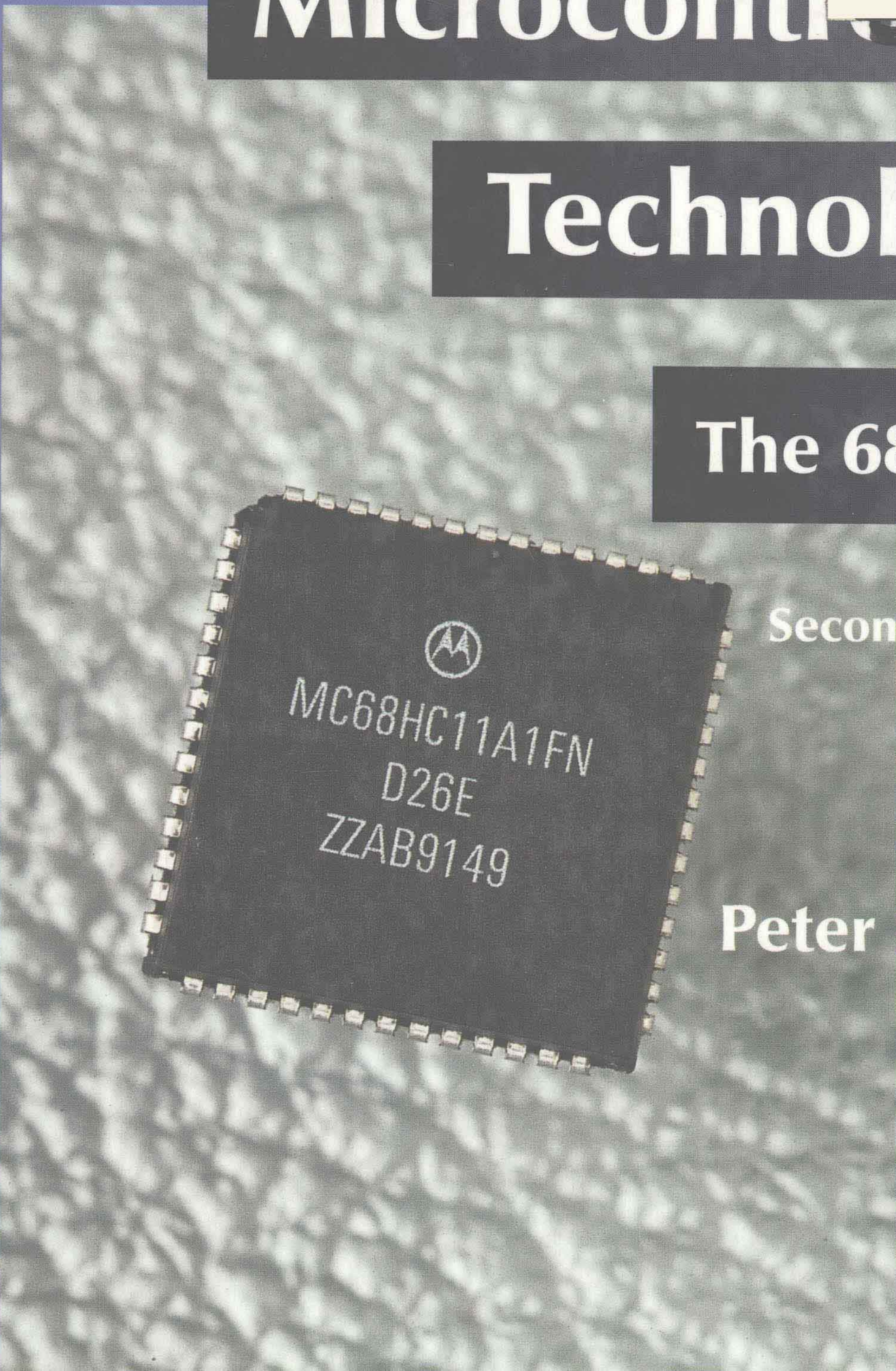
Microcontroller Technology

The 68HC11



Second Edition

Peter Spasov

SECOND EDITION

MICROCONTROLLER TECHNOLOGY THE 68HC11

Peter Spasov

Sir Sandford Fleming College

PRENTICE HALL
Englewood Cliffs, New Jersey Columbus, Ohio

To my wife Renate

Library of Congress Cataloging-in-Publication Data

Spasov, Peter.
Microcontroller technology : the 68HC11 / Peter Spasov.—2nd ed.
p. cm.
Includes bibliographical references and index.
ISBN 0-13-362724-1
1. Automatic control. 2. Digital control systems. I. Title
TJ223.M53S63 1996
629.8'9—dc20 95-16282
CIP

