Power-Efficient Computer Architectures

Recent Advances

Synthesis Lectures on Computer Architecture

Synthesis Lectures on Computer Architecture publishes 50- to 100-page publications on topics pertaining to the science and art of designing, analyzing, selecting and interconnecting hardware components to create computers that meet functional, performance and cost goals. The scope will largely follow the purview of premier computer architecture conferences, such as ISCA, HPCA, MICRO, and ASPLOS.

The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Second edition
Luiz André Barroso, Jimmy Clidaras, and Urs Hölzle
2013

Shared-Memory Synchronization
Michael L. Scott
2013

Resilient Architecture Design for Voltage Variation
Vijay Janapa Reddi and Meeta Sharma Gupta
2013

Multithreading Architecture
Mario Nemirovsky and Dean M. Tullsen
2013

Performance Analysis and Tuning for General Purpose Graphics Processing Units (GPGPU)
Hyesoon Kim, Richard Vuduc, Sara Baghsorkhi, Jee Choi, and Wen-mei Hwu
2012

Automatic Parallelization: An Overview of Fundamental Compiler Techniques
Samuel P. Midkiff
2012

Phase Change Memory: From Devices to Systems
Moinuddin K. Qureshi, Sudhanva Gurumurthi, and Bipin Rajendran
2011

Multi-Core Cache Hierarchies
Rajeev Balasubramonian, Norman P. Jouppi, and Naveen Muralimanohar
2011

A Primer on Memory Consistency and Cache Coherence
Daniel J. Sorin, Mark D. Hill, and David A. Wood
2011

Dynamic Binary Modification: Tools, Techniques, and Applications
Kim Hazelwood
2011

Quantum Computing for Computer Architects, Second Edition
Tzvetan S. Metodi, Arvin I. Faruque, and Frederic T. Chong
2011

High Performance Datacenter Networks: Architectures, Algorithms, and Opportunities
Dennis Abts and John Kim
2011

Processor Microarchitecture: An Implementation Perspective
Antonio González, Fernando Latorre, and Grigorios Magklis
2010

Transactional Memory, 2nd edition
Tim Harris, James Larus, and Ravi Rajwar
2010

Computer Architecture Performance Evaluation Methods
Lieven Eeckhout
2010

Introduction to Reconfigurable Supercomputing
Marco Lanzagorta, Stephen Bique, and Robert Rosenberg
2009

On-Chip Networks
Natalie Enright Jerger and Li-Shiuan Peh
2009

The Memory System: You Can't Avoid It, You Can't Ignore It, You Can't Fake It
Bruce Jacob
2009

Fault Tolerant Computer Architecture
Daniel J. Sorin
2009

The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines
Luiz André Barroso and Urs Hölzle
2009

Computer Architecture Techniques for Power-Efficiency
Stefanos Kaxiras and Margaret Martonosi
2008

Chip Multiprocessor Architecture: Techniques to Improve Throughput and Latency
Kunle Olukotun, Lance Hammond, and James Laudon
2007

Transactional Memory
James R. Larus and Ravi Rajwar
2006

Quantum Computing for Computer Architects
Tzvetan S. Metodi and Frederic T. Chong
2006

Power-Efficient Computer Architectures: Recent Advances

Magnus Själander, Margaret Martonosi, and Stefanos Kaxiras

www.morganclaypool.com

ISBN: 9781627056458 paperback
ISBN: 9781627056465 ebook

DOI 10.2200/S00611ED1V01Y201411CAC030

A Publication in the Morgan & Claypool Publishers series
SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE

Lecture #30
Series Editor: Margaret Martonosi, *Princeton University*
Founding Editor Emeritus: Mark D. Hill, *University of Wisconsin, Madison*
Series ISSN
Print 1935-3235 Electronic 1935-3243

Power-Efficient Computer Architectures

Recent Advances

Magnus Själander
Uppsala University

Margaret Martonosi
Princeton University

Stefanos Kaxiras
Uppsala University

SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE #30

MORGAN & CLAYPOOL PUBLISHERS

ABSTRACT

As Moore's Law and Dennard scaling trends have slowed, the challenges of building high-performance computer architectures while maintaining acceptable power efficiency levels have heightened. Over the past ten years, architecture techniques for power efficiency have shifted from primarily focusing on module-level efficiencies, toward more holistic design styles based on parallelism and heterogeneity. This work highlights and synthesizes recent techniques and trends in power-efficient computer architecture.

