

Contents

Preface xiii

1. Introduction to Microcontrollers 1

- 1.1 Computer System 2
 - 1.1.1 Central Processing Unit 2
 - CPU Components 2
 - 1.1.2 Memory 3
 - 1.1.3 I/O Unit 3
 - 1.1.4 System Bus 4
 - 1.1.5 How does the CPU Read Data from the Memory Chip? 5
 - 1.1.6 What can a Computer do? 5
 - 1.1.7 How does Computer Execute Program Instructions? 5
 - 1.2 Microprocessor, Microcomputer and Microcontroller 5
 - 1.2.1 Microprocessor 5
 - 1.2.2 Microcomputer 6
 - 1.2.3 Microcontroller 6
 - 1.2.4 Comparison between Microprocessor and Microcontroller 7
 - 1.3 Classification of Microcontrollers 7
 - 1.3.1 Word Length: 4, 8, 16, 32, 64-bit Microcontrollers 7
 - 1.3.2 Memory Architecture: Von Neumann and Harvard Architectures 8
 - 1.3.3 Core Architecture: Microcoded and Hardwired Designs 8
 - 1.3.4 Instruction Set Architectures: CISC and RISC 9
 - 1.4 Choosing a Microcontroller 10
 - 1.5 Applications of Microcontrollers 10
 - 1.6 History and Introduction to The 8051 Microcontroller Family 11
 - 1.7 Overview of The 8051 Family 11
 - 1.7.1 Features of the 8051 (MCS 51) Family 11
 - 1.7.2 8051 Variants and Enhancements 12
 - 1.7.3 Comparison between MCS 51, PIC, AVR and HCS11/12 Families 13
 - 1.7.4 Advantages of using 8051 Family of Microcontrollers 13
 - 1.8 Embedded Systems 15
 - Embedded Microcontrollers 15
- Points to Remember* 15
Objective Questions 16
Review Questions with Answers 17
Exercise 18

2. Programming Model and Architecture of the 8051 19

- 2.1 The 8051 Architecture 20
 - ALU 21
 - Memory 21
 - Peripherals 21
 - Timing and Control Unit 21
 - Oscillator 21
 - 2.2 Programming Model of The 8051 21
 - 2.3 On-Chip Memory Organization 22
 - Special Function Registers (SFRs) 22
 - Internal RAM 23
 - Internal ROM 23
 - 2.3.1 Special Function Registers (SFRs) 23
 - Accumulator: A 23
 - B 23
 - PSW 23
 - Program Counter: PC 24
 - Data Pointer: DPTR 24
 - Stack Pointer: SP 25
 - I/O Port Registers (latches): P0, P1, P2 and P3 25
 - Peripheral Data Registers: TL0, TH0, TL1, TH1, and SBUF 25
 - Peripheral Control Registers: IP, IE, TMOD, TCON, SCON, and PCON 25
 - 2.3.2 Internal RAM 25
 - Register Banks 25
 - Bit Addressable Memory 27
 - General-Purpose RAM 28
 - 2.3.3 Internal ROM 29
 - 2.4 External Memory Organization 29
- Points to Remember* 30
Objective Questions 31
Review Questions with Answers 32
Exercise 33

3. Program Development Process and Tools 34

- 3.1 Programming Language 35
 - 3.1.1 Machine Language 35
 - 3.1.2 Assembly Language 35
 - 3.1.3 High-Level Language 35
 - 3.1.4 Comparison between Programming Languages 36
 - 3.1.5 Why Assembly Language? 36
- 3.2 Assembly Language Structure 37
 - 3.2.1 Label 37

