Lab: 2
Familiarization with arithmetic instructions

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	X	AC	X	P	X	CY

Load the following program

```
8000 MVI B, 55
8002 ADD B
8003 ADD B
8004 ADD B
8005 SUB A
8006 STC
8007 CMC
8008 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 MVI A, A0
8002 MVI B, 20
8004 ADD B
8005 ADI 50
8007 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 BC has 5780H and register pair DE has 4293H. To add these two numbers let's see the program

```
8000 LXI B, 5780
8003 LXI D, 4293
8006 MOV A, C
8007 ADD E
8008 MOV C, A
8009 MOV A, B
800A ADD D
800B MOV B, A
800C RST 5
```

Load the program and run it. Check the result at register pair BC. 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=-------------
```

Assignment

1. 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 HL 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 BC has 3456H and register pair DE 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 BC 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=-------------
```

Assignments

2. 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
3. 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 HL reg pair and the result is stored in HL reg pair

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

e.g., DAD B instruction adds the content of reg. pair BC with reg. pair HL and the result is stored in HL pair. Suppose reg. pair BC contains A4B9H and reg. pair DE 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 BC and the flag condition after the execution

Output:

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

Assignment

4. 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                      8006 MOV B, A
8002 MVI C, 45                      8007 MOV A, C
8004 MOV A, B                       8008 DCR A
8005 INR A                          8009 MOV C, A
```

```
800A RST 5
```

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 BC pair has 34FFH, then INX B increases the contents of reg. pair BC to 3500H. Similarly if reg pair DE has 2300H, then DCX D decreases the content of reg pair DE 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------------------
```

Assignment

5. 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

6. 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
8003 LXI D, 2668
8006 MOV A, C
8007 ADD E
8008 DAA
8009 MOV C, A
800A MOV A, B
800B ADC D
800C DAA
800D MOV B, A
800E RST 5
```

Load this program and see the content of reg. pair BC 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=---------
```

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

7. 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	

8. 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	

9. 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.