KEYWORDS

power, architecture, parallelism, heterogeneity

Contents

CHAPTER 1

Introduction

1.1 FROM THE BEGINNING...

Managing the power dissipation of current computer systems is a Grand Challenge problem. Power affects computer systems at all scales: from the computational capacity of our large-scale data centers [18], to the processing performance of our high-end servers [25], and the battery life and performance of our mobile devices [45, 152, 207].

To today's computer architects, the emergence of power as a grand challenge [91] may seem like a relatively recent issue, but the reality is that the very earliest computer systems faced vexing power challenges. For example, the ENIAC computer first became operational in 1946, and its initial press release [191] included this alarming text[1] about its kilowatts of power dissipation:

> *The ENIAC consumes 150 kilowatts. This power is supplied by a three-phase, regulated, 240-volt, 60-cycle power line. The power consumption may be broken up as follows; 80 kilowatts for heating the tubes 45 kilowatts for generating d.c. voltages, 20 kilowatts for driving the ventilator blower and 5 kilowatts for the auxiliary card machines.*

Over the decades that have followed, computer systems benefited from technology refinements that improved circuit performance, cost, and power. Gordon Moore's predictions of technology scaling linked integration levels (transistors per chip) to production cost [138]. For many years, these cost-driven integration improvements also translated quite naturally into performance improvements.

Nearly concurrently, Dennard articulated a scaling principle that would lower supply voltages as transistors became smaller [42]. It is Dennard scaling that enables the transistor increases predicted by Moore's Law to be parlayed into performance improvements and power savings.

Despite the benefits of Dennard Scaling, the power dissipation of integrated systems has spiked before. Most notably, some of the high-performance bipolar ECL processors of the late 1980s and early 1990s dissipated over 100 W [100]. While these designs were impressive and offered then unmatched performance, the costs and challenges of designing and packaging such chips [99] are considered to have played a major role in the adoption of CMOS technology for *high-performance* designs. The challenge for us today is that we have reached a similarly difficult operating point regarding CMOS power dissipation, but without any viable alternative technology available to turn to next.

[1] The text's punctuation is as in the original.

Table 1.1: Dennard scaling rules [42]

Device or Circuit Parameter	Scaling Factor
Device dimension T_{ox}, L, W	1/k
Doping concentration Na	k
Voltage V	1/k
Current I	1/k
Capacitance eA/t	1/k
Delay time per circuit VC/I	1/k
Power dissipation per circuit VI	$1/k^2$
Power density VI/A	1

In addition to technology trends, our industry is also driven by application trends that further drive the need for power-efficient computing systems. Compared to 30 years ago, much more of today's computing market (phones, laptops, tablets, games) is at least moderately mobile, and therefore designed with battery life as an important characteristic. In years past, mobile technology would simply "trickle down" from the server world, as old designs are shrunk in new processes. Now the mobile domain is a distinct target with great need for nimble, adaptive power/performance tradeoffs [70]. At the other extreme—the server and enterprise end of the spectrum—power also matters more than ever. Data centers co-locate thousands of high-end servers, and are often limited by their ability to offer sufficient power and cooling to sustain their desired execution throughput [19].

Thus, both technology and application drivers have placed us at a point where power-efficient computation is both vital to future computer systems' viability, and also increasingly difficult to achieve. The following sections elaborate on this.

