

19CSE303: Embedded Systems

LABSHEET 2: ASSEMBLY LANGUAGE PROGRAMMING OF 8085 MICROPROCESSORS

Name: Vinayak V Thayil

Roll Number:AM.EN.U4CSE21161

1. ONE'S COMPLEMENT OF AN 8-BIT NUMBER

LDA 5100H ; Load Address Of Number In H-L Register Pair (Lower byte data).

CMA ; Complement Accumulator

STA 5101H ; Store The Result

HLT ; Terminate Program

INPUT		OUTPUT	
Memory location	Data	Memory location	Data
5100H	56	5101H	199



2. 2'S COMPLEMENT OF AN 8-BIT NUMBER

LDA 5100H ; Load Address Of Number In H-L Register Pair (Lower byte data).

CMA ; Complement Accumulator

INR A ; Increment

STA 5101H ; Store The Result

HLT ; Terminate Program

INPUT		OUTPUT	
Memory location	Data	Memory location	Data
5100H	56	5101H	200

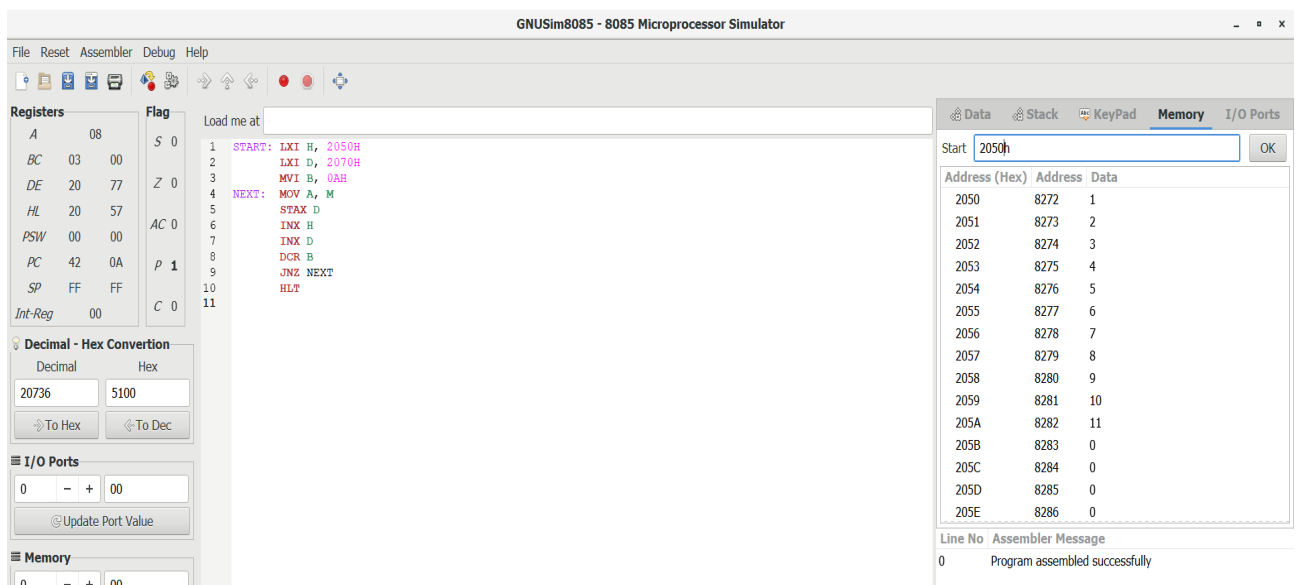
19CSE303: Embedded Systems



3. Show the output in 8085 simulator for the following programs. Attach the screenshot of final status of PC and relevant registers.

a) Copy 10 numbers stored from the location 2050H to the new location 2070H

```
START: LXI H, 2050H
      LXI D, 2070H
      MVI B, 0AH
NEXT: MOV A, M
      STAX D
      INX H
      INX D
      DCR B
      JNZ NEXT
      HLT
```



