

Processor Microarchitecture

An Implementation Perspective

Synthesis Lectures on Computer Architecture

Editor

Mark D. Hill, *University of Wisconsin*

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.

Processor Microarchitecture: An Implementation Perspective

Antonio González, Fernando Latorre, and Grigorios Magklis

2011

Transactional Memory, 2nd edition

Tim Harris, James Larus, and Ravi Rajwar

2010

Computer Architecture Performance Evaluation Models

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 2011

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.

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

ISBN: 978-3-031-00601-2 paperback

ISBN: 978-3-031-01729-2 ebook

DOI 10.1007/978-3-031-01729-2

A Publication in the Springer series

SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE #12

Lecture #12

Series Editor: Mark D. Hill, University of Wisconsin

Series ISSN

ISSN 1935-3235 print

ISSN 1935-3243 electronic

Processor Microarchitecture

An Implementation Perspective

Antonio González

Intel and Universitat Politècnica de Catalunya

Fernando Latorre and Grigorios Magklis

Intel

SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE #12

ABSTRACT

This lecture presents a study of the microarchitecture of contemporary microprocessors. The focus is on implementation aspects, with discussions on their implications in terms of performance, power, and cost of state-of-the-art designs. The lecture starts with an overview of the different types of microprocessors and a review of the microarchitecture of cache memories. Then, it describes the implementation of the fetch unit, where special emphasis is made on the required support for branch prediction. The next section is devoted to instruction decode with special focus on the particular support to decoding x86 instructions. The next chapter presents the allocation stage and pays special attention to the implementation of register renaming. Afterward, the issue stage is studied. Here, the logic to implement out-of-order issue for both memory and non-memory instructions is thoroughly described. The following chapter focuses on the instruction execution and describes the different functional units that can be found in contemporary microprocessors, as well as the implementation of the bypass network, which has an important impact on the performance. Finally, the lecture concludes with the commit stage, where it describes how the architectural state is updated and recovered in case of exceptions or misspeculations.

This lecture is intended for an advanced course on computer architecture, suitable for graduate students or senior undergrads who want to specialize in the area of computer architecture. It is also intended for practitioners in the industry in the area of microprocessor design. The book assumes that the reader is familiar with the main concepts regarding pipelining, out-of-order execution, cache memories, and virtual memory.

KEYWORDS

processor microarchitecture, cache memories, instructions fetching, register renaming, instruction decoding, instruction issuing, instruction execution, misspeculation recovery

Contents

1.	Introduction	1
1.1	Classification of Microarchitectures	1
1.1.1	Pipelines/Nonpipelined Processors.....	1
1.1.2	In-Order/Out-of-Order Processors.....	2
1.1.3	Scalar/Superscalar Processors	2
1.1.4	Vector Processors	3
1.1.5	Multicore Processors.....	3
1.1.6	Multithreaded Processors	3
1.2	Classification of Market Segments	3
1.3	Overview of a Processor	4
1.3.1	Overview of the Pipeline	5
2.	Caches	9
2.1	Address Translation	10
2.2	Cache Structure Organization.....	11
2.2.1	Parallel Tag and Data Array Access.....	12
2.2.2	Serial Tag and Data Array Access.....	13
2.2.3	Associativity Considerations.....	15
2.3	Lockup-Free Caches	15
2.3.1	Implicitly Addressed MSHRs	16
2.3.2	Explicitly Addressed MSHRs	16
2.3.3	In-Cache MSHRs	17
2.4	Multiported Caches.....	17
2.4.1	True Multiported Cache Design	17
2.4.2	Array Replication	18
2.4.3	Virtual Multiporting.....	18
2.4.4	Multibanking.....	18

2.5	Instruction Caches.....	19
2.5.1	Multiported vs. Single Ported.....	19
2.5.2	Lockup Free vs. Blocking	20
2.5.3	Other Considerations	20
3.	The Instruction Fetch Unit.....	21
3.1	Instruction Cache	22
3.1.1	Trace Cache.....	23
3.2	Branch Target Buffer.....	23
3.3	Return Address Stack.....	24
3.4	Conditional Branch Prediction.....	25
3.4.1	Static Prediction	25
3.4.2	Dynamic Prediction.....	25
4.	Decode	31
4.1	RISC Decoding.....	31
4.2	The x86 ISA.....	32
4.3	Dynamic Translation	34
4.4	High-Performance x86 Decoding	35
4.4.1	The Instruction Length Decoder	35
4.4.2	The Dynamic Translation Unit	36
5.	Allocation.....	39
5.1	Renaming through the Reorder Buffer.....	40
5.2	Renaming through a Rename Buffer.....	41
5.3	Merged Register File.....	42
5.4	Register File Read	43
5.5	Recovery in Case of Misspeculation	44
5.6	Comparison of the Three Schemes.....	44
6.	The Issue Stage.....	47
6.1	Introduction.....	47
6.2	In-Order Issue Logic.....	47
6.3	Out-of-Order Issue Logic.....	48
6.3.1	Issue Process when Source Operands Are Read before Issue	49
6.3.1.1	Issue Queue Allocation.....	50
6.3.1.2	Instruction Wakeup	51

6.3.1.3	Instruction Selection.....	54
6.3.1.4	Entry Reclamation.....	54
6.3.2	Issue Process when Source Operands Are Read after Issue.....	55
6.3.2.1	Read Port Reduction	57
6.3.3	Other Implementations for Out-of-Order Issue	58
6.3.3.1	Distributed Issue Queue	58
6.3.3.2	Reservation Stations	58
6.4	Issue Logic for Memory Operations	58
6.4.1	Nonspeculative Memory Disambiguation	59
6.4.1.1	Case Study 1: Load Ordering and Store Ordering on an AMD K6 Processor	60
6.4.1.2	Case Study 2: Partial Ordering on a MIPS R10000 Processor	62
6.4.2	Speculative Memory Disambiguation	65
6.4.2.1	Case Study: Alpha 21264	66
6.5	Speculative Wakeup of Load Consumers	67
7.	Execute.....	69
7.1	Functional Units	71
7.1.1	The Integer Arithmetic and Logical Unit	71
7.1.2	Integer Multiplication and Division	71
7.1.3	The Address Generation Unit	71
7.1.4	The Branch Unit	73
7.1.5	The Floating-Point Unit	74
7.1.6	The SIMD Unit	75
7.2	Result Bypassing.....	78
7.2.1	Bypass in a Small Out-of-Order Machine	80
7.2.2	Multilevel Bypass for Wide Out-of-Order Machines	81
7.2.3	Bypass for In-Order Machines	83
7.2.4	Organization of Functional Units.....	86
7.3	Clustering.....	87
7.3.1	Clustering the Bypass Network.....	87
7.3.2	Clustering with Replicated Register Files	88
7.3.3	Clustering with Distributed Issue Queue and Register Files.....	89

8.	The Commit Stage	91
8.1	Introduction.....	91
8.2	Architectural State Management.....	92
8.2.1	Architectural State Based on a Retire Register File.....	93
8.2.2	Architectural State Based on a Merged Register File.....	95
8.3	Recovery of the Speculative State.....	96
8.3.1	Recovery from a Branch Misprediction.....	96
8.3.1.1	Handling Branch Mispredictions on an ROB-Based Architecture with RRF.....	97
8.3.1.2	Handling Branch Mispredictions on a Merged Register File.....	98
8.3.2	Recovery from an Exception.....	99
	References	101
	Author Biographies.....	105