Module No: 1

A. Familiarization with data transfer instructions
B. Familiarization with arithmetic operations

(Note: Assume the value as hexadecimal if not specified.)

A. Data Transfer Instructions

Data transfer instructions copy data from the register (or I/O or memory) called the source to another register (or I/O or memory) called destination. Following different instructions are involved in data transfer. The data transfer instructions copy data from a source into a destination without modifying the contents of the source. Data transfer instructions do not affect flags.

a) Immediate Data Transfer

This type of instructions copy the immediate data to the destination register. The mnemonic for the immediate data transfer are `MVI` and `LXI`, which means move immediate data. They are used as follows

```
MVI R, data (8-bit)
```

e.g., `MVI A, 94` copies immediate data 94H to accumulator register

Load the following program

```
8000 3E     MVI A, 94
8001 94
8002 EF     RST 5
```

Run the program in single step mode and examine the content of register A before and after the execution of instruction `MVI A, 94`

```
Output:
Before executing: A= . . . .
After executing:  A= . . . .
```

```
LXI Rp data (16-bit)
```

Rp denotes the register pair B, D, H or SP

e.g. `LXI B, 4534` copies immediate data 4534H to register pair B

Load the following program

```
8000 01     LXI B, 2464
8001 64
8002 24
8003 EF     RST 5
```

Run the program in single step mode and examine the content of register pair B before and after the execution of instruction `LXI B, 2464`

```
Output:
Before executing: B= -------- , C--------
After executing:  B= -------- , C--------
```

Assignment

1. Load `1A, 2B, 3C, 4D, 5E, 6F and 7A` to the accumulator, register B, C, D, E, H and L respectively. Examine the content of each register before and after the execution of each instruction. (Use `MVI` instruction)
2. Load `AABB, CCDD and EEFF` into register pair B, D and H respectively. Examine the content of each register before and after the execution of each instruction. (Use `LXI` instruction)

b) Transferring data among registers/memory

This type of instruction copies the data from source register/memory to the destination register/memory. The mnemonic is `MOV`. It is used as follows

```
MOV R, R
MOV R, M
MOV M, R
```

e.g. `MOV B, D` copies content of register D to register B.

For memory related data transfer the H and L registers are used to hold the memory address.

If register pair H has 8090H then, `MOV B, M` copies the content of memory location 8090 to register B.

Load the following program

```
8000  26 80       MVI H, 80
8002  2E 90       MVI L, 90
8004  46          MOV B, M
8005  2E A0       MVI L, A0
8007  70          MOV M, B
8008  EF          RST 5

8090 33           DATA
```

Run the program in single step mode and examine the contents of registers B and H before and after the execution of each instruction. Also examine the content at memory location 80A0 before and after execution of the program.

Output:

```
Before executing: B= -------- , H=--------, 80A0=--------
After executing:  B= -------- , C=--------, 80A0=--------
```

Assignment

3. Load 4455H and 6677H in register pair B and D respectively and exchange their contents. Examine the register contents before and after execution.
4. Write a program to copy content of memory location at 8080 to 8090.

c) Data transfer from and to memory (load/store)

These types of instructions are used to copy the data from memory to specific registers and from specific registers to memory.

LHLD, SHLD, LDAX, STAX, LDA, STA are the instructions for this purpose.

`LHLD 16-bit address` copies the contents of the memory location pointed by 16-bit address in register L and copies the content of next memory location in register H.

`SHLD 16-bit address` copies the contents of register pair H to the memory location pointed by 16-bit address.

`LDAX B/D` copies the content of memory location pointed by register pair B or D to accumulator.

`STAX B/D` copies the content of accumulator to memory location pointed by register pair B or D.

`LDA 16-bit address` copies the contents of a memory location specified by 16-bit address to accumulator.

`STA 16-bit address` copies the contents accumulator to the memory location specified by 16-bit address.

Load the following program

```
8000 01 90 80           LXI B, 8090
8003 11 A0 80           LXI D, 80A0
8006 0A                 LDAX B
8007 12                 STAX D
8008 EF                 RST 5

8090 AA                 DATA
```

Run the program in single step mode and note the contents of accumulator and registers B, C, D and E in each step. Also note the content of memory address 80A0 before and after execution.

