**** **GUJARAT TECHNOLOGICAL UNIVERSITY **

**Government Engineering College, Bhavnagar**

Subject: **Microprocessor & Interfacing**

**B.E. C.E. Semester-6th**

(**Computer Branch**)

**Submitted By:**

**Name: Vankani Arjun BakulBhai**

**Enrollment: 180210107060**

**Prof. Hardi Sanghavi Prof. KARSHAN KANDORIYA**

(Faculty Guide) (Head of the Department)

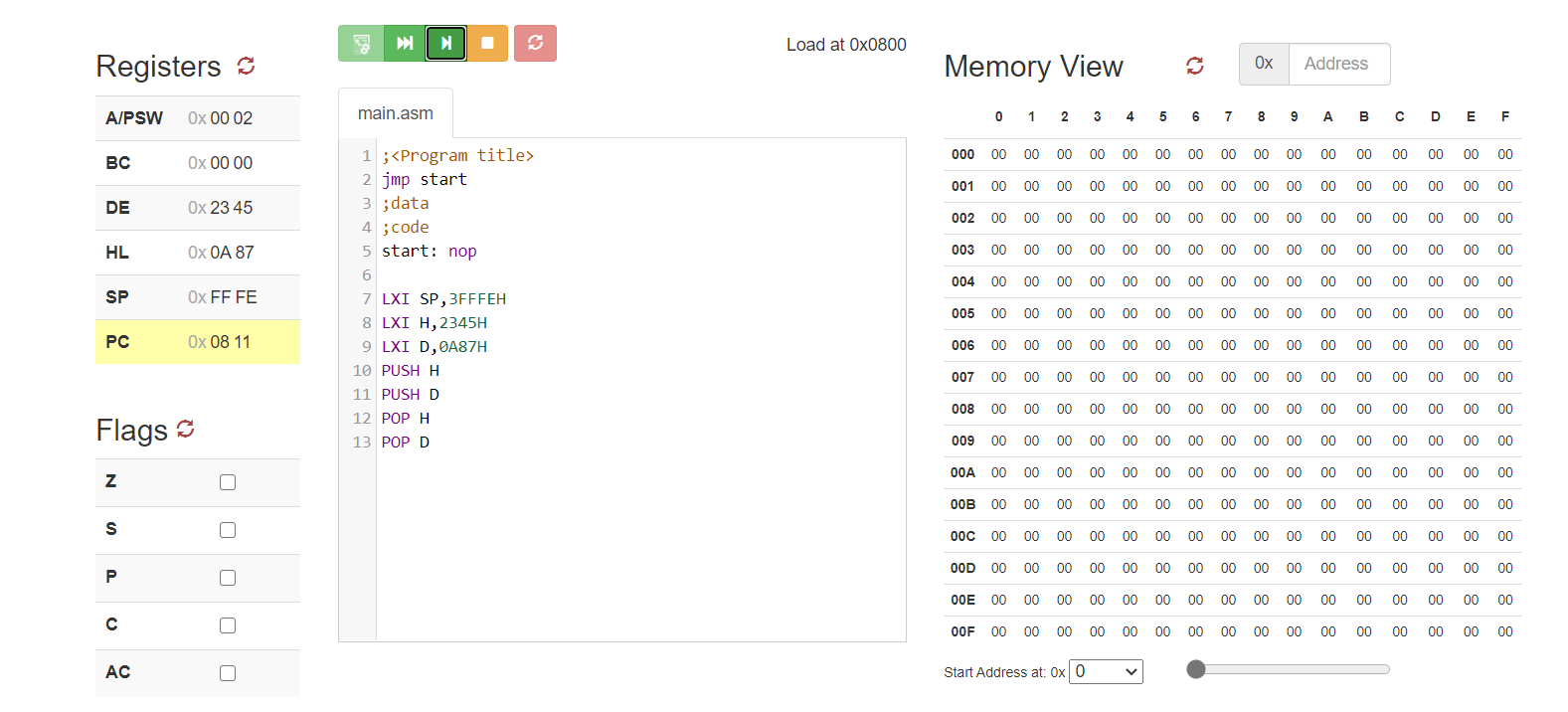
Academic Year (2020-21)

|  |  |  |
| --- | --- | --- |
| **SR No.** | **(Lab work)** | **Page No** |
| **1** | **LAB 1** | **03** |
| **2** | **LAB 2** | **05** |
| **3** | **LAB 3** | **07** |
| **4** | **LAB 4** | **09** |
| **5** | **LAB 5** | **11** |
| **6** | **LAB 6** | **14** |
| **7** | **LAB 7** | **15** |
| **8** | **LAB 8** | **19** |
| **9** | **LAB 9** | **21** |
| **10** | **LAB 10** | **24** |
| **11** | **LAB 11** | **31** |

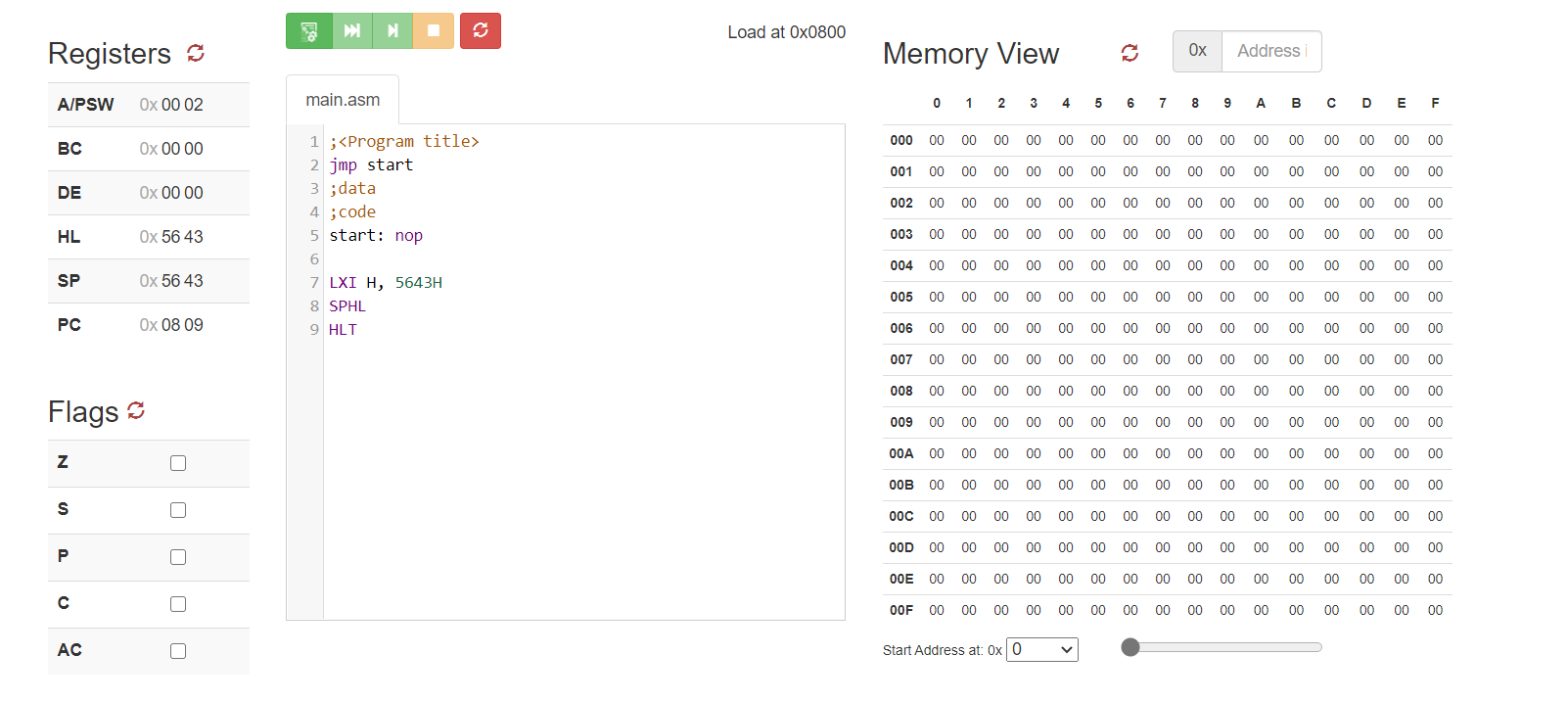
**Microprocessor & Interfacing**

**(Lab session-01)**

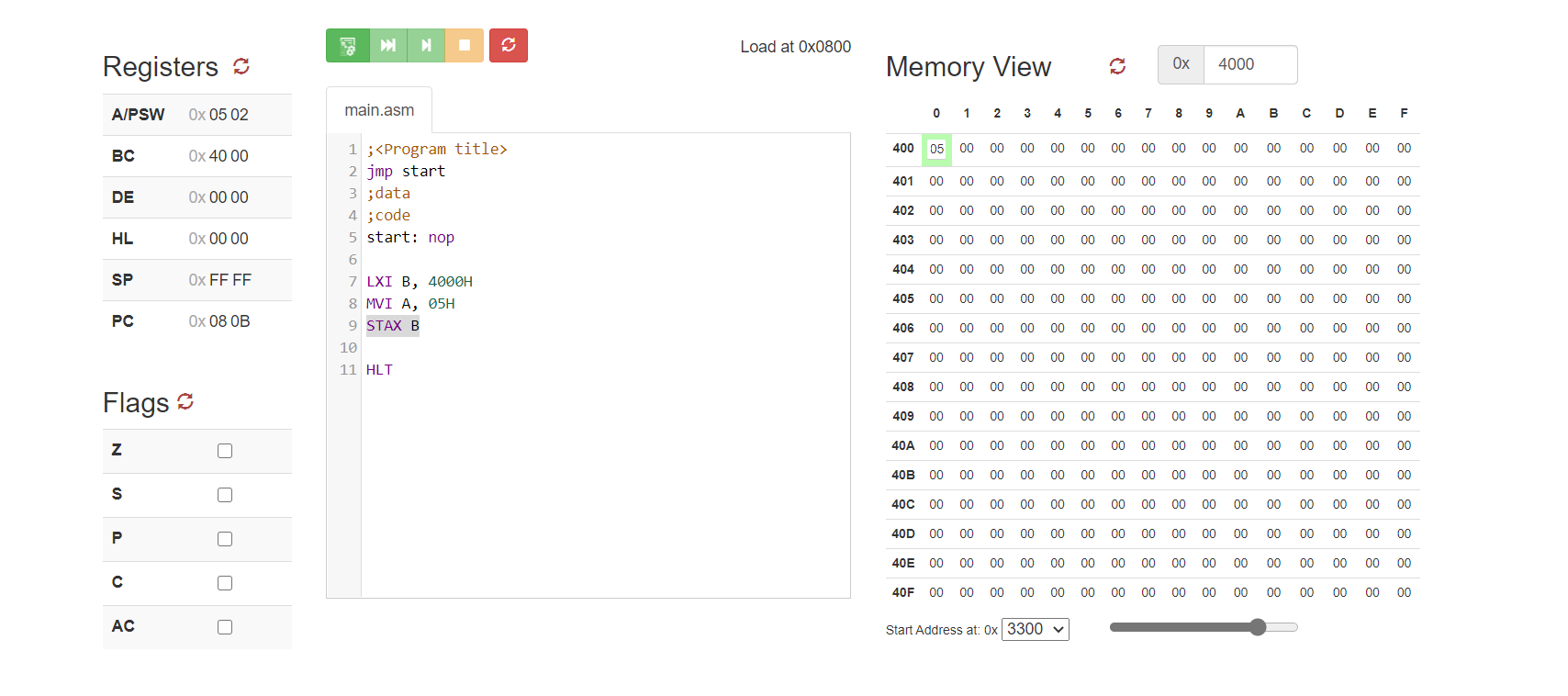
1. **Write an 8085 program to exchange the contents of HL=2345H and DE=0A87H register pairs using the push and pop instructions.**



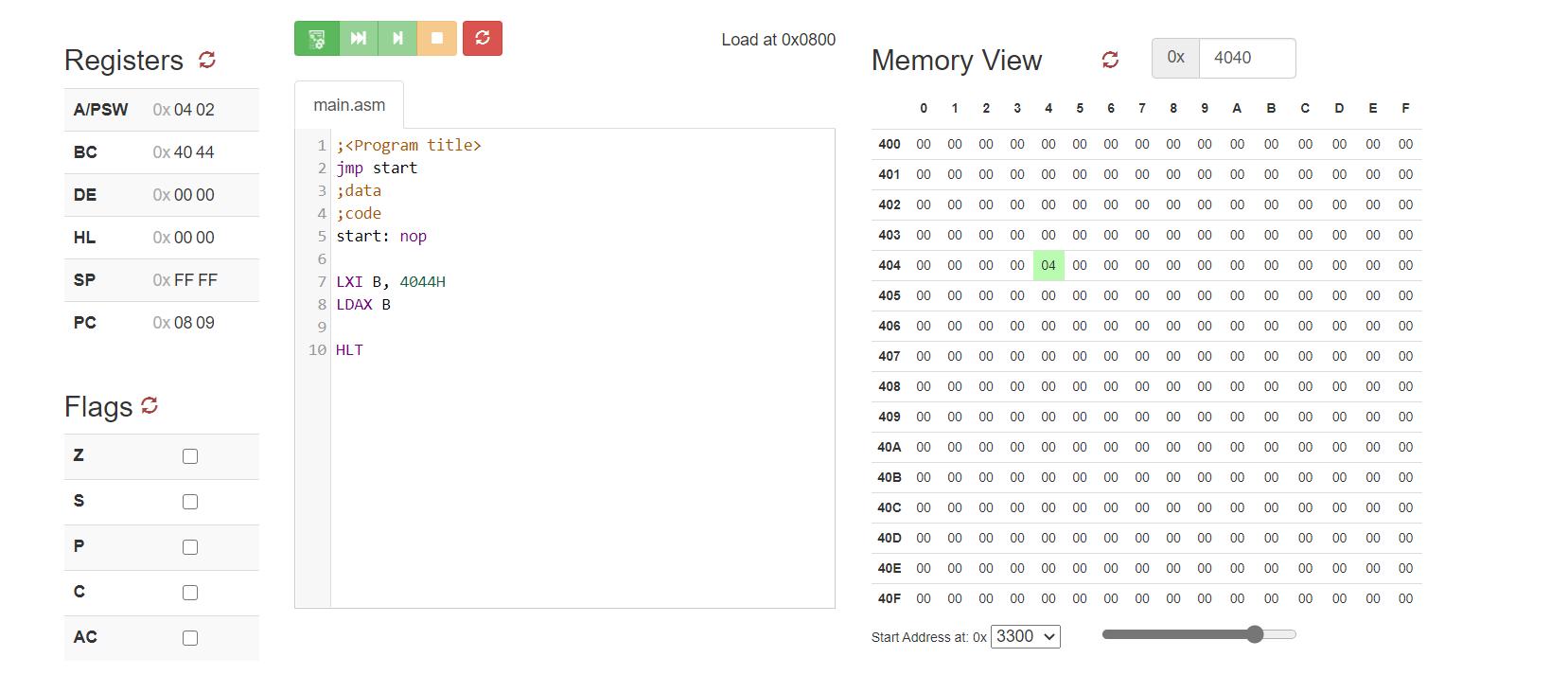
1. **Write a 8085 program to store the content of HL=5643H register pair on to SP.**



1. **Write an 8085 program to store the contents of A register into memory location 4000H. (Hint: use STAX instruction)**



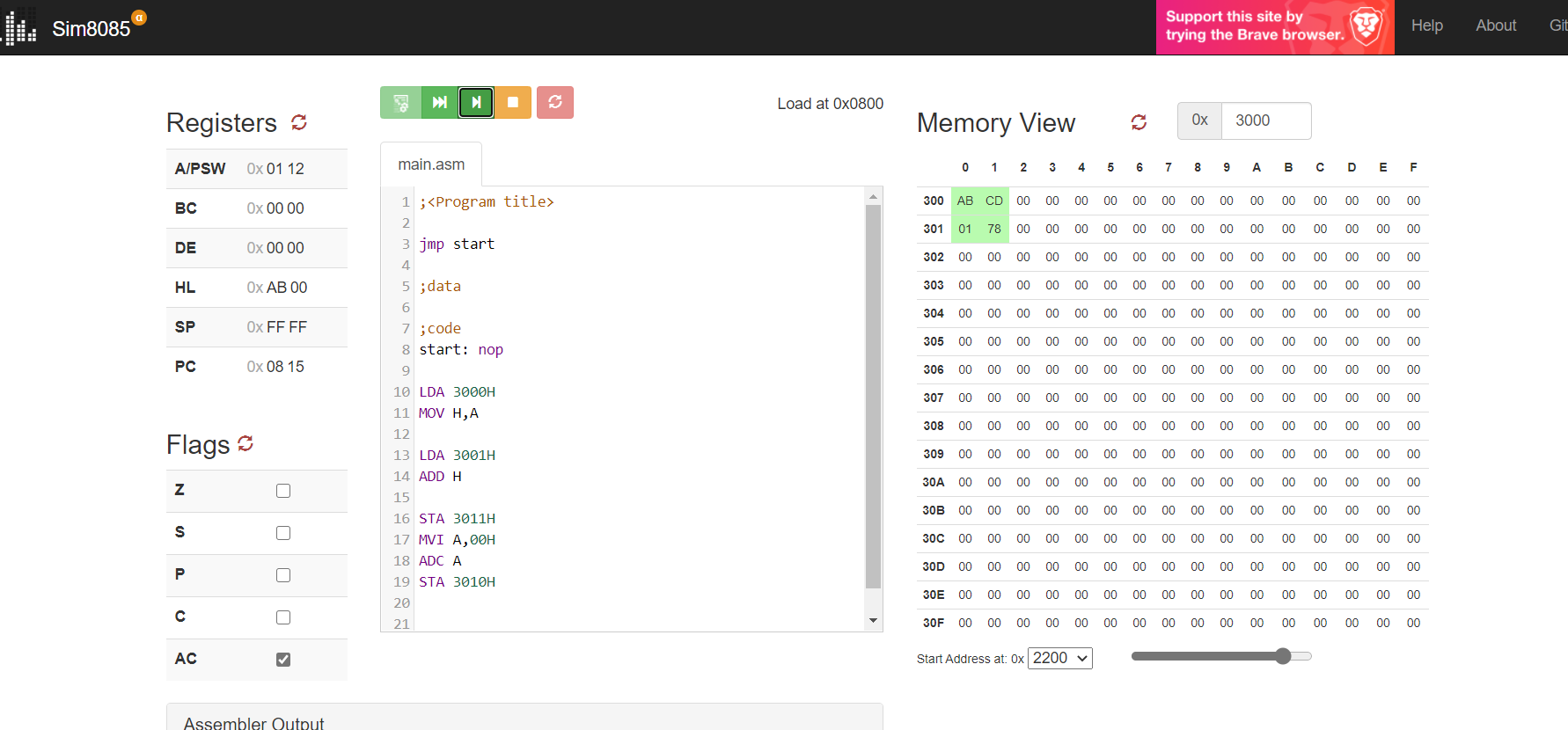
1. **Write an 8085 program to get the contents of memory location 4040H into A register only. (Hint: use LDAX instruction)**