Acquisitions Editor: Charles E. Stewart, Jr.
Production Manager: Mary Irvin
Editorial/production supervision: WordCrafters Editorial Services, Inc.
Cover design: Graphica
Cover photo: © Phil Matt Photography
Prepress/manufacturing buyer: Deidra Schwartz

This book was set in Times Roman by Bi-Comp, Incorporated and was printed and bound by RR Donnelley & Sons Company, Harrisonburg, VA. The cover was printed by Phoenix Color Corp.

Printed in the United States of America

10 9 8 7 6 5 4

ISBN 0-13-362724-1

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*

Preface

Microcontrollers are used in the industrial world to control many types of equipment, ranging from consumer to specialized devices. They have replaced older types of controllers, including microprocessors. Also, there is a growing need for off-line support of a computer's main processor. The demand will grow as more and more equipment uses more intelligence. Applications range from controlling engines in modern automobiles to controlling laser printers and other computer peripherals. One consumer application is a washing machine controller that adjusts the wash cycle based on load size, fabric type, and amount of dirt.

This book illustrates how to use a popular microcontroller, the Motorola 68HC11. The 68HC11 is relatively easy to work with, yet it has most of the features essential for a complete control system. Thus the student of control automation can use it to work with control systems at the component level. The interested layperson can also use it as a tool to understand and experiment with computer and data communications systems.

The goals of the second edition remain the same as those of the first edition. The book provides the background knowledge needed to understand and use microcontrollers in general as well as the 68HC11 specifically. It starts at an introductory level, explaining the applications and origins of microcontrollers. Next, a programmer's view of the machine is developed. Finally, we describe machine hardware and how to connect it to the outside world for control applications.

The reader will find the book useful in learning how to use the 68HC11. Original manufacturers' data books do not provide the background explanations that a novice needs. Experienced users will find the book to be a useful accompaniment to the manufacturer's original documentation.

The book can be used as a textbook in introductory, interfacing, and industrial control courses. To use the book, the reader should have a background understanding of digital logic and numbering systems. Some computer programming knowledge is useful but not absolutely essential. Although this book was written to be used in college courses, the interested hacker or hobbyist should find the material useful for self-study. Others with a formal education from the past should also find useful information. The book can accompany hands-on exercises using a microcontroller training kit or simulation software. Where possible, concepts are presented generically, to help the reader understand other microcontrollers and related devices.

To permit flexible use of the book, we have organized it into five parts:

Part 1. Microcontroller technology is introduced in Chapter 1: what it is, how it evolved, and how to use it. The chapter continues with explanations of terminology and parts of the microcontroller. Finally, essential memory concepts are reviewed.

Part 2. Chapters 2 to 4 cover programming concepts, the language used to instruct the microcontroller, and how to use registers and memory. Also covered is how to produce, use, and document programs.

Part 3. Chapters 5 to 7 cover operation of the chip itself. Topics include the system bus, operating modes, clocked operation, and memory technology.

Part 4. Chapters 8 to 12 deal with the subsystems for parallel, serial, programmable timer, and analog interfacing. The basic software techniques to use these systems are presented. Also introduced are some common hardware designs used to connect the microcontroller to sensors and actuators. An overview of all subsystems without the processor-specific details is presented in Chapter 8.

Part 5. Control methods are illustrated in Chapter 13, where it is shown how programs use the subsystems for control applications. Chapters 14 and 15 cover the industry to date, presenting a survey of typical applications, choices in choosing microcontrollers, and characteristics of other microcontrollers.

