

Name:- Bambhaniya Jaydev Lakhman Bhai

ID:- 19CEUOF027

Roll No:- CE008

Lab:-03

1. To study about the LDAX,STAX,LHLD and SHLD instructions. Study about the jump instructions.

Ans:-

1. LDAX :

- Load accumulator indirect.
- The content of the designated register pair point to a memory location, this instruction copies that contents into accumulator.
- LDAX B

2. STAX :

- Store the accumulator indirect.
- The content of the accumulator are copied into the memory location specified by the contents of the register pair.
- STAX B

3. LHLD :

- Load H-L register direct.
- The content of the specified address is copied into L register and the content of the H register is store by its next memory location.
- LHLD 2040H

4. SHLD :

- Store H-L register direct.
- The content of the L register is copied into specified address and content of the H register is copied into the next memory location.
- SHLD 2550H

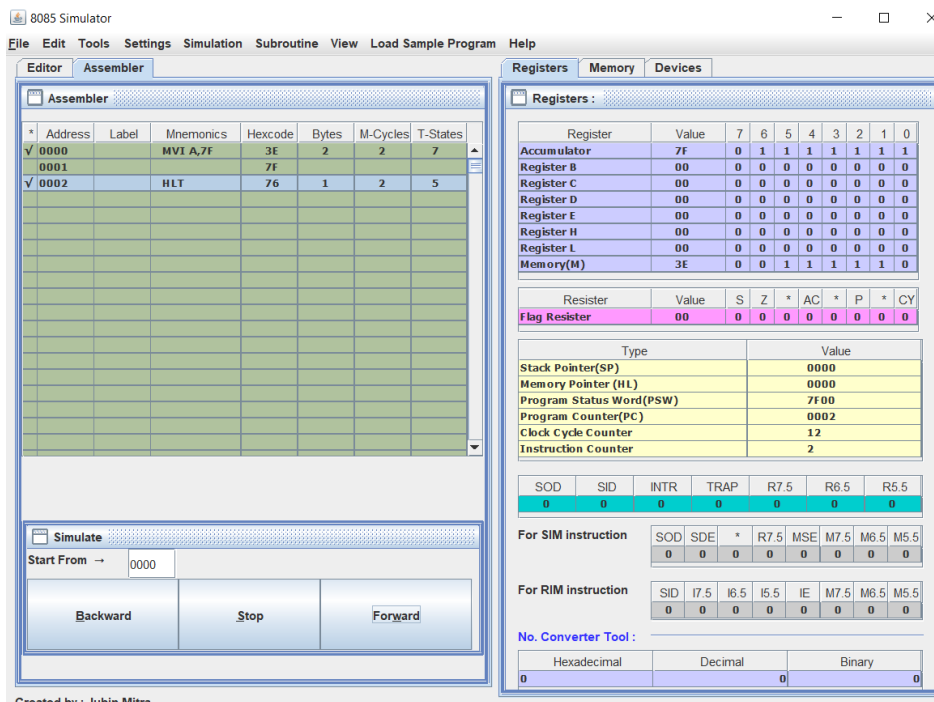
❖ Jump Instructions

- It alters the normal sequential flow; this is conditionally or unconditionally.
- E.g JMP 2034H
- The program sequence is transferred to the memory location specified by the 16-bit address given in the operand.
- JZ, JNZ, JNC, JC, JPE, JPO, JM etc...
- JZ – Jump if zero, JNZ -- Jump if not zero.
- JC – Jump if carry, JNC – Jump if no carry.
- JPE – Jump if even parity, JPO – Jump if odd parity.
- JM – Jump if Minus, JP – Jump if positive.

2. Identify the contents of the accumulator/registers and the flag status as the following instructions are executed.

a) MVI A,7F

- As you can see value 7F is stored in the accumulator and no any flag or other register is changed.



b) ORA A

- ORA function is do the ORing with the accumulator, here operand is also accumulator and the value of accumulator is 00 so no change in accumulator, but as you can see the Zero flag is set and also the parity flag is also set.

8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor Assembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
0000		ORA A	B7	1	1	4

Simulate

Start From → 0000

Backward Stop Forward

Created by : Jubin Mitra

Registers Memory Devices

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	B7	1	0	1	1	0	1	1	1

Register	Value	S	Z	* AC	* P	* CY
Flag Register	44	0	1	0	1	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer(HL)	0000
Program Status Word(PSW)	0044
Program Counter(PC)	0002
Clock Cycle Counter	8
Instruction Counter	2

SOD	SID	INTR	TRAP	R7.5	R6.5	R5.5
0	0	0	0	0	0	0

For SIM instruction

SOD	SDE	* R7.5	MSE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0

For RIM instruction

SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

No. Converter Tool :

Hexadecimal	Decimal	Binary
0		0

c) CPI A2H

- CPI instruction is doing the comparison between current value of accumulator and the operand passed, here the value of accumulator value is zero, so the A2 is higher than 00 so the Carry flag is set.