1.2 THE END OF DENNARD SCALING AND THE SWITCH TO MULTICORES

The power problem as we have faced it over the past decade is largely due to two effects. First, it is primarily a consequence of the end of the Dennard scaling rules (Table 1.1) that parlayed Moore's Law into performance and power benefits for more than three decades. Dennard scaling rests on several key shifts that can be made when transitioning to a smaller feature size. For example, smaller transistors can switch quickly at lower supply voltages, resulting in more power efficient circuits and keeping the power density constant. But supply voltages cannot drop forever. A breakdown of Dennard scaling occurred when voltages dropped low enough to make static power consumption a major issue. Second, even if Dennard scaling had continued on-track, our propensity for faster clock rates and larger die sizes meant that each generation's power dissipation was scaling up faster than Dennard effects were able to hold it in check.

In particular, a key advantage of CMOS technology for many years was its lack of static power dissipation. That is, the complementary p- and n-networks in CMOS gates (theoretically) do not allow any path from supply voltage to ground, consuming power only when switching (dynamic power and some glitch power). Static power consumption was therefore safely ignored at the architectural level. However, when technology scaling broke the 100 nm barrier, transistors showed their analog nature: they are never truly off, and this allows sub-threshold leakage currents to flow. Worse, sub-threshold leakage currents are exponential to threshold voltage reductions. In Dennard scaling, the major mechanism to improve power efficiency is the reduction of the supply voltage, which assumes a reduction of the threshold voltage (since the difference of the two voltages dictates transistor switching speed). The rise of static power brought a complete stop to the power benefits architects took for granted for many technology generations. One-time reductions of static power consumption are possible but the trends remain the same with scaling. For example, current technologies employ multi-gate transistors also known as FinFETs. In these transistors, a fin between the drain and source is "wrapped" by silicon in a non-planar fashion to enable the gate to better encompass the channel, which reduces leakage. As one particular example, Intel switched to 3D or tri-gate transistors in their 22 nm technology [24]. While this change provided a step reduction in leakage going from 32 nm to 22 nm, further reductions will be limited in subsequent scalings.

In the 1980s and early 1990s (the heyday of Moore's Law scaling), architects primarily improved performance by exploiting instruction-level parallelism (ILP)—parallelism found in the dynamic instruction stream during execution of a program. To discover and exploit this parallelism, significant hardware resources were thrown at the problem. Sophisticated techniques such as out-of-order execution, branch prediction and speculative execution, register renaming, memory dependence prediction, among others, were developed for this purpose. These approaches can be highly complex and do not scale well. This results in diminishing performance returns (number of instructions executed in parallel) for increasing hardware investments. Dynamic power scales even worse, deteriorating the power efficiency of such approaches. In fact, as illustrated in Figure 1.1, power dissipation scales as performance raised to the 1.73 power for the typical ILP core: a Pentium 4 is about six times the performance of an i486 at 23 times the power [71]!

The shift to multicore architectures started in 2004 as a reaction to this looming problem of increased power consumption and power density. Effectively, we abandoned frequency scaling (which resulted in significant increases in both dynamic and static power consumption) in favor of laying down more cores on the same chip. This dramatic shift to chip multiprocessors (CMPs) in the past decade is a response to the power wall and the end of Dennard scaling. In particular, Borkar et al. [26] walks through an example for 45 nm technology that is still instructive. For a 45 nm chip with 150M transistors, Figure 1.2 shows a range of possible options for implementing the processor. To abide by the total limit of 150M transistors, one can use more logic transistors (x-axis) in opposition with fewer cache transistors. The resulting power dissipation is shown on the left y-axis, and the resulting cache capacity is shown on the right y-axis. As one increases

