Chapter 1 : Introduction to Microprocessors **1-1 to 1-12**

✓ **Syllabus Topic :** Microprocessor, Microprocessor Instruction Set **1-1**
1.1 Basics of Microprocessors and its Instruction Set 1-1
1.1.1 Microprocessor Characteristics 1-3
✓ **Syllabus Topic :** Computer Languages, From Large Computers to Single-Chip Microcontrollers **1-4**
1.1.2 Computer Languages, From Large Computers to Single-Chip Microcontrollers 1-4
✓ **Syllabus Topic :** Microprocessor Architecture and Its Operation **1-4**
1.2 Microprocessor Architecture and Its Operation 1-4
✓ **Syllabus Topic :** Microcomputer System, Memory, I/O Devices **1-6**
1.3 Memory and I/O Device Accessing 1-6
1.3.1 Address Bus 1-6
1.3.2 Data Bus 1-8
1.3.3 Control Bus 1-8
✓ **Syllabus Topic :** Microcomputer System, Logic Devices and Interfacing **1-9**
1.4 Microcomputer System, Logic Devices and Interfacing **(April 2017)** 1-9
✓ **Syllabus Topic :** Applications of Microprocessors and Microprocessor-based System **1-10**
1.5 Applications of Microprocessor Microprocessor-based System 1-10
1.6 Comprehensive University Question and Answer 1-10

Chapter 2 : Introduction to 8085 **2-1 to 2-27**

✓ **Syllabus Topic :** Introduction **2-1**
2.1 8085 Features 2-1
✓ **Syllabus Topic :** 8085 Microprocessor Unit **2-2**
2.2 8085 Microprocessor Architecture (Functional Block Diagram) **(April 2017)** 2-2
2.2.1 Register Organization 2-8
2.2.1(A) Temporary Registers 2-8
2.2.1(B) General Purpose Registers 2-9
2.2.1(C) Special Purpose Registers 2-10
2.2.2 Interrupt Control 2-14
2.2.3 Serial I/O Control Group 2-14
2.2.4 Timing and Control 2-14
2.3 8085 Microprocessor Pin-Out and Signals 2-15
2.4 Address, Data and Control Buses 2-21
2.5 Demultiplexing Address and Data Bus 2-22
2.6 Generation of Control Signals 2-23
2.7 Reset Circuit 2-25
✓ **Syllabus Topic :** 8085 – Based Microcomputer **2-26**
2.8 8085 based Microcomputer (Minimum System of 8085) 2-26

2.9 Comprehensive University Question and Answer 2-27

Chapter 3 : Introduction to Programming **3-1 to 3-13**

✓ **Syllabus Topic :** The 8085 Programming Model **3-1**
3.1 The 8085 Programmer's Model **(April 2017)** 3-1
✓ **Syllabus Topic :** Instruction Classification **3-3**
3.2 Instruction Classification **(April 2017)** 3-3
✓ **Syllabus Topic :** Instruction **3-4**
3.3 Introduction to Programming 3-4
3.4 Timing Diagram 3-5
3.4.1(A) Instruction Cycle 3-5
3.4.1(B) Machine Cycle 3-6
3.4.1(C) T State 3-6
3.5 8085 Machine Cycles and their Timings 3-7
3.5.1 OPCODE Fetch Cycle 3-7
3.5.2 Operand Fetch or Memory Read Cycle **(April 2017)** 3-8
3.5.3 Memory Write Cycle 3-9
3.5.4 I/O Read and I/O Write Machine Cycles 3-9
✓ **Syllabus Topic :** Data Storage **3-10**
3.6 Addressing Modes / Data Storage methods 3-10
3.7 Comprehensive University Questions and Answers 3-13

Chapter 4 : Overview of 8085 Instructions Set **4-1 to 4-52**

✓ **Syllabus Topic :** Overview of Instruction Set **4-1**
4.1 Introduction 4-1
✓ **Syllabus Topic :** Data Transfer Operations **4-1**
4.2 Data Transfer Instructions **(April 2017)** 4-1
4.2.1 MOV Rd, Rs 4-1
4.2.2 MOV R, M 4-2
4.2.3 MOV M, R 4-2
4.2.4 MVI R, Data 4-3
4.2.5 MVI M, Data 4-3
4.2.6 LXI Rp, 16-bit Data 4-4
4.2.7 LDA Address 4-4
4.2.8 STA Address **(April 2017)** 4-5
4.2.9 LHLD Address **(April 2017)** 4-5
4.2.10 SHLD Address **(April 2017)** 4-6
4.2.11 LDAX Rp 4-7
4.2.12 STAX Rp 4-7
4.2.13 XCHG **(April 2017)** 4-8
✓ **Syllabus Topic :** Writing Assembling and Executing Simple Programs **4-8**

