INSTITUTE OF COMPUTER TECHNOLOGY

B-TECH COMPUTER SCIENCE ENGINEERING 2025-26

SUBJECT: MICROCONTROLLER & APPLICATIONS

NAME: Rahul Prajapati

ENRLL. NO: 23162171020

BRANCH: CYBER SECURITY

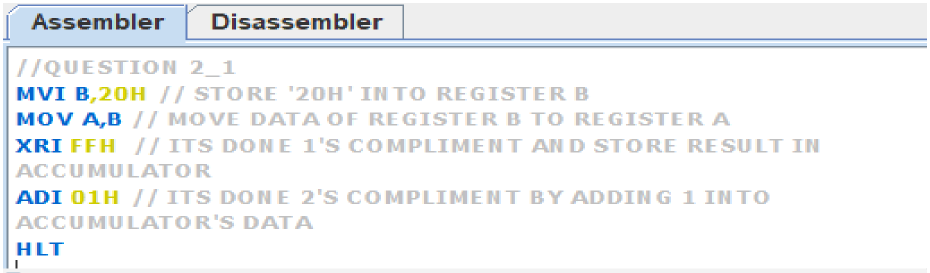
BATCH: 52

PRACTICAL_02

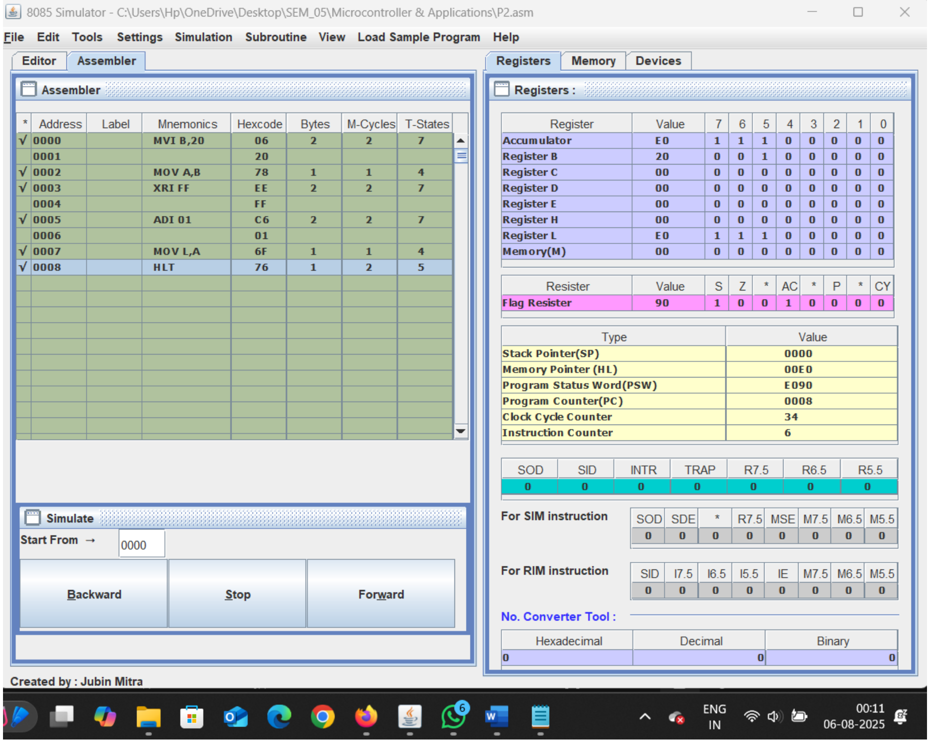
Aim: Learning Programs using Logical Instructions like ANA, ANI, ORA, ORI, XRA, XRI, CMA, RAL, RRC, RAR, CMP, CPI etc.

Exercise :

1. To find 2's complement of 8-bit data without CMA instruction. Take 8-bit data (Last two digit of your enrollment number) in B register and store 2's complement of that number in Reg-L.



```
//QUESTION 2_1
MVI B,20H // STORE '20H' INTO REGISTER B
MOV A,B // MOVE DATA OF REGISTER B TO REGISTER A
XRI FFH // ITS DONE 1'S COMPLIMENT AND STORE RESULT IN
ACCUMULATOR
ADI 01H // ITS DONE 2'S COMPLIMENT BY ADDING 1 INTO
ACCUMULATOR'S DATA
HLT
```



*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
√	0000		MVI B,20	06	2	2	7
	0001			20			
√	0002		MOV A,B	78	1	1	4
√	0003		XRI FF	EE	2	2	7
	0004			FF			
√	0005		ADI 01	C6	2	2	7
	0006			01			
√	0007		MOV L,A	6F	1	1	4
√	0008		HLT	76	1	2	5

Register	Value	7	6	5	4	3	2	1	0
Accumulator	E0	1	1	1	0	0	0	0	0
Register B	20	0	0	1	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	E0	1	1	1	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	90	1	0	0	1	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	00E0
Program Status Word(PSW)	E090
Program Counter(PC)	0008
Clock Cycle Counter	34
Instruction Counter	6

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

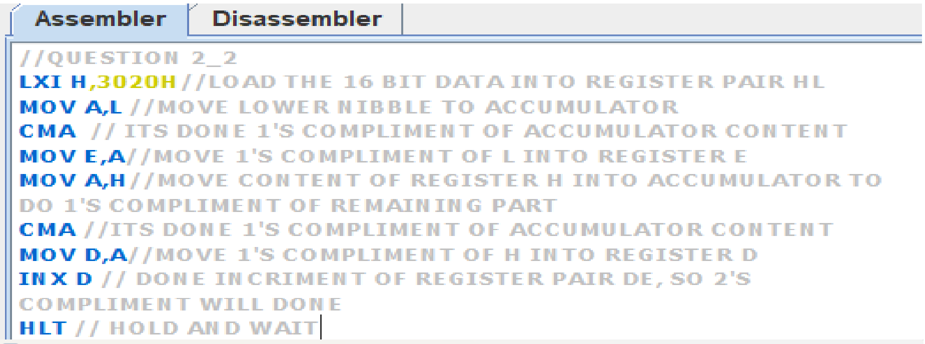
No. Converter Tool :