3.2.2	Instructions	37
	Instruction Size	37
3.2.3	Comments	38
3.3	Assembly-Language-Program Example	38
3.4	Program Execution Process	39
3.5	Software and Hardware Development Tools	39
3.5.1	Design and Documentation Tools	39
	Flowcharts	40
	Pseudo-codes	40
3.5.2	Software Development Tools	40
	Editor	40
	Assembler	41
	Cross Assembler	42
	Compiler	42
	Linker	42
	Simulator	42
	Debugger	42
3.5.3	Hardware Development Tools	43
	Emulator or In Circuit	
	Emulator (ICE)	43
	Logic Analyzer	43
	Target System	43
3.6	Integrated Development Environments (IDE)	43
3.7	Assembler Directives	43
	Originate-ORG	43
	Define Byte-DB	44
	Define Word-DW	44
	Equate-EQU	44
	END	44
3.8	Program Development Process	45
	Create Source Code	45
	Assemble all Source Code Files	45
	Link all the Object Code Files	45
	Test the Program for Correctness	46
3.9	Loading Program into Microcontroller	47
	Parallel Programming	47
	Serial Programming	47
3.10	The Intel Hex File Format	48
	<i>Points to Remember</i>	49
	<i>Objective Questions</i>	50
	<i>Review Questions with Answers</i>	51
	<i>Exercise</i>	52
4.	Addressing Modes and Data	
	Movement Instructions	53
4.1	Why Data Movement Instructions First?	54
4.2	Addressing Modes	54
	Acronyms used in the Instructions	55
4.2.1	Immediate Addressing Mode	55
	Notations for Numbers of Different	
	Number Systems	56
	Use of Expressions	56
4.2.2	Register Addressing Mode	56
4.2.3	Direct Addressing Mode	57
4.2.4	Indirect Addressing Mode	58
	Register Indirect Addressing Mode	58
	Indexed Addressing Mode	59
4.3	Operand Modifiers: # and @	60
4.4	External Memory Data Movements	61
4.4.1	Data Memory Access	61
4.4.2	Program Memory Access	62
4.5	Data Exchange	63
4.6	Push and Pop Instructions	63
	<i>Points to Remember</i>	65
	<i>Objective Questions</i>	67
	<i>Review Questions with Answers</i>	68
	<i>Exercise</i>	69
5.	Arithmetic and Logical Instructions	70
5.1	Arithmetic Instructions	71
5.1.1	Addition	71
	Addition of Multi-Byte Numbers	72
5.1.2	Subtraction	74
5.1.3	Signed Arithmetic	75
	Positive Numbers	75
	Negative Numbers	75
	Overflow	77
	Addition of Unlike Signed Numbers	77
	Addition of Like Signed Numbers	78
	Subtraction of Unlike Signed Numbers	80
	Subtraction of Like Signed Numbers	81
	Recovering a Result from Overflow	82
5.1.4	Decimal (Binary Coded Decimal—	
	BCD) Arithmetic	83
	DA A Decimal Adjust Accumulator	
	for Addition	84
5.1.5	Multiplication	85
	Use of OV in Multiplication	85
5.1.6	Division	86
	Use of OV in Division	86
5.1.7	Increment and Decrement	86
5.2	Logical Instructions	89
5.2.1	Byte Operations	89
	AND Operation	89
	OR Operation	90
	EX-OR Operation	90
	Logical Operations with Ports	91
5.2.2	Unary Operations	92
	Summary of Arithmetic and	
	Logical Instructions	94
	<i>Points to Remember</i>	95
	<i>Objective Questions</i>	96
	<i>Review Questions with Answers</i>	97
	<i>Exercise</i>	98

6. Bit-Processing Instructions 99

- 6.1 Bit Addressability 100
- 6.2 Bit-Addressable Memories 100
 - 6.2.1 Bit-Addressable Internal RAM 100
 - 6.2.2 Bit-Addressable Special Function Registers 101
- 6.3 Bit-Processing Instructions 104
 - 6.3.1 Instruction using Carry 104
 - 6.3.2 Other Instructions 105
 - 6.3.3 Conditional Jump Instructions 107
- Summary of Bit-Processing Instructions 108
- Points to Remember* 108
- Objective Questions* 109
- Review Questions with Answers* 109
- Exercise* 110

