

Carl Hamacher et al.: Computer Organization, 5E(ISBN 0-07-112218-4).

Copyright © 2002 by The McGraw-Hill Companies, Inc. All rights reserved.

Jointly published by China Machine Press/McGraw-Hill. This edition may be sold in the People's Republic of China only. This book cannot be re-exported and is not for sale outside the People's Republic of China.

本书英文影印版由美国McGraw-Hill公司授权机械工业出版社在中国大陆境内独家出版发行, 未经出版者许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

版权所有, 侵权必究。

**本书版权登记号: 图字: 01-2002-2178**

**图书在版编目(CIP)数据**

计算机组成(英文版·第5版)/(美)哈马克(Hamacher, C.)等著.

-北京:机械工业出版社, 2002.6

(经典原版书库)

书名原文: Computer Organization, 5E

ISBN 7-111-10346-7

I. 计… II. ①哈… III. 计算机体系结构-英文 IV. TP303

中国版本图书馆CIP数据核字(2002)第038014号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 华 章

北京昌平奔腾印刷厂印刷·新华书店北京发行所发行

2002年6月第1版第1次印刷

850mm×1168mm 1/32·26印张

印数: 0 001-3 000册

定价: 48.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

# 出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域中取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及收藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：针对本科生的核心课程，剔除外版菁华而成“国外经典教材”系列；对影印版的教材，则单独开辟出“经典原版书库”；定位在高级教程和专业参考的“计算机科学丛书”还将保持原来的风格，继续出版新的品种。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

“经典原版书库”是响应教育部提出的使用原版国外教材的号召，为国内高校的计算机教学度身订造的。在广泛地征求并听取丛书的“专家指导委员会”的意见后，我们最终选定了这30多种篇幅内容适度、讲解鞭辟入里的教材，其中的大部分已经被MIT、Stanford、U.C. Berkley、C.M.U.等世界名牌大学采用。丛书不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

电子邮件：[hzedu@hzbook.com](mailto:hzedu@hzbook.com)

联系电话：(010) 68995265

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

# 专家指导委员会

(按姓氏笔画顺序)

尤晋元	王珊	冯博琴	史忠植	史美林
石教英	吕建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅宏	程旭	程时端	谢希仁
裘宗燕	戴葵			

# ABOUT THE AUTHORS

**Carl Hamacher** received his B.A.Sc. degree in engineering physics from the University of Waterloo, Canada, an M.Sc. degree in electrical engineering from Queen's University, Kingston, Canada, and a Ph.D. degree in electrical engineering from Syracuse University, New York. From 1968 to 1990 he was at the University of Toronto, where he was a Professor in the Departments of Electrical Engineering and Computer Science. He served as director of the Computer Systems Research Institute during 1984 to 1988, and as chairman of the Division of Engineering Science during 1988 to 1990. Since January 1991 he has been a Professor of Electrical and Computer Engineering at Queen's University. He served as the dean of the Faculty of Applied Science from 1991 to 1996. During 1978 to 1979, he was a visiting scientist at the IBM Research Laboratory in San Jose, California. In 1986, he was a research visitor at the Laboratory for Circuits and Systems associated with the University of Grenoble in France. In 1996 to 1997, he was a visiting professor in the Computer Science Department at the University of California at Riverside and in the LIP6 Laboratory of the University of Paris VI, France.

His research interests are in multiprocessors and multicomputers, focusing on their interconnection networks.

**Zvonko Vranesic** received his B.A.Sc., M.A.Sc., and Ph.D. degrees, in electrical engineering from the University of Toronto. From 1963 to 1965 he worked as a design engineer with the Northern Electric Co., Ltd. in Bramalea, Ontario. In 1968, he joined the University of Toronto, where he is now a Professor in the Department of Electrical and Computer Engineering and the Department of Computer Science. During 1978 to 1979, he was a senior visitor at the University of Cambridge, England, and during 1984 to 1985 he was at the University of Paris VI, France. In 2000 to 2001, he was a principal software engineer at Altera Corporation in Toronto. From 1995 to 2000, he served as chair of the Division of Engineering Science at the University of Toronto.