Hexadecimal	Decimal	Binary
0	0	0

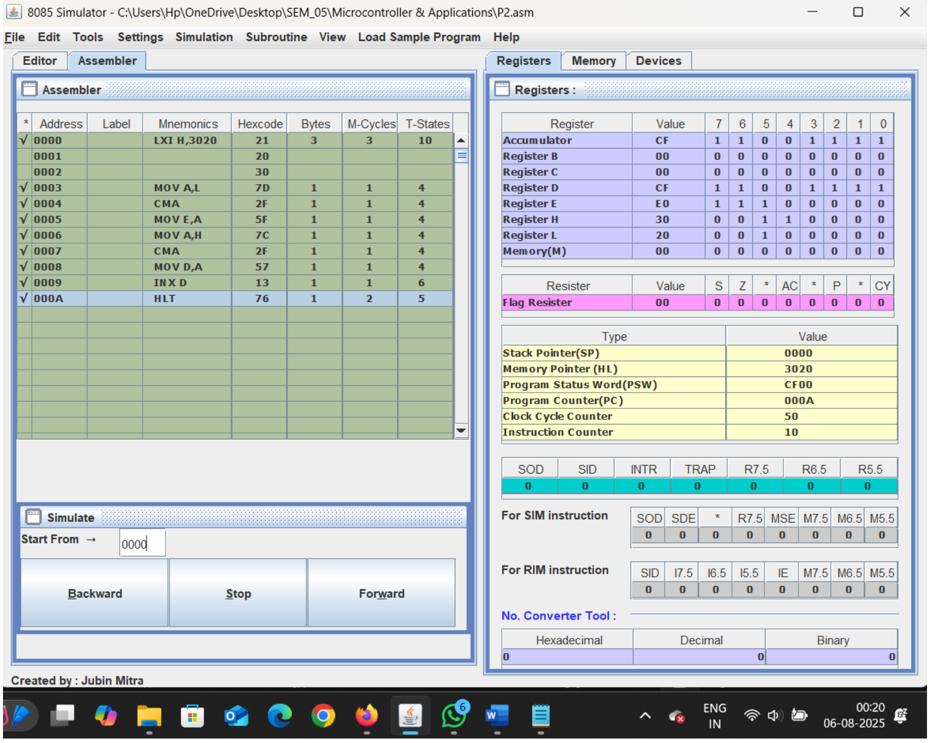
Simulate

Start From → 0000

2. To find 2's complement of given 16-bit number. Take 16-bit number (3 appended with last three digits of your enrollment number) in Register pair HL and 2's complement in DE register pair.



```
//QUESTION 2_2
LXI H,3020H//LOAD THE 16 BIT DATA INTO REGISTER PAIR HL
MOV A,L //MOVE LOWER NIBBLE TO ACCUMULATOR
CMA // ITS DONE 1'S COMPLIMENT OF ACCUMULATOR CONTENT
MOV E,A//MOVE 1'S COMPLIMENT OF L INTO REGISTER E
MOV A,H//MOVE CONTENT OF REGISTER H INTO ACCUMULATOR TO
DO 1'S COMPLIMENT OF REMAINING PART
CMA //ITS DONE 1'S COMPLIMENT OF ACCUMULATOR CONTENT
MOV D,A//MOVE 1'S COMPLIMENT OF H INTO REGISTER D
INX D // DONE INCRIMENT OF REGISTER PAIR DE, SO 2'S
COMPLIMENT WILL DONE
HLT // HOLD AND WAIT
```



*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
√	0000		LXI H,3020	21	3	3	10
	0001			20			
	0002			30			
√	0003		MOV A,L	7D	1	1	4
√	0004		CMA	2F	1	1	4
√	0005		MOV E,A	5F	1	1	4
√	0006		MOV A,H	7C	1	1	4
√	0007		CMA	2F	1	1	4
√	0008		MOV D,A	57	1	1	4
√	0009		INX D	13	1	1	6
√	000A		HLT	76	1	2	5

Start From → 0000

Register	Value	7	6	5	4	3	2	1	0
Accumulator	CF	1	1	0	0	1	1	1	1
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	CF	1	1	0	0	1	1	1	1
Register E	E0	1	1	1	0	0	0	0	0
Register H	30	0	0	1	1	0	0	0	0
Register L	20	0	0	1	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	3020
Program Status Word(PSW)	CF00
Program Counter(PC)	000A
Clock Cycle Counter	50
Instruction Counter	10

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 :

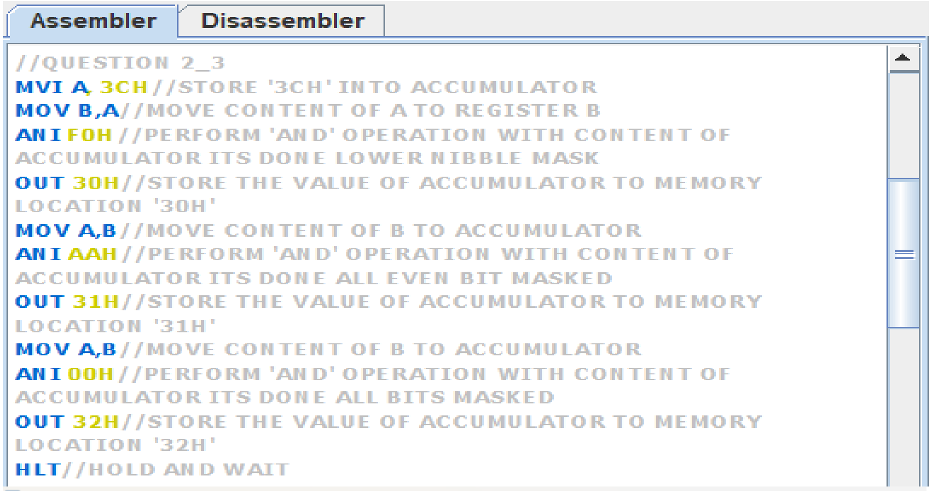
Hexadecimal	Decimal	Binary
0	0	0

3. To learn masking patterns and hence making specific bits to zero. Take one 8-bit data that is multiplied by 3 with last two digits of your enrollment number. Perform operations to result as follows and display on the port 30H, 31H and 32H respectively:

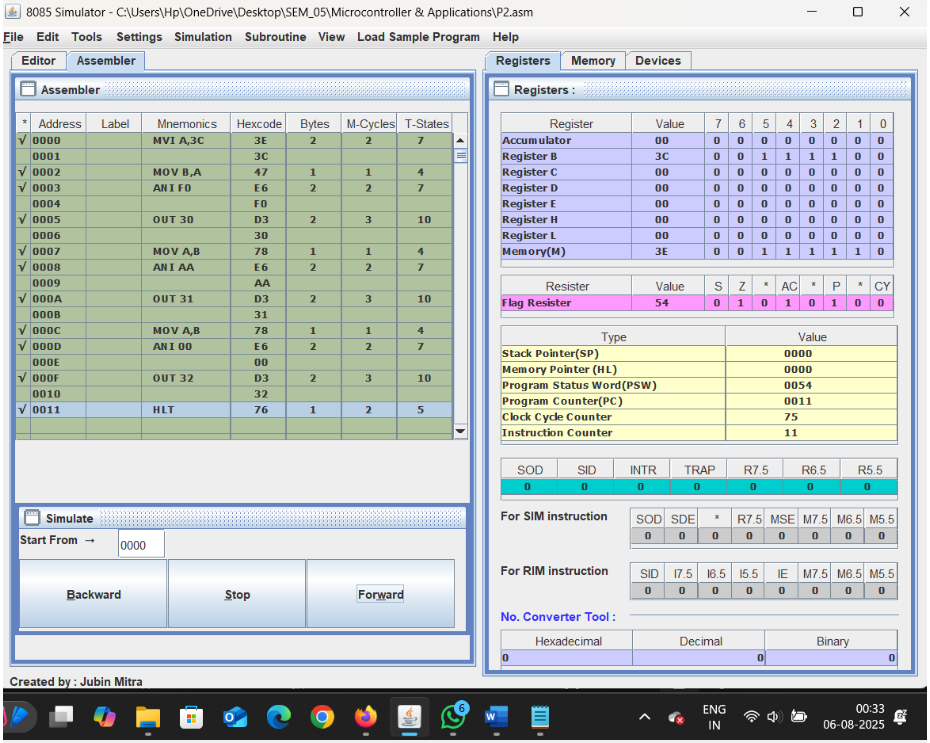
Case 1: Lower nibble should be masked and upper nibble should remain unchanged.

