KARIYA RAJ 21162101011

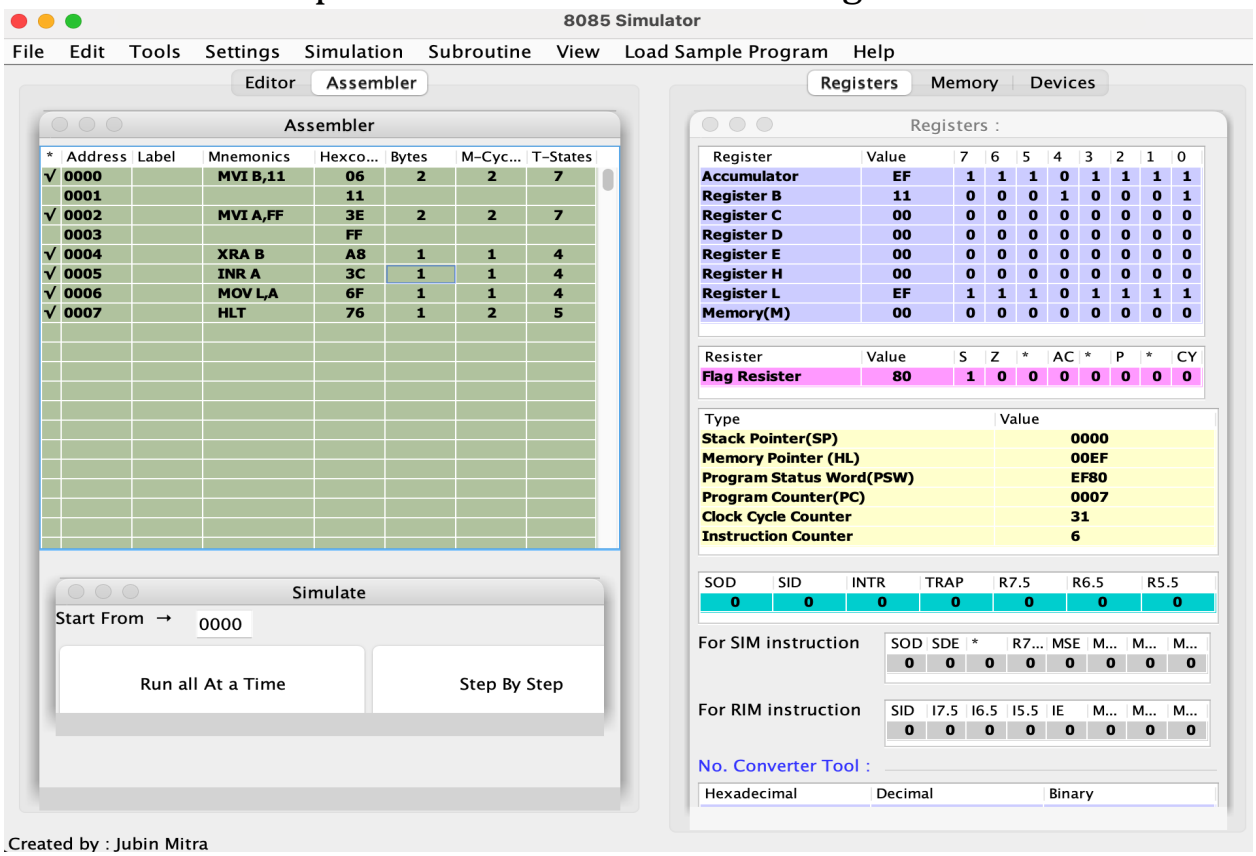
M&A

PRACTICAL -2

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

Exercise : (Solution must be handwritten in book)

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.



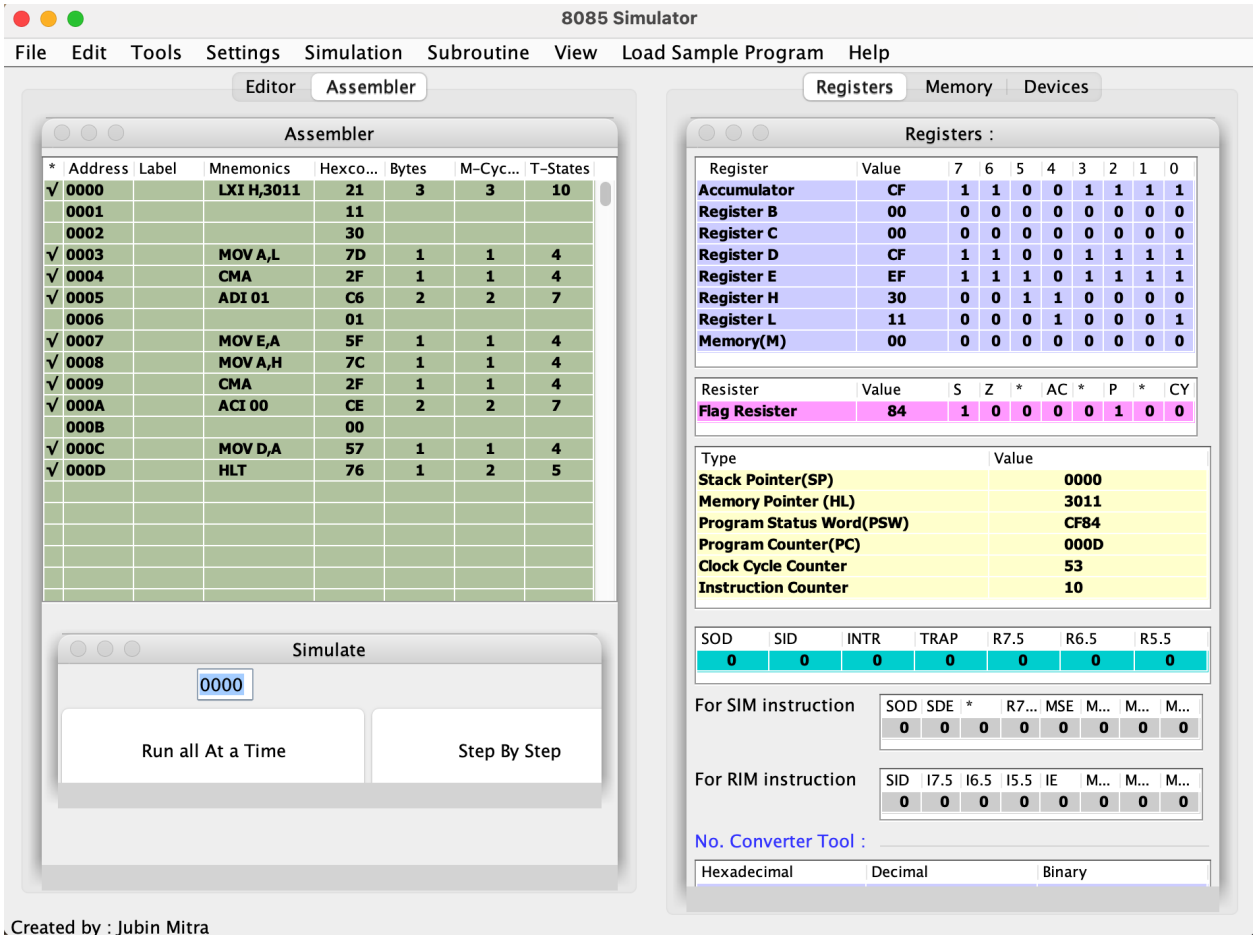
KARIYA RAJ 21162101011

A: Code:

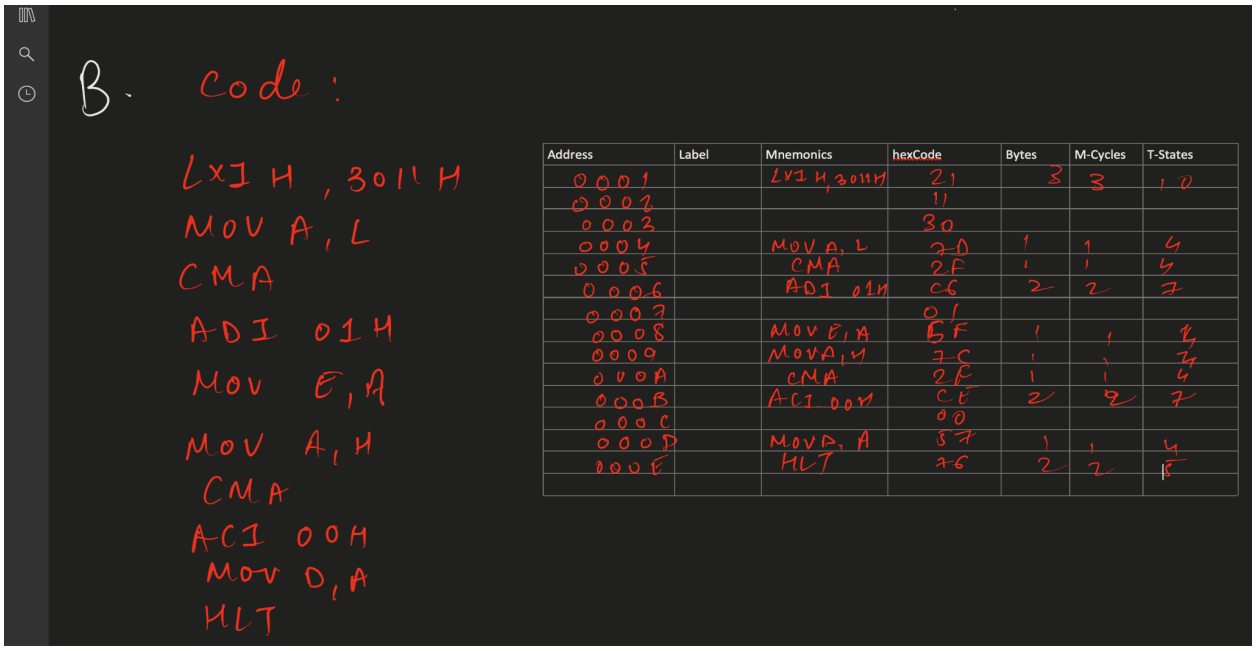
```
MVI B, 11H
MVI A, FFH
XRA B
INR A
MOV L, A
HLT
```

Address	Label	Mnemonics	hexCode	Bytes	M-Cycles	T-States
0000		MVI B, 11	06	2	2	7
0001			11			
0002		MVI A, FF	3E	2	2	7
0003			FF			
0004		XRA B	A8	1	1	4
0005		INR A	3C	1	1	4
0006		MOV L, A	6F	1	1	4
0007		HLT	76	1	2	5

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.