7. Program-Flow Control Instructions 111

- 7.1 Jump Instructions 112
 - 7.1.1 Unconditional Jumps 112
 - Short Jump 112
 - Absolute Jump 114
 - Long Jump 115
 - 7.1.2 Conditional Jumps 117
 - Bit Jumps 117
 - Byte Jumps 118
- 7.2 Loops 119
 - Nested Loops 121
- 7.3 Calls and Subroutines 122
 - Relation of Calls and Stack 122
 - Cautions while Developing Subroutines 125
- 7.4 Stack Initialization and Overflow 127
- 7.5 Time-Delay Generation using Software 127
 - NOP (No Operation) 127
- Summary of Program-Flow Control Instructions 129
- Points to Remember* 130
- Objective Questions* 130
- Review Questions with Answers* 131
- Exercise* 133

8. Look-Up Tables and Jump Tables 134

- 8.1 Look-Up Tables and their Usage 135
- 8.2 Faster Evaluation of Functions 135
- 8.3 Miscellaneous Conversions 136
- 8.4 The 8051 and Look-Up Tables 136
- 8.5 Jump Tables 140
- Points to Remember* 141
- Objective Questions* 141
- Exercise* 142

9. Code Conversions, Array Processing and 16 Bit Arithmetic 143

- 9.1 Code Conversions 144

- 9.2 Array Processing 148

- 9.3 16 Bit Operations 153

- 9.4 Other Programs 155

Points to Remember 158

Exercise 158

10. Timing and Instruction Execution 159

- 10.1 The Clock Pulse 160
- 10.2 Machine Cycle 160
- 10.3 Instructions Timing 160
 - 10.3.1 1 Byte–1 Machine Cycle Instructions 161
 - 10.3.2 2 Byte–1 Machine Cycle Instructions 161
 - 10.3.3 1 Byte–2 Machine Cycle Instructions 161
 - 10.3.4 2 Byte–2 Machine Cycle Instructions 162
- 10.4 External Memory Access 162
 - 10.4.1 External RAM (Data Memory) Access 162
 - 10.4.2 Data Memory Read/Write Cycle 163
 - 10.4.3 External ROM (Code Memory) Access 164
- 10.5 8051 Instruction Execution 165
 - 10.5.1 MOV A, Operand 165
 - 10.5.2 ADD A, Operand 166
 - 10.5.3 LCALL Label 169
 - 10.5.4 MOVX A, @ DPTR 169

Points to Remember 169

Objective Questions 171

Review Questions with Answers 171

Exercise 172

11. The 8051 Hardware, System Design and Troubleshooting 173

- 11.1 The 8051 Pin Diagram 174
- 11.2 The 8051 Pin Description 174
- 11.3 Power Consumption Control of The 8051 178
 - Idle Mode 178
 - Power Down Mode 178
 - Tip for Power Saving 178
- 11.4 Design of The 8051 based System 179
 - Reset Circuit 179
 - Clock Circuit 179
 - Pull-up Resistors 179
 - Demultiplexer 179
- 11.5 Troubleshooting 8051 based Systems 179
- 11.6 Program Memory Protection 180
- Points to Remember* 180
- Objective Questions* 181
- Review Questions with Answers* 181
- Exercise* 182

12. The 8051 Programming in C 183

- 12.1 Data Types for The 8051 Supported by Cx51 Compiler 184
 - 12.1.1 Native Word Size: char 184

