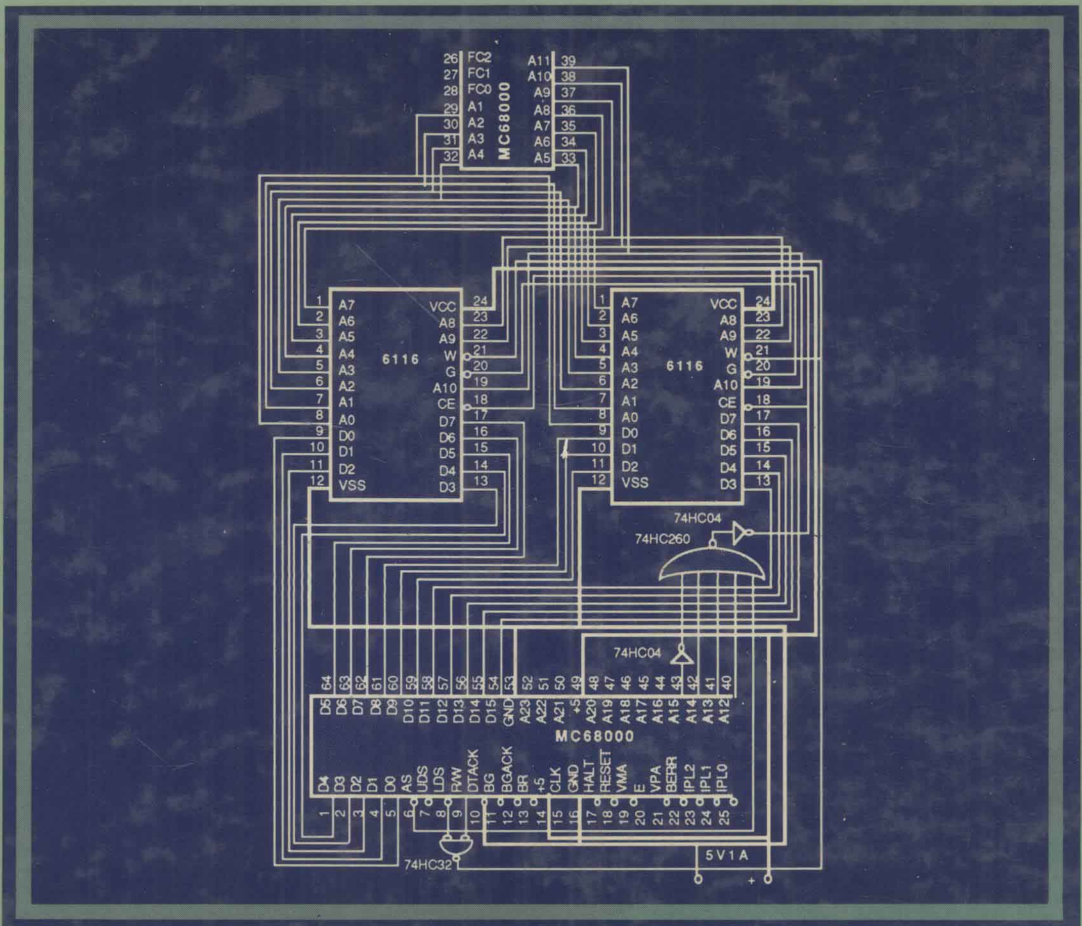
16-AND 32-BIT MICROCOMPUTER INTERFACING

Programming Examples in C and M68000 Family Assembly Language



G.J. LIPOVSKI

16- and 32-Bit Microcomputer Interfacing: Program Examples in C and M68000 Family Assembly Language

