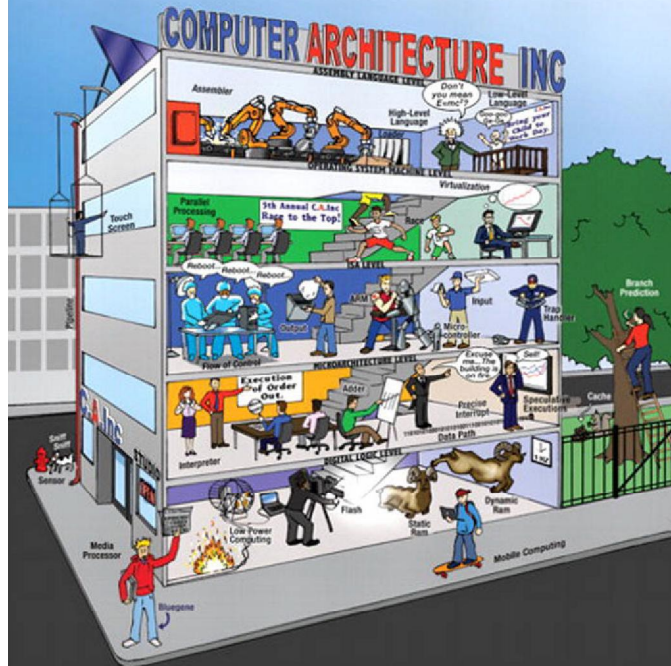


# STRUCTURED COMPUTER ORGANIZATION

SIXTH EDITION

Andrew S. Tanenbaum • Todd Austin



*This page intentionally left blank*

**STRUCTURED  
COMPUTER ORGANIZATION**

SIXTH EDITION

*This page intentionally left blank*

# STRUCTURED COMPUTER ORGANIZATION

SIXTH EDITION

**ANDREW S. TANENBAUM**

*Vrije Universiteit  
Amsterdam, The Netherlands*

**TODD AUSTIN**

*University of Michigan  
Ann Arbor, Michigan, United States*

**PEARSON**

Boston Columbus Indianapolis New York San Francisco Upper Saddle River  
Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montreal Toronto  
Delhi Mexico City Sao Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

Editorial Director, ECS: Marcia Horton  
Executive Editor: Tracy Johnson (Dunkelberger)  
Associate Editor: Carole Snyder  
Director of Marketing: Christy Lesko  
Marketing Manager: Yez Alayan  
Senior Marketing Coordinator: Kathryn Ferranti  
Director of Production: Erin Gregg  
Managing Editor: Jeff Holcomb  
Associate Managing Editor: Robert Engelhardt  
Manufacturing Buyer: Lisa McDowell  
Art Director: Anthony Gemmellaro  
Cover Illustrator: Jason Consalvo  
Manager, Rights and Permissions: Michael Joyce  
Media Editor: Daniel Sandin  
Media Project Manager: Renata Butera  
Printer/Binder: Courier/Westford  
Cover Printer: Lehigh-Phoenix Color/Hagerstown

Credits and acknowledgments borrowed from other sources and reproduced, with permission, in this textbook appear in the Credits section in the end matter of this text.

Copyright © 2013, 2006, 1999 Pearson Education, Inc., publishing as Prentice Hall. All rights reserved. Printed in the United States of America. This publication is protected by Copyright, and permission should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. To obtain permission(s) to use material from this work, please submit a written request to Pearson Education, Inc., Permissions Department, One Lake Street, Upper Saddle River, New Jersey 07458, or you may fax your request to 201-236-3290.

Many of the designations by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed in initial caps or all caps.

#### **Library of Congress Cataloging-in-Publication Data**

Tanenbaum, Andrew S.,

Structured computer organization / Andrew S. Tanenbaum, Todd Austin. -- 6th ed.

p. cm.

Includes bibliographical references and index.