His current research interests include computer architecture, field-programmable VLSI technology, and multiple-valued logic systems. He is a coauthor of three other books: *Fundamentals of Digital Logic with VHDL Design*, *Microcomputer Structures*, and *Field-Programmable Gate Arrays*. In 1990, he received the Wighton Fellowship for "innovative and distinctive contributions to undergraduate laboratory instruction."

**Safwat Zaky** received his B.Sc. degree in electrical engineering and B.Sc. in mathematics, both from Cairo University, Egypt, and his M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto. From 1969 to 1972 he was with Bell Northern Research, Bramalea, Ontario, where he worked on applications of electro-optics and magnetics in mass storage and telephone switching. In 1973, he joined the University of Toronto, where he is now a Professor in the Department of Electrical and Computer Engineering and the Department of Computer Science. Presently, he

serves as chair of the Department of Electrical and Computer Engineering. From 1980 to 1981, he was a senior visitor at the Computer Laboratory, University of Cambridge, England.

His research interests are in the areas of computer architecture, reliability of digital circuits, and electromagnetic compatibility. He is a coauthor of the book *Microcomputer Structures* and is a recipient of the IEEE Third Millennium Medal.

# CONTENTS

Preface xvii

## Chapter 1

### BASIC STRUCTURE OF COMPUTERS 1

- 1.1 Computer Types 2
- 1.2 Functional Units 3
  - 1.2.1 Input Unit 4
  - 1.2.2 Memory Unit 4
  - 1.2.3 Arithmetic and Logic Unit 5
  - 1.2.4 Output Unit 6
  - 1.2.5 Control Unit 6
- 1.3 Basic Operational Concepts 7
- 1.4 Bus Structures 9
- 1.5 Software 10
- 1.6 Performance 13
  - 1.6.1 Processor Clock 14
  - 1.6.2 Basic Performance Equation 14
  - 1.6.3 Pipelining and Superscalar Operation 15
  - 1.6.4 Clock Rate 16
  - 1.6.5 Instruction Set: CISC and RISC 16
  - 1.6.6 Compiler 17
  - 1.6.7 Performance Measurement 17
- 1.7 Multiprocessors and Multicomputers 18
- 1.8 Historical Perspective 19
  - 1.8.1 The First Generation 19
  - 1.8.2 The Second Generation 20
  - 1.8.3 The Third Generation 20
  - 1.8.4 The Fourth Generation 20
  - 1.8.5 Beyond the Fourth Generation 21
  - 1.8.6 Evolution of Performance 21
- 1.9 Concluding Remarks 21
  - Problems 22
  - References 23

## Chapter 2

### MACHINE INSTRUCTIONS AND PROGRAMS 25

- 2.1 Numbers, Arithmetic Operations, and Characters 27

- 2.1.1 Number Representation 27
- 2.1.2 Addition of Positive Numbers 28
- 2.1.3 Addition and Subtraction of Signed Numbers 29
- 2.1.4 Overflow in Integer Arithmetic 32
- 2.1.5 Characters 33
- 2.2 Memory Locations and Addresses 33
  - 2.2.1 Byte Addressability 33
  - 2.2.2 Big-endian and Little-endian Assignments 35
  - 2.2.3 Word Alignment 36
  - 2.2.4 Accessing Numbers, Characters, and Character Strings 36
- 2.3 Memory Operations 36
- 2.4 Instructions and Instruction Sequencing 37
  - 2.4.1 Register Transfer Notation 37
  - 2.4.2 Assembly Language Notation 38
  - 2.4.3 Basic Instruction Types 38
  - 2.4.4 Instruction Execution and Straight-Line Sequencing 42
  - 2.4.5 Branching 44
  - 2.4.6 Condition Codes 46
  - 2.4.7 Generating Memory Addresses 47
- 2.5 Addressing Modes 48
  - 2.5.1 Implementation of Variables and Constants 49
  - 2.5.2 Indirection and Pointers 50
  - 2.5.3 Indexing and Arrays 52
  - 2.5.4 Relative Addressing 56
  - 2.5.5 Additional Modes 56
- 2.6 Assembly Language 58
  - 2.6.1 Assembler Directives 59
  - 2.6.2 Assembly and Execution of Programs 62
  - 2.6.3 Number Notation 64
- 2.7 Basic Input/Output Operations 64
- 2.8 Stacks and Queues 68
- 2.9 Subroutines 72
  - 2.9.1 Subroutine Nesting and the Processor Stack 73
  - 2.9.2 Parameter Passing 74
  - 2.9.3 The Stack Frame 75
- 2.10 Additional Instructions 81