8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor Assembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
0000		CPI A2	7E	2	2	7

Simulate

Start From → 0000

Backward Stop Forward

Created by : Jubin Mitra

Registers Memory Devices

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	FE	1	1	1	1	1	1	1	0

Register	Value	S	Z	* AC	* P	* CY
Flag Register	01	0	0	0	0	1

Type	Value
Stack Pointer(SP)	0000
Memory Pointer(HL)	0000
Program Status Word(PSW)	0001
Program Counter(PC)	0002
Clock Cycle Counter	7
Instruction Counter	1

SOD	SID	INTR	TRAP	R7.5	R6.5	R5.5
0	0	0	0	0	0	0

For SIM instruction

SOD	SDE	* R7.5	MSE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0

For RIM instruction

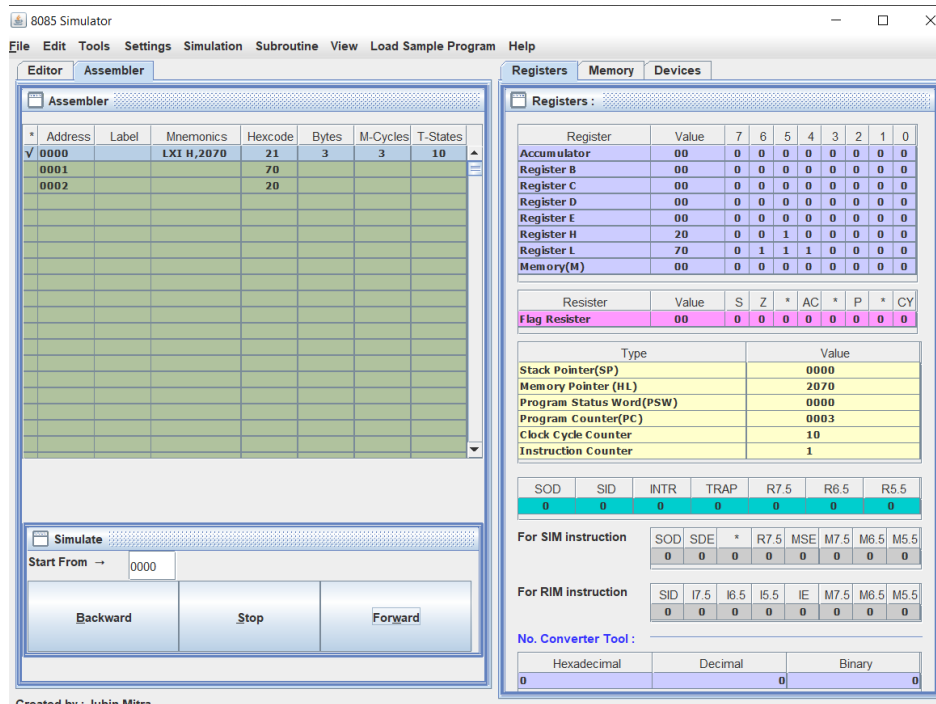
SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

No. Converter Tool :