Typically, an introductory course would include Parts 1 and 2. We also recommend including Chapter 8. Advanced or more intensive courses would include Parts 3 and 4. Either type of course can use Part 5, provided that Chapter 8 is also covered.

Although the book contains some data sheets, it is not a complete reference. The readers should refer to the literature listed in Appendix D for complete data.

▶ SECOND EDITION CHANGES

Chapters 2 and 3 contain more illustrations and explanations about basic concepts. This makes the book more useful to the beginner. For similar reasons, Appendix G contains a section describing fractions and normalizing. Chapter 13 contains a new section about fuzzy logic. This logic technique has become very popular since the first edition. Chapter 13 also introduces a third-party C cross compiler since the one mentioned in the first edition is no longer supported. Since the Internet has grown phenomenally, the second edition lists Internet resources for the 68HC11 in a new appendix. Minor changes include new developments in the 68HC11 family, a brief introduction to Petri nets, and more information on how a C program uses the stack.

▶ HOW TO USE THIS BOOK

Most important to success using this book is an enthusiasm for computer and related technology. If you are already familiar with assembly language programming, you may wish to skim Chapters 1 and 2 for new information instead of reading them in detail. If you are not familiar with digital logic and binary numbers, you should also read Appendix G.

We use generic principles whenever possible. You should be able to work with any microcontroller after studying this book. But it is necessary to cover a specific microcontroller in depth in order to understand them all in general. In this book we have chosen the 68HC11. However there are many variations of the 68HC11. When explaining specific details, we generally refer to the 68HC11A8, unless we state otherwise.

To understand microcontrollers, you also have to know a lot of *jargon*. You cannot understand the concepts without knowing the jargon—but you cannot understand the jargon without knowing the concepts. We get around this dilemma by introducing details step by step.

The appendices provide references and further clarification of terms used in the book. Appendix A lists the details of the 68HC11 instructions used throughout the book. You will probably use it often. Appendix B is the quick reference section. Use it as a cross-reference to look up other 68HC11 details as needed. Appendix A and B are necessary if you do not have 68HC11 data sheets or a manual.

Appendix C is a glossary of terms. Most terms are explained in the text as required. New terms are indicated in italic type upon occurrence in the text. You may wish to reread earlier sections of the book to review terms when used in another context. The glossary provides an additional reference.

A lot of abbreviations are used in the technical literature. Unfortunately, these frustrate many readers. In practice, most technical data books use abbreviations, often without explanation. In this book each abbreviation is defined the first time it is used, and many are redefined upon subsequent appearances. You can also use Appendix C to look up an abbreviation.

Sources of further information on some topics are listed in Appendix D. Also listed are suppliers for some of the commercially available products mentioned in the book. We encourage you to use the bulletin board mentioned in Appendix D.

Some conventions are defined in Appendix E. Since there are many ways to do the same thing, we felt it necessary to define a *standard way*. The conventions are those used by many in the industry. For example, we use Motorola's dollar sign ($) convention to refer to a method of coding numbers (in this case the sign does not indicate dollars). This ensures that we all understand the sign to have the same meaning.

The *header files* used by some program examples are listed in Appendix F. Note that a header file is explained in Appendix F.

The basics of digital logic and number conversions are explained in Appendix G. You should read this section early if you are not already familiar with these topics.

If you need basic information about waveforms, refer to Appendix H.

Finally, we included a reference to Internet resources in Appendix I. It lists sources of free software and information that is available for the 68HC11 microcontroller.

Supplements

Prentice Hall also offers a lab manual, written by Robert John Dirkman and John Leonard of the University of Massachusetts, that can be used with this book. It is called the *Microprocessor Applications Laboratory Workbook*. Instructors can contact Prentice Hall for a copy of the "Instructor's Solutions Manual with Transparency Masters" to accompany this book.