- 2.10.1 Logic Instructions 81
- 2.10.2 Shift and Rotate Instructions 82
- 2.10.3 Multiplication and Division 86
- 2.11 Example Programs 86
  - 2.11.1 Vector Dot Product Program 86
  - 2.11.2 Byte-Sorting Program 87
  - 2.11.3 Linked Lists 89
- 2.12 Encoding of Machine Instructions 94
- 2.13 Concluding Remarks 98
  - Problems 98

## Chapter 3

### ARM, MOTOROLA, AND INTEL INSTRUCTION SETS 103

#### Part I The ARM Example 104

- 3.1 Registers, Memory Access, and Data Transfer 104
  - 3.1.1 Register Structure 105
  - 3.1.2 Memory Access Instructions and Addressing Modes 106
  - 3.1.3 Register Move Instructions 113
- 3.2 Arithmetic and Logic Instructions 113
  - 3.2.1 Arithmetic Instructions 113
  - 3.2.2 Logic Instructions 115
- 3.3 Branch Instructions 116
  - 3.3.1 Setting Condition Codes 117
  - 3.3.2 A Loop Program for Adding Numbers 118
- 3.4 Assembly Language 118
  - 3.4.1 Pseudo-Instructions 120
- 3.5 I/O Operations 121
- 3.6 Subroutines 122
- 3.7 Program Examples 126
  - 3.7.1 Vector Dot Product Program 126
  - 3.7.2 Byte-Sorting Program 127
  - 3.7.3 Linked-List Insertion and Deletion Subroutines 127

#### Part II The 68000 Example 130

- 3.8 Registers and Addressing 131
  - 3.8.1 The 68000 Register Structure 131
  - 3.8.2 Addressing 131
- 3.9 Instructions 136
- 3.10 Assembly Language 140
- 3.11 Program Flow Control 141

- 3.11.1 Condition Code Flags 141
- 3.11.2 Branch Instructions 141
- 3.12 I/O Operations 145
- 3.13 Stacks and Subroutines 146
- 3.14 Logic Instructions 151
- 3.15 Program Examples 152
  - 3.15.1 Vector Dot Product Program 152
  - 3.15.2 Byte-Sorting Program 153
  - 3.15.3 Linked-List Insertion and Deletion Subroutines 154

#### Part III The IA-32 Pentium Example 155

- 3.16 Registers and Addressing 156
  - 3.16.1 IA-32 Register Structure 156
  - 3.16.2 IA-32 Addressing Modes 159
- 3.17 IA-32 Instructions 164
  - 3.17.1 Machine Instruction Format 168
- 3.18 IA-32 Assembly Language 170
- 3.19 Program Flow Control 171
  - 3.19.1 Conditional Jumps and Condition Code Flags 171
  - 3.19.2 Unconditional Jump 173
- 3.20 Logic and Shift/Rotate Instructions 173
  - 3.20.1 Logic Operations 173
  - 3.20.2 Shift and Rotate Operations 173
- 3.21 I/O Operations 174
  - 3.21.1 Memory-Mapped I/O 174
  - 3.21.2 Isolated I/O 175
  - 3.21.3 Block Transfers 176
- 3.22 Subroutines 177
- 3.23 Other Instructions 182
  - 3.23.1 Multiply and Divide Instructions 182
  - 3.23.2 Multimedia Extension (MMX) Instructions 183
  - 3.23.3 Vector (SIMD) Instructions 184
- 3.24 Program Examples 184
  - 3.24.1 Vector Dot Product Program 184
  - 3.24.2 Byte-Sorting Program 185
  - 3.24.3 Linked-List Insertion and Deletion Subroutines 185
- 3.25 Concluding Remarks 188
  - Problems 188
  - References 201