4.3 Writing, Assembling and Executing Simple Programs 4-8
✓ **Syllabus Topic :** Arithmetic Operations : Arithmetic Instructions Related to Register, Memory and Immediate Data **4-11**
4.4 Arithmetic Instructions related to Register, Memory and Immediate Data 4-11
4.4.1 ADD R 4-11
4.4.2 ADD M 4-11
4.4.3 ADC R 4-12
4.4.4 ADC M 4-12
4.4.5 ADI Data **(April 2017)** 4-13
4.4.6 ACI Data 4-13
✓ **Syllabus Topic :** Additional Data Transfer and 16-bit Arithmetic Instructions **4-14**
4.5 Additional Data Transfer and 16-bit Arithmetic Operations 4-14
4.5.1 DAD R_p 4-14
4.5.2 SUB R 4-15
4.5.3 SUB M 4-15
4.5.4 SBB R **(April 2017)** 4-16
4.5.5 SBB M 4-17
4.5.6 SUI Data 4-18
4.5.7 SBI Data 4-19
4.5.8 DAA 4-20
4.5.9 INR R 4-21
4.5.10 DCR R 4-21
4.5.11 INR M 4-22
4.5.12 DCR M 4-22
4.5.13 INX R_p 4-23
4.5.14 DCX R_p 4-23
✓ **Syllabus Topic :** Writing and Assembling Program **4-24**
4.6 Writing and Assembling Programs 4-24
✓ **Syllabus Topic :** Logic Operation, Logic Operation : Rotate, Compare **4-29**
4.7 Logical Group Instructions : Logical Instructions, Compare, Rotate and Shift Instructions 4-29
4.7.1 ANA R 4-29
4.7.2 ANA M 4-30
4.7.3 ANI Data 4-30
4.7.4 ORA R 4-31
4.7.5 ORA M 4-32
4.7.6 ORI Data **(April 2017)** 4-32
4.7.7 XRA R **(April 2017)** 4-33
4.7.8 XRA M 4-34
4.7.9 XRI Data 4-35
4.7.10 CMA 4-35

4.7.11 CMC 4-36
4.7.12 STC 4-36
4.7.13 CMP R 4-37
4.7.14 CMP M 4-38
4.7.15 CPI Data 4-39
4.7.16 RLC **(April 2017)** 4-40
4.7.17 RRC **(April 2017)** 4-41
4.7.18 RAL **(April 2017)** 4-42
4.7.19 RAR **(April 2017)** 4-43
✓ **Syllabus Topic :** Writing Assembly Languages Programs, Debugging a Program **4-43**
4.8 Programs 4-43
4.9 Comprehensive University Questions and Answers 4-52

Chapter 5 : Stack Control and Branching Group **5-1 to 5-61**

✓ **Syllabus Topic :** Stack **5-1**
5.1 Introduction to Stack **(April 2017)** 5-1
5.2 Stack Related Instructions **(April 2017)** 5-2
5.2.1 PUSH R_p 5-2
5.2.2 POP R_p 5-3
5.2.3 SPHL 5-4
5.2.4 XTHL **(April 2017)** 5-4
5.2.5 LXI SP, data 5-5
✓ **Syllabus Topic :** Branch Operation **5-5**
5.3 Branch Operation 5-5
5.3.1 JMP Address 5-6
5.3.2 Conditional JUMP Instruction **(April 2017)** 5-6
5.3.3 PCHL 5-7
5.3.4 CALL Address **(April 2017)** 5-8
✓ **Syllabus Topic :** Conditional CALL Instruction **5-9**
5.3.5 Conditional CALL Instruction 5-9
✓ **Syllabus Topic :** Return Instruction **5-10**
5.3.6 RET 5-10
5.3.7 Conditional RET Instruction 5-11
5.4 Machine Control Instructions 5-11
5.4.1 NOP 5-11
5.4.2 HLT 5-12
5.4.3 EI 5-12
5.4.4 DI 5-12
5.4.5 RIM 5-13
5.4.6 SIM 5-14
5.4.7 IN 8 bit Address 5-15