G. J. LIPOVSKI

University of Texas
& Naval Postgraduate School

Prentice Hall
Englewood Cliffs, New Jersey 07632

Lipovski, G. Jack.
16- and 32-bit microcomputer interfacing: program examples in C and M68000 family assembly / G.J. Lipovski
p. cm.
Includes index.
ISBN 0-13-812074-9
1. Microcomputers. 2. Computer interfaces. 3. Motorola 68000 (Microprocessor) 4. Motorola 68030 (Microprocessor) I. Title.
TK7887.5.L55 1990
621.39'81—dc20 89-35675
CIP

Editorial/production supervision: *Kerry Reardon*
Cover design: *Wanda Lubelska*
Manufacturing buyer: *Denise Duggan*

© 1990 by Prentice-Hall, Inc.
A Division of Simon & Schuster
Englewood Cliffs, New Jersey 07632

The publisher offers discounts on this book when ordered in bulk quantities. For more information, write:

Special Sales/College Marketing
College Technical and Reference Division
Prentice Hall
Englewood Cliffs, New Jersey 07632

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-812074-9

PRENTICE-HALL INTERNATIONAL (UK) LIMITED, *London*
PRENTICE-HALL OF AUSTRALIA PTY. LIMITED, *Sydney*
PRENTICE-HALL CANADA INC., *Toronto*
PRENTICE-HALL HISPANOAMERICANA, S.A., *Mexico*
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*
PRENTICE-HALL OF JAPAN, INC., *Tokyo*
SIMON & SCHUSTER ASIA PTE. LTD., *Singapore*
EDITORA PRENTICE-HALL DO BRASIL, LTDA., *Rio de Janeiro*

16- and 32-Bit Microcomputer Interfacing: Program Examples in C and M68000 Family Assembly Language

Dedicated to my father

Joseph Lipovski

此为试读，需要完整PDF请访问： www.ertongbook.com

Preface

Microcomputer interfacing is a significant part of computer scientists' and computer engineers' backgrounds. They should understand the fundamental and pervasive notion of the hardware-software tradeoff. They should be familiar with operating systems and their handling of input/output. Finally, because they might have to interface a microcomputer to external equipment, to monitor or control external processes, some computer scientists or computer engineers should be capable of doing this. Microcomputer interfacing - the subject of this book - covers these concepts.

Of fundamental importance, any computer scientist or computer engineer should understand what can be done effectively in hardware, what is easily done in software, and what alternative approaches in both are worth considering in a design. This understanding of the hardware-software tradeoff is missing in purely hardware electrical engineering courses or purely software computer science courses, but is necessary for the intelligent design of either hardware or software systems. While an understanding of the hardware-software tradeoff pervades many aspects of computer science and engineering, and appears in many courses therein, nowhere is it so clearly seen as it is in interfacing. A course on interfacing is fundamentally a course on hardware-software tradeoffs.

Of secondary but still major importance, most computer scientists or computer engineers must understand operating systems. Many operating systems courses cover time-sharing and performance modeling, which are important in operating systems used in large mainframe computers. Personal computers may become more important than mainframe computers as time goes on. Operating systems in personal computers mainly support high-level languages. Their input/output infrastructure is one of the most important support functions they provide for high-level languages. A course on interfacing is a course on the infrastructure for high-level language input/output.

A third rationale for this course is that some computer scientists or computer engineers have to interface a microcomputer to external equipment. They provide a basic service and command a premium salary. They need to have a thorough knowledge of the principles of interfacing. A comprehensive course should be provided for them.

However, one of the least important aspects of microcomputer interfacing is the detailed understanding of a particular machine. While details are important for you to get hands-on interfacing experience, these details can easily swamp you. Even though we cover the popular 68000/68030, you are likely to interface some other microcomputer. You must keep details in perspective; they are only a means to an end. Another common but unimportant aspect of interfacing is the use of assembler language. High-level languages like C are more efficient and are being more frequently used in industry.

While some books on interfacing put great emphasis on a particular machine, on special techniques used in it, and on assembler language programming, this book puts greater emphasis on hardware-software tradeoffs, operating system and high-level language interfacing, and the principles of interfacing. Because of this, we believe this book offers a better perspective than other interfacing textbooks. Its approach to interfacing provides a significant part of the computer scientists' and computer engineers' background.