19CSE303: Embedded Systems

Data Stack KeyPad Memory I/O Ports		
Start	2070h	
Address (Hex)	Address	Data
2070	8304	1
2071	8305	2
2072	8306	3
2073	8307	4
2074	8308	5
2075	8309	6
2076	8310	7
2077	8311	8
2078	8312	9
2079	8313	10
207A	8314	0
207B	8315	0
207C	8316	0
207D	8317	0
207E	8318	0

b) 8-bit Addition without carry

;Addition of 2 numbers 15H and 10H

LDA 7000H

MOV B,A

LDA 7001H

ADD B

STA 7002H

HLT

The screenshot shows the 8085 assembly simulator interface. The 'Registers' panel on the left shows the A register containing 15 (hex 0F). The 'Flags' panel shows the Zero flag (Z) is set to 0. The 'Memory' panel on the right shows the program code stored at addresses 7000H to 7009H:

Address (Hex)	Address	Data
7000	28672	21
7001	28673	16
7002	28674	37
7003	28675	0
7004	28676	0
7005	28677	0
7006	28678	0
7007	28679	0
7008	28680	0
7009	28681	0

The 'Assembler Message' panel at the bottom shows: '0 Program assembled successfully'.

c) 8-bit Addition with carry

;Addition of 2 numbers 250 and 30

MVI C,00H

LDA 7000H

MOV B,A

LDA 7001H

ADD B

JNC ahead

INR C

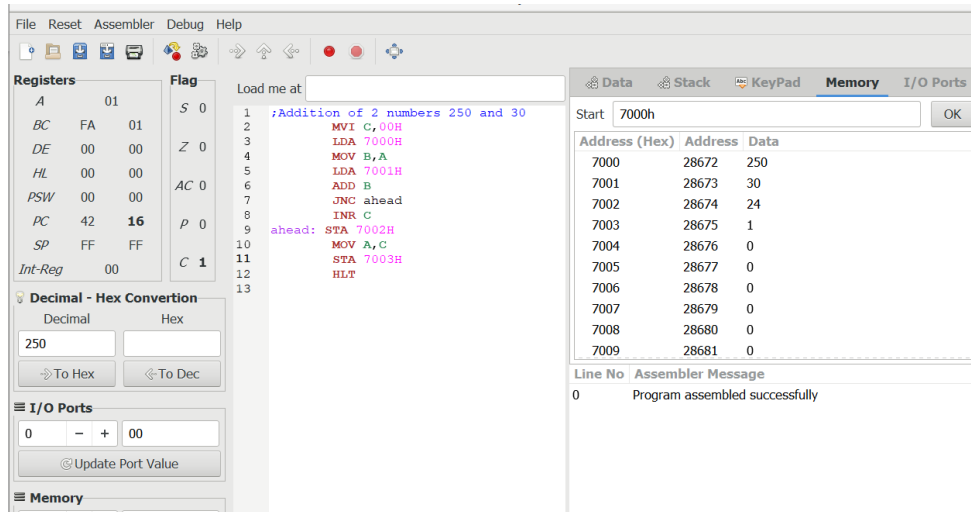
ahead: STA 7002H

MOV A,C

STA 7003H

HLT

19CSE303: Embedded Systems



d) 8-bit Subtraction

;Subtraction of 2 numbers 40H and 15H

LDA 3000H

MOV B,A

LDA 3001H

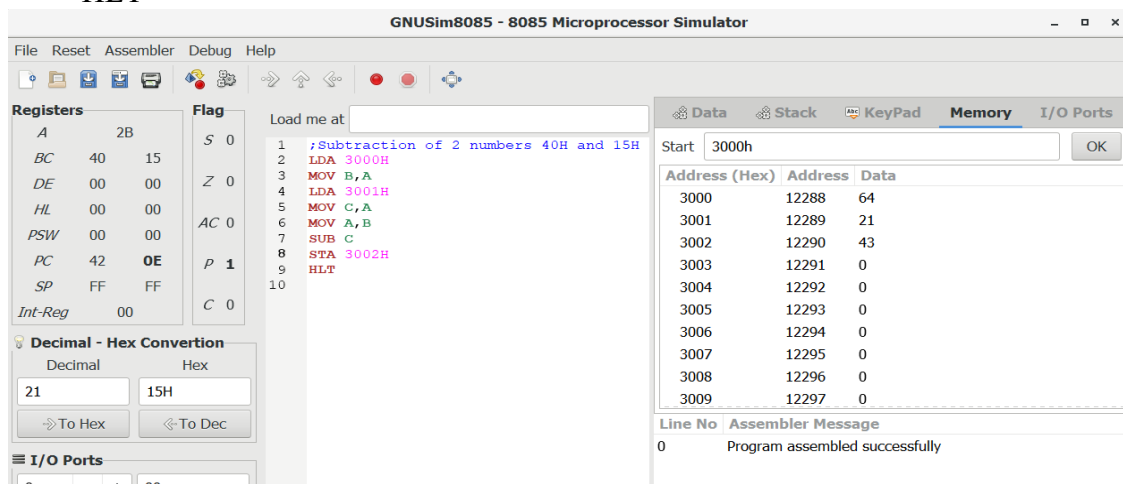
MOV C,A

MOV A,B

SUB C

STA 3002H

HLT



e) 8-bit Multiplication

;Multiplying 2 numbers 12H and 03H

MVI A,12H

MOV B,A

MVI C,03H

MVI A,00H

Label1: ADD B

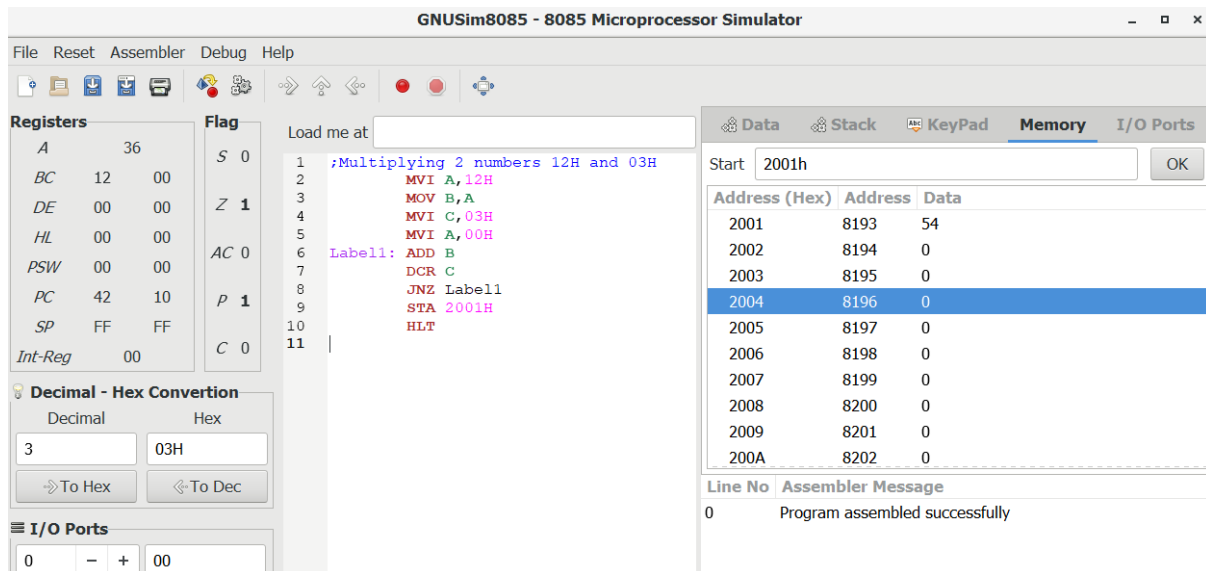
DCR C

JNZ Label1

STA 2001H

HLT

19CSE303: Embedded Systems



f) 8-bit Division

;Division of 2 numbers : Dividend -> 4AH & Divisor -> 08H

LDA 2050H ; Load Divisor in Accumulator

MOV B,A ; Copy Divisor to Register B

LDA 2051H ; Load Dividend in Accumulator

MVI C, 00H ; Initialize register C for Quotient

Next: CMP B ; Compare the no. in B with the no. in Accumulator

JC Loop ; Jump if B>A to Loop

INR C ; Increment Register C (Quotient)

