



EXPLAINING PARALLEL ARCHITECTURE DESIGN

By Natalie Enright Jerger

Michel Dubois, Murali Annavaram, and Per Stenström, *Parallel Computer Organization and Design*, Cambridge Univ. Press, 2012, ISBN: 978-0521886758, 560 pp.



Parallel Computer Organization and Design provides a comprehensive look at advanced computer architecture topics

from the perspective of parallel computer organization. Topics in parallelism broadly span instruction-level parallelism through thread-level parallelism.

With the rise of multicore processors in the last decade, it has become imperative that parallelism be integrated into today's computer science and engineering curricula. When I first heard that this book was in preparation, I was excited. The book promised to address a critical need in the teaching of computer architecture: an up-to-date, comprehensive, and accessible textbook and reference on parallel architectures. I had been looking for a book that would cover a range of topics from basic in-order processors and caches to out-of-order processors and multiprocessor systems.

When I was asked to review this book, I was anxious to see if the book effectively addresses key concepts in parallel architecture while being written at a level appropriate for both undergraduate and graduate students. Upon reading this book, the answer to both of these questions is largely "yes." In terms of content and approach, the book is spot-on in addressing the pressing need to teach students about computer architecture from the

vantage point of now-ubiquitous parallel architectures. In terms of the material's instructive level, the text is quite approachable. However, there might be some gaps for undergraduates that would need to be supplemented through lectures and additional material. For example, my students come in without exposure to the classical five-stage, in-order pipeline. The book covers this, but more details might be needed to bridge the gap from computer organization to computer architecture. For graduate students, the text can easily be supplemented with research papers relevant to each topic, but it doesn't need to be—the text stands on its own quite nicely.

The book reads well and is quite clean. It's largely free of clutter, focusing on key concepts from a qualitative and high-level perspective. Moreover, it doesn't overburden the reader with quantitative results or put undue focus on specific product implementations.

Content

The co-authors of this book are leading experts in the field of computer architecture. Their different areas of expertise are evident in the thoughtful and thorough treatment of a broad range of topics. Topics are self-contained and laid out in an appealing manner from a pedagogical perspective.

This book covers the following topics:

- an introduction to computer architecture;
- the impact of technology on architecture;
- processor microarchitecture, starting from the classical five-stage, in-order pipeline through dynamically scheduled pipelines, very long instruction word (VLIW) architectures, and vector machines;
- memory hierarchies, including caches and virtual memory;
- multiprocessor systems, including coherence, interconnection networks, synchronization, and memory consistency;
- chip multiprocessors and multithreading; and
- methods for quantitative evaluation.

In terms of content, all the expected topics are there. In addition, there were a few chapters that stood out as possibly unique in their coverage; these provide a nice supplement to conventional topics. In particular, chapters 2 and 9 provide content that often isn't covered in any depth in a computer architecture textbook. Chapter 2 provides a technology review and overview (on topics such as transistor basics, power consumption, and reliability) geared towards students who might need to brush up on this material or might have

less exposure due to a computer science rather than engineering background. This chapter also provides nice coverage of issues related to power and reliability that are often overlooked (such as low-power design techniques and the architectural vulnerability factor). Subsequent chapters don't depend on this material, yet its inclusion is a nice reference for students.

Chapter 9 provides an overview of research methodologies in computer architecture and a thorough discussion of various simulation platforms. This chapter also provides some simulation exercises for students. Other computer architecture books that I'm aware of have similarly provided simulator assignments, but are missing such a nice taxonomy and discussion of simulators spanning performance, power, and thermal considerations.

Each chapter provides a nice set of exercises to reinforce the key material. The authors have presented thoughtful and thought-provoking exercises. Creating fresh questions can always be a challenge, but I found these to be quite well done (especially the questions on processor architecture and multiprocessor systems).

Strengths and Weaknesses

One aspect of the book's organization that I really liked was how self-contained each chapter was. The authors effectively encapsulated all the key concepts within a single chapter. This comes at the cost of being repetitive in subsequent chapters, but this material can easily be skipped or can provide students with often-needed reinforcement. Much of the material is independent from previous chapters, and

when prior material is needed, the authors do a nice job of briefly summarizing the key concepts needed in a particular chapter. This lets an instructor pick and choose chapters that suit them or allows material to be taught concurrently in multiple courses. For example, chapter 5 introduces multiprocessor systems and covers basic cache-coherence protocols. Chapter 7 presents an in-depth discussion of memory-ordering considerations, and briefly introduces the reader to cache coherence to allow this chapter to stand on its own.

Criticisms of the text are mostly minor. Considering the book as more of a reference text than a textbook, I would have appreciated references to provide the reader with pointers to primary source material on the topics covered. The density and level of depth also varied in the treatment of different subjects. For example, the treatment of issues related to memory ordering and consistency is thorough, while being written in a way that's both a handy reference and easily accessible to those unfamiliar with these topics. Yet some of the treatment of dynamically scheduled pipelines seemed brief; coverage of the specifics of Tomasulo's algorithm is one example. Finally, I felt that each chapter (and the book as a whole) ends rather abruptly. I would have liked some "putting it all together" type of material to help summarize and refocus students on key concepts.

The book's thorough approach to teaching parallel architecture has been sorely needed, and this textbook will become an indispensable

resource to students and educators alike. The expertise of the authors is evident. I look forward to using this book in my next offering of computer architecture.



Natalie Enright Jerger is an assistant professor in the Edward S. Rogers Department of Electrical and Computer Engineering at the University of Toronto. Her research focuses on many-core computer architectures and interconnection networks, and she currently teaches computer organization and computer architecture courses. Enright Jerger has a PhD in electrical engineering from the University of Wisconsin-Madison. Contact her at enright@eecg.toronto.edu.

cn Selected articles and columns from IEEE Computer Society publications are also available for free at <http://ComputingNow.computer.org>.



**Richard E. Merwin
Student Leadership
Scholarship**

IEEE Computer Society is offering \$40,000 in student scholarships, from \$1,000 and up, to recognize and reward active student volunteer leaders who show promise in their academic and professional efforts.

Graduate students and undergraduate students in their final two years, enrolled in a program in electrical or computer engineering, computer science, information technology, or a well-defined computer-related field, are eligible. IEEE Computer Society student membership is required.

Apply now! Application deadline is 30 September 2013. For more information, go to www.computer.org/scholarships, or email patricia.edwards@computer.org.

To join IEEE Computer Society, visit www.computer.org/membership.