## Chapter 4

### INPUT/OUTPUT ORGANIZATION 203

- 4.1 Accessing I/O Devices 204
- 4.2 Interrupts 208



- 4.2.1 Interrupt Hardware 210
- 4.2.2 Enabling and Disabling Interrupts 211
- 4.2.3 Handling Multiple Devices 213
- 4.2.4 Controlling Device Requests 217
- 4.2.5 Exceptions 218
- 4.2.6 Use of Interrupts in Operating Systems 220
- 4.3 Processor Examples 224
  - 4.3.1 ARM Interrupt Structure 224
  - 4.3.2 68000 Interrupt Structure 229
  - 4.3.3 Pentium Interrupt Structure 231
- 4.4 Direct Memory Access 234
  - 4.4.1 Bus Arbitration 237
- 4.5 Buses 240
  - 4.5.1 Synchronous Bus 241
  - 4.5.2 Asynchronous Bus 244
  - 4.5.3 Discussion 247
- 4.6 Interface Circuits 248
  - 4.6.1 Parallel Port 248
  - 4.6.2 Serial Port 257
- 4.7 Standard I/O Interfaces 259
  - 4.7.1 Peripheral Component Interconnect (PCI) Bus 261
  - 4.7.2 SCSI Bus 266
  - 4.7.3 Universal Serial Bus (USB) 272
- 4.8 Concluding Remarks 283
  - Problems 283
  - References 289

## Chapter 5

### THE MEMORY SYSTEM 291

- 5.1 Some Basic Concepts 292
- 5.2 Semiconductor RAM Memories 295
  - 5.2.1 Internal Organization of Memory Chips 295
  - 5.2.2 Static Memories 297
  - 5.2.3 Asynchronous Drains 299
  - 5.2.4 Synchronous DRAMs 302
  - 5.2.5 Structure of Larger Memories 305
  - 5.2.6 Memory System Considerations 307
  - 5.2.7 Rambus Memory 308
- 5.3 Read-Only Memories 309
  - 5.3.1 ROM 310
  - 5.3.2 PROM 311
  - 5.3.3 EPROM 311
  - 5.3.4 EEPROM 311
  - 5.3.5 Flash Memory 312
- 5.4 Speed, Size, and Cost 313
- 5.5 Cache Memories 314
  - 5.5.1 Mapping Functions 316
  - 5.5.2 Replacement Algorithms 321
  - 5.5.3 Example of Mapping Techniques 322
  - 5.5.4 Examples of Caches in Commercial Processors 325
- 5.6 Performance Considerations 329
  - 5.6.1 Interleaving 330
  - 5.6.2 Hit Rate and Miss Penalty 332
  - 5.6.3 Caches on the Processor Chip 334
  - 5.6.4 Other Enhancements 335
- 5.7 Virtual Memories 337
  - 5.7.1 Address Translation 339
- 5.8 Memory Management Requirements 343
- 5.9 Secondary Storage 344
  - 5.9.1 Magnetic Hard Disks 344
  - 5.9.2 Optical Disks 352
  - 5.9.3 Magnetic Tape Systems 358
- 5.10 Concluding Remarks 359
  - Problems 360
  - References 366

## Chapter 6

### ARITHMETIC 367

- 6.1 Addition and Subtraction of Signed Numbers 368
  - 6.1.1 Addition/Subtraction Logic Unit 369
- 6.2 Design of Fast Adders 371
  - 6.2.1 Carry-Lookahead Addition 372
- 6.3 Multiplication of Positive Numbers 376
- 6.4 Signed-Operand Multiplication 380
  - 6.4.1 Booth Algorithm 380
- 6.5 Fast Multiplication 383
  - 6.5.1 Bit-Pair Recoding of Multipliers 384
  - 6.5.2 Carry-Save Addition of Summands 385
- 6.6 Integer Division 390
- 6.7 Floating-Point Numbers and Operations 393
  - 6.7.1 IEEE Standard for Floating-Point Numbers 394
  - 6.7.2 Arithmetic Operations on Floating-Point Numbers 398
  - 6.7.3 Guard Bits and Truncation 399

