



# **THE 8051 MICROCONTROLLER**

**FOURTH EDITION**

**I. Scott MacKenzie**

*York University*

**Raphael C.-W. Phan**

*Swinburne University of Technology  
(Sarawak Campus)*



*Pearson Education International*



# PREFACE

This book examines the hardware and software features of the MCS-51 family of microcontrollers. The intended audience is college or university students of electronics, computer technology, and electrical or computer engineering, or practicing technicians or engineers interested in learning about microcontrollers.

The means to effectively fulfill that audience's informational needs were tested and refined in the development of this book. In its prototype form, *The 8051 Microcontroller* was the basis of a fifth-semester course for college students in computer engineering. As detailed in Chapter 11, students built an 8051 single-board computer as part of this course. That computer, in turn, has been used as the target system for a final, sixth-semester "project" course in which students design, implement, and document a "product" controlled by the 8051 microcontroller and incorporating original software and hardware.

Because the 8051—like all microcontrollers—contains a high degree of functionality, the book emphasizes architecture and programming rather than electrical details. The software topics are delivered in the context of Intel's assembler (ASM51) and linker/locator (RL51).

Four new chapters are included in this edition, and the main additional feature is the information about using 8051 C programming as an alternative to the assembly language used in earlier editions. Programming in C allows for structured programs and is especially useful in coding big and complex 8051-based projects.

All examples are annotated to assist both the student and the teacher. The examples begin by stating a problem followed by a straightforward solution. Then, following the solution, there is a discussion that explores the inner workings of the problem and the solution. The approach is to explain and to elaborate, taking into account different perspectives that enter into the example.

It is our view that courses on microprocessors or microcontrollers are inherently more difficult to deliver than courses in, for example, digital systems, because a linear sequence of topics is hard to devise. The very first program that is demonstrated to students brings with it significant assumptions, such as a knowledge of the CPU's programming model and addressing modes, the distinction between an address and the content of an address, and so on. For this reason, a course based on this book should not attempt to follow strictly the sequence presented. Chapter 1 is a good starting point, however. It serves as a general introduction to microcontrollers, with particular emphasis on the distinctions between microcontrollers and microprocessors.

Chapter 2 introduces the hardware architecture of the 8051 microcontroller and its counterparts that form the MCS-51 family. Concise examples are presented using short sequences of instructions. Instructors should be prepared at this point to introduce, in parallel, topics from Chapters 3 and 7 and Appendices A and C to support the requisite software knowledge in these examples. Appendix A is particularly valuable, since it contains in a single figure the entire 8051 instruction set.

Chapter 3 introduces the instruction set, beginning with definitions of the 8051's addressing modes. The instruction set has convenient categories of instructions (data transfer, branch, etc.) that facilitate a step-wise presentation. Numerous brief examples demonstrate each addressing mode and each type of instruction.

Chapters 4, 5, and 6 progress through the 8051's on-chip features, beginning with the timers, advancing to the serial port (which requires a timer as a baud rate generator), and concluding with interrupts. The examples in these chapters are longer and more complex than those presented earlier. Instructors are wise not to rush into these chapters; it is essential that students gain solid understanding of the 8051's hardware architecture and instruction set before advancing to these topics.

Many of the topics in Chapter 7 will be covered, by necessity, in progressing through the first six chapters. Nevertheless, this chapter is perhaps the most important for developing in students the potential to undertake large-scale projects. Advanced topics such as assemble-time expression evaluation, modular programming, linking and locating, and macro programming will be a significant challenge for many students. At this point, the importance of hands-on experience cannot be overemphasized. Students should be encouraged to experiment by entering the examples in the chapter into the computer and observing the output and error messages provided by ASM51, RL51, and the object-to-hex conversion utility (OH).

Chapter 8 lays the foundation for C programming the 8051. It highlights differences between this higher-level language compared to assembly language, and differences between conventional C language for computer systems and C for an embedded microcontroller such as the 8051.

Some advanced topics relating to programming methods, style, and the development environment are presented in Chapters 9 and 10. These chapters address larger, more conceptual topics important in professional development environments.