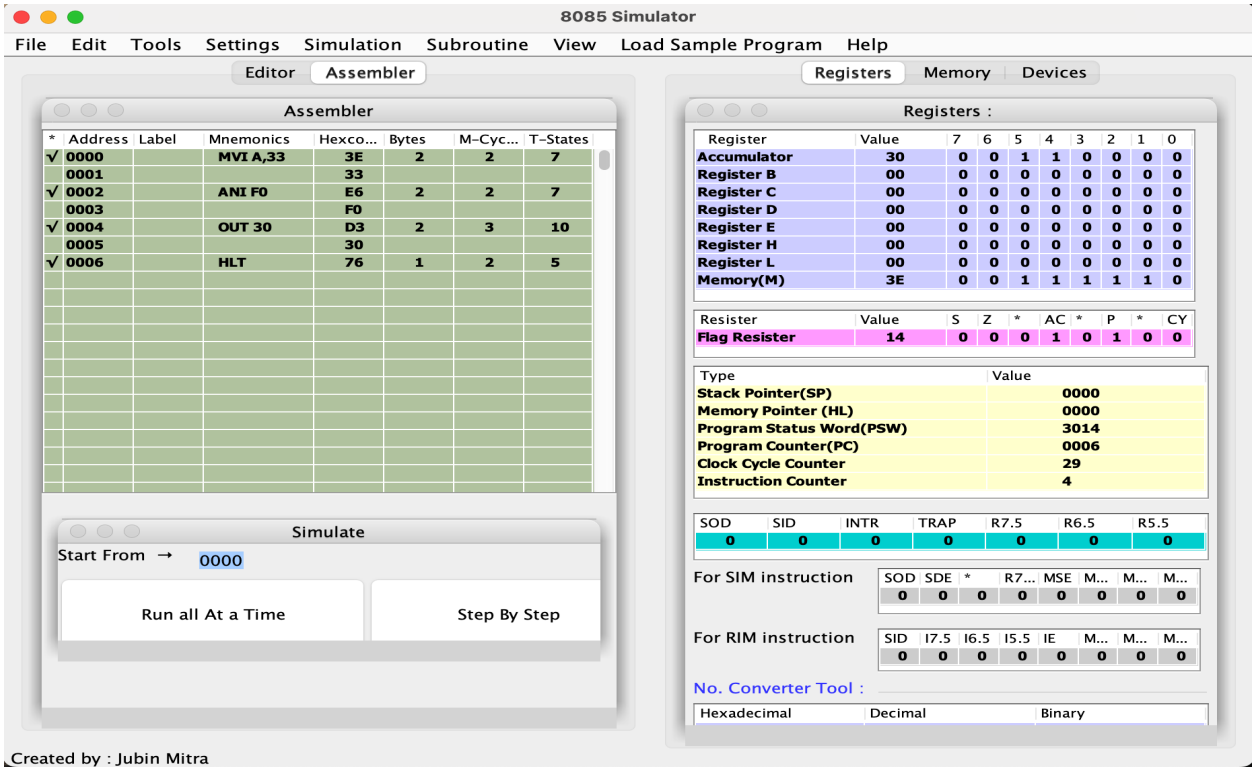


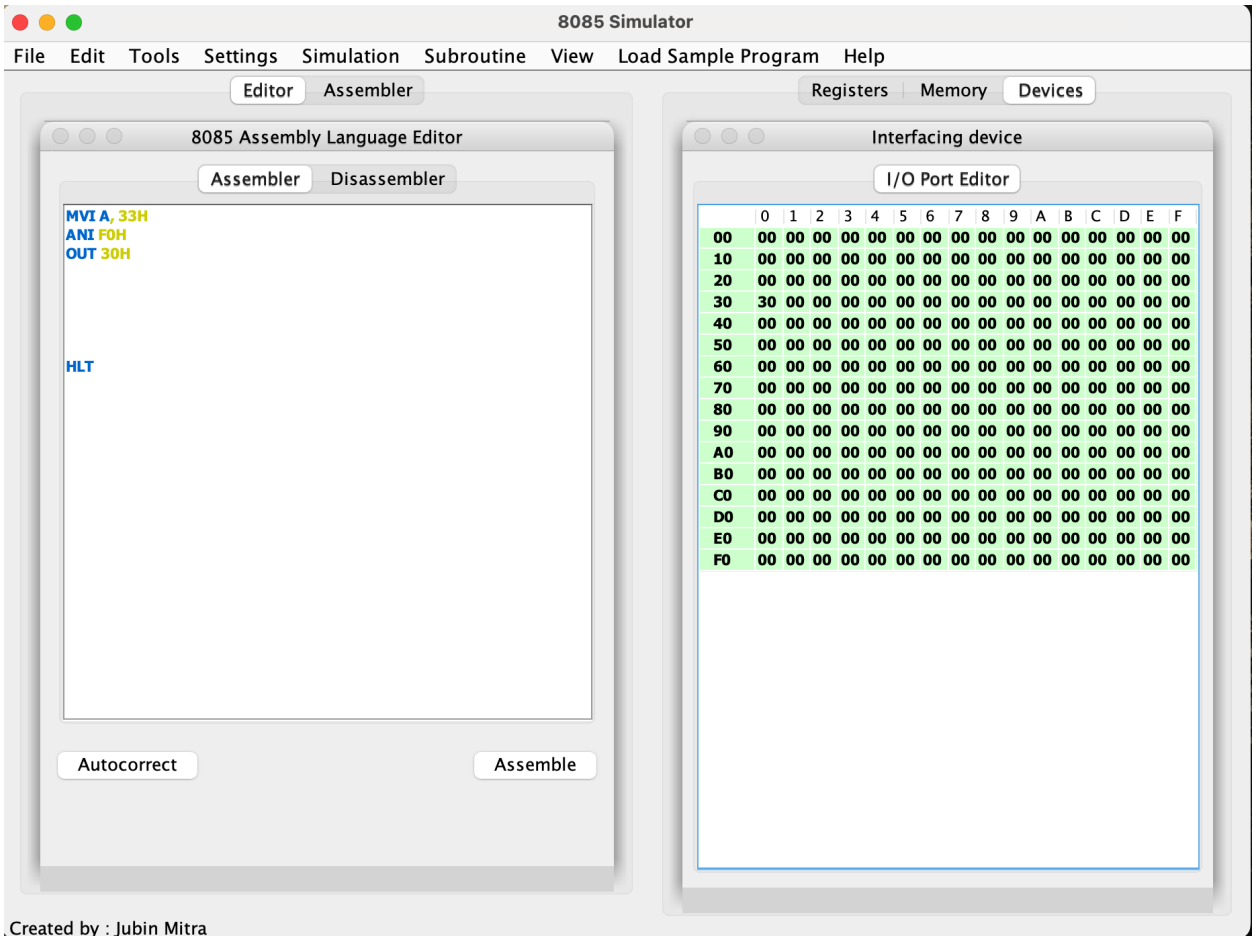
KARIYA RAJ 21162101011



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:

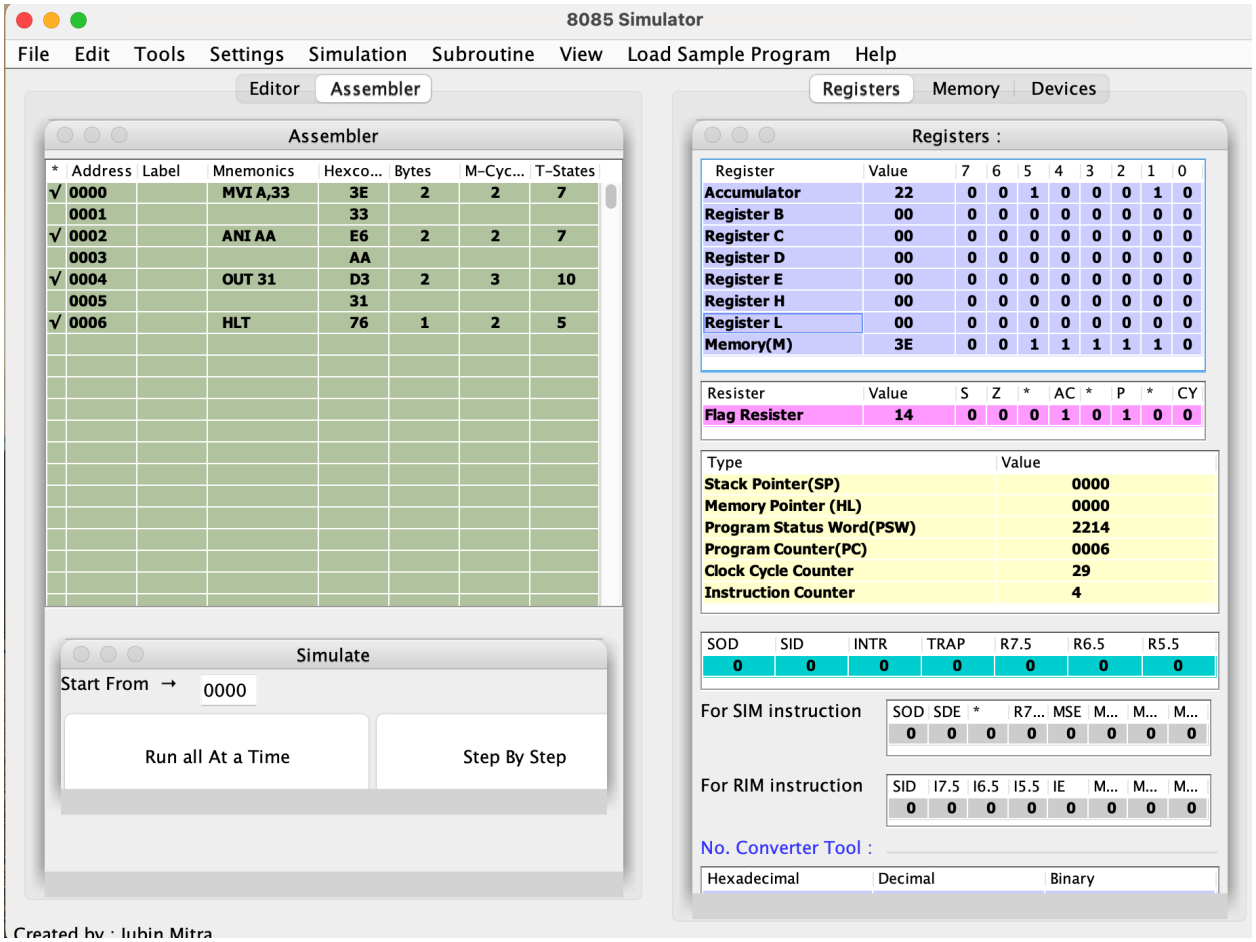
Case 1: Lower nibble should be masked and upper nibble should remain unchanged.

