VSIT | Vidyalankar School of Information Technology
NAAC Accredited College

LAB MANUAL

SUBJECT : Microprocessor Architecture

YEAR : First Year BSc IT SEMESTER : 2

ACADEMIC YEAR : 2017-18

PRACTICAL 1

TITLE:

Simple Data Transfer programs

PRIOR CONCEPTS:

1. Data Transfer Instructions
2. Arithmetic Instructions
3. Logical Instructions
4. Branching Group Instructions
5. Machine Control Group Instructions

NEW CONCEPTS:

1. Programing using Simple data transfer instructions

THEORY:

This practical consists of programs which involve simple data transfer instructions

a. Store the data byte 32H into memory location 4000H.

Address	Label	Mnemonics	Hexcode	Bytes
0000		MVI A,32	3E	2
0001			32	
0002		STA 4000	32	3
0003			00	
0004			40	
0005		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

b. Exchange the contents of memory locations 2000H and 4000H.

Program:

Address	Label	Mnemonics	Hexcode	Bytes
0000		LDA 2000	3A	3
0001			00	
0002			20	
0003		MOV B,A	47	1
0004		LDA 4000	3A	3
0005			00	
0006			40	
0007		STA 2000	32	3
0008			00	
0009			20	
000A		MOV A,B	78	1
000B		STA 4000	32	3
000C			00	
000D			40	
000E		HLT	76	1

OBSERVATION:

RESULT:
Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

PRACTICAL 2

TITLE:

Simple Arithmetic and Logical Programs

PRIOR CONCEPTS:

1. Data Transfer Instructions
2. Arithmetic Instructions
3. Logical Instructions
4. Branching Group Instructions
5. Machine Control Group Instructions

NEW CONCEPTS:

1. Programing using Simple data transfer instructions

THEORY:

This practical consists of programs which involve arithmetic programs of 8-bit addition, 8-bit subtraction, 16-bit addition and 16-bit subtraction as well as simple logical programs of 1's compliment and 2's complement.

a. Subtract the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H.

Program:

Address	Label	Mnemonics	Hexcode	Bytes
0000		LXI H,4001	21	3
0001			01	
0002			40	
0003		LDA 2000	3A	3
0004			00	
0005			20	
0006		SUB M	96	1
0007		INX H	23	1
0008		MOV M,A	77	1
0009		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

b. Subtract two 8-bit numbers. 33H – 22H

Address	Label	Mnemonics	Hexcode	Bytes
0000		MVI A,33	3E	2
0001			33	
0002		SUI 22	D6	2
0003			22	
0004		HLT	76	1

Program:

OBSERVATION:

RESULT:
Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

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.

Program:

Address	Label	Mnemonics	Hexcode	Bytes
0000		LHLD 4000	2A	3
0001			00	
0002			40	
0003		XCHG	EB	1
0004		LHLD 4002	2A	3
0005			02	
0006			40	
0007		DAD D	19	1
0008		SHLD 4004	22	3
0009			04	
000A			40	
000B		HLT	76	1

OBSERVATION:

RESULT:
Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

d. Add the contents of memory locations 4000H and 4001H and place the result in the memory locations 4002Hand 4003H.

Program:

Address	Label	Mnemonics	Hexcode	Bytes
0000		MVI D,00	16	2
0001			00	
0002		LXI H,4000	21	3
0003			00	
0004			40	
0005		MOV A,M	7E	1
0006		INX H	23	1
0007		ADD M	86	1
0008		JNC SKIP	D2	3
0009			0C	
000A			00	
000B		INR D	14	1
000C	SKIP	INX H	23	1
000D		MOV M,A	77	1
000E		INX H	23	1
000F		MOV M,D	72	1
0010		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

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.

Program:

Address	Label	Mnemonics	Hexcode	Bytes
0000		LDA 4000	3A	3
0001			00	
0002			40	
0003		LXI H,4002	21	3
0004			02	
0005			40	
0006		SUB M	96	1
0007		STA 4004	32	3
0008			04	
0009			40	
000A		LDA 4001	3A	3
000B			01	
000C			40	
000D		INX H	23	1
000E		SBB M	9E	1
000F		STA 4005	32	3
0010			05	
0011			40	
0012		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

f. Find the 1's complement of the number stored at memory location 4400H and store the complemented number at memory location 4300H.

Program:

Address	Label	Mnemonics	Hexcode
0000		LDA 4400	3A
0001			00
0002			44
0003		CMA	2F
0004		STA 4300	32
0005			00
0006			43
0007		HLT	76

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

g. Find the 2's complement of the number stored at memory location 4200H and store the complemented number at memory location 4300H.

Program:

Address	Label	Mnemonics	Hexcode	Bytes
0000		LDA 4400	3A	3
0001			00	
0002			44	
0003		CMA	2F	1
0004		STA 4300	32	3
0005			00	
0006			43	
0007		INR A	3C	1
0008		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

PRACTICAL 3

TITLE:

Packing and unpacking operations.