- 6.7.4 Implementing Floating-Point Operations 400
- 6.8 Concluding Remarks 403
  - Problems 403
  - References 410
- Chapter 7**
- BASIC PROCESSING UNIT 411**
- 7.1 Some Fundamental Concepts 412
  - 7.1.1 Register Transfers 415
  - 7.1.2 Performing an Arithmetic or Logic Operation 415
  - 7.1.3 Fetching a Word from Memory 418
  - 7.1.4 Storing a Word in Memory 420
- 7.2 Execution of a Complete Instruction 421
  - 7.2.1 Branch Instructions 422
- 7.3 Multiple-Bus Organization 423
- 7.4 Hardwired Control 425
  - 7.4.1 A Complete Processor 428
- 7.5 Microprogrammed Control 429
  - 7.5.1 Microinstructions 432
  - 7.5.2 Microprogram Sequencing 435
  - 7.5.3 Wide-Branch Addressing 437
  - 7.5.4 Microinstructions with Next-Address Field 440
  - 7.5.5 Prefetching Microinstructions 443
  - 7.5.6 Emulation 443
- 7.6 Concluding Remarks 445
  - Problems 446
- Chapter 8**
- PIPELINING 453**
- 8.1 Basic Concepts 454
  - 8.1.1 Role of Cache Memory 456
  - 8.1.2 Pipeline Performance 458
- 8.2 Data Hazards 461
  - 8.2.1 Operand Forwarding 462
  - 8.2.2 Handling Data Hazards in Software 464
  - 8.2.3 Side Effects 464
- 8.3 Instruction Hazards 465
  - 8.3.1 Unconditional Branches 466
  - 8.3.2 Conditional Branches and Branch Prediction 470
- 8.4 Influence on Instruction Sets 476
  - 8.4.1 Addressing Modes 476
  - 8.4.2 Condition Codes 478
- 8.5 Datapath and Control Considerations 479
- 8.6 Superscalar Operation 481
  - 8.6.1 Out-of-Order Execution 483
  - 8.6.2 Execution Completion 485
  - 8.6.3 Dispatch Operation 486
- 8.7 UltraSPARC II EXAMPLE 486
  - 8.7.1 SPARC Architecture 487
  - 8.7.2 UltraSPARC II 493
  - 8.7.3 Pipeline Structure 493
- 8.8 Performance Considerations 503
  - 8.8.1 Effect of Instruction Hazards 504
  - 8.8.2 Number of Pipeline Stages 505
- 8.9 Concluding Remarks 506
  - Problems 506
  - Reference 509
- Chapter 9**
- EMBEDDED SYSTEMS 511**
- 9.1 Examples of Embedded Systems 512
  - 9.1.1 Microwave Oven 512
  - 9.1.2 Digital Camera 514
  - 9.1.3 Home Telemetry 516
- 9.2 Processor Chips for Embedded Applications 517
- 9.3 A Simple Microcontroller 518
  - 9.3.1 Parallel I/O Ports 518
  - 9.3.2 Serial I/O Interface 521
  - 9.3.3 Counter/Timer 523
  - 9.3.4 Interrupt Control Mechanism 525
- 9.4 Programming Considerations 525
  - 9.4.1 Polling Approach 526
  - 9.4.2 Interrupt Approach 529
- 9.5 I/O Device Timing Constraints 531
  - 9.5.1 C Program for Transfer via a Circular Buffer 533
  - 9.5.2 Assembly Language Program for Transfer via a Circular Buffer 534
- 9.6 Reaction Timer — An Example 535
  - 9.6.1 C Program for the Reaction Timer 537
  - 9.6.2 Assembly Language Program for the Reaction Timer 537
  - 9.6.3 Final Comments 541
- 9.7 Embedded Processor Families 541
  - 9.7.1 Microcontrollers Based on the Intel 8051 542
  - 9.7.2 Motorola Microcontrollers 542
  - 9.7.3 ARM Microcontrollers 543

- 9.8 Design Issues 544
- 9.9 System-on-a-Chip 546
  - 9.9.1 FPGA Implementation 547
- 9.10 Concluding Remarks 549
  - Problems 550
  - References 552

