Future Processors: Flexible and Modular

Charlie Johnson
IBM
3605 Hwy 52 N
Rochester, MN 55901
1-507-253-2004
charliej@us.ibm.com

Jeff Welser
IBM
650 Harry Road
San Jose, CA 95120
1-408-927-1017
welser@us.ibm.com

ABSTRACT
The ability to continue increasing processor frequency and single thread performance is being severely limited by exponential increases in leakage and active power. To continue to improve system performance, future designs will rely on increasing numbers of smaller, more power efficient cores and special purpose accelerators integrated on a chip. In this paper, we describe how these trends are leading to more modular, SoC-like designs for future processor chips, which can still achieve very high throughput performance while using simplified components and a cost efficient design methodology.

Categories and Subject Descriptors
C.0 [**Computer Systems Organization**]: General – *system architectures.*

General Terms
Performance, Design, Economics, Standardization.

Keywords
Multiprocessor, accelerators, SoC.

1. INTRODUCTION
For the past 30 years, a large portion of the annual improvement in computer system performance has come from transistor scaling. Shrinking the transistor at each generation not only allowed more transistors per chip, but also enabled chip frequency to increase as the transistor switching speed increased, while switching power density remained approximately constant. Unfortunately, scaling also results in exponential increases in leakage power, and by the 90nm node, this leakage power density began approaching the active power density (see Figure 1). Unlike the many previous technological barriers that have challenged the industry's ability to continue classical scaling in the past – and have been overcome – this leakage power increase is fundamental to the physics of an MOS transistor. So while future technology nodes will undoubtedly be able to continue packing more transistors onto a chip each year, the transistor speed increases will be more limited by practical power dissipation. To the microprocessor designer, this means there will continue to be increasing numbers of transistors to utilize for performance and function, but the frequency growth rate will be much more limited.

CODES+ISSS'05, Sept. 19–21, 2005, Jersey City, New Jersey, USA.

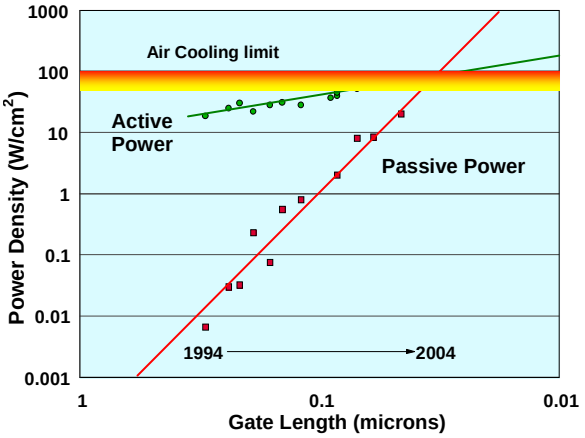

Figure 1. Exponential power density increase as technology scales

In the following sections, we'll discuss the impact these technology trends are having on microprocessor design, why this is leading to more modular SoC designs even for high end processor chips, and how these chips can continue to deliver very high throughput performance even if increases in individual core frequency is limited by power.

2. IMPACT OF POWER ON CORE DESIGN
Microprocessor cores have traditionally focused on single thread performance (i.e. a single application running on a single microprocessor). This form of performance grew at an annual rate of ~60% in the early- to mid-1990's. In the mid-1990's to early 2000's, the annual growth rate slowed to ~40%, and currently it is slowing further to ~20% [1, 2]. Semiconductor frequency has always played a big role in this single thread performance growth rate, typically providing 15% or more gain on an annual basis [1, 2]. However, due to the now significant tradeoffs between device performance and passive power, that source of frequency growth will be slowing dramatically.

What about the other sources of single thread performance growth in microprocessors? There are many design techniques which could theoretically pick up the slack due to the loss of technology frequency growth [3]. However, the frequency and single thread performance "war" between the microprocessor developers over the last 10-15 years has already implemented most of these. Hence, just about every classic single thread performance lever is

plateauing in a fashion not dissimilar to the transistor scaling problem.