Hexadecimal	Decimal	Binary
0		0

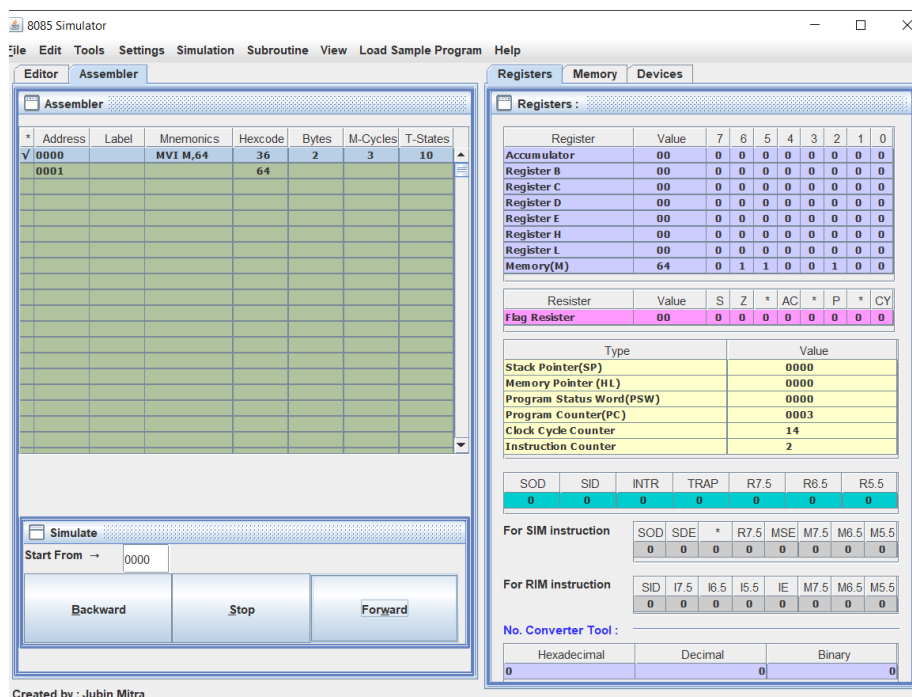
d) LXI H,2070H

- LXI is do nothing just value is set in the H-L register so as you can see the H is 20 and the L is 70 and no any flag is set.



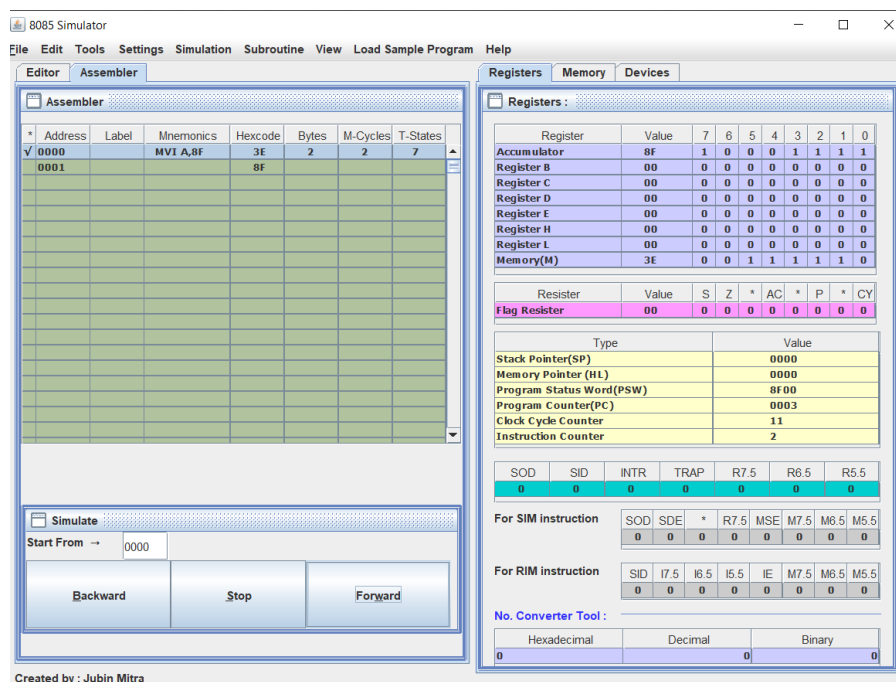
e) MVI M,64H

- MVI instruction set the value of register of Memory whichever passed, here M is passed means the current H-L register at the memory location the value 64 is stored. So no any change is flags.