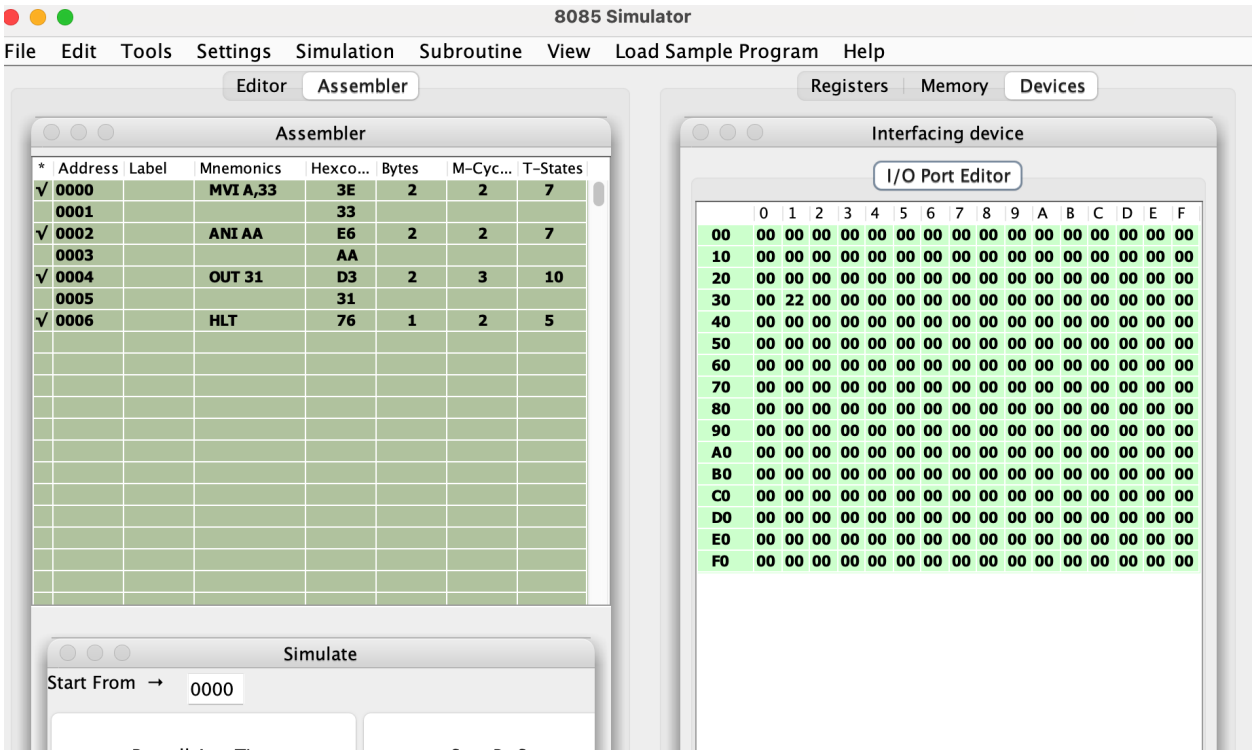




Case 2: All even bits shall be masked.

KARIYA RAJ 21162101011





KARIYA RAJ 21162101011

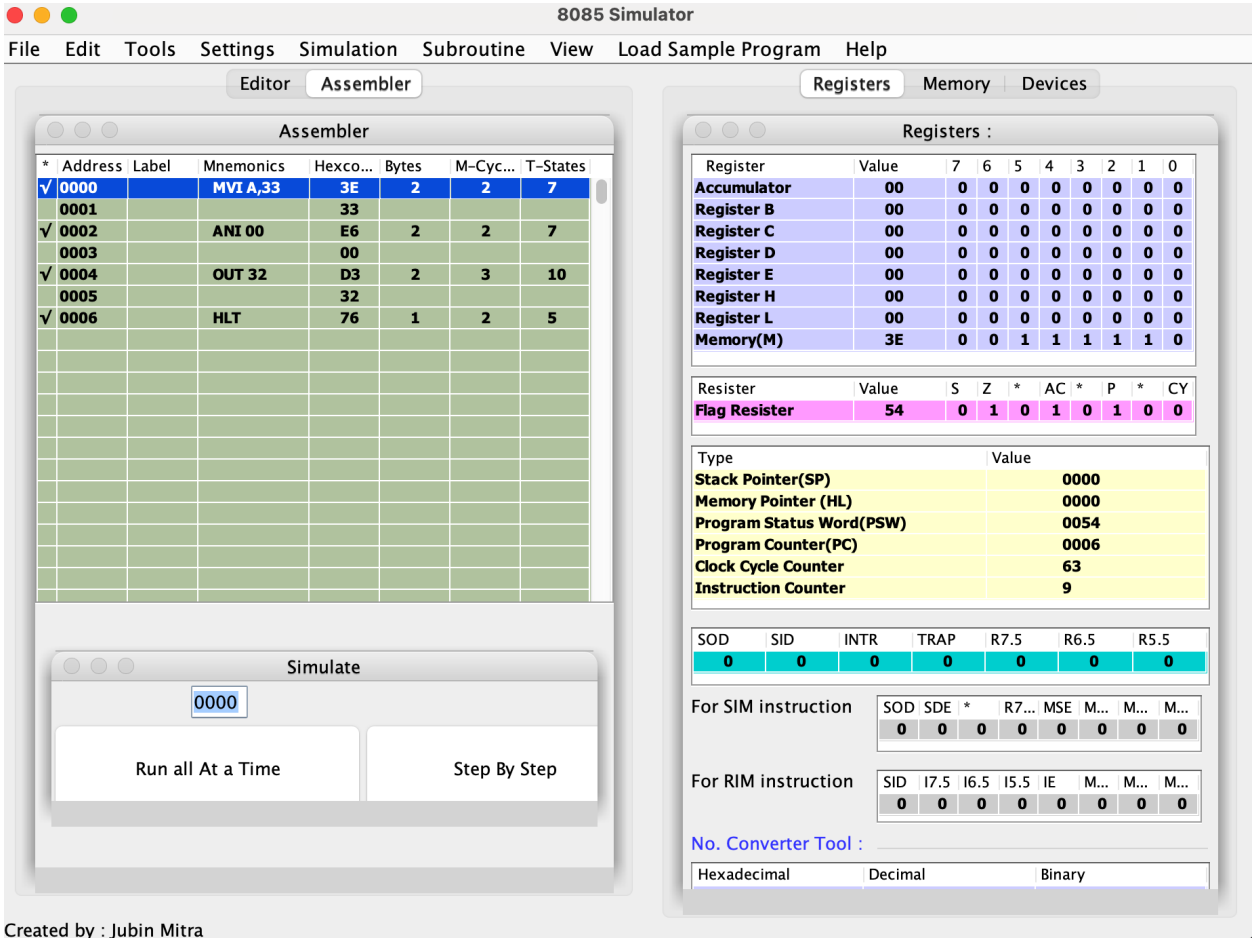
3 (case-2):