Output:

```
Before executing: 80A0=--------
After executing:  80A0=--------
```

Reg/Step	**1st**	**2nd**	**3rd**	**4th**
B				
C				
D				
E				

Load the following program

```
8000 2A 00 81           LHLD 8100
8003 22 00 82           SHLD 8200
8006 EF                 RST 5

8100 AA                 DATA
```

```
8101 BB                 DATA
```

Run the program in single step mode and note the contents of registers H and L in each step. Also note the content of memory address 8200 and 8201 before and after execution.

Output:

Before executing: 8200=--------,8201=-----------------
After executing: 8200=--------,8201=-----------------

Reg/Step	1st	2nd
H		
L		

Load the following program

```
8000 3A 50 80           LDA 8050
8003 32 60 50           STA 8060
8006 EF                 RST 5

8050 99                 DATA
```

Run the program in single step mode and note the contents of accumulator in each step. Also note the content of memory address 8060 before and after execution.

Output:

Before executing: 8060=--------
After executing: 8060=--------

Reg/Step	1st	2nd
A		

Assignment

5. Load 11H, 22H, 33H, 44H and 55H into accumulator and registers B, C, D and E respectively, and store these data in memory location starting at 9000H to 9004H.
6. The data 44H and 66H are stored in memory location 9000H and 9001H respectively. Write a program to transfer these data to memory location 9050H and 9051H respectively. (Hint: use LHLD and SHLD)

d) Exchanging the contents of register pair D and H

This type of instructions used to exchange the contents of register pair D with the contents of register pair H. The content of register D is exchanged with the content of register H, and the content of register E is exchanged with the content of register L.

XCHG

Load the following program

```
8000 11 33 22           LXI D 2233
8003 21 66 55           LXI H 5566
8006 EB                 XCHG
8007 EF                 RST 5
```

Run the program in single step mode and examine the contents register pair D and H before and after execution of each instruction.

Output:

Before executing: DE=-------------,HL=--------
After executing: DE=-------------,HL=--------

Assignment

7. Write a program to exchange the content of register pair B and D with the use of XCHG instruction. Assume data AABBH and CCDDH are in register pair B and D respectively. Examine the register contents before and after execution.

e) Sending data to output port

This type of instructions used to transfer the data from accumulator to the output port.

OUT 8-Bit port address

e.g. OUT 40 sends the data from accumulator to output port A.

Load the following program

```
8000  3E    80          MVI A, 80
8002  D3    43          OUT 43
8004  3E    12          MVI A, 12
8006  01    56 34       LXI B, 3456
8009  D3    40          OUT 40
800B  78                MOV A, B
800C  D3    41          OUT 41
800E  79                MOV A, C
800F  D3    42          OUT 42
8011  EF                RST 5
```

Run the program in single step mode and examine the contents of accumulator, register B and C, and output ports after execution of each instructions.

Output:

Reg/Step	**1st**	**2nd**	**3rd**	**4th**	**5th**	**6th**	**7th**	**8th**	**9th**
A									
B									
C									
Port A (40)									
Port B (41)									
Port C (42)									

B. Arithmetic operations

The 8085 microprocessor performs various arithmetic operations, such as additions, subtractions, increment and decrement. These arithmetic operations have the following mnemonics. Almost all the arithmetic instructions affect the flags to reflect the result with few exceptions

ADD: Add	Add the contents of a register/memory
ADI: Add Immediate	Add immediate 8-bit data
SUB: Subtract	Subtract the contents of a register/memory
SUI: Subtract Immediate	Subtract immediate 8-bit data
INR: Increment	Increase the contents of a register by 1
DCR: Decrement	Decrease the contents of a register by 1

a) Introduction to flags

There are five flip-flops that are set and reset according to the result of some operations in accumulator and other registers. These flags are very useful for microprocessor to perform conditional branching operations.

If an arithmetic or logic operation results zero the flip-flop called the **Zero** (Z) **flag** is set to one. Similarly after an addition of two numbers if sum in the accumulator is larger than eight bits the flip-flop is used to indicate a carry called **Carry** (CY) **flag** is set to one. Similarly auxiliary carry (AC) is set if there is a carry from lower nibble, sign flag (S) is set if the MSB is 1, parity flag (P) is set if the no of 1's of the result is even.