12.1.2	Additional Data Types for the 8051	185		
	Bit	185		
	sfr	185		
	sbit	185		
12.1.3	Implementing Infinite Loops in a C Program	188		
12.1.4	Bit Addressing in the C Language	191		
12.2	Accessing Memory Areas of The 8051	195		
	12.2.1 Internal RAM (Data Memory)	195		
	12.2.2 External RAM (Data Memory)	196		
	12.2.3 Program/Code Memory	196		
12.3	Bit Addressable Variables	196		
12.4	Interrupt-Service Routines	197		
	‘using’ attribute	197		
12.5	Operators	197		
12.6	Data Serialization using Port Pins	201		
12.7	Rotate Operations in C	202		
	12.7.1 Left Rotation	203		
	12.7.2 Right Rotation	203		
12.8	Pointers	204		
12.9	Pointers to Absolute Addresses	205		
12.10	Time Delays in C	206		
12.11	Increasing the Code Efficiency	207		
	12.11.1 Variable Size	207		
	12.11.2 Use of Unsigned Data Types	207		
	12.11.3 Use of Bit Variables	207		
	12.11.4 Inline Functions	207		
	12.11.5 Use of Internal RAM	207		
	12.11.6 Inline Assembly/Hand-Coded Assembly	207		
	12.11.7 Avoid Standard Library Routines	207		
	12.11.8 Use of Intrinsic Functions	207		
12.12	Performance Comparison between Assembly and C Programs	207		
	<i>Points to Remember</i>	212		
	<i>Objective Questions</i>	212		
	<i>Review Questions with Answers</i>	213		
	<i>Exercise</i>	214		
13.	Input/Output Ports	215		
13.1	The 8051 Ports	216		
	13.1.1 Port 1	216		
	Configuring the Port as an Input	217		
	Configuring the Port as an Output	217		
	13.1.2 Port 0	219		
	Port 0 as an Address/Data Bus	219		
	13.1.3 Port 2	220		
	13.1.4 Port 3	221		
13.2	Reading Latch Versus Reading Port Pin	225		
	Logical Operation with Ports	226		
13.3	Port Current Capabilities	227		
	<i>Points to Remember</i>	227		
	<i>Objective Questions</i>	227		
	<i>Review Questions with Answers</i>	228		
	<i>Exercise</i>	229		
14.	Timers	230		
14.1	Need of Timers	231		
14.2	How Does a Timer Operate?	231		
14.3	Timers in The 8051	231		
	14.3.1 TMOD (Timer Mode Control) Register	232		
	Gate	232		
	C/T	233		
	M1 and M0	234		
	14.3.2 TCON Register	235		
14.4	Timer Circuits as an Interval Timer	236		
	14.4.1 Timer Mode 0	236		
	14.4.2 Timer Mode 1	236		
	Operation of the Timer in Mode 1	236		
	Initial Value to be Loaded in Timer Registers	237		
	Square-Wave Generation using Timers	238		
	Timer Mode 0	243		
	14.4.3 Timer Mode 2	244		
	Operation of Mode 2	245		
	14.4.4 Generating Larger Delays	247		
	14.4.5 Timer Mode 3	250		
	14.4.6 Reading the Value of a Timer	250		
14.5	Timer as an Event Counter	252		
	Simulation Procedure	253		
14.6	Frequency Measurement using Timers	254		
	<i>Points to Remember</i>	256		
	<i>Objective Questions</i>	256		
	<i>Review Questions with Answers</i>	257		
	<i>Exercise</i>	258		
15.	Serial Communications	259		
15.1	Need for the Serial Communication	260		
15.2	Synchronous and Asynchronous Serial Communications	260		
	15.2.1 Format of Asynchronous Serial Data Frame	261		
	15.2.2 Rate of Data Transfer	262		
15.3	RS 232 : Serial Data Transmission Standard	262		
	15.3.1 Hand-shaking Process between DTE and DCE	262		
	15.3.2 RS232 to TTL Interfacing	263		
15.4	UART	264		
	15.4.1 UART Features	265		
	15.4.2 SBUF (Serial Data Buffer) Register—One Name—Two Registers	265		
	SCON—Serial Control Register	265		