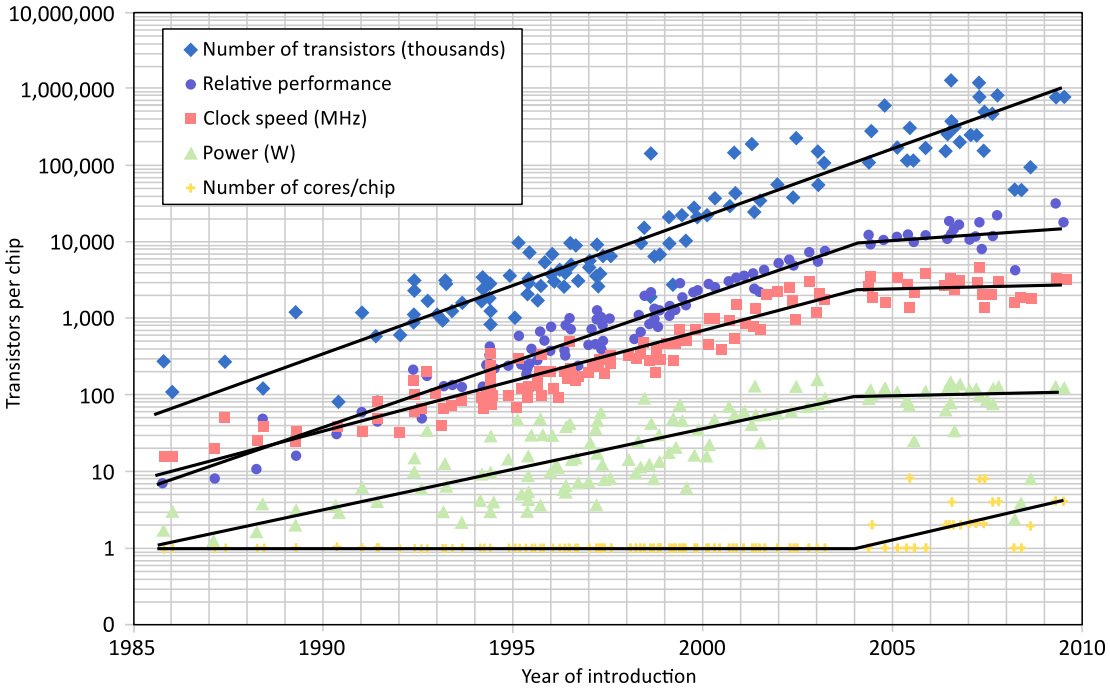


Figure 1.1: Moore's Law and corollaries. Data shows scaling trends, with clear shifts in trend lines at roughly 2004 [37].

the number of transistors devoted to logic, the power dissipation increases (because caches are "cool" from a power standpoint). Pollack's rule [154] argues that microprocessor performance scales roughly as the square root of its complexity, where the logic transistor count is often used as a proxy to quantify complexity.

From these rules of thumb, multiple parallel cores essentially always beat monolithic single cores on power-normalized performance. For example, Figure 1.3 shows three approaches that use parallel cores to enhance throughput while maintaining the same power envelope [26]. Case A (far left of Figure 1.2) harnesses 6-way parallelism at a fairly coarse-grain, and is out-performed by Case B (far right), which is more aggressively parallel, when enough thread/task level parallelism exist in the workload. Case C represents heterogeneous parallelism, in which two large cores are mixed with several small ones, to good effect. An even more heterogeneous approach would be to include some specialized accelerators, which use very few transistors or chip area, but have large performance and power benefits when applicable.

Overall, these examples and rules of thumb begin to explain the direction that industry has taken: a quick and aggressive adoption of medium-scale, on-chip parallelism. Parallelism helps with the impending power wall, by offering a path to high performance that does not rely on