ISBN-13: 978-0-13-291652-3

ISBN-10: 0-13-291652-5

1. Computer programming. 2. Computer organization. I. Austin, Todd. II. Title.

QA76.6.T38 2013

005.1--dc23

2012021627

10 9 8 7 6 5 4 3 2 1

**PEARSON**

ISBN 10: 0-13-291652-5

ISBN 13: 978-0-13-291652-3

*AST: Suzanne, Barbara, Marvin, Aron and Nathan*

*TA: To Roberta, who made space (and time) for me to finish this project.*

*This page intentionally left blank*



# CONTENTS

**PREFACE**

**xix**

## **1 INTRODUCTION**

- 1.1 STRUCTURED COMPUTER ORGANIZATION 2
  - 1.1.1 Languages, Levels, and Virtual Machines 2
  - 1.1.2 Contemporary Multilevel Machines 5
  - 1.1.3 Evolution of Multilevel Machines 8
  
- 1.2 MILESTONES IN COMPUTER ARCHITECTURE 13
  - 1.2.1 The Zeroth Generation—Mechanical Computers (1642–1945) 13
  - 1.2.2 The First Generation—Vacuum Tubes (1945–1955) 16
  - 1.2.3 The Second Generation—Transistors (1955–1965) 19
  - 1.2.4 The Third Generation—Integrated Circuits (1965–1980) 21
  - 1.2.5 The Fourth Generation—Very Large Scale Integration (1980–?) 23
  - 1.2.6 The Fifth Generation—Low-Power and Invisible Computers 26
  
- 1.3 THE COMPUTER ZOO 28
  - 1.3.1 Technological and Economic Forces 28
  - 1.3.2 The Computer Spectrum 30
  - 1.3.3 Disposable Computers 31
  - 1.3.4 Microcontrollers 33
  - 1.3.5 Mobile and Game Computers 35
  - 1.3.6 Personal Computers 36
  - 1.3.7 Servers 36
  - 1.3.8 Mainframes 38

- 1.4 EXAMPLE COMPUTER FAMILIES 39
  - 1.4.1 Introduction to the x86 Architecture 39
  - 1.4.2 Introduction to the ARM Architecture 45
  - 1.4.3 Introduction to the AVR Architecture 47
- 1.5 METRIC UNITS 49
- 1.6 OUTLINE OF THIS BOOK 50

## 2 COMPUTER SYSTEMS

- 2.1 PROCESSORS 55
  - 2.1.1 CPU Organization 56
  - 2.1.2 Instruction Execution 58
  - 2.1.3 RISC versus CISC 62
  - 2.1.4 Design Principles for Modern Computers 63
  - 2.1.5 Instruction-Level Parallelism 65
  - 2.1.6 Processor-Level Parallelism 69
- 2.2 PRIMARY MEMORY 73
  - 2.2.1 Bits 74
  - 2.2.2 Memory Addresses 74
  - 2.2.3 Byte Ordering 76
  - 2.2.4 Error-Correcting Codes 78
  - 2.2.5 Cache Memory 82
  - 2.2.6 Memory Packaging and Types 85
- 2.3 SECONDARY MEMORY 86
  - 2.3.1 Memory Hierarchies 86
  - 2.3.2 Magnetic Disks 87
  - 2.3.3 IDE Disks 91
  - 2.3.4 SCSI Disks 92
  - 2.3.5 RAID 94
  - 2.3.6 Solid-State Disks 97
  - 2.3.7 CD-ROMs 99
  - 2.3.8 CD-Recordables 103
  - 2.3.9 CD-Rewritables 105
  - 2.3.10 DVD 106
  - 2.3.11 Blu-ray 108

- 2.4 INPUT/OUTPUT 108
  - 2.4.1 Buses 108
  - 2.4.2 Terminals 113
  - 2.4.3 Mice 118
  - 2.4.4 Game Controllers 120
  - 2.4.5 Printers 122
  - 2.4.6 Telecommunications Equipment 127
  - 2.4.7 Digital Cameras 135
  - 2.4.8 Character Codes 137
- 2.5 SUMMARY 142

### **3 THE DIGITAL LOGIC LEVEL**

- 3.1 GATES AND BOOLEAN ALGEBRA 147
  - 3.1.1 Gates 148
  - 3.1.2 Boolean Algebra 150
  - 3.1.3 Implementation of Boolean Functions 152
  - 3.1.4 Circuit Equivalence 153
- 3.2 BASIC DIGITAL LOGIC CIRCUITS 158
  - 3.2.1 Integrated Circuits 158
  - 3.2.2 Combinational Circuits 159
  - 3.2.3 Arithmetic Circuits 163
  - 3.2.4 Clocks 168
- 3.3 MEMORY 169
  - 3.3.1 Latches 169
  - 3.3.2 Flip-Flops 172
  - 3.3.3 Registers 174
  - 3.3.4 Memory Organization 174
  - 3.3.5 Memory Chips 178
  - 3.3.6 RAMs and ROMs 180
- 3.4 CPU CHIPS AND BUSES 185
  - 3.4.1 CPU Chips 185
  - 3.4.2 Computer Buses 187
  - 3.4.3 Bus Width 190
  - 3.4.4 Bus Clocking 191
  - 3.4.5 Bus Arbitration 196
  - 3.4.6 Bus Operations 198