code:

```
MVI A,33H
ANI AAH
OUT 31H
HLT
```

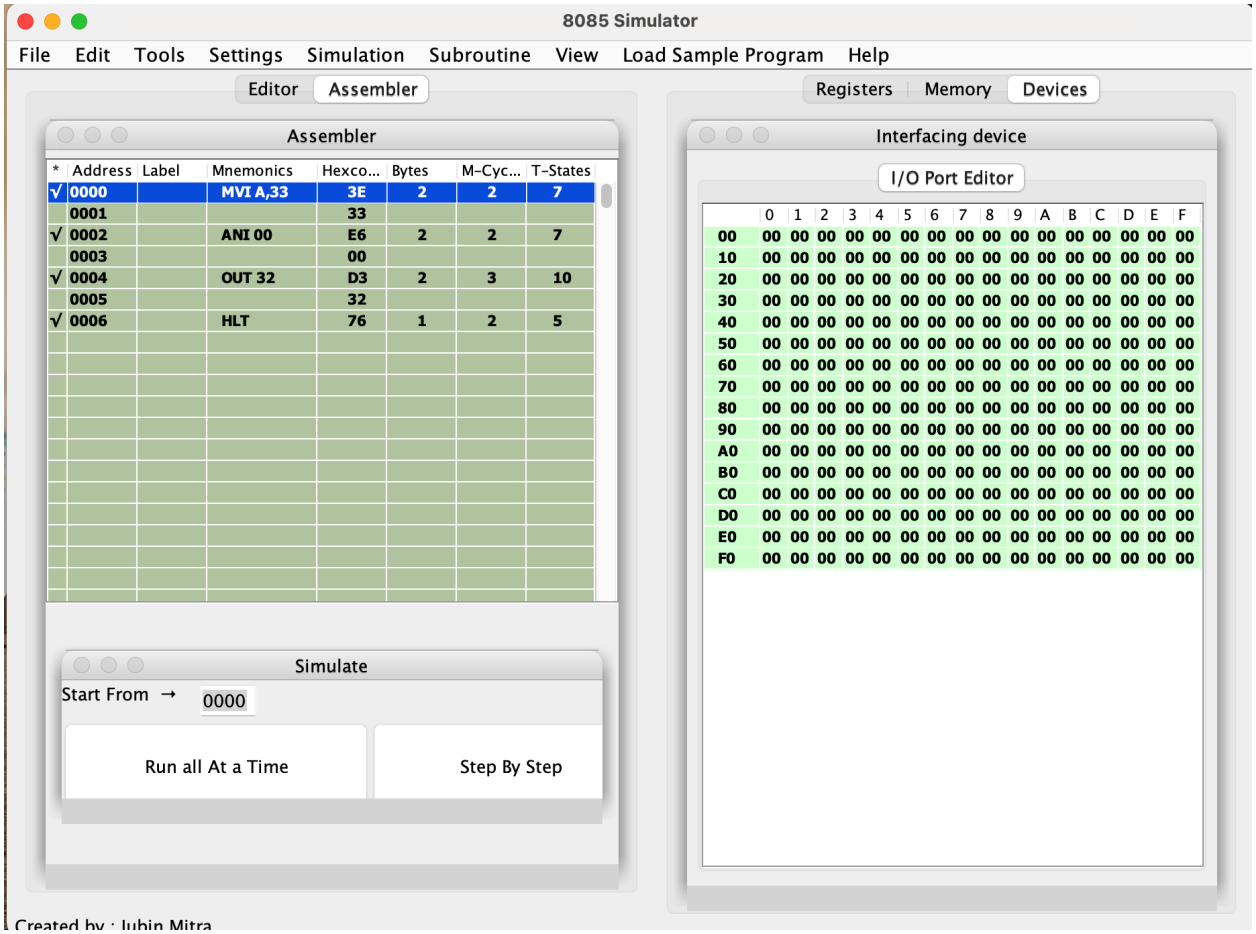
Address	Label	Mnemonics	hexCode	Bytes	M-Cycles	T-States
0000		MVI A,33H	3E	2	2	7
0001			33			
0002		ANI AAH	E6	2	2	7
0003			AA			
0004		OUT	D3	2	3	10
0005			32			
0006		HLT	76	1	2	5

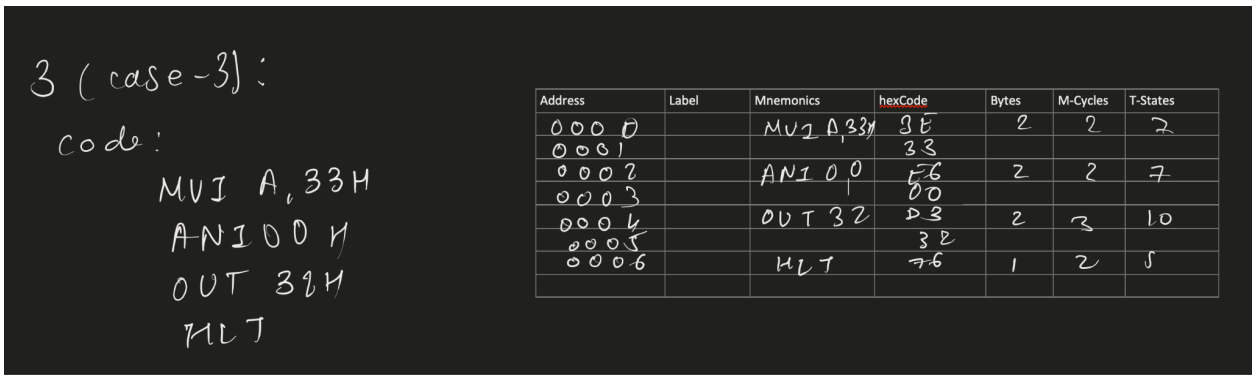
Case 3: Answer after masking becomes zero.



