

# 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 or computer technology, 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* has been the basis of a fifth semester course for college students in computer engineering. As detailed in Chapter 10, students build 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.

Since 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).

It is my 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.) which 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),

11866

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 over-emphasized. 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).

Some advanced topics relating to programming methods, style, and the development environment are presented in Chapters 8 and 9. These chapters address larger, more conceptual topics important in professional development environments.

Chapter 10 presents several design examples incorporating selected hardware with supporting software. The software is fully annotated and is the real focus in these examples. The second edition includes two additional interfaces; a digital-to-analog output interface using an MC1408 8-bit DAC, and an analog-to-digital input interface using an ADC0804 8-bit ADC. One of the designs in Chapter 10 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-51 during the years that I have taught 8051-based courses to computer engineering students. Shortly after the release of the first edition of this text, URDA, Inc. (Pittsburgh, Pennsylvania) began manufacturing and marketing a PC-board version of the SBC-51. This has proven to be a cost-effective solution to implementing a complete lecture-plus-lab package for teaching the 8051 microcontroller to technology students. Contact URDA at 1-800-338-0517 for more information.

Finally, each chapter contains questions further exploring the concepts presented. This new edition includes 128 end-of-chapter questions—almost double the number in the first edition. A solutions manual is available to instructors from the publisher.

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. I also thank the following persons who reviewed the manuscript and offered invaluable comments, criticism, and suggestions: Antony Alumkal, Austin Community College; Omer Farook, Purdue University—Calumet; David Jones, Lenoir Community College; Roy Seigel, DeVry Institute; and Chandra Sekhar, Purdue University—Calumet.

I. Scott MacKenzie

---

---

# CONTENTS

<b>1</b>	<b>INTRODUCTION TO MICROCONTROLLERS</b>	<b>1</b>
1.1	Introduction	1
1.2	Terminology	3
1.3	The Central Processing Unit	3
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	7
1.6.2	Human Interface Devices	7
1.6.3	Control/Monitor Devices	7
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	10
1.9.2	Applications	10
1.9.3	Instruction Set Features	11
1.10	New Concepts	12
1.11	Gains and Losses: A Design Example	13
	Problems	15
<b>2</b>	<b>HARDWARE SUMMARY</b>	<b>17</b>
2.1	MCS-51 <sup>®</sup> Family Overview	17
2.2	Once Around the Pins	19
2.2.1	Port 0	20
2.2.2	Port 1	20

2.2.3	Port 2	20
2.2.4	Port 3	20
2.2.5	PSEN (Program Store Enable)	21
2.2.6	ALE (Address Latch Enable)	21
2.2.7	EA (External Access)	21
2.2.8	RST (Reset)	21
2.2.9	On-chip Oscillator Inputs	21
2.2.10	Power Connections	22
<b>2.3</b>	<b>I/O Port Structure</b>	<b>22</b>
<b>2.4</b>	<b>Memory Organization</b>	<b>22</b>
2.4.1	General Purpose RAM	23
2.4.2	Bit-addressable RAM	24
2.4.3	Register Banks	26
<b>2.5</b>	<b>Special Function Registers</b>	<b>26</b>
2.5.1	Program Status Word	27
2.5.2	B Register	28
2.5.3	Stack Pointer	29
2.5.4	Data Pointer	29
2.5.5	Port Registers	29
2.5.6	Timer Registers	30
2.5.7	Serial Port Registers	30
2.5.8	Interrupt Registers	31
2.5.9	Power Control Register	31
<b>2.6</b>	<b>External Memory</b>	<b>32</b>
2.6.1	Accessing External Code Memory	33
2.6.2	Accessing External Data Memory	33
2.6.3	Address Decoding	36
2.6.4	Overlapping the External Code and Data Spaces	36
<b>2.7</b>	<b>8032/8052 Enhancements</b>	<b>37</b>
<b>2.8</b>	<b>Reset Operation</b>	<b>38</b>
<b>2.9</b>	<b>Summary</b>	<b>39</b>
	<b>Problems</b>	<b>40</b>