Acknowledgments

The author would like to express his deepest gratitude to everyone who contributed to the development of this book. In addition to a number of faculty members, students, and colleagues in industry who helped in no small part, special thanks are due to David Mielke, J. A. Hernandez Jr., Robert Steigerwald, Eugene Gugliemo, Anthony J. Davis, Matthew J. Kohler, and D. Antonopoulis for finding errors in the manuscript. This text was prepared and run off using a Macintosh and LaserWriter, running WriteNow. The cooperation of Motorola in providing chips and information made this book's development a real pleasure. I am pleased to observe that the Motorola 68000, 68020, and 68030 are incredibly powerful components and vehicles for teaching a considerable range of concepts.

About the Author

G. Jack Lipovski has taught electrical engineering and computer science at the University of Texas since 1976, and is currently on leave, occupying the Grace Hopper Chair of Computer Science at the Naval Postgraduate School. He is a computer architect internationally recognized for his design of the pioneering database computer, CASSM, and the parallel computer, TRAC. His expertise in microcomputers is also internationally recognized by his being a past director of Euromicro and an editor of IEEE Micro. Dr. Lipovski has published more than 70 papers, largely in the proceedings of the annual symposium on computer architecture, the IEEE transactions on computers, and the national computer conference. He has authored five books and edited three. He has served as chairman of the IEEE Computer Society Technical Committee on Computer Architecture, member of the Computer Society Governing Board, and chairman of the Special Interest Group on Computer Architecture of the Association for Computer Machinery. He received his Ph.D. degree from the University of Illinois, 1969, and has taught at the University of Florida. He has consulted for Harris Semiconductor, designing a microcomputer, and for the Microelectronics and Computer Corporation, studying parallel computers. His current interests include parallel computing, database computer architectures, artificial intelligence computer architectures, and microcomputers.