PRIOR CONCEPTS:

1. Data Transfer Instructions
2. Arithmetic Instructions
3. Logical Instructions
4. Branching Group Instructions
5. Machine Control Group Instructions

NEW CONCEPTS:

1. Packing of BCD numbers
2. Unpacking of BCD numbers

THEORY:

This practical consists of programs which involve packing of unpacked BCD numbers as well as unpacking of Packed BCD

- Example for packing unpacked BCD:

If we have two unpacked BCD numbers: 04H in memory location 4200H and 09H in memory location 4201H, then after packing the result should be either 94H if LSB was in 4200H or 49H if LSB was in 4201H.

- Example for unpacking packed BCD:

If we have unpacked BCD number 38H, then after unpacking the result should be two separated BCD numbers 03H and 08H.

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.**

Address	Label	Mnemonics	Hexcode	Bytes
0000		LDA 4201	3A	3
0001			01	
0002			42	
0003		RLC	07	1
0004		RLC	07	1
0005		RLC	07	1
0006		RLC	07	1
0007		MOV B,A	47	1
0008		LDA 4200	3A	3
0009			00	
000A			42	
000B		ADD B	80	1
000C		STA 4300	32	3
000D			00	
000E			43	
000F		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

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.

Address	Label	Mnemonics	Hexcode	Bytes
0000		LXI H,4300	21	3
0001			00	
0002			43	
0003		LDA 4200	3A	3
0004			00	
0005			42	
0006		ANI 0F	E6	2
0007			0F	
0008		MOV M,A	77	1
0009		LDA 4200	3A	3
000A			00	
000B			42	
000C		ANI F0	E6	2
000D			F0	
000E		RLC	07	1
000F		RLC	07	1
0010		RLC	07	1
0011		RLC	07	1
0012		INX H	23	1
0013		MOV M,A	77	1
0014		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

PRACTICAL 4

TITLE:

Register Operations.

PRIOR CONCEPTS:

1. Data Transfer Instructions
2. Arithmetic Instructions
3. Logical Instructions
4. Branching Group Instructions
5. Machine Control Group Instructions

NEW CONCEPTS:

1. Rotating 8-bit and 16-bit data
2. Operations of flag register and stack

THEORY:

This practical consists of programs which involve rotating of 8-bit and 16-bit data, Operation on flag register and stack operations.

When any 8-bit data is shifted to left / right by 4 bits its nibbles get exchanged.

When any data is doubled (added to itself) it gets shifted to left by 1 bit

a. **Write a program to shift an eight bit data four bits right. Assume that data is in register C.**

Address	Label	Mnemonics	Hexcode	Bytes
0000		MOV A,C	79	1
0001		RRC	0F	1
0002		RRC	0F	1
0003		RRC	0F	1
0004		RRC	0F	1
0005		MOV C,A	4F	1
0006		HLT	76	1

OBSERVATION:

RESULT:
Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

b. Write a program to shift a 16-bit data 1 bit left. Assume data is in the HL register pair.

Address	Label	Mnemonics	Hexcode	Bytes
0000		DAD H	29	1
0001		JNC SKIP	D2	3
0002			05	
0003			00	
0004		INR L	2C	1
0005	SKIP	HLT	76	1

OBSERVATION:

RESULT:
Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

c. Write a set of instructions to alter the contents of flag register in 8085.

Address	Label	Mnemonics	Hexcode	Bytes
0000		LXI SP,7060	31	3
0001			60	
0002			70	
0003		PUSH PSW	F5	1
0004		POP B	C1	1
0005		MOV A,C	79	1
0006		CMA	2F	1
0007		MOV C,A	4F	1
0008		PUSH B	C5	1
0009		POP PSW	F1	1
000A		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

d. Write a program to count number of l's in the contents of D register and store the count in the B register.

Address	Label	Mnemonics	Hexcode	Bytes
0000		MVI C,08	0E	2
0001			08	
0002		MVI B,00	06	2
0003			00	
0004		MOV A,D	7A	1
0005	LOOP	RRC	0F	1
0006		JNC SKIP	D2	3
0007			0A	
0008			00	
0009		INR B	04	1
000A	SKIP	DCR C	0D	1
000B		JNZ LOOP	C2	3
000C			05	
000D			00	
000E		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

PRACTICAL 5

TITLE:

Multiple Memory locations programs.