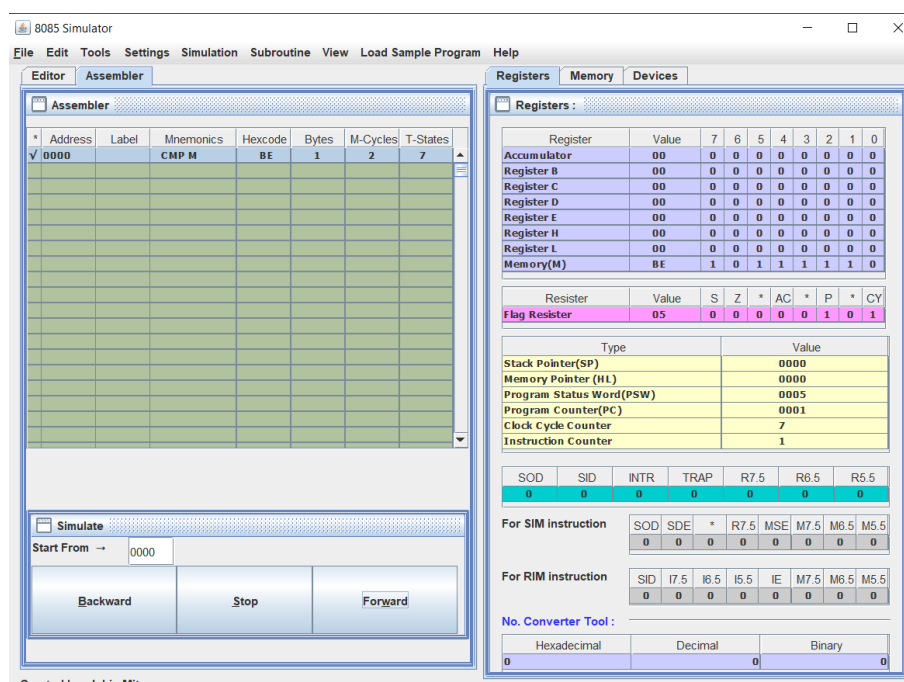
f) MVI A,8FH

- As I tell you in above example MVI change the value of register(Accumulator) to 8F, so any changes in the flag.



g) CMP M

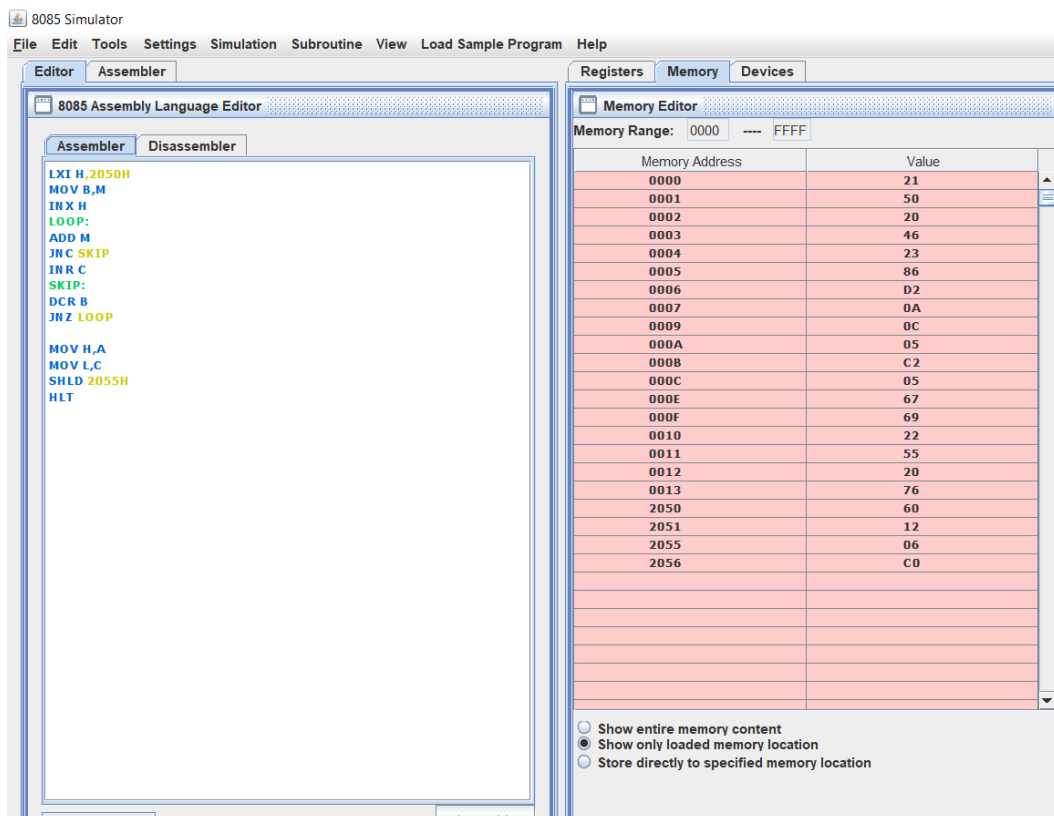
- CMP instruction is used to compare the value of accumulator and the register or value of memory which is passed, here the value of accumulator is 00 and the currently the value of memory is considered as higher than the accumulator so the carry flag is set, and as well as the parity flag is also set.