15.4.3	Serial Port Control (SCON) Register	265			
15.5	Modes of Operation	266			
15.5.1	Mode 0—8 bit Shift Register Mode	266			
	Transmission	266			
	Reception	266			
	Baud Rate for Mode 0	269			
15.5.2	Mode 1—Standard 8-bit UART Mode	269			
	Transmission	269			
	Reception	270			
	Baud Rate for Mode 1	270			
	Software Development to Transmit and Receive Data Serially	272			
	Simulation Result	274			
	Simulation Procedure	275			
15.5.3	Mode 2—Multiprocessor Communication	280			
	Transmission	281			
	Reception	281			
	Multiprocessor Communication	281			
	Baud Rate for Mode 2	282			
15.5.4	Mode 3	282			
15.6	Second Serial Port in The DS89C4x0	283			
	<i>Points to Remember</i>	285			
	<i>Objective Questions</i>	285			
	<i>Review Questions with Answers</i>	287			
	<i>Exercise</i>	288			
16.	Interrupts	289			
16.1	Need of Interrupts	290			
16.1.1	How are Interrupts Useful?	290			
16.2	Interrupts in The 8051	290			
16.2.1	Reset as a Special Interrupt	292			
16.3	Interrupt Handling and Execution	292			
16.4	Programming the Interrupts	293			
16.4.1	Interrupt Enable (IE) Register	293			
16.4.2	Interrupt Priority (IP) Register	294			
16.5	Timer Interrupts	296			
16.5.1	Programming of Timer Interrupts	296			
16.5.2	Simultaneous and Independent use of both the Timers	297			
16.6	External Interrupts	304			
16.6.1	Level-Triggered External Interrupts	304			
16.6.2	Transition (Edge) Triggered External Interrupts	304			
	Simulation Procedure	305			
16.6.3	Pulse Generation using External Interrupt	306			
16.6.4	Sampling of Edge-Triggered Interrupts	306			
16.7	Serial Port Interrupts	307			
	Simulation Procedure	308			
	Simulation Result	310			
16.8	Interrupt Priorities	312			
16.9	Nested and Multiple Interrupts	312			
16.10	Blocking Conditions	313			
16.11	Interrupt Latency	314			
16.12	Generating Interrupts using Instructions	315			
16.13	Cautions While Developing Interrupt Service Routines	315			
16.14	Dilemma: Use Interrupt or Polling?	315			
16.15	Project: Full-Duplex System	316			
	Problem Statement	316			
	Program Development	317			
	Suggested Modification	318			
	<i>Points to Remember</i>	318			
	<i>Objective Questions</i>	318			
	<i>Review Questions with Answers</i>	320			
	<i>Exercise</i>	320			
17.	Interfacing Keyboards	322			
17.1	Keyboard Design Considerations	323			
17.1.1	Mechanical Properties of the Switches	323			
17.1.2	Key Debouncing using Hardware	324			
17.1.3	Key Debouncing using Software	324			
17.2	Keyboard Configurations	324			
17.2.1	Simple Keyboard Configuration (Using I/O Pins directly)	324			
	Advantages of a Simple Keyboard	324			
	Disadvantages of Simple Keyboard	325			
	Key-Press Detection and the Code Generation	325			
	Algorithm	325			
	Program for a Simple Keyboard	325			
	Key Identification using the Hardware Technique	327			
	Advantage of Hardware Technique	327			
17.2.2	Matrix Keyboard Configuration	328			
	Key-Code Generation	328			
	Key Identification and Key Code Generation using Counters	329			
	Key Identification and Key-Code Generation using the Look-Up Table	331			
	<i>Points to Remember</i>	331			
	<i>Objective Questions</i>	332			
	<i>Review Questions with Answers</i>	332			
	<i>Exercise</i>	332			
18.	Interfacing Display Devices: LED, Seven Segment Display and LCD	333			
18.1	Light Emitting Diodes	334			
	Applications of LEDs	336			
18.2	Seven-Segment Display	337			