The first major design source for achieving additional frequency independent of technology has been pipelining. However, as pipelines have become deeper and the number of logic levels between latches has decreased, the power and device count have increased dramatically. This is due to the increased number of latches in a design and the fact that the clocking rate doubles every time the pipeline depth is doubled. For example, a doubling of the pipeline depth would double the number of latches, which are then all clocked twice as frequently, squaring the active power impact from pipelining alone (not to mention the additional leakage power associated with all of the additional latches and re-powering buffers). Most designs today have settled on about 8-10 logic levels between latches [4], with latches representing about 25% of all of the logic devices (although some designs have gone as far as having only 4-5 logic levels between latches, with latches representing then 50% of the logic devices).

Since power is limiting the ability of executing instructions at a faster rate, a second approach has been to execute more instructions at once [3]. This approach requires the designs to increase both the issue width of the core and the number of execution units available to run the instructions. Again, it turns out the complexity of the control logic is non-linear with the issue width. So a 4-issue implementation is more than two times the complexity of a 2-issue implementation, and 8-issue is more than two times more complex than 4-issue. A few designs actually have attempted to implement up to 8-issue, but due to the complexity increase and to a lack of inherent instruction level parallelism in most programs that compilers or cores could exploit in this manner, the industry has settled on about 4-issue as optimum. Similarly a technique called out-of-order execution (which, within limits, allows instructions to complete in any order in an attempt to get around the lack of instruction level parallelism) also requires substantially more transistors to implement, and yields only about a 20% improvement in single thread performance.

A third design approach would be to try new circuit families or clever circuit techniques to allow the existing designs to operate faster in a given technology. However, most of the high performance circuit techniques require intensive design resources, have less process margin, and/or dissipate inordinately large active or passive power. Hence, good old static circuits dominate more and more of the logic designs, particularly as constantly changing technology definitions make it extremely difficult to get all circuit types to scale uniformly with last minute technology tweaks.

As a result of these design constraints, a typical RISC microprocessor design today issues about 4 instructions per cycle, has 4 or so integer execution units, probably allows out-of-order execution, has a pipeline requiring 8-10 logic levels per cycle, and is implemented with primarily static circuits (with perhaps a few dynamic circuits thrown in for specific macros). Very few designs will go much beyond this for complexity, area and power reasons. The last levers to pull to get more single thread performance out of the core then are process sorting and using elevated supply voltages. Two limitations arise here. First, these are one time levers – once used, additional relative performance is not available from the next design with these techniques. Second, both of these levers increase power in a non-linear fashion, while improving performance in at best a linear fashion. Beyond the core, there has also been a substantial increase in the use of caches to improve performance, but again the single thread performance gains quickly saturate within area limitations.

3. ADVANTAGES OF SoC DESIGN

So power and complexity are causing single thread performance growth to begin to saturate, or at least slowing to a rate of growth substantially lower than it has been historically. There is however another interesting phenomenon resulting from this saturation of single thread performance for high-end microprocessors. All along there have been low-end and embedded processors tracking these micro-architectural advances at a delayed rate of implementation. As the base high-end core concept evolution slows, the embedded cores begin to incorporate proportionally more of these, to the point that they begin to architecturally take on many of the attributes of the high-end cores

This convergence in architecture means that currently there is a pretty dramatic roll-off in the single thread performance gain relative to the development resources, area and power consumed to achieve it. So much so that's its possible to develop cores in a fraction of the area and power, but with most of the performance. These simpler cores not only require fewer development resources (which can be approximated by comparing the number of unique logic transistors between two designs) but are easier to get to market in a more rapid fashion. Time-to-Market (TTM) is a key competitive metric for most microprocessors. The performance of many high-end embedded cores is becoming good enough for many commercial and HPC applications; reference for example the success of IBM's BlueGene, which has risen to the top of the HPC list utilizing low power 32b embedded microprocessors in ordinary foundry technology [5]. These high-end embedded cores tend to be optimized for foundry technologies, which is also where a lot of other high-end library elements, such as I/O interfaces and eDRAM cache macros, already exist. The combination of low power cores and very dense eDRAM caches can make for a powerful performance combination, particularly in a power and cost constrained environment such as blades, where external cache hierarchy is non-existent.