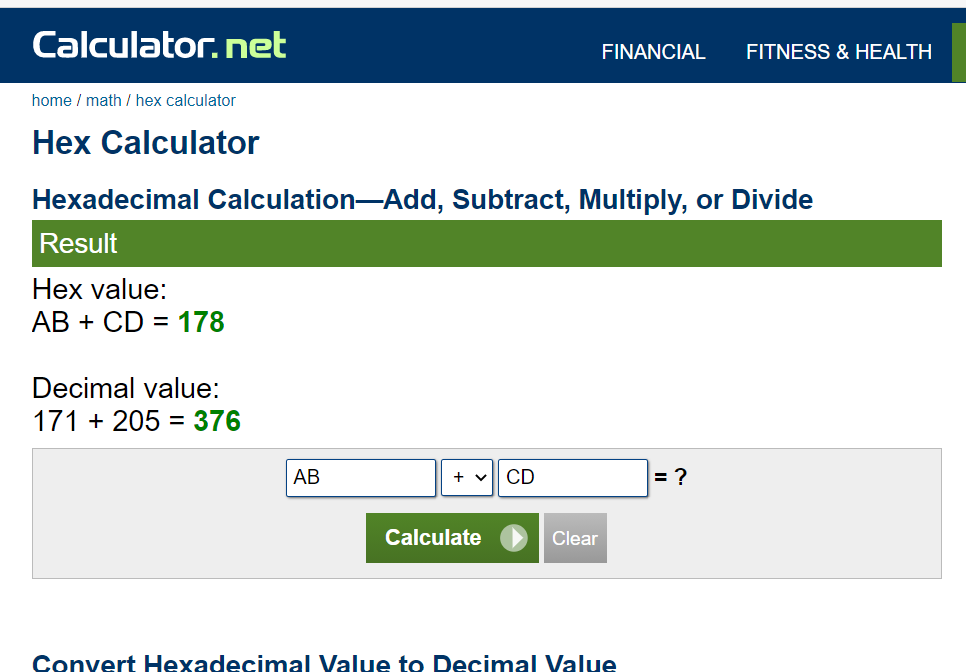


**Microprocessor & Interfacing**

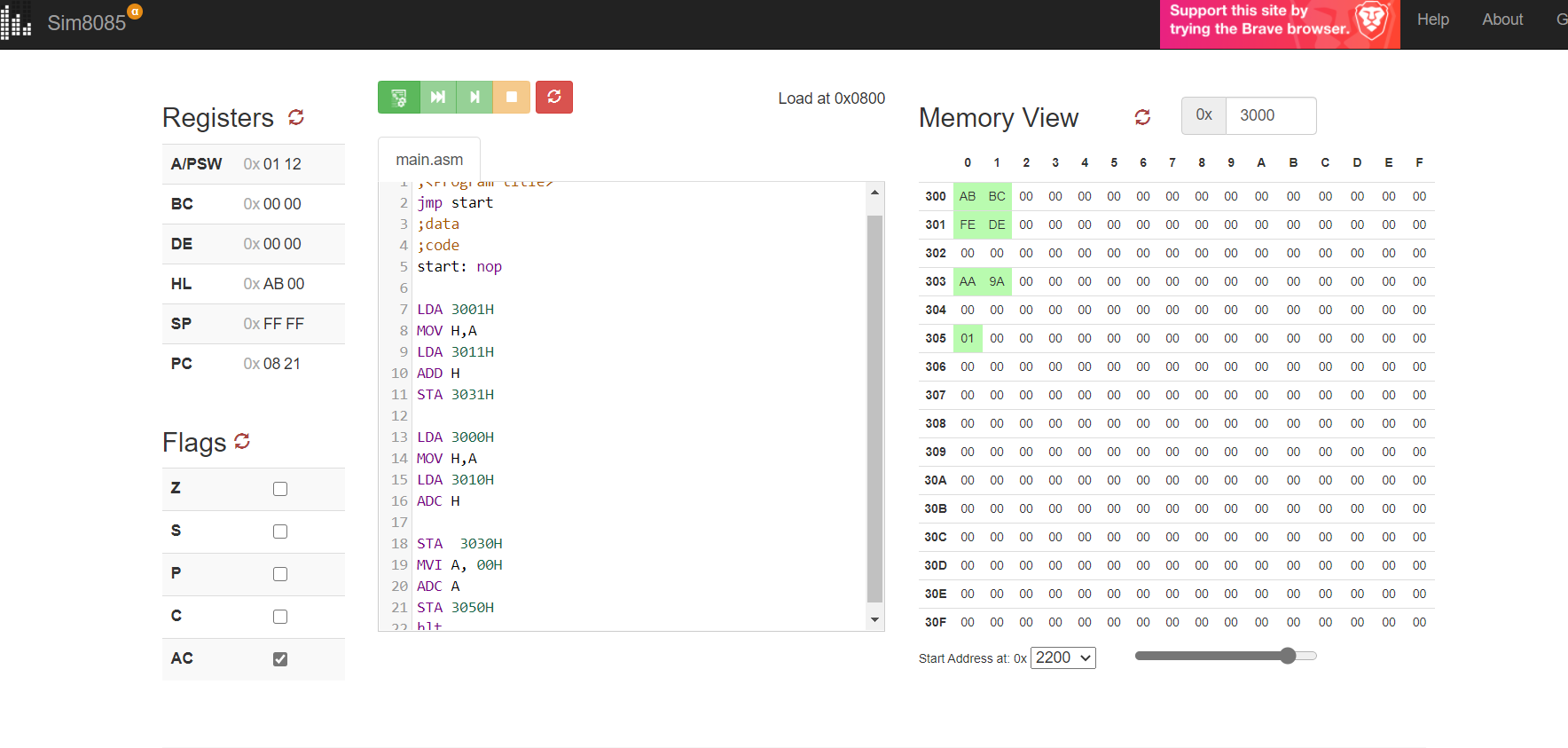
**(Lab session-02)**

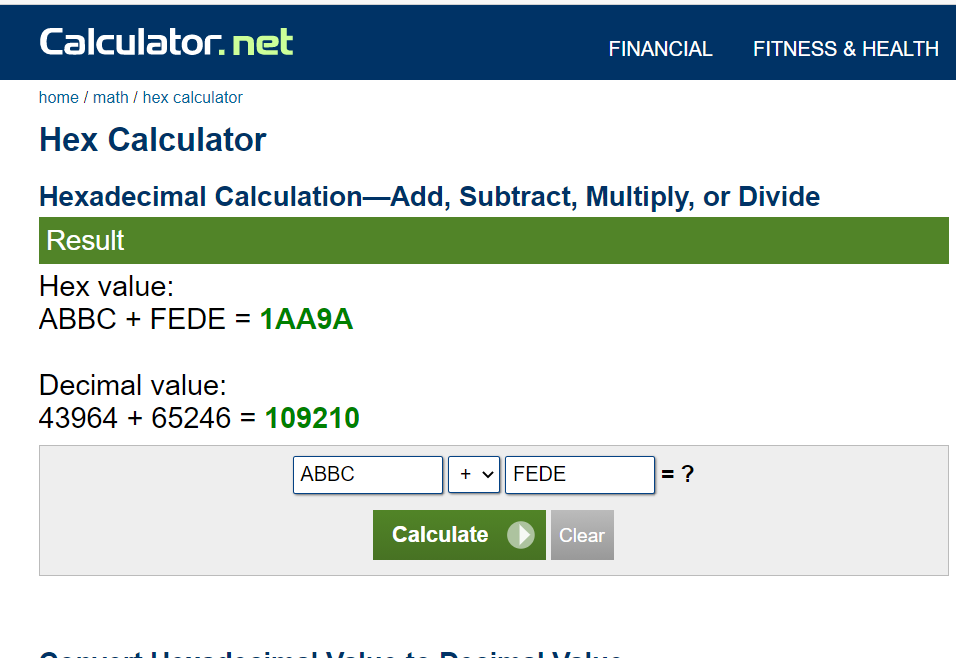
* **Addition of 8 bit**

****

****

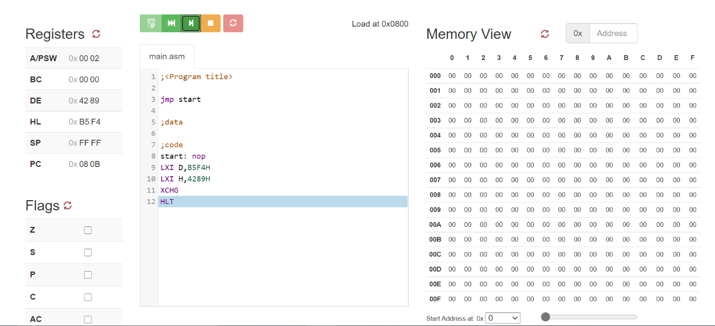
* **Addition of 16 bit**

****

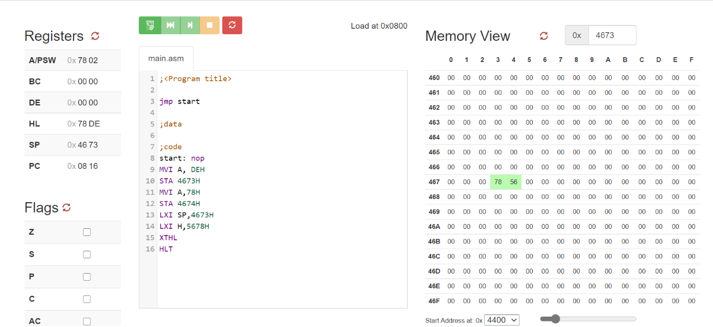
****

**Microprocessor & Interfacing**

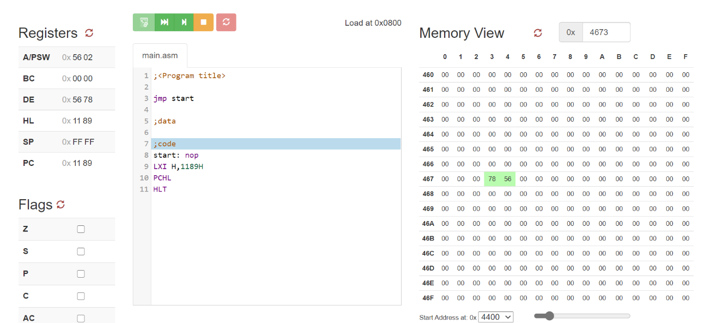
**(Lab session-03)**

**Q-1) Write an Assembly language program to exchange the contents of DE and HL register pair. The contents of DE=B5F4H and HL=4289H.  
**

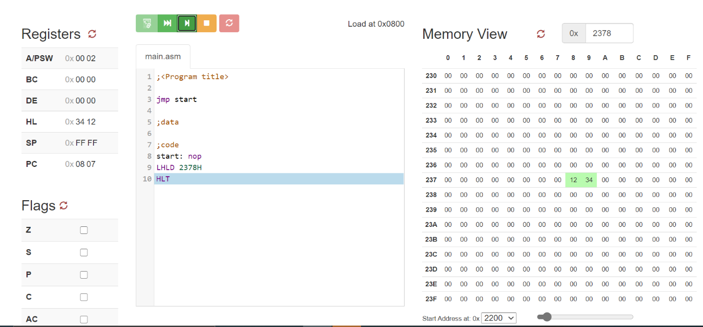
**Q-2) Write an Assembly language program to exchange the content of HL register pair which is 5678H with the contents stored at 4673H and 4674H which are DEH and 78H respectively.**

****

**Q-3) Write an Assembly language program to initialize the content of PC register with 1189H using HL register pair.**

****

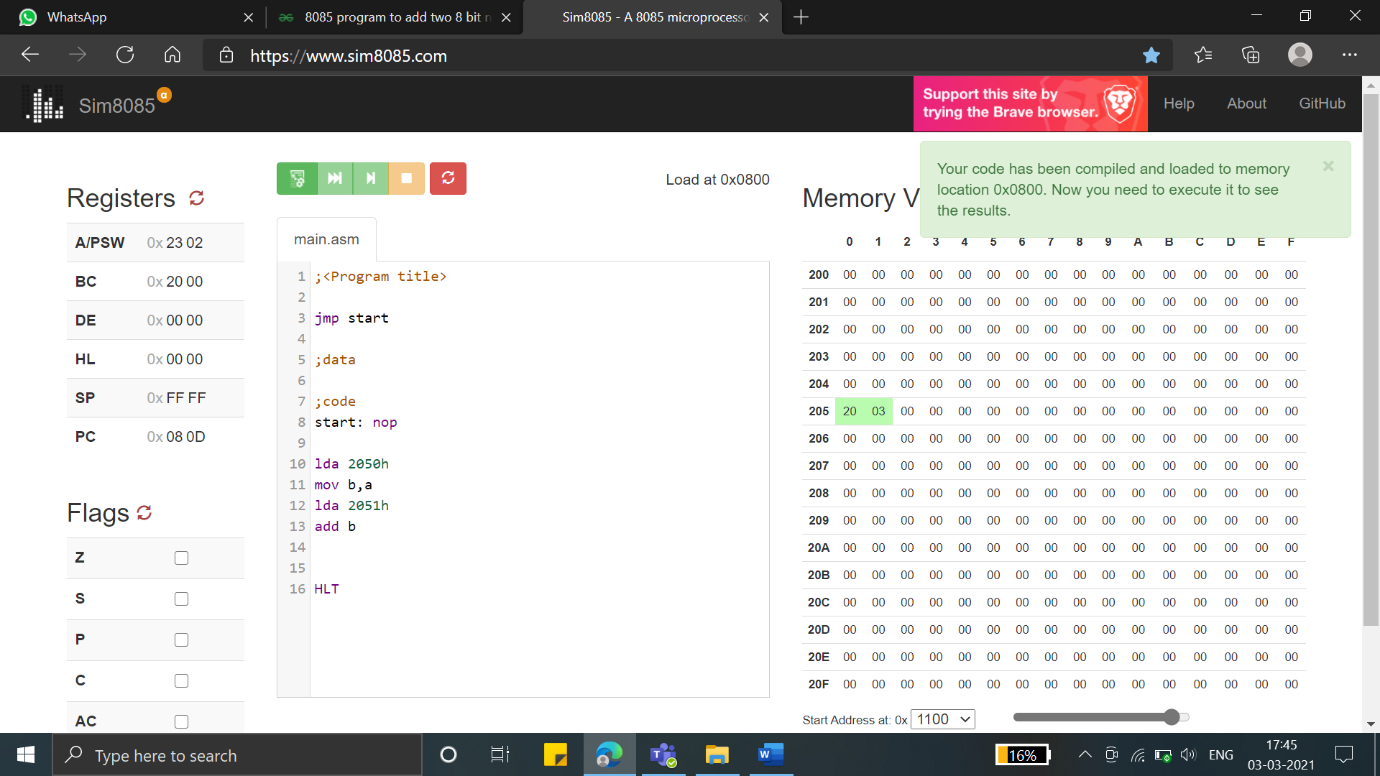
**Q-4) Write an Assembly language program to load the contents stored at two consecutive memory locations starting from 2378H into HL register pair. Use only one single instruction to complete the process of loading data.**



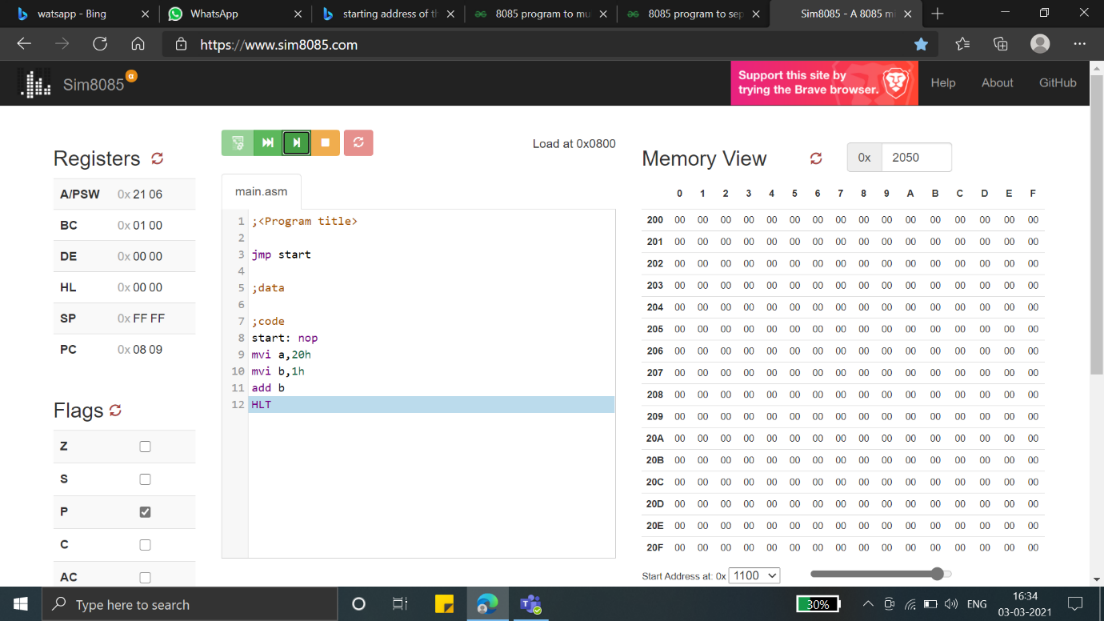
**Microprocessor & Interfacing**

**(Lab session-04)**

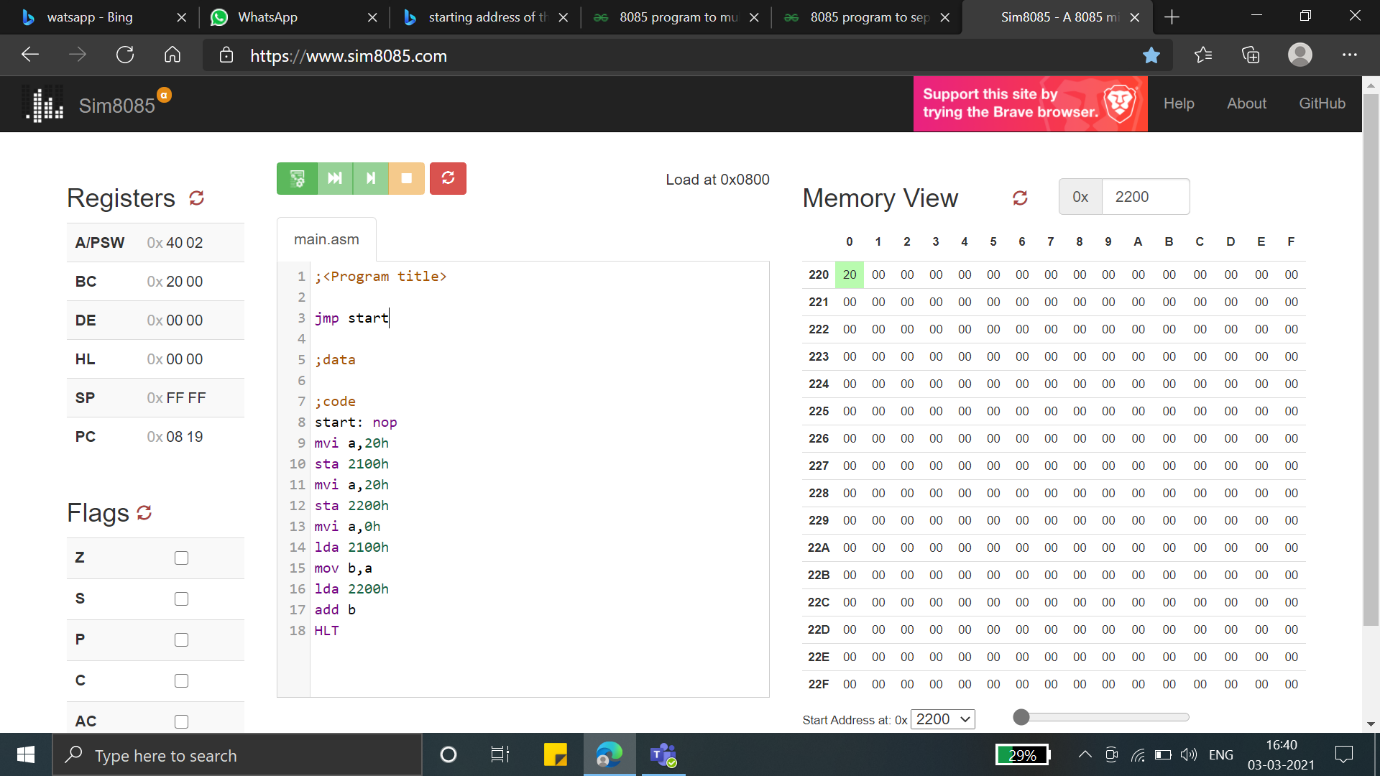
1. **Write an assembly language program to add two 8-bit numbers stored at address 2050 and address 2051 in 8085 microprocessors. The starting address of the program is taken as 2000.**

****

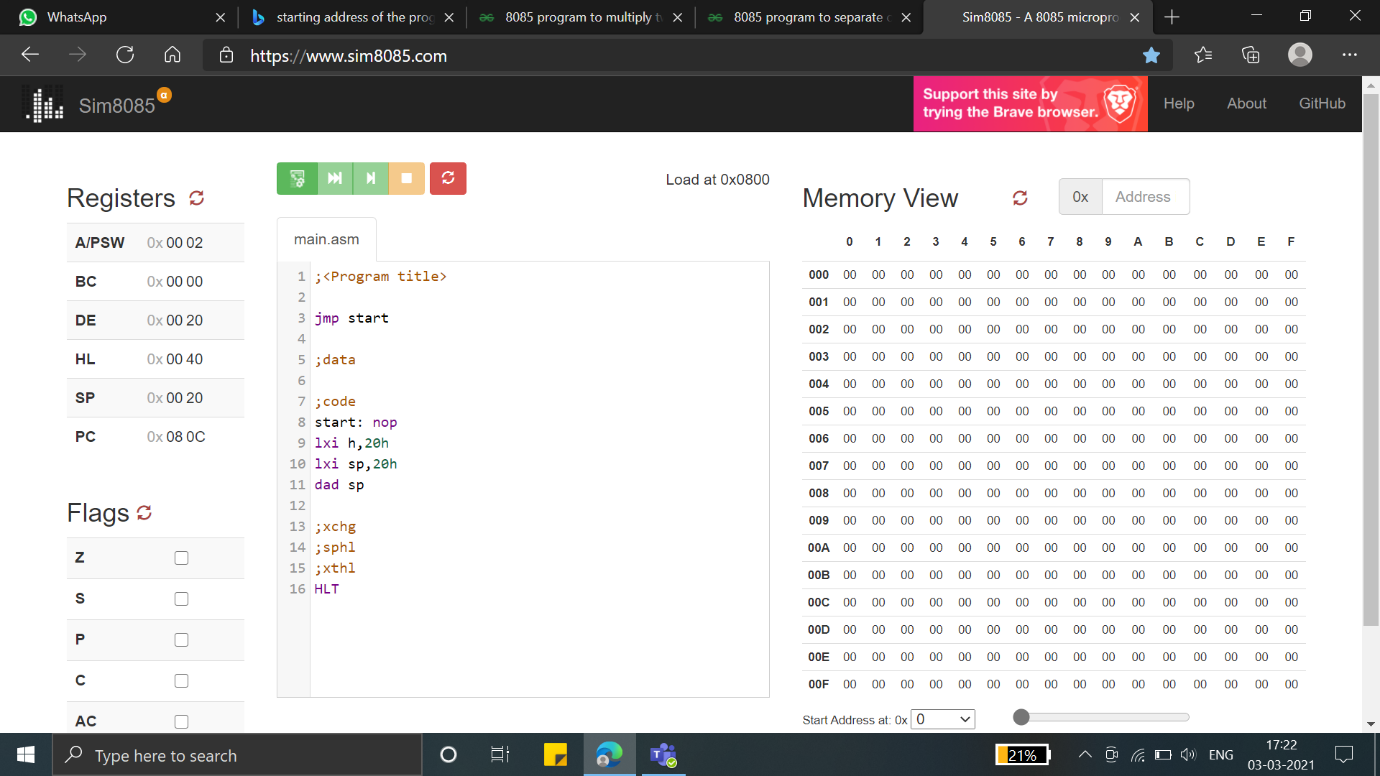
1. **Write assembly language program to do addition of two 8-bit numbers.**

****

1. **Write an 8085 program to add two 16-bit nos stored in memory locations 2100H and 2200H respectively.**

****

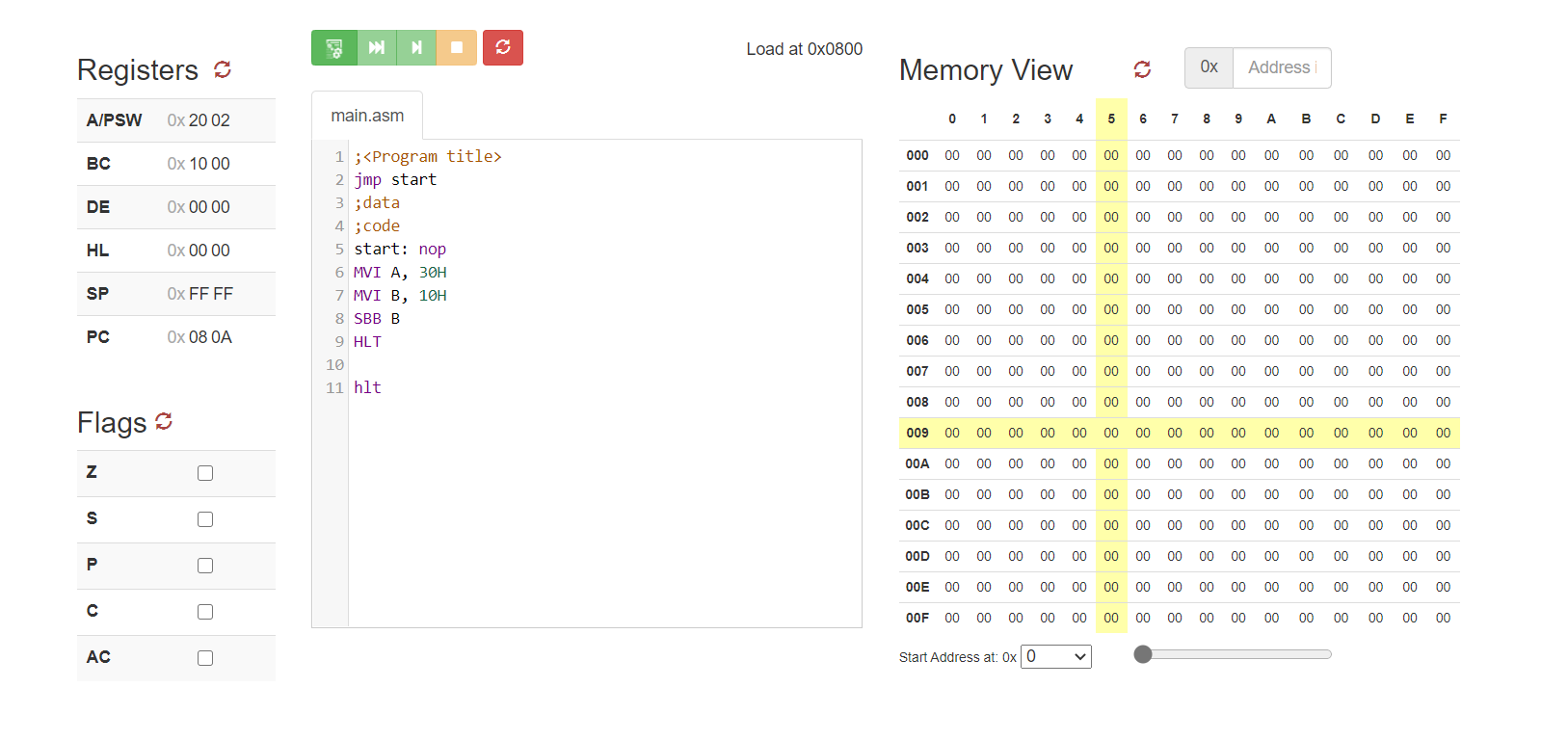
1. **Write an 8085 program to add the contents of HL and SP register pairs.**