Chapter 11 presents several design examples incorporating selected hardware with supporting software. The software is fully annotated and is the real focus in these examples. The fourth edition includes several additional interfaces: a liquid crystal display (LCD), the 8255, an RS-232 serial interface, a Centronics parallel interface, sensors, relays, and a stepper motor. One of the designs in Chapter 11 is the SBC-51 - the 8051 single-board computer. The SBC-51 can form the basis of a course on the 8051 microcontroller. A short monitor program is included (see Appendix G), which is sufficient to get "up and running." A development environment also requires a host computer, which doubles as a dumb terminal for controlling the SBC-51 after programs have been downloaded for execution.

Many dozens of students have wire-wrapped prototype versions of the SBC during years that Scott has taught 8051-based courses to computer engineering students. Raphael also thanks his Microprocessor Fundamentals, Microprocessor Applications, and Embedded

Microcontrollers students, who enthusiastically undertook assignments and projects based on the 8051.

There is also a new chapter, Chapter 12, on the design and interface examples given in Chapter 11, but with the solutions in C rather than in assembly language.

Chapter 13 presents some more advanced examples of 8051 projects for students and concentrates on the discussion of design choices and the importance of pseudo code in the design process, prior to the actual coding.

Chapter 14 talks briefly about some 8051 derivative devices that are descendants of the 8051 but with enhancements such as increases in speed and memory size, additional built-in peripherals, and enhanced network capabilities and security mechanisms.

Also worth mentioning is the treatment of smart cards and data security in this edition, notably in Chapters 12, 13, and 14, and in Appendix J. This information is included because of the increasing popularity of smart cards using 8-bit microcontrollers such as the 8051 to run security software to protect confidential information.

The book makes extensive use of and builds on Intel's literature on the MCS-51 devices. In particular, Appendix C contains the definitions of all 8051 instructions, and Appendix E contains the 8051 data sheet. Intel's cooperation is gratefully acknowledged.

All the 8051 C examples in this edition have been compiled, debugged, and tested with Keil's µvision2 IDE, available for download at <http://www.keil.com>. We also thank the following for their review and invaluable comments, criticism, and suggestions: Dwight Egbert, University of Nevada; Marty Kaliski, Cal Polytech State University; Claude Kansaku, Oregon Institute of Technology; and Ron Tinkham, Santa Fe Community College. Raphael thanks his wife, Grace, for her understanding and patience, and for sacrificing all the nights, weekends, and public holidays to keep him company in writing this edition. In fact, without her gentle nudges, this edition would not have been completed. This edition is dedicated to her.

I. Scott MacKenzie Raphael C.-W.  
Phan



# CONTENTS

## **1 INTRODUCTION TO MICROCONTROLLERS 1**

1.1	Introduction	1
1.2	Terminology	3
1.3	The Central Processing Unit	4
1.4	Semiconductor Memory: RAM and ROM	5
1.5	The Buses: Address, Data, and Control	6
1.6	Input/Output Devices	7
	1.6.1 Mass Storage Devices   1.6.2 Human Interface Devices   1.6.3 Control/Monitor Devices	
1.7	Programs: Big and Small	8
1.8	Micros, Minis, and Mainframes	9
1.9	Microprocessors vs. Microcontrollers	10
	1.9.1 Hardware Architecture   1.9.2 Applications   1.9.3 Instruction Set Features	
1.10	New Concepts	12
1.11	Gains and Losses: A Design Example	13
	Problems	14

## **2 HARDWARE SUMMARY 17**

2.1	MCS-51 Family Overview	17
2.2	Once Around the Pins	18
	2.2.1 Port 0   2.2.2 Port 1   2.2.3 Port 2   2.2.4 Port 3	
	2.2.5 PSEN (Program Store Enable)   2.2.6 ALE (Address Latch Enable)   2.2.7 EA (External Access)   2.2.8 RST (Reset)   2.2.9 On-Chip Oscillator Inputs	
	2.2.10 Power Connections	
2.3	I/O Port Structure	22
2.4	Timing and the Machine Cycle	23
2.5	Memory Organization	24