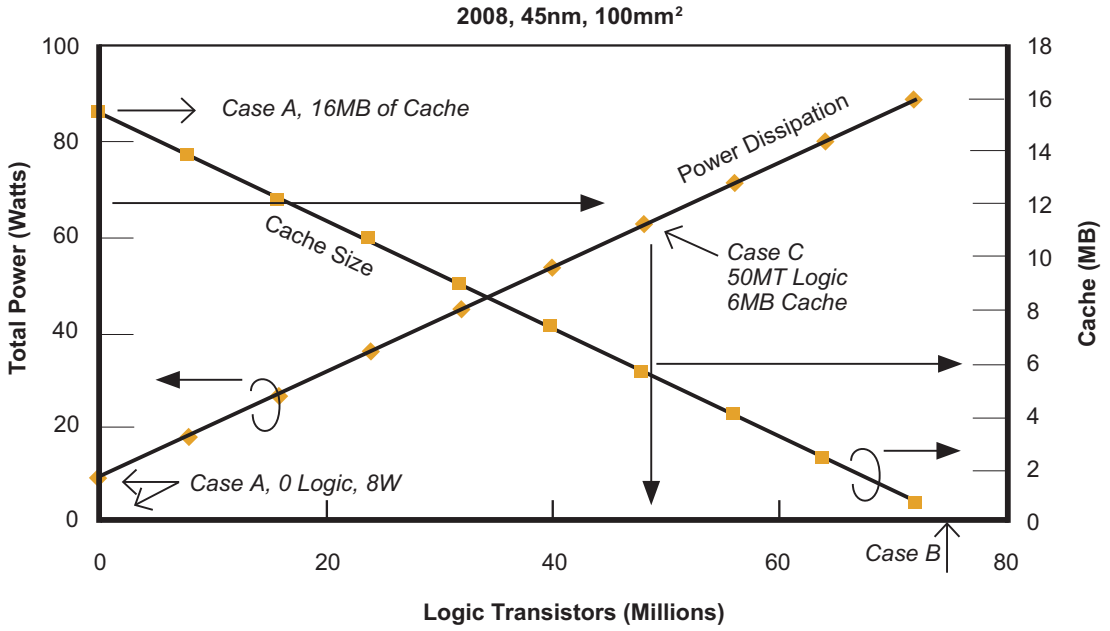


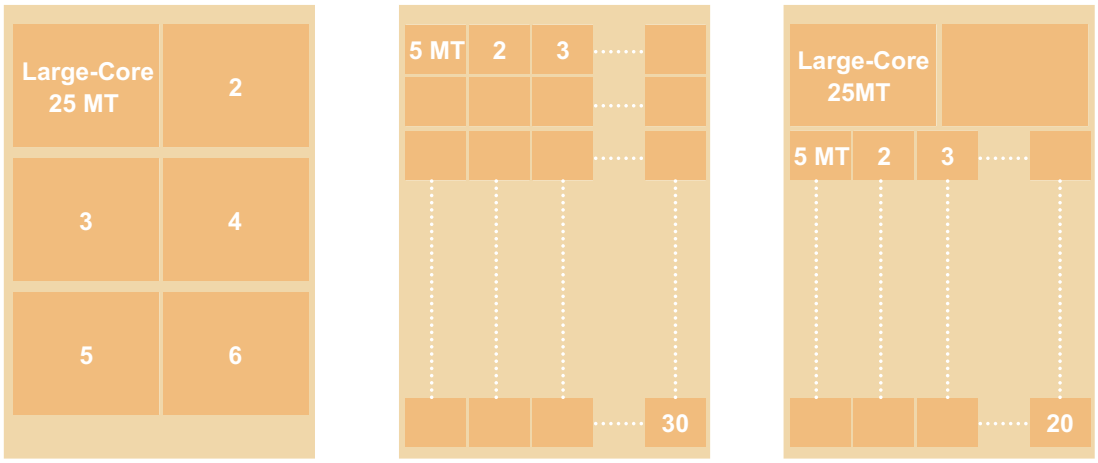
Figure 1.2: A range of implementation options trading off processor area devoted to cache, and resulting power tradeoffs [26].

high clock rates and high supply voltages. Parallelism—particularly heterogeneous parallelism—also helps with the so-called "utilization wall" [186] and the "Dark Silicon" problem [54], as discussed next.

1.3 DARK SILICON, THE UTILIZATION WALL, AND THE RISE OF THE HETEROGENEOUS PARALLELISM

Our inability to scale a single core to further exploit ILP in a power efficient manner turned computer architecture toward exploring alternative kinds of parallelism (task/thread parallelism, data parallelism). Multicore and manycore architectures are designed for explicit parallelism, and recalling Figures 1.2 and 1.3, they offer greater performance-per-watt than large monolithic approaches. Unfortunately, even homogeneous CMPs will not be sufficient to solve the power problem for more than a few more generations [137]. This road is also faced with the same problems as with the single core architecture: we are unable to efficiently extract sufficient speedup from parallel programs (Amdahl's Law [7]).