****

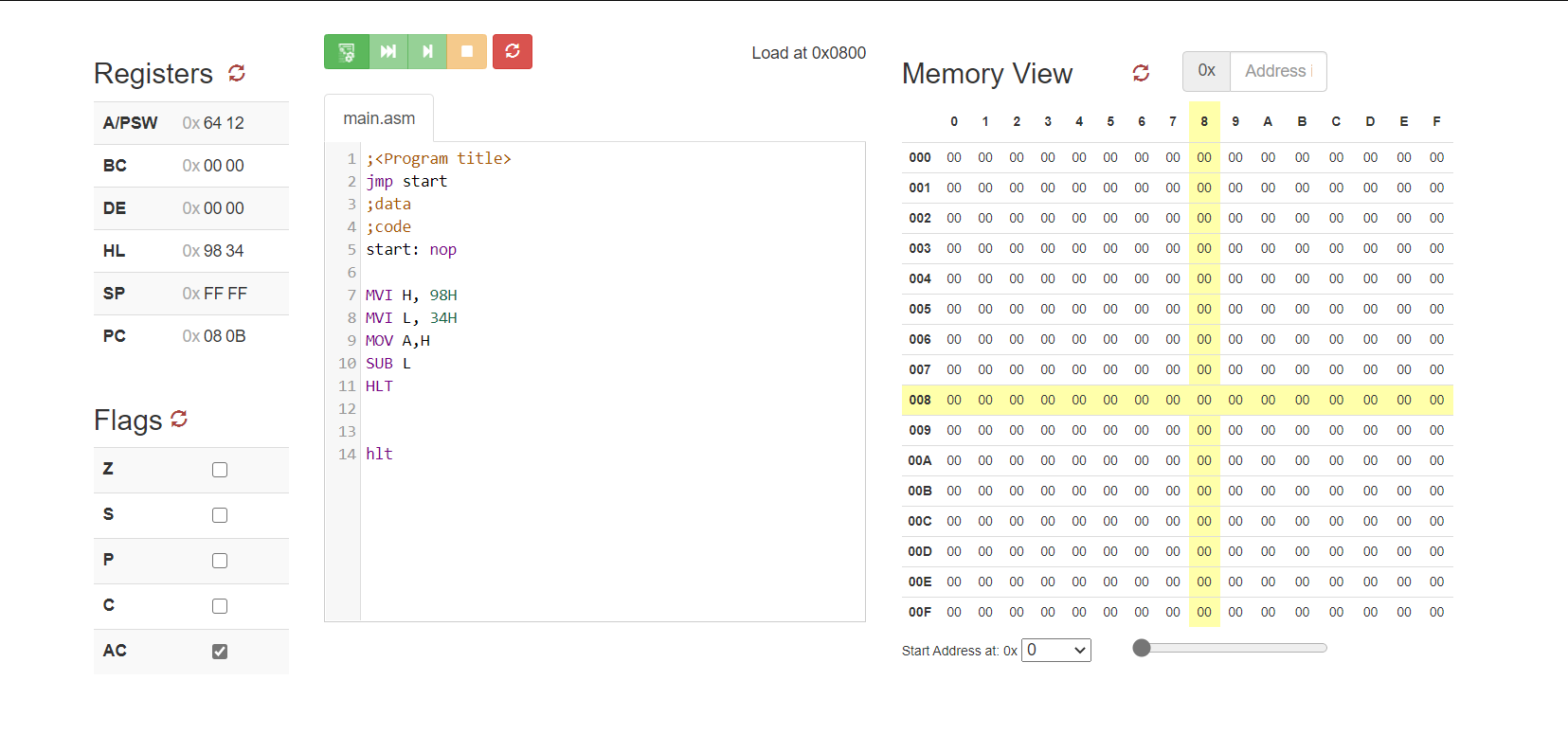
**Microprocessor & Interfacing**

**(Lab session-05)**

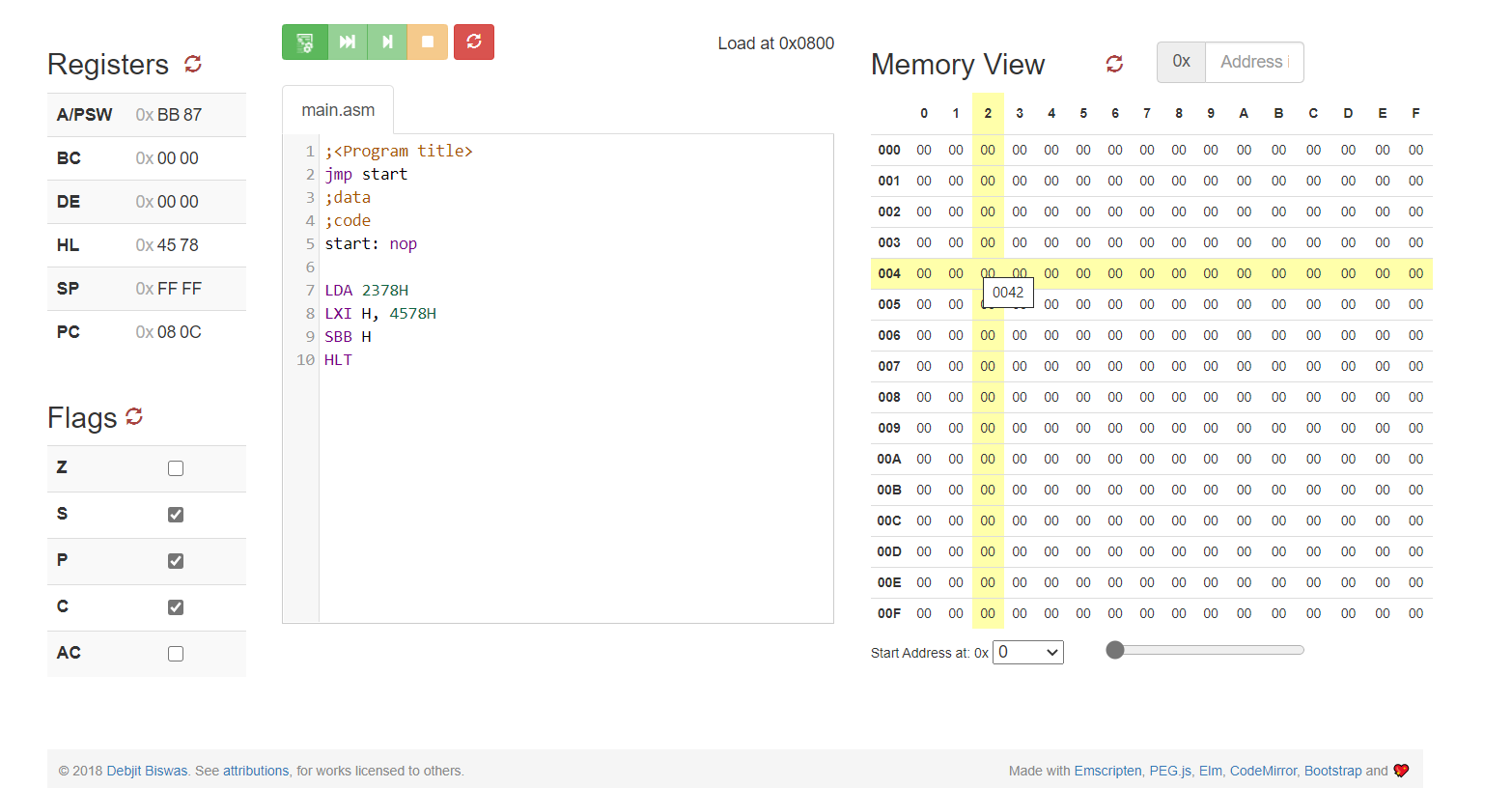
1. **Write an 8085 program to subtract two 8-bit numbers with borrow bit. Contents are A=30H and B=10H**

****

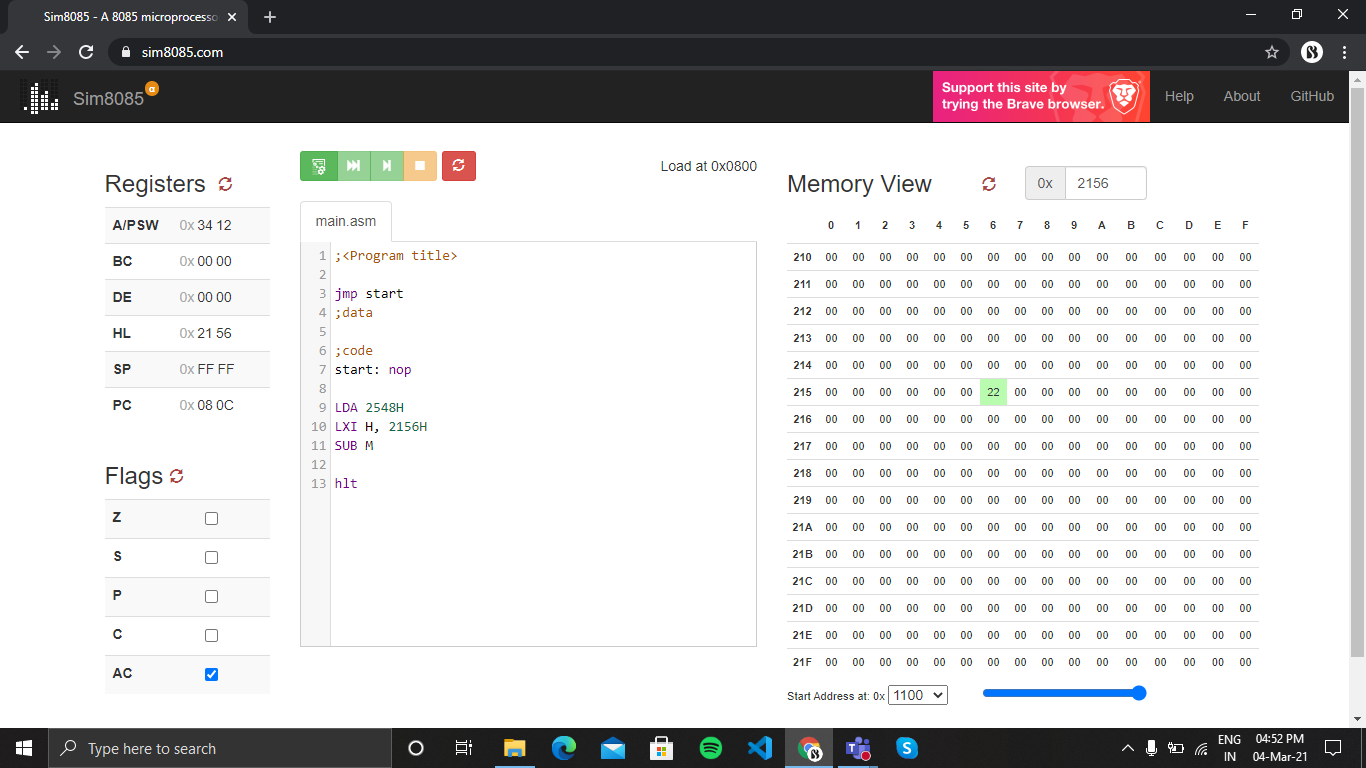
1. **Write an 8085 program to subtract two 8-bit numbers without borrow bit. Contents are H=98H and L=34H**



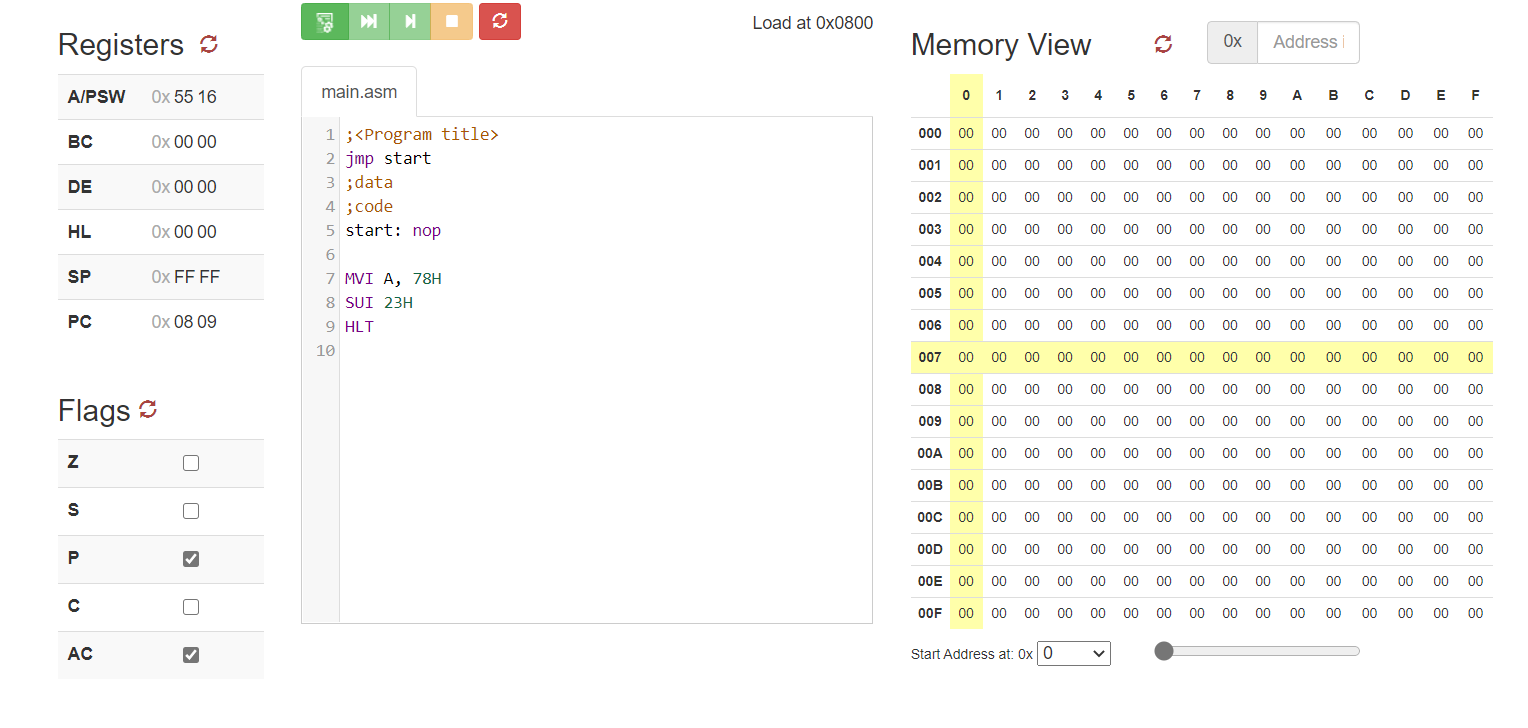
1. **Write an 8085 program to subtract two 8-bit numbers with borrow bit. Contents are [2378H] =09H and [4578H] =03H**

****

1. **Write an 8085 program to subtract two 8-bit numbers without borrow bit. Contents are [2548H] =56H and [2156H] =22H**



1. **Write an 8085 program to subtract 23H data from the content of accumulator A which is 78H.**

****

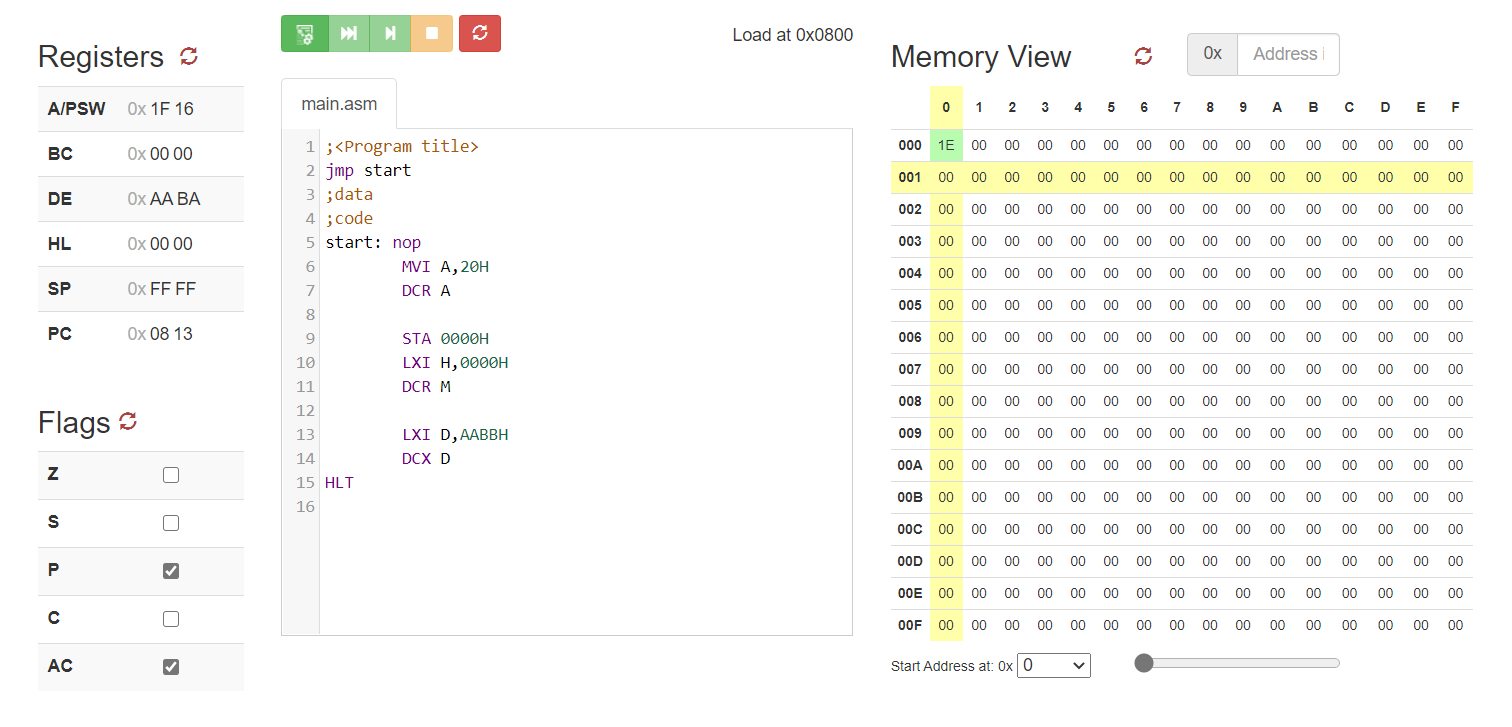
1. **Write an 8085 program to subtract 38H data and carry bit from the content of accumulator A which is 98H.**

****

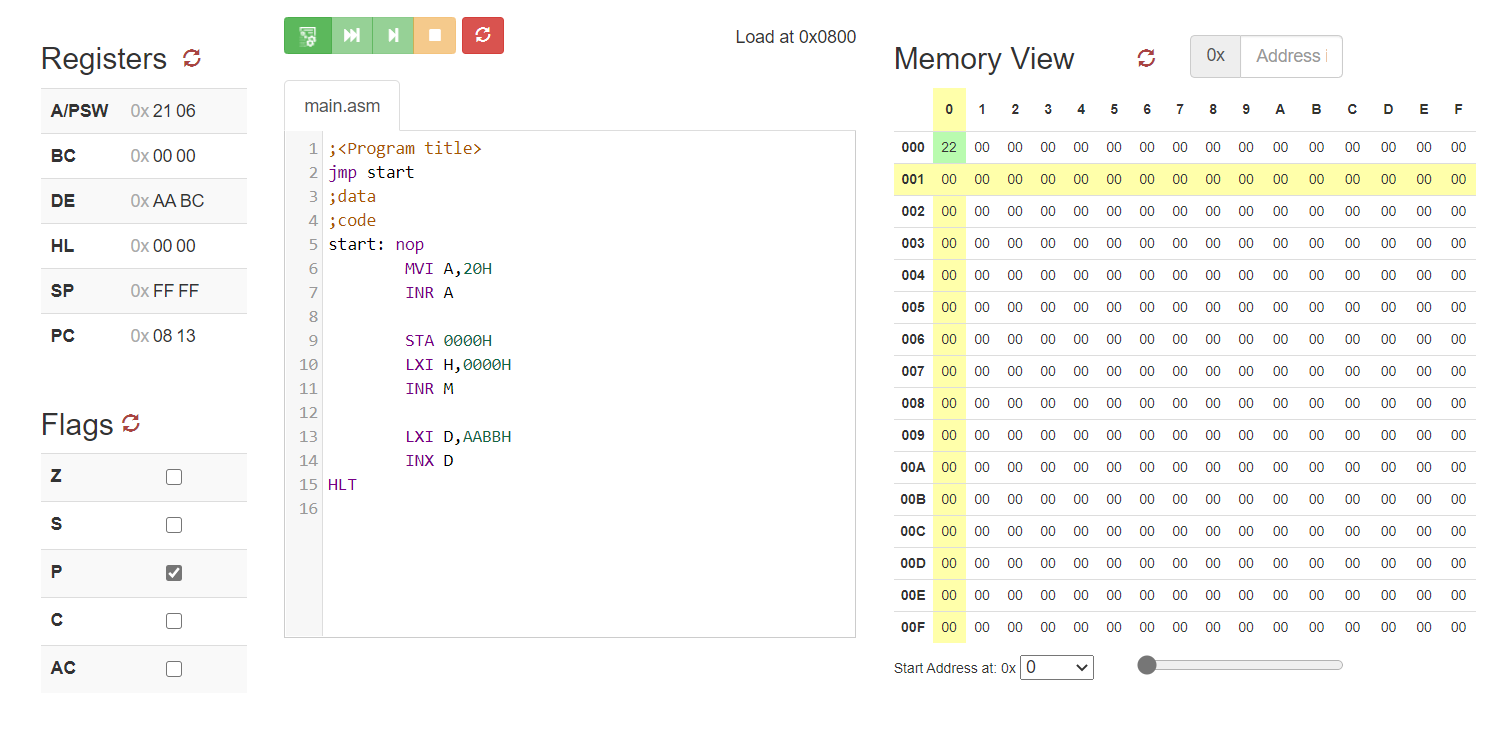
**Microprocessor & Interfacing**

**(Lab session-06)**

* **Increment Decrement Instructions**
* **Write an 8085 program to demonstrate increment operation for 8-bit and 16-bit data.**



* **Write an 8085 program to demonstrate decrement operation for 8-bit and 16-bit data.**