5.4.8 OUT 8 Bit Address **(April 2017)** 5-16
✓ **Syllabus Topic :** Restart **5-16**
5.4.9 RST n 5-16
✓ **Syllabus Topic :** Programming Techniques : Looping, Counting, Indexing, Dynamic Debugging **5-18**
5.5 Looping, Counting and Indexing 5-18
✓ **Syllabus Topic :** Subroutine **5-33**
5.6 Subroutines 5-33
✓ **Syllabus Topic :** Advanced Concepts of Subroutine **5-34**
5.6.1 Parameter Passing Techniques 5-34
5.6.2 Nested Subroutines 5-38
✓ **Syllabus Topic :** Counters and Time Delay (Illustrative Program : Hexadecimal, Modulo-ten Counter) Debugging Counter and Time-Delay Programs **5-41**
5.7 Time Delay and Counters **(April 2017)** 5-41
5.7.1 Hexadecimal, Modulo-ten Counters 5-41
5.7.2 Timing Delay using Nested Loops 5-42
✓ **Syllabus Topic :** Generating Pulse Waveforms **5-47**
5.8 8085 Serial I/O and Generating Pulse Waveforms 5-47
5.8.1 Solved Examples 5-47
✓ **Syllabus Topic :** BCD to Binary Conversion, Binary to BCD Conversion, BCD-to-Seven-Segment-LED Code Conversion, Binary to ASCII and ASCII to Binary Code Conversion, BCD Addition, BCD Subtraction, Introduction To Advanced Instructions and Applications, Multiplication, Subtraction With Carry **5-50**
5.9 Code Conversion : BCD to Binary, Binary to BCD, Binary to ASCII, ASCII to Binary, BCD Addition, BCD Subtraction, Introduction To Advanced Instructions and Applications, Multiplication, Subtraction With Carry, BCD-to-Seven-Segment-LED Code Conversion **(April 2017)** 5-50
✓ **Syllabus Topic :** Testing and Troubleshooting Memory Interfacing 5-57
5.10 Testing and Troubleshooting Memory Interfacing 5-57
✓ **Syllabus Topic:** Microprocessors-Based Software Development system, Operating System and Programming Tools, Assemblers and Cross-Assemblers, Writing Program Using Cross Assemblers. **5-57**
5.11 Microprocessor Based Software Development Tools **(April 2017)** 5-58
5.12 Comprehensive University Questions and Answers 5-59

Chapter 6 : Interrupt System of 8085 **6-1 to 6-12**

6.1 Basic Definitions of Interrupts 6-1
✓ **Syllabus Topic :** The 8085 Interrupt **6-1**
6.2 8085 Interrupt Structure **(April 2017)** 6-1

6.2.1 INTR 6-5
6.2.2 Interrupt Acknowledge Machine Cycle 6-5
✓ **Syllabus Topic :** 8085 Vectored Interrupts **6-6**
6.2.3 Interrupt Vector Locations **(April 2017)** 6-6
6.2.4 Interrupt Priority Structure and Pending Interrupts 6-7
✓ **Syllabus Topic :** RST vs Software Interrupt **6-7**
6.3 Software Interrupts 6-7
6.4 Masking / Unmasking of Interrupts 6-8
✓ **Syllabus Topic :** Additional I/O Concepts and Processes, Interfacing Input Devices **6-11**
6.5 Interfacing Input Devices 6-11
6.6 Comprehensive University Questions and Answers 6-12

Chapter 7 : 8155 PPI **7-1 to 7-23**

7.1 8155 Programmable Peripheral Interface (PPI) 7-1
7.2 8155 Functional Block Diagram 7-1
7.3 Control Register Format 7-5
7.4 Timer Section 7-6
7.4.1 Solved Examples 7-8
✓ **Syllabus Topic :** Basic Interfacing concepts, Interfacing Output Displays, Interfacing Input Devices, Memory Mapped I/O, Testing and Troubleshooting I/O Interfacing Circuits, Interfacing 8155 Memory Segment **7-10**
7.5 8155 Port Modes 7-10
7.5.1 Solved Examples 7-14
7.6 Status Word 7-16
7.7 Interfacing 8155 Memory Segment with 8085 in I/O Mapped I/O 7-22