Furthermore, some postulate a near future in which the number of *dynamically active* transistors on a die may be greatly constrained, forming the "utilization wall" [186]. The concept of the utilization wall is that power envelopes may lead to scenarios in which few (perhaps 20% or less)



Large-Core Homogeneous	
Large-core throughput	1
Small-core throughput	
Total throughput	6

(a)

Small-Core Homogeneous	
Large-core throughput	
Small-core throughput	Pollack's Rule $(5/25)^{0.5}$=0.45
Total throughput	13

(b)

Small-Core Homogeneous	
Large-core throughput	1
Small-core throughput	Pollack's Rule $(5/25)^{0.5}$=0.45
Total throughput	11

(c)

Figure 1.3: Enhancing throughput while maintaining power envelope [26].

of a chip's transistors can be "on" at a time. The argument for this possible future is exemplified in Figure 1.4. If transistor density increases in line with Moore's Law, a 45 nm chip will shrink to one-quarter the size at 22 nm in 2014, and one-sixteenth at 11 nm in 2020. Using the ITRS roadmap [91] for scaling, the smaller chips would be more efficient, drawing the same power at 22 nm even though the peak frequency increases by a factor of 1.6, and 40% less at 11 nm with 2.4 peak clock speed. But, if we maintain the same chip area, we can pack four times the number of transistors at 22 nm and 16 times at 11 nm. For the same initial power budget this means that only 25% of the transistors can be powered-up in 22 nm, and 10% in 11 nm. These results are also supported in recent academic studies [54].

The answer to the challenge of the utilization wall is the rise of heterogeneous architectures where some general-purpose cores are augmented by other cores of different microarchitectures or even specialized accelerators that offer outstanding performance-per-watt by being very lean hardware designs for a particular computational purpose. The approach of heterogeneous paral-

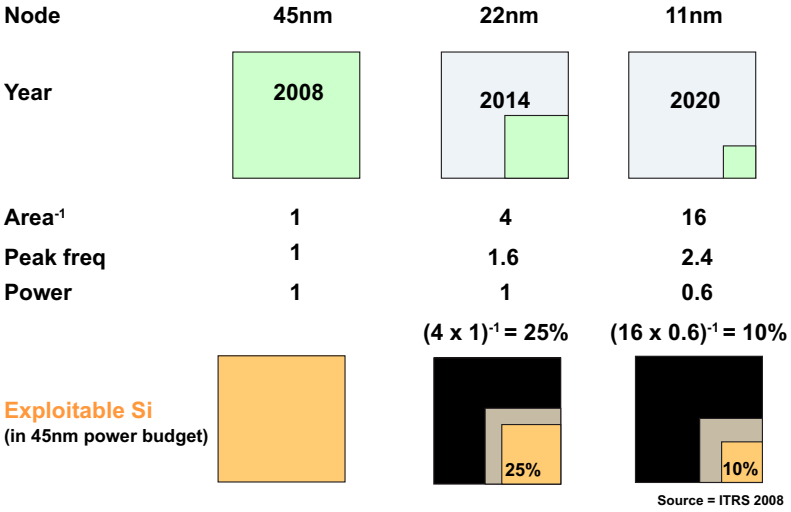


Figure 1.4: A depiction of Dark Silicon trends as seen by ARM [2, 143].

lelism with specialized accelerators is well-suited for "Dark Silicon" scenarios. A large number of accelerators can be built on the same chip to be woken up only when needed. These heterogeneous architectures are fast becoming the dominant paradigm after multicore.

In fact, we do not need to speculate about future heterogeneity, as heterogeneous parallel computing is here today. If we examine the product lines from the major chip manufacturers we see that they now have separate multicore x86 architectures targeted at high performance (2–12 cores, 100W, 100 GFLOPs) and low power (1–2 cores, 10W, 10 GFLOPs), and are integrating data-parallel graphics cores onto their CPU devices with distinct programming and memory models [28]. In the embedded world, there are a range of cores at different performance/efficiency points (1–8 cores, 2W, 10 GMIPS) with a range of programmable graphics cores [133]. NVIDIA, Samsung, and Qualcomm all sell heterogeneous ARM/GPU processors with many fixed-function accelerator blocks for the smart phone market [12, 144, 159], and there are multiple start-ups with 64–100 core devices [183] for networking and telecom. This present-day processor heterogeneity forces system and software designers to address the difficult optimization challenge of choosing the right processor (both at design time and runtime) for their product's power and performance requirements.