**Microprocessor & Interfacing**

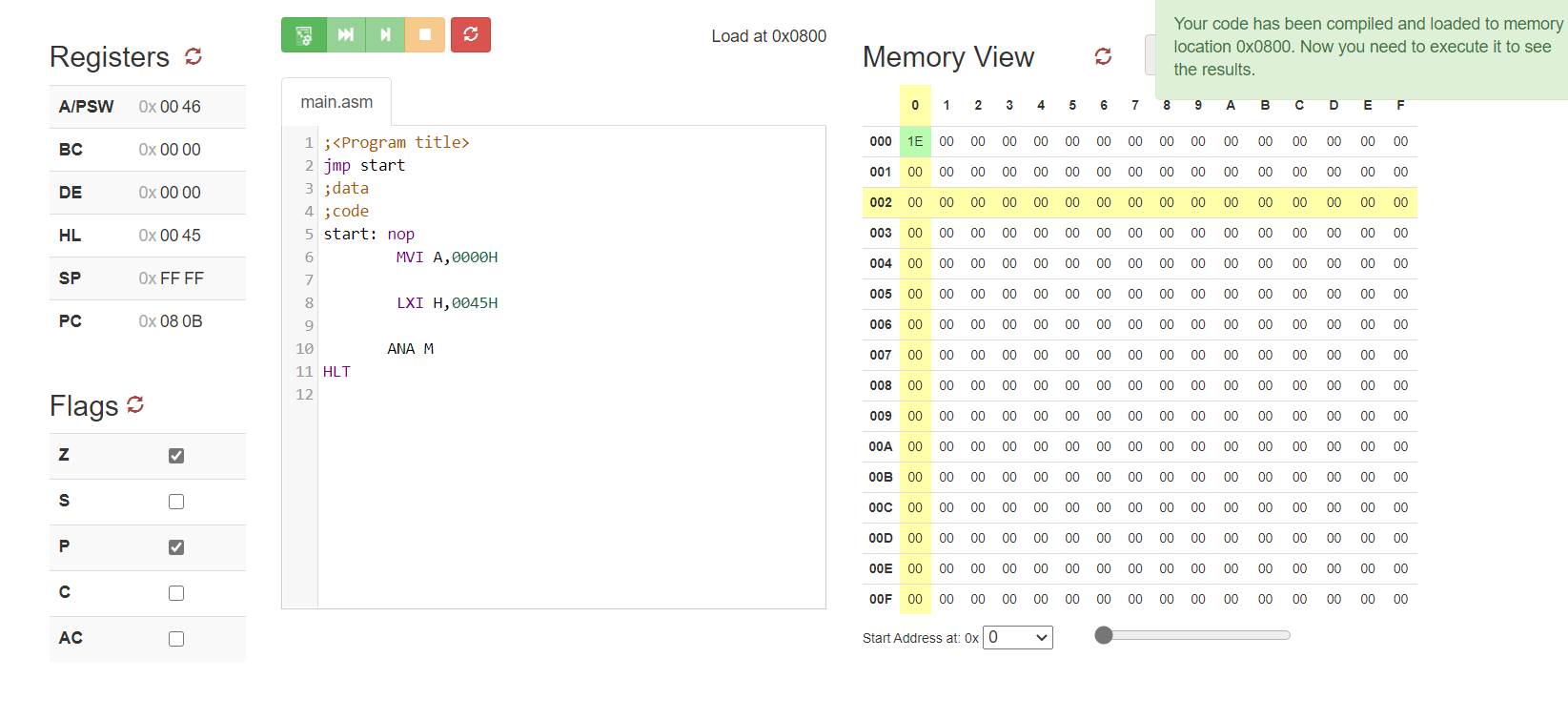
**(Lab session-07)**

* **Demonstrate AND, OR and XOR instructions through 8085 programs for following:**

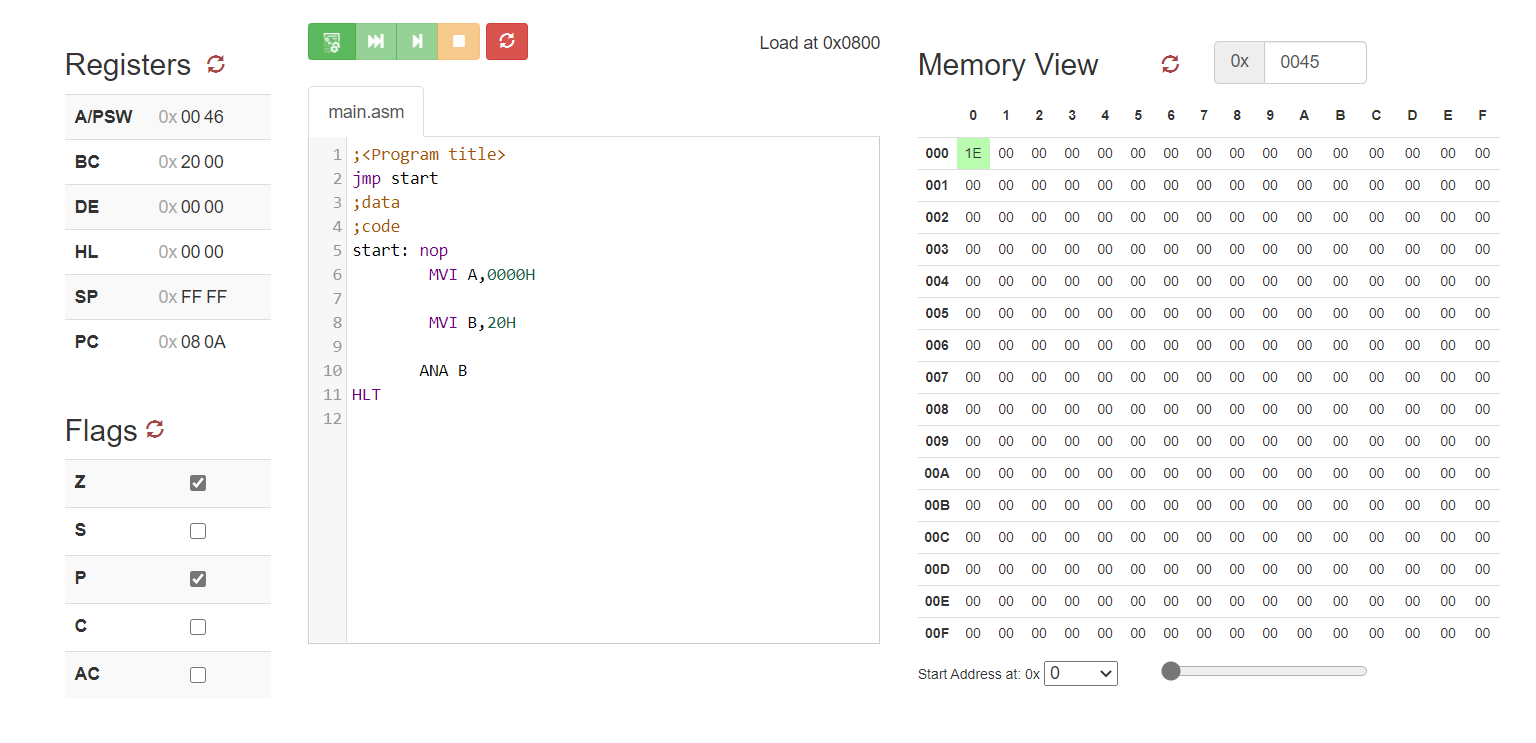
**Assume A contains 89H, Memory Address=0045H**