Created by : Jubin Mitra

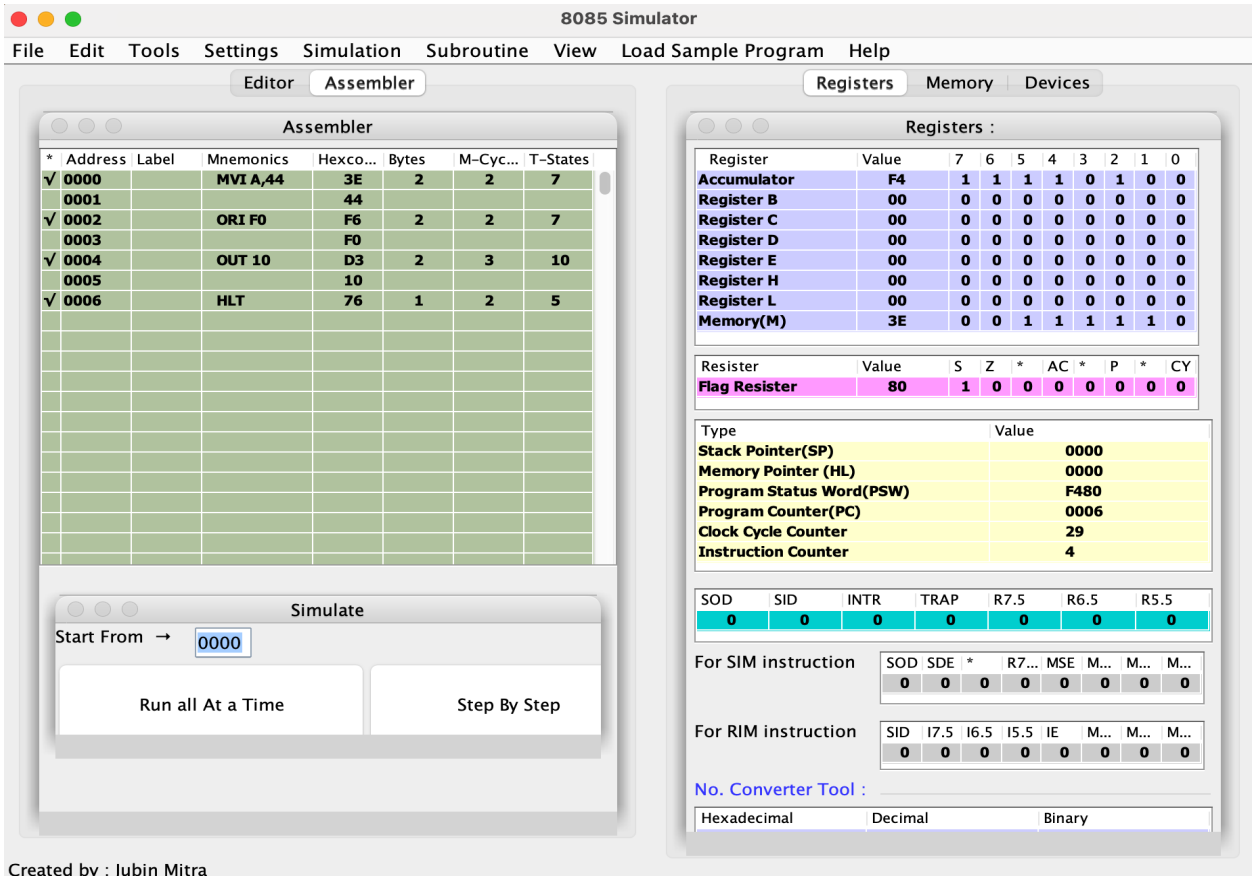
KARIYA RAJ 21162101011

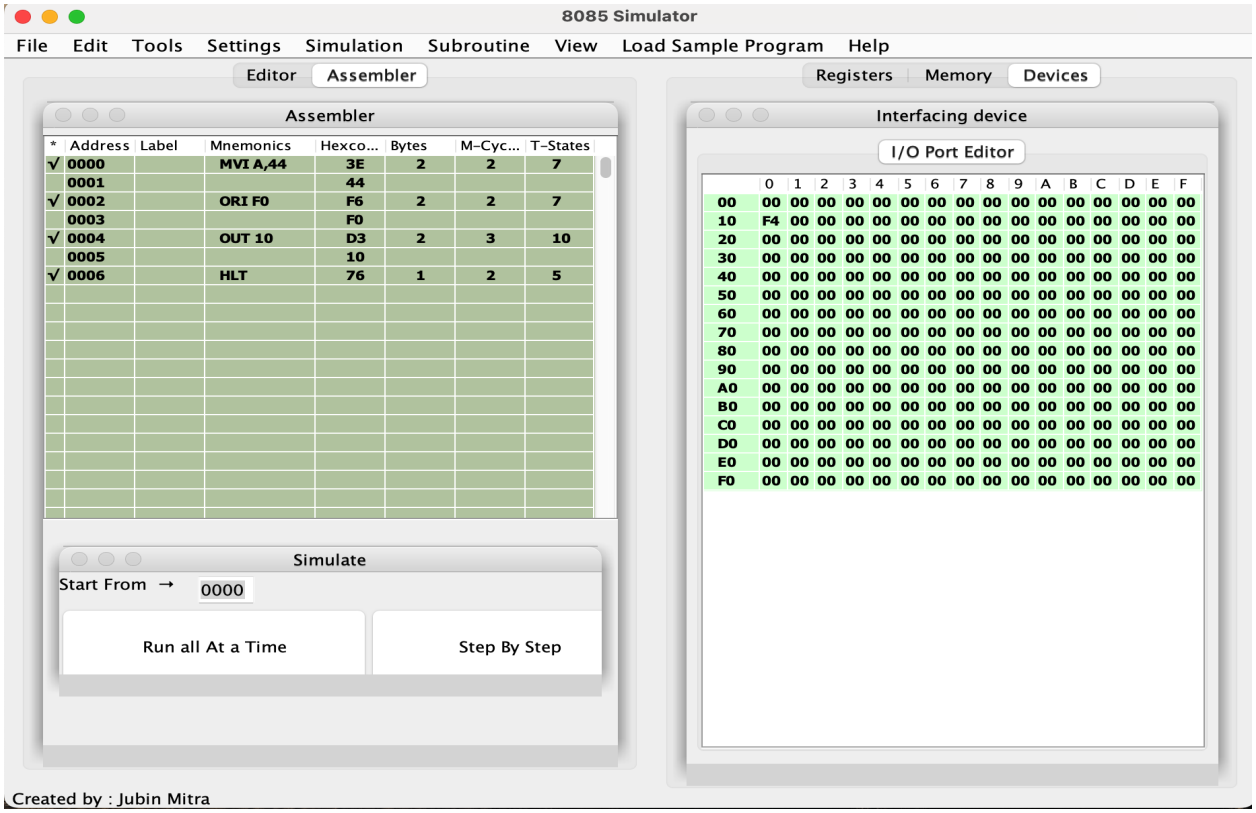




4. To learn unmasking patterns and hence making specific bits to one. Take one 8-bit that is multiplied by 4 with the last two digits of your enrollment number. Perform operations to result as follows and display on the port 10H, 11H and 12H
respectively:
Case 1: Upper nibble should be unmasked and lower nibble should remain unchanged.

KARIYA RAJ 21162101011





KARIYA RAJ 21162101011

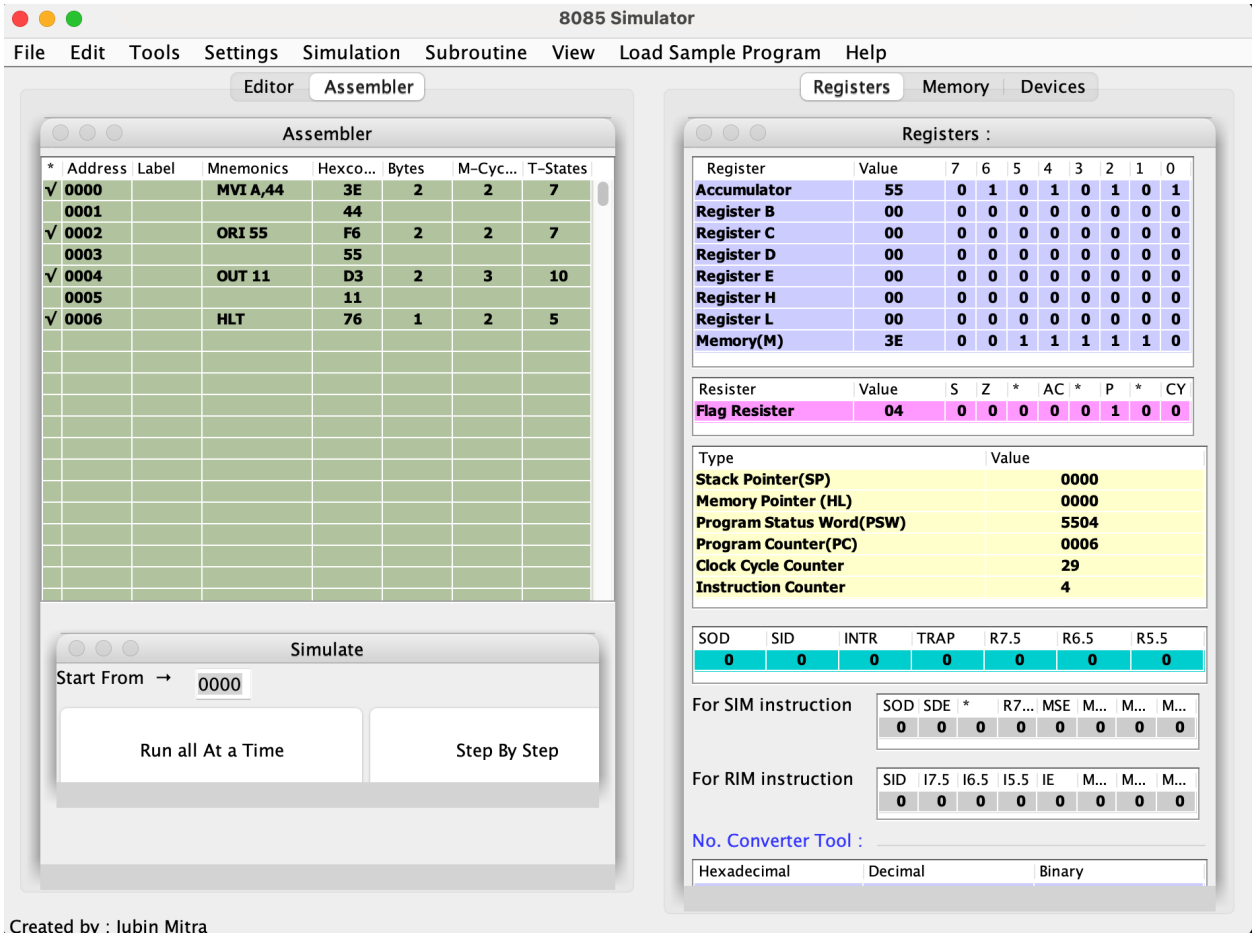
4 (case-1):

code:

```
MVI A,44H
ORI F0H
OUT 10H
HLT
```

Address	Label	Mnemonics	hexCode	Bytes	M-Cycles	T-States
0000		MVI A,44H	3E	2	2	7
0001			44			
0002		ORI F0	F6	2	2	7
0003			F0			
0004		OUT 10	D3	2	3	10
0005			10			
0006		HLT	76	1	2	5

Case 2: All odd bits shall be unmasked.



*	Address	Label	Mnemonics	Hexco...	Bytes	M-Cyc...	T-States
√	0000		MVI A,44	3E	2	2	7
	0001			44			
√	0002		ORI 55	F6	2	2	7
	0003			55			
√	0004		OUT 11	D3	2	3	10
	0005			11			
√	0006		HLT	76	1	2	5

Register	Value	7	6	5	4	3	2	1	0
Accumulator	55	0	1	0	1	0	1	0	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	04	0	0	0	0	0	1	0	0

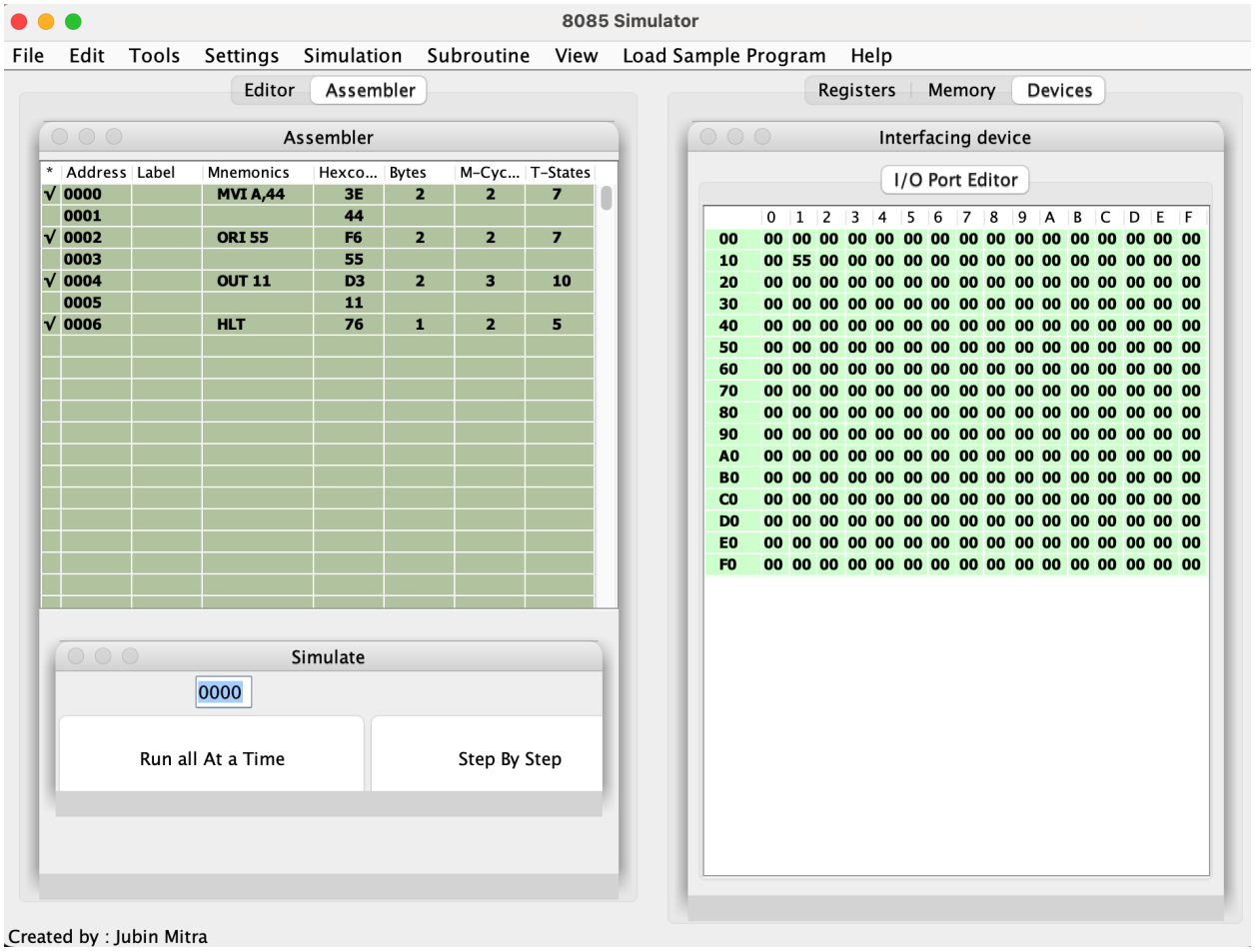
Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	5504
Program Counter(PC)	0006
Clock Cycle Counter	29
Instruction Counter	4