Beyond simply considering heterogeneity in the types of instruction-programmable cores on-chip, the field is also increasingly considering approaches involving specialized accelerators that may not be instruction-programmable, and that are tuned to particular application kernels of interest. Specialized accelerators are a particularly natural response to the Dark Silicon scenario in which we may have many more transistors than what we can power up at once. With these "dormant" transistors we could build a plethora of specialized accelerators that cost little either

in terms of "active" area or power when not in use. The expectations of generality—all transistors must be useful to all applications—shift considerably in a Dark Silicon world, and what once might have been viewed as "niche" accelerators become a viable method for achieving performance goals under dramatic power constraints.

1.4 OTHER ISSUES AND FUTURE DIRECTIONS

Overall, computer systems have reached an intriguing inflection point. For architects, power has been *a* fundamental design constraint for well over a decade now, with the initial reaction being fairly localized, per-module efforts to improve power efficiency. These efforts have been the equivalent of turning lights off in unused rooms of one's house—very sensible, but insufficient in leverage to dramatically change the overall power-performance design landscape. The second wave in power-aware computing has been the recent and seismic shift toward on-chip parallelism.

Software and Programmability Issues: In many ways, the hardware industry's shift toward parallelism has occurred much faster than the abilities of the software and systems designers to react. We know how to build CMPs, and we must build them to keep Moore's Law rolling along. But we do not yet know how to program them efficiently—both in terms of software development time and in terms of getting the best power-performance outcomes from them. Furthermore, the shift toward on-chip accelerators offers even greater programmability challenges. Finally, there are a host of programmability concerns that emanate from the basic goal of elevating power to a first-class design constraint alongside performance. For example, from a power perspective, information on the relative criticality of different communication or computation operations may be very useful, but current programming models offer few abstractions or constructs to help programmers manage this.

Reliability Tradeoffs: Until now, power-performance tradeoffs have been viewed by architects as a two-dimensional optimization landscape. There is emerging research, however, on the possibilities of *three-dimensional* optimization scenarios in which power, performance, and *reliability* are traded off against each other. Such tradeoffs are already frequently considered at the device and circuit level, but in ways that enable the architecture and software levels to be shielded from their effects; abstraction layers give the impression of perfect reliability even when device or circuit tricks are being employed [52].

Intuitively, there seem to be rich opportunities for raising the abstraction layer at which reliability, energy, and performance are traded off, in order to enable architects to exploit them as well. For example, operating with smaller supply voltage noise margins (by lowering supply voltage) may offer high leverage on power savings, at the expense of possible calculation or storage errors. Likewise, reducing or eliminating parity/checksum protection on memory or interconnect also seems to offer some intuitive power/reliability tradeoff possibility. The key research questions in this space, however, focus on whether the power/performance benefits achievable through some approaches are large enough to be appealing given the serious impact of relaxing reliability guarantees to software.

Beyond the Processor Core: Much of the "first wave" of power optimizations focused on the CPU itself, because the most serious thermal and power density concerns were experienced there. And even more specifically, most optimizations were focused on the CPU's processor cores and cache memories. As we look, however, to future power issues and ideas, there is a growing need to look beyond the processor core. Data communications and on-chip interconnect will play an increasingly important role in power dissipation, especially since the adoption of parallelism has led to much higher levels of data motion and inter-processor communication in many cases. One also needs to consider the energy issues related to the memory hierarchy as well. Chapter 4 covers these topics in this book, but considerable future work in this area is likely to be forthcoming.

1.5 ABOUT THE BOOK

To conclude this introduction, we include here some further explanations about the book that may be helpful to readers in finding relevant material and in comparing with contents from a prior Synthesis Lecture by Kaxiras and Martonosi [103].