- 3.5 EXAMPLE CPU CHIPS 201
  - 3.5.1 The Intel Core i7 201
  - 3.5.2 The Texas Instruments OMAP4430 System-on-a-Chip 208
  - 3.5.3 The Atmel ATmega168 Microcontroller 212
- 3.6 EXAMPLE BUSES 214
  - 3.6.1 The PCI Bus 215
  - 3.6.2 PCI Express 223
  - 3.6.3 The Universal Serial Bus 228
- 3.7 INTERFACING 232
  - 3.7.1 I/O Interfaces 232
  - 3.7.2 Address Decoding 233
- 3.8 SUMMARY 235

## **4 THE MICROARCHITECTURE LEVEL**

- 4.1 AN EXAMPLE MICROARCHITECTURE 243
  - 4.1.1 The Data Path 244
  - 4.1.2 Microinstructions 251
  - 4.1.3 Microinstruction Control: The Mic-1 253
- 4.2 AN EXAMPLE ISA: IJVM 258
  - 4.2.1 Stacks 258
  - 4.2.2 The IJVM Memory Model 260
  - 4.2.3 The IJVM Instruction Set 262
  - 4.2.4 Compiling Java to IJVM 265
- 4.3 AN EXAMPLE IMPLEMENTATION 267
  - 4.3.1 Microinstructions and Notation 267
  - 4.3.2 Implementation of IJVM Using the Mic-1 271
- 4.4 DESIGN OF THE MICROARCHITECTURE LEVEL 283
  - 4.4.1 Speed versus Cost 283
  - 4.4.2 Reducing the Execution Path Length 285
  - 4.4.3 A Design with Prefetching: The Mic-2 291
  - 4.4.4 A Pipelined Design: The Mic-3 293
  - 4.4.5 A Seven-Stage Pipeline: The Mic-4 300

- 4.5 IMPROVING PERFORMANCE 303
  - 4.5.1 Cache Memory 304
  - 4.5.2 Branch Prediction 310
  - 4.5.3 Out-of-Order Execution and Register Renaming 315
  - 4.5.4 Speculative Execution 320
- 4.6 EXAMPLES OF THE MICROARCHITECTURE LEVEL 323
  - 4.6.1 The Microarchitecture of the Core i7 CPU 323
  - 4.6.2 The Microarchitecture of the OMAP4430 CPU 329
  - 4.6.3 The Microarchitecture of the ATmega168 Microcontroller 334
- 4.7 COMPARISON OF THE I7, OMAP4430, AND ATMEGA168 336
- 4.8 SUMMARY 337

## **5 THE INSTRUCTION SET**

- 5.1 OVERVIEW OF THE ISA LEVEL 345
  - 5.1.1 Properties of the ISA Level 345
  - 5.1.2 Memory Models 347
  - 5.1.3 Registers 349
  - 5.1.4 Instructions 351
  - 5.1.5 Overview of the Core i7 ISA Level 351
  - 5.1.6 Overview of the OMAP4430 ARM ISA Level 354
  - 5.1.7 Overview of the ATmega168 AVR ISA Level 356
- 5.2 DATA TYPES 358
  - 5.2.1 Numeric Data Types 358
  - 5.2.2 Nonnumeric Data Types 359
  - 5.2.3 Data Types on the Core i7 360
  - 5.2.4 Data Types on the OMAP4430 ARM CPU 361
  - 5.2.5 Data Types on the ATmega168 AVR CPU 361
- 5.3 INSTRUCTION FORMATS 362
  - 5.3.1 Design Criteria for Instruction Formats 362
  - 5.3.2 Expanding Opcodes 365
  - 5.3.3 The Core i7 Instruction Formats 367
  - 5.3.4 The OMAP4430 ARM CPU Instruction Formats 368
  - 5.3.5 The ATmega168 AVR Instruction Formats 370