### **3 INSTRUCTION SET SUMMARY**

43

<b>3.1</b>	<b>Introduction</b>	<b>43</b>
<b>3.2</b>	<b>Addressing Modes</b>	<b>43</b>
3.2.1	Register Addressing	44
3.2.2	Direct Addressing	45
3.2.3	Indirect Addressing	46
3.2.4	Immediate Addressing	47
3.2.5	Relative Addressing	47
3.2.6	Absolute Addressing	48

3.2.7	Long Addressing	49
3.2.8	Indexed Addressing	49
<b>3.3</b>	<b>Instruction Types</b>	<b>50</b>
3.3.1	Arithmetic Instructions	50
3.3.2	Logical Instructions	51
3.3.3	Data Transfer Instructions	52
3.3.4	Boolean Instructions	54
3.3.5	Program Branching Instructions	55
<b>Problems</b>		<b>58</b>
<b>4</b>	<b>TIMER OPERATION</b>	<b>63</b>
<b>4.1</b>	<b>Introduction</b>	<b>63</b>
<b>4.2</b>	<b>Timer Mode Register (TMOD)</b>	<b>64</b>
<b>4.3</b>	<b>Timer Control Register (TCON)</b>	<b>66</b>
<b>4.4</b>	<b>Timer Modes and the Overflow Flag</b>	<b>66</b>
4.4.1	13-Bit Timer Mode (Mode 0)	66
4.4.2	16-Bit Timer Mode (Mode 1)	67
4.4.3	8-Bit Auto-Reload Mode (Mode 2)	68
4.4.4	Split Timer Mode (Mode 3)	68
<b>4.5</b>	<b>Clocking Sources</b>	<b>68</b>
4.5.1	Interval Timing	68
4.5.2	Event Counting	68
<b>4.6</b>	<b>Starting, Stopping, and Controlling the Timers</b>	<b>69</b>
<b>4.7</b>	<b>Initializng and Accessing Timer Registers</b>	<b>71</b>
4.7.1	Reading a Timer “On the Fly”	71
<b>4.8</b>	<b>Short Intervals and Long Intervals</b>	<b>72</b>
<b>4.9</b>	<b>8052 Timer 2</b>	<b>76</b>
4.9.1	Auto-Reload Mode	77
4.9.2	Capture Mode	77
<b>4.10</b>	<b>Baud Rate Generation</b>	<b>78</b>
<b>4.11</b>	<b>Summary</b>	<b>78</b>
<b>Problems</b>		<b>79</b>
<b>5</b>	<b>SERIAL PORT OPERATION</b>	<b>81</b>
<b>5.1</b>	<b>Introduction</b>	<b>81</b>
<b>5.2</b>	<b>Serial Port Control Register</b>	<b>81</b>
<b>5.3</b>	<b>Modes of Operation</b>	<b>82</b>
5.3.1	8-Bit Shift Register (Mode 0)	82
5.3.2	8-Bit UART with Variable Baud Rate (Mode 1)	84