Case 2: All even bits shall be masked.

Case 3: Answer after masking becomes zero.



```
//QUESTION 2_3
MVI A, 3CH//STORE '3CH' INTO ACCUMULATOR
MOV B,A//MOVE CONTENT OF A TO REGISTER B
ANI F0H//PERFORM 'AND' OPERATION WITH CONTENT OF ACCUMULATOR ITS DONE LOWER NIBBLE MASK
OUT 30H//STORE THE VALUE OF ACCUMULATOR TO MEMORY LOCATION '30H'
MOV A,B//MOVE CONTENT OF B TO ACCUMULATOR
ANI AAH//PERFORM 'AND' OPERATION WITH CONTENT OF ACCUMULATOR ITS DONE ALL EVEN BIT MASKED
OUT 31H//STORE THE VALUE OF ACCUMULATOR TO MEMORY LOCATION '31H'
MOV A,B//MOVE CONTENT OF B TO ACCUMULATOR
ANI 00H//PERFORM 'AND' OPERATION WITH CONTENT OF ACCUMULATOR ITS DONE ALL BITS MASKED
OUT 32H//STORE THE VALUE OF ACCUMULATOR TO MEMORY LOCATION '32H'
HLT//HOLD AND WAIT
```



*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
√	0000		MVI A,3C	3E	2	2	7
	0001			3C			
√	0002		MOV B,A	47	1	1	4
√	0003		ANI F0	E6	2	2	7
	0004			F0			
√	0005		OUT 30	D3	2	3	10
	0006			30			
√	0007		MOV A,B	78	1	1	4
√	0008		ANI AA	E6	2	2	7
	0009			AA			
√	000A		OUT 31	D3	2	3	10
	000B			31			
√	000C		MOV A,B	78	1	1	4
√	000D		ANI 00	E6	2	2	7
	000E			00			
√	000F		OUT 32	D3	2	3	10
	0010			32			
√	0011		HLT	76	1	2	5

Start From → 0000

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	3C	0	0	1	1	1	1	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)	3E	0	0	1	1	1	1	1	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	54	0	1	0	1	0	1	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0054
Program Counter(PC)	0011
Clock Cycle Counter	75
Instruction Counter	11

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

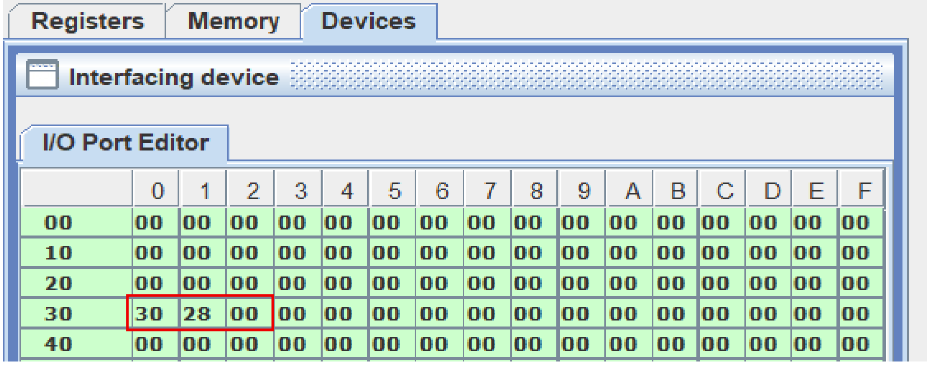
For RIM instruction

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	0

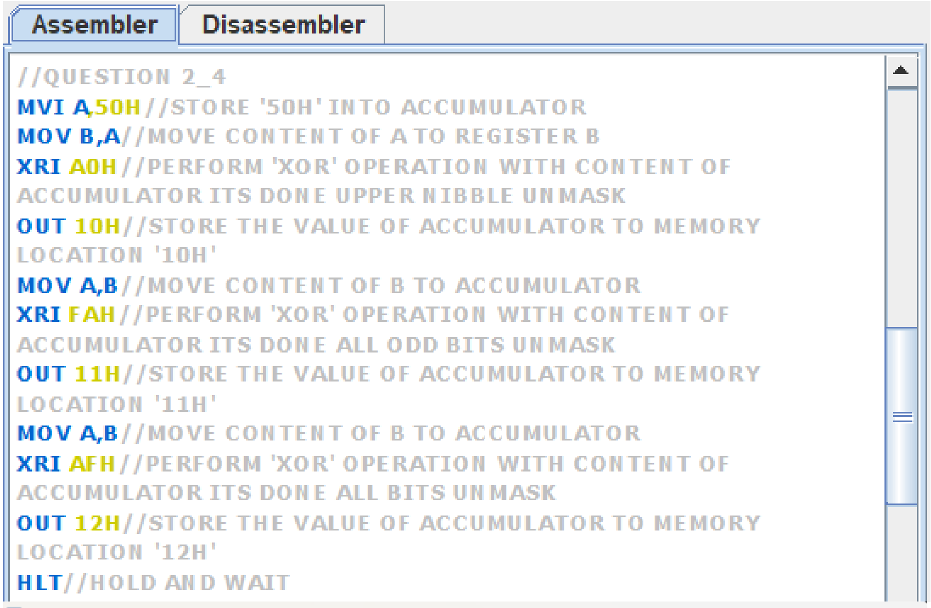
Created by : Jubin Mitra



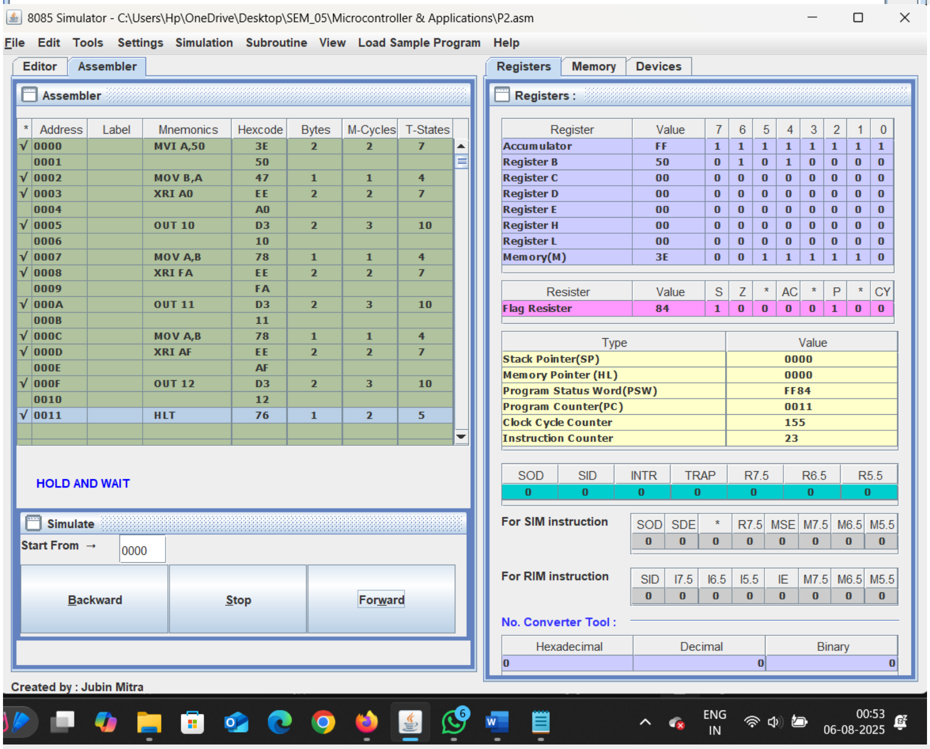
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	30	28	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