PRIOR CONCEPTS:

1. Data Transfer Instructions
2. Arithmetic Instructions
3. Logical Instructions
4. Branching Group Instructions
5. Machine Control Group Instructions

NEW CONCEPTS:

1. Programing with multiple memory locations

THEORY:

This practical consists of programs which deal with multiple memory locations. In other words these programs deal with a block or series of data bytes.
This practical includes programs like –

- Addition of all the elements in a series of data bytes ignoring carry
- Addition of all the elements in a series of data bytes considering carry
- Multiplication by successive addition
- Division of 16-bit dividend by 8-bit divisor
- Finding the number of negative elements from the given series of data bytes
- Finding the largest number from the series of given numbers

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.**

Program i:

Address	Label	Mnemonics	Hexcode	Bytes
0000		XRA A	AF	1
0001		LXI H,4200	21	3
0002			00	
0003			42	
0004		MOV C,M	4E	1
0005	LOOP	INX H	23	1
0006		ADD M	86	1
0007		DCR C	0D	1
0008		JNZ LOOP	C2	3
0009			05	
000A			00	
000B		STA 4300	32	3
000C			00	
000D			43	
000E		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

Program ii.

Address	Label	Mnemonics	Hexcode	Bytes
0000		XRA A	AF	1
0001		MOV D,A	57	1
0002		LXI H,4200	21	3
0003			00	
0004			42	
0005		MOV C,M	4E	1
0006	LOOP	INX H	23	1
0007		ADD M	86	1
0008		JNC SKIP	D2	3
0009			0C	
000A			00	
000B		INR D	14	1
000C	SKIP	DCR C	0D	1
000D		JNZ LOOP	C2	3
000E			06	
000F			00	
0010		LXI H,4300	21	3
0011			00	
0012			43	
0013		MOV M,A	77	1
0014		INX H	23	1
0015		MOV M,D	72	1
0016		HLT	76	1

OBSERVATION:

RESULT:
Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

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.

Address	Label	Mnemonics	Hexcode	Bytes
0000		XRA A	AF	1
0001		MOV D,A	57	1
0002		LXI H,2200	21	3
0003			00	
0004			22	
0005		MOV C,M	4E	1
0006		INX H	23	1
0007	LOOP	ADD M	86	1
0008		JNC SKIP	D2	3
0009			0C	
000A			00	
000B		INR D	14	1
000C	SKIP	DCR C	0D	1
000D		JNZ LOOP	C2	3
000E			07	
000F			00	
0010		LXI H,2300	21	3
0011			00	
0012			23	
0013		MOV M,A	77	1
0014		INX H	23	1
0015		MOV M,D	72	1
0016		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

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.

Address	Label	Mnemonics	Hexcode	Bytes
0000		LXI D,0000	11	3
0001			00	
0002			00	
0003		LHLD 2200	2A	3
0004			00	
0005			22	
0006		LDA 2202	3A	3
0007			02	
0008			22	
0009		MOV B,A	47	1
000A		MOV A,L	7D	1
000B	LOOP	CMP B	B8	1
000C		JNC DOWN	D2	3
000D			18	
000E			00	
000F		MOV L,A	6F	1
0010		MOV A,H	7C	1
0011		CPI 00	FE	2
0012			00	
0013		JZ ESC	CA	3
0014			1D	
0015			00	
0016		DCR H	25	1
0017		MOV A,L	7D	1
0018	DOWN	INX D	13	1
0019		SUB B	90	1
001A		JMP LOOP	C3	3
001B			0B	
001C			00	
001D	ESC	XCHG	EB	1
001E		SHLD 2300	22	3
001F			00	
0020			23	
0021		MOV A,E	7B	1
0022		STA 2302	32	3
0023			02	
0024			23	
0025		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

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.

Address	Label	Mnemonics	Hexcode	Bytes
0000		MVI D,00	16	2
0001			00	
0002		LXI H,2200	21	3
0003			00	
0004			22	
0005		MOV C,M	4E	1
0006	LOOP	INX H	23	1
0007		MOV A,M	7E	1
0008		RLC	07	1
0009		JNC POSTV	D2	3
000A			0D	
000B			00	
000C		INR D	14	1
000D	POSTV	DCR C	0D	1
000E		JNZ LOOP	C2	3
000F			06	
0010			00	
0011		MOV A,D	7A	1
0012		STA 2300	32	3
0013			00	
0014			23	
0015		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

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.

Address	Label	Mnemonics	Hexcode	Bytes
0000		XRA A	AF	1
0001		LXI H,2200	21	3
0002			00	
0003			22	
0004		MOV C,M	4E	1
0005	LOOP	INX H	23	1
0006		CMP M	BE	1
0007		JNC SKIP	D2	3
0008			0B	
0009			00	
000A		MOV A,M	7E	1
000B	SKIP	DCR C	0D	1
000C		JNZ LOOP	C2	3
000D			05	
000E			00	
000F		STA 2300	32	3
0010			00	
0011			23	
0012		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

PRACTICAL 6

TITLE:

Calculations with respect to memory locations.

