

General-Purpose Graphics Processor Architectures

Synthesis Lectures on Computer Architecture

Editor

Margaret Martonosi, *Princeton University*

Founding Editor Emeritus

Mark D. Hill, *University of Wisconsin, Madison*

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.

General-Purpose Graphics Processor Architectures

Tor M. Aamodt, Wilson Wai Lun Fung, and Timothy G. Rogers
2018

Compiling Algorithms for Heterogeneous Systems

Steven Bell, Jing Pu, James Hegarty, and Mark Horowitz
2018

Architectural and Operating System Support for Virtual Memory

Abhishek Bhattacharjee and Daniel Lustig
2017

Deep Learning for Computer Architects

Brandon Reagen, Robert Adolf, Paul Whatmough, Gu-Yeon Wei, and David Brooks
2017

On-Chip Networks, Second Edition

Natalie Enright Jerger, Tushar Krishna, and Li-Shiuan Peh
2017

Space-Time Computing with Temporal Neural Networks

James E. Smith
2017

Hardware and Software Support for Virtualization

Edouard Bugnion, Jason Nieh, and Dan Tsafir
2017

Datacenter Design and Management: A Computer Architect's Perspective

Benjamin C. Lee
2016

A Primer on Compression in the Memory Hierarchy

Somayeh Sardashti, Angelos Arelakis, Per Stenström, and David A. Wood
2015

Research Infrastructures for Hardware Accelerators

Yakun Sophia Shao and David Brooks
2015

Analyzing Analytics

Rajesh Bordawekar, Bob Blainey, and Ruchir Puri
2015

Customizable Computing

Yu-Ting Chen, Jason Cong, Michael Gill, Glenn Reinman, and Bingjun Xiao
2015

Die-stacking Architecture

Yuan Xie and Jishen Zhao
2015

Single-Instruction Multiple-Data Execution

Christopher J. Hughes
2015

Power-Efficient Computer Architectures: Recent Advances

Magnus Själander, Margaret Martonosi, and Stefanos Kaxiras
2014

FPGA-Accelerated Simulation of Computer Systems

Hari Angepat, Derek Chiou, Eric S. Chung, and James C. Hoe
2014

A Primer on Hardware Prefetching

Babak Falsafi and Thomas F. Wenisch
2014

On-Chip Photonic Interconnects: A Computer Architect's Perspective

Christopher J. Nitta, Matthew K. Farrens, and Venkatesh Akella

2013

Optimization and Mathematical Modeling in Computer Architecture

Tony Nowatzki, Michael Ferris, Karthikeyan Sankaralingam, Cristian Estan, Nilay Vaish, and David Wood

2013

Security Basics for Computer Architects

Ruby B. Lee

2013

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 Bagsorkhi, 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, Second 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

© Springer Nature Switzerland AG 2022

Reprint of original edition ©Morgan & Claypool 2018

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopy, recording, or any other except for brief quotations in printed reviews, without the prior permission of the publisher.

General-Purpose Graphics Processor Architectures

Tor M. Aamodt, Wilson Wai Lun Fung, and Timothy G. Rogers

ISBN: 978-3-031-00631-9 paperback

ISBN: 978-3-031-01759-9 ebook

ISBN: 978-3-031-00056-0 hardcover

DOI 10.1007/978-3-031-01759-9

A Publication in the Springer series

SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE

Lecture #44

Series Editor: Margaret Martonosi, *Princeton University*

Founding Editor Emeritus: Mark D. Hill, *University of Wisconsin, Madison*

Series ISSN

Print 1935-3235 Electronic 1935-3243

General-Purpose Graphics Processor Architectures

Tor M. Aamodt
University of British Columbia

Wilson Wai Lun Fung
Samsung Electronics

Timothy G. Rogers
Purdue University

SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE #44

ABSTRACT

Originally developed to support video games, graphics processor units (GPUs) are now increasingly used for general-purpose (non-graphics) applications ranging from machine learning to mining of cryptographic currencies. GPUs can achieve improved performance and efficiency versus central processing units (CPUs) by dedicating a larger fraction of hardware resources to computation. In addition, their general-purpose programmability makes contemporary GPUs appealing to software developers in comparison to domain-specific accelerators. This book provides an introduction to those interested in studying the architecture of GPUs that support general-purpose computing. It collects together information currently only found among a wide range of disparate sources. The authors led development of the GPGPU-Sim simulator widely used in academic research on GPU architectures.

The first chapter of this book describes the basic hardware structure of GPUs and provides a brief overview of their history. Chapter 2 provides a summary of GPU programming models relevant to the rest of the book. Chapter 3 explores the architecture of GPU compute cores. Chapter 4 explores the architecture of the GPU memory system. After describing the architecture of existing systems, Chapters 3 and 4 provide an overview of related research. Chapter 5 summarizes cross-cutting research impacting both the compute core and memory system.

This book should provide a valuable resource for those wishing to understand the architecture of graphics processor units (GPUs) used for acceleration of general-purpose applications and to those who want to obtain an introduction to the rapidly growing body of research exploring how to improve the architecture of these GPUs.