The 8085 has five flags to indicate five different types of data conditions. They are Zero (Z), Carry (CY), Sign (S), Parity (P) and Auxiliary Carry (AC) flags.

Flags				
S	Z	AC	P	CY

Load the following program

```
8000 06 55       MVI B, 55
8002 80                ADD B
8003 80                ADD B
8004 80                ADD B
8005 97                SUB A
8006 37                STC
```

```
8007 3F                 CMC
8008 EF                 RST 5
```

Run the program in single step and examine the content of accumulator and flags in each step.

Output:

Reg/Step	**1st**	**2nd**	**3rd**	**4th**	**5th**	**6th**	**7th**
A							
Flag (CY)							
Flag (P)							
Flag (AC)							
Flag (Z)							
Flag (S)							

b) Adding with Accumulator

These types of instructions add the content of the destination register/memory or immediate data with the accumulator, and result is also stored in accumulator. The flags are modified as according to the result of the addition. Different types of addition instructions are

`ADD R`	add register content to accumulator
`ADD M`	add memory content to accumulator
`ADI DATA(1-BYTE)`	add immediate data to accumulator
`ADC R`	add register to accumulator with carry
`ADC M`	add memory content to accumulator with carry
`ACI DATA(1-BYTE)`	add immediate data to accumulator with carry

e.g. `ADD B` adds the content of register B to the content of accumulator and result is stored in accumulator.

`ADI DATA(1-BYTE)` adds the immediate 1-byte data to the content of accumulator and result is stored in accumulator.

`ADC B` adds the content of register B and carry flag to the content of accumulator and result is stored in accumulator.

`ACI DATA(1-BYTE)` adds the immediate 1-byte data and carry flag to the content of accumulator and result is stored in accumulator.

Load the following program

```
8000 3E A0        MVI A, A0
8002 06 20        MVI B, 20
8004 80           ADD B
8005 C6 50        ADI 50
8007 EF           RST 5
```

Run this program in single step mode and examine the register content and flag in each step.

Output:

Reg/Step	**1st**	**2nd**	**3rd**	**4th**
A				
B				
Flag (CY)				
Flag (P)				
Flag (AC)				
Flag (Z)				
Flag (S)				

Add with carry instructions are used in 16-bit addition. Lower bytes of each number are added by simply add instruction and while adding higher byte add with carry instruction is used to account for the carry generated by low-order bytes.

Let register pair B has 5780H and register pair D has 4293H. To add these two numbers let's see the program

```
8000 01 80 57           LXI B, 5780
8003 11 93 42           LXI D, 4293
8006 79                 MOV A, C
8007 83                 ADD E
8008 4F                 MOV C, A
8009 78                 MOV A, B
800A 82                 ADD D
800B 47                 MOV B, A
800C EF                 RST 5
```

Load the program and run it. Check the result at register pair B. Is it correct?

Now change the instruction at 800A to `ADC D` (op-code `8A`). Again run it and check the result. Is it correct now? Why?

Output:

Before changing code: BC=-------------

After changing code: BC=-------------

Comment:

Assignment

8. Two 16-bit data are stored in memory as shown in table below. Write a program to add 16-bit data at memory location 9000 (4790H) and at 9010 (6283H), and display the result in output ports. Use port A for lower byte and port B for higher byte.

c) Subtract from Accumulator

These types of instructions subtract the content of the destination register/memory from the accumulator. The result is calculated by using the 2’s complement method. The reg/mem content is first converted to 2’s complement and the calculated 2’s complemented is added with the accumulator content. The flag contents are modified as according to the result of the addition. But the carry flag is complemented after the 2’complement addition. The subtraction instructions are identified by `SUB` mnemonic. Different types of subtraction instruction are

SUB R	subtract reg from acc
SUB M	subtract memory content from acc
SUI byte	subtract immediate data from acc
SBB R	subtract reg from acc with borrow
SBB M	subtract memory content from acc with borrow
SBI byte	subtract immediate data from acc with borrow

The `SUB R/M` instruction subtracts the content of register/memory from the accumulator and the result is stored in the accumulator

e.g., `SUB C` instruction subtracts the content of the register c from the accumulator and the result is stored in the accumulator