PRIOR CONCEPTS:

1. Data Transfer Instructions
2. Arithmetic Instructions
3. Logical Instructions
4. Branching Group Instructions
5. Machine Control Group Instructions

NEW CONCEPTS:

1. Sorting given set of numbers in ascending order and descending order
2. Even number additions
3. Odd number additions

THEORY:

This practical consists of programs which deal with calculations with memory locations. This practical includes programs like –

- Sorting the given numbers in ascending order
- Addition of only even numbers of the list
- Addition of only odd numbers of the list
- Finding square of the number in given memory location
- Finding the given data byte in the list of data bytes

a. **Write a program to sort given 10 numbers from memory location 7200H in the ascending order.**

Address	Label	Mnemonics	Hexcode	Bytes
0000	UP	MVI D,00	16	2
0001			00	
0002		MVI C,09	0E	2
0003			09	
0004		LXI H,7200	21	3
0005			00	
0006			72	
0007	LOOP	MOV A,M	7E	1
0008		INX H	23	1
0009		CMP M	BE	1
000A		JC DOWN	DA	3
000B			14	
000C			00	
000D		MOV B,M	46	1
000E		MOV M,A	77	1
000F		DCX H	2B	1
0010		MOV M,B	70	1
0011		INX H	23	1
0012		MVI D,01	16	2
0013			01	
0014	DOWN	DCR C	0D	1
0015		JNZ LOOP	C2	3
0016			07	
0017			00	
0018		MOV A,D	7A	1
0019		RRC	0F	1
001A		JC UP	DA	3
001B			00	
001C			00	
001D		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

b. Calculate the sum of series of even numbers from the list of numbers. The length of the list is in memory location 7200H and the series itself begins from memory location 7201H. Assume the sum to be 8 bit number so you can ignore carries and store the sum at memory location 7300H.

Address	Label	Mnemonics	Hexcode	Bytes
0000		MVI D,00	16	2
0001			00	
0002		LXI H,7200	21	3
0003			00	
0004			72	
0005		MOV C,M	4E	1
0006	LOOP	INX H	23	1
0007		MOV A,M	7E	1
0008		RRC	0F	1
0009		JC ODD	DA	3
000A			0F	
000B			00	
000C		RLC	07	1
000D		ADD D	82	1
000E		MOV D,A	57	1
000F	ODD	DCR C	0D	1
0010		JNZ LOOP	C2	3
0011			06	
0012			00	
0013		LXI H,7300	21	3
0014			00	
0015			73	
0016		MOV M,D	72	1
0017		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

c. Calculate the sum of series of odd numbers from the list of numbers. The length of the list is in memory location 7200H and the series itself begins from memory location 7201H. Assume the sum to be 8 bit number so you can ignore carries and store the sum at memory location 7300H.

Address	Label	Mnemonics	Hexcode	Bytes
0000		MVI D,00	16	2
0001			00	
0002		LXI H,7200	21	3
0003			00	
0004			72	
0005		MOV C,M	4E	1
0006	LOOP	INX H	23	1
0007		MOV A,M	7E	1
0008		RRC	0F	1
0009		JNC EVEN	D2	3
000A			0F	
000B			00	
000C		RLC	07	1
000D		ADD D	82	1
000E		MOV D,A	57	1
000F	EVEN	DCR C	0D	1
0010		JNZ LOOP	C2	3
0011			06	
0012			00	
0013		LXI H,7300	21	3
0014			00	
0015			73	
0016		MOV M,D	72	1
0017		HLT	76	1

OBSERVATION:

RESULT:
Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

d. Find the square of the given number at memory location 7100H and store the result from memory location 7200H.

Address	Label	Mnemonics	Hexcode	Bytes
0000		XRA A	AF	1
0001		MOV D,A	57	1
0002		LXI H,7100	21	3
0003			00	
0004			71	
0005		MOV C,M	4E	1
0006	LOOP	ADD M	86	1
0007		JNC SKIP	D2	3
0008			0B	
0009			00	
000A		INR D	14	1
000B	SKIP	DCR C	0D	1
000C		JNZ LOOP	C2	3
000D			06	
000E			00	
000F		LXI H,7200	21	3
0010			00	
0011			72	
0012		MOV M,A	77	1
0013		INX H	23	1
0014		MOV M,D	72	1
0015		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

e. Search the given byte in the list of 10 numbers stored in the consecutive memory locations and store the address of memory location in the memory locations 7200H and 7201H. Assume byte is in the C register and starting address of the list is 7100H. If byte is not found store 00 at 7200H and 7201H.

Address	Label	Mnemonics	Hexcode	Bytes
0000		MVI D,0A	16	2
0001			0A	
0002		LXI H,7100	21	3
0003			00	
0004			71	
0005	LOOP	MOV A,M	7E	1
0006		CMP C	B9	1
0007		JZ FOUND	CA	3
0008			12	
0009			00	
000A		INX H	23	1
000B		DCR D	15	1
000C		JNZ LOOP	C2	3
000D			05	
000E			00	
000F		LXI H,0000	21	3
0010			00	
0011			00	
0012	FOUND	SHLD 7200	22	3
0013			00	
0014			72	
0015		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

PRACTICAL 7

TITLE:

Calculations on memory locations.