▶ ACKNOWLEDGMENTS

I would like to thank Hugh K. McGugan of Motorola Training Organization, Canada, for technical literature, review of some chapters, and the assistance he always gave willingly and promptly. Without his help this book would not exist. I also thank Alan D. Niemi of Lake Superior State University, Sault Sainte Marie, Michigan, for his valuable comments and advice when reviewing sections of the preliminary manuscript. Helpful advice was received from V. S. Anandu of Southwest Texas State University, San Antonio, Texas, and from Jeffrey Rankinen of Pennsylvania College of Technology during review of the manuscript. Also, I thank my students and teaching colleagues whose experiences led me to make some of the changes in the second edition. Any deficiencies are mine because I did not always follow their advice.

I am grateful to Hitoshi Doi and Chuck Westfall of Canon Incorporated, Japan, for valuable information about their model EOS cameras. In addition, I thank Richard Man of ImageCraft, USA, for providing information as well as the software for their C cross compiler. Finally, thanks to Linda Zuk of WordCrafters, who provided excellent and professional support during the editorial and production phase.

Finally, a very special thanks to my wife, Renate, who provided invaluable emotional support. I also thank my three-year-old daughter, Emilie, and recently-arrived Hannah, who both provide much joy.

Contents

▶ PART 1 Introducing Microcontroller Technology 1

1 Microcontroller Concepts 1

1.1 What Is a Microcontroller? and What Is It Used For? 1
1.1.1 The General View, 2 1.1.2 Terminology and Conventions, 2 1.1.3 The Technical View, 4 1.1.4 The Specific View, 4

1.2 History 5
1.2.1 Early Computing Machines, 5 1.2.2 Digital Computers, 5 1.2.3 Microprocessors, 5 1.2.4 Microcontrollers, 6 1.2.5 The Motorola MC68HC11, 6

1.3 Top-Down View of Microcontroller Systems 9
1.3.1 The Microcontroller System, 9 1.3.2 The Microcontroller Unit (MCU), 10 1.3.3 The Central Processing Unit (CPU), 13

1.4 Memory Concepts 14
1.4.1 Organization, 14 1.4.2 Semiconductor Memory, 15

1.5 Microcontroller Memory Map 16

▶ PART 2 Software 18

2 Programming 8

2.1 Assembly and Other Programming Languages 18

2.2 Source Code, Object Code, and the Assembler 20
2.2.1 Machine Language, 21 2.2.2 Assembly Language, 23 2.2.3 Examples, 25 2.2.4 Manual Assembly, 26

2.3 Using High-Level Languages 27
2.3.1 General Concepts, 27 2.3.2 The C Language for Microcontrollers, 28

2.4 Fetch/Execute Operation of the Central Processing Unit (CPU) 30

2.5 The Instruction Set and Addressing Modes 34
2.5.1 Instruction Set References, 34 2.5.2 Types of Instructions, 36 2.5.3 Addressing Modes, 36 2.5.4 The Prebyte, 37 2.5.5 Inherent Addressing Mode, 37 2.5.6 Listing and Execution Conventions, 40 2.5.7 Stopping a Program, 42 2.5.8 Immediate Addressing Mode, 42 2.5.9 Direct and Extended Addressing Modes, 45 2.5.10 Indexed Addressing Mode, 50

2.5.11 Memory Dump Convention, 58 *2.5.12 Addressing Review, 59*
2.6 Basic Operations 60
2.6.1 Data Handling, 60 *2.6.2 Arithmetic, 62* *2.6.3 Logic, 64* *2.6.4 Other Operations, 68*
2.7 Microcontroller Arithmetic and the Condition Code Register 68
2.7.1 Two's Complement and the Sign Bit, 69 *2.7.2 Carry, Overflow, Zero, and Half Carry, 71* *2.7.3 Binary-Coded-Decimal (BCD) Arithmetic, 79* *2.7.4 Multiplication, 80* *2.7.5 Integer Division, 81* *2.7.6 Fractional Division, 82* *2.7.7 Floating-Point Numbers, 83*
2.8 Program Flow Control Using Looping and Branching 85
2.8.1 Flow Control, 85 *2.8.2 Conditional Branches, 87* *2.8.3 Relative Addressing, 91* *2.8.4 Secondary Memory Reference Instructions, 94* *2.8.5 Jump Instructions, 96* *2.8.6 Relocatable Programs, 97*
2.9 Summary 97
Exercises 98