Similarly `SUB M` instruction subtracts the content of the memory pointed by reg pair H from the accumulator

`SUI byte` instruction subtracts immediate data from the accumulator

e.g., `SUI 34` instruction subtracts 34H from accumulator

`SBB R/M` and `SBI byte` instructions does the same operation as `SUB R/M AND SUI byte` except these instructions take account of the borrow in the previous case. If the carry flag is set during subtraction, it indicated borrow

Load the following program

```
8000 MVI A, 67
8002 MVI B, 56
8004 SUB B
8005 SUI 34
```

```
8006 RST 5
```

Run this program in single step mode and see the flag contents and register contents in each step

Output:

Reg/Step	1st	2nd	3rd	4th
A				
B				
Flag (CY)				
Flag (P)				
Flag (AC)				
Flag (Z)				
Flag (S)				

Generally subtract with borrow instructions are used in 16-bit subtraction. Let register pair B has 3456H and register pair D has 1297H. The solution to this problem is done as follows`8000 LXI B, 3456`

```
8003 LXI D, 1297
8006 MOV A, C
8007 SUB E
8008 MOV C, A
8009 MOV A, B
800A SUB D
800B MOV B, A
800C RST 5
```

load this program and run it. Is the result at register pair B correct? What is the output and what should be the correct output. Now change the instruction at `800A` to `SBB B`. See the output is it correct now? Explain why.

Output:

```
Before changing code: BC=-------------
After changing code:  BC=-------------
```

Comment:

Assignments

9. Suppose the accumulator content is 25H and register B has 48H. Write a program to subtract the reg B from accumulator and store the result in reg C. Again swap the acc content and reg B content and subtract the reg B content from acc content. See the flag content and the reg content before and after the subtraction
10. Suppose the memory content is as shown in the table along side. Write a program to subtract 16-bit data at memory location 8092 (2697H) from data at memory location 8090 (7963H). Store the result at memory location 8094 in reverse order. (16-bit data is always stored in two memory locations in reverse order)

8090	63
8091	79
8092	97
8093	26
8094	
8095	

c) 16-bit Addition (DAD)

This type of instruction is used for the 16-bit addition. Only the carry flag is affected with this instruction. The mnemonic for this type of the instruction is `DAD`, which means double add. `DAD` instruction adds the content of the reg pair with the H reg pair and the result is stored in H reg pair

```
DAD Rp      ;Rp can be B, D, H & SP
```

e.g., DAD B instruction adds the content of reg. pair B with reg. pair H and the result is stored in H pair. Suppose reg. pair B contains A4B9H and reg. pair D has 329AH. These data can be added as

```
8000 LXI B, A4B9                        8008 DAD D
8003 LXI D, 329A                        8009 MOV C, L
8006 MOV L, C                           800A MOV B, H
8007 MOV H, B                           800B RST 5
```

Load this program and observe the output at reg. pair B and the flag condition after the execution

Output:

```
After execution:  BC=-------, Flags=-------------------
```

Assignment

11. Suppose the memory content is as shown in the table along side. Write a program to Add the 16-bit data at memory location 8084 with the 16 bit data at 8086 and store the result at 8088. Use DAD instruction in adding 16 bit data. Verify the result with the 16-bit addition using ADC instruction.

8084	A2
8085	79
8086	4B
8087	C4
8088	
8089	

d) Increment and Decrement Instructions (INR, DCR, INX, DCX)

These instructions are used to increase the reg/mem content by one. These instructions does not require accumulator to be the operand.

INR and DCR are the instructions for increasing and decreasing reg/mem data by one. All the flags except carry are affected by these instructions. These instructions are used as follows

```
INR R       increase reg. content by one
INR M       increase mem. content pointed by H&L by one
DCR R       decrease reg. content by one
DCR M             decrease mem. content pointed by H&L by one
```

e.g., INR D increases the content of reg. D by one and INR M increases the content of the memory location pointed by H&L.

DCR C decreases the content of reg. C by one and DCR M decrease the content of memory location pointed by H&L

Load the following program

```
8000 MVI B, 00                           8007 MOV A, C
8002 MVI C, 45                           8008 DCR A
8004 MOV A, B                            8009 MOV C, A
8005 INR A                               800A RST 5
8006 MOV B, A
```

Run this program and see the content of register A and flag condition before and after the execution of INR and DCR instructions.