PRIOR CONCEPTS:

1. Data Transfer Instructions
2. Arithmetic Instructions
3. Logical Instructions
4. Branching Group Instructions
5. Machine Control Group Instructions

NEW CONCEPTS:

1. Block transfer
2. Splitting of nibbles

THEORY:

This practical consists of programs which deal with calculations related with memory locations. In other words these programs deal with a block or series of data bytes. This practical includes programs like –

- Sorting a series of given numbers in descending order
- Transfer a data block from one location to another
- Split the nibbles of a given number

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.**

Address	Label	Mnemonics	Hexcode	Bytes
0000		LXI H,2200	21	3
0001			00	
0002			22	
0003		LXI D,2300	11	3
0004			00	
0005			23	
0006		MVI C,32	0E	2
0007			32	
0008	BACK	MOV A,M	7E	1
0009		ANI 01	E6	2
000A			01	
000B		JNZ SKIP	C2	3
000C			11	
000D			00	
000E		MOV A,M	7E	1
000F		STAX D	12	1
0010		INX D	13	1
0011	SKIP	INX H	23	1
0012		DCR C	0D	1
0013		JNZ BACK	C2	3
0014			08	
0015			00	
0016		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

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.

Address	Label	Mnemonics	Hexcode	Bytes
0000		MVI C,FF	0E	2
0001			FF	
0002		LXI H,30FF	21	3
0003			FF	
0004			30	
0005		LXI D,314F	11	3
0006			4F	
0007			31	
0008	BACK	MOV A,M	7E	1
0009		STAX D	12	1
000A		DCX H	2B	1
000B		DCX D	1B	1
000C		DCR C	0D	1
000D		JNZ BACK	C2	3
000E			08	
000F			00	
0010		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

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.**

Address	Label	Mnemonics	Hexcode	Bytes
0000		LXI H,2040	21	3
0001			40	
0002			20	
0003		MOV C,M	4E	1
0004	REPEAT	INX H	23	1
0005		MOV A,M	7E	1
0006		ORA A	B7	1
0007		JPO PAREVEN	E2	3
0008			0C	
0009			00	
000A		ORI 80	F6	2
000B			80	
000C	PAREV...	MOV M,A	77	1
000D		DCR C	0D	1
000E		JNZ REPEAT	C2	3
000F			04	
0010			00	
0011		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

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.

Address	Label	Mnemonics	Hexcode	Bytes
0000		LXI H,6000	21	3
0001			00	
0002			60	
0003		MVI C,00	0E	2
0004			00	
0005		MVI B,00	06	2
0006			00	
0007		MVI E,00	1E	2
0008			00	
0009	BEGIN	MOV A,M	7E	1
000A		CPI 00	FE	2
000B			00	
000C		JZ ZERONUM	CA	3
000D			18	
000E			00	
000F		ANI 80	E6	2
0010			80	
0011		JNZ NEGNUM	C2	3
0012			1C	
0013			00	
0014		INR D	14	1
0015		JMP LAST	C3	3
0016			1D	
0017			00	
0018	ZERON...	INR E	1C	1
0019		JMP LAST	C3	3
001A			1D	
001B			00	
001C	NEGNUM	INR B	04	1
001D	LAST	INX H	23	1
001E		INR C	0C	1
001F		MOV A,C	79	1
0020		CPI 32	FE	2
0021			32	
0022		JNZ BEGIN	C2	3
0023			09	
0024			00	
0025		LXI H,7000	21	3
0026			00	
0027			70	
0028		MOV M,B	70	1
0029		MOV M,E	73	1
002A		INX H	23	1
002B		MOV M,D	72	1

OBSERVATION:

RESULT:
Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

e. Write an assembly language program to generate fibonacci number.

Address	Label	Mnemonics	Hexcode	Bytes
0000		MVI D,0A	16	2
0001			0A	
0002		MVI B,00	06	2
0003			00	
0004		MVI C,01	0E	2
0005			01	
0006		MOV A,B	78	1
0007	BACK	ADD C	81	1
0008		MOV B,C	41	1
0009		MOV C,A	4F	1
000A		DCR D	15	1
000B		JNZ BACK	C2	3
000C			07	
000D			00	
000E		HLT	76	1

OBSERVATION:

RESULT:
Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

f. Program to calculate the factorial of a number between 0 to 8.