18.2.1	Segment Multiplexing within one Seven-Segment Display	337	19.2	Digital-to-Analog Converters	373
18.2.2	Digit Multiplexing	337	19.2.1	DAC Parameters	373
18.3	Liquid Crystal Display (LCD)	340	19.2.2	Common DAC chips	374
18.3.1	Pin Description for LCD	340	19.2.3	DAC AD557 Chip	374
18.3.2	LCD Commands	341	19.2.4	DAC 0808 Chip	378
18.3.3	Initialization of the LCD using the Internal Reset Circuit	341		Operation of DAC0808	378
18.3.4	Software Initialization of the LCD	341	19.2.5	Applications of DACs	381
18.3.5	LCD Timing	341	19.3	Temperature Sensor: LM35	382
18.3.6	Modes of operation	341	19.3.1	LM35 Specifications	382
	8-bit Mode	344	19.3.2	Common Temperature Sensors	383
	Importance of Monitoring Busy Flag	348	19.3.3	Applications of Temperature Sensors	383
	4-bit Mode	350	19.4	Infrared (IR) Sensors	384
18.4	Project: Simple Burglar Alarm System	350	19.4.1	TSOP 17xx IR Receivers	384
	Program Development	351	19.4.2	Interfacing of TSOP 17xx with the 8051	385
	Suggested Modifications	353	19.4.3	Applications of IR Sensors	385
	<i>Points to Remember</i>	353		Project: <i>Temperature Monitoring System</i>	385
	<i>Objective Questions</i>	353		Project: <i>Function Generator</i>	389
	<i>Review Questions with Answers</i>	354		<i>Points to Remember</i>	393
	<i>Exercise</i>	354		<i>Objective Questions</i>	394
				<i>Review Questions with Answers</i>	395
				<i>Exercise</i>	395
19.	Interfacing ADC, DAC and Sensors	355	20.	Interfacing Relays, Opto-Couplers, Stepper and DC Motors	396
19.1	Analog to Digital Converters	356	20.1	Relays	397
19.1.1	Need for Analog-to-Digital Converters	356	20.1.1	Relay Operation	397
19.1.2	Methods of Conversion	356	20.1.2	Relay Driver Circuits and Interfacing	397
19.1.3	ADC Parameters	356		Driver Circuit Operation	398
19.1.4	Common ADC Chips	357		Advantages of using a Relay	399
19.1.5	ADC 0801/02/03/04/05 Chips	358		Drawbacks of Electromechanical Relays	399
	Analog inputs	358	20.1.3	Parameters of Relays	399
	Analog Input Voltage Range	358		Input Coil Side (Low Power, Controlling Side)	399
	Digital Output	359		Output Switch Side (High Power, Load Side)	399
	Clock Source	359	20.2	Opto-Coupler	400
19.1.6	Handshaking Process between the Microcontroller and ADC 0804 Chip	360	20.2.1	Opto-Coupler Operation	400
19.1.7	ADC 0808/0809 Chips	362	20.2.2	Applications of Opto-Couplers	401
19.1.8	Serial ADC Chips	364	20.3	Stepper Motors	401
	Serial ADC Chip MAX1112/ MAX1113	364	20.3.1	Permanent-Magnet Stepper Motors	401
	Pin Description of MAX1112/1113	365	20.3.2	Unipolar and Bipolar Stepper Motors	402
	Control Byte of MAX1112/ MAX1113	366	20.3.3	Direction and Speed Control	404
	Sending Control Byte to MAX 1112/ MAX1113	367	20.3.4	Interfacing with the 8051	404
	Reading Digital Output Data (Result) from the Serial ADC Chip	367		Circuit Operation	405
	Common Serial ADC Chips	370	20.3.5	Rotation of Motor for Specified Angle	408
19.1.9	On-chip ADCs	371	20.3.6	Applications of Stepper Motors	409
	P89LPC768	371	20.4	DC Motors	409
19.1.10	Applications of ADCs	373	20.4.1	Analog Speed Control	409
			20.4.2	Digital Speed Control	410
			20.4.3	Direction Control	410