	2.5.1 General-Purpose RAM   2.5.2 Bit-Addressable RAM   2.5.3 Register Banks	
2.6	Special Function Registers	28
	2.6.1 Program Status Word   2.6.2 B Register   2.6.3 Stack Pointer   2.6.4 Data Pointer   2.6.5 Port Registers   2.6.6 Timer Registers   2.6.7 Serial Port Registers   2.6.8 Interrupt Registers   2.6.9 Power Control Register	
2.7	External Memory	36
	2.7.1 Accessing External Code Memory   2.7.2 Accessing External Memory   2.7.3 Address Decoding   2.7.4 Overlapping the External Code and Data Spaces	
2.8	8032/8052 Enhancements	41
2.9	Reset Operation	43
	Summary	44
	Problems	44

### **3 INSTRUCTION SET SUMMARY 49**

3.1	Introduction	49
3.2	Addressing Modes	50
	3.2.1 Register Addressing   3.2.2 Direct Addressing   3.2.3 Indirect Addressing   3.2.4 Immediate Addressing   3.2.5 Relative Addressing   3.2.6 Absolute Addressing   3.2.7 Long Addressing   3.2.8 Indexed Addressing	
3.3	Instruction Types	59
	3.3.1 Arithmetic Instructions   3.3.2 Logical Instructions   3.3.3 Data Transfer Instructions   3.3.4 Boolean Instructions   3.3.5 Program Branching Instructions	
	Summary	78
	Problems	78

### **4 TIMER OPERATION 87**

4.1	Introduction	87
4.2	Timer Mode Register (TMOD)	89
4.3	Timer Control Register (TCON)	89
4.4	Timer Modes and the Overflow Flag	90
	4.4.1 13-Bit Timer Mode (Mode 0)   4.4.2 16-Bit Timer Mode (Mode 1)   4.4.3 8-Bit Auto-Reload Mode (Mode 2)   4.4.4 Split Timer Mode (Mode 3)	
4.5	Clocking Sources	92
	4.5.1 Interval Timing   4.5.2 Event Counting	

4.6	Starting, Stopping, and Controlling the Timers	93
4.7	Initializing and Accessing Timer Registers	95
	4.7.1 Reading a Timer "on the Fly"	
4.8	Short, Medium, and Long Intervals	96
4.9	Producing Exact Frequencies	102
	4.9.1 Eliminating Round-off Errors   4.9.2 Compensating for Overhead Due to Instructions	
4.10	8052 Timer 2	105
	4.10.1 Auto-Reload Mode   4.10.2 Capture Mode	
4.11	Baud Rate Generation	106
	Summary	107
	Problems	107

## 5 SERIAL PORT OPERATION 111

5.1	Introduction	111
5.2	Serial Communication	111
5.3	Serial Port Buffer Register (SBUF)	112
5.4	Serial Port Control Register (SCON)	113
5.5	Modes of Operation	113
	5.5.1 8-Bit Shift Register (Mode 0)   5.5.2 8-Bit UART with Variable Baud Rate (Mode 1)   5.5.3 9-Bit UART with Fixed Baud Rate (Mode 2)   5.5.4 9-Bit UART with Variable Baud Rate (Mode 3)	
5.6	Full Duplex Serial Communication: Issues	117
5.7	Initialization and Accessing Serial Port Registers	118
	5.7.1 Receiver Enable   5.7.2 The Ninth Data Bit   5.7.3 Adding a Parity Bit   5.7.4 Interrupt Flags	
5.8	Multiprocessor Communications	119
5.9	Serial Port Baud Rates	120
	5.9.1 Using Timer 1 as the Baud Rate Clock	
	Summary	127
	Problems	128

## 6 INTERRUPTS 131

6.1	Introduction	131
6.2	8051 Interrupt Organization	132
	6.2.1 Enabling and Disabling Interrupts   6.2.2 Interrupt Priority   6.2.3 Polling Sequence	