SUB B ; Subtract B from A

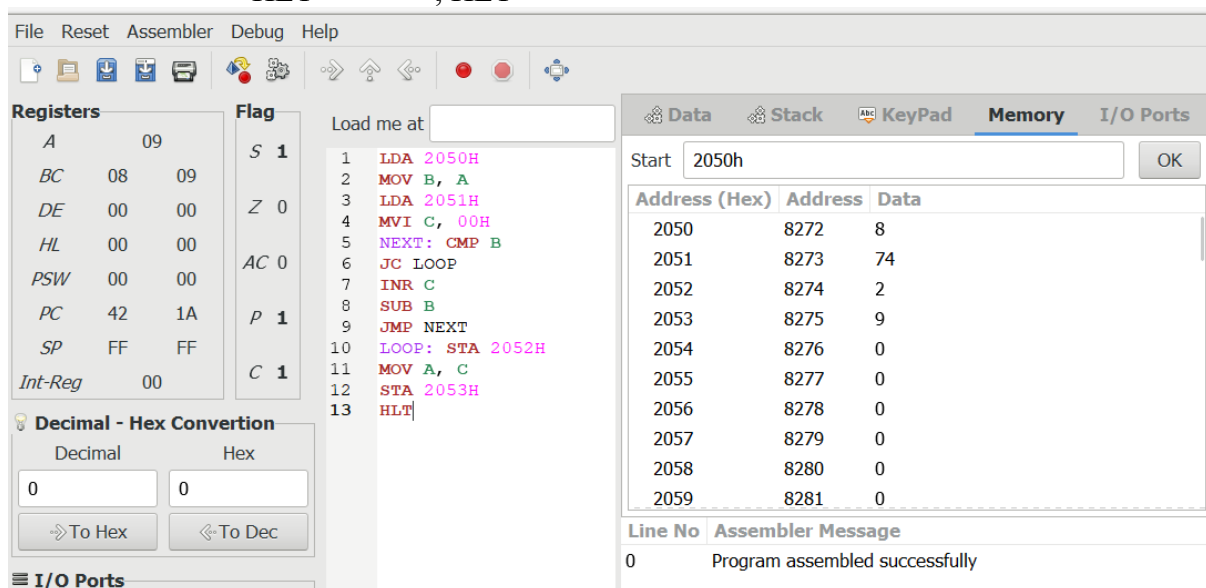
JMP Next ; Repeat the above steps till A becomes smaller than B

Loop: STA 2052H ; Store the remainder at memory address 2052

MOV A,C ; Move the contents of C to Accumulator

STA 2053H ; Store the Quotient at memory address 2053

HLT ; HLT



19CSE303: Embedded Systems

g) 16-bit Addition

;Addition of 2 numbers 1020H & 2040H

LHLD 8501H ; Get first 16-bit number in HL

XCHG ; Save first 16-bit number in DE

LHLD 8503H ; Get second 16-bit number in HL

MOV A,E ; Get lower byte of the first number

ADD L ; Add lower byte of the second number

MOV L,A ; Store the result in L register

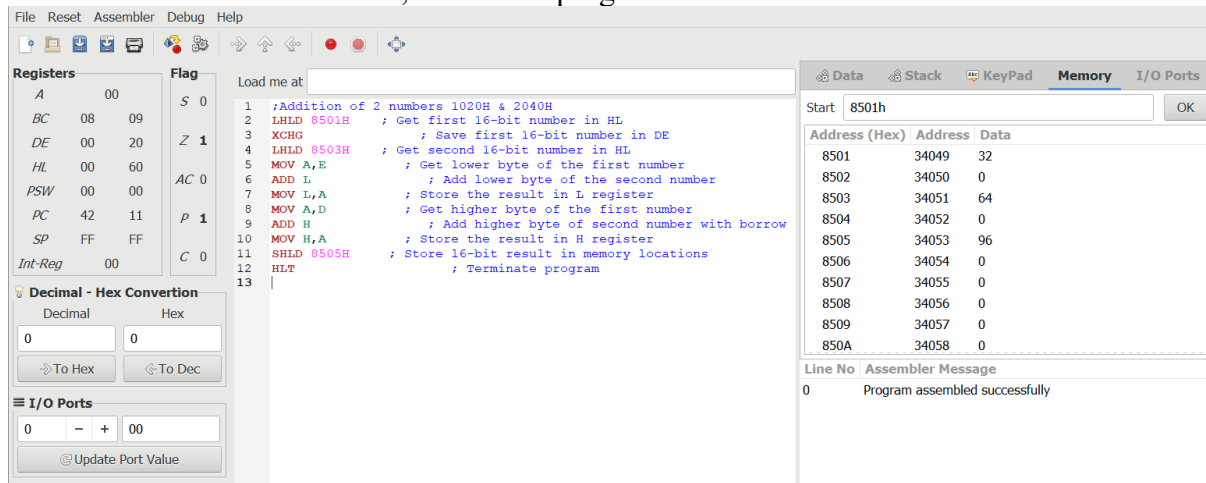
MOV A,D ; Get higher byte of the first number