4. To learn unmasking patterns and hence making specific bits to one. Take one 8-bit that is multiplied by 4 with last two digits of your enrollment number. Perform operations to result as follows and display on the port 10H, 11H and 12H respectively:

   Case 1: Upper nibble should be unmasked and lower nibble should remain unchanged.
   Case 2: All odd bits shall be unmasked.
   Case 3: Answer after unmasking becomes all bits one.



```
//QUESTION 2_4
MVI A,50H//STORE '50H' INTO ACCUMULATOR
MOV B,A//MOVE CONTENT OF A TO REGISTER B
XRI A0H//PERFORM 'XOR' OPERATION WITH CONTENT OF ACCUMULATOR ITS DONE UPPER NIBBLE UNMASK
OUT 10H//STORE THE VALUE OF ACCUMULATOR TO MEMORY LOCATION '10H'
MOV A,B//MOVE CONTENT OF B TO ACCUMULATOR
XRI FAH//PERFORM 'XOR' OPERATION WITH CONTENT OF ACCUMULATOR ITS DONE ALL ODD BITS UNMASK
OUT 11H//STORE THE VALUE OF ACCUMULATOR TO MEMORY LOCATION '11H'
MOV A,B//MOVE CONTENT OF B TO ACCUMULATOR
XRI AFH//PERFORM 'XOR' OPERATION WITH CONTENT OF ACCUMULATOR ITS DONE ALL BITS UNMASK
OUT 12H//STORE THE VALUE OF ACCUMULATOR TO MEMORY LOCATION '12H'
HLT//HOLD AND WAIT
```



8085 Simulator - C:\Users\Hp\OneDrive\Desktop\SEM_05\Microcontroller & Applications\P2.asm

*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
√	0000		MVI A,50	3E	2	2	7
	0001			50			
√	0002		MOV B,A	47	1	1	4
√	0003		XRI A0	EE	2	2	7
	0004			A0			
√	0005		OUT 10	D3	2	3	10
	0006			10			
√	0007		MOV A,B	78	1	1	4
√	0008		XRI FA	EE	2	2	7
	0009			FA			
√	000A		OUT 11	D3	2	3	10
	000B			11			
√	000C		MOV A,B	78	1	1	4
√	000D		XRI AF	EE	2	2	7
	000E			AF			
√	000F		OUT 12	D3	2	3	10
	0010			12			
√	0011		HLT	76	1	2	5

HOLD AND WAIT

Start From → 0000

Register	Value	7	6	5	4	3	2	1	0
Accumulator	FF	1	1	1	1	1	1	1	1
Register B	50	0	1	0	1	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)	3E	0	0	1	1	1	1	1	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	84	1	0	0	0	0	1	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	FF84
Program Counter(PC)	0011
Clock Cycle Counter	155
Instruction Counter	23

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

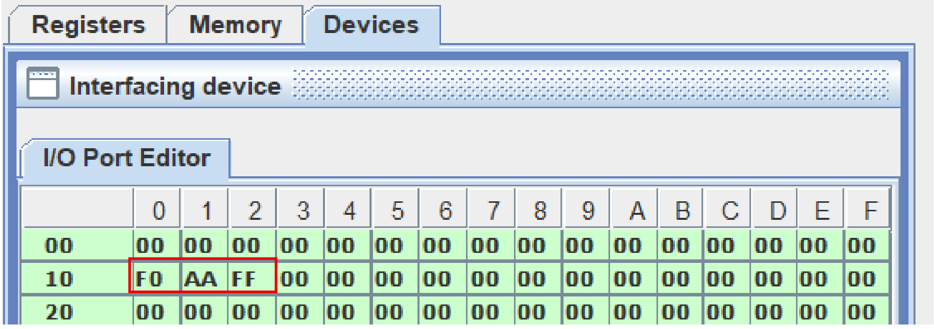
For RIM instruction

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 :

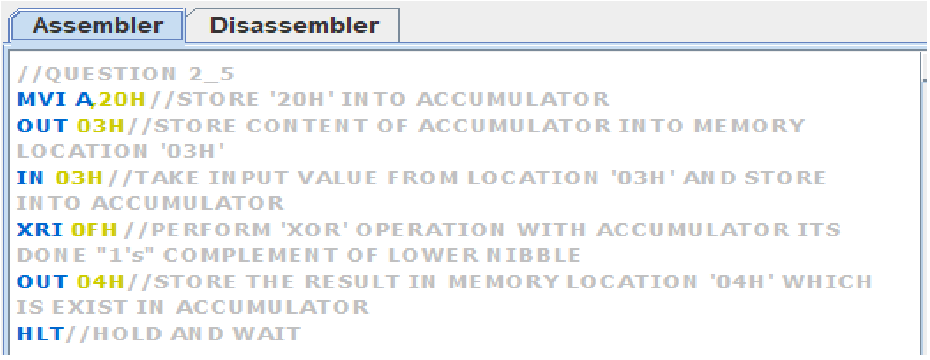
Hexadecimal	Decimal	Binary
0	0	0

Created by : Jubin Mitra

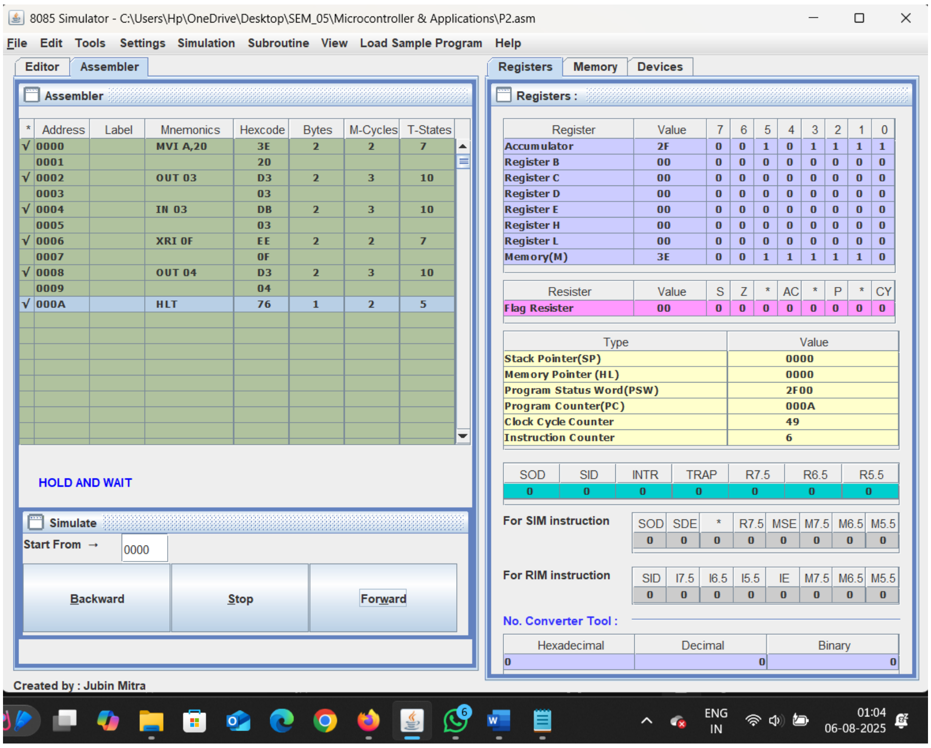


	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10	F0	AA	FF	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