Output:

```
Before execution: A=--------, Flags------------------
After  execution:  A=--------, Flags=------------------
```

INX and DCX are the instructions for increasing and decreasing register pair content (16-bit data). These instructions affect no flags. They are used as follows

```
INX Rp            increase reg pair by one
DCX Rp            decrease reg pair by one
```

Rp can be B, D, H, SP

e.g., if reg B pair has 34FFH, then INX B increases the contents of reg. pair B to 3500H. Similarly if reg pair D has 2300H, then DCX D decreases the content of reg pair D to 22FFH

Load the following program

```
8000 LXI B, 341F
8003 INX B
8004 LXI D, 231E
```

```
8007 DCX D
8008 RST 5
```

Run this program and see the content of register and flags before and after the execution. Instead of using `INX B` and `DCX D` instructions can't we use `INR C` and `DCR E` instructions? Can we always use `INR` and `DCR` instructions to decrease the reg pair content? Give reasons

```
Output:
Before execution: B=----, D=----, Flags=------------------
After execution:  B=----, D=----, Flags=------------------
```

Comment:

Assignment

12. The data in the memory is as shown in the table along side. Write a program to increase the 16-bit data at memory location 80A2 and decrease the 16-bit data at memory location 80A4.

80A2	FF
80A3	A2
80A4	00
80A5	26

13. Use `INR` and `DCR` instructions for the increment and decrement operation to increase 16-bit data in the above problem? (Hint: use `INR` and `DCR` instruction for the lower byte data)

e) Decimal Operation (DAA)

DAA is the instruction to adjust the accumulator content after the binary addition operation. The `DAA` instruction uses `CY` and `Y` flags for the conversion. This instruction assumes the operands of the addition are BCD numbers but not binary. If the value of the low-order four bits (D3-D0) in the accumulator is greater than 9 or if AC flag is set, the instruction adds 6 (06) to the low-order four bits. If the value of the high-order four bits (D7-D4) in the accumulator is greater than 9 or if the carry flag is set, the instruction adds 6 (60) to the high-order four bits. All the flags are affected to reflect the result of DAA.

e.g., if 39 BCD is added with 12 BCD the result is 4B in binary and 51 in BCD. The use of `DAA` adjust the accumulator for the decimal result

Load the following program

```
8000 MVI A, 15
8002 MVI B, 87
8004 ADD B
8005 DAA
```

Load this program and examine the accumulator content before and after `ADD` instruction, and after `DAA` instruction. Also note the flag conditions at these steps.

```
Output:
Before ADD instruction: A=----, Flags=------------------
After DAA instruction:  A=----, Flags=------------------
```

For the 16-bit decimal addition the `DAA` is used as follow

```
8000 LXI B, 4985                            800A MOV A, B
8003 LXI D, 2668                            800B ADC D
8006 MOV A, C                               800C DAA
8007 ADD E                                  800D MOV B, A
8008 DAA                                    800E RST 5
8009 MOV C, A
```

Load this program and see the content of reg. pair B at the end of the program. Is this value the correct decimal value if the operands are BCD numbers? What will be the binary sum of 4B85 and 2668?

```
Output:
After Execution:  BC=---------
Comment:
```

Decimal subtraction can be done by using 10's complement method. For the BCD subtraction process convert the subtrahend to 10's complement and add it with minuend. Use DAA after 10's complement addition.

Assignment

14. The table along side shows BCD data. Write a program to add the content at memory locations 818A and 818B. Store the BCD result at memory location 818C.

818A	29
818B	45
818C	

15. Write a program to Add the 16-bit BCD data at memory location 8284 with the 16-bit BCD data at 8286 and store the BCD result at 8288.

8284	13
8285	79
8286	26
8287	48
8288	
8289	

16. Suppose reg H has 34 and reg D has 12. Write program to subtract BCD value in reg D from the BCD value in reg H. Store the BCD result in reg H.
17. Using the same data of problem 15 write a program to subtract BCD data at 8286 from BCD data at 8284 and store the BCD result at 8288.