6.3	Processing Interrupts	136
	6.3.1 Interrupt Vectors	
6.4	Program Design Using Interrupts	137
	6.4.1 Small Interrupt Service Routines   6.4.2 Large Interrupt Service Routines	
6.5	Timer Interrupts	139
6.6	Serial Port Interrupts	142
6.7	External Interrupts	143
6.8	Interrupt Timings	148
	Summary	149
	Problems	150

## **7 ASSEMBLY LANGUAGE PROGRAMMING 151**

7.1	Introduction	151
7.2	Assembler Operation	152
	7.2.1 Pass One   7.2.2 Pass Two	
7.3	Assembly Language Program Format	155
	7.3.1 Label Field   7.3.2 Mnemonic Field   7.3.3 Operand Field   7.3.4 Comment Field   7.3.5 Special Assembler Symbols   7.3.6 Indirect Address   7.3.7 Immediate Data   7.3.8 Data Address   7.3.9 Bit Address   7.3.10 Code Address   7.3.11 Generic Jumps and Calls	
7.4	Assemble-Time Expression Evaluation	160
	7.4.1 Number Bases   7.4.2 Character Strings   7.4.3 Arithmetic Operators   7.4.4 Logical Operators   7.4.5 Special Operators   7.4.6 Relational Operators   7.4.7 Expression Examples   7.4.8 Operator Precedence	
7.5	Assembler Directives	164
	7.5.1 Assembler State Control   7.5.2 Symbol Definition   7.5.3 Storage Initialization/Reservation   7.5.4 Program Linkage   7.5.5 Segment Selection Directives	
7.6	Assembler Controls	173
7.7	Linker Operation	173
7.8	Annotated Example: Linking Relocatable Segments and Modules	176
	7.8.1 ECHO.LST   7.8.2 IO.LST   7.8.3 EXAMPLE.M51	
7.9	Macros	183
	7.9.1 Parameter Passing   7.9.2 Local Labels   7.9.3 Repeat Operations   7.9.4 Control Flow Operations	
	Summary	188
	Problems	188



## **8 8051 C PROGRAMMING 191**

- 8.1 Introduction 191
- 8.2 Advantages and Disadvantages of 8051 C 191
- 8.3 8051 C Compilers 192
- 8.4 Data Types 193
- 8.5 Memory Types and Models 197
- 8.6 Arrays 198
- 8.7 Structures 199
- 8.8 Pointers 199
  - 8.8.1 A Pointer's Memory Type V 8.8.2 Typed Pointers |
  - 8.8.3 Untyped Pointers
- 8.9 Functions 202
  - 8.9.1 Parameter Passing | 8.9.2 Return Values
- 8.10 Some 8051 C Examples 204
  - 8.10.1 The First Program || 8.10.2 Timers | 8.10.3 Serial Port | 8.10.4 Interrupts
- Summary 214
- Problems 214

## **9 PROGRAM STRUCTURE AND DESIGN 217**

- 9.1 Introduction 217
- 9.2 Advantages and Disadvantages of Structured Programming 219
- 9.3 The Three Structures 220
  - 9.3.1 Statements | 9.3.2 The Loop Structure | 9.3.3 The Choice Structure
- 9.4 Pseudo Code Syntax 234
- 9.5 Assembly Language Programming Style 237
  - 9.5.1 Labels | 9.5.2 Comments || 9.5.3 Comment Blocks | 9.5.4 Saving Registers on the Stack | 9.5.5 The Use of Equates || 9.5.6 The Use of Subroutines | 9.5.7 Program Organization
- 9.6 8051 C Programming Style 243
  - 9.6.1 Comments | 9.6.2 The Use of Defines | 9.6.3 The Use of Functions | 9.6.4 The Use of Arrays and Pointers | 9.6.5 Program Organization
- Summary 245
- Problems 245

## **10 TOOLS AND TECHNIQUES FOR PROGRAM DEVELOPMENT 247**

- 10.1 Introduction 247
- 10.2 The Development Cycle 247
  - 10.2.1 Software Development | 10.2.2 Hardware Development
- 10.3 Integration and Verification 251
  - 10.3.1 Software Simulation | 10.3.2 Hardware Emulation | 10.3.3 Execution from RAM | 10.3.4 Execution from EPROM | 10.3.5 The Factory Mask Process
- 10.4 Commands and Environments 255
- Summary 257
- Problems 257