20.4.4	Pulse-Width Modulation (PWM)	412
20.4.5	DC Motor Driver Circuits	412
20.4.6	Interfacing DC Motors with the 8051	414
Project: <i>DC Motor-Speed-Control System</i>		416
Project: <i>Automatic Street Light Control System</i>		420
<i>Points to Remember</i>		421
<i>Objective Questions</i>		422
<i>Review Questions with Answers</i>		422
<i>Exercise</i>		423

21. Interfacing External Memory and Real-Time Clock 424

21.1	Memory Interfacing and its Need	425
21.2	Memory Chips	425
21.2.1	Address Signals	425
21.2.2	Data Signals	426
21.2.3	Control Signals	426
21.3	Semiconductor Memory Devices	426
21.3.1	Volatile Memory	426
	SRAM: Static RAM	427
	DRAM: Dynamic RAM	427
21.3.2	Nonvolatile Memory	427
	ROM	427
	PROM: Programmable ROM	428
	EPROM: Erasable Programmable ROM	428
	EEPROM: Electrically Erasable and Programmable ROM	428
	Flash	428
	NVRAM: Nonvolatile RAM	428
21.4	Memory Map and Address Decoding	429
21.4.1	Signals used in Memory Interfacing	429
21.4.2	The 8051 and the Corresponding Memory Chip Signals	429
	Address Lines ($A_{15}-A_0$)	429
	Data Lines (D_7-D_0)	429
21.4.3	Address Decoder using Logic Gates	430
	Memory Read and Write Operations	431
21.4.4	Address Decoder using Decoder Chip	431
21.5	Program/Code Memory Interfacing	433
21.6	Data/RAM Memory Interfacing	435
21.7	Data Memory Using ROM	437
21.8	ROM as Data as well as Program Memory	437
21.9	RAM as Data as well as Program Memory	438
21.10	On-Chip EEPROM Programming in AT89S8253	438
	Steps to Write (Program) a Byte in Data EEPROM	440
	Steps to Write a Page (Maximum 32 Bytes) in Data EEPROM	441
	Steps to Read a Byte(s) from Data EEPROM	441
21.11	Real-Time Clock	441

21.11.1	DS12887: Real-time Clock Chip	443
	Main Features of the DS12887	443
21.11.2	Address Map	443
21.11.3	Interfacing DS12887 with the 8051	443
21.11.4	Programming the DS12887	444
	Oscillator Control Bits	444
	Setting the Time and Date	445
	Reading the Time and Date	447
21.11.5	Square Wave Output	448
21.11.6	Alarms	448
21.11.7	Periodic Interrupts	450
21.11.8	Update Cycles	450
21.11.9	Interrupt Sources	450

Points to Remember 450

Objective Questions 450

Review Questions with Answers 451

Exercise 452

22. I2C and SPI Protocols 453

22.1	Inter Integrated Circuit (IIC or I ² C)	454
	I2C Bus Features	455
22.2	I2C Bus Hardware Configuration	454
22.2.1	V_{CC} or V_{DD}	455
22.2.2	I2C Devices	456
22.3	I2C Protocol	455
22.3.1	START and STOP Conditions	456
22.3.2	Data Validity	457
22.3.3	Data Transfer Operations on I2C Bus	457
22.3.4	Write Operation	457
22.3.5	Read Operation	459
22.3.6	Arbitration and the Clock Synchronization	460
22.3.7	Clock Stretching	460
22.3.8	Burst Read/Write Modes	461
22.4	Driving The I2C Bus	460
22.4.1	I2C Interface Module of P89C66x Microcontrollers	462
22.4.2	Programming I2C Interface of P89C66x	463
	Using I2C Interface as a Master	463
	Initialization of I2C Module	463
	Generate START Condition	463
	Transmit Data	464
	Read Data	464
	Generate a STOP Condition	464
22.4.3	Interfacing PCF8594C-2 Serial EEPROM	466
	Addressing of PCF8594C-2	467
	Write Operation	467
	Read Operation	467
	24xxx and AT24Cxxxx Serial EEPROMs	467
	Using I2C Device as a Slave	470