## Chapter 10

### COMPUTER PERIPHERALS 553

- 10.1 Input Devices 554
  - 10.1.1 Keyboard 554
  - 10.1.2 Mouse 555
  - 10.1.3 Trackball, Joystick, and Touchpad 556
  - 10.1.4 Scanners 557
- 10.2 Output Devices 558
  - 10.2.1 Video Displays 558
  - 10.2.2 Flat-Panel Displays 559
  - 10.2.3 Printers 560
  - 10.2.4 Graphics Accelerators 561
- 10.3 Serial Communication Links 563
  - 10.3.1 Asynchronous Transmission 566
  - 10.3.2 Synchronous Transmission 568
  - 10.3.3 Standard Communications Interfaces 571
- 10.4 Concluding Remarks 574
  - Problems 575

## Chapter 11

### PROCESSOR FAMILIES 577

- 11.1 The ARM Family 579
  - 11.1.1 The Thumb Instruction Set 579
  - 11.1.2 Processor and CPU Cores 580
- 11.2 The Motorola 680X0 and ColdFire Families 582
  - 11.2.1 68020 Processor 582
  - 11.2.2 Enhancements in 68030 and 68040 Processors 584
  - 11.2.3 68060 Processor 585
  - 11.2.4 The ColdFire Family 585
- 11.3 The Intel IA-32 Family 585
  - 11.3.1 IA-32 Memory Segmentation 586
  - 11.3.2 Sixteen-Bit Mode 588
  - 11.3.3 80386 and 80486 Processors 588
  - 11.3.4 Pentium Processor 589
  - 11.3.5 Pentium Pro Processor 589
  - 11.3.6 Pentium II and III Processors 590

- 11.3.7 Pentium 4 Processor 590
- 11.3.8 Advanced Micro Devices IA-32 Processors 591
- 11.4 The PowerPC Family 591
  - 11.4.1 Register Set 591
  - 11.4.2 Memory Addressing Modes 592
  - 11.4.3 Instructions 592
  - 11.4.4 PowerPC Processors 592
- 11.5 The Sun Microsystems SPARC Family 594
- 11.6 The Compaq Alpha Family 596
  - 11.6.1 Instruction and Addressing Mode Formats 596
  - 11.6.2 Alpha 21064 Processor 597
  - 11.6.3 Alpha 21164 Processor 597
  - 11.6.4 Alpha 21264 Processor 597
- 11.7 The Intel IA-64 Family 598
  - 11.7.1 Instruction Bundles 598
  - 11.7.2 Conditional Execution 598
  - 11.7.3 Speculative Loads 600
  - 11.7.4 Registers and the Register Stack 600
  - 11.7.5 Itanium Processor 602
- 11.8 A Stack Processor 603
  - 11.8.1 Stack Structure 604
  - 11.8.2 Stack Instructions 606
  - 11.8.3 Hardware Registers in the Stack 610
- 11.9 Concluding Remarks 612
  - Problems 612
  - References 614

## Chapter 12

### LARGE COMPUTER SYSTEMS 617

- 12.1 Forms of Parallel Processing 619
  - 12.1.1 Classification of Parallel Structures 619
- 12.2 Array Processors 620
- 12.3 The Structure of General-Purpose Multiprocessors 622
- 12.4 Interconnection Networks 624
  - 12.4.1 Single Bus 624
  - 12.4.2 Crossbar Networks 625
  - 12.4.3 Multistage Networks 626
  - 12.4.4 Hypercube Networks 628
  - 12.4.5 Mesh Networks 630
  - 12.4.6 Tree Networks 630
  - 12.4.7 Ring Networks 631
  - 12.4.8 Practical Considerations 632
  - 12.4.9 Mixed Topology Networks 636
  - 12.4.10 Symmetric Multiprocessors 636

- 12.5 Memory Organization in Multiprocessors 637
- 12.6 Program Parallelism and Shared Variables 638
  - 12.6.1 Accessing Shared Variables 640
  - 12.6.2 Cache Coherence 641
  - 12.6.3 Need for Locking and Cache Coherence 645
- 12.7 Multicomputers 645
  - 12.7.1 Local Area Networks 646
  - 12.7.2 Ethernet (CSMA/CD) Bus 646
  - 12.7.3 Token Ring 647
  - 12.7.4 Network of Workstations 647
- 12.8 Programmer's View of Shared Memory and Message Passing 648
  - 12.8.1 Shared Memory Case 648
  - 12.8.2 Message-Passing Case 651
- 12.9 Performance Considerations 653
  - 12.9.1 Amdahl's Law 654
  - 12.9.2 Performance Indicators 656
- 12.10 Concluding Remarks 656
  - Problems 657
  - References 660