Contents

Preface xvii

Acknowledgments xviii

About the Author xix

1 Microcomputer Architecture 1

1-1 An Introduction to the Microcomputer 2

1-1.1 Computer Architecture 2

1-1.2 The Instruction 8

1-1.3 Microcomputers 11

1-2 The Simplified 68000 Instruction Set 14

1-2.1 Addressing Modes for the Simplified 68000 15

1-2.2 The Instruction Set of the Simplified 68000 21

1-3 The 68000 and 68020/68030 Instruction Sets * 34

1-3.1 68000 and 68020/68030 Operand Sizes * 35

1-3.2 68000 and 68020/68030 Addressing Modes * 40

1-3.3 68000 and 68020/68030 Instructions * 44

1-4 Conclusions 50

Problems 52

2 Programming Microcomputers 57

2-1 Introduction to C 58

2-2 Data Structures 65

2-2.1 Assembler Directives 65

2-2.2 Indexable Data Structures 69

2-2.3 Sequential Data Structures 73

2-2.4 Directives and Structures with 8-Bit and 32-Bit Elements * 78

2-2.5 Special Addressing Modes in the 68000 and 68020/68030 * 80

2-3 Writing Clear Assembler Language Programs 81

2-3.1 Documentation 81

2-3.2 Programming Style 83

2-4 Subroutines and Arguments 88

2-4.1 Passing Parameters (Conceptual Level) 89

2-4.2 Assembler Language Parameter Passing Techniques 91

2-4.3 Assembler Language Calling and Returning Mechanisms 96

2-4.4 A Blueprint for Writing Assembler Language Subroutines 98

2-4.5 Assembler Language Subroutine Documentation 99

2-4.6 Some C Functions Using Character Strings * 102

2-5 Conclusions 104

Problems 105

3 Bus Hardware and Signals 109
3-1 Digital Hardware 110
3-1.1 Modules and Signals 110
3-1.2 Drivers, Registers, and Memories 113
3-2 Control Signals and Their Sequences in a Microprocessor 121
3-2.1 Data Transfers and Microprograms 121
3-2.2 The Fetch-Execute Cycle 126
3-3 Interfacing to the Bus 131
3-3.1 Address and Data Bus Timing 132
3-3.2 Address Decoding 139
3-3.3 The Design of a (2048,16) Memory for an MC68000 142
3-3.4 Dynamic Memory 145
3-3.5 Some 68000, 68020, and 68030 Memory Techniques * 148
3-3.6 Backplanes for Multiboard Computer Systems 154
3-4 Conclusions 157
Problems 159

4 Parallel and Serial Input-Output 166
4-1 Input-Output Architecture and Simple I/O Ports 167
4-1.1 Isolated and Memory-Mapped I/O 167
4-1.2 I/O Ports in an Address Space 169
4-1.3 Basic Input and Output Ports 170
4-1.4 Specific 68000 and 68020/68030 I/O Instructions * 177
4-1.5 Summary of I/O Device Design 179
4-2 Input-Output Software 180
4-2.1 I/O Transfers and Buffered Input and Output 180
4-2.2 Programmed Logic Control 183
4-2.3 Table and Linked List Interpreters 185
4-3 Programmable Parallel Input-Output Devices 192
4-3.1 Problems with LSI I/O and the Ritual 193
4-3.2 The M6821 Input-Output Registers 194
4-3.3 M68230 and M68901 Parallel Input-Output Devices 200
4-4 Input-Output Indirection 205
4-4.1 Indirect Input-Output 205
4-4.2 Synchronous Serial Input-Output 210
4-4.3 Coprocessors as Immediate I/O Devices * 215
4-4.4 Design Considerations With Regard to Indirection 224
4-5 An Example: An IC Tester 225
4-6 Conclusions 230
Problems 232

5 Interrupts and Alternatives 239
5-1 Synchronization in I/O Devices 240
5-1.1 Synchronization Principles of I/O Devices 240
5-1.2 The M6821 Interrupt Mechanism 243

5-1.3 Synchronization Mechanisms for the M68901 and M68230 Parallel Devices 246
5-2 Slow Synchronization Mechanisms 251
5-2.1 Real-Time Synchronization 251
5-2.2 Gadfly Synchronization 254
5-2.3 Single Interrupts 258
5-2.4 Polled Interrupts 264
5-2.5 Vectored Interrupts 269
5-2.6 Vectored Interrupts in the M68000 * 274
5-3 Fast Synchronization Mechanisms 280
5-3.1 Direct Memory Access, Context Switching, and Coprocessing 280
5-3.2 The M68020 Coprocessor I/O Device * 288
5-3.3 Shuttle, Indirect, and Time-Multiplexed Memories 289
5-4 Conclusions 291
Problems 293

6 Analog Interfacing 300
6-1 Input and Output Transducers 302
6-1.1 Positional Transducers 302
6-1.2 Radiant Energy Transducers 304
6-1.3 Other Transducers 306
6-2 Basic Analog Processing Components 307
6-2.1 Transistors and Silicon-Controlled Rectifiers 307
6-2.2 Basic Linear Integrated Circuits 310
6-2.3 Practical Linear Integrated Circuits 313
6-3 Signal Conditioning Using OP AMPs and Analog Switches 319
6-3.1 Filters 320
6-3.2 Selection of Inputs and Control of Gain 322
6-3.3 Nonlinear Amplification 325
6-4 A Keyboard and LED Display Module 327
6-4.1 Key Debouncing 327
6-4.2 Keyboard Scanning Techniques 330
6-4.3 Displays 333
6-5 Converters 338
6-5.1 Digital-to-Analog Converters 339
6-5.2 Analog-to-Digital Converters 343
6-6 Data Acquisition Systems 349
6-6.1 Basic Operation of a Data Acquisition System 349
6-6.2 The MC145040 Chip 351
6-6.3 Applications of Data Acquisition Systems in Control Systems 352
6-7 Conclusions 355
Problems 357