3. Multiply two numbers given at address 2050H and 2051H respectively.

Ans:-

```
LXI H,2050H
MOV B,M
INX H
LOOP:
ADD M
JNC SKIP
INR C
SKIP:
DCR B
JNZ LOOP
MOV H,A
MOV L,C
SHLD 2055H
HLT
```



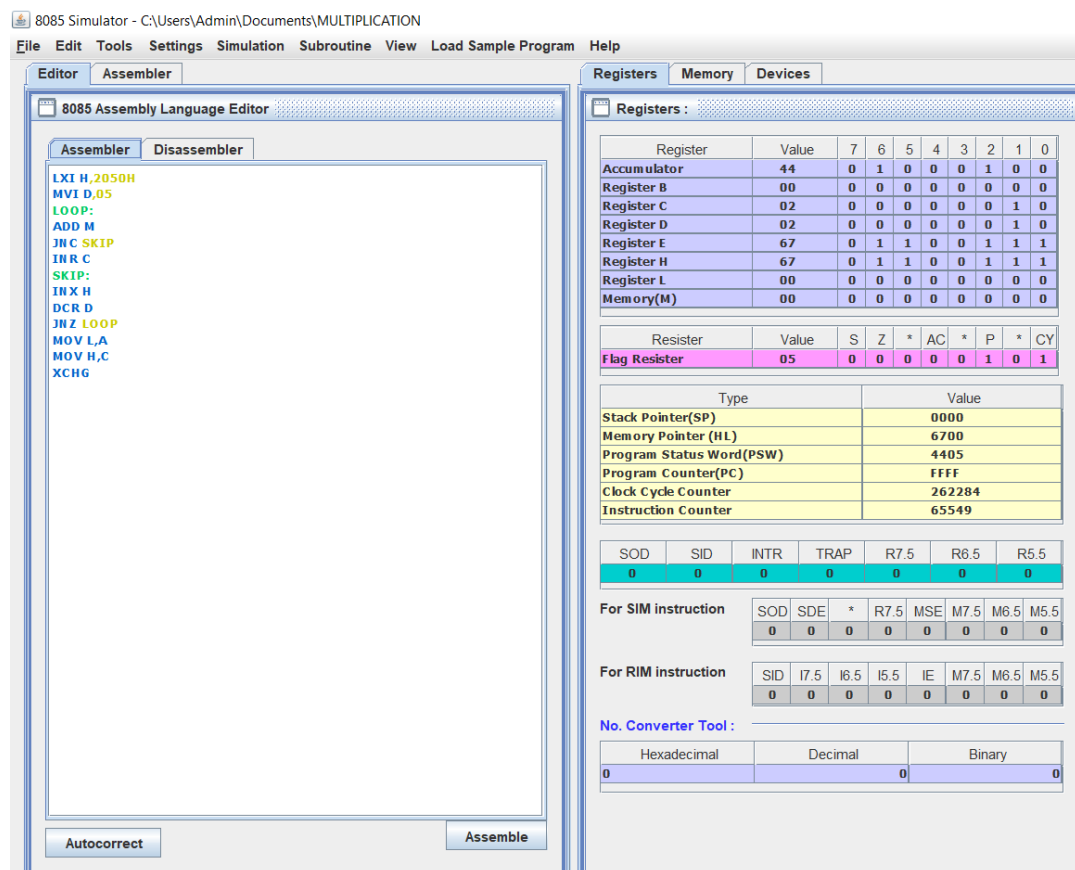
4. Write a program to add the 5 bytes of data starting at address 2050H and store the result at DE pair.

Ans:-

```

LXI H,2050H
MVI D,05
LOOP:
ADD M
JNC SKIP
INR C
SKIP:
INX H
DCR D
JNZ LOOP
MOV L,A
MOV H,C
XCHG

```



5. Using the jump instruction transfer block of 8 bytes of data starting from 2055H to 3055H in reverse order.

Ans:-

```

LXI B,205C
LXI H,3055

```

```

MVI D,08
LOOP:
LDAX B
MOV M,A
DCR D
INX H
DCX B
JNZ LOOP

```

8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor Assembler

8085 Assembly Language Editor

Assembler Disassembler

```

LXI B,205C
LXI H,3055
MVI D,08
LOOP:
LDAX B
MOV M,A
DCR D
INX H
DCX B
JNZ LOOP

```

Autocorrect Assemble

Registers Memory Devices

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	08	0	0	0	0	1	0	0	0
Register B	06	0	0	0	0	0	1	1	0
Register C	55	0	1	0	1	0	1	0	1
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	30	0	0	1	1	0	0	0	0
Register L	5D	0	1	0	1	1	1	0	1
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	04	0	0	0	0	0	1	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	305D
Program Status Word(PSW)	0804
Program Counter(PC)	FFFF
Clock Cycle Counter	262410
Instruction Counter	65566

SOD	SID	INTR	TRAP	R7.5	R6.5	R5.5
0	0	0	0	0	0	0

For SIM instruction

SOD	SDE	*	R7.5	MSE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

For RIM instruction

SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

No. Converter Tool :

Hexadecimal	Decimal	Binary
0		0