3 The Stack, Subroutines, Interrupts, and Resets 106
3.1 Introducing the Stack 106
3.2 Using the Stack to Store Data 108
3.3 Using Subroutines 111
3.4 Modular Programming Using Subroutines 116
3.4.1 Top-Down Approach to Problem Solving, 116 *3.4.2 Subroutine Libraries, 120* *3.4.3 Documentation and Using Subroutines, 121* *3.4.4 Simple Parameter Passing, 124*
3.5 Subroutine Operation 125
3.5.1 Subroutine Use of the Stack, 125 *3.5.2 Nested Subroutine Details, 128*
3.6 Concept of Interrupts 130
3.6.1 Resets, 130 *3.6.2 Interrupts, 133*
3.7 Interrupt Vectors 134
3.8 Interrupt Operation 137
3.8.1 Interrupt Masks and Enables, 137 *3.8.2 Stacking the Registers, 138* *3.8.3 Interrupt Priority and Nested Interrupts, 139*
3.9 Hardware Interrupts and Resets 140
3.9.1 Power-On and External Reset, 140 *3.9.2 Other Processor Resets, 140* *3.9.3 Illegal Opcode Trap, 142* *3.9.4 Nonmaskable Interrupt (XIRQ), 143* *3.9.5 Interrupt Request (IRQ), 145* *3.9.6 Interrupt Polling, 145*
3.10 Software and CPU Control Interrupts 146
3.10.1 Software Interrupt (SWI), 146 *3.10.2 CPU Control and Monitoring, 146*
3.11 The Kiss of Death: Stack Overflow 147

3.12 Summary 148
Exercises 149

4 Cross Assembly and Program Development 153

4.1 Introduction to Program Development 153
4.2 Format of the Source Code 154
4.3 Code and Data Segments 158
4.4 Pseudo-operations 160
4.5 The Assembler Two-Pass Process 170
4.6 Assembler Options and Preprocessor Directives 172
4.7 Hex and Binary Files 177
4.8 Documentation Files 180
4.9 Simulation 184
4.10 Evaluation Boards and Emulation 185
4.11 Summary and Final Word 188
Exercises 189

▶ PART 3 Hardware 193

5 Bus Concepts and Modes of Operation 193

5.1 Introduction 193
5.2 The Bus 194
5.3 Tristate 196
5.4 Address Decoding 199
5.4.1 Address Decoding Circuit, 199 *5.4.2 Address Decoding Design Example, 201* *5.4.3 Partially and Fully Decoded Systems, 201*
5.5 Modes of Operation 203
5.6 Single-Chip Operating Mode 204
5.7 Expanded Multiplexed Operating Mode 205
5.8 Special Bootstrap Mode 206
5.9 Special Test Operating Mode 208
5.10 Summary 208
Exercises 209

6 Microcontroller Hardware 211

6.1 Semiconductor Technology 211
6.1.1 The Integrated Circuit, 211 *6.1.2 Some Transistor Theory, 212* *6.1.3 Fabrication, 215* *6.1.4 HCMOS and Other Logic Families, 215* *6.1.5 CMOS and HCMOS Handling Precautions, 216*
6.2 Chip Specifications 219
6.2.1 Data Sheets, 219 *6.2.2 Maximum Ratings, 220*

6.2.3 Recommended Operating Conditions, 220 6.2.4 DC Electrical Characteristics, 220 6.2.5 AC Electrical Characteristics, 222

6.3 Memory Technology 224

6.3.1 Review and Overview, 224 6.3.2 Read-Only Memory (ROM), 224 6.3.3 Random Access Memory (RAM), 225

6.4 Erasable Programmable Read-Only Memory (EPROM) 226

6.5 Flash Memory 229

6.6 Electrically Erasable Programmable Read-Only Memory (EEPROM) 230

6.6.1 Organization and Applications, 230 6.6.2 The Register Block, 230 6.6.3 EEPROM Operation and Specifications, 231 6.6.4 Programmer's Guide to Changing EEPROM, 232