- 5.4 ADDRESSING 371
  - 5.4.1 Addressing Modes 371
  - 5.4.2 Immediate Addressing 372
  - 5.4.3 Direct Addressing 372
  - 5.4.4 Register Addressing 372
  - 5.4.5 Register Indirect Addressing 373
  - 5.4.6 Indexed Addressing 374
  - 5.4.7 Based-Indexed Addressing 376
  - 5.4.8 Stack Addressing 376
  - 5.4.9 Addressing Modes for Branch Instructions 379
  - 5.4.10 Orthogonality of Opcodes and Addressing Modes 380
  - 5.4.11 The Core i7 Addressing Modes 382
  - 5.4.12 The OMAP4440 ARM CPU Addressing Modes 384
  - 5.4.13 The ATmega168 AVR Addressing Modes 384
  - 5.4.14 Discussion of Addressing Modes 385
  
- 5.5 INSTRUCTION TYPES 386
  - 5.5.1 Data Movement Instructions 386
  - 5.5.2 Dyadic Operations 387
  - 5.5.3 Monadic Operations 388
  - 5.5.4 Comparisons and Conditional Branches 390
  - 5.5.5 Procedure Call Instructions 392
  - 5.5.6 Loop Control 393
  - 5.5.7 Input/Output 394
  - 5.5.8 The Core i7 Instructions 397
  - 5.5.9 The OMAP4430 ARM CPU Instructions 400
  - 5.5.10 The ATmega168 AVR Instructions 402
  - 5.5.11 Comparison of Instruction Sets 402
  
- 5.6 FLOW OF CONTROL 404
  - 5.6.1 Sequential Flow of Control and Branches 405
  - 5.6.2 Procedures 406
  - 5.6.3 Coroutines 410
  - 5.6.4 Traps 413
  - 5.6.5 Interrupts 414
  
- 5.7 A DETAILED EXAMPLE: THE TOWERS OF HANOI 417
  - 5.7.1 The Towers of Hanoi in Core i7 Assembly Language 418
  - 5.7.2 The Towers of Hanoi in OMAP4430 ARM Assembly Language 418

- 5.8 THE IA-64 ARCHITECTURE AND THE ITANIUM 2 420
  - 5.8.1 The Problem with the IA-32 ISA 421
  - 5.8.2 The IA-64 Model: Explicitly Parallel Instruction Computing 423
  - 5.8.3 Reducing Memory References 423
  - 5.8.4 Instruction Scheduling 424
  - 5.8.5 Reducing Conditional Branches: Predication 426
  - 5.8.6 Speculative Loads 429
- 5.9 SUMMARY 430

## 6 THE OPERATING SYSTEM

- 6.1 VIRTUAL MEMORY 438
  - 6.1.1 Paging 439
  - 6.1.2 Implementation of Paging 441
  - 6.1.3 Demand Paging and the Working-Set Model 443
  - 6.1.4 Page-Replacement Policy 446
  - 6.1.5 Page Size and Fragmentation 448
  - 6.1.6 Segmentation 449
  - 6.1.7 Implementation of Segmentation 452
  - 6.1.8 Virtual Memory on the Core i7 455
  - 6.1.9 Virtual Memory on the OMAP4430 ARM CPU 460
  - 6.1.10 Virtual Memory and Caching 462
- 6.2 HARDWARE VIRTUALIZATION 463
  - 6.2.1 Hardware Virtualization on the Core I7 464
- 6.3 OSM-LEVEL I/O INSTRUCTIONS 465
  - 6.3.1 Files 465
  - 6.3.2 Implementation of OSM-Level I/O Instructions 467
  - 6.3.3 Directory Management Instructions 471
- 6.4 OSM-LEVEL INSTRUCTIONS FOR PARALLEL PROCESSING 471
  - 6.4.1 Process Creation 473
  - 6.4.2 Race Conditions 473
  - 6.4.3 Process Synchronization Using Semaphores 478
- 6.5 EXAMPLE OPERATING SYSTEMS 480
  - 6.5.1 Introduction 482
  - 6.5.2 Examples of Virtual Memory 488