Chapter 8 : 8085 based Single-board Microcomputer **8-1 to 8-12**

✓ **Syllabus Topic :** 8085 Based Single-Board Microcomputer **8-1**
8.1 Address Decoding 8-1
✓ **Syllabus Topic :** Memory Interfacing, Illustrative Example : Designing Memory for the MCTS Project **8-1**
8.2 Memory Interfacing and Designing Memory for Microprocessor Controlled Temperature System (MCTS) Project 8-1
✓ **Syllabus Topic :** Testing and Troubleshooting Memory Interfacing Circuit **8-12**
8.3 Testing and Troubleshooting Memory Interfacing Circuit 8-12

Chapter 9 : Intel Pentium and Advanced Processors **9-1 to 9-59**

✓ **Syllabus Topic :** Introduction, Special Pentium Registers **9-1**
9.1 Registers of Pentium 9-1
9.1.1 General Purpose Registers 9-2

9.1.2 Pointer Registers of Pentium Processor 9-3
9.1.3 Instruction Pointer 9-4
9.1.4 Flags Register 9-4
9.1.5 Segment Registers 9-6
✓ **Syllabus Topic :** Memory Management **9-6**
9.1.6 Memory Management Registers 9-6
9.1.7 Control Registers 9-6
9.1.8 Debug Registers 9-9
9.2 Processing Modes of 80386 9-11
9.3 Real and Virtual Mode 9-12
9.3.1 Addressing in Real Mode 9-14
9.4 Protection Mode 9-15
9.4.1 Memory Management in Protected Mode of Pentium 9-15
9.4.2 Segmentation in Protection Mode 9-15
9.4.2.1 Accessing Global Memory Location 9-17
9.4.2.2 Accessing Local Memory Location 9-19
9.4.3 Implementation of Paging in Protected Mode 9-20
9.5 Specialities of Pentium Processors 9-22
9.6 Pentium Processor Block Diagram **(April 2017)** 9-23
✓ **Syllabus Topic :** Pentium Instructions **9-26**
9.7 New Instructions in Pentium **(April 2017)** 9-26
9.7.1 Model Specific Register Instructions 9-27
9.7.2 CPUID **(April 2017)** 9-29
✓ **Syllabus Topic :** Pentium Pro Microprocessor **9-32**
9.8 Pentium Pro Processor 9-32
9.8.1 Internal Structure of the Pentium Pro Processor 9-33
9.8.2 Pentium Pro Internal Memory System 9-34
✓ **Syllabus Topic :** Special Pentium Pro Features **9-35**
9.8.3 Special Pentium Pro Features 9-35
✓ **Syllabus Topic :** Introduction to Pentium II and its Software Changes **9-36**
9.9 Pentium 2 Processor 9-36
9.9.1 Pentium II Software Changes 9-37
✓ **Syllabus Topic :** Pentium IV **9-38**
9.10 Pentium-4 NetBurst Architecture 9-38
9.10.1 Pentium 4 Architecture 9-41
✓ **Syllabus Topic :** Core 2, i3, i5 and i7 ..... **9-43**
9.11 Comparative Study of Multi-core Processors i3, i5 and i7 **(April 2017)** 9-43

9.12 Reduced Instruction Set Computer Principles 9-44
9.12.1 RISC Versus CISC 9-44
✓ **Syllabus Topic :** Sun SPARC Register File **9-45**
9.12.2 Register Window 9-45
9.12.3 ON-Chip Register File versus Cache Evaluation 9-46
9.13 Introduction to Sun SPARC Family 9-46
9.14 SPARC Registers 9-46
9.14.1 General Purpose Register 9-47
9.14.2 IU Control/Status Registers 9-49
9.14.3 FPU Control/Status Registers 9-51
✓ **Syllabus Topic :** SUN SPARC Microprocessor : Data Types **9-54**
9.15 Data Formats **(April 2017)** 9-54
9.15.1 Integer Data Formats 9-54
9.15.2 Floating Point 9-55
✓ **Syllabus Topic :** SUN SPARC Microprocessor : Instruction Format **9-56**
9.16 SPARC Instruction Formats 9-56
✓ **Syllabus Topic :** Sun SPARC Architecture **9-57**
9.17 Architecture of Sun Super SPARC **(April 2017)** 9-57
9.18 Comprehensive University Questions and Answers 9-59
- **Model Question Papers ..... M-1 to M-04**
- **List of Practicals ..... P-1 to P-24**
- **Solved University Question Papers of Nov. 17 and April 18 ..... A-1 to A-12**
- **Solved University Question Papers of Nov. 18 and April 19 ..... B-1 to B-8**