6.7 Configuration Control Register (CONFIG) 236

6.8 RAM and I/O Mapping Register (INIT) 236

6.9 Summary 237

Exercises 237

7 Clocked Operation 240

7.1 Timing Diagrams 240

7.2 System Clock 242

7.3 Bus Operations 244

7.4 Cycle-by-Cycle Operation 250

7.5 Oscilloscope Test Loops 252

7.6 Logic Analyzer 254

7.7 Transmission Line Effects 257

7.8 Summary 261

Exercises 262

▶ PART 4 Interfacing 264

8 Interfacing Concepts 264

8.1 Introduction 264

8.2 Input/Output Subsystems and Registers 266

8.2.1 Different Subsystems, 266 8.2.2 I/O Registers, 266 8.2.3 I/O Ports, 268 8.2.4 Port Replacement Unit, 268

8.3 Memory or Input/Output Mapping 269

8.4 Interfacing Using Polling or Interrupts 271

8.4.1 Polled I/O, 271 8.4.2 Interrupt-Driven I/O, 273

8.5 The Parallel I/O Subsystem 274

8.6 Serial Systems 275

8.6.1 Synchronous Serial I/O Subsystem, 275 8.6.2 Asynchronous Serial I/O Subsystem, 277

8.7 Programmable Timer I/O Subsystem 278

8.8 Analog/Digital I/O Subsystem 280

8.9 The I/O Subsystem Registers 283
8.9.1 Control Register and System Configuration, 283 8.9.2 Status Registers, 285 8.9.3 Data Registers, 286
8.10 Interface Standards 288
8.11 Summary 289
Exercises 290

9 Parallel Input/Output 292

9.1 Introduction to the Subsystem 293
9.2 Seven-Segment (LED) Display Output 293
9.2.1 The Seven-Segment Display, 293 9.2.2 Seven-Segment Code, 295 9.2.3 Driving the Seven-Segment Display, 295 9.2.4 Software Seven-Segment Code Look-up, 297 9.2.5 Hardware Decoder, 298 9.2.6 Multiplexed Displays, 298
9.3 Liquid Crystal Displays (LCDs) 301
9.3.1 Operation, 301 9.3.2 Seven-Segment LCD, 302 9.3.3 Dot Matrix LCD, 303 9.3.4 LCD Excitation Waveforms, 307
9.4 Keyboard Interfacing 308
9.4.1 The Matrix Keyboard, 308 9.4.2 Keyboard Decoding for 2 out of 7, 310 9.4.3 Debouncing, 314 9.4.4 Software Drivers for Matrix Keyboard, 314 9.4.5 Using Hardware Decoding Chips, 320
9.5 Other User Input/Outputs 321
9.6 Strobed Input/Output 322
9.6.1 Input Strobe, 322 9.6.2 Output Strobe, 324 9.6.3 Configuration, 326
9.7 Full Handshake Input/Output 327
9.7.1 Input Handshake, 328 9.7.2 Output Handshake, 332
9.8 Parallel Interface Standards 336
9.8.1 Centronics Parallel Interface, 336 9.8.2 The IEEE-488 General Purpose Instrumentation Bus, 344 9.8.3 Small Computer Systems Interface (SCSI), 347
9.9 Summary 350
9.9.1 Parallel I/O Subsystems Summary, 350 9.9.2 Parallel I/O Applications Summary, 353
Exercises 354

10 The Serial Subsystems 357

10.1 Asynchronous Communications Systems 357
10.1.1 A Serial Communications Primer, 357 10.1.2 Protocols and Flow Control, 364 10.1.3 Communication Channel Operation, 365 10.1.4 Modems, 365
10.2 The Serial Communications Interface (SCI) 368
10.2.1 The SCI Subsystem in the 68HC11, 368 10.2.2 Transmit Operation, 369 10.2.3 Receive Operation, 370 10.2.4 SCI Software, 371