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

SOD	SDE	*	R7...	MSE	M...	M...	M...
0	0	0	0	0	0	0	0

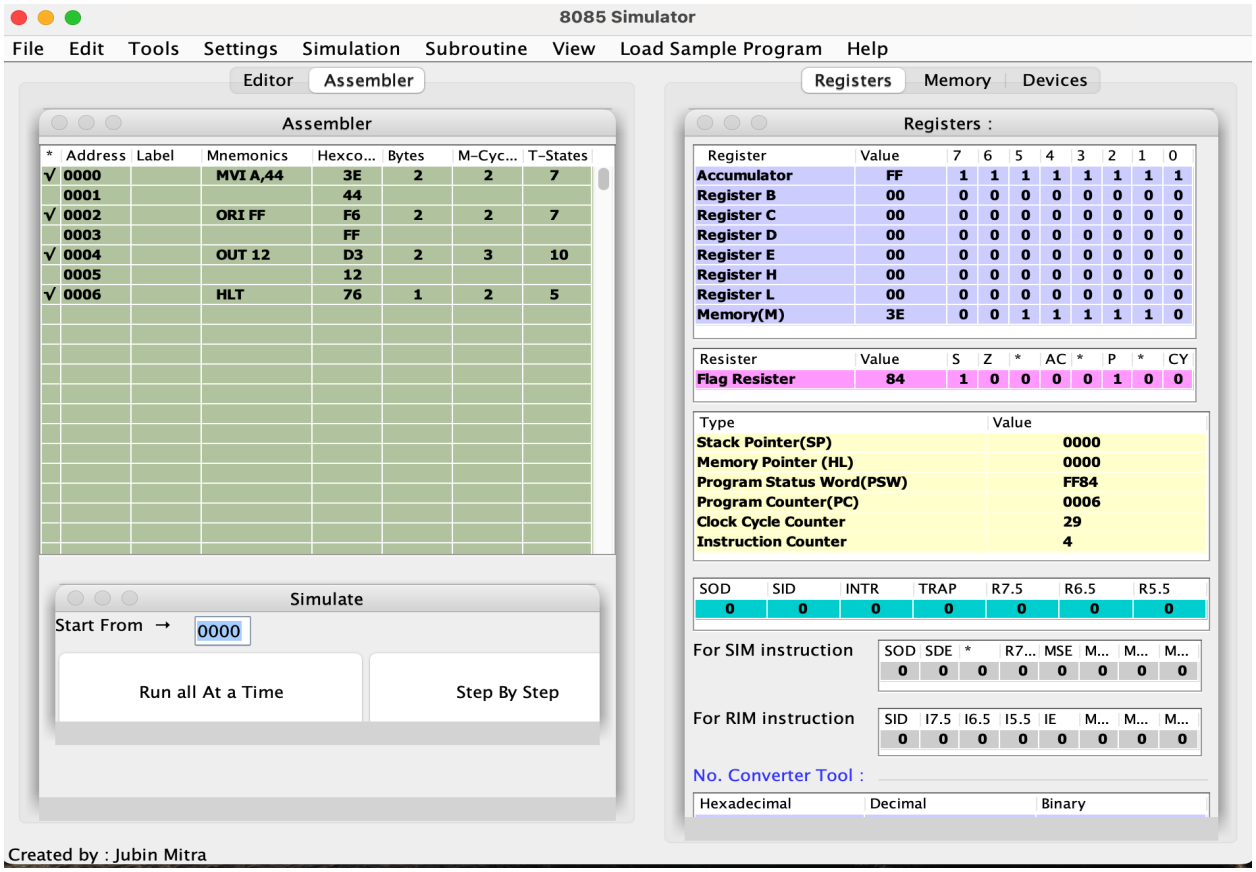
SID	I7.5	I6.5	I5.5	IE	M...	M...	M...
0	0	0	0	0	0	0	0

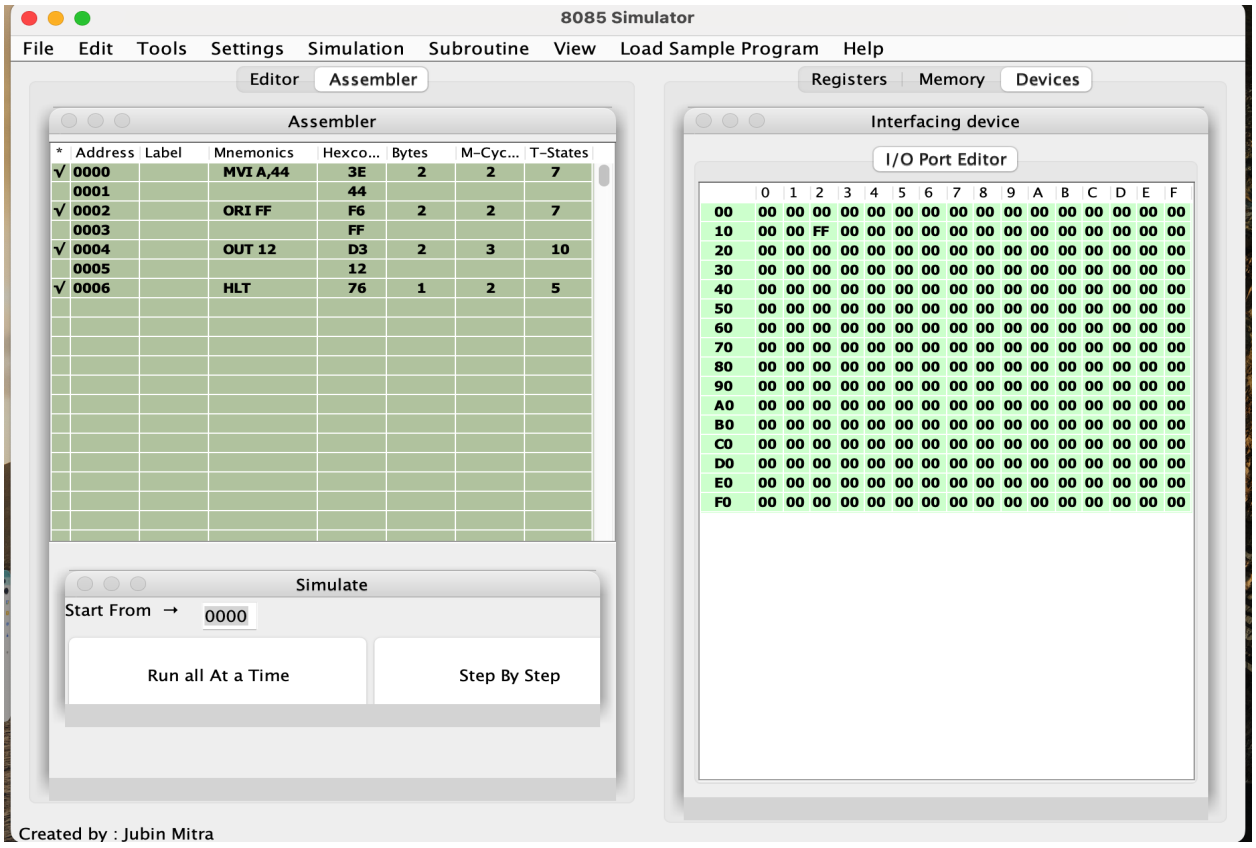
KARIYA RAJ 21162101011



Case 3: Answer after unmasking becomes all bits one.