❑❑❑

LIST OF PROGRAMS

Program No.	Name of the Program	Page Nos.
Ex. 4.3.1	Write a program to load data 05H into the accumulator and data 20H into the B register.	4-8
Ex. 4.3.2	Write a program to exchange the contents of memory location D000H and D001H.	4-8
Ex. 4.6.1	Add two 8-bit numbers.	4-24
Ex. 4.6.2	Subtract two 8 bit numbers.	4-24
Ex. 4.6.3	Add two 16-bit numbers.	4-25
Ex. 4.6.4	Subtract two 16 bit Numbers.	4-27
Ex. 4.8.1	Find the 1's complement of a number.	4-42
Ex. 4.8.2	Find 2's complement of a number.	4-43
Ex. 4.8.3	Program to mask lower nibble.	4-44
Ex. 4.8.4	Program to mask upper nibble.	4-44
Ex. 4.8.5	Pack the Two Unpacked BCD Numbers.	4-45
Ex. 4.8.6	Unpack the packed BCD number.	4-46
Ex. 4.8.7	Program to shift data within 8 bit register.	4-47
Ex. 4.8.8	Right shift data within 16 bit register.	4-48
Ex. 4.8.9	Left shift 16-bit data within 16-bit register.	4-48
Ex. 4.8.10	Write the assembly language program for 8085 to unpack the packed BCD number stored at memory location E002H. Store result at B004H and B005H. Also draw the flowchart for the same.	4-49
Ex. 4.8.11	Write ALP to convert five packed BCD bytes to unpacked BCD bytes for 8085 based microcomputer.	4-50
Ex. 5.5.1	Program to calculate the sum of series of 8 bit numbers assuming sum to be 8 bit.	5-18
Ex. 5.5.2	Program to calculate the sum of series of 8 bit numbers assuming sum to be 16 bit.	5-19
Ex. 5.5.3	Write an assembly language program to add ten, 8 bit numbers stored from the memory locations F000 H. The result of the addition is 16 bit and it is to be stored in register pair DE.	5-21
Ex. 5.5.4	Program to sort the numbers in ascending order.	5-21
Ex. 5.5.5	Program to sort the numbers in descending order.	5-23
Ex. 5.5.6	Multiply two 8 bit numbers using successive addition method.	5-24
Ex. 5.5.7	Transfer a block of N bytes from source to destination. (Non overlapped block transfer)	5-25
Ex. 5.5.8	Divide 16 bit number by an 8 bit number.	5-26
Ex. 5.5.9	Program to find the number of negative numbers in the array.	5-27