Main:

Address	Label	Mnemonics	Hexcode	Bytes
0000		LXI SP,27FF	31	3
0001			FF	
0002			27	
0003		LDA 2200	3A	3
0004			00	
0005			22	
0006		CPI 02	FE	2
0007			02	
0008		JC LAST	DA	3
0009			1A	
000A			00	
000B		MVI D,00	16	2
000C			00	
000D		MOV E,A	5F	1
000E		DCR A	3D	1
000F		MOV C,A	4F	1
0010		CALL 0025	CD	3
0011			25	
0012			00	
0013		XCHG	EB	1
0014		SHLD 2201	22	3
0015			01	
0016			22	
0017		JMP END	C3	3
0018			1D	
0019			00	
001A	LAST	LXI H,0001	21	3
001B			01	
001C			00	
001D	END	SHLD 2201	22	3
001E			01	
001F			22	
0020		HLT	76	1

Subroutine:

0025	FACTO	LXI H,0000	21	3
0026			00	
0027			00	
0028		MOV B,C	41	1
0029	BACK	DAD D	19	1
002A		DCR B	05	1
002B		JNZ BACK	C2	3
002C			29	
002D			00	
002E		XCHG	EB	1
002F		DCR C	0D	1
0030		CNZ FACTO	C4	3
0031			25	
0032			00	
0033		RET	C9	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

PRACTICAL 8

TITLE:

String operations in assembly programs

PRIOR CONCEPTS:

1. Data Transfer Instructions
2. Arithmetic Instructions
3. Logical Instructions
4. Branching Group Instructions
5. Machine Control Group Instructions

NEW CONCEPTS:

1. Programing consisting string operations

THEORY:

This practical consists of string operations

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.**

Address	Label	Mnemonics	Hexcode	Bytes
0000		LXI H,2131	21	3
0001			31	
0002			21	
0003		LXI D,2135	11	3
0004			35	
0005			21	
0006	AGAIN	MOV A,M	7E	1
0007		STAX D	12	1
0008		DCX D	1B	1
0009		DCX H	2B	1
000A		MOV A,L	7D	1
000B		CPI 05	FE	2
000C			05	
000D		JNZ AGAIN	C2	3
000E			06	
000F			00	
0010		INX H	23	1
0011		LXI D,2200	11	3
0012			00	

0013			22	
0014	REPE	LDAX D	1A	1
0015		MOV M,A	77	1
0016		INX D	13	1
0017		INX H	23	1
0018		MOV A,E	7B	1
0019		CPI 04	FE	2
001A			04	
001B		JNZ REPE	C2	3
001C			14	
001D			00	
001E		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

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.

Address	Label	Mnemonics	Hexcode	Bytes
0000		LXI H,021D	21	3
0001			1D	
0002			02	
0003	REPE	LXI D,2109	11	3
0004			09	
0005			21	
0006		MOV A,M	7E	1
0007		STAX D	12	1
0008		INX D	13	1
0009		INX H	23	1
000A		MOV A,L	7D	1
000B		CPI 32	FE	2
000C			32	
000D		JNZ REPE	C2	3
000E			03	
000F			00	
0010		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

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.

Address	Label	Mnemonics	Hexcode	Bytes
0000		LXI H,2200	21	3
0001			00	
0002			22	
0003		MOV E,M	5E	1
0004		MVI D,00	16	2
0005			00	
0006		INX H	23	1
0007		MOV A,M	7E	1
0008		LXI H,0000	21	3
0009			00	
000A			00	
000B		MVI B,08	06	2
000C			08	
000D	MULT	DAD H	29	1
000E		RAL	17	1
000F		JNC SKIP	D2	3
0010			13	
0011			00	
0012		DAD D	19	1
0013	SKIP	DCR B	05	1
0014		JNZ MULT	C2	3
0015			0D	
0016			00	
0017		SHLD 2300	22	3
0018			00	
0019			23	
001A		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	Z	-	AC	-	P	-	CY

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.

Address	Label	Mnemonics	Hexcode	Bytes
0000		MVI E,00	1E	2
0001			00	
0002		LHLD 2200	2A	3
0003			00	
0004			22	
0005		LDA 2300	3A	3
0006			00	
0007			23	
0008		MOV B,A	47	1
0009		MVI C,08	0E	2
000A			08	
000B	NEXT	DAD H	29	1
000C		MOV A,E	7B	1
000D		RLC	07	1
000E		MOV E,A	5F	1
000F		MOV A,H	7C	1
0010		SUB B	90	1
0011		JC SKIP	DA	3
0012			16	
0013			00	
0014		MOV H,A	67	1
0015		INR E	1C	1
0016	SKIP	DCR C	0D	1
0017		JNZ NEXT	C2	3
0018			0B	
0019			00	
001A		MOV A,E	7B	1
001B		STA 2401	32	3
001C			01	
001D			24	
001E		MOV A,H	7C	1
001F		STA 2410	32	3
0020			10	
0021			24	
0022		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

e. DAA instruction is not present. Write a sub routine which will perform the same task as DAA.