5.3.3	9-Bit UART with Fixed Baud Rate (Mode 2)	86
5.3.4	9-Bit UART with Variable Baud Rate (Mode 3)	87
<b>5.4</b>	<b>Initialization and Accessing Serial Port Registers</b>	<b>87</b>
5.4.1	Receiver Enable	87
5.4.2	The 9th Data Bit	87
5.4.3	Adding a Parity Bit	87
5.4.4	Interrupt Flags	88
<b>5.5</b>	<b>Multiprocessor Communications</b>	<b>88</b>
<b>5.6</b>	<b>Serial Port Baud Rates</b>	<b>89</b>
5.6.1	Using Timer 1 as the Baud Rate Clock	90
<b>5.7</b>	<b>Summary</b>	<b>94</b>
	<b>Problems</b>	<b>94</b>
<b>6</b>	<b>INTERRUPTS</b>	<b>97</b>
<b>6.1</b>	<b>Introduction</b>	<b>97</b>
<b>6.2</b>	<b>8051 Interrupt Organization</b>	<b>98</b>
6.2.1	Enabling and Disabling Interrupts	98
6.2.2	Interrupt Priority	99
6.2.3	Polling Sequence	100
<b>6.3</b>	<b>Processing Interrupts</b>	<b>100</b>
6.3.1	Interrupt Vectors	102
<b>6.4</b>	<b>Program Design Using Interrupts</b>	<b>102</b>
6.4.1	Small Interrupt Service Routines	104
6.4.2	Large Interrupt Service Routines	104
<b>6.5</b>	<b>Serial Port Interrupts</b>	<b>107</b>
<b>6.6</b>	<b>External Interrupts</b>	<b>109</b>
<b>6.7</b>	<b>Interrupt Timings</b>	<b>113</b>
<b>6.8</b>	<b>Summary</b>	<b>114</b>
	<b>Problems</b>	<b>115</b>
<b>7</b>	<b>ASSEMBLY LANGUAGE PROGRAMMING</b>	<b>117</b>
<b>7.1</b>	<b>Introduction</b>	<b>117</b>
<b>7.2</b>	<b>Assembler Operation</b>	<b>118</b>
7.2.1	Pass One	119
7.2.2	Pass Two	119
<b>7.3</b>	<b>Assembly Language Program Format</b>	<b>120</b>
7.3.1	Label Field	120
7.3.2	Mnemonic Field	122
7.3.3	Operand Field	122
7.3.4	Comment Field	122

7.3.5	Special Assembler Symbols	122
7.3.6	Indirect Address	123
7.3.7	Immediate Data	123
7.3.8	Data Address	124
7.3.9	Bit Address	124
7.3.10	Code Address	124
7.3.11	Generic Jumps and Calls	124
<b>7.4</b>	<b>Assemble-Time Expression Evaluation</b>	<b>125</b>
7.4.1	Number Bases	126
7.4.2	Character Strings	126
7.4.3	Arithmetic Operators	126
7.4.4	Logical Operators	127
7.4.5	Special Operators	127
7.4.6	Relational Operators	127
7.4.7	Expression Examples	128
7.4.8	Operator Precedence	129
<b>7.5</b>	<b>Assembler Directives</b>	<b>129</b>
7.5.1	Assembler State Control	129
7.5.2	Symbol Definition	130
7.5.3	Storage Initialization/Reservation	132
7.5.4	Program Linkage	135
7.5.5	Segment Selection Directives	137
<b>7.6</b>	<b>Assembler Controls</b>	<b>138</b>
<b>7.7</b>	<b>Linker Operations</b>	<b>140</b>
<b>7.8</b>	<b>Annotated Example: Linking Relocatable Segments and Modules</b>	<b>141</b>
7.8.1	ECHO.LST	141
7.8.2	IO.LST	146
7.8.3	EXAMPLES.M51	147
<b>7.9</b>	<b>Macros</b>	<b>148</b>
7.9.1	Parameter Passing	149
7.9.2	Local Labels	150
7.9.3	Repeat Operations	151
7.9.4	Control Flow Operations	152
<b>Problems</b>	<b>152</b>	

<b>8</b>	<b>PROGRAM STRUCTURE AND DESIGN</b>	<b>155</b>
8.1	Introduction	155
8.2	Advantages and Disadvantages of Structured Programming	157
8.3	The Three Structures	158
8.3.1	Statements	158
8.3.2	The Loop Structure	158
8.3.3	The Choice Structure	165