KEYWORDS

GPGPU, computer architecture

Contents

Preface	xv
Acknowledgments	xvii
1 Introduction	1
1.1 The Landscape of Computation Accelerators	1
1.2 GPU Hardware Basics	2
1.3 A Brief History of GPUs	6
1.4 Book Outline	7
2 Programming Model	9
2.1 Execution Model	9
2.2 GPU Instruction Set Architectures	14
2.2.1 NVIDIA GPU Instruction Set Architectures	14
2.2.2 AMD Graphics Core Next Instruction Set Architecture	17
3 The SIMT Core: Instruction and Register Data Flow	21
3.1 One-Loop Approximation	22
3.1.1 SIMT Execution Masking	23
3.1.2 SIMT Deadlock and Stackless SIMT Architectures	26
3.1.3 Warp Scheduling	31
3.2 Two-Loop Approximation	33
3.3 Three-Loop Approximation	35
3.3.1 Operand Collector	38
3.3.2 Instruction Replay: Handling Structural Hazards	40
3.4 Research Directions on Branch Divergence	41
3.4.1 Warp Compaction	42
3.4.2 Intra-Warp Divergent Path Management	47
3.4.3 Adding MIMD Capability	52
3.4.4 Complexity-Effective Divergence Management	54
3.5 Research Directions on Scalarization and Affine Execution	57
3.5.1 Detection of Uniform or Affine Variables	57

3.5.2	Exploiting Uniform or Affine Variables in GPU	60
3.6	Research Directions on Register File Architecture	62
3.6.1	Hierarchical Register File	63
3.6.2	Drowsy State Register File	64
3.6.3	Register File Virtualization	64
3.6.4	Partitioned Register File	65
3.6.5	RegLess	65
4	Memory System	67
4.1	First-Level Memory Structures	67
4.1.1	Scratchpad Memory and L1 Data Cache	68
4.1.2	L1 Texture Cache	72
4.1.3	Unified Texture and Data Cache	73
4.2	On-Chip Interconnection Network	75
4.3	Memory Partition Unit	75
4.3.1	L2 Cache	75
4.3.2	Atomic Operations	76
4.3.3	Memory Access Scheduler	76
4.4	Research Directions for GPU Memory Systems	77
4.4.1	Memory Access Scheduling and Interconnection Network Design	77
4.4.2	Caching Effectiveness	78
4.4.3	Memory Request Prioritization and Cache Bypassing	78
4.4.4	Exploiting Inter-Warp Heterogeneity	80
4.4.5	Coordinated Cache Bypassing	81
4.4.6	Adaptive Cache Management	81
4.4.7	Cache Prioritization	82
4.4.8	Virtual Memory Page Placement	82
4.4.9	Data Placement	83
4.4.10	Multi-Chip-Module GPUs	84
5	Crosscutting Research on GPU Computing Architectures	85
5.1	Thread Scheduling	85
5.1.1	Research on Assignment of Threadblocks to Cores	86
5.1.2	Research on Cycle-by-Cycle Scheduling Decisions	88
5.1.3	Research on Scheduling Multiple Kernels	92
5.1.4	Fine-Grain Synchronization Aware Scheduling	93
5.2	Alternative Ways of Expressing Parallelism	93

5.3	Support for Transactional Memory	96
5.3.1	Kilo TM	96
5.3.2	Warp TM and Temporal Conflict Detection	98
5.4	Heterogeneous Systems	99
	Bibliography	103
	Authors' Biographies	121

Preface

This book is intended for those wishing to understand the architecture of graphics processor units (GPUs) and to obtain an introduction to the growing body of research exploring how to improve their design. It is assumed readers have a familiarity with computer architecture concepts such as pipelining and caches and are interested in undertaking research and/or development related to the architecture of GPUs. Such work tends to focus on trade-offs between different designs, and thus this book is written with a view to providing insights into such trade-offs so that the reader can avoid having to learn by trial and error what is already known to experienced designers.

To help achieve this, the book collects together into one resource many relevant bits of information currently found among a wide range of disparate sources such as patents, product documents, and research papers. It is our hope this will help reduce the time it takes for a student or practitioner just starting to do their own research to become productive.

While this book covers aspects of current GPU designs, it also attempts to “synthesize” published research. This is partly due to necessity, as very little has been said by vendors on the microarchitecture of specific GPU products. In describing a “baseline” GPGPU architecture, this book relies both upon published product descriptions (journal papers, whitepapers, manuals) and, in some cases, descriptions in patents. The details found in patents may differ substantially from the microarchitecture of actual products. In some cases, microbenchmark studies have clarified for researchers some details, but in others our baseline represents our “best guess” based upon publicly available information. Nonetheless, we believe this will be helpful as our focus is understanding architecture trade-offs that have already been studied or might be interesting to explore in future research.

Several portions of this book focus on summarizing the many recent research papers on the topic of improving GPU architectures. As this topic has grown significantly in popularity in recent years, there is too much to cover in this book. As such, we have had to make difficult choices about what to cover and what to leave out.

Tor M. Aamodt, Wilson Wai Lun Fung, and Timothy G. Rogers
April 2018

Acknowledgments

We would like to thank our families for their support while writing this book. Moreover, we thank our publisher, Michael Morgan and editor, Margaret Martonosi, for the extreme patience they have shown while this book came together. We also thank Carole-Jean Wu, Andreas Moshovos, Yash Ukidave, Aamir Raihan, and Amruth Sandhupatla for providing detailed feedback on early drafts of this book. Finally, we thank Mark Hill for sharing his thoughts on strategies for writing Synthesis Lectures and specific suggestions for this book.

Tor M. Aamodt, Wilson Wai Lun Fung, and Timothy G. Rogers
April 2018