5. Get data byte from input port 03H and complement Lower Nibble. Store the result on next memory location.



```
//QUESTION 2_5
MVI A,20H//STORE '20H' INTO ACCUMULATOR
OUT 03H//STORE CONTENT OF ACCUMULATOR INTO MEMORY
LOCATION '03H'
IN 03H//TAKE INPUT VALUE FROM LOCATION '03H' AND STORE
INTO ACCUMULATOR
XRI 0FH//PERFORM 'XOR' OPERATION WITH ACCUMULATOR ITS
DONE "1's" COMPLEMENT OF LOWER NIBBLE
OUT 04H//STORE THE RESULT IN MEMORY LOCATION '04H' WHICH
IS EXIST IN ACCUMULATOR
HLT//HOLD AND WAIT
```



8085 Simulator - C:\Users\Hp\OneDrive\Desktop\SEM_05\Microcontroller & Applications\P2.asm
File Edit Tools Settings Simulation Subroutine View Load Sample Program Help
Editor Assembler
Registers Memory Devices
Assembler
Registers :
Simulate
Backward Stop Forward
Created by : Jubin Mitra
ENG IN
01:04
06-08-2025

*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
√	0000		MVI A,20	3E	2	2	7
	0001			20			
√	0002		OUT 03	D3	2	3	10
	0003			03			
√	0004		IN 03	DB	2	3	10
	0005			03			
√	0006		XRI 0F	EE	2	2	7
	0007			0F			
√	0008		OUT 04	D3	2	3	10
	0009			04			
√	000A		HLT	76	1	2	5

HOLD AND WAIT

Start From → 0000

Register	Value	7	6	5	4	3	2	1	0
Accumulator	2F	0	0	1	0	1	1	1	1
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)	3E	0	0	1	1	1	1	1	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	2F00
Program Counter(PC)	000A
Clock Cycle Counter	49
Instruction Counter	6

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

For SIM instruction

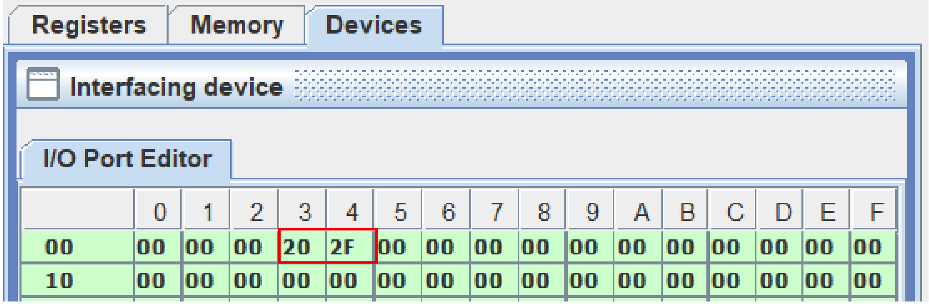
SOD	SDE	*	R7.5	MSE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

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

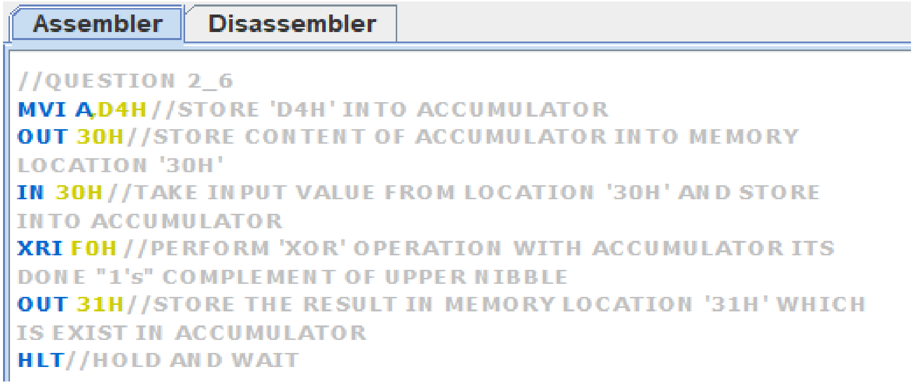
No. Converter Tool :

Hexadecimal	Decimal	Binary
0	0	0

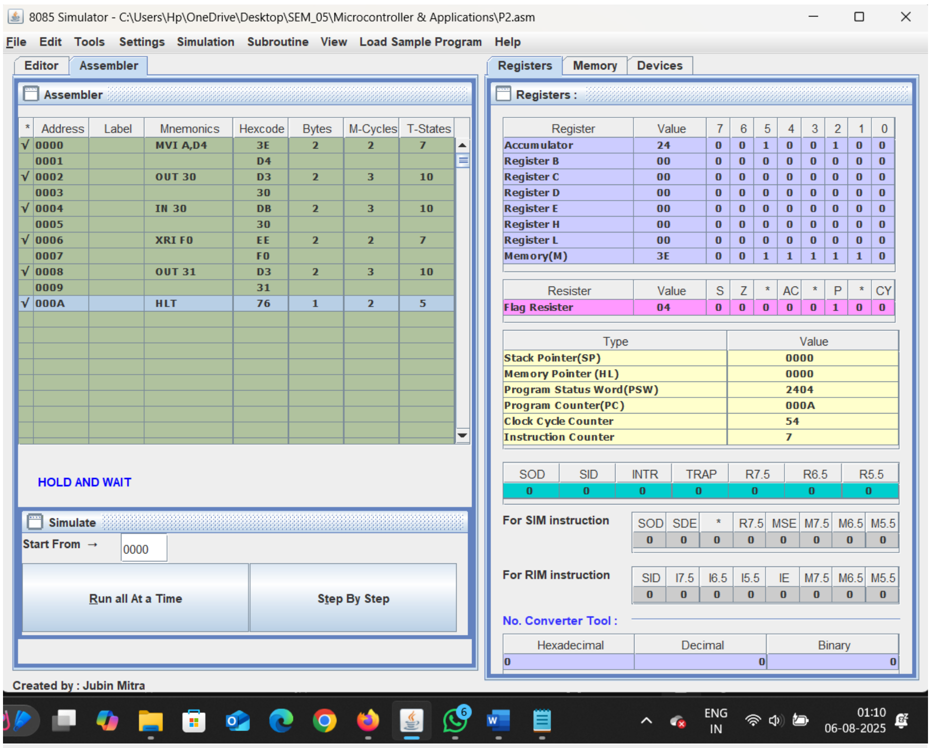


	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	00	00	20	2F	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