7 Counters and Timers 364
7-1 Counter-Timer Chip Subsystems 365

此为试读，需要完整PDF请访问：www.ertongbook.com

7-1.1 Overview of the M6840 365
7-1.2 Overview of Counter/Timers in the M68230 and M68901 369
7-1.3 68000 and 68020/68030 Instructions for Counter/Timer Chips * 373
7-2 Signal Generation 375
7-2.1 The Square Wave Generator 375
7-2.2 The Pulse Generator 385
7-2.3 Real-Time Clock 388
7-2.4 Generation Modes of the M6840 394
7-3 Frequency Analog Measurement 395
7-3.1 Frequency Measurement 396
7-3.2 Period Measurement 400
7-3.3 Pulse Width Measurement 402
7-3.4 Signal Measurement Modes of the M6840 405
7-3.5 Counter/Timer Modes of the M68230 and M68901 409
7-4 Conclusions 410
Problems 412

8 Communication Systems **418**
8-1 Communications Principles 419
8-2 Signal Transmission 422
8-2.1 Voltage and Current Linkages 422
8-2.2 Frequency Shift Keyed Links Using Modems 426
8-2.3 Optical Data Transmission 428
8-3 UART Link Protocol 430
8-3.1 UART Transmission and Reception by Software 430
8-3.2 The UART 435
8-3.3 The M6850 438
8-3.4 The M14469 442
8-3.5 The Serial Communication Interface Subsystems in the M68681 and M68901 446
8-4 Other Protocols 451
8-4.1 Synchronous Bit-Oriented Protocols 451
8-4.2 IEEE-488 Bus Standard 455
8-4.3 The Smart Computer System Interface (SCSI) 457
8-5 Conclusions 458
Problems 460

9 Storage and Display Systems **465**
9-1 Storage Systems 466
9-1.1 The 5 1/4" Floppy Disk Format 466
9-1.2 The Western Digital 1772 Floppy Disk Controller 470
9-2 Display Systems 479
9-2.1 Raster Display Format 480
9-2.2 The Signetics 2670-73 CRT Controller Chip Set 481
9-3 Conclusions 496
Problems 497

10 Operating System and Device Drivers **500**
10-1 What Is an Operating System? 500
10-1.1 Three Views of an Operating System 501
10-1.2 Functions and Features of an Operating System (OS-9) 503
10-1.3 Compontents of an Operating System (OS-9) 509
10-1.4 Environment of an I/O Read System Call 511
10-2 A Device Driver 514
10-2.1 Initialization 517
10-2.2 Read Subroutine 520
10-4.3 Interrupt Handler 523
10-2.4 Other Parts 528
10-3 Conclusions 531
Problems 532

Appendix A. A Downloader For C and Assmembler Language 537
Index 545

List of Figures

Figure Title **Page**

1-1. Analogy to the von Neumann Computer. 4
1-2. Registers and Memory in a 68000 (Simplified). 5
1-3. Microprocessor, Microcomputer, and System. 13
1-4. Coding of MOVE Instructions. 21
1-5. Coathanger Diagrams. 28
1-6. A Flow Chart of a Decision Tree. 31
1-7. Flow Chart of the DBcc Loop. 33
1-8. Memory in the 68000. 35
1-9. Registers in the 68000. 36
1-10. Registers in the 68020 and 68030 (Processor Module). 37
1-11. Memory in the 68020/68030. 38
1-12. Memory Spaces in the 68000 or 68020/68030. 39
1-13. Index Addressing in the 68020/68030. 42

2-1. Conditional Statements. 61
2-2. Case Statements. 62
2-3. Loop Statements. 63
2-4. A Huffman Coding Tree. 75
2-5. A Program Segment. 83
2-6. Nested Program Segments. 85
2-7. Coathanger Diagram Showing a Bug Using Global Variables. 93

