper.

II. Modeling Power in 2000

When we began this study, architecture level power models were unavailable for academic research -- Wattch [3] would also be introduced to the community in 2000, too late for our use. To pursue this research, we needed an accurate power model that we could connect to our performance simulator. Academic-industry collaboration was critical in this case, allowing us to modify and adapt an Intel power model [4] to integrate into the SMTSIM simulator. The power model utilized an area- and activity-based approach, where an architectural event was measured using activity counters and each event was modeled to have a dynamic power cost. Wattch also implemented a similar methodology. In the research community, the existence of power simulation infrastructure (and Wattch, in particular) would be transforming and enable an explosion in power and energy-related architectural research.

III. Multithreaded Architecture and Energy Efficiency

One of the initial motivations for this work was the observation that incorrect speculation consumed power with no payoff, and multithreaded processors naturally rely less heavily on speculation to get performance than single-threaded processors. To better understand this phenomenon, we utilized a baseline metric, Energy per Useful Instruction (E/UI), which quantified how much energy was expended on average by the instructions that are actually committed (as opposed to those executed via speculation but not necessarily committed). In the worst cases, about 35%-40% of the energy used to run a single-threaded program was being wasted due to uncommitted instructions. We found that the more threads were running, the more efficient the processor was (in terms of E/UI).

Earlier work on multithreaded architecture had demonstrated its natural area efficiency, as the transistor investment required to get one thread running could be amortized to enable a second, third, etc. hardware context with minimal incremental area. What was not obvious at the time was that this was also true for power -- a small incremental cost in power could result in high performance gains, creating significant increase in energy efficiency.

IV. Adressing Power

While energy efficiency has high value, in many cases, power is still a significant constraint -- much more now than

then. We showed that multithreading could also address the power problem. In particular, we introduced three optimizations to control power that were unique to a multithreaded processor.

The first optimization looked at trading off the execution bandwidth of a CPU for the number of threads that CPU could run. We found that multiple threads coupled with a modest pipeline would on average give us better performance than an aggressive pipeline with a single hardware context. The multithreaded architecture, then, could provide both reduced power and higher performance. As a result, this gives us an architecture that no longer trades off power for energy, but improves both.

The second optimization exploited the opportunity to control the power demand dynamically by throttling execution when a thermal sensor exceeded some threshold -- a technique that would later be called "dynamic thermal management" [5]. The most effective mechanism reacts to power above a threshold by stopping fetch for some subset of the threads, thus dynamically reducing the multithreading level of the core temporarily. This mechanism is attractive because it gives us a back-off state that still provides high performance.

Thirdly, we re-examined prior work on fetch policies for simultaneous multithreading. While the prior work focused only on performance optimization, in this case we examined thread selection policies that maximized energy efficiency by minimizing wasted execution bandwidth. We did this by favoring threads that are least likely to introduce wasted speculation -- either counting the number of unresolved branches or unresolved low-confidence branches, and give priority for fetch to those threads with the lowest counts.

V. Investigating Further

This work led us to dive deeper into the concept of wasted energy. We looked further into fully detailing the sources of wasted energy in the paper: "Architecture-Level Power Optimizations: What are the Limits?" [6]. In that work, we were able to quantify wasted energy and categorize the waste into 3 types: unnecessary instructions (instructions that do not change program state), uncommitted instructions, and energy waste due to incorrectly sized processor structures.

VI. Contribution to the Processors of Today

There are a number of microprocessor designs in industry that achieve high instruction throughput, high energy efficiency, and low power by combining multithreading support with simpler pipelines, as suggested by our first optimization. For example, Intel's low-power Atom architecture features hardware multithreading support.

This paper introduced the idea of combining thermal sensors with multithreading mechanisms to gracefully reduce execution demands for power. This idea of adapting runtime behavior based on thermal sensors was explored more fully in a later paper from Princeton [5], without considering multithreading effects. Today, nearly all high performance processors have thermal sensing and some ability to adapt runtime behavior to eliminate thermal emergencies.

References

[1] S. Gunther, F. Binns, D. Carmean, and J. Hall, "Managing the Impact of Increasing Microprocessor Power Consumption", Intel Technology Journal, Vol. 5, No. 1, 2001.

[2] J. Seng, D. Tullsen, and G. Cai, "Power-Sensitive Multithreaded Architecture", International Conference on Computer Design 2000, Sept. 2000, pp. 199-206.

[3] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations", 27th International Symposium on Computer Architecture, June 2000, pp. 83-94.

[4] G. Cai and C.H. Lim, "Architecture Level Power/Performance Optimization and Dynamic Power Estimation", Cool Chips Tutorial at 32nd International Symposium on Microarchitecture, November 1999

[5] D. Brooks and M. Martonosi, "Dynamic Thermal Management for High-Performance Microprocessors", 7th International Symposium on High-Performance Computer Architecture, January 2001.

[6] J. Seng and D. Tullsen, "Architecture-Level Power Optimizations: What are the Limits?", Journal of Instruction Level Parallelism, Vol. 7, January 2005.