6. Get data byte from input port 30H and complement Upper Nibble. Store the result on next memory location.



```
//QUESTION 2_6
MVI A,D4H//STORE 'D4H' INTO ACCUMULATOR
OUT 30H//STORE CONTENT OF ACCUMULATOR INTO MEMORY
LOCATION '30H'
IN 30H//TAKE INPUT VALUE FROM LOCATION '30H' AND STORE
INTO ACCUMULATOR
XRI F0H//PERFORM 'XOR' OPERATION WITH ACCUMULATOR ITS
DONE "1's" COMPLEMENT OF UPPER NIBBLE
OUT 31H//STORE THE RESULT IN MEMORY LOCATION '31H' WHICH
IS EXIST IN ACCUMULATOR
HLT//HOLD AND WAIT
```



8085 Simulator - C:\Users\Hp\OneDrive\Desktop\SEM_05\Microcontroller & Applications\P2.asm

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help
Editor Assembler
Registers Memory Devices

Assembler

*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
√	0000		MVI A,D4	3E	2	2	7
	0001			D4			
√	0002		OUT 30	D3	2	3	10
	0003			30			
√	0004		IN 30	DB	2	3	10
	0005			30			
√	0006		XRI F0	EE	2	2	7
	0007			F0			
√	0008		OUT 31	D3	2	3	10
	0009			31			
√	000A		HLT	76	1	2	5

HOLD AND WAIT

Simulate

Start From → 0000

Run all At a Time
Step By Step

Created by : Jubin Mitra

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	24	0	0	1	0	0	1	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)	3E	0	0	1	1	1	1	1	0

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

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	2404
Program Counter(PC)	000A
Clock Cycle Counter	54
Instruction Counter	7

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

For SIM instruction

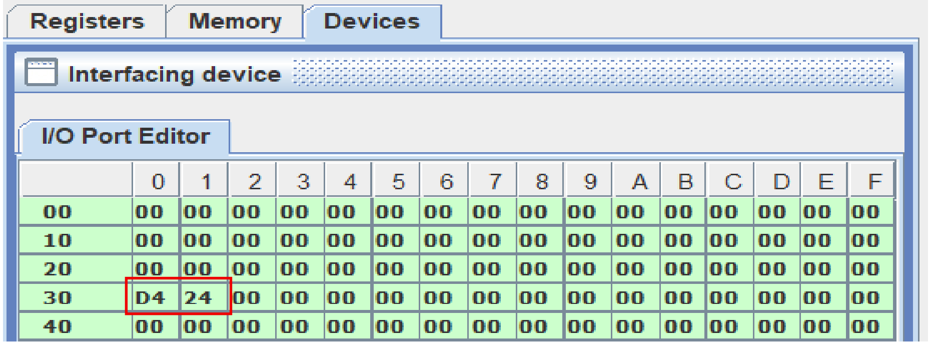
SOD	SDE	*	R7.5	MSE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

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	0



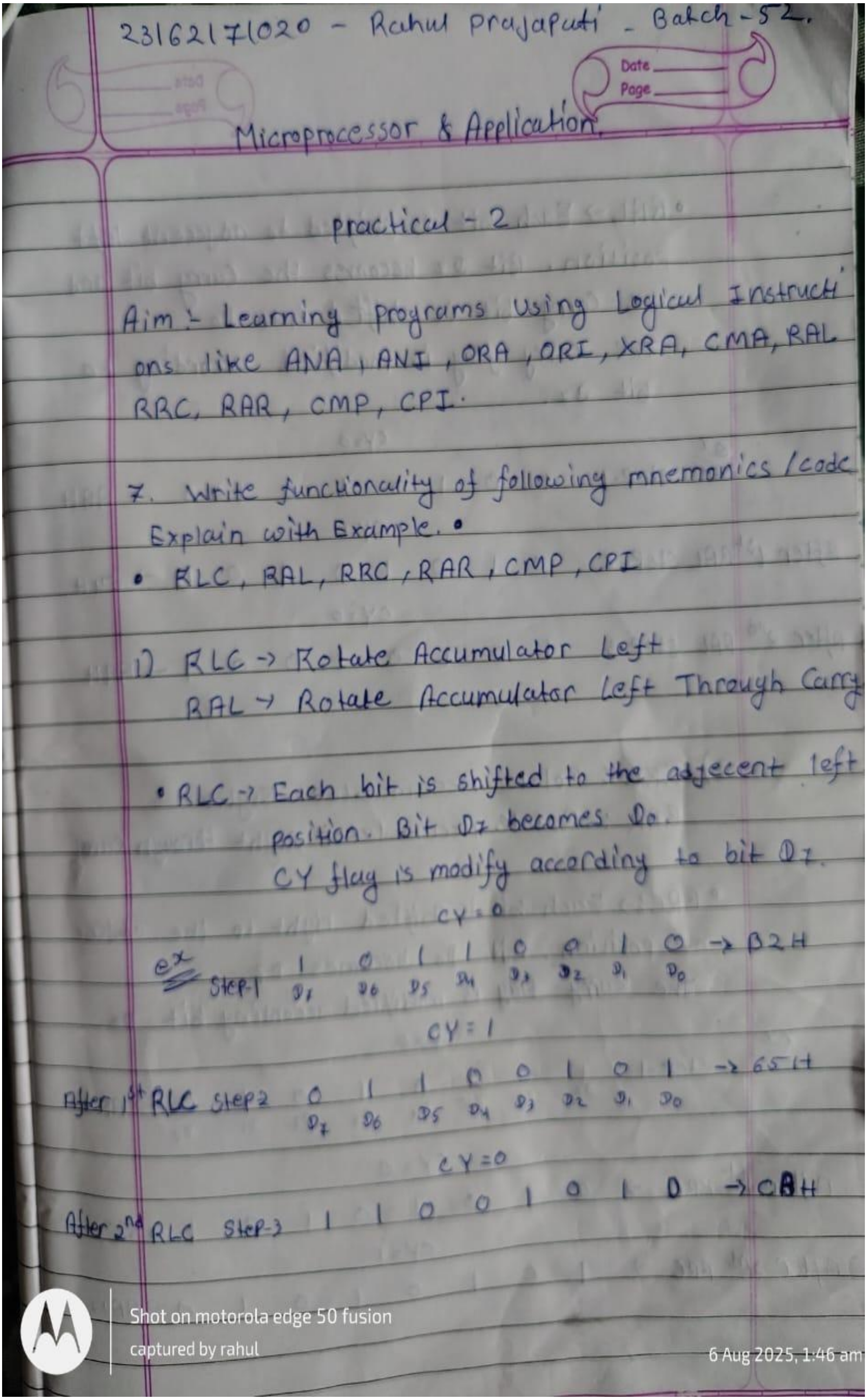
Interfacing device

I/O Port Editor

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	D4	24	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

7. Write functionality of following mnemonics/code. Explain with example

- RLC:-Rotate Accumulator Left
- RAL:- Rotate Accumulator Left Through Carry
- RRC:- Rotate Accumulator Right
- RAR:- Rotate Accumulator Right Through Carry
- CMP R:- Compare Register Data With Accumulator
- CPI [8 bit data]:- Compare 8-Bit Data With Accumulator



23162171020 - Rahul prajapati - Batch - 52.