**1) ANA R, ANA M, ANI 56H**

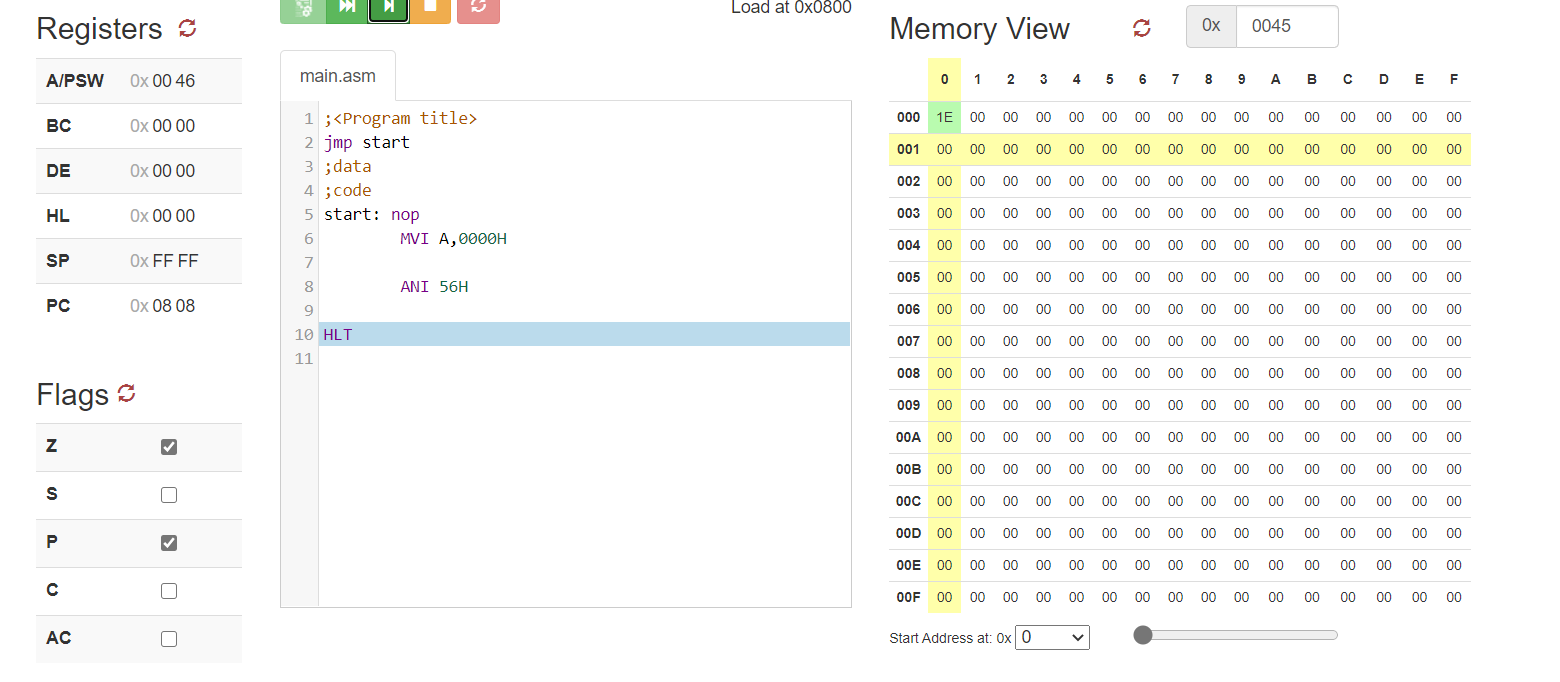
* **ANA M**



* **ANA R**

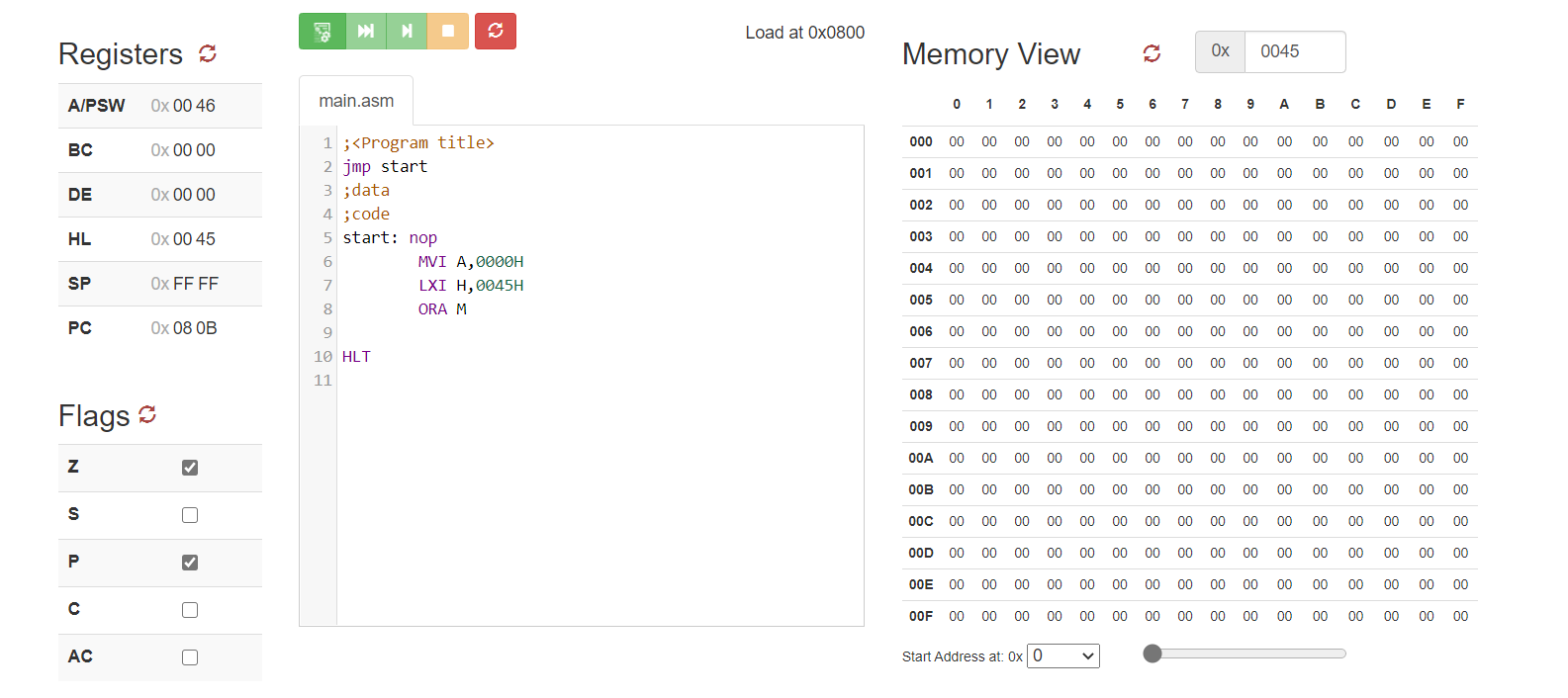


* **ANI 56H**

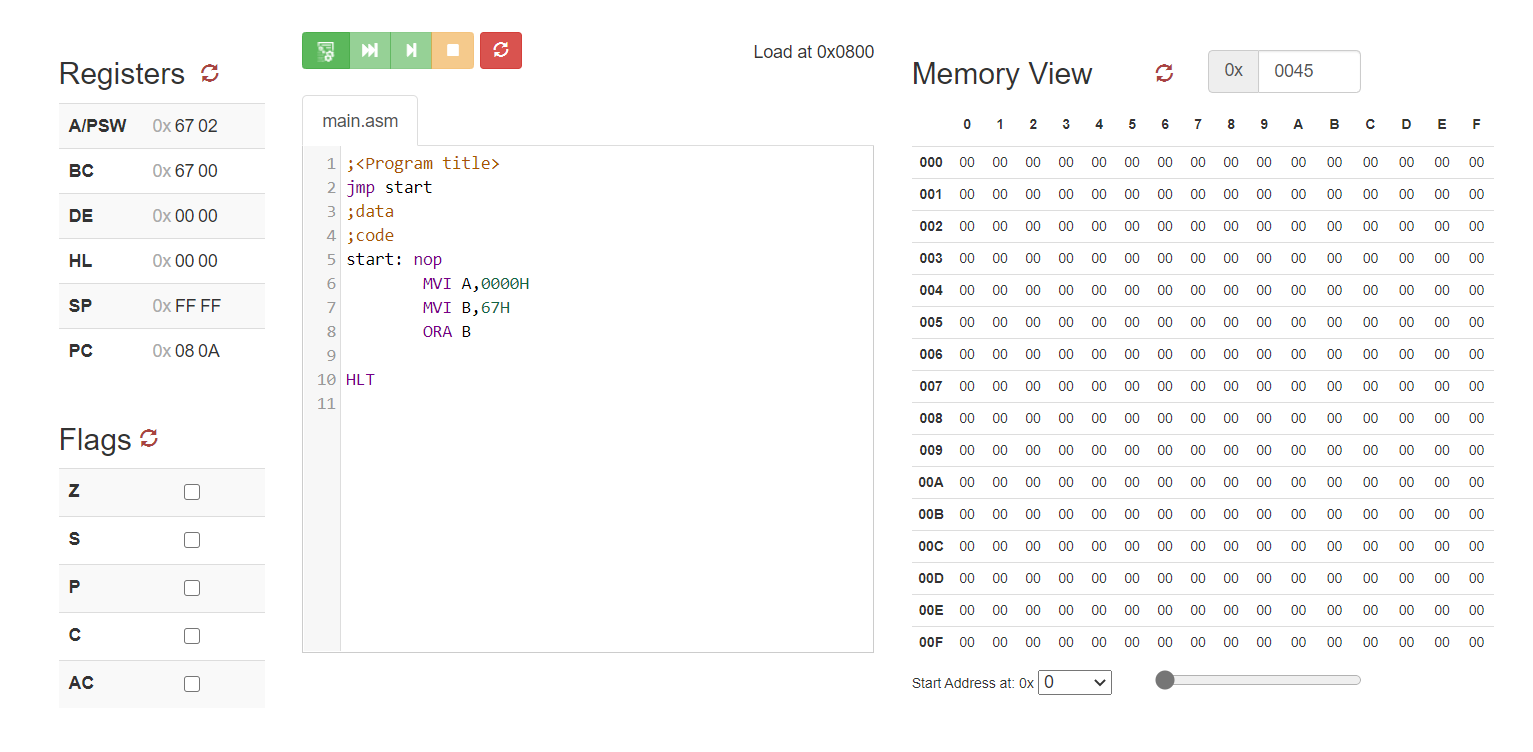


**2) ORA R, ORA M, ORI 34H**

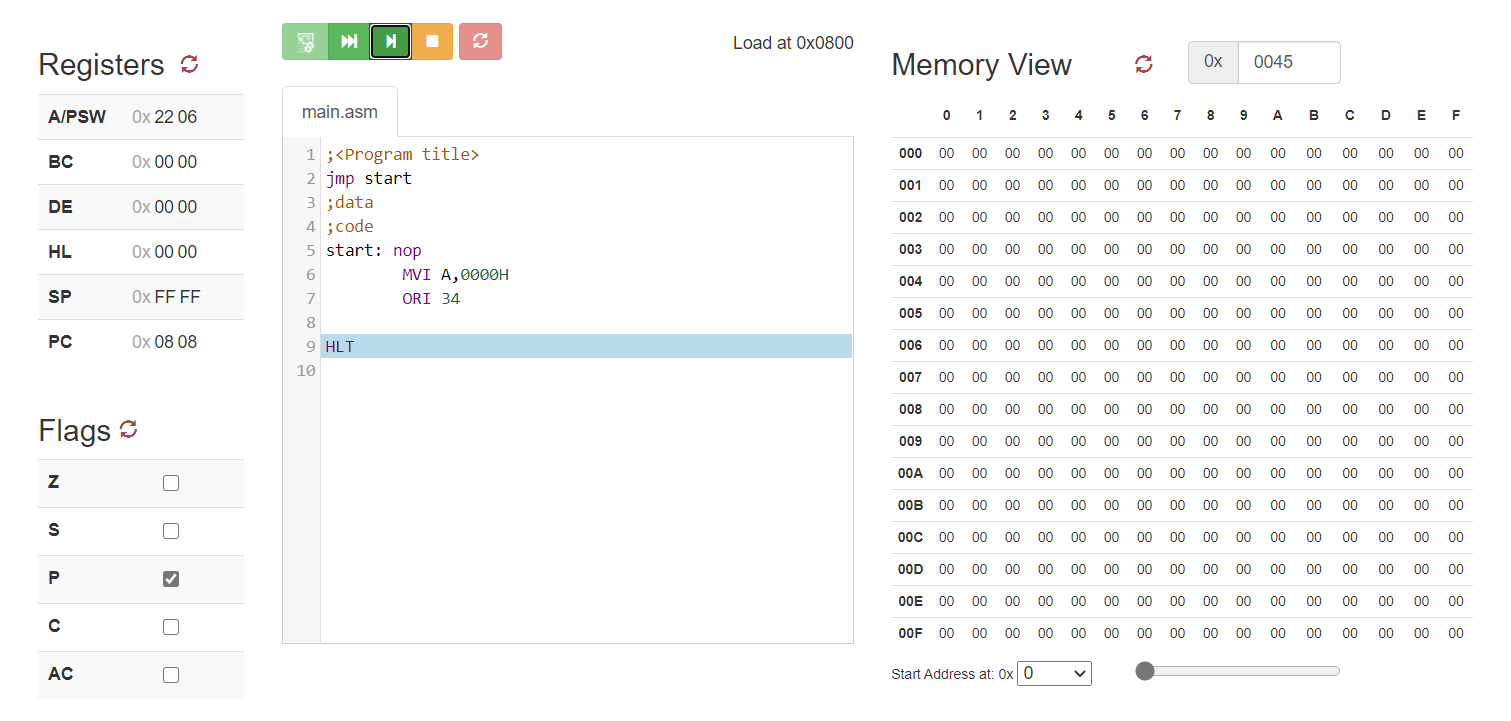
* **ORA** M



* **ORA R:**

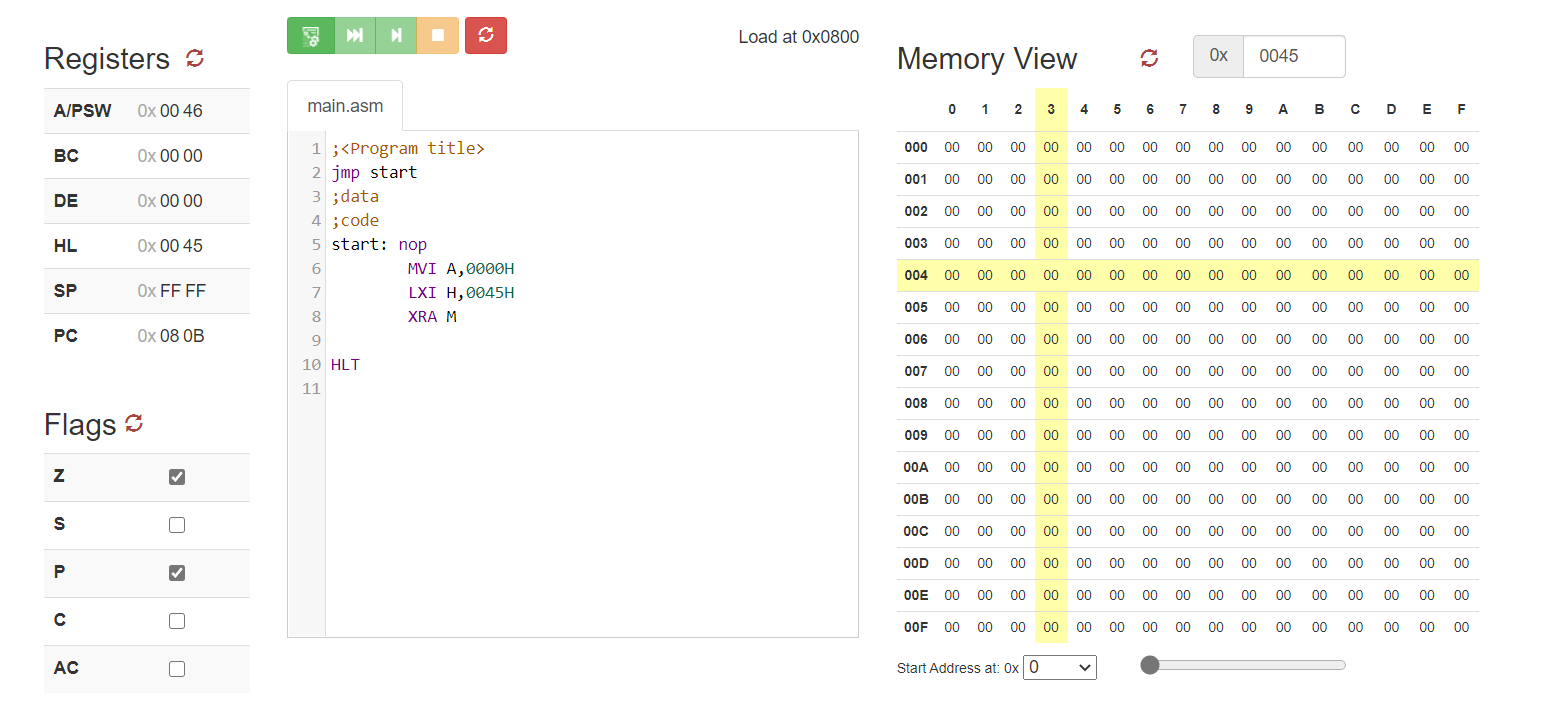


* **ORI 34H**

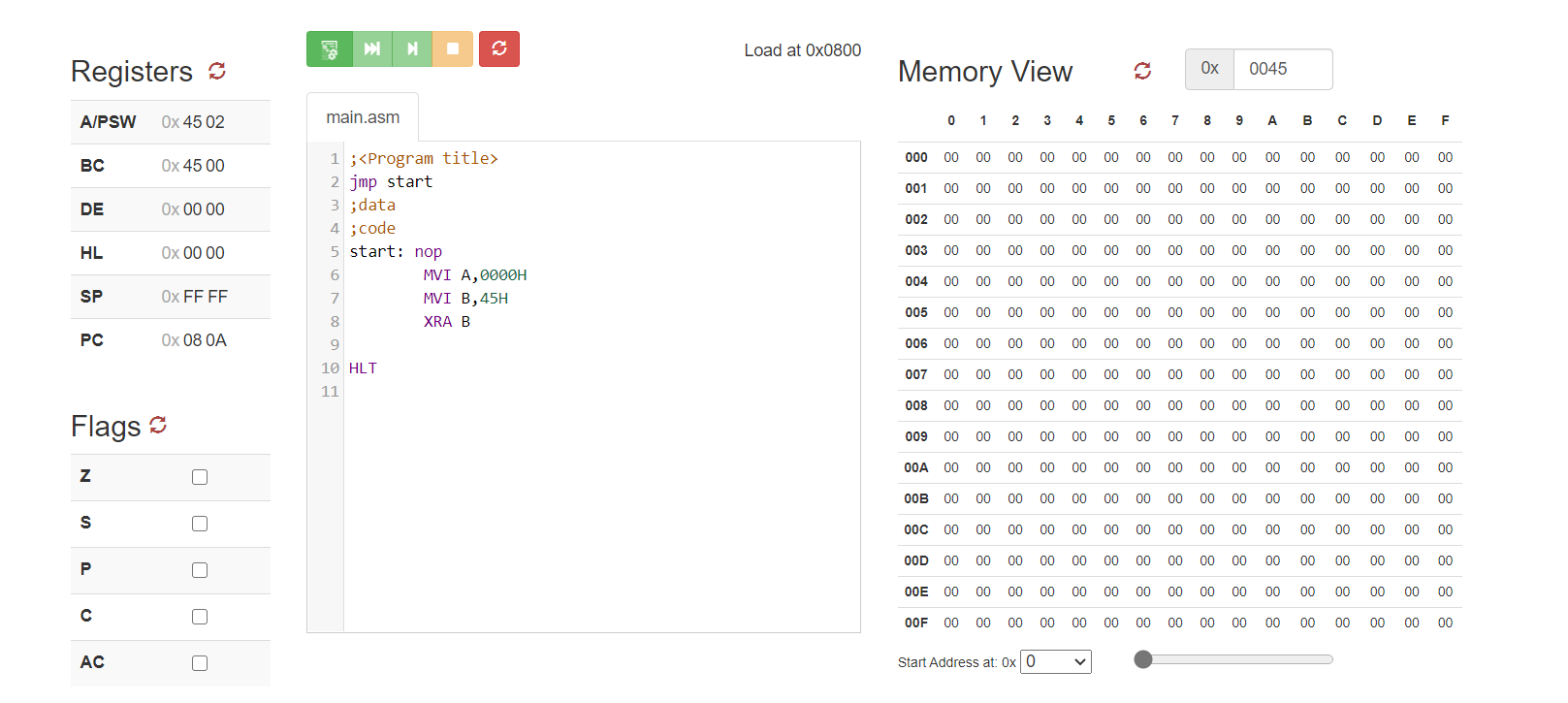


**3) XRA R, XRA M, XRI 67H**

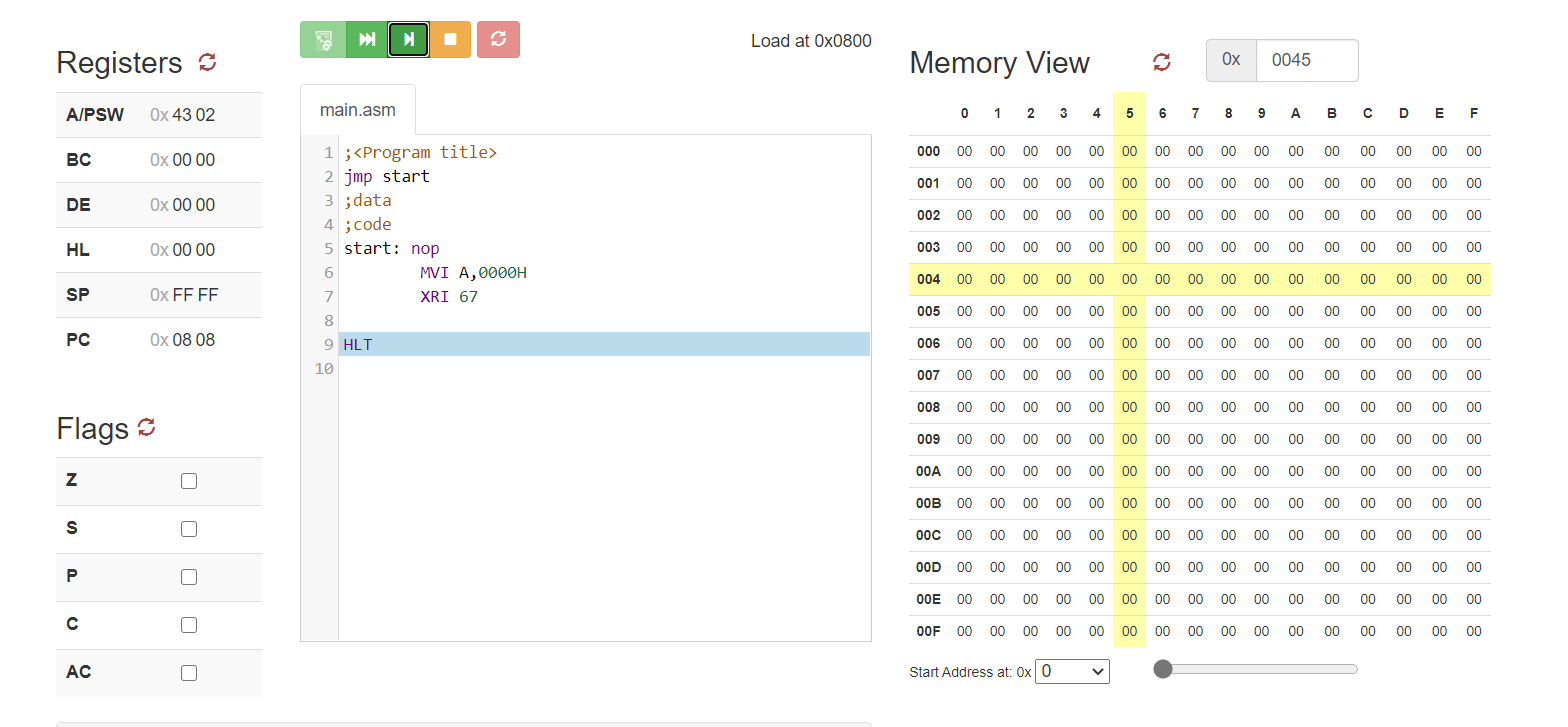
* **XRA M**



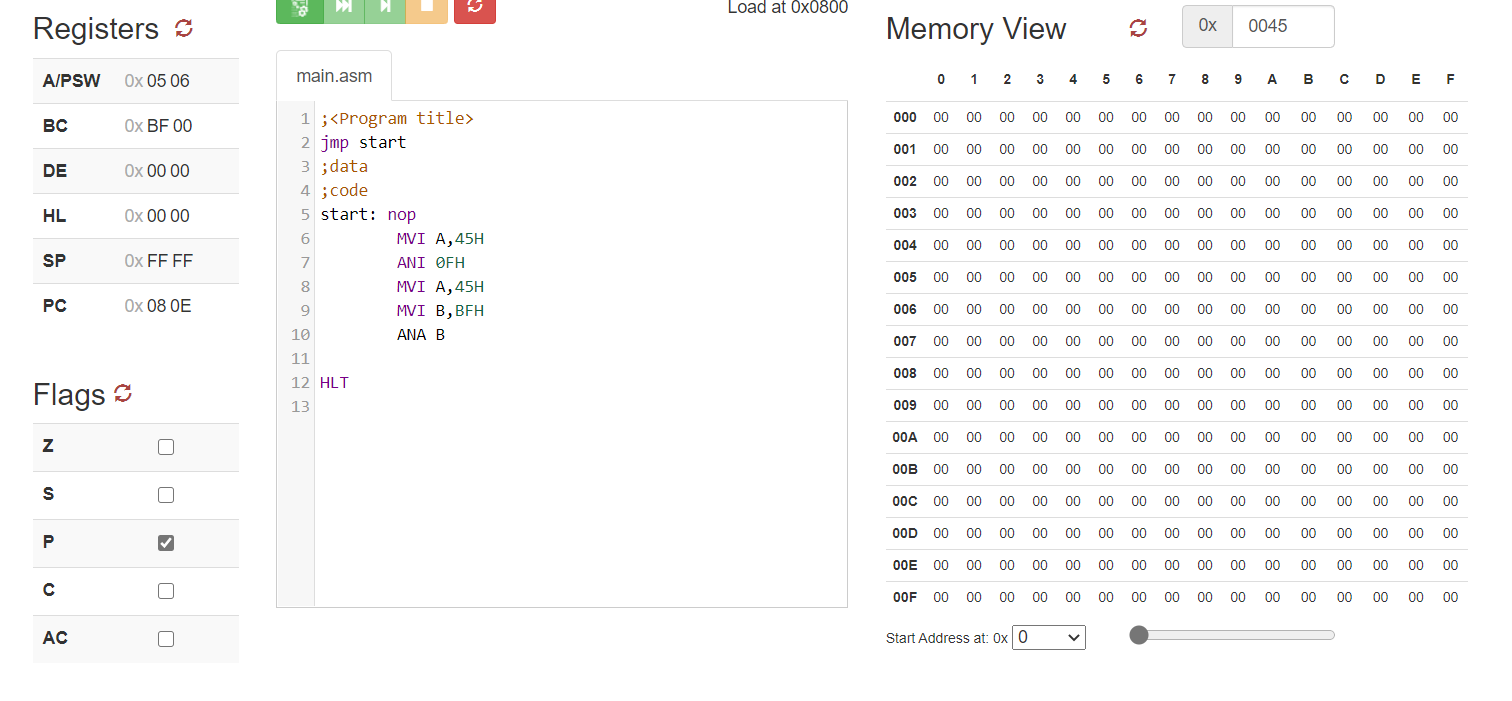
**XRA R**



* **XRI 67H**



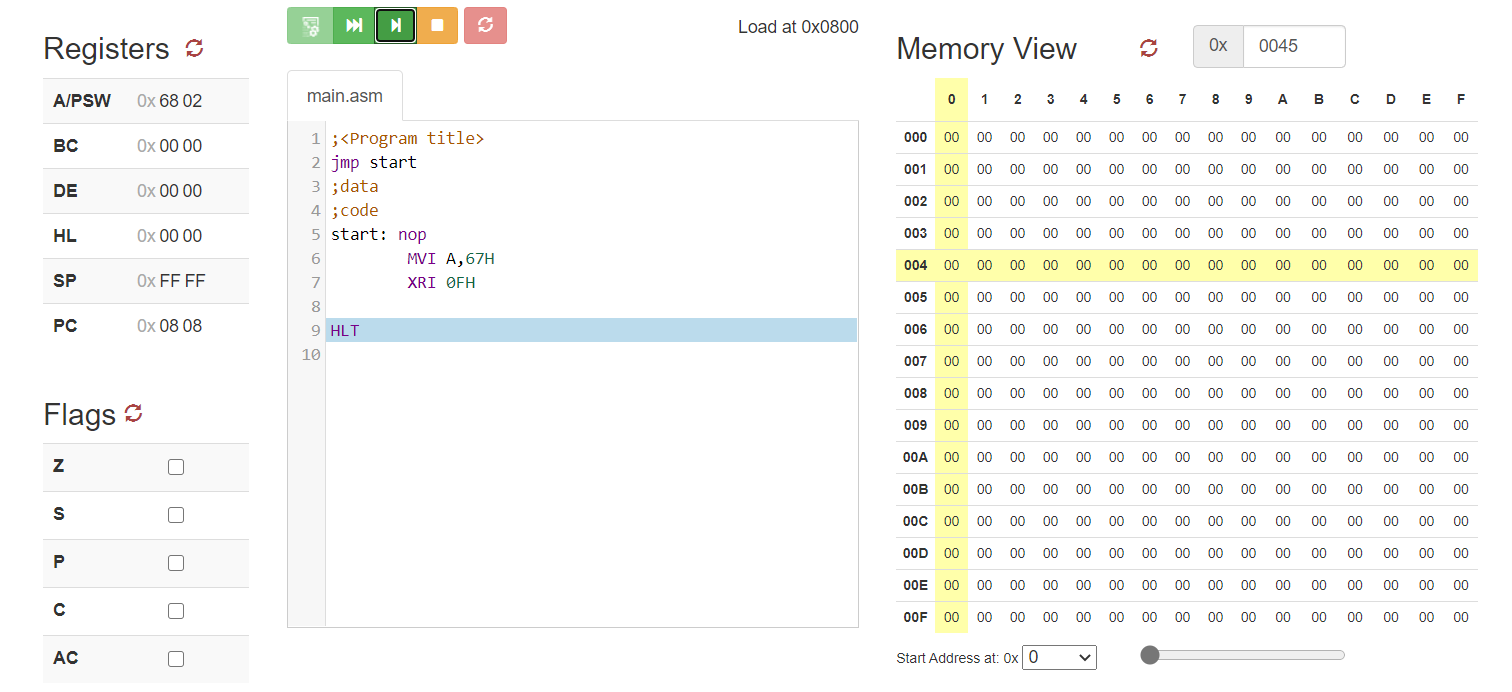
* **Write an 8085 program to clear the higher 4 bits of contents inside register A. Content of Register A = 45H. (ANI 0FH)**

****

* **Write an 8085 program to set the higher 3 bits of contents inside register A. Content of Register A = 27H. (ORI E0H)**

****

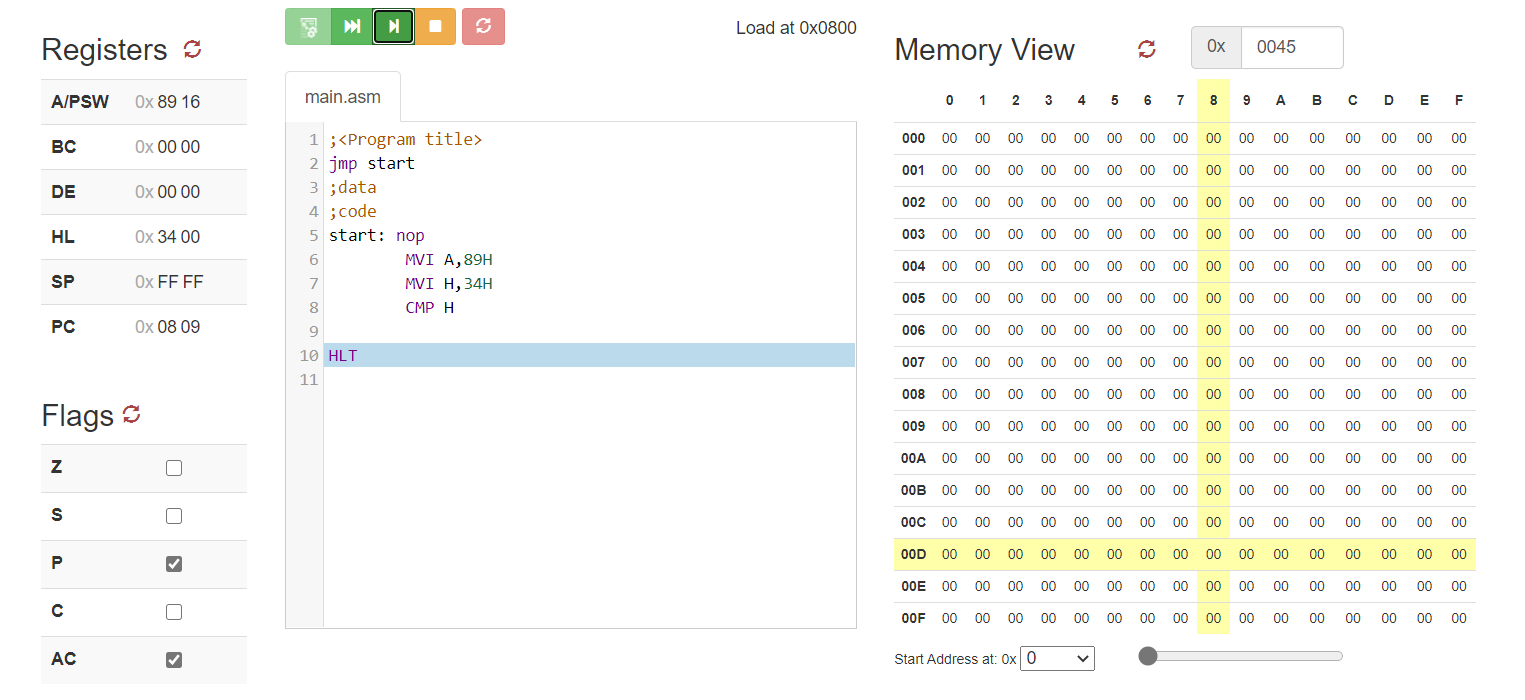
* **Write an 8085 program to complement the lower 4 bits of contents inside register A. Content of Register A = 67H. (XRI 0FH)**

****

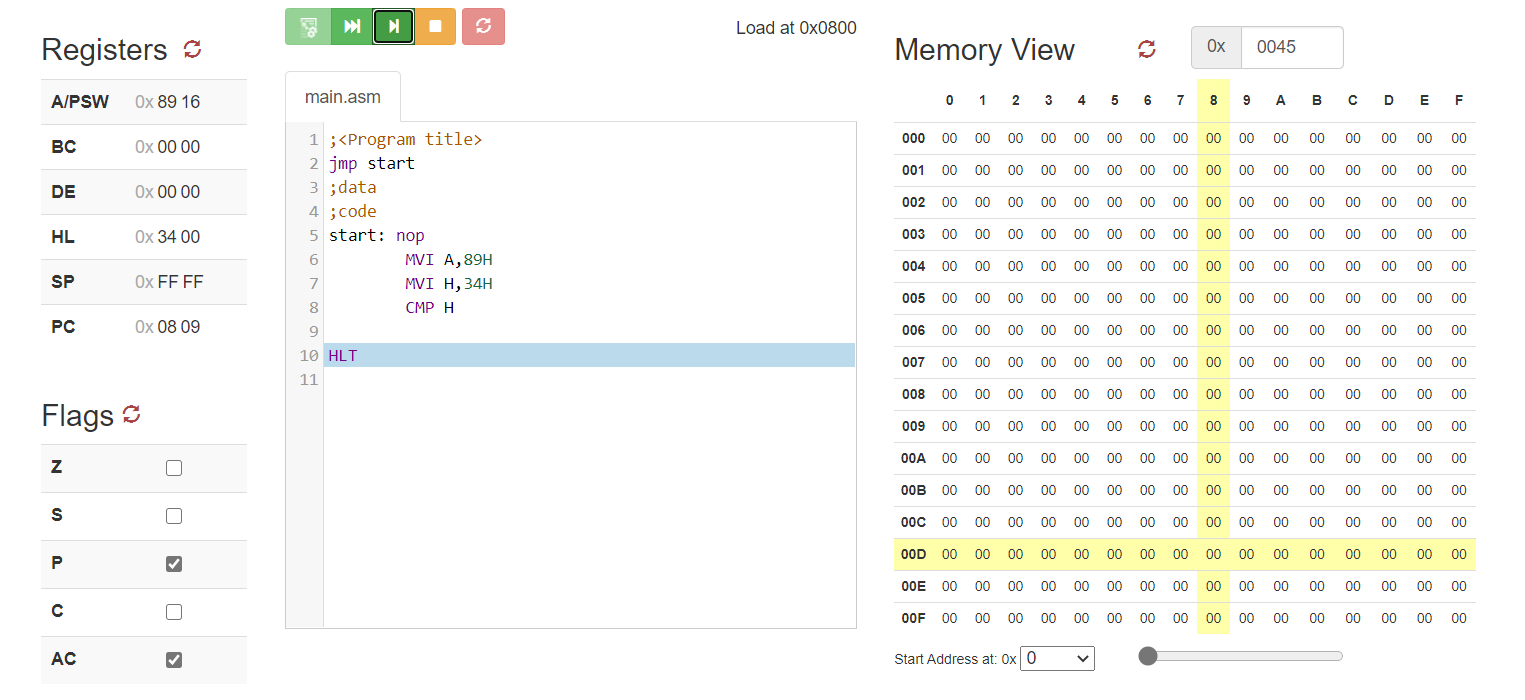
**Microprocessor & Interfacing**

**(Lab session-08)**

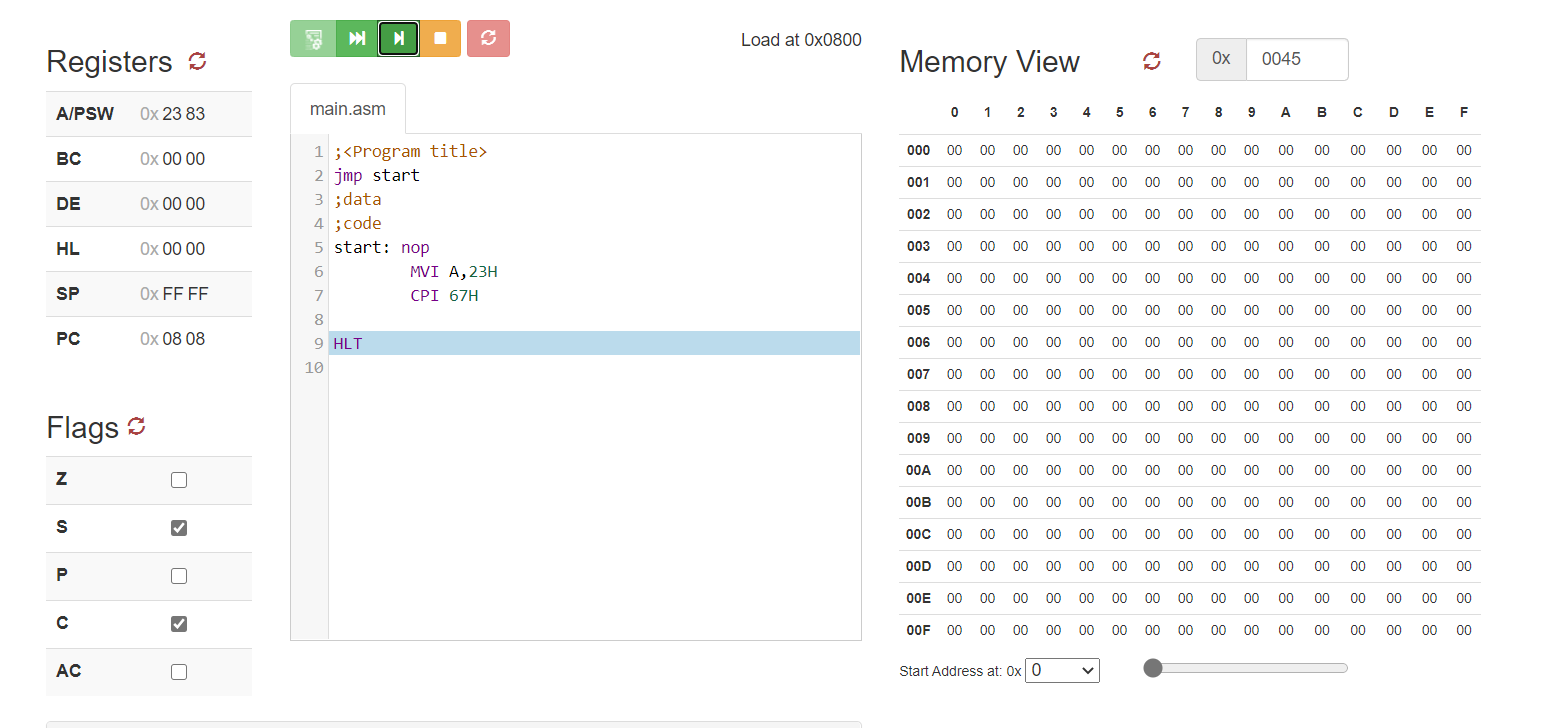
* **Demonstrate Compare instructions through 8085 programs**
* **Write an 8085 program to compare the contents of register A and H which are as follows: A=89H, H=34H. Also derive the conclusion by the end of this operation in reference to the contents of flag register.**