Program No.	Name of the Program	Page Nos.
Ex. 5.5.10	Program to find maximum number in the array.	5-28
Ex. 5.5.11	To reverse the contents of block of N bytes and transfer them from source to destination.	5-29
Ex. 5.6.1	Write a program to compute a ! + b !, a and b are available, store final result at location C101.	5-37
Ex. 5.7.1	Write a subroutine for 8085 to generate delay of 100 (assume 320 ns clock cycle).	5-40
Ex. 5.7.2	Write a subroutine for 8085 to generate a delay of 10 ms (assume 0.333 μs clock cycle).	5-40
Ex. 5.7.3	Write a program to generate a delay of 0.5 sec if the crystal frequency is 5 MHz.	5-41
Ex. 5.7.4	Write a program for displaying binary up counter. Counter should count numbers from 00 to FF H and it should increment after every 0.5 sec. Assume operating frequency of 8085 equal to 5 MHz. Display routine is available.	5-41
Ex. 5.7.5	Design a UP - DOWN counter to count continuously from 00 to 99 and back from 99 to 00, with a delay of 1 seconds.	5-42
Ex. 5.8.1	Prepare a program to generate a square wave of 1 kHz using SOD pin of 8085.	5-43
Ex. 5.8.2	Write a program to transmit B register contents to the peripheral via SOD line. The asynchronous format is given as follows (1) Number of start bits = 1 (2) Number of stop bits = 1 (3) Even parity bit after data (4) CP U clock frequency = 3 MHz. Assume Baud rate.	5-44
Ex. 5.9.1	Write a program to convert BCD to Binary.	46
Ex. 5.9.2	Write a program to convert Binary to BCD.	5-46
Ex. 5.9.3	Write a program in 8085 to convert a 8 bit binary number stored at 1000 H into equivalent gray number. Store the gray number at same location.	5-47
Ex. 5.9.4	Write a program to convert ASCII to Binary.	5-48
Ex. 5.9.5	Write a program to convert Binary to ASCII.	5-48
Ex. 5.9.6	Two decimal numbers six digits each, are stored in BCD package form. Each number occupies a sequence of byte in the memory. The starting address of first number is 6000H. Write an assembly language program that adds these two numbers and stores the sum in the same format starting from memory location 6200H.	5-49
Ex. 5.9.7	Add two 4 digit BCD numbers in HL and DE register pairs and store result in memory locations, 2300H and 2301H. Ignore carry after 16 bit.	5-50
Ex. 5.9.8	Subtract the BCD number stored in E register from the number stored in the D register.	5-50
Ex. 5.9.9	Write an assembly language program to multiply 2 BCD numbers.	5-51

Program No.	Name of the Program	Page Nos.
Ex. 5.9.10	Write an 8085 assembly program to convert BCD to seven segment LED code conversion	5-51
Ex. 5.10.1	Write a program to test RAM by writing '1' and reading it back and later writing '0' (zero) and reading it back. RAM addresses to be checked are 4000H to 40FFH. In case of any error, it is indicated by writing 01H at port 10.	5-52
Ex. 6.4.1	Write a program to mask RST 7.5, RST 6.5 and Unmask RST 5.5 and to output serial data logic '1' on SOD pin, using SIM instruction.	6-10
Ex. 6.5.1	Interface 8 keys using interrupt driven I/O technique.	6-11
Ex. 7.4.1	Write a program to initialise 8155 in configuration given below (i) Port A-output with handshake (ii) Port B- simple input (iii) Start timer (iv) Interrupt enable port A	7-8
Ex. 7.4.2	Generate a continuous square wave of 1 kHz using 8155.	7-9
Ex. 7.5.1	An 8155 is used with port A as output, Port B as input and Port C as input in simple I/O mode. The 8155 is enabled when A_{15} to A_8 = 00110XXX. Write the instruction sequence to program the 8155 for this mode of operation.	7-14
Ex. 7.5.2	An 8085 is using 8155 as a interfacing device with (i) Port A as input device, Port B as input device and Port C as handshake signals. (ii) Port A as input device, Port B as output device and Port C as handshake signals. (iii) Port A as output device, Port B as output device and Port C as handshake signals. (iv) Port A as output device, Port B as input in simple I/O and Port C, 3 bits as handshake signals and remaining 3 bits as output bits. The 8155 CWR address is 30 H. Write the instruction sequence to program 8155 for each of the cases given above.	7-15
Ex. 7.6.1	Interface 8155 with 8085 in simple I/O mode. Generate triangular waveform using DAC0808. The output voltage should swing between 0 – 5V.Use memory mapped I/O.	7-17
Ex. 7.6.2	Write a program to generate a square wave of 2 KHz using 8155 timer. Timer operate at 3 MHz.	7-19
Ex. 7.6.3	With a neat labelled diagram show how six – 7 segment displays can be interfaced to 8085 through 8155. Write a program to display " HELLO" on that display.	7-19
Ex. 8.2.1	Interface 4 KB of EPROM with starting address from 0000H and 2 KB of RAM with starting address followed by EPROM.	8-1
Ex. 8.2.2	Interface 8K of EPROM and 8 KB of RAM using 4KB devices.	8-3
Ex. 8.2.3	Interface 8 KB EPROM and 4 KB RAM to 8085 using absolute decoding technique.	8-5