## APPENDIX A: LOGIC CIRCUITS 661

- A.1 Basic Logic Functions 662
  - A.1.1 Electronic Logic Gates 665
- A.2 Synthesis of Logic Functions 666
- A.3 Minimization of Logic Expressions 668
  - A.3.1 Minimization Using Karnaugh Maps 671
  - A.3.2 Don't-Care Conditions 674
- A.4 Synthesis with NAND and NOR Gates 674
- A.5 Practical Implementation of Logic Gates 678
  - A.5.1 CMOS Circuits 681
  - A.5.2 Propagation Delay 686
  - A.5.3 Fan-In and Fan-Out Constraints 687
  - A.5.4 Tri-state Buffers 687
  - A.5.5 Integrated Circuit Packages 688
- A.6 Flip-Flops 690
  - A.6.1 Gated Latches 690
  - A.6.2 Master-Slave Flip-Flop 694
  - A.6.3 Edge Triggering 694
  - A.6.4 T Flip-Flop 697

- A.6.5 JK Flip-Flop 697
- A.6.6 Flip-Flops with Preset and Clear 698
- A.7 Registers and Shift Registers 699
- A.8 Counters 702
- A.9 Decoders 703
- A.10 Multiplexers 705
- A.11 Programmable Logic Devices (PLDs) 705
  - A.11.1 Programmable Logic Array (PLA) 707
  - A.11.2 Programmable Array Logic (PAL) 710
  - A.11.3 Complex Programmable Logic Devices (CPLDs) 711
- A.12 Field-Programmable Gate Arrays 712
- A.13 Sequential Circuits 714
  - A.13.1 An Example of an Up/Down Counter 714
  - A.13.2 Timing Diagrams 718
  - A.13.3 The Finite State Machine Model 719
  - A.13.4 Synthesis of Finite State Machines 720
- A.14 Concluding Remarks 724
  - Problems 724
  - References 731

## APPENDIX B: ARM INSTRUCTION SET 733

- B.1 Instruction Encoding 734
  - B.1.1 Arithmetic and Logic Instructions 734
  - B.1.2 Memory Load and Store Instructions 741
  - B.1.3 Block Load and Store Instructions 744
  - B.1.4 Branch and Branch with Link Instructions 747
  - B.1.5 Machine Control Instructions 747
- B.2 Other ARM Instructions 750
  - B.2.1 Coprocessor Instructions 750
  - B.2.2 Versions v4 and v5 Instructions 750
- B.3 Programming Experiments 750

## APPENDIX C: MOTOROLA 68000 INSTRUCTION SET 751

## **APPENDIX D: INTEL IA-32 INSTRUCTION SET 769**

- D.1 Instruction Encoding 770
  - D.1.1 Addressing Modes 772
- D.2 Basic Instructions 773
  - D.2.1 Conditional Jump Instructions 782
  - D.2.2 Unconditional Jump Instructions 782
- D.3 Prefix Bytes 782
- D.4 Other Instructions 783
  - D.4.1 String Instructions 783

- D.4.2 Floating-Point, MMX, and SSE  
Instructions 784

- D.5 Sixteen-Bit Operation 785
- D.6 Programming Experiments 785

## **APPENDIX E: CHARACTER CODES AND NUMBER CONVERSION 789**

- E.1 Character Codes 790
- E.2 Decimal-to-Binary Conversion 793

## **INDEX 795**

# PREFACE

This book is intended for use in a first-level course on computer organization in electrical engineering, computer engineering, and computer science curricula. The book is self-contained, assuming only that the reader has a basic knowledge of computer programming in a high-level language. Many students who study computer organization will have had an introductory course on digital logic circuits. Therefore, this subject is not covered in the main body of the book. However, we have provided an extensive appendix on logic circuits for those students who need it.