- 6.5.3 Examples of OS-Level I/O 492
- 6.5.4 Examples of Process Management 503

6.6 SUMMARY 509

## **7 THE ASSEMBLY LANGUAGE LEVEL**

- 7.1 INTRODUCTION TO ASSEMBLY LANGUAGE 518
  - 7.1.1 What Is an Assembly Language? 518
  - 7.1.2 Why Use Assembly Language? 519
  - 7.1.3 Format of an Assembly Language Statement 520
  - 7.1.4 Pseudoinstructions 522
- 7.2 MACROS 524
  - 7.2.1 Macro Definition, Call, and Expansion 524
  - 7.2.2 Macros with Parameters 526
  - 7.2.3 Advanced Features 527
  - 7.2.4 Implementation of a Macro Facility in an Assembler 528
- 7.3 THE ASSEMBLY PROCESS 529
  - 7.3.1 Two-Pass Assemblers 529
  - 7.3.2 Pass One 530
  - 7.3.3 Pass Two 534
  - 7.3.4 The Symbol Table 535
- 7.4 LINKING AND LOADING 536
  - 7.4.1 Tasks Performed by the Linker 538
  - 7.4.2 Structure of an Object Module 541
  - 7.4.3 Binding Time and Dynamic Relocation 542
  - 7.4.4 Dynamic Linking 545
- 7.5 SUMMARY 549

## **8 PARALLEL COMPUTER ARCHITECTURES**

- 8.1 ON-CHIP PARALELLISM 554
  - 8.1.1 Instruction-Level Parallelism 555
  - 8.1.2 On-Chip Multithreading 562
  - 8.1.3 Single-Chip Multiprocessors 568



- 8.2 COPROCESSORS 574
  - 8.2.1 Network Processors 574
  - 8.2.2 Graphics Processors 582
  - 8.2.3 Cryptoprocessors 585
- 8.3 SHARED-MEMORY MULTIPROCESSORS 586
  - 8.3.1 Multiprocessors vs. Multicomputers 586
  - 8.3.2 Memory Semantics 593
  - 8.3.3 UMA Symmetric Multiprocessor Architectures 598
  - 8.3.4 NUMA Multiprocessors 606
  - 8.3.4 COMA Multiprocessors 615
- 8.4 MESSAGE-PASSING MULTICOMPUTERS 616
  - 7.4.1 Interconnection Networks 618
  - 8.4.2 MPPs—Massively Parallel Processors 621
  - 8.4.3 Cluster Computing 631
  - 8.4.4 Communication Software for Multicomputers 636
  - 8.4.5 Scheduling 639
  - 8.4.6 Application-Level Shared Memory 640
  - 8.4.7 Performance 646
- 8.5 GRID COMPUTING 652
- 8.6 SUMMARY 655

## **9 BIBLIOGRAPHY 659**

## **A BINARY NUMBERS 669**

- A.1 FINITE-PRECISION NUMBERS 669
- A.2 RADIX NUMBER SYSTEMS 671
- A.3 CONVERSION FROM ONE RADIX TO ANOTHER 673
- A.4 NEGATIVE BINARY NUMBERS 675
- A.5 BINARY ARITHMETIC 678

**B FLOATING-POINT NUMBERS 681**

- B.1 PRINCIPLES OF FLOATING POINT 682
- B.2 IEEE FLOATING-POINT STANDARD 754 684

**C ASSEMBLY LANGUAGE PROGRAMMING 691**

- C.1 OVERVIEW 692
  - C.1.1 Assembly Language 692
  - C.1.2 A Small Assembly Language Program 693
- C.2 THE 8088 PROCESSOR 694
  - C.2.1 The Processor Cycle 695
  - C.2.2 The General Registers 695
  - C.2.3 Pointer Registers 698
- C.3 MEMORY AND ADDRESSING 699
  - C.3.1 Memory Organization and Segments 699
  - C.3.2 Addressing 701
- C.4 THE 8088 INSTRUCTION SET 705
  - C.4.1 Move, Copy and Arithmetic 705
  - C.4.2 Logical, Bit and Shift Operations 708
  - C.4.3 Loop and Repetitive String Operations 708
  - C.4.4 Jump and Call Instructions 709
  - C.4.5 Subroutine Calls 710
  - C.4.6 System Calls and System Subroutines 712
  - C.4.7 Final Remarks on the Instruction Set 715
- C.5 THE ASSEMBLER 715
  - C.5.1 Introduction 715
  - C.5.2 The ACK-Based Tutorial Assembler as88 716
  - C.5.3 Some Differences with Other 8088 Assemblers 720
- C.6 THE TRACER 721
  - C.6.1 Tracer Commands 723
- C.7 GETTING STARTED 725

C.8	EXAMPLES	726
C.8.1	Hello World Example	726
C.8.2	General Registers Example	729
C.8.3	Call Command and Pointer Registers	730
C.8.4	Debugging an Array Print Program	734
C.8.5	String Manipulation and String Instructions	736
C.8.6	Dispatch Tables	740
C.8.7	Buffered and Random File Access	742