3-1. Voltage Waveforms, Signals, and Variables. 110
3-2. Some Common Gates. 114
3-3. Logic Diagrams for a Popular Driver and Register. 115
3-4. 16R4 PAL Used in Microcomputer Designs. 120
3-5. Bus with Control Signals. 123
3-6. Connections to the MC68000. 126
3-7. MC68000 Microprocessor. 128
3-8. MC68000 Memory Cycle Sequences. 131
3-9. Some Timing Relationships. 132
3-10. Timing Relationships for the MC68000L8. 133
3-11. Timing Requirements of the Motorola MCM65116-20. 135
3-12. Analysis of Memory Timing. 137
3-13. Wait Cycle Control Logic. 139
3-14. A Memory Map. 140
3-15. A Programmable Decoder. 141
3-16. An MCM6116 Memory System. 143
3-17. Practical Memory System. 144
3-18. A Dynamic Memory Chip. 145
3-19. HM511000P10 1 Mbit Dynamic Memory Timing. 147
3-20. A Fully Decoded MCM6116 Memory System. 149
3-21. Timing Relationships for the MC68020R16 and MC68030R20. 153

3-22. Bus for Problems 3-7 to 3-9. 160
3-23. Timing Characteristics for Several 2114s. 161
3-24. Common Gates Used for Decoders. 163

4-1. Addressing I/O Ports. 170
4-2. Logic Diagrams for Basic I/O Ports. 171
4-3. Logic Diagram of a Readable Output Port. 174
4-4. Address Output Techniques. 176
4-5. A 2-Byte Output Port. 178
4-6. Traffic Light. 183
4-7. Mealy Sequential Machine. 188
4-8. A Linked-List Structure. 190
4-9. The M6821. 196
4-10. 68000 Compatible I/O Devices with Parallel I/O. 201
4-11. Block Diagrams of the 68901 and 68230. 203
4-12. 6818A Time-of-Day Chip. 206
4-13. Simple Serial Input/Output Ports. 211
4-14. Configurations of Simple Serial Input/Output Registers. 212
4-15. Flow Chart for Series Serial Data Output. 213
4-16. Coprocessor Instructions. 216
4-17. Coprocessor Port Address Map. 217
4-18. Coprocessor Response. 218
4-19. Connecting the MC68881 Floating-Point Coprocessor. 220
4-20. Execution of Floating-Point Coprocessor Instructions. 222
4-21. A Logic Diagram for a Chip Tester. 228

5-1. Paper Tape. 241
5-2. State Diagram for I/O Devices. 242
5-3. M6821 Device A Interrupt/Output Mechanisms. 244
5-4. Synchronization Mechanisms for the M68901 Parallel Devices. 246
5-5. Synchronization Mechanisms for the M68230 Parallel Devices. 248
5-6. Flow Chart for Real-Time Input. 252
5-7. Connections to a Paper Tape Reader. 255
5-8. Flow Chart for Gadfly Input. 255
5-9. Interrupt Request Path. 259
5-10. Multiple Interrupt Sources. 265
5-11. Flow Chart for Interrupt Polling. 265
5-12. External Vectored Interrupts Using the M68901. 272
5-13. State Saved on Stack when Processing Exceptions. 275
5-14. Reset Hardware. 277
5-15. Connecting an M68440 to the M68000 Bus. 281
5-16. Channel 0 of the M68440. 282
5-17. DMA System. 284
5-18. Connections for Context Switching. 286
5-19. A Coprocessor Input Device. 288
5-20. Fast Synchronization Mechanisms Using Memory Organizations. 290