****

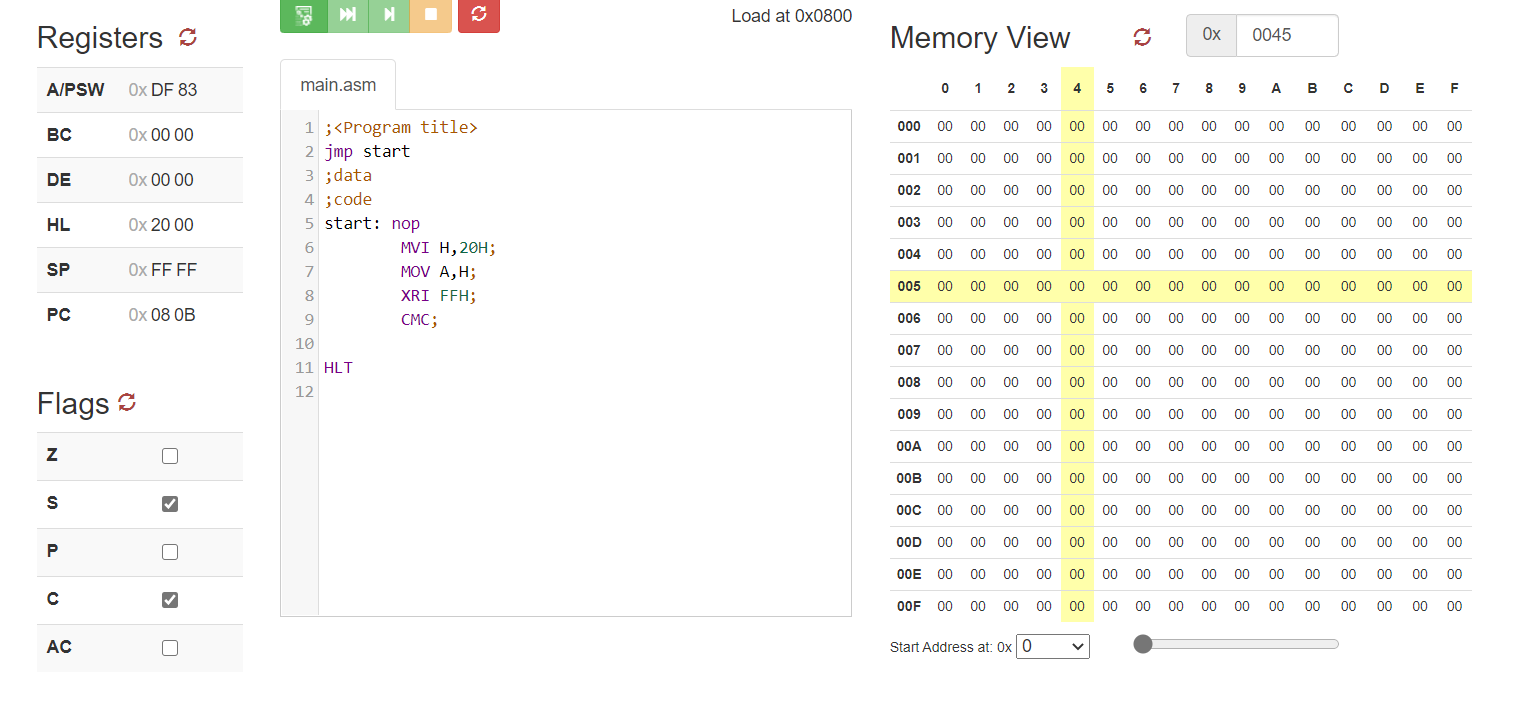
* **Write an 8085 program to compare the contents of register A and content at memory location 2436H which are as follows: A=5BH, [2436] = 5BH. Also derive the conclusion by the end of this operation in reference to the contents of flag register.**

****

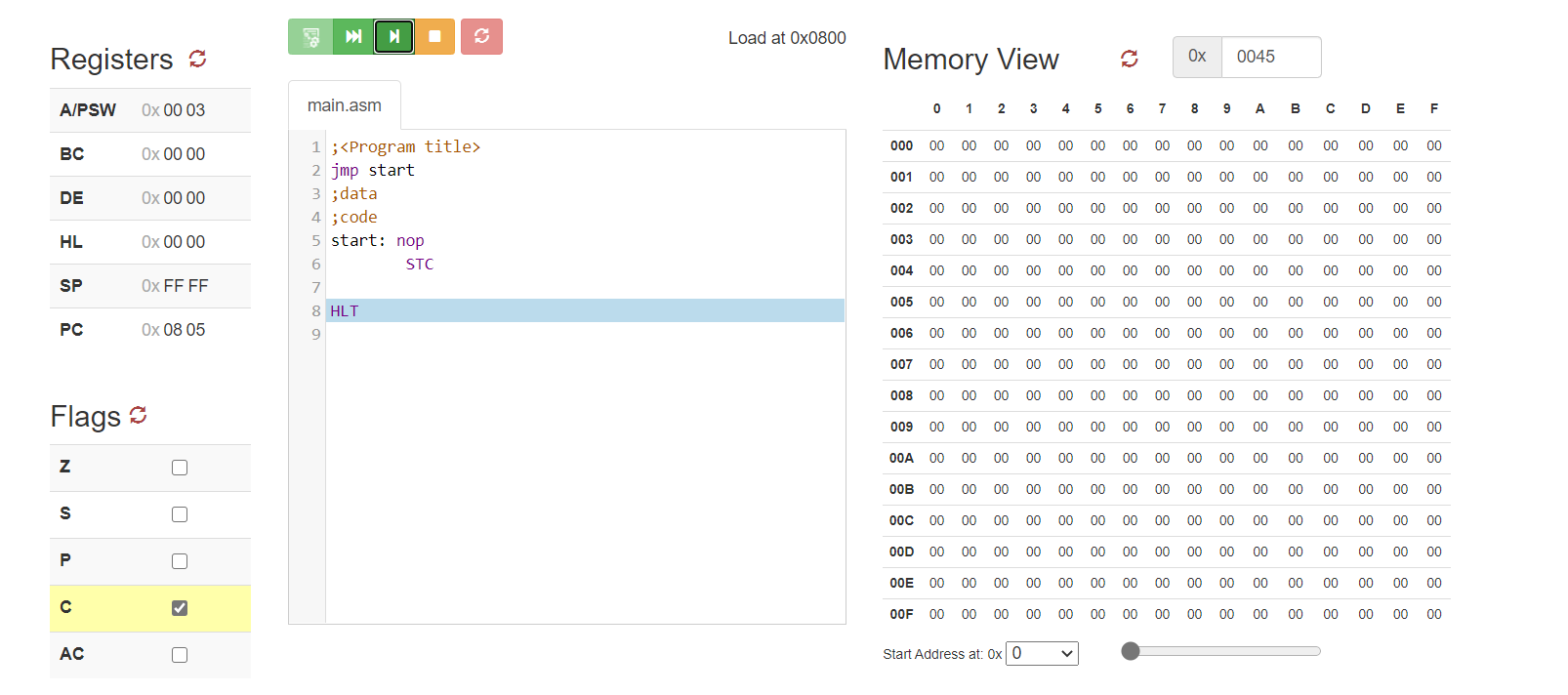
* **Write an 8085 program to compare 67H with the contents of Register A=23H. Also derive the conclusion by the end of this operation in reference to the contents of flag register.**

****

* **Write an 8085 program to complement the current content of Register H and Carry Flag (CY).**

****

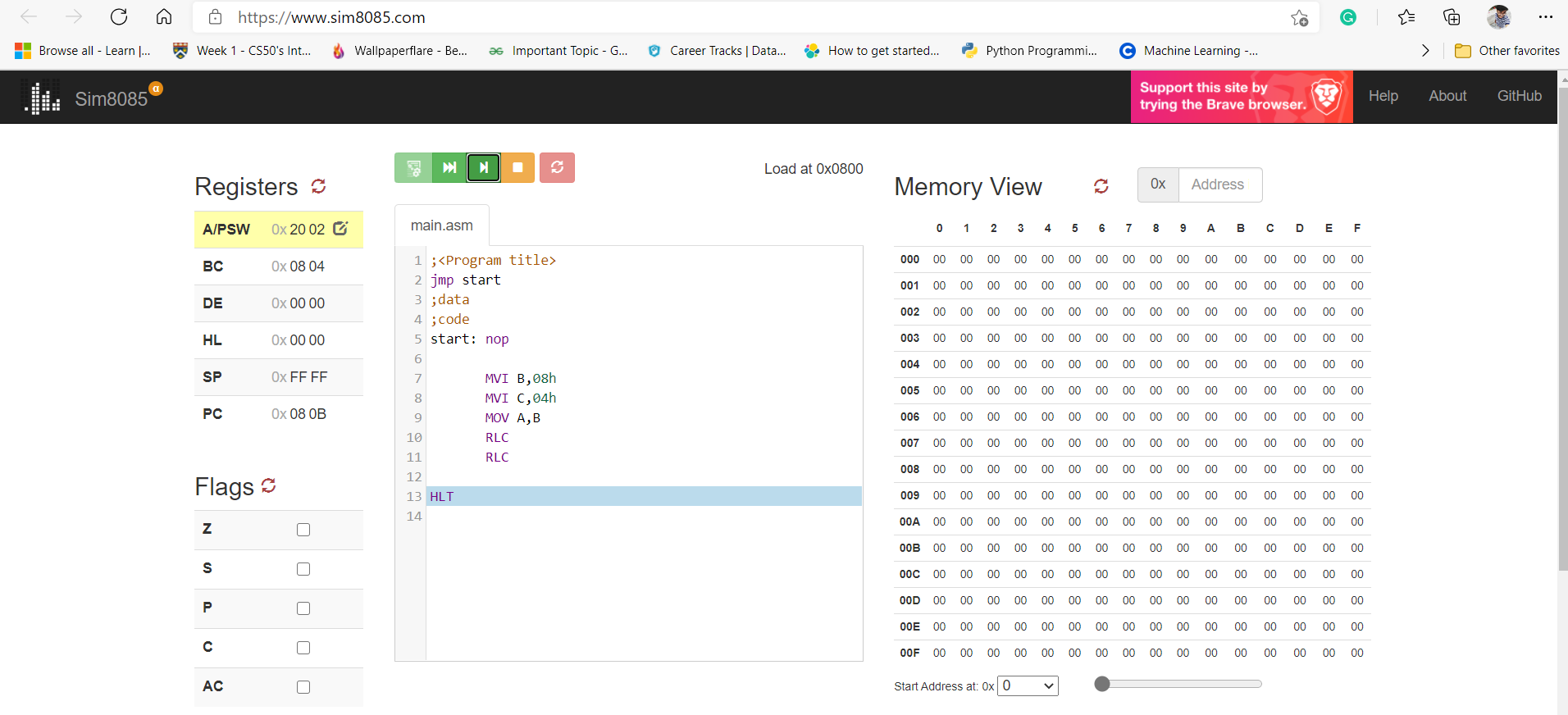
* **Write an 8085 program to set the carry flag**

****

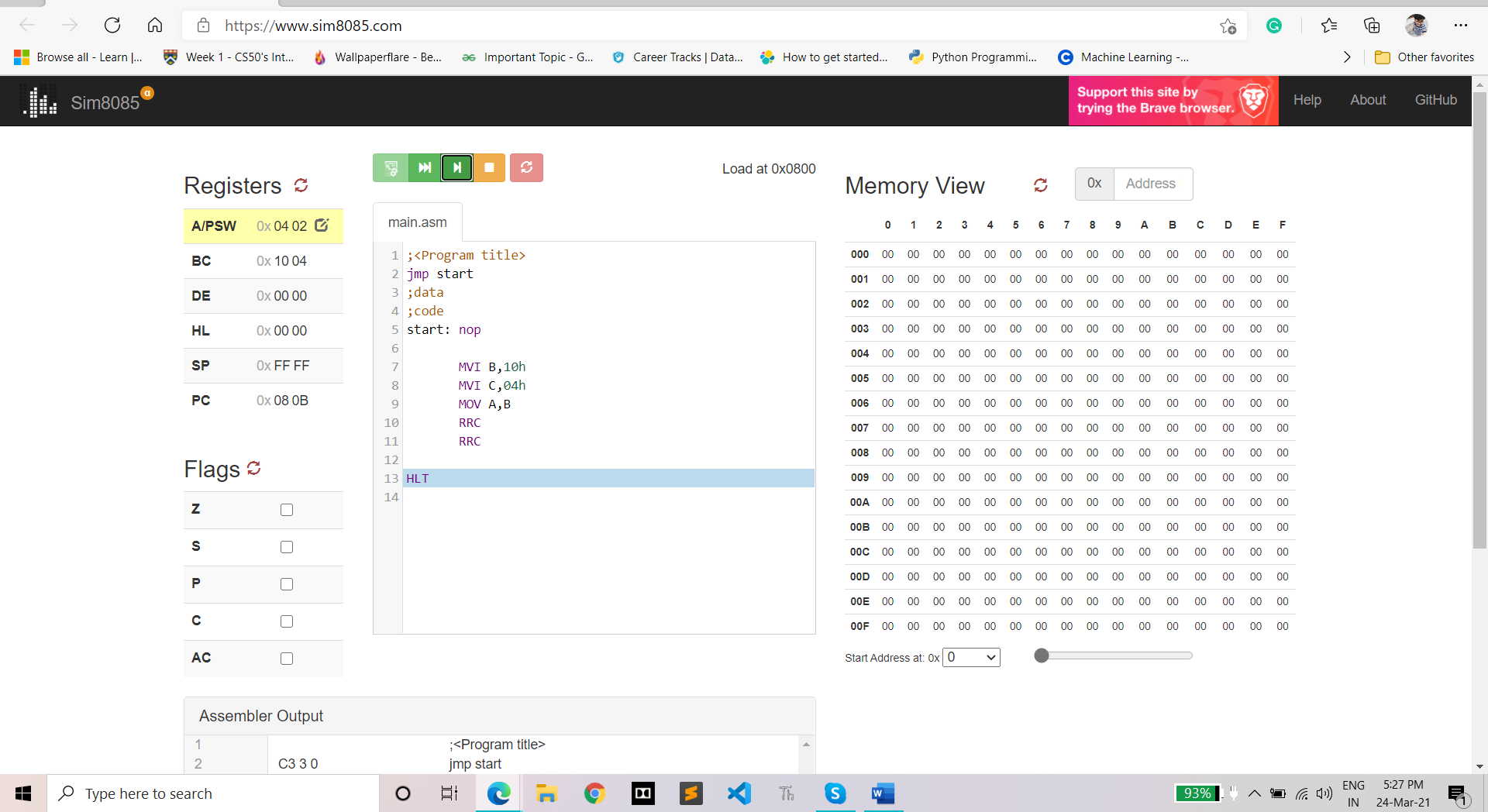
**Microprocessor & Interfacing**

**(Lab session-09)**

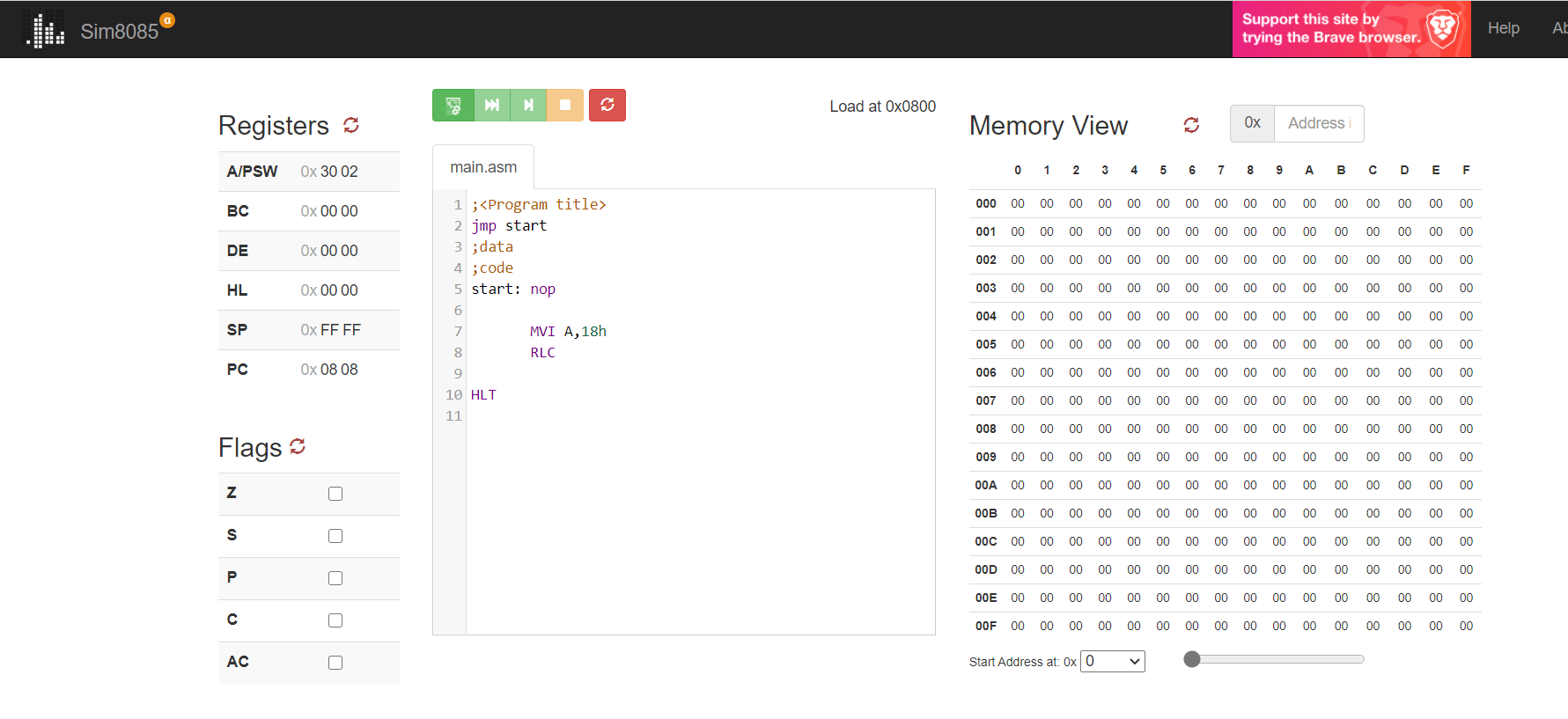
1. **Write an 8085 program to perform multiplication of the contents of Register B with the contents of Register C using logical instructions. The contents of B=08H and C=04H.**



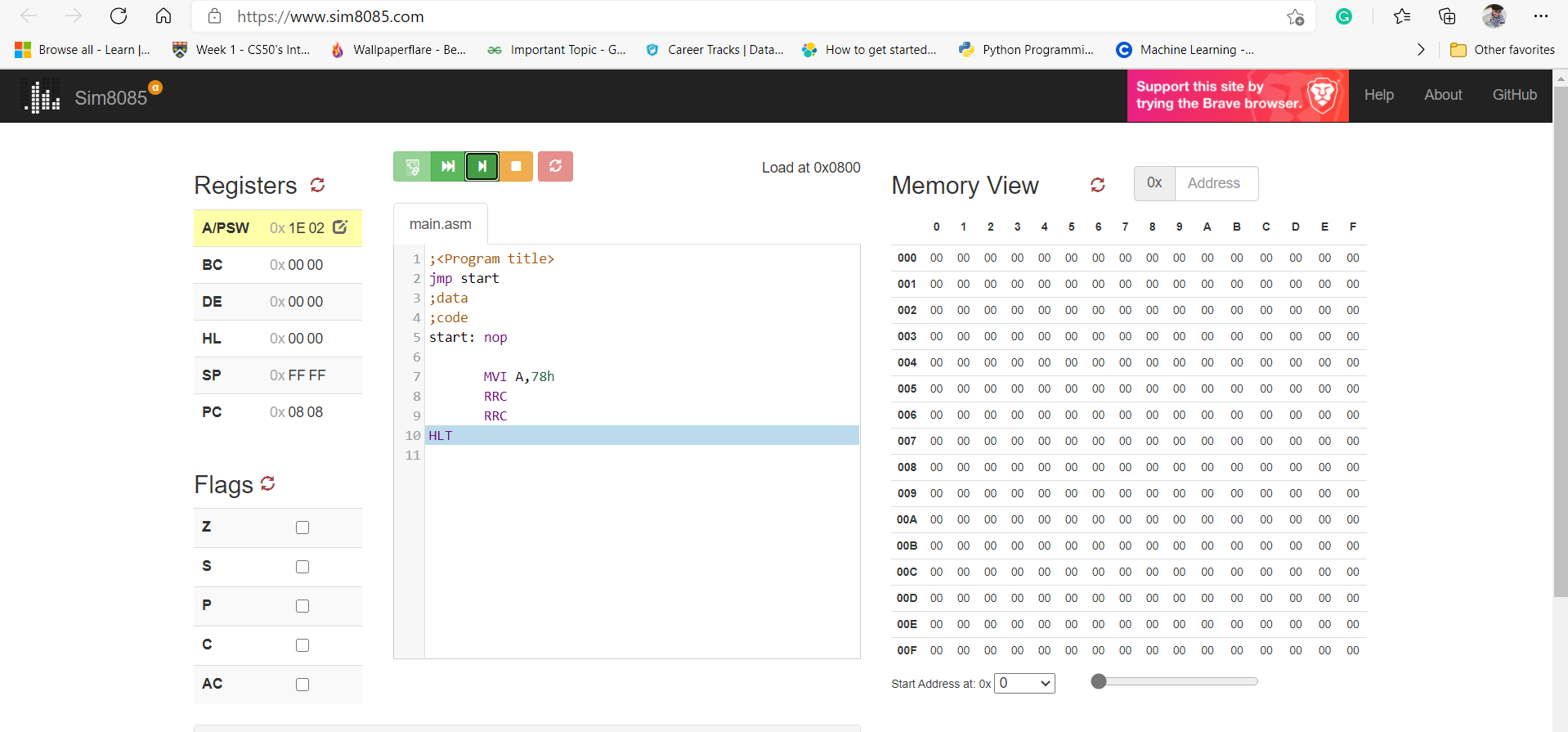
1. **Write an 8085 program to perform division of the contents of Register B by the contents of Register C using logical instructions. The contents of B=10H and C=04H.**



