

# Computer Organization and Design, 5th Edition: The Hardware/Software Interface

## 1 Computer Abstractions and Technology

1.1 Introduction

1.2 Eight Great Ideas in Computer Architecture

1.3 Below Your Program

1.4 Under the Covers

1.5 Technologies for Building Processors and Memory

1.6 Performance

1.7 The Power Wall

1.8 The Sea Change: The Switch from Uniprocessors to Multiprocessors

~~1.9 Real Stuff: Benchmarking the Intel Core i7~~

1.10 Fallacies and Pitfalls

1.11 Concluding Remarks

~~1.12 Historical Perspective and Further Reading~~

~~1.13 Exercises~~

## 2 Instructions: Language of the Computer

2.1 Introduction

2.2 Operations of the Computer Hardware

2.3 Operands of the Computer Hardware

2.4 Signed and Unsigned Numbers

2.5 Representing Instructions in the Computer

2.6 Logical Operations

2.7 Instructions for Making Decisions

2.8 Supporting Procedures in Computer Hardware

2.9 Communicating with People

2.10 MIPS Addressing for 32-Bit immediates and addresses

2.11 Parallelism and Instructions: Synchronization

2.12 Translating and Starting a Program

- 2.13 A C Sort Example to Put It All Together
- 2.14 Arrays versus Pointers
- ~~2.15 Advanced Material: Compiling C and Interpreting Java~~
- ~~2.16 Real Stuff: ARM v7 (32-bit) Instructions~~
- ~~2.17 Real Stuff: x86 Instructions~~
- ~~2.18 Real Stuff: ARM v8 (64-bit) Instructions~~
- 2.19 Fallacies and Pitfalls
- 2.20 Concluding Remarks
- ~~2.21 Historical Perspective and Further Reading~~
- ~~2.22 Exercises~~

### **3 Arithmetic for Computers**

- 3.1 Introduction
- 3.2 Addition and Subtraction
- 3.3 Multiplication
- 3.4 Division
- ~~3.5 Floating Point~~
- ~~3.6 Parallelism and Computer Arithmetic: Subword Parallelism~~
- ~~3.7 Real Stuff: x86 Streaming SIMD Extensions and Advanced Vector Extensions~~
- ~~3.8 Going Faster: Subword Parallelism and Matrix Multiply~~
- 3.9 Fallacies and Pitfalls
- 3.10 Concluding Remarks
- 3.11 Historical Perspective and Further Reading
- ~~3.12 Exercises~~

### **4 The Processor**

- 4.1 Introduction
- 4.2 Logic Design Conventions
- 4.3 Building a Datapath
- 4.4 A Simple Implementation Scheme
- 4.5 An Overview of Pipelining
- 4.6 Pipelined Datapath and Control

- 4.7 Data Hazards: Forwarding versus Stalling
- 4.8 Control Hazards
- 4.9 Exceptions
- 4.10 Parallelism via Instructions
- 4.11 Real Stuff: The ARM Cortex-A8 and Intel Core i7 Pipelines
- 4.12 Going Faster: Instruction-Level Parallelism and Matrix Multiply
- 4.13 Advanced Topic: an Introduction to Digital Design Using a Hardware Design Language to Describe and Model a Pipeline and More Pipelining Illustrations
- 4.14 Fallacies and Pitfalls
- 4.15 Concluding Remarks
- 4.16 Historical Perspective and Further Reading
- 4.17 Exercises XXX

## **5 Large and Fast: Exploiting Memory Hierarchy**

- 5.1 Introduction
- 5.2 Memory Technologies
- 5.3 The Basics of Caches
- 5.4 Measuring and Improving Cache Performance
- 5.5 Dependable Memory
- 5.6 Virtual Machines
- 5.7 Virtual Memory
- 5.8 A Common Framework for Memory Hierarchy
- 5.9 Using a Finite-State Machine to Control a Simple Cache
- 5.10 Parallelism and Memory Hierarchies: Cache Coherence
- 5.11 Parallelism and Memory Hierarchy: Redundant Arrays of Inexpensive Disks
- 5.12 Advanced Material: Implementing Cache Controllers
- 5.13 Real Stuff: The ARM Cortex-A8 and Intel Core i7 Memory Hierarchies
- 5.14 Going Faster: Cache Blocking and Matrix Multiply
- 5.15 Fallacies and Pitfalls
- 5.16 Concluding Remarks
- 5.17 Historical Perspective and Further Reading