## **11 DESIGN AND INTERFACE EXAMPLES 259**

- 11.1 Introduction 259
- 11.2 The SBC-51 259
- 11.3 Hexadecimal Keypad Interface 265
- 11.4 Interface to Multiple 7-Segment LEDs 267
- 11.5 Interface to Liquid Crystal Displays (LCDs) 273
- 11.6 Loudspeaker Interface 276
- 11.7 Nonvolatile RAM Interface 277
- 11.8 Input/Output Expansion 282
  - 11.8.1 Using Shift Registers | 11.8.2 Using the 8255
- 11.9 RS232 (EIA-232) Serial Interface 291
- 11.10 Centronics Parallel Interface 294
- 11.11 Analog Output 296
- 11.12 Analog Input 300
- 11.13 Interface to Sensors 303
- 11.14 Interface to Relays 306
- 11.15 Stepper Motor Interface 310
- Summary 315
- Problems 315

## **12 DESIGN AND INTERFACE EXAMPLES IN C 319**

- 12.1 Introduction 319
- 12.2 Hexadecimal Keypad Interface 319
- 12.3 Interface to Multiple 7-Segment LEDs 323

12.4	Interface to Liquid Crystal Displays (LCDs)	325
12.5	Loudspeaker Interface	327
12.6	Nonvolatile RAM Interface	329
12.7	Input/Output Expansion	333
12.8	RS232 (EIA-232) Serial Interface	337
12.9	Centronics Parallel Interface	339
12.10	Analog Output	341
12.11	Analog Input	342
12.12	Interface to Sensors	344
12.13	Interface to Relays	346
12.14	Stepper Motor Interface	347
	Problems	350

## **13** EXAMPLE STUDENT PROJECTS 353

13.1	Introduction	353
13.2	Home Security System	353
	13.2.1 Project Description   13.2.2 System Specifications	
	13.2.3 System Design   13.2.4 Software Design	
13.3	Elevator System	355
	13.3.1 Project Description   13.3.2 System Specifications	
	13.3.3 System Design   13.3.4 Software Design	
13.4	Tic-Tac-Toe	358
	13.4.1 Project Description   13.4.2 System Specifications	
	13.4.3 Software Design	
13.5	Calculator	363
	13.5.1 Project Description   13.5.2 System Specifications	
	13.5.3 Software Design	
13.6	Micromouse	366
	13.6.1 Project Description   13.6.2 System Specifications	
	13.6.3 System Design   13.6.4 Software Design	
13.7	A Soccer-Playing Robot	369
	13.7.1 Project Description   13.7.2 System Specifications	
	13.7.3 System Design   13.7.4 Software Design	
13.8	A Smart Card Application	371
	13.8.1 Basic Security Concepts   13.8.2 Project Description	
	13.8.3 System Specifications   13.8.4 Software Design	
	Summary	373
	Problems	374

## **14 8051 DERIVATIVES 377**

- 14.1 Introduction 377
- 14.2 MCS-151TM™ and MCS-251TM™ 377
- 14.3 Microcontrollers with Flash Memory and NVRAM 377
- 14.4 Microcontrollers with ADCs and DACs 378
- 14.5 High-Speed Microcontrollers 378
- 14.6 Network Microcontrollers 379
- 14.7 Secure Microcontrollers 379
- Summary 379
- Problems 380

## **APPENDICES**

- A Quick Reference Chart 381
- B Opcode Map 383
- C Instruction Definitions 385
- D Special Function Registers 431
- E 8051 Data Sheet 439
- F ASCII Code Chart 455
- G MON51—An 8051 Monitor Program 457
- H A Guide to Keil's  $\mu$ Vision2 IDE 499
- I A Guide to the 8052 Simulator 507
- J The Advanced Encryption Standard 515
- K Sources of 8051 Development Products 521

## **BIBLIOGRAPHY 527**

## **INDEX 529**