Address	Label	Mnemonics	Hexcode	Bytes
0000		LXI SP,27FF	31	3
0001			FF	
0002			27	
0003		MOV E,A	5F	1
0004		ANI 0F	E6	2
0005			0F	
0006		CPI 0A	FE	2
0007			0A	
0008		JC SKIP	DA	3
0009			11	
000A			00	
000B		MOV A,E	7B	1
000C		ADI 06	C6	2
000D			06	
000E		JMP SECOND	C3	3
000F			1C	
0010			00	
0011	SKIP	PUSH PSW	F5	1
0012		POP B	C1	1

0013		MOV A,C	79	1
0014		ANI 10	E6	2
0015			10	
0016		JZ SECOND	CA	3
0017			1C	
0018			00	
0019		MOV A,E	7B	1
001A		ADI 06	C6	2
001B			06	
001C	SECOND	MOV E,A	5F	1
001D		ANI F0	E6	2
001E			F0	
001F		RRC	0F	1
0020		RRC	0F	1
0021		RRC	0F	1
0022		RRC	0F	1
0023		CPI 0A	FE	2
0024			0A	
0025		JC SKIP	DA	3
0026			11	
0027			00	
0028		MOV A,E	7B	1
0029		ADI 60	C6	2
002A			60	
002B		JMP LAST	C3	3
002C			34	
002D			00	
002E	SKIP1	JNC LAST	D2	3
002F			34	
0030			00	
0031		MOV A,E	7B	1
0032		ADI 60	C6	2
0033			60	
0034	LAST	HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

PRACTICAL 9

TITLE:

Calculations on memory locations.

PRIOR CONCEPTS:

1. Data Transfer Instructions
2. Arithmetic Instructions
3. Logical Instructions
4. Branching Group Instructions
5. Machine Control Group Instructions

NEW CONCEPTS:

1. Programing related to RAM

THEORY:

This practical consists of programs operating on memory locations.

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 40FFH to 40FFH. In case of any error, it is indicated by writing 01H at port 10.

Address	Label	Mnemonics	Hexcode	Bytes
0000		LXI H,4000	21	3
0001			00	
0002			40	
0003	BACK	MVI M,FF	36	2
0004			FF	
0005		MOV A,M	7E	1
0006		CPI FF	FE	2
0007			FF	
0008		JNZ ERROR	C2	3
0009			21	
000A			00	
000B		INX H	23	1
000C		MOV A,H	7C	1
000D		CPI 50	FE	2
000E			50	
000F		JNZ BACK	C2	3
0010			03	
0011			00	
0012		LXI H,4000	21	3

0013			00	
0014			40	
0015	BACKL	MVI M,00	36	2
0016			00	
0017		MOV A,M	7E	1
0018		CPI 00	FE	2
0019			00	
001A		INX H	23	1
001B		MOV A,H	7C	1
001C		CPI 50	FE	2
001D			50	
001E		JNZ BACK	C2	3
001F			03	
0020			00	
0021	ERROR	HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

b. Arrange an array of 8 bit unsigned no in descending order.

Address	Label	Mnemonics	Hexcode	Bytes
0000	START	MVI B,00	06	2
0001			00	
0002		LXI H,4150	21	3
0003			50	
0004			41	
0005		MOV C,M	4E	1
0006		DCR C	0D	1
0007		INX H	23	1
0008	LOOP	MOV A,M	7E	1
0009		INX H	23	1
000A		CMP M	BE	1
000B		JNC LOOP	D2	3
000C			08	
000D			00	
000E		MOV D,M	56	1
000F		MOV M,A	77	1
0010		DCR H	25	1
0011		MOV M,D	72	1
0012		INX H	23	1
0013		MVI B,01	06	2
0014			01	
0015	LOOP	DCR C	0D	1
0016		JNZ LOOP	C2	3
0017			08	
0018			00	
0019		DCR B	05	1
001A		JZ START	CA	3
001B			00	
001C			00	
001D		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	Z	-	AC	-	P	-	CY

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.**

Address	Label	Mnemonics	Hexcode	Bytes
0000		LXI H,4150	21	3
0001			50	
0002			41	
0003		MVI B,08	06	2
0004			08	
0005		MVI A,54	3E	2
0006			54	
0007	LOOP	RRC	0F	1
0008		JC LOOP2	DA	3
0009			10	
000A			00	
000B		MVI M,00	36	2
000C			00	
000D		JMP COMMON	C3	3
000E			12	
000F			00	
0010	LOOP2	MVI M,01	36	2
0011			01	
0012	COMM...	INX H	23	1
0013		DCR B	05	1
0014		JNZ LOOP	C2	3
0015			07	
0016			00	
0017		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

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.