10.3 SCI Registers 377
10.3.1 Serial Communication Data Register (SCDR), 377 10.3.2 The Baud Rate Register (BAUD), 377 10.3.3 Serial Communication Control Register 1 (SCCR1), 378 10.3.4 Serial Communication Control Register 2 (SCCR2), 380 10.3.5 Serial Communication Status Register (SCSR), 383 10.3.6 Receive Errors, 385
10.4 Synchronous Serial Input/Output 388
10.5 The Serial Peripheral Interface (SPI) 390
10.6 SPI Topologies and Applications 391
10.6.1 Bus Topology, 391 10.6.2 Microcontroller I/O with Bus Topology, 392 10.6.3 Cascade Topology, 394 10.6.4 Microcontroller I/O with Cascade Topology, 395 10.6.5 Summary of SPI Data Transfer Operations, 396
10.7 SPI Software 397
10.7.1 Configuration and Registers, 397 10.7.2 Master SPI Operation, 401 10.7.3 Slave SPI Operation, 402 10.7.4 Sample Driver Routines, 403 10.7.5 Transfer to Peripherals with Odd Word Lengths, 405
10.8 SPI Error Handling 406
10.9 Network Communications 407
10.10 Summary 409
10.10.1 Asynchronous Communications, 409 10.10.2 The Serial Communications Interface (SCI), 410 10.10.3 Synchronous Serial Systems, 410 10.10.4 The Serial Peripheral Interface (SPI), 411
Exercises 411

11 Programmable Timer Operations 415
11.1 Overview 415
11.2 Output Compare 421
11.2.1 Sequence of Operations, 421 11.2.2 One-Shot Pulse Example, 424 11.2.3 Square-Wave and Pulse-Width Modulation (PWM) Outputs, 426 11.2.4 Stepper Motor Outputs, 432
11.3 Input Capture 437
11.3.1 Sequence of Operations, 438 11.3.2 Pulse Width Example, 439 11.3.3 Period (Frequency, Speed) Measurement, 440 11.3.4 Measuring Long Periods, 445
11.4 Pulse Accumulator 448
11.4.1 Function Description, 448 11.4.2 Event Counting, 449 11.4.3 Gated Time Accumulation, 454
11.5 Real-Time Interrupt 457
11.6 Summary 461
Exercises 461

12 The Analog Converter Subsystem 464
12.1 Concepts and Terminology 465

12.2 Transducers 469
12.2.1 Sensors, 469 *12.2.2 Signal Conditioning, 473*
12.2.3 Actuators, 475
12.3 Digital-to-Analog Converters 477
12.4 Analog-to-Digital Converters 484
12.5 The Built-in A/D Subsystem 488
12.6 Summary 494
Exercises 495

▶ PART 5 The Microcontroller World 497

13 Application Control Software 497
13.1 Boolean Logic Control 497
13.2 Sequencers 499
13.3 Sequential Machines 503
13.4 Feedback Control of the Analog World 510
13.5 Digital Signal Processors 517
13.6 Using the C Language 524
13.6.1 Why C?, 525 *13.6.2 Control Examples, 527* *13.6.3 The PID Controller Using C, 533* *13.6.4 C Function Library, 536* *13.6.5 User-Defined Functions, 541* *13.6.6 Behind the Scenes: Using the Stack, 543* *13.6.7 Pointers, 546* *13.6.8 Other Variable Types and Features, 549*
13.7 Multitasking 549
13.8 Fuzzy Logic 557
13.8.1 Why Use It and What Is It?, 557 *13.8.2 Overview, 557* *13.8.3 Create a Knowledge Base, 559* *13.8.4 Fuzzy Logic Inference, 567* *13.8.5 Development and Further Information, 571*
13.9 Summary 572
Exercises 574

14 Applications 577
14.1 Automotive-Engine Control 577
14.2 Consumer Electronics: Automatic Camera 584
14.3 Computer Peripherals: Dot Matrix Printer 592
14.4 Society and Microcontroller Technology 603

15 Eight-Bit Microcontroller Families 605
15.1 Microcontrollers and the Semiconductor Industry 606
15.2 Motorola Families 608
15.2.1 The 68HC11 Family, 608 *15.2.2 The 68HC05 Family, 610*
15.2.3 The 68HC16 Family, 613 *15.2.4 The 68HC08 Family, 615*
15.3 The Intel 8051 Core 617
15.4 Survey of Other Families 626
15.5 Choosing a Microcontroller 628
15.6 Summary 632