ADD H ; Add higher byte of second number with borrow

MOV H,A ; Store the result in H register

SHLD 8505H ; Store 16-bit result in memory locations

HLT ; Terminate program



h) 16-bit Subtraction

;Subtraction of 2 numbers 2040H & 1020H

LHLD 8501H ; Get first 16-bit number in HL

XCHG ; Save first 16-bit number in DE

LHLD 8503H ; Get second 16-bit number in HL

MOV A,E ; Get lower byte of the first number

SUB L ; Subtract lower byte of the second number

MOV L,A ; Store the result in L register

MOV A,D ; Get higher byte of the first number

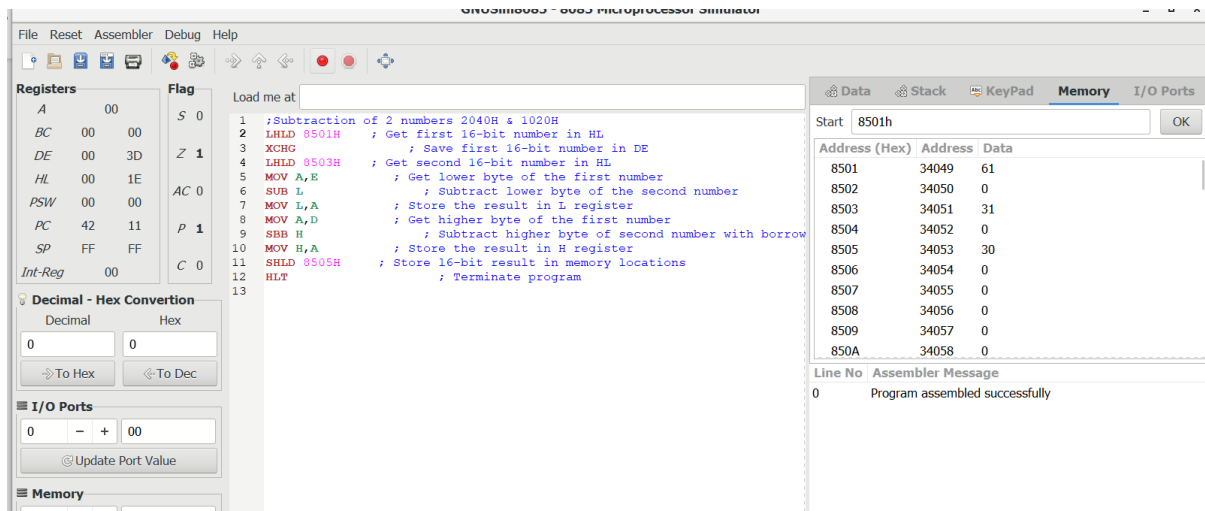
SBB H ; Subtract higher byte of second number with borrow