## 5.18 Exercises

## 6 Parallel Processors from Client to Cloud

### 6.1 Introduction

### 6.2 The Difficulty of Creating Parallel Processing Programs

### 6.3 SISD, MIMD, SIMD, SPMD, and Vector

### 6.4 Hardware Multithreading

### 6.5 Multicore and Other Shared Memory Multiprocessors

### 6.6 Introduction to Graphics Processing Units

### 6.7 Clusters and Other Message-Passing Multiprocessors

### 6.8 Introduction to Multiprocessor Network Topologies

### 6.9 Communicating to the Outside World: Cluster Networking

### 6.10 Multiprocessor Benchmarks and Performance Models

### 6.11 Real Stuff: Benchmarking Intel Core i7 versus NVIDIA Fermi GPU

### 6.12 Going Faster: Multiple Processors and Matrix Multiply

### 6.13 Fallacies and Pitfalls

### 6.14 Concluding Remarks

### 6.15 Historical Perspective and Further Reading

### 6.16 Exercises

## APPENDICES

### A Assemblers, Linkers, and the SPIM Simulator

#### A.1 Introduction A-3

#### A.2 Assemblers A-10

#### A.3 Linkers A-18

#### A.4 Loading A-19

#### A.5 Memory Usage A-20

#### A.6 Procedure Call Convention A-22

#### A.7 Exceptions and Interrupts A-33

#### A.8 Input and Output A-38

#### A.9 SPIM A-40

#### A.10 MIPS R2000 Assembly Language A-45

A.11 Concluding Remarks	A-81
<del>A.12 Exercises</del>	<del>A-82</del>
B The Basics of Logic Design	
B.1 Introduction	B-3
B.2 Gates, Truth Tables, and Logic Equations	B-4
B.3 Combinational Logic	B-9
<del>B.4 Using a Hardware Description Language</del>	<del>B-20</del>
B.5 Constructing a Basic Arithmetic Logic Unit	B-26
<del>B.6 Faster Addition: Carry Lookahead</del>	<del>B-38</del>
B.7 Clocks	B-48
B.8 Memory Elements: Flip-Flops, Latches, and Registers	B-50
B.9 Memory Elements: SRAMs and DRAMs	B-58
B.10 Finite-State Machines	B-67
B.11 Timing Methodologies	B-72
<del>B.12 Field Programmable Devices</del>	<del>B-78</del>
B.13 Concluding Remarks	B-79
<del>B.14 Exercises</del>	<del>B-80</del>
ONLINE CONTENT	
<del>C Graphics and Computing GPUs</del>	
<del>C.1 Introduction</del>	<del>C-3</del>
<del>C.2 GPU System Architectures</del>	<del>C-7</del>
<del>C.3 Programming GPUs</del>	<del>C-12</del>
<del>C.4 Multithreaded Multiprocessor Architecture</del>	<del>C-25</del>
<del>C.5 Parallel Memory System</del>	<del>C-36</del>
<del>C.6 Floating Point Arithmetic</del>	<del>C-41</del>
<del>C.7 Real Stuff: The NVIDIA GeForce 8800</del>	<del>C-46</del>
<del>C.8 Real Stuff: Mapping Applications to GPUs</del>	<del>C-55</del>
<del>C.9 Fallacies and Pitfalls</del>	<del>C-72</del>
<del>C.10 Concluding Remarks</del>	<del>C-76</del>

C.11 Historical Perspective and Further Reading C-77

D Mapping Control to Hardware

D.1 Introduction D-3

D.2 Implementing Combinational Control Units D-4

D.3 Implementing Finite-State Machine Control D-8

D.4 Implementing the Next-State Function with a Sequencer D-22

D.5 Translating a Microprogram to Hardware D-28

D.6 Concluding Remarks D-32

D.7 Exercises D-33

E A Survey of RISC Architectures for Desktop, Server, and Embedded Computers

E.1 Introduction E-3

E.2 Addressing Modes and Instruction Formats E-5

E.3 Instructions: The MIPS Core Subset E-9

E.4 Instructions: Multimedia Extensions of the Desktop/Server RISCs E-16

E.5 Instructions: Digital Signal Processing Extensions of the Embedded RISCs E-19

E.6 Instructions: Common Extensions to MIPS Core E-20

E.7 Instructions Unique to MIPS-64 E-25

E.8 Instructions Unique to Alpha E-27

E.9 Instructions Unique to SPARC v.9 E-29

E.10 Instructions Unique to PowerPC E-32

E.11 Instructions Unique to PA-RISC 2.0 E-34

E.12 Instructions Unique to ARM E-36

E.13 Instructions Unique to Thumb E-38

E.14 Instructions Unique to SuperH E-39

E.15 Instructions Unique to M32R E-40

E.16 Instructions Unique to MIPS-16 E-40

E.17 Concluding Remarks E-43