KARIYA RAJ 21162101011





KARIYA RAJ 21162101011

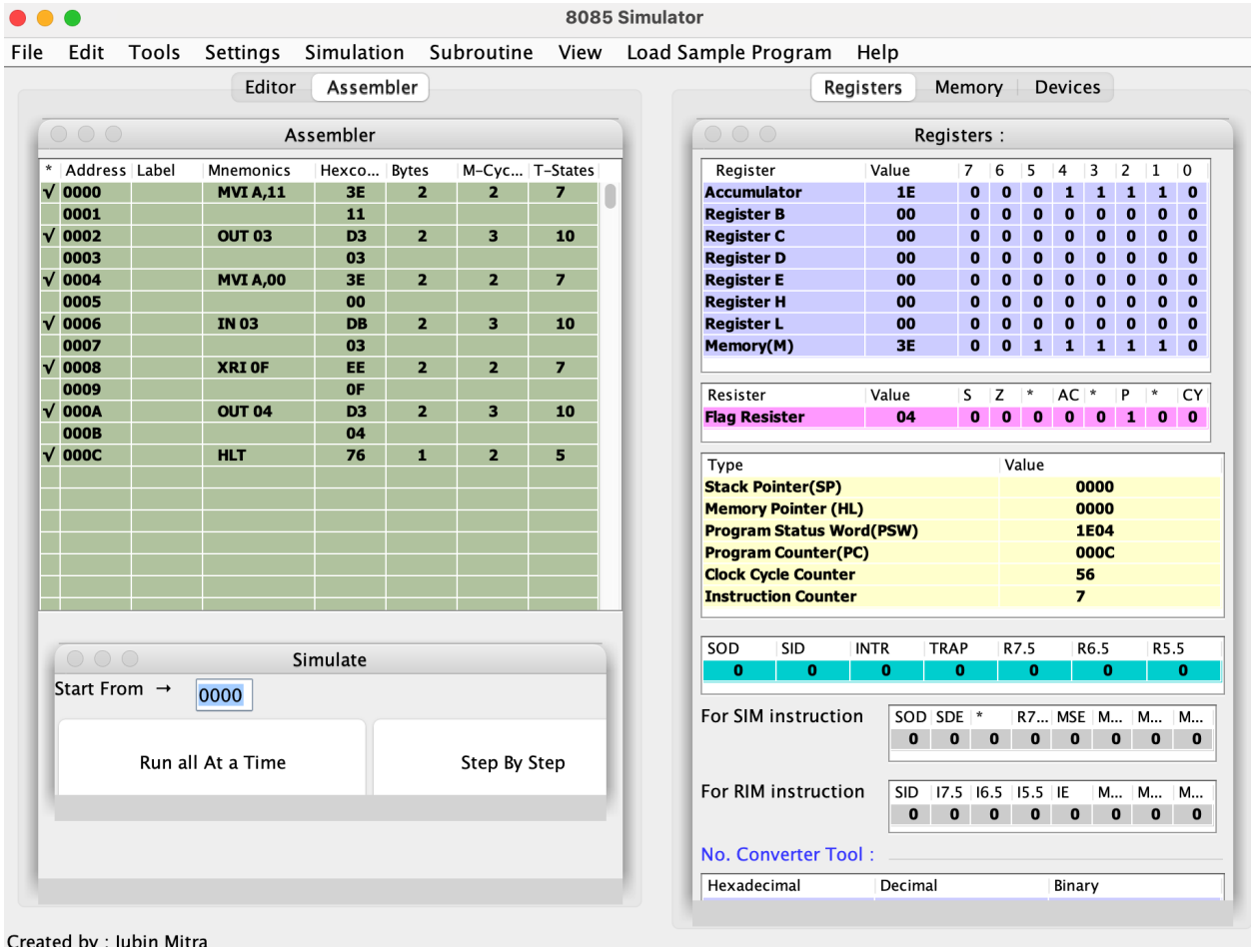
4. (case-3):

Code:

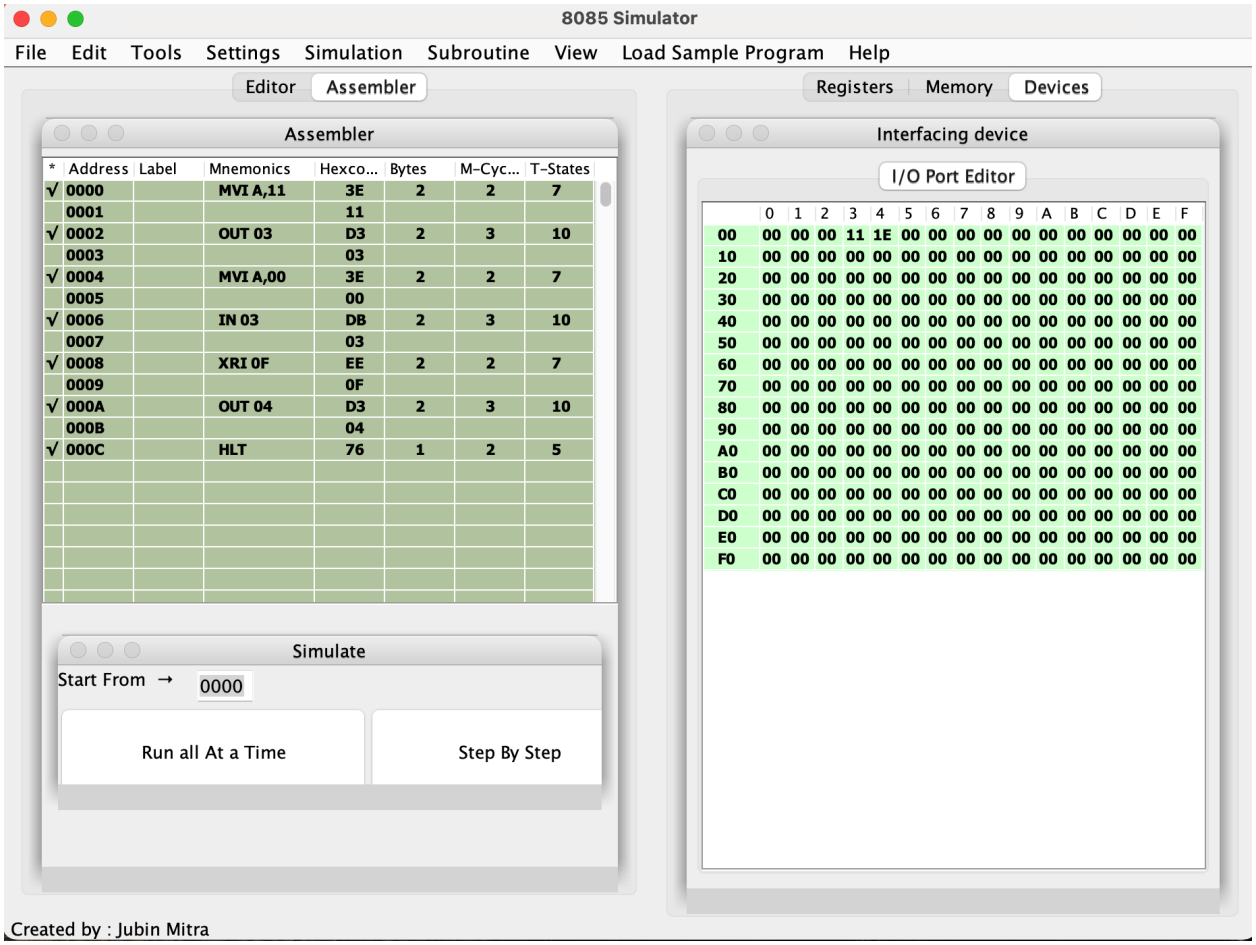
```
MVI A,44H
ORI FFH
OUT 12H
HLT
```

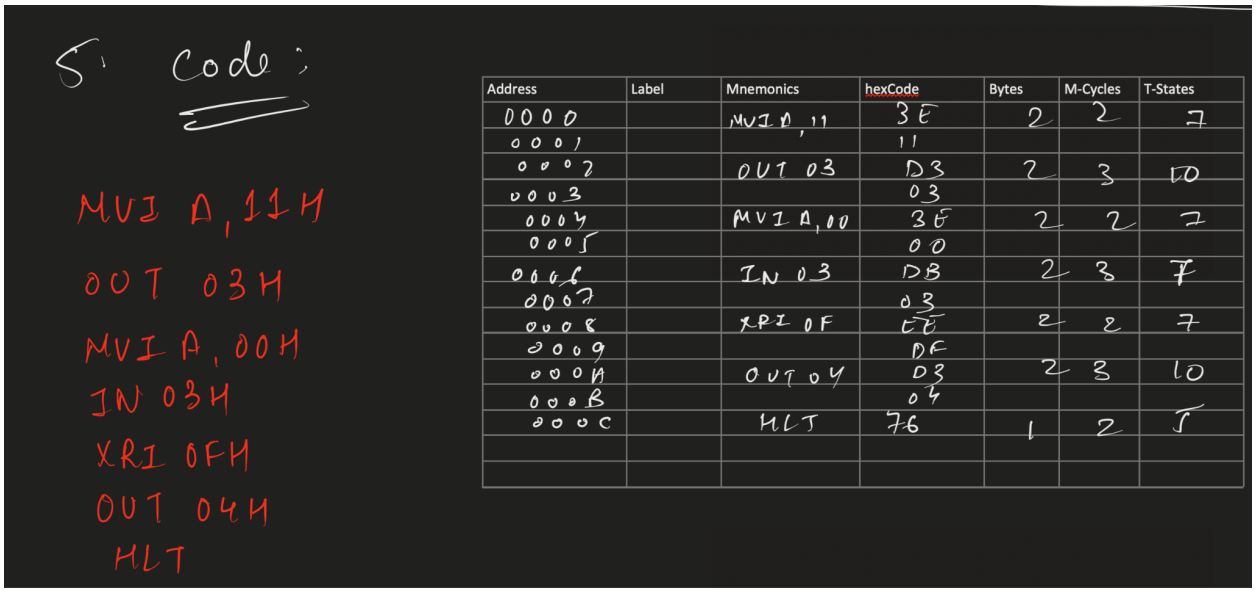
Address	Label	Mnemonics	hexCode	Bytes	M-Cycles	T-States
0000		MVI A,44	3E	2	2	7
0001			44			
0002		ORI FF	F6	2	2	7
0003			FF			
0004		OUT 12	D3	2	3	10
0005			12			
0006		HLT	76	1	2	5

