

The Memory System

You Can't Avoid It,
You Can't Ignore It,
You Can't Fake It

Synthesis Lectures on Computer Architecture

Editor

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

Synthesis Lectures on Computer Architecture publishes 50 to 150 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 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, Urs Hölzle
2009

Computer Architecture Techniques for Power-Efficiency

Stefanos Kaxiras, Margaret Martonosi
2008

Chip Multiprocessor Architecture: Techniques to Improve Throughput and Latency

Kunle Olukotun, Lance Hammond, James Laudon
2007

Transactional Memory

James R. Larus, Ravi Rajwar
2006

Quantum Computing for Computer Architects

Tzvetan S. Metodi, Frederic T. Chong
2006

© Springer Nature Switzerland AG 2022

Reprint of original edition ©Morgan & Claypool 2009

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.

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

Bruce Jacob

ISBN: 978-3-031-00596-1 paperback

ISBN: 978-3-031-01724-7 ebook

DOI 10.1007/978-3-031-01724-7

A Publication in the Springer series

SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE

Lecture #7

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

Series ISSN

Synthesis Lectures on Computer Architecture

Print 1935-3235 Electronic 1935-3243

The Memory System

**You Can't Avoid It,
You Can't Ignore It,
You Can't Fake It**

Bruce Jacob
University of Maryland

with contributions by

Sadagopan Srinivasan
Intel

David T. Wang
MetaRAM

ABSTRACT

Today, computer-system optimization, at both the hardware and software levels, must consider the details of the memory system in its analysis; failing to do so yields systems that are increasingly inefficient as those systems become more complex. This lecture seeks to introduce the reader to the most important details of the memory system; it targets both computer scientists and computer engineers in industry and in academia. Roughly speaking, computer scientists are the users of the memory system, and computer engineers are the designers of the memory system. Both can benefit tremendously from a basic understanding of how the memory system really works: the computer scientist will be better equipped to create algorithms that perform well, and the computer engineer will be better equipped to design systems that approach the optimal, given the resource limitations. Currently, there is consensus among architecture researchers that the memory system is "the bottleneck," and this consensus has held for over a decade. Somewhat inexplicably, most of the research in the field is still directed toward improving the CPU to better tolerate a slow memory system, as opposed to addressing the weaknesses of the memory system directly. This lecture should get the bulk of the computer science and computer engineering population up the steep part of the learning curve. Not every CS/CE researcher/developer needs to do work in the memory system, but, just as a carpenter can do his job more efficiently if he knows a little of architecture, and an architect can do his job more efficiently if he knows a little of carpentry, giving the CS/CE worlds better intuition about the memory system should help them build better systems, both software and hardware.

KEYWORDS

memory systems, storage systems, memory scheduling, system simulation, memory-system design, prefetching, hash-associative cache, virtual memory, superpages, memory power, storage power, translation lookaside buffers, cache design, DRAM systems, memory bandwidth, memory latency, memory trends

Contents

	Prelude: Why Should I Care About the Memory System?	1
1	Primers	3
1.1	Your Code Does Not Run in a Vacuum	3
1.1.1	Data and its Representation 3	
1.1.2	Variables and Stack Allocation 7	
1.1.3	‘Random Access’ is Anything But 10	
1.2	Performance Perspective	15
1.3	Memory-System Organization and Operation	19
1.4	State of the (DRAM) Union	25
2	It Must Be Modeled Accurately	29
2.1	Some Context	30
2.2	Modeling the Memory System	32
2.3	Comparing the Models	34
2.4	Let’s Add Prefetching to the Mix	39
2.5	Summary	40
3	... and It Will Change Soon	43
3.1	Problems and Trends	43
3.1.1	The use of Multiple Cores Increases Working-Set Demands 43	
3.1.2	Multicore Bandwidth Requirement is Roughly 1GB/s per Core 44	
3.1.3	TLB Reach does not Scale Well (... or at all, Really) 45	
3.1.4	You Cannot Physically Connect to all the DRAM you can Afford to Purchase 46	
3.1.5	DRAM Refresh is Becoming Expensive in Both Power and Time 46	

3.1.6	Flash is Eating Disk's Lunch	47
3.1.7	For Large Systems, Power Dissipation of DRAM Exceeds that of CPUs	47
3.1.8	On-Chip Cache Hierarchies are Complex	48
3.1.9	Disk Access is Still Slow	50
3.1.10	There are too Many Wires on a Typical Motherboard as it is	50
3.1.11	Numerous New Technologies are in Development	50
3.2	Some Obvious Conclusions	50
3.2.1	A New DRAM-System Organization is Needed	51
3.2.2	Flash Needs to be Integrated	51
3.2.3	Possibly Revisit Superpages	51
3.3	Some Suggestions	52
3.3.1	Fully Buffered DIMM, take 2 (aka "BOMB")	52
3.3.2	Some Uses for Flash	55
3.3.3	Superpages (Take 2) and SuperTLBs	56
3.3.4	The Hash-Associative Cache	56
3.4	Virtual Memory in the Age of Cheap Memory	59
	Postlude: You Can't Fake It	61
	Bibliography	65
	Biography	69