Date ______
Page ______

Microprocessor & Application.

practical - 2

Aim :- Learning programs using Logical Instructions like ANA, ANI, ORA, ORI, XRA, CMA, RAL RRC, RAR, CMP, CPI.

7. Write functionality of following mnemonics / code Explain with Example. •

• RLC, RAL, RRC, RAR, CMP, CPI

1) RLC -> Rotate Accumulator Left
RAL -> Rotate Accumulator Left Through Carry

• RLC -> Each bit is shifted to the adjecent left position. Bit D_7 becomes D_0.
CY flag is modify according to bit D_7.

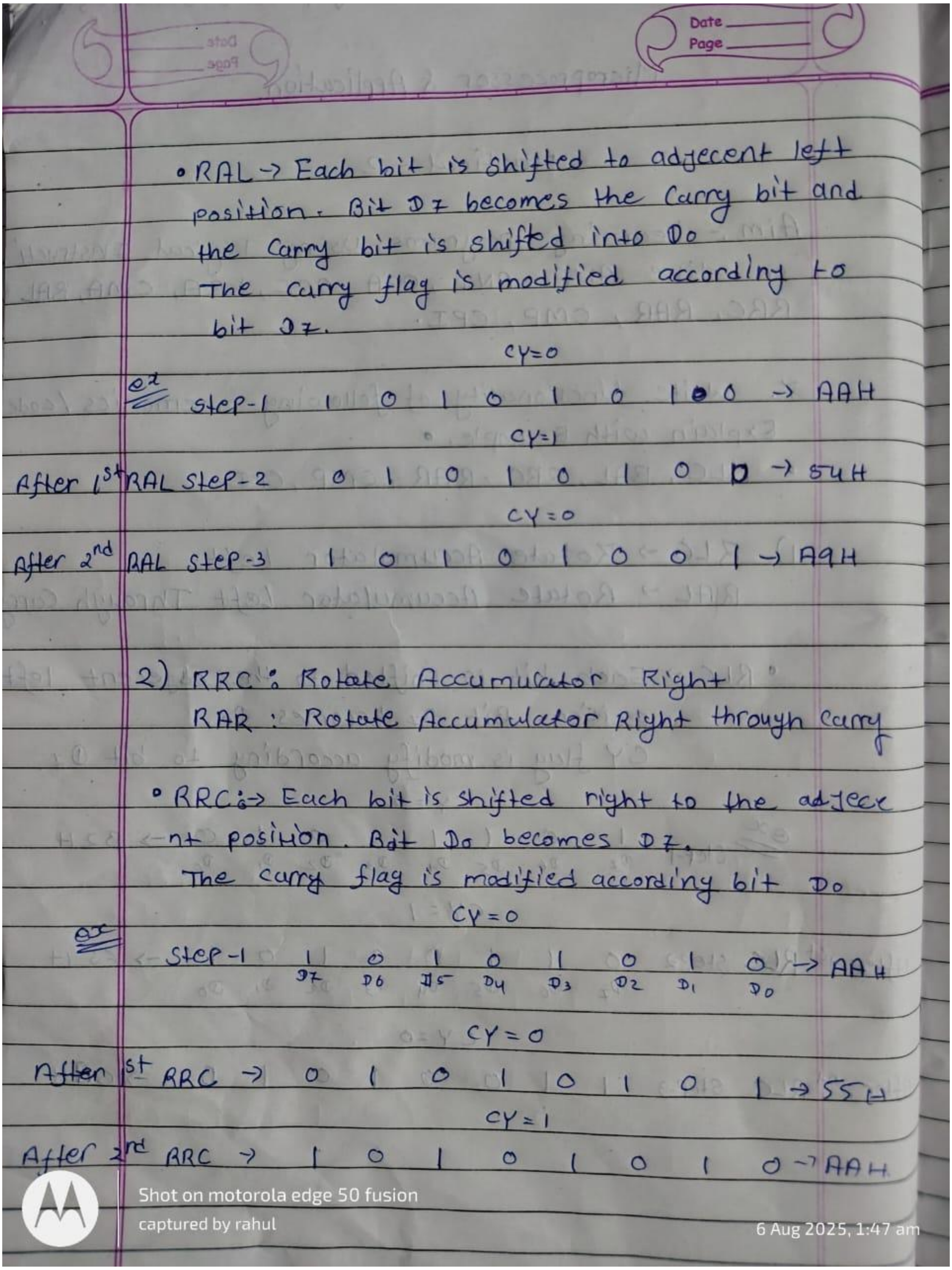
ex

	CY = 0								
Step-1	1	0	1	1	0	0	1	0	-> B2H
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	

	CY = 1								
After 1st RLC step2	0	1	1	0	0	1	0	1	-> 65H
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	

	CY = 0								
After 2nd RLC Step-3	1	1	0	0	1	0	1	0	-> CBH

Shot on motorola edge 50 fusion
captured by rahul
6 Aug 2025, 1:46 am



- RAL → Each bit is shifted to adjecent left position. Bit D7 becomes the Carry bit and the Carry bit is shifted into D0
  The carry flag is modified according to bit D7.

CY=0

ex Step-1 1 0 1 0 1 0 1 0 → AAH

CY=1

After 1st RAL Step-2 0 1 0 1 0 1 0 0 → 54H

CY=0

After 2nd RAL Step-3 1 0 1 0 1 0 0 1 → A9H

2) RRC : Rotate Accumulator Right
RAR : Rotate Accumulator Right through carry

- RRC:→ Each bit is shifted right to the adjecent position. Bit D0 becomes D7.
  The carry flag is modified according bit D0

CY=0

ex Step-1 1 0 1 0 1 0 1 0 → AAH

D7 D6 D5 D4 D3 D2 D1 D0

CY=0

After 1st RRC → 0 1 0 1 0 1 0 1 → 55H

CY=1

After 2nd ARC → 1 0 1 0 1 0 1 0 → AAH

Shot on motorola edge 50 fusion
captured by rahul
6 Aug 2025, 1:47 am

• RAR → Each bit is shifted right to the adjecen position. Bit D_0 becomes the carry bit, and the carry bit shifted to into D_7.

	CY = 0								
eg: Step-1	1	0	1	1	1	0	0	1	→ B9H
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
	CY = 1								
After 1st RAR →	0	1	0	1	1	1	0	0	→ 5CH
	CY = 0								
After 2nd RAR →	1	0	1	0	1	1	1	0	→ ADH.