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



KARIYA RAJ 21162101011

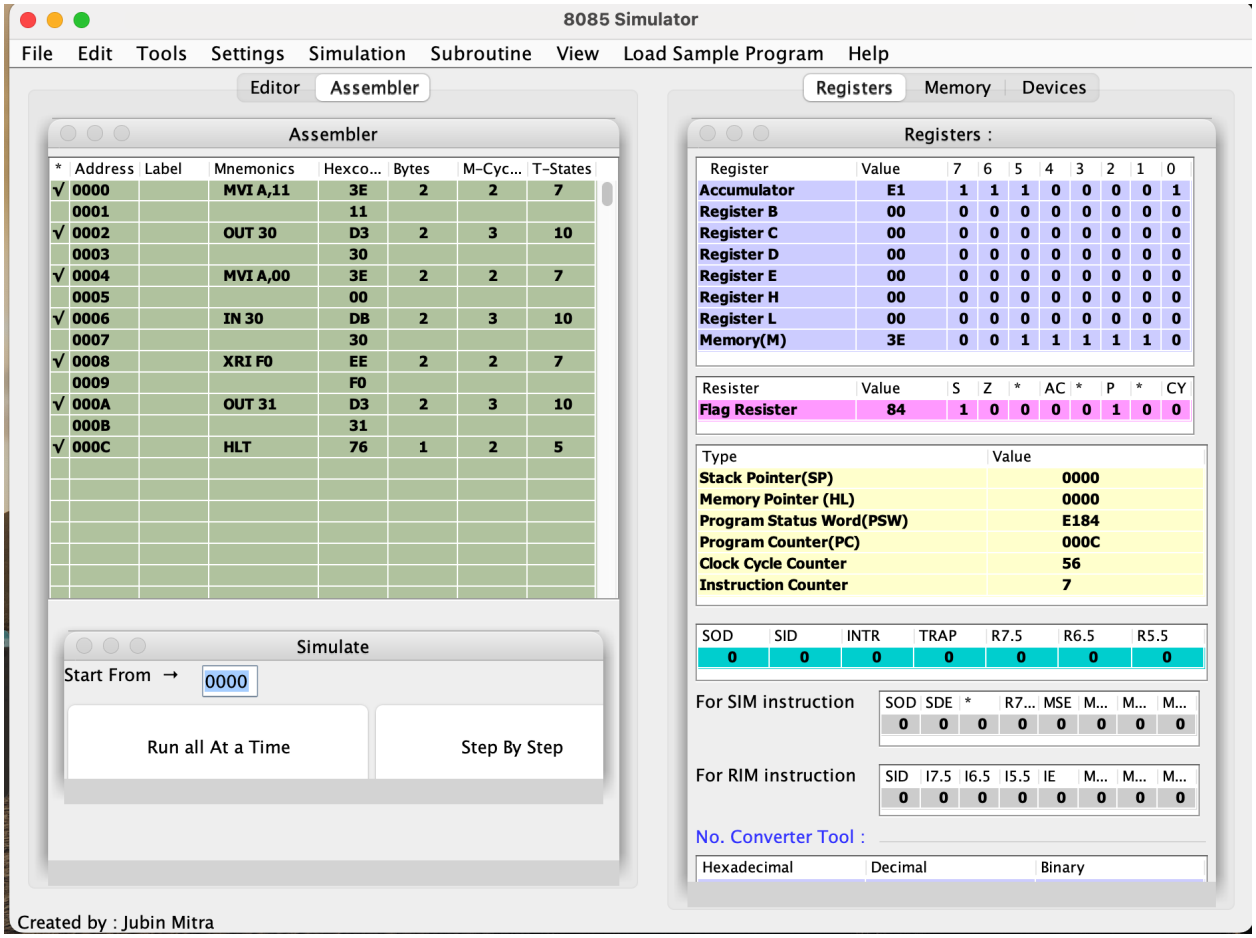




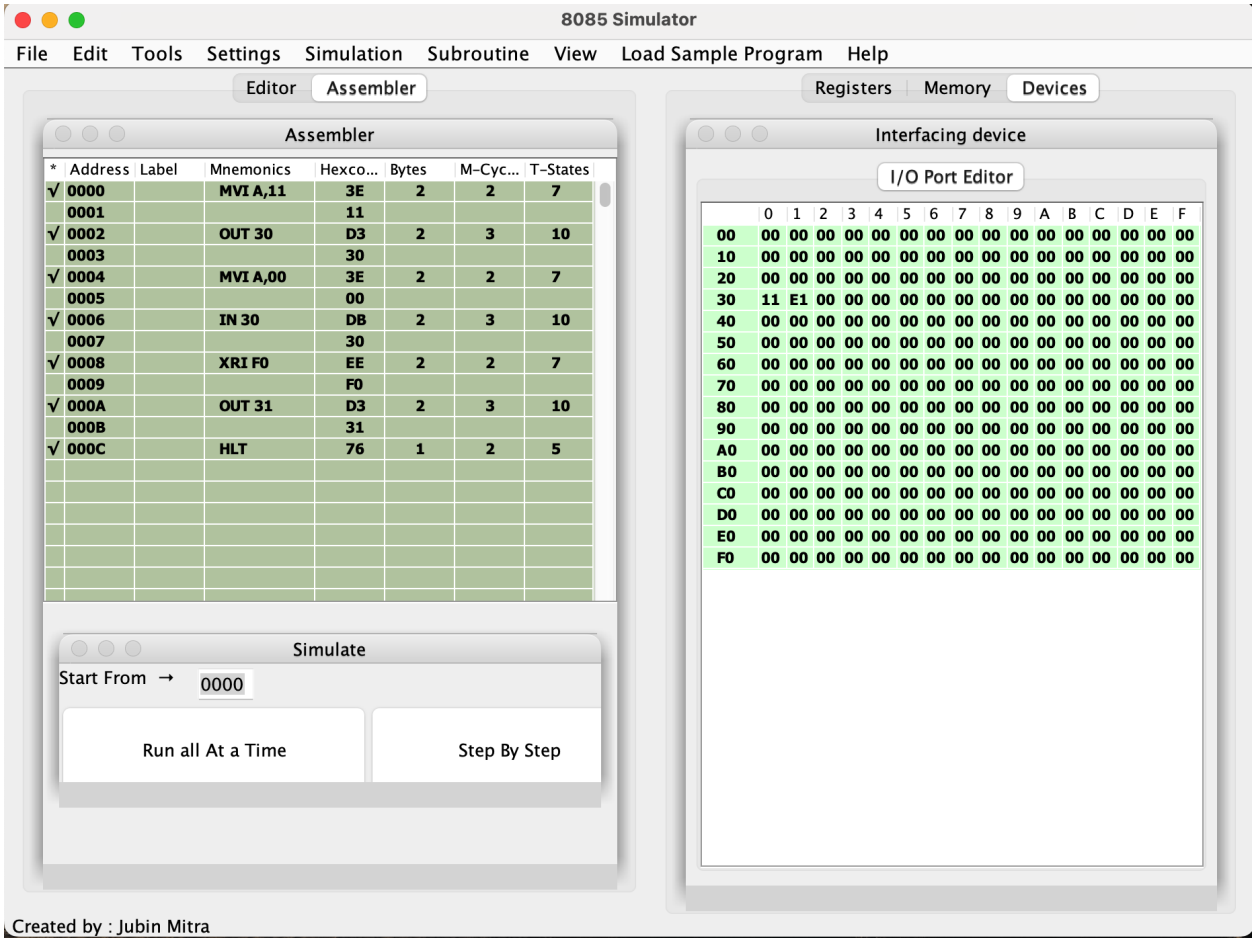
Address	Label	Mnemonics	hexCode	Bytes	M-Cycles	T-States
0000		MVI A, 11	3E	2	2	7
0001			11			
0002		OUT 03	D3	2	3	10
0003			03			
0004		MVI A,00	3E	2	2	7
0005			00			
0006		IN 03	DB	2	3	10
0007			03			
0008		XRI 0F	EE	2	2	7
0009			0F			
000A		OUT 04	D3	2	3	10
000B			04			
000C		HLT	76	1	2	5

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

KARIYA RAJ 21162101011



KARIYA RAJ 21162101011



G.) Code :

```
MVI A, 11H
OUT 30H
MVI A, 00H
IN 30H
XRI F0H
OUT 31H
HLT
```

Address	Label	Mnemonics	hexCode	Bytes	M-Cycles	T-States
0000		MVI A, 11	3E	2	2	7
0001			11			
0002		OUT 30	D3	2	3	10
0003			30			
0004		MVI A, 00	3E	2	2	7
0005			00			
0006		IN 30	DB	2	3	7
0007			30			
0008		XRI F0	EE	2	2	7
0009			F0			
000A		OUT 31	D3	2	3	10
000B			31			
000C		HLT	76	1	2	5

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

• RLC, RAL, RRC, RAR, CMP, CPI

7.	1	RLC	RLC Stands for "Rotate Left Accumulator". It rotates the accumulator contents to the left by 1-bit position.	MVI A, 11H RLC HLT
	2	RAL	RAL Stands for "Rotate Accumulator Left". It rotates the accumulator content to left by 1-bit Pos.	MVI A, 11H RAL HLT
	3	RRC	RRC Stands for "Rotate Right Accumulator". It rotates eight bits in the accumulator & one bit in the carry flag one bit position.	MVI A, 11H RRC HLT
	4	RAR	RAR Stands for "Rotate Accumulator Right including cy flag in rotation", It rotates the accumulator contents to the right by 1-bit position.	MVI A, 11H RAR HLT

5		CMP	The CMP instruction compares two operands. It is generally used in conditional execution. This basically subtracts one operand from the other comparing whether the operands are equal or not.	MOV A, 11H
				MOV B, 11H
				CMP B
				HLT

6		CPI	CPI stands for "compare immediate with Accumulator", for any 8 bit data. This inst^n is used to compare Accumulator with 8-bit immediate data.	MVI A, 11H CPI, 10H HLT