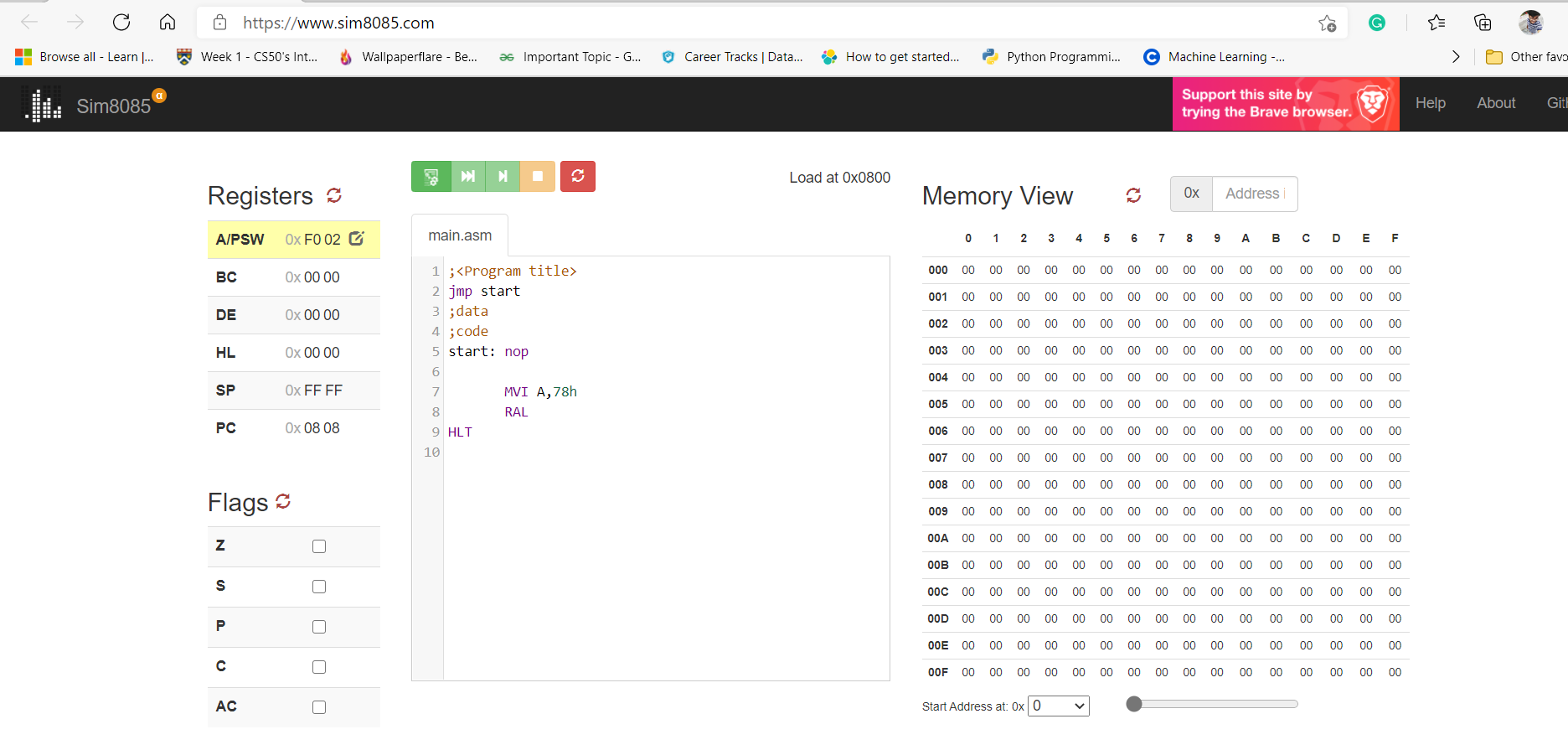
1. **Calculate the decimal value of the number in the accumulator before and after the Rotate instructions are executed and explain the mathematical functions performed by the instructions**
2. **MVI A 18H RLC**



**b) MVI A, 78H RRC RRC**



1. **Write an 8085 program to demonstrate the rotate left through carry flag instruction. Assume A=78H and CY=0**



1. **Write n 8085 program to demonstrate the rotate right through carry flag instruction. Assume B=34H and CY=1**



**Microprocessor & Interfacing**

**(Lab session-10)**

1. **Write 8085 program to call to a label INCR. Perform increment operation for the contents of Register A at label INCR and get back to main program and move the new value to Register B and then halt the program.**

**➢ Program:**

MVI A,27H; MOV IMMEDEATE 27H INTO ACCUMULATOR

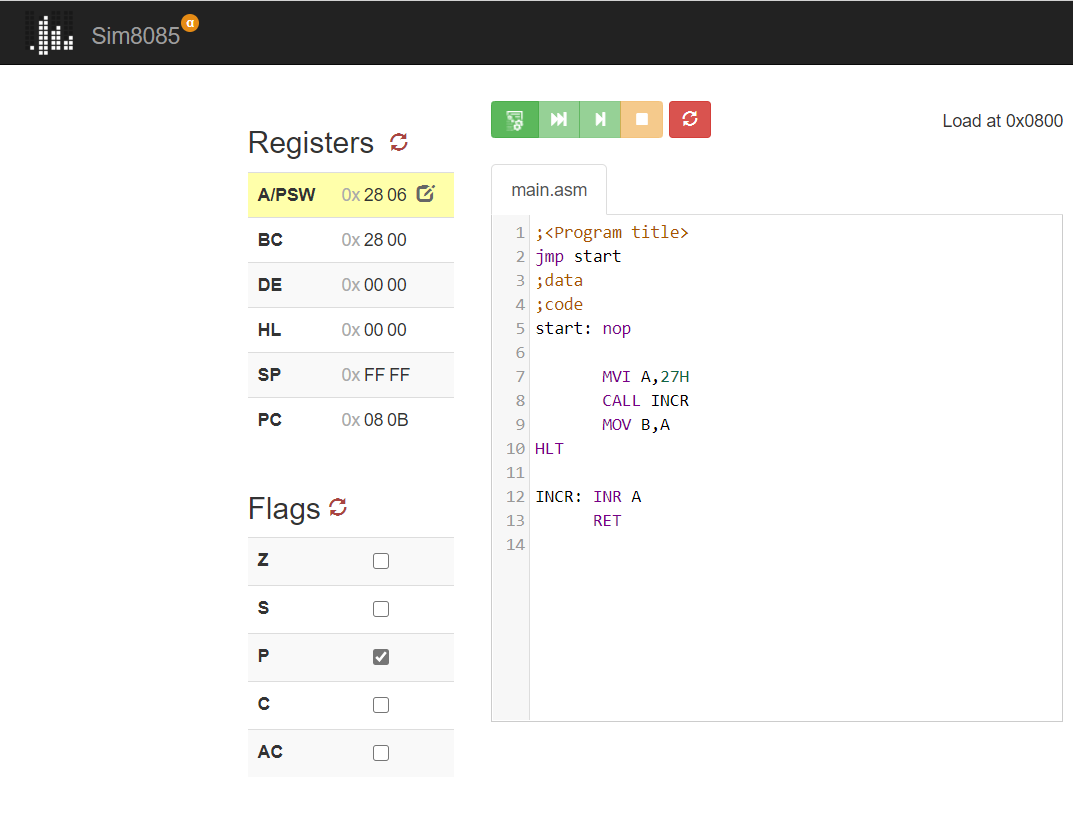
CALL INCR; CALL LABEL INCR LABEL

MOV B, A; MOV CONTENT OF A INTO B

HLT

INCR: INR A; INCREMENT THE ACCUMULATOR

RET; RETURN BACK TO MAIN PROGRAM



**2) Write 8085 program to demonstrate the working of given list of conditional call instructions:**

**1-> CC - carry=1**

➢ **Program:**

INITIALLY VALUE OF ACCUMULATOR IS 0

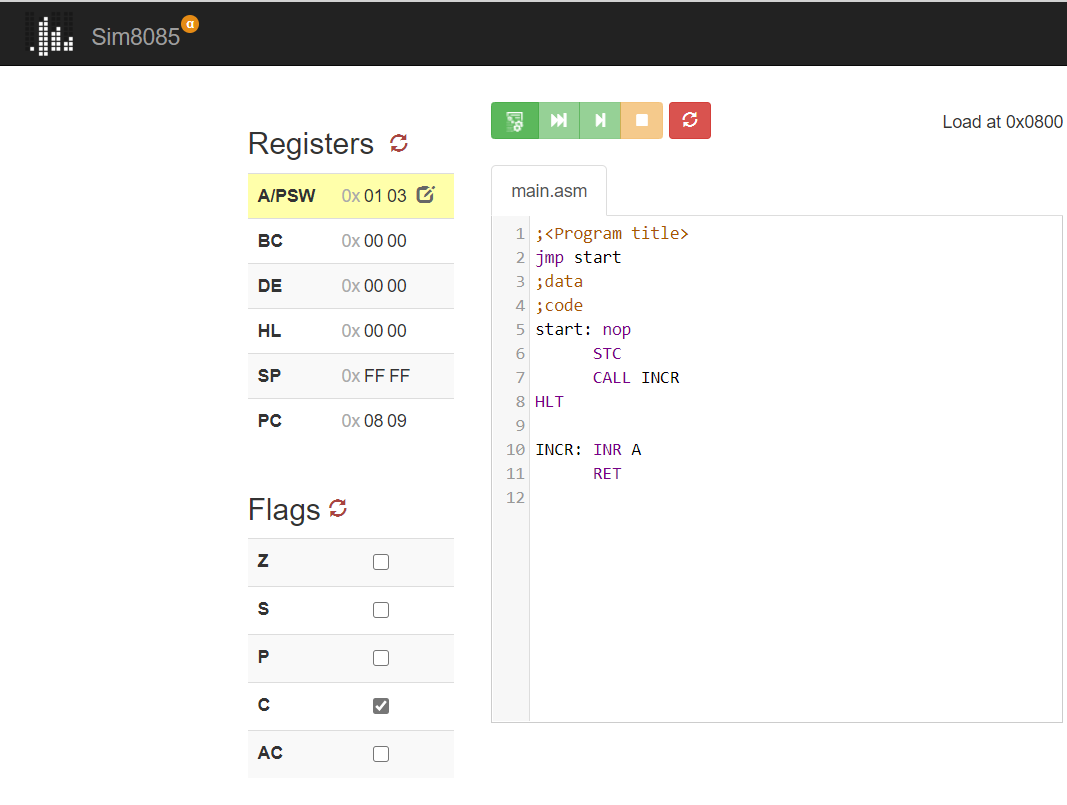
STC; SET CARRY FLAG

CC INCR; CALL INCR LABEL

HLT

INCR: INR A; INCREMENT THE CONTENT OF ACCUMULATOR

RET



**2-> CNZ - zero=0**

* **Program:**

; INITIALLY C CONTENT 0H

MVI B,26H; MOVE IMMIDIATE 26 IN REGISTER B

MVI A,27H; MOV IMMIDIATE 27 IN ACCUMULATOR

SUB B; SUBTRACT B FROM A..A <= A-B ;

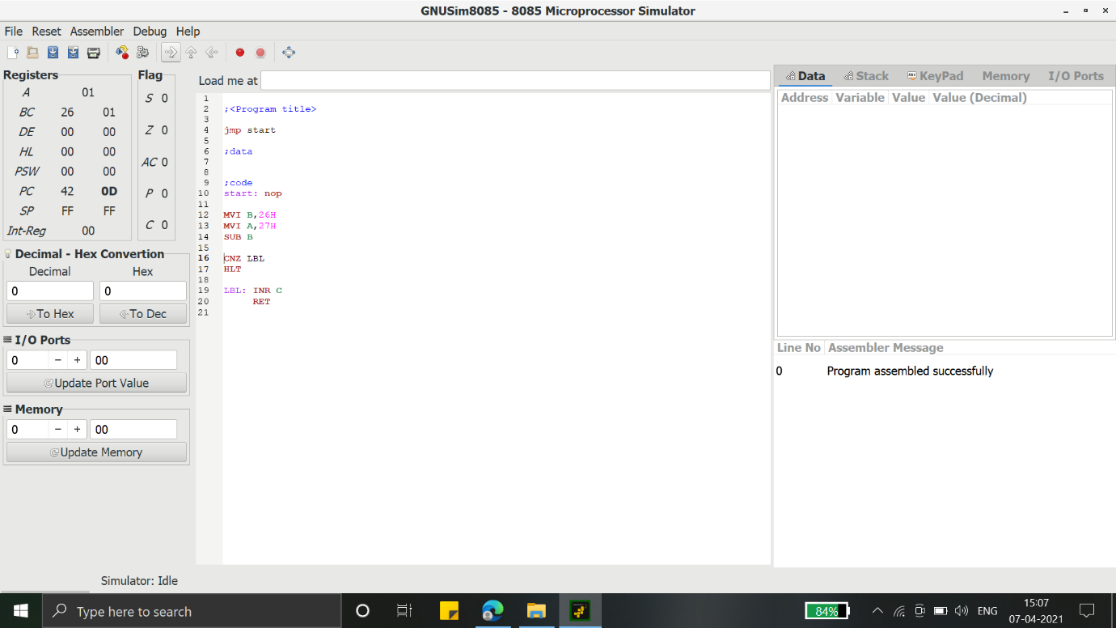
BUT HERE FINAL RESULT IS NOT ZERO MEANS Z WILL BE RESET

CNZ LABEL

HLT

LABEL: INR C

RET

****

**3 -> CPE – parity=1**

**Program:**

MVI A ,07H; MOV IMMMIDIATE 07H DATA INTO ACCUMULATOR

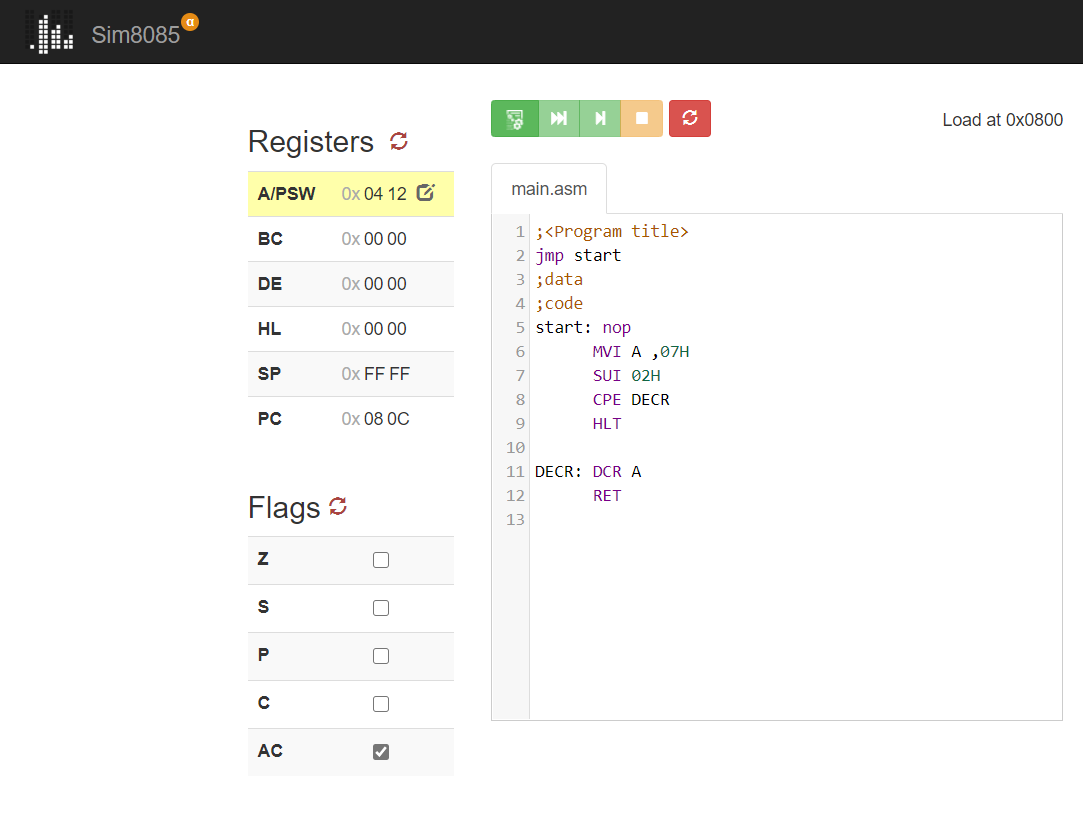
SUI 02H; A <= A - 02

CPE DECR; CALL DECREMENT LABEL WHEN PARITY FLAG IS SET

HLT

DECR: DCR A; DECREMENT THE VALUE OF ACCUMULATOR

RET; RETURN TO THE MAIN PROGRAM



**4 -> CP – sign=0**

**Program:**

MVI A, 27H; TRANSFER IMMIDIATE 27H DATA INTO A

MVI B, 20H; TRANSFER IMMIDIATE 20H DATA INTO B

SUB B; A <= A-B <= 7H;

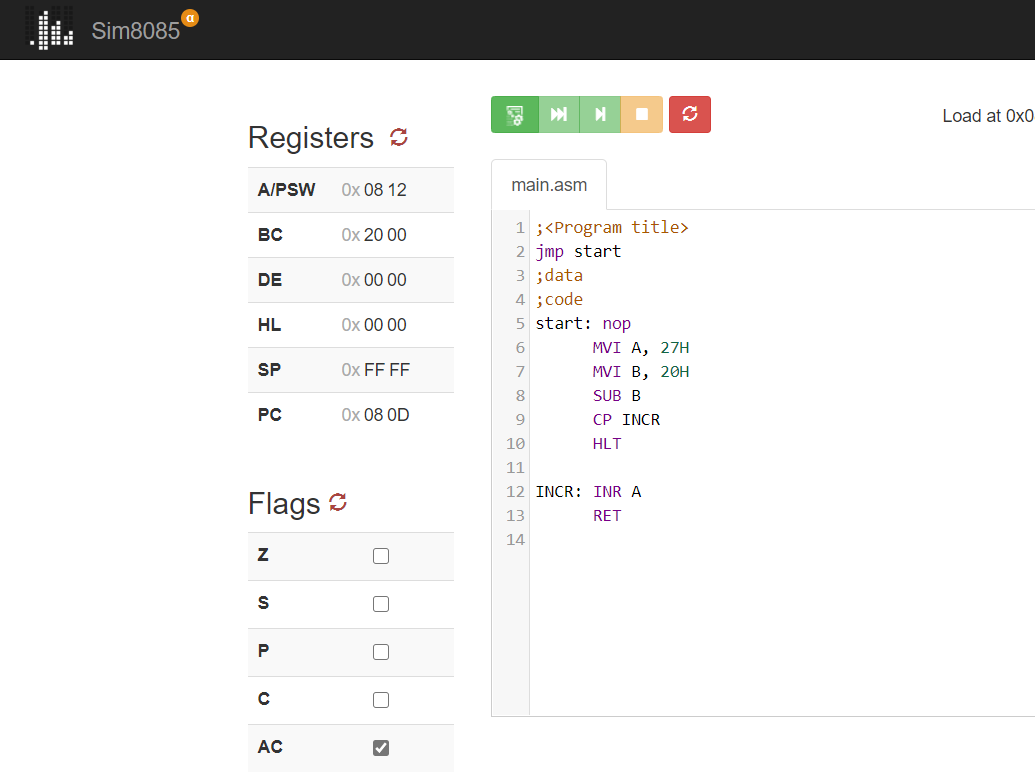
HERE FINAL RESULT WILL BE POSITIVE MEANS SIGN FLAG = 0

CP INCR; CALL INCR LABEL

HLT

INCR: INR A

RET



**5 -> RNC - carry=0**

**➢ Program:**

CALL START; CALL INCR LABEL

HLT

START: MVI A, 27H; TRANSFER IMMIDIATE 27H DATA INTO A

MVI B , 20H ; TRANSFER IMMIDIATE 20H DATA INTO B

SUB B; A <= A-B <= 7H;

HERE FINAL RESULT WILL NOT BE GENERATE ANY CARRY BIT;

SO, C = 0

RNC; RETURN TO MAIN PROGRAM ON NO CARRY (C = 0)



**6 -> RZ - zero=1**

**➢ Program:**

CALL START; CALL INCR LABEL

HLT

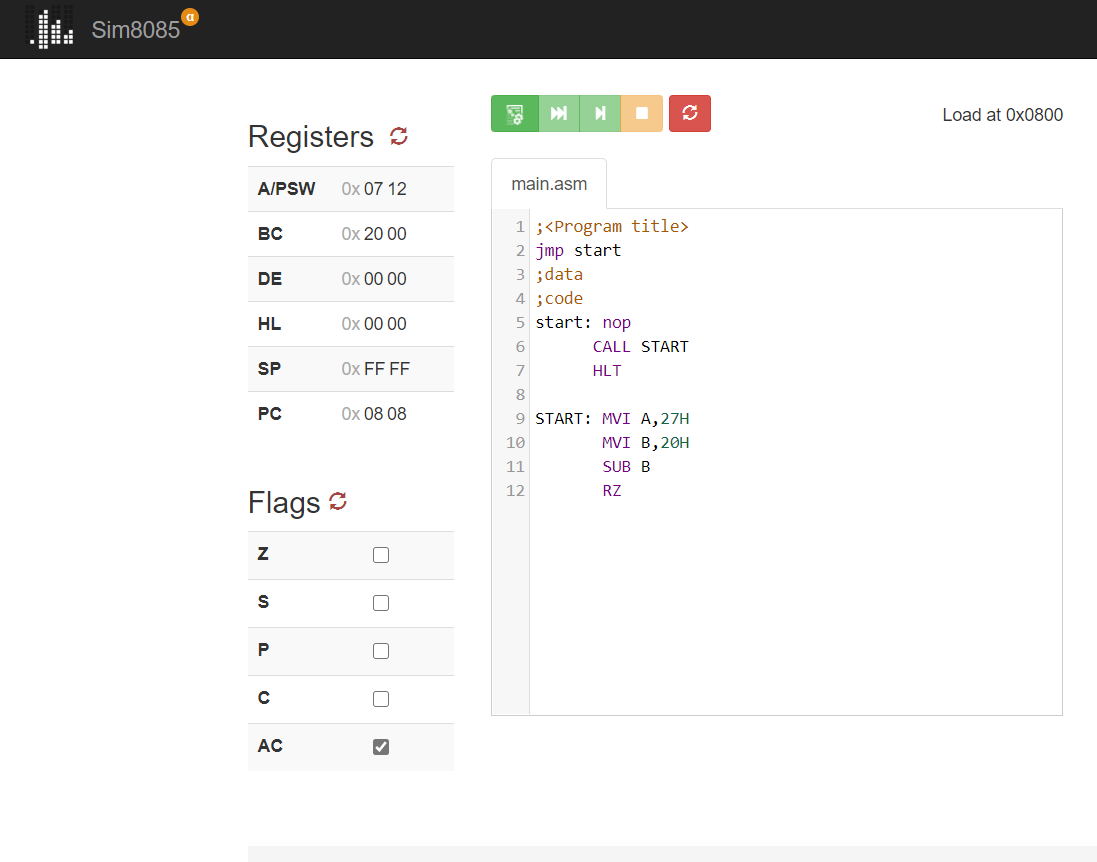
START: MVI A, 20H; TRANSFER IMMIDIATE 27H DATA INTO A