1.5.1 DIFFERENCES FROM THE PRIOR SYNTHESIS LECTURE [103]

We view the two books as largely complementary. The first book offered extensive details on the sorts of local, per-module power optimizations that comprised the industry's "first wave" response to the power challenge. In this current book, we take a more holistic view. As a result, both the structure and content of the book have changed dramatically.

There are three core chapters, which synthesize highlights of power-efficient computer architecture techniques. Chapter 2 covers voltage and frequency scaling issues, with a particular emphasis on trends and techniques that have emerged in the years since the first edition of the book. Chapter 3 covers techniques related to specialization and heterogeneity that have emerged with greater prominence in the five years since Dark Silicon began to emerge. Chapter 4 covers the power implications of data motion and storage, again with a particular emphasis on more recent techniques and trends. Finally, Chapter 5 concludes the book.

We note that while the power dissipation of main memory has emerged as an important problem, we feel that this topic is too broad to be covered well as part of this book. Thus, this book does not cover main memory issues in earnest, and we hope that another synthesis lecture will take on this topic in detail.

Finally, a note on power modeling approaches. These were covered in the first book [103]. While new tools and modeling environments have been created in the years since then (e.g., [76, 122, 127]), these tools employ fairly similar basic philosophies and approaches as previous generations of tools. Thus due to space and scope constraints we have chosen not to cover them further here.

1.5.2 TARGET AUDIENCE

This book is written for researchers who have taken a basic course in computer architecture, and are interested in becoming somewhat fluent in the power implications of architectural choices. We envision it being particularly useful for a new graduate student who may be familiar with the basics of computer design and architectural simulations, but perhaps has been less exposed to power issues. In addition, systems researchers from related fields (e.g., operating systems, compilers, parallel programming, and others) may find the book useful for understanding some of the architectural viewpoints and issues interposed between the technology challenges emerging "from below" and the applications trends "from above."

Authors' Biographies

MAGNUS SJÄLANDER

Magnus Själander is a Research Associate at Uppsala University. He received both his Ph.D. degree (2008) and Lic.Eng. degree (2006) in Computer Engineering from Chalmers University of Technology, Sweden. He has been a visiting researcher at NXP Semiconductors, worked at Aeroflex Gaisler, been a post-doctoral researcher at Chalmers University of Technology, and a research scientist at Florida State University. Själander's research interests include energy-efficient computing, high-performance and low-power digital circuits, micro-architecture and memory-system design, and hardware-software interaction.

MARGARET MARTONOSI

Margaret Martonosi is the Hugh Trumbull Adams '35 Professor of Computer Science at Princeton University, where she has been on the faculty since 1994. She also holds an affiliated faculty appointment in Princeton's Electrical Engineering Department. Martonosi's research interests are in computer architecture and mobile computing, with particular focus on power-efficient systems. Her work has included the development of the Wattch power modeling tool and the Princeton ZebraNet mobile sensor network project for the design and real-world deployment of zebra tracking collars in Kenya. Her current research focuses on hardware-software interface approaches to manage heterogeneous parallelism and power-performance tradeoffs in systems ranging from smartphones to chip multiprocessors to large-scale data centers. Martonosi is a Fellow of both IEEE and ACM. She was the 2013 recipient of the Anita Borg Institute Technical Leadership Award. She has also received the 2013 NCWIT Undergraduate Research Mentoring Award and the 2010 Princeton University Graduate Mentoring Award.

STEFANOS KAXIRAS

Stefanos Kaxiras is a full professor at Uppsala University, Sweden. He holds a Ph.D. degree in Computer Science from the University of Wisconsin. In 1998, he joined the Computing Sciences Center at Bell Labs (Lucent) and later Agere Systems. In 2003 he joined the faculty of the ECE Department of the University of Patras, Greece and in 2010 became a full professor at Uppsala University, Sweden. Kaxiras' research interests are in the areas of memory systems, and multiprocessor/multicore systems, with a focus on power efficiency. He has co-authored more than 100 research papers and 13 US patents, participated in five major European research projects, and

currently receives funding from Sweden's business incubator and innovation agency VINNOVA. Kaxiras is a Distinguished ACM Scientist and IEEE member.