MOV H,A ; Store the result in H register

SHLD 8505H ; Store 16-bit result in memory locations

HLT ; Terminate program

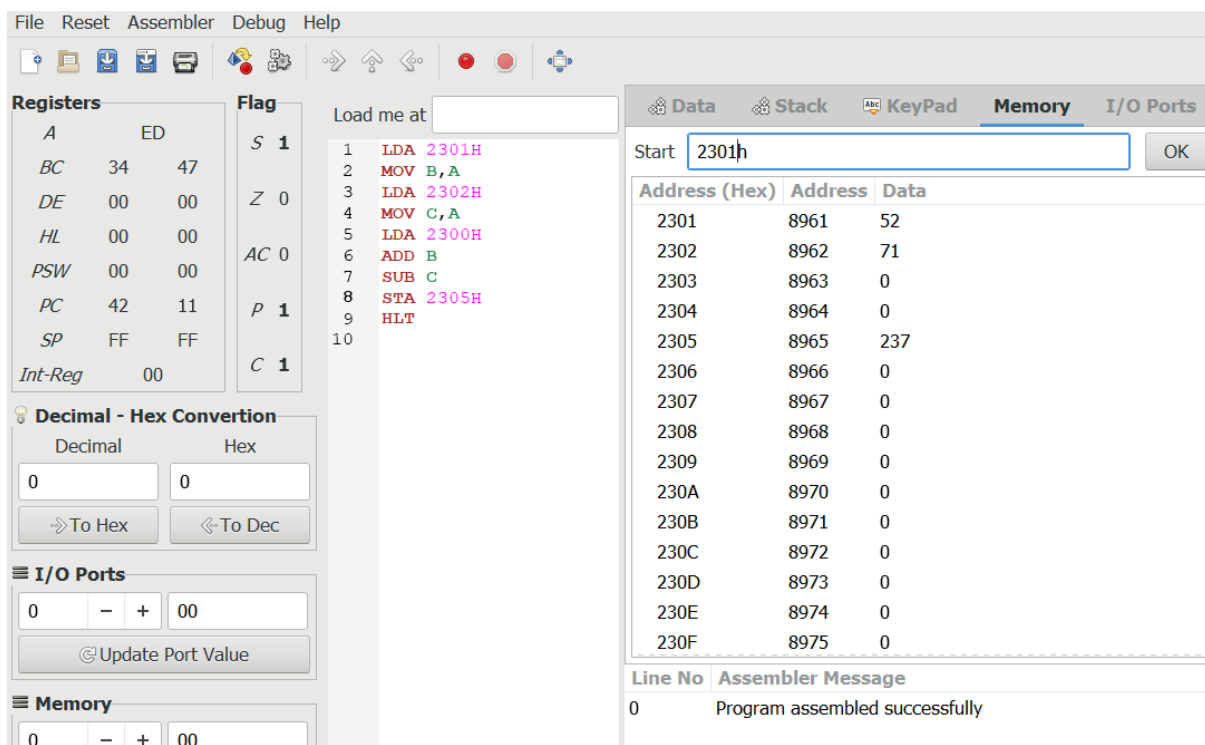
19CSE303: Embedded Systems



4. Write an assembly program in mnemonic code for 8085 processor to calculate

(A+B)-C where the value of

- A stored the value 34H at address 2300H
- B stored the value 47H at address 2301H
- C stored the value ABH at address 2302H



19CSE303: Embedded Systems

5. Write an ALP (Assembly Level Language) to add 10 numbers stored in the consecutive memory locations starting from 2000H. Write the code using conditional jump instruction. Store the final result in 200AH.

The screenshot displays an 8085 assembly simulator interface. The main window shows the assembly code being executed:

```
1  START: LXI H, 201H
2          MVI B, 0AH
3          MVI D, 0H
4  NEXT: MOV A, M
5          ADD D
6          MOV D, A
7          INX H
8          DCR B
9          JNZ NEXT
10         STA 200H
11         HLT
```

The left panel shows the state of the 8085 registers and flags:

Register	Value	Flag	Value
A	37	S	0
BC	00 47	Z	1
DE	37 00	AC	0
HL	02 0B	P	1
PSW	00 00	C	0
PC	42 15		
SP	FF FF		
Int-Reg	00		

The right panel shows the memory dump starting at 200h:

Address (Hex)	Address	Data
0200	512	55
0201	513	1
0202	514	2
0203	515	3
0204	516	4
0205	517	5
0206	518	6
0207	519	7
0208	520	8
0209	521	9

The bottom panel shows the assembler message:

```
Line No  Assembler Message
0        Program assembled successfully
```