Initialization of I2C Module	470
Monitor the Bus Condition	470
Transmit Data	470
Read Data	470
22.5 I2C Devices	469
22.6 Serial Peripheral Interface	470
22.6.1 SPI Operation	471
22.6.2 Clock Polarity and Phase in SPI Device	472
22.6.3 SPI Bus Configurations	473
22.7 AT89S825x	472
22.7.1 SPI Interface Module of AT89S825x Microcontrollers	474
22.7.2 Interfacing MAX512/13 with SPI Bus	477
Serial-Input Data Format and Control Codes for MAX512/13	478
22.7.3 Interfacing MAX512/13 with AT89S8253	478
22.8 SPI Devices	479
22.9 Comparison between I2C and SPI Protocols	479
<i>Points to Remember</i>	479
<i>Objective Questions</i>	480
<i>Review Questions with Answers</i>	481
<i>Exercise</i>	482
23. The 8051 Variants, AVR and PIC Microcontrollers	483
23.1 The 8051 Enhancements	484
23.1.1 Additional 128 Bytes of On-Chip RAM	485
23.1.2 Timer 2	485
23.1.3 Maximum Clock Speed	485
Clocks/Machine Cycle	485
23.1.4 Program Memory Identification	485
23.2 8051 Variants from NXP (Philips)	485
23.3 8051 Variants from Atmel Corporation	485
23.4 8051 Variants from Dallas Semiconductor	485
23.5 8051 Variants from Silicon Laboratories	486
23.6 Common On-Chip Peripherals	486
23.6.1 Watchdog Timer	487
23.6.2 Controller Area Network (CAN)	488
23.6.3 Analog Comparator	488
23.6.4 Pulse-width Modulator	488
23.6.5 ADC and DAC	488
23.6.6 Real-Time Clock (RTC)	488
23.6.7 Other Peripherals and Features	488
Multiple DPTRs	489
Four Levels of Interrupt Priorities	489
In-System Programming (ISP)	489
In-Application Programming (IAP)	489
23.7 MCS 151/251 Microcontrollers	488
23.8 MCS 96 Microcontrollers	489
23.9 AVR Microcontrollers	490
23.9.1 AVR ATmega Family	493
23.9.2 Programming Model of ATmega16 Family of Microcontrollers	493
General-Purpose Registers: R0–R31	493
STATUS Register	494
Program Counter	495
Stack Pointer	495
Data Memory	495
Data EEPROM	495
Program Memory	496
I/O Ports and Peripherals	496
23.10 PIC Microcontrollers	495
23.10.1 PIC18 Family	497
23.10.2 Programming Model of PIC18 Family of Microcontrollers	498
Working Register (WREG)	498
Bank Select Register (BSR)	499
File Select Registers (FSR)	499
STATUS Register	499
Table Pointer	500
Program Counter	500
Stack and Stack Pointer	500
Special Function Registers (SFRs)	500
Data Memory	500
Program Memory	501
Data EEPROM Memory	501
I/O Ports	502
<i>Points to Remember</i>	501
<i>Objective Questions</i>	501
<i>Review Questions with Answers</i>	502
<i>Exercise</i>	502
Appendix A: The 8051 Instruction Set Summary	503
Appendix B: Using Keil μ Vision 4.0 IDE	525
Appendix C: Instructions Arranged Functionally	540
Appendix D: ASCII Codes	549
Appendix E: Special Function Registers Quick View	551
Index	555