Program No.	Name of the Program	Page Nos.
Ex. 8.2.4	Explain how to interface an (i) 8K × 8 EPROM using 2K × 8 chips (ii) 16K × 8 RAM using 2K × 8 chips with 8085 microprocessor. Draw memory map and interface diagram.	8-7
Ex. 8.2.5	(i) Interface 16 KB RAM memory chip to 8085 using absolute decoding technique. (ii) Modify the address decoding circuit in the above design to incorporate partial decoding.	8-10
Ex. 8.3.1	Write a program to test RAM by writing '1' and reading it back and later writing '0' (zero) and reading it back. RAM addresses to be checked are 4000H to 40FFH. In case of any error, it is indicated by writing 01H at port 10.	8-12

List of Practicals

Program No.	Name of the Program	Page Nos.
Perform the following Operations related to memory locations.		P-1
Program 1(a)	Store the data byte 32H into memory location 4000H.	P-1
Program 1(b)	Exchange the contents of memory locations 2000H and 4000H.	P-1
Simple assembly language programs.		P-1
Program 2(a)	Subtract the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H.	P-1
Program 2(b)	Subtract two 8-bit numbers.	P-2
Program 2(c)	Add the 16-bit number in memory locations 4000H and 4001H to the 16-bit number in memory locations 4002H and 4003H. The most significant eight bits of the two numbers to be added are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.	P-2
Program 2(d)	Add the contents of memory locations 4000H and 4001H and place the result in the memory locations 4002H and 4003H.	P-2
Program 2(e)	Subtract the 16-bit number in memory locations 4002H and 4003H from the 16-bit number in memory locations 4000H and 4001H. The most significant eight bits of the two numbers are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.	P-3
Program 2(f)	Find the 1's complement of the number stored at memory location 4400H and store the complemented number at memory location 4300H.	P-3
Program 2(g)	Find the 2's complement of the number stored at memory location 4200H and store the complemented number at memory location 4300H.	P-3

Program No.	Name of the Program	Page Nos.
Packing and unpacking operations.		P-4
Program 3(a)	Pack the two unpacked BCD numbers stored in memory locations 4200H and 4201H and store result in memory location 4300H. Assume the least significant digit is stored at 4200H.	P-4
Program 3(b)	Two digit BCD number is stored in memory location 4200H. Unpack the BCD number and store the two digits in memory locations 4300H and 4301H such that memory location 4300H will have lower BCD digit.	P-4
Register Operations.		P-5
Program 4(a)	Write a program to shift an eight bit data four bits right. Assume that data is in register C.	P-5
Program 4(b)	Program to shift a 16-bit data 1 bit left. Assume data is in the HL register pair.	P-5
Program 4(c)	Write a set of instructions to alter the contents of flag register in 8085.	P-5
Program 4(d)	Write a program to count number of 1's in the contents of D register and store the count in the B register.	P-5
Multiple memory locations.		P-6
Program 5(a)	Calculate the sum of series of numbers. The length of the series is in memory location 4200H and the series begins from memory location 4201H. i. Consider the sum to be 8 bit number. So, ignore carries. Store the sum at memory location 4300H. ii. Consider the sum to be 16 bit number. Store the sum at memory locations 4300H and 4301H.	P-6
Program 5(b)	Multiply two 8-bit numbers stored in memory locations 2200H and 2201H by repetitive addition and store the result in memory locations 2300H and 2301H.	P-7
Program 5(c)	Divide 16 bit number stored in memory locations 2200H and 2201H by the 8 bit number stored at memory location 2202H. Store the quotient in memory locations 2300H and 2301H and remainder in memory locations 2302H and 2303H.	P-7
Program 5(d)	Find the number of negative elements (most significant bit 1) in a block of data. The length of the block is in memory location 2200H and the block itself begins in memory location 2201H. Store the number of negative elements in memory location 2300H.	P-8
Program 5(e)	Find the largest number in a block of data. The length of the block is in memory location 2200H and the block itself starts from memory location 2201H. Store the maximum number in memory location 2300H. Assume that the numbers in the block are all 8 bit unsigned binary numbers.	P-8
Calculations with respect to memory locations.		P-9
Program 6(a)	Write a program to sort given 10 numbers from memory location 2200H in the ascending order.	P-9