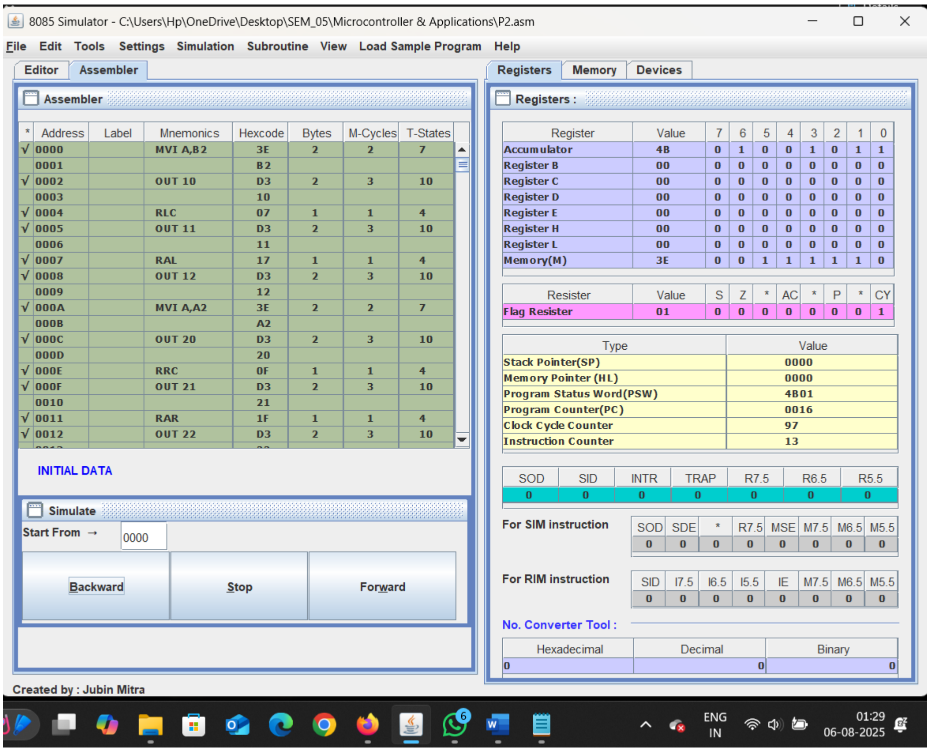
★ CMP :- Compare the content of register with the content of A for less then / equal to or greater than.

★. CPI 8bit : Compare 8 bit data with the contents of A for less than, equal to, or grecter than.

→ How to Decide condition?

Compare (A − R)	Z	CY	meaning.
A = R	Z = 1	CY = 0	Equal
A < R	Z = 0	CY = 1	A is less than R
A > R	Z = 0	CY = 0	A is grecter than R.

Shot on motorola edge 50 fusion
captured by rahul
6 Aug 2025, 1:47 am



8085 Simulator - C:\Users\Hp\OneDrive\Desktop\SEM_05\Microcontroller & Applications\P2.asm

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help
Editor Assembler
Assembler

*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
√	0000		MVI A,B2	3E	2	2	7
	0001			B2			
√	0002		OUT 10	D3	2	3	10
	0003			10			
√	0004		RLC	07	1	1	4
√	0005		OUT 11	D3	2	3	10
	0006			11			
√	0007		RAL	17	1	1	4
√	0008		OUT 12	D3	2	3	10
	0009			12			
√	000A		MVI A,A2	3E	2	2	7
	000B			A2			
√	000C		OUT 20	D3	2	3	10
	000D			20			
√	000E		RRC	0F	1	1	4
√	000F		OUT 21	D3	2	3	10
	0010			21			
√	0011		RAR	1F	1	1	4
√	0012		OUT 22	D3	2	3	10

INITIAL DATA

Start From → 0000

Register	Value	7	6	5	4	3	2	1	0
Accumulator	4B	0	1	0	0	1	0	1	1
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)	3E	0	0	1	1	1	1	1	0

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

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	4B01
Program Counter(PC)	0016
Clock Cycle Counter	97
Instruction Counter	13

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

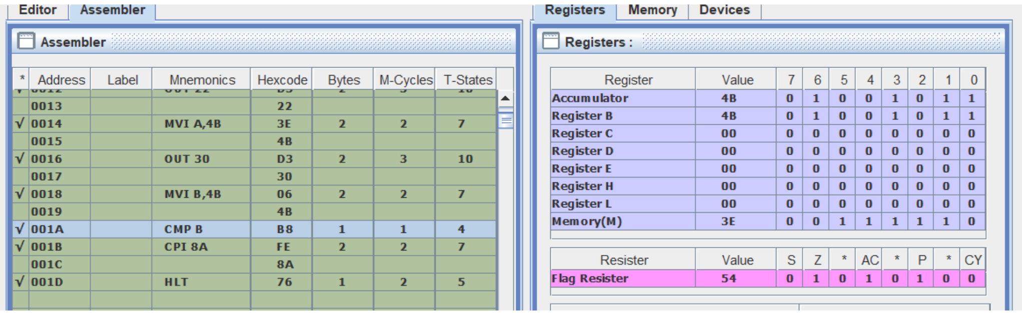
SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

No. Converter Tool :

Hexadecimal	Decimal	Binary
0	0	0

Created by : Jubin Mitra

- CMP :- Comparision Between Value Of Register ‘A’ And ‘B’ .
  Therefore, The Value Of FLAG “Z=1” And “CY=0” So The Both Value Are Equal.

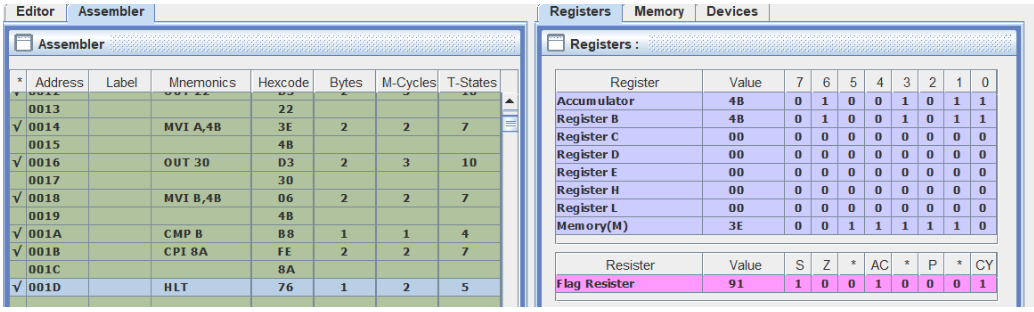


*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
	0013			22			
√	0014		MVI A,4B	3E	2	2	7
	0015			4B			
√	0016		OUT 30	D3	2	3	10
	0017			30			
√	0018		MVI B,4B	06	2	2	7
	0019			4B			
√	001A		CMP B	B8	1	1	4
√	001B		CPI 8A	FE	2	2	7
	001C			8A			
√	001D		HLT	76	1	2	5

Register	Value	7	6	5	4	3	2	1	0
Accumulator	4B	0	1	0	0	1	0	1	1
Register B	4B	0	1	0	0	1	0	1	1
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)	3E	0	0	1	1	1	1	1	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	54	0	1	0	1	0	1	0	0

- CPI :- Comparision Between Data And Register ‘A’ .
- Therefore, The Value Of FLAG “Z=0” And “CY=1” So The Value Of Accumulator Is Less Then Data.



*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
	0013			22			
√	0014		MVI A,4B	3E	2	2	7
	0015			4B			
√	0016		OUT 30	D3	2	3	10
	0017			30			
√	0018		MVI B,4B	06	2	2	7
	0019			4B			
√	001A		CMP B	B8	1	1	4
√	001B		CPI 8A	FE	2	2	7
	001C			8A			
√	001D		HLT	76	1	2	5

Register	Value	7	6	5	4	3	2	1	0
Accumulator	4B	0	1	0	0	1	0	1	1
Register B	4B	0	1	0	0	1	0	1	1
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)	3E	0	0	1	1	1	1	1	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	91	1	0	0	1	0	0	0	1