Once one enters the regime of "good enough" single thread performance thru the use of high end embedded microprocessors and eDRAM caches, the ability to utilize SoC design practices presents itself as an obvious methodology to immediately reduce development expenses and TTM simultaneously. A key part of this environment is the use of standardized external interfaces such as DDR2 for memory, PCI for IO and Hyper Transport for high speed interconnect between chips. Historically, the proprietary custom high-end microprocessors mostly shied away from standard interfaces, in favor of eking out a little more performance from a custom design. This only further exacerbated the rapidly growing development bill for what was apparently limited performance advantage. In order to facilitate a more modular design, the next obvious place to pursue standardized interfaces is the on-chip interconnect between the microprocessor cores, the standardized external interfaces, and other SoC library IP elements. This has not been pursued to any significant degree to date for high-performance on-chip bus designs, but could be in the future as SoC design practices become common place for many microprocessor chips.

4. MULTICORE CHIP PERFORMANCE

With the focus moving away from the microprocessor core itself and single thread performance (because it is "good enough" and too expensive to grow substantially), more emphasis will be put on optimizing the number of cores, the cache hierarchy and interconnect on the chip. The performance scaling of large numbers of small, power efficient cores will be the next major performance focal point. Just as the industry chased "frequency" and "single thread performance" in the last ten years, they will likely change to a race of "number of cores on a die" and the efficient performance scaling of those cores. SoC design practices and the development of new chip and system tools will help designers more efficiently explore these new design spaces.

As area and power efficiency become stronger influences on the microprocessor designs, some companies will begin to try even smaller, less complex cores which utilize only subset ISA's for performance targeted at specific applications (these cores are often called accelerators or off-load engines). Basically, these cores eliminate many instructions from a traditional ISA because they are either infrequently used or not used by the application being targeted. Just recently IBM, with partners Sony and Toshiba, unveiled the Cell microprocessor chip [6]. This chip contained a single "general purpose" microprocessor (i.e. one which supported the full ISA) and eight highly area and power efficient accelerators with a unique ISA. It is expected that this type of design will provide >10x performance improvement in the highly specialized application of image rendering used in today's games.

5. SUMMARY

As power constraints are limiting the industry's ability to increase single thread performance, future design focus will move away from resource-intensive unique processor and custom component attributes, and become more focused on creating high-performance system structures that utilize more standard components which are "good enough." The cores and components will be designed with power efficiency as a first order concern, to allow the highest level of integration on a chip, and with standard interfaces, to increase design modularity and to allow very rapid TTM for new chips. The ability to innovate quickly by utilizing this flexibility to introduce new function, integrated at the chip or package level, will be a key differentiator for satisfying customer needs going forward.

One of the biggest challenges that will be faced, however, is how to enable software to efficiently utilize these new massively parallel architectures and special purpose hardware combinations. Tight integration between chip, system, and software design – particularly for the lower levels of the software stack, including O/S, middleware, and compiler technologies – from the beginning of the design cycle will be key to unleashing the potential of these new architectures, without causing untenable churn to application creators. Companies with a focus on co-design of all of these levels will undoubtedly be the most successful in this new environment.

6. ACKNOWLEDGMENTS

Our thanks to Bijan Davari and the Next Generation Computing team at IBM.

7. REFERENCES

[1] SIA, et al. *International Technology Roadmap for Semiconductors*, 2004 update. http://www.itrs.net/Common/2004Update/2004Update.htm

[2] Ekman, M., et al. *An In-Depth Look at Computer Performance Growth*. Technical Report 2004-9, Chalmers University of Technology, Göteborg, Sweden, 2004.

[3] Lee, B. and Brooks, D. Effects of Pipeline Complexity on SMT/CMP Power-Performance Efficiency. In *Proceedings of the Workshop on Complexity-Effective Design. (WCED '05)* (Madison, WI, USA, June 5, 2005). p. 4-12. http://www.csl.cornell.edu/~albonesi/wced05/wced05.pdf

[4] Hrishikesh, M.S., et al. The Optimal Logic Depth per Pipeline Stage is 6 to 8 FO4 Inverter Delays. In *Proceedings of the 29th International Symposium on Computer Architecture. (ISCA '02)* (Anchorage, AK, USA, May 25-29, 2002). IEEE Press, 2002, p. 14-24.

[5] Gara, A. et al. Overview of the BlueGene/L system architecture. *IBM Journal of Res. & Dev., 49*, 2/3 (Mar./May 2005), p. 195-212.

[6] Pham, D. et al. The Design and Implementation of a First-Generation CELL Processor. In *ISSCC Digest of Technical Papers.* (San Francisco, CA, USA, Feb 6-10, 2005) p. 184-5.