The book reflects our experience in teaching computer organization to three distinct groups of undergraduates: electrical and computer engineering undergraduates, computer science specialists, and engineering science undergraduates. We have always approached the teaching of courses in this area from a practical point of view. Thus, a key consideration in shaping the contents of the book has been to illustrate the principles of computer organization using examples drawn from commercially available computers. Our main examples are based on the following processors: ARM, Motorola 680X0, Intel Pentium, and Sun UltraSPARC.

It is important to recognize that digital system design is not a straightforward process of applying optimal design algorithms. Many design decisions are based largely on heuristic judgment and experience. They involve cost/performance and hardware/software tradeoffs over a range of alternatives. It is our goal to convey these notions to the reader.

We have endeavored to provide sufficient details to encourage the student to dig beyond the surface when dealing with ideas that seem to be intuitively obvious. We believe that this is best accomplished by giving real examples that are adequately documented. Block diagrams are a powerful means of describing organizational features of a computer. However, they can easily lead to an oversimplified view of the problems involved. Hence, they must be accompanied by the details of implementation alternatives.

The book is aimed at a one-semester course in engineering or computer science programs. It is suitable for both hardware- and software-oriented students. Even though the emphasis is on hardware, we have addressed a number of software issues, including basic aspects of compilers and operating systems related to instruction execution performance, coordination of parallel operations at the system level, and real-time applications. An understanding of hardware/software interaction and tradeoffs is necessary for computer specialists.

## THE SCOPE OF THE BOOK

We now review the topics covered in sequence, chapter by chapter. The first eight chapters cover the basic principles of computer organization, operation, and performance.

The remaining four chapters deal with embedded systems, peripheral devices, processor family evolution patterns, and large computer systems.

Chapter 1 provides an overview of computer hardware and software and informally introduces terms that are dealt with in more depth in the remainder of the book. This chapter discusses the basic functional units and the ways they are interconnected to form a complete computer system. The role of system software is introduced and basic aspects of performance evaluation are discussed. A brief treatment of the history of computer development is also provided.

Chapter 2 gives a methodical treatment of machine instructions, addressing techniques, and instruction sequencing. Basic aspects of 2's-complement arithmetic are introduced to facilitate the discussion of the generation of effective addresses. Program examples at the machine instruction level, expressed in a generic assembly language, are used to discuss loops, subroutines, simple input-output programming, sorting, and linked list operations.

Chapter 3 illustrates implementation of the concepts introduced in Chapter 2 on three commercial processors—ARM, 68000, and Pentium. The ARM processor illustrates the RISC design style, the 68000 has an easy-to-teach CISC design, while the Pentium represents the most successful commercial design that combines the elements of both the CISC and RISC styles. The material is organized into three independent and complete parts. Each part includes all of the examples from Chapter 2 implemented in the context of the specific processor. It is sufficient to cover only one of the three parts to provide the continuity needed to follow the rest of the book. If laboratory experiments using one of the three processors are associated with the course, the relevant part of Chapter 3 can be covered in parallel with Chapter 2.

Input-output organization is developed in Chapter 4. The basics of I/O data transfer synchronization are presented, and a series of increasingly complex I/O structures are explained. Interrupts and direct-memory access methods are described in detail, including a discussion of the role of software interrupts in operating systems. Bus protocols and standards are also presented, with the PCI, SCSI, and USB standards being used as representative commercial examples.

Semiconductor memories, including SDRAM, Rambus, and Flash memory implementations, are discussed in Chapter 5. Caches and multiple-module memory systems are explained as ways for increasing main memory bandwidth. Caches are discussed in some detail, including performance modeling. Virtual-memory systems, memory management, and rapid address translation techniques are also presented. Magnetic and optical disks are discussed as components in the memory hierarchy.

Chapter 6 treats the arithmetic unit of a computer. Logic design for fixed-point add, subtract, multiply, and divide hardware, operating on 2's-complement numbers, is described. Lookahead adders and high-speed multipliers are explained, including descriptions of the Booth multiplier recoding and carry-save addition techniques. Floating-point number representation and operations, in the context of the IEEE Standard, are presented.

Chapter 7 begins with a register-transfer-level treatment of the implementation of instruction fetching and execution in a processor. This is followed by a discussion of processor implementation by both hardwired and microprogrammed control.

















































































































