Program No.	Name of the Program	Page Nos.
Program 6(b)	Calculate the sum of series of even numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 8 bit number so you can ignore carries and store the sum at memory location.	P-9
Program 6(c)	Calculate the sum of series of odd numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 16-bit. Store the sum at memory locations 2300H and 2301H.	P-10
Program 6(d)	Find the square of the given numbers from memory location 6100H and store the result from memory location 7000H.	P-11
Program 6(e)	Search the given byte in the list of 50 numbers stored in the consecutive memory locations and store the address of memory location in the memory locations 2200H and 2201H. Assume byte is in the C register and starting address of the list is 2000H. If byte is not found store 00 at 2200H and 2201H.	P-11
Program 6(f)	Two decimal numbers six digits each, are stored in BCD package form. Each number occupies a sequence of byte in the memory. The starting address of first number is 6000H Write an assembly language program that adds these two numbers and stores the sum in the same format starting from memory location 6200H.	P-12
Program 6(g)	Add 2 arrays having ten 8-bit numbers each and generate a third array of result. It is necessary to add the first element of array 1 with the first element of array-2 and so on. The starting addresses of array1, array2 and array3 are 2200H, 2300H and 2400H, respectively.	P-13
Assembly Programs on Memory Locations		P-13
Program 7(a)	Write an assembly language program to separate even numbers from the given list of 50 numbers and store them in the another list starting from 2300H. Assume starting address of 50 number list is 2200H.	P-13
Program 7(b)	Write assembly language program with proper comments for the following A block of data consisting of 256 bytes is stored in memory starting at 3000H. This block is to be shifted (relocated) in memory from 3050H onwards. Do not shift the block or part of the block anywhere else in the memory.	P-14
Program 7(c)	Add even parity to a string of 7-bit ASCII characters. The length of the string is in memory location 2040H and the string itself begins in memory location 2041H. Place even parity in the most significant bit of each character.	P-14
Program 7(d)	A list of 50 numbers is stored in memory, starting at 6000H. Find number of negative, zero and positive numbers from this list and store these results in memory locations 7000H, 7001H, and 7002H respectively.	P-15

Program No.	Name of the Program	Page Nos.
Program 7(e)	Write an assembly language program to generate Fibonacci number.	P-15
Program 7(f)	Program to calculate the factorial of a number between 0 to 8.	P-16
String Operations in Assembly Programs		P-16
Program 8(a)	Write an 8085 assembly language program to insert a string of four characters from the tenth location in the given array of 50 characters.	P-16
Program 8(b)	Write an 8085 assembly language program to delete a string of 4 characters from the tenth location in the given array of 50 characters.	P-17
Program 8(c)	Multiply the 8-bit unsigned number in memory location 2200H by the 8-bit unsigned number in memory location 2201H. Store the 8 least significant bits of the result in memory location 2300H and the 8 most significant bits in memory location 2301H.	P-17
Program 8(d)	Divide the 16-bit unsigned number in memory locations 2200H and 2201H (most significant bits in 2201H) by the B-bit unsigned number in memory location 2300H store the quotient in memory location 2400H and remainder in 2401H.	P-18
Program 8(e)	DAA instruction is not present. Write a sub routine which will perform the same task as DAA.	P-19
Calculations on Memory Locations		P-20
Program 9(a)	To test RAM by writing '1' and reading it back and later writing '0' (zero) and reading it back. RAM addresses to be checked are 4000H to 40FFH. In case of any error, it is indicated by writing 01H at port 10.	P-20
Program 9(b)	Arrange an array of 8 bit unsigned no in descending order.	P-20
Program 9(c)	Transfer ten bytes of data from one memory to another memory block. Source memory block starts from memory location 2200H where as destination memory block starts from memory location 2300H.	P-21
Program 9(d)	Write a program to find the Square Root of an 8 bit binary number. The binary number is stored in memory location 4200H and store the square root in 4201H.	P-21
Program 9(e)	Write a simple program to Split a HEX data into two nibbles and store it in memory.	P-22
Operations on BCD Numbers		P-22
Program 10(a)	Add two 4 digit BCD numbers in HL and DE register pairs and store result in memory locations, 2300H and 2301H. Ignore carry after 16 bit.	P-22
Program 10(b)	Subtract the BCD number stored in E register from the number stored in the D register.	P-23
Program 10(c)	Write an assembly language program to multiply 2 BCD numbers.	P-23

❑❑❑