8.4	Pseudo Code Syntax	171	
8.5	Assembly Language Programming Style	174	
8.5.1	Labels	174	
8.5.2	Comments	175	
8.5.3	Comment Blocks	176	
8.5.4	Saving Registers on the Stack	176	
8.5.5	The Use of Equates	176	
8.5.6	The Use of Subroutines	176	
8.5.7	Program Organization	179	
8.6	Summary	179	
	Problems	179	
<b>9</b>	<b>TOOLS AND TECHNIQUES FOR PROGRAM DEVELOPMENT</b>		<b>181</b>
9.1	Introduction	181	
9.2	The Development Cycle	181	
9.2.1	Software Development	182	
9.2.2	Hardware Development	183	
9.3	Integration and Verification	185	
9.3.1	Software Simulation	185	
9.3.2	Hardware Emulation	187	
9.3.3	Execution from RAM	187	
9.3.4	Execution from EPROM	188	
9.3.5	The Factor Mask Process	188	
9.4	Commands and Environments	189	
9.5	Summary	191	
	Problems	191	
<b>10</b>	<b>DESIGN AND INTERFACE EXAMPLES</b>		<b>193</b>
10.1	Introduction	193	
10.2	The SBC-51	193	
10.3	Hexadecimal Keypad Interface	199	
10.4	Interface to Multiple 7-Segment LEDs	201	
10.5	Loudspeaker Interface	205	
10.6	Non-Volatile RAM Interface	208	
10.7	Input/Output Expansion	215	
10.8	Analog Output	217	
10.9	Analog Input	222	
10.10	Summary	225	
	Problems	225	



**APPENDIXES**

- A Quick Reference Chart 229**
- B OPCODE Map 231**
- C Instruction Definitions 233**
- D Special Function Registers 277**
- E 8051 Data Sheet 287**
- F ASCII Code Chart 303**
- G MON51—An 8051 Monitor Program 305**
- H Sources of 8051 Development Products 347**

**BIBLIOGRAPHY 351**

**INDEX 353**

---

# INTRODUCTION TO MICROCONTROLLERS

## 1.1 INTRODUCTION

Although computers have only been with us for a few decades, their impact has been profound, rivaling that of the telephone, automobile, or television. Their presence is felt by us all, whether computer programmers or recipients of monthly bills printed by a large computer system and delivered by mail. Our notion of computers usually categorizes them as “data processors,” performing numeric operations with inexhaustible competence.

We confront computers of a vastly different breed in a more subtle context performing tasks in a quiet, efficient, and even humble manner, their presence often unnoticed. As a central component in many industrial and consumer products, we find computers at the supermarket inside cash registers and scales; at home in ovens, washing machines, alarm clocks, and thermostats; at play in toys, VCRs, stereo equipment, and musical instruments; at the office in typewriters and photocopiers; and in industrial equipment such as drill presses and phototypesetters. In these settings computers are performing “control” functions by interfacing with the “real world” to turn devices on and off and to monitor conditions. **Microcontrollers** (as opposed to microcomputers or microprocessors) are often found in applications such as these.

It’s hard to imagine the present world of electronic tools and toys without the microprocessor. Yet this single-chip wonder has barely reached its twentieth birthday. In 1971 Intel Corporation introduced the 8080, the first successful microprocessor. Shortly thereafter, Motorola, RCA, and then MOS Technology and Zilog introduced similar devices: the 6800, 1801, 6502, and Z80, respectively. Alone these integrated circuits (ICs) were rather helpless (and they remain so); but as part of a single-board computer (SBC) they became the central component in useful products for learning about and designing with microprocessors. These SBCs, of which the *D2* by Motorola, *KIM-1* by MOS Technology, and *SDK-85* by Intel are the most memorable, quickly found their way into design labs at colleges, universities, and electronics companies.

A device similar to the microprocessor is the microcontroller. In 1976 Intel introduced the 8748, the first device in the MCS-48™ family of microcontrollers. Within a single integrated circuit containing over 17,000 transistors, the 8748 delivered a CPU,