6-1. Waveforms of Alternating Voltage Signals. 300
6-2. Position Transducers. 303
6-3. Power Output Devices. 308
6-4. Triac Control Techniques. 309
6-5. Operational Amplifier Circuits. 311
6-6. Analog Switches. 313
6-7. A Practical Voltage Follower. 317
6-8. Practical Analog Switches. 318
6-9. 555 Timer. 319
6-10. Some Filters. 321
6-11. Selecting Inputs for a Stereo Preamplifier. 322
6-12. Computer Control of Amplification and Filtering. 323
6-13. Nonlinear Signal Conditioning. 326
6-14. Contact Bounce and Its Prevention. 328
6-15. Keyboard Structures. 332
6-16. Seven-Segment Displays. 335
6-17. Special Serial Interface Chips for LED and LCD Displays. 337
6-18. D-to-A Converters. 341
6-19. The MC144110. 343
6-20. A-to-D Conversion Techniques. 344
6-21. Frequency-to-Voltage Conversion. 347
6-22. Voltage-to-Frequency Conversion. 348
6-23. Data Acquisition System. 350
6-24. Serial Interfaced A-to-D Converter. 348
6-25. The Spline Technique. 353
6-26. Feedback Control. 353
6-27. Some OP AMP Circuits. 359

7-1. The M6840. 366
7-2. The Basic Counter Device. 369
7-3. The M68230 Counter/Timer Subsystem. 370
7-4. The M68901 Counter/Timer Subsystem. 372
7-5. The M68901 Counter/Timer A. 373
7-6. Phase-Locked Loop. 377
7-7. An M6840 Gated Square Wave Generator. 379
7-8. An M68230 Square Wave Generator. 380
7-9. An M68901 Square Wave Generator. 381
7-10. A Touch-Tone Generator. 384
7-11. An M6840 Single-Shot Pulse Generator. 386
7-12. An M6840 Real-Time Clock. 389
7-13. An M68230 Real-Time Clock. 391
7-14. An M68901 Real-Time Clock. 392
7-15. An M6840 Event Counter or a Frequency Counter. 397
7-16. An M68230 Event Counter or a Frequency Counter. 398
7-17. An M6840 Period or Pulse Width Timer. 400
7-18. An M68230 Pulse Width Timer. 402

7-19. An M68901 Pulse Width Timer. 404
7-20. State Diagrams for Period and Pulse Width Measurements. 407

8-1. Peer-to-Peer Communication in Different Levels. 419
8-2. Drivers and Receivers. 423
8-3. Originating a Call on a Modem. 428
8-4. A GPC Linkage for Optical Communication. 430
8-5. Frame Format for UART Signals. 431
8-6. Block Diagram of a UART (IM6403). 436
8-7. Transmitter Signals. 437
8-8. The M6850. 439
8-9. Output of a Test Program. 441
8-10. The M14469. 443
8-11. MC68901 Serial Communication Subsystem. 447
8-12. The MC68681 DUART. 449
8-13. Synchronous Formats. 452
8-14. IEEE-488 Bus Handshaking Cycle. 455
8-15. SCSI Timing. 457

9-1. Bit and Byte Storage for FM and MFM Encoding of 5 1/4" Floppy Disks. 467
9-2. Organization of Sectors and Tracks on a Disk Surface. 468
9-3. A Special Byte. 469
9-4. Floppy Disk Controller. 471
9-5. Programmer's View of the WD1772. 472
9-6. The Raster Scan Display Used in Television. 479
9-7. The Composite Video Signal. 480
9-8. Five Approaches to CRT System Design. 483
9-9. Characteristics of a CRT Display. 485
9-10. A CRT Display System. 488

10-1. Memory Map for OS-9. 504
10-2. Data Storage for OS-9. 505
10-3. Module Relationships for a Terminal in OS-9. 509
10-4. I$READ Call to a Terminal in OS-9. 512
10-5. Data Used in the ACIA Driver. 513
10-6. Subroutines and Vector of Addresses in a Device Driver. 517
10-7. Flow Chart for the Read Subroutine. 522
10-8. Flow Chart for the Receive Interrupt Handler Subroutine. 527

此为试读，需要完整PDF请访问：www.ertongbook.com