Appendix A Instruction Set Summary 633

Appendix B Quick Reference 641

Appendix C Glossary 651

Appendix D Further Information 664

Appendix E Conventions 671

Appendix F Header and Library File Source Listings 678

Appendix G Digital Logic and Binary Codes 688

Appendix H Basic Waveforms 704

Appendix I Internet Resources 706

INDEX 713

PART 1
Introducing Microcontroller Technology

1

Microcontroller Concepts

1.1 WHAT IS A MICROCONTROLLER? AND WHAT IS IT USED FOR?

The Swiss Army Knife. "Choose," the voice says. What do you choose? You may take only one tool for your journey. The journey will take you through forests, plains, mountains, and deserts. You need something small, light, and useful. You choose the Swiss Army knife and the voice "nods" respectfully. It is a good choice. Depending on the model, it has assorted blades, saw blade, scissors, can opener, corkscrew (most important!), tweezers, toothpick, and possibly more.

"What is a microcontroller?" the voice asks. You say: "It is a (computerized) Swiss Army knife." The voice is satisfied. Indeed, the microcontroller bears much resemblance to the Swiss Army knife. It is a single device, in this case a chip. The microcontroller follows instructions, reads information, stores information, communicates, measures time, and switches things on and off. It also does other things, depending on the model. Despite its power and versatility, you may not hear much about microcontrollers. Most computer literature describes the processor chips used in computers. But most single-chip programmable devices are microcontrollers.

If you are the type of person who likes to take things apart, you will find microcontrollers in all kinds of places. The most common place is under the hood of almost any car produced since 1985. Consumer items include televisions, compact disc players, washing machines, telephones, and microwave ovens. Office computers use microcontrollers in addition to their main processor to control peripherals such as keyboards and printers. Automated manufacturing systems use microcontrollers in production equipment such as robots and conveyor lines.

Now, we will take three views of microcontrollers: general, technical, and specific. After we consider the general view, we consider the technical and specific aspects of microcontrollers.

1.1.1 The General View

One could say that a microcontroller is a programmable single-chip integrated circuit (IC) that controls the operation of a system. Often, the system to be controlled is a machine. One of the most common applications is automotive control. This includes engine fuel injection control, transmission control, suspension and ride control, instrument displays, and braking systems. In fact, the 68HC11 microcontroller managed some engine control functions of first-place winners of Indianapolis 500 races for five years in a row.

Refer to Figure 1.1 for an application in controlling air/fuel mixture in an automobile engine. The amount of air and fuel and the timing of the ignition spark determine the fuel efficiency and amount of exhaust emission. The microcontroller unit uses the sensor inputs to control the ignition timing for maximum fuel efficiency and minimum exhaust emissions. It also looks at how fast the inputs change. The exhaust gas sensor measures the oxygen level so that the system can correct for any error. The microcontroller can be programmed to handle conditions when the engine is cold or warm, and accelerating or cruising. Also, the driver could instruct the unit to optimize speed and acceleration performance if necessary. With some extra inputs, the microcontroller unit could maintain constant speed, such as in a cruise control.

We went through this example in some detail to illustrate what could be involved in a control system using a microcontroller. For a more detailed understanding of air/fuel ratio control, refer to reference 1.

Automatic cameras also use microcontrollers. They control exposure and focus. Another application is the computer mouse. It typically uses a microcontroller to read the mouseball movement, sense the pushbutton positions, and handle communications with the computer.

1.1.2 Terminology and Conventions

We will be using words, abbreviations, and acronyms specifically used in the computer and microcontroller industry. Although we have minimized the use of technical jargon, its use is sometimes unavoidable. When we first use a term, we italicize it and you will generally find a definition in the glossary (Appendix C). You will have to understand some terms in context when you first encounter them. You may also check the index to locate other sections where a term is used.

In the technical description that follows (Section 1.1.3), we use a number of technical terms. Since the computer and microcontroller field is changing rapidly,