MVI B, 20H; TRANSFER IMMIDIATE 20H DATA INTO B

SUB B; A <= A-B <= 0H;

HERE FINAL RESULT WILL BE ZERO MEANS Z=1

RZ; RETURN TO MAIN PROGRAM ON NO CARRY (Z = 0)



**6 -> RPO - parity=0**

**➢ Program:**

CALL START; CALL INCR LABEL

HLT

START: MVI A, 027H; TRANSFER IMMIDIATE FFH DATA INTO A

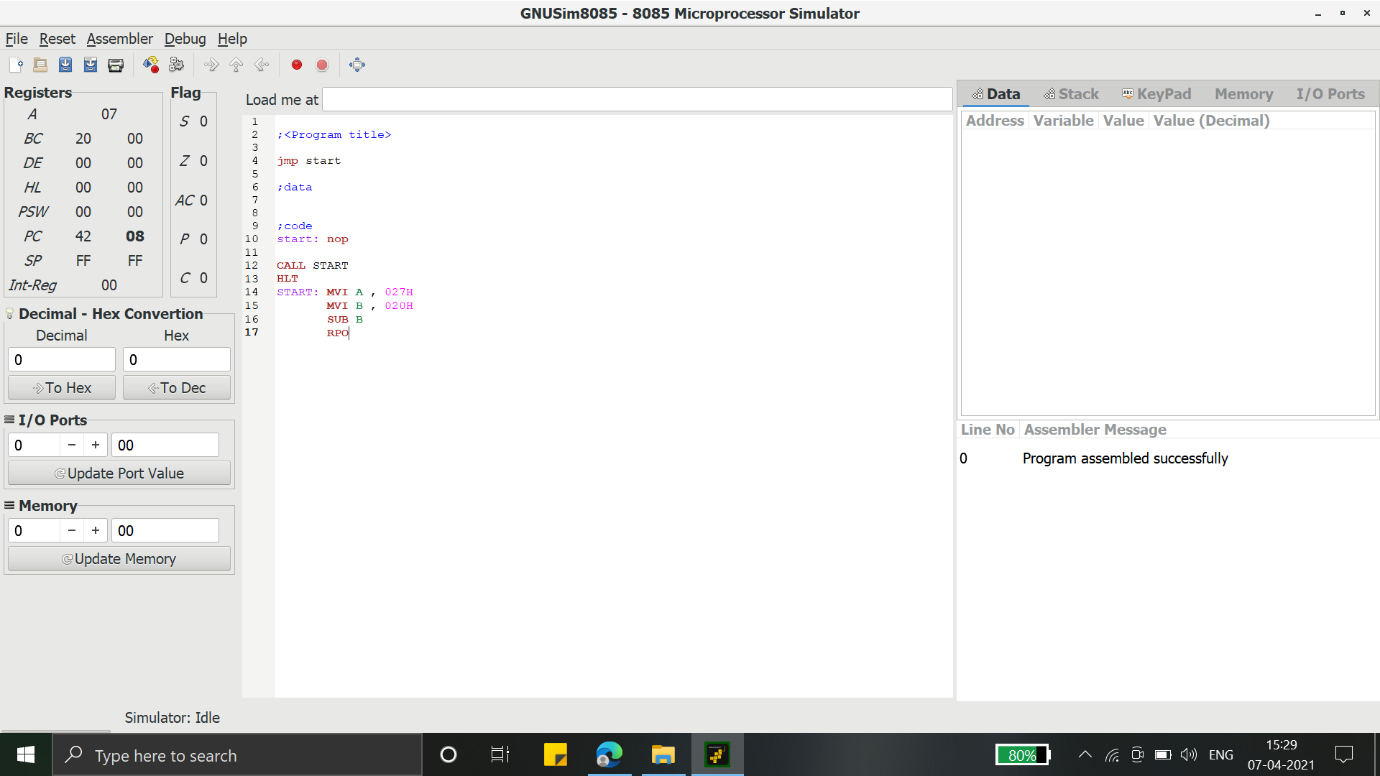
MVI B, 020H; TRANSFER IMMIDIATE FFH DATA INTO B

SUB B; A <= A-B <= 7H;

HERE FINAL RESULT WILL BE 7H IT WILL NOT BE GENETATED;

ANY PARITY BIT MEANS P =0

RPO; RETURN TO MAIN PROGRAM ON NO PARITY BIT (P = 0)

****

**7 -> RM - sign=1**

**➢ Program:**

CALL START; CALL INCR LABEL

HLT

START: MVI A, 20H; TRANSFER IMMIDIATE FFH DATA INTO A

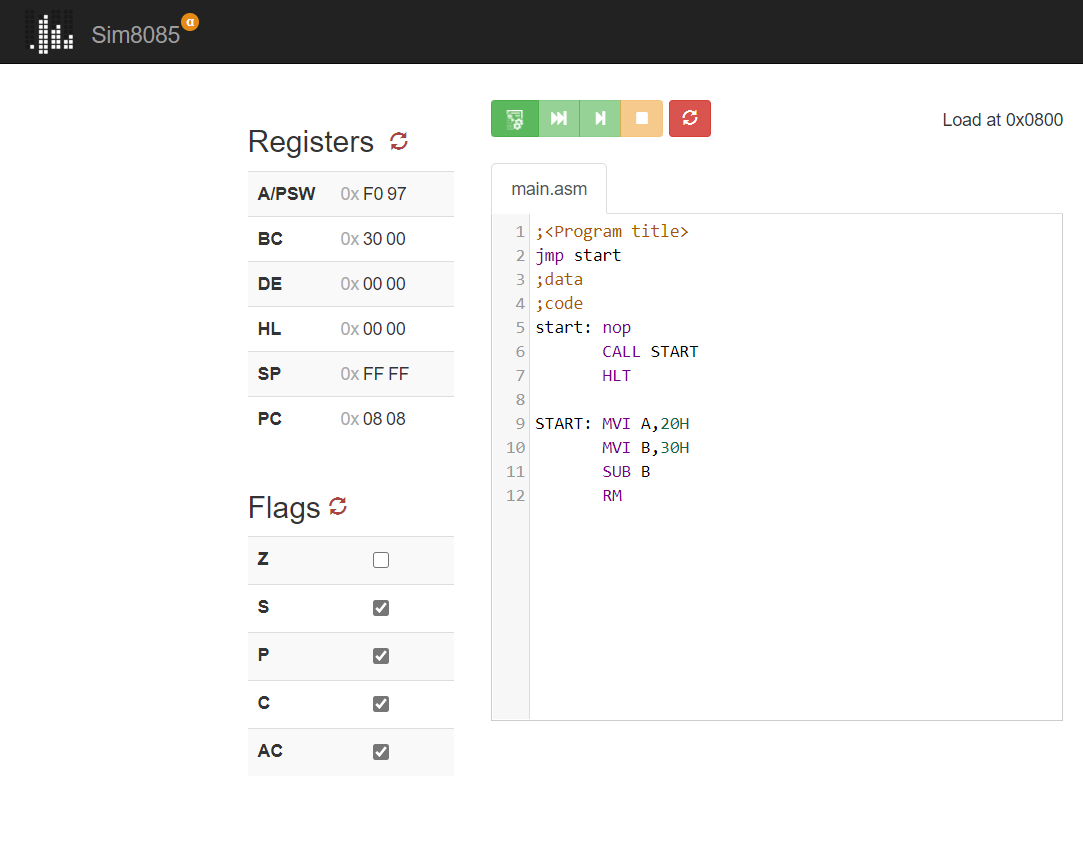
MVI B,30H; TRANSFER IMMIDIATE FFH DATA INTO B

SUB B; A <= A-B <= -1H;

HERE FINAL RESULT WILL BE NEGATIVE;

SO, S = 1 (SET)

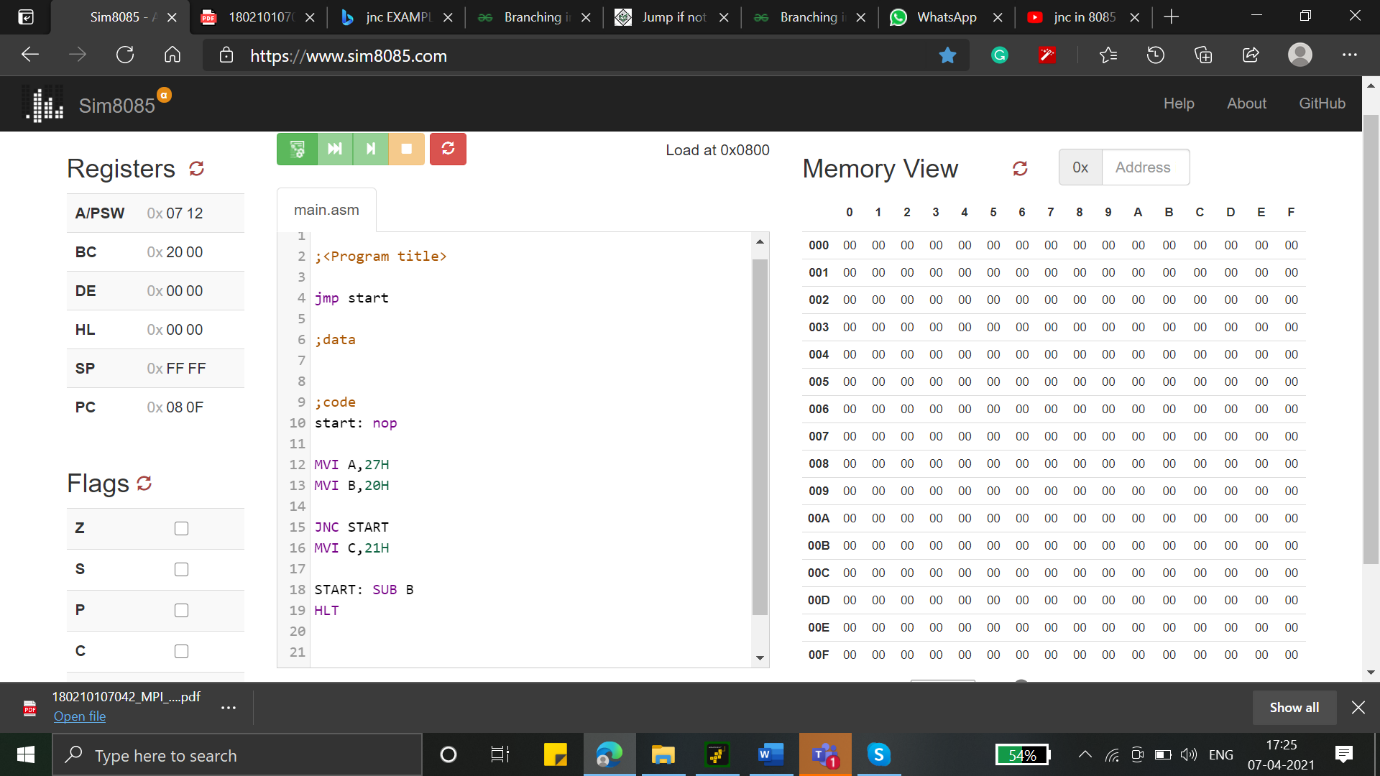
RM; RETURN TO MAIN PROGRAM ON SIGN BIT (S = 1)



**Microprocessor & Interfacing**

**(Lab session-11)**

**Write an 8085 program to demonstrate the working of given list of conditional jump Instructions: JNC - carry=0 JZ - zero=1 JPE - parity=0 JM - sign=1.**

1. **JNC - carry=0**

**Program:**

MVI A,27H

MVI B,20H

JNC START

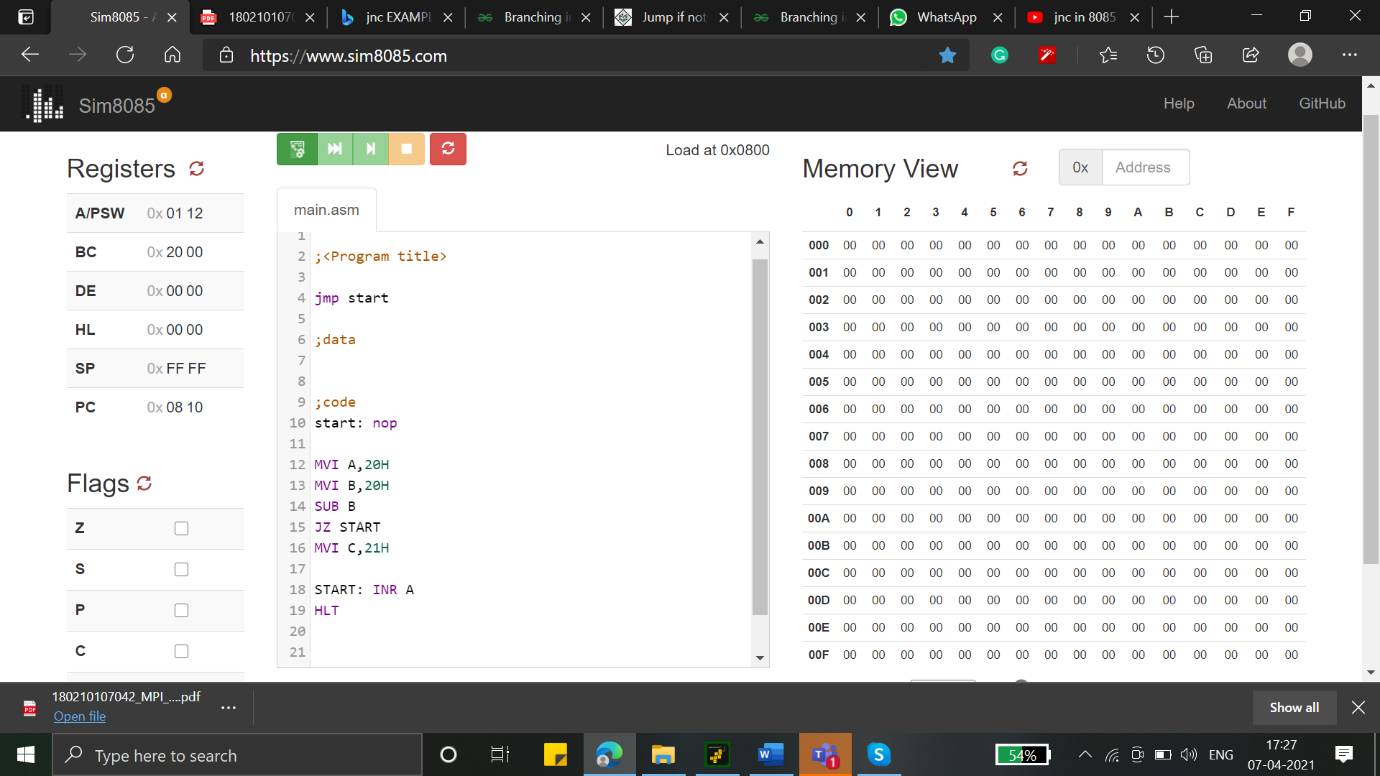
MVI C,21H

START: SUB B

HLT

1. **JZ - zero=1**

**Program:**



MVI A,20H

MVI B,20H

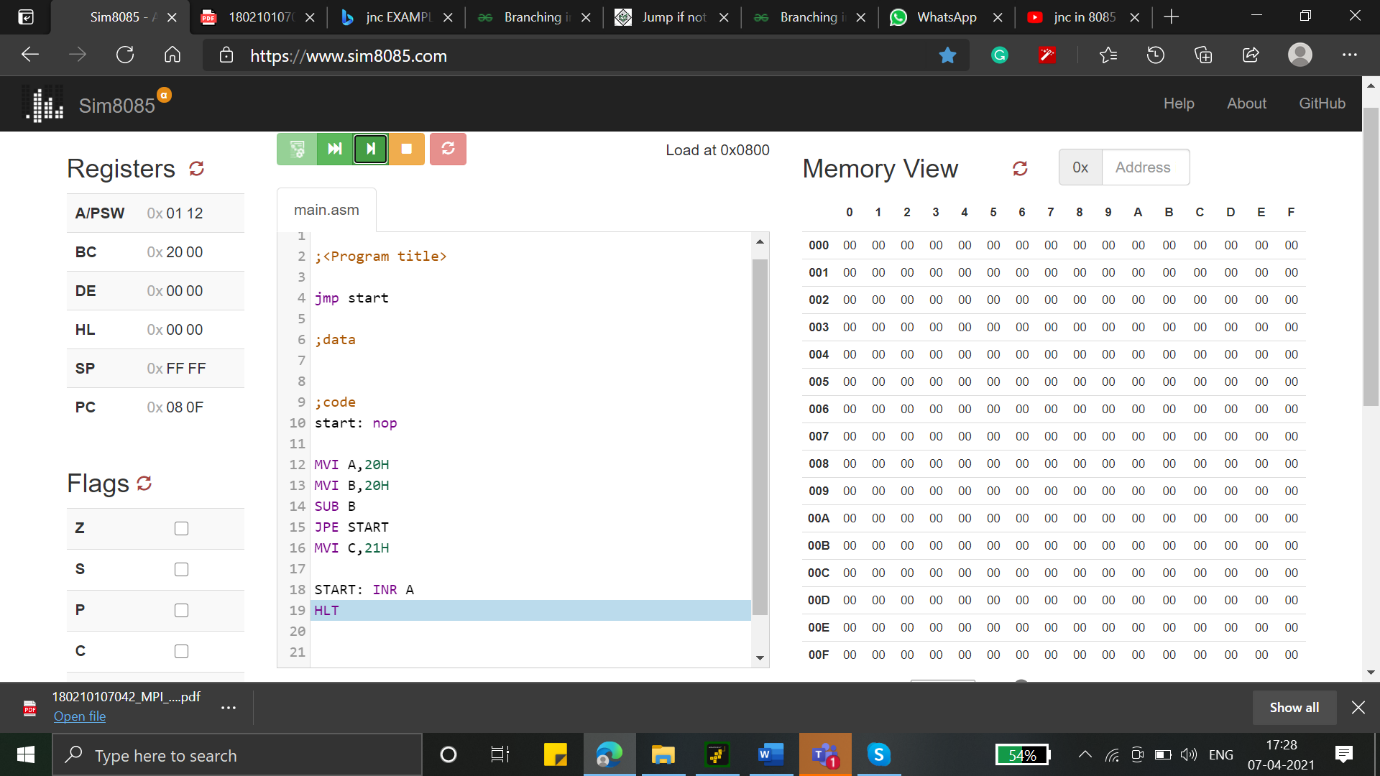
SUB B

JZ START

MVI C,21H

START: INRA

HLT

1. **JPE - parity=0**

**Program:**

MVI A,20H

MVI B,20H

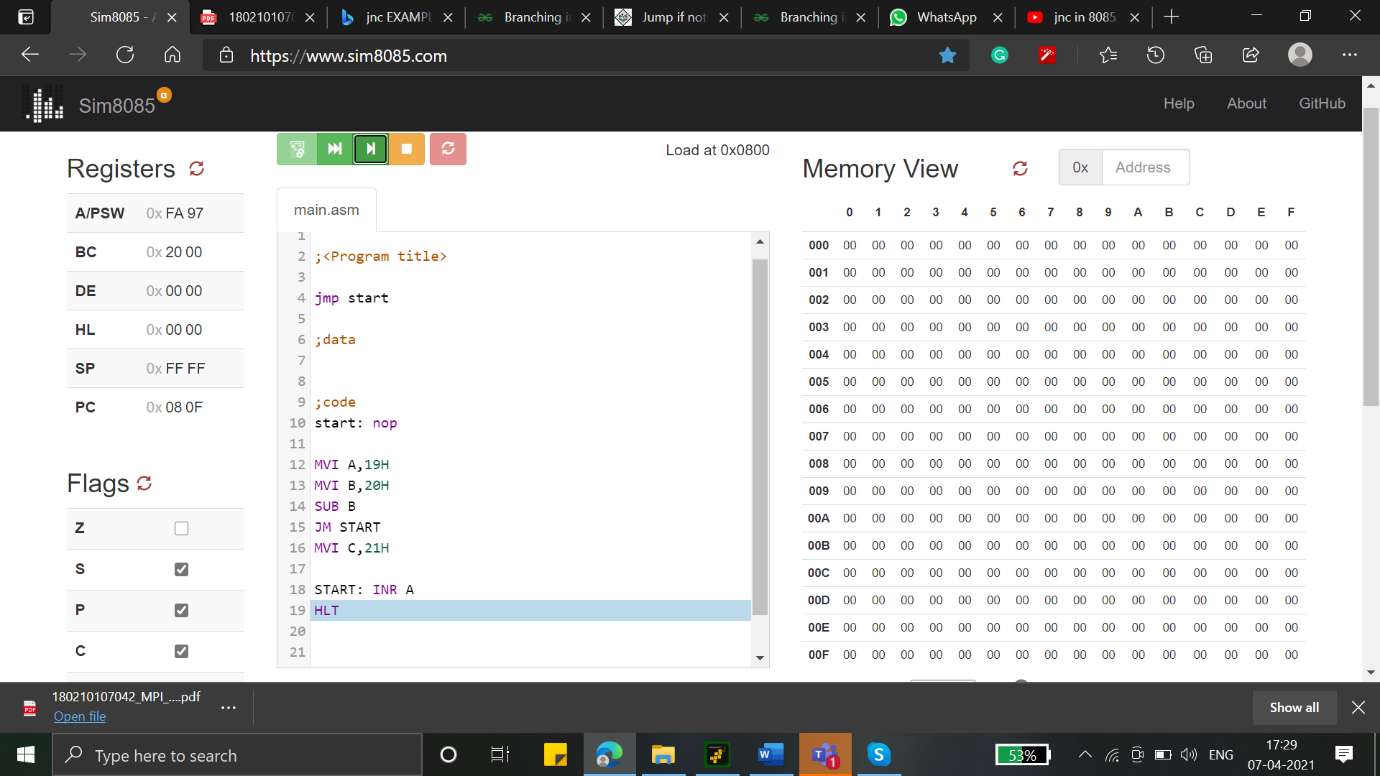
SUB B

JPE START

MVI C,21H

START: INRA

HLT

1. **JM - sign=1.**

**Program:**

MVI A,20H

MVI B,20H

SUB B

JM START

MVI C,21H

START: INRA

HLT