Main:

Address	Label	Mnemonics	Hexcode	Bytes
0000		LDA 4200	3A	3
0001			00	
0002			42	
0003		MOV B,A	47	1
0004		MVI C,02	0E	2
0005			02	
0006		CALL 0020	CD	3
0007			20	
0008			00	
0009	REP	MOV E,D	5A	1
000A		MOV A,B	78	1
000B		MOV C,D	4A	1
000C		CALL 0020	CD	3
000D			20	
000E			00	
000F		MOV A,D	7A	1
0010		ADD E	83	1
0011		MVI C,02	0E	2
0012			02	
0013		CALL 0020	CD	3
0014			20	
0015			00	
0016		MOV A,E	7B	1
0017		CMP D	BA	1
0018		JNZ REP	C2	3
0019			09	
001A			00	
001B		STA 4201	32	3
001C			01	
001D			42	
001E		HLT	76	1

Subroutine:

0020		MVI D,00	16	2
0021			00	
0022	NEXT	SUB C	91	1
0023		INR D	14	1
0024		CMP C	B9	1
0025		JNC NEXT	D2	3
0026			22	
0027			00	
0028		RET	C9	1

OBSERVATION:

RESULT:
Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

e. Write a simple program to Split a HEX data into two nibbles and store it in memory program.

Address	Label	Mnemonics	Hexcode	Bytes
0000		LXI H,4200	21	3
0001			00	
0002			42	
0003		MOV B,M	46	1
0004		MOV A,B	78	1
0005		ANI 0F	E6	2
0006			0F	
0007		INX H	23	1
0008		MOV M,A	77	1
0009		MOV A,B	78	1
000A		ANI F0	E6	2
000B			F0	
000C		RRC	0F	1
000D		RRC	0F	1
000E		RRC	0F	1
000F		RRC	0F	1
0010		INX H	23	1
0011		MOV M,A	77	1
0012		HLT	76	1

OBSERVATION:

RESULT:
Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

PRACTICAL 10

TITLE:

Operations on BCD numbers.

PRIOR CONCEPTS:

1. Data Transfer Instructions
2. Arithmetic Instructions
3. Logical Instructions
4. Branching Group Instructions
5. Machine Control Group Instructions

NEW CONCEPTS:

1. Addition of BCD numbers
2. Subtraction of BCD numbers
3. Multiplication of BCD numbers

THEORY:

This practical consists of programs which deal with calculations related with memory locations. In other words these programs deal with a block or series of data bytes.
This practical includes programs like –

- Addition of two 4-digit BCD numbers
- BCD Subtraction
- BCD Multiplication

a. Add two 4-digit BCD numbers in HL and DE register pairs and store result in memory locations, 7300H and 7301H. Ignore carry after 16 bit.

Address	Label	Mnemonics	Hexcode	Bytes
0000		MOV A,L	7D	1
0001		ADD E	83	1
0002		DAA	27	1
0003		MOV L,A	6F	1
0004		MOV A,H	7C	1
0005		ADC D	8A	1
0006		DAA	27	1
0007		MOV H,A	67	1
0008		SHLD 7300	22	3
0009			00	
000A			73	
000B		HLT	76	1

OBSERVATION:

RESULT:
Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

b. Subtract the BCD number stored in E register from the number stored in the D register.

Address	Label	Mnemonics	Hexcode	Bytes
0000		MVI A,99	3E	2
0001			99	
0002		SUB E	93	1
0003		INR A	3C	1
0004		ADD D	82	1
0005		DAA	27	1
0006		HLT	76	1

OBSERVATION:

RESULT:
Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**

c. Write an assembly language program to multiply 2 BCD numbers.

Address	Label	Mnemonics	Hexcode	Bytes
0000		MVI C,05	0E	2
0001			05	
0002		MVI B,00	06	2
0003			00	
0004		LXI H,0000	21	3
0005			00	
0006			00	
0007		MVI E,02	1E	2
0008			02	
0009		MVI D,00	16	2
000A			00	
000B	BACK	DAD D	19	1
000C		MOV A,L	7D	1
000D		ADI 00	C6	2
000E			00	
000F		DAA	27	1
0010		MOV L,A	6F	1
0011		MOV A,H	7C	1
0012		ACI 00	CE	2
0013			00	
0014		DAA	27	1
0015		MOV H,A	67	1
0016		MOV A,B	78	1
0017		ADI 01	C6	2
0018			01	
0019		DAA	27	1
001A		MOV B,A	47	1
001B		CMP C	B9	1
001C		JNZ BACK	C2	3
001D			0B	
001E			00	
001F		HLT	76	1

OBSERVATION:

RESULT:

Registers:

[A] = [B] =

[C] = [D] =

[E] = [F] =

[H] = [L] =

[PC] = [SP] =

Flag Register:

S	**Z**	**-**	**AC**	**-**	**P**	**-**	**CY**