MAHARISHI DAYANAND UNIVERSITY

Practical File

College Name: Delhi Global Institute of Technology

Student Name: Bazgha Razi

Registration Number:*191380214*

Subject Name: Microprocessor Lab

Subject Code: LC-ESC-321G

Session: 2021-2022

INDEX

Experiment 1

Aim: Write a 8085 assembly language program to add two 8-bit numbers.

Tool Used: Sim 8085(www.sim8085.com)

Program:

Memory Address	Hex Code	Mnemonics	Comments
0800H 0801H	3E 52	MVI A, 52H	Copy value 52H in accumulator.
0802H 0803H	06 24	MVI B, 24H	Copy value 24H in register B.
0804H	80	Add B	Add content of accumulator with register B and result in accumulator.
0805H 0806H 0807H	32 00 00	STA 0000H	Store sum in memory 0000H
0808H	76	HLT	End the program.

Before Execution	After Execution
A = 00H	A = 52H
B = 00H	B = 24H
0000H = 00H	0000H = 76H

Result: Two 8-bit numbers were added and result is stored in memory.

Experiment 2

Aim: Write a 8085 program to add two 16-bit numbers stored in memory.

Tool Used: Sim 8085(www.sim8085.com)

Program:

Memory Address	Hex Code	Mnemonics	Comments
0800H 0801H 0802H	2A 00 00	LHLD 0000H	Store value in HL register.
0803H	EB	XCHG	Exchange the value of HL with DE.
0804H 0805H 0806H	2A 02 00	LHLD 0002H	Store second value in HL register.
0807H	19	DAD D	Add DE and HL register.
0808H 0809H 080AH	22 04 00	SHLD 0004H	Store sum in memory 0004H and 0005H
080BH	76	HLT	End the program.

Before Execution	After Execution
HL = 0000H	HL = 76CCH
DE = 0000H	DE = 1C15H
0004H = 00H 0005H = 00H	0004H = CCH 0005H = 76H

Result: Two 16-bit numbers were added and result is stored in memory.

Experiment 3

Aim: Write a 8085 program to find 2’s complement of a number stored in memory 0000H. The result will be stored in memory 0001H.

Tool Used: Sim 8085(www.sim8085.com)

Program:

Memory Address	Hex Code	Mnemonics	Comments
0800H 0801H 0802H	3A 00 00	LDA 0000H	Get the number.
0803H	2F	CMA	Complement number.
0804H 0805H	C6 01	ADI 01H	2’s complement of number.
0806H 0807H 0808H	32 01 00	STA 0001H	Store complement in memory 0001H
0809H	76	HLT	End the program.

Before Execution	After Execution
PC = 0000H	PC = 080AH
SP= 0000H	SP = FFFFH
0001H = 00H	0000H = ABH

Result: 2’s complement of a number stored in memory 0001H.

Experiment 4

Aim: Write a 8085 program to multiply two 8-bit numbers stored in memory 0000H and 0001H. Product is stored in memory 0002H and 0003H.

Tool Used: Sim 8085(www.sim8085.com)

Program:

Memory Address	Hex Code	Mnemonics	Comments
0800H 0801H 0802H	3A 00 00	LDA 0000H	Get the number.
0803H	5F	MOV E, A	Move value of A in E register.
0804H 0805H	16 00	MVI D, 00H	Copy value 00H in D register.
0806H 0807H 0808H	3A 01 00	LDA 0001H	Get the number from 0001H.
0809H	4F	MOV C,A	Move the value of A in register C.
080AH 080BH 080CH	21 00 00	LXI H, 0000H	Load 0000H in HL.
080DH	19	AGAIN: DAD D	Add content of M in accumulator.
080EH	0D	DCR C	Decrement the value of regiter C.
080FH 0810H 0811H	C2 0d 08	JNZ AGAIN	Jump if z=0.
0812H 0813H 0814H	22 02 00	SHLD 0002H	Store product in memory 0002H
0815H	76	HLT	End the program.

Before Execution	After Execution
HL = 0000H	HL = 0142H
DE = 0000H	DE = 00 A1
PC = 0000H	PC = 081CH
SP= 0000H	SP = FFFFH
0002H = 00H	0002H = 42H
0003H = 00H	0003H = 01H

Result: Product is stored in memory 0002H, 0003H.

Experiment 5

Aim: Write a 8085 program to find the largest number in a block of data stored at 0000H to 0004H.

Tool Used: Sim 8085(www.sim8085.com)

Program:

Memory Address	Hex Code	Mnemonics	Comments
0800H 0801H 0802H	3A 10 00	LDA 0010H	Get the number.
0803H	4F	MOV C, A	Initialize counter.
0804H	AF	XRA A	Clear the accumulator using XRA.
0805H 0806H 0807H	21 00 00	LXI H, 0000H	Point to the first data.
0808H	BE	BACK : CMP M	Compare the value of

			accumulator and memory.
0809H 080AH 080BH	D2 0d 08	JNC SKIP	Jump when CY=0
080CH	7E	MOV A, M	Move the value of memory into accumulator.
080DH	23	SKIP : INX H	Increment the value of HL register.
080EH	0D	DCR C	Decrement the value of counter C.
080FH 0810H 0811H	C2 08 08	JNZ BACK	Jump if z=0
0812H 0813H 0814H	32 06 00	STA 0005H	Store largest number in memory 0006H.
0815H	76	HLT	End the program.

Before Execution	**After Execution**
HL = 0000H	HL = 0005H
PC = 0000H	PC = 0819H
SP= 0000H	SP = FFFFH
0005H = 00H	0005H = BAH

Result: Largest number stored at memory address 0005H.

Assembler Directive

Assembler directives are predefined alphabetical strings which helps the assembler to understand the assembly language programs properly and generate the machine codes. Some assembler directives are commonly used in 8086 assembly language programs:

DB: Define byte
DW: Define Word
ASSUME: Assume logical segment name
END: End of program
ENDS: End of segment
SEG: Segment of a label
PROC: Procedure
LABEL: Label
SEGMENT: Indicates start of a logical segment

MASM Installation

1. Download DOSBOX for your PC from the below link: https://www.dosbox.com/download.php?main=1
2. Download 8086.zip file.
3. Make 8086 folder in C drive.
4. Done, MASM installed on your PC.

Steps for Executing Program

1. Open DOSBOX and mount C drive in which your 8086 folder present.
   ```
   Z:>mount c c:\8086
   ```
2. Now navigate into drive C.
   ```
   Z:> c:\
   ```
3. For writing assembly program, execute the given command:
   ```
   C:> edit file_name.asm
   ```
4. After writing your assembly program save the file and exit and then run the following command:
   ```
   C:>masm file_name.asm
   C:>link file_name.obj
   ```
   Keep pressing enter and then write the next command:
   ```
   C:> debug file_name.exe
   ```
   Then press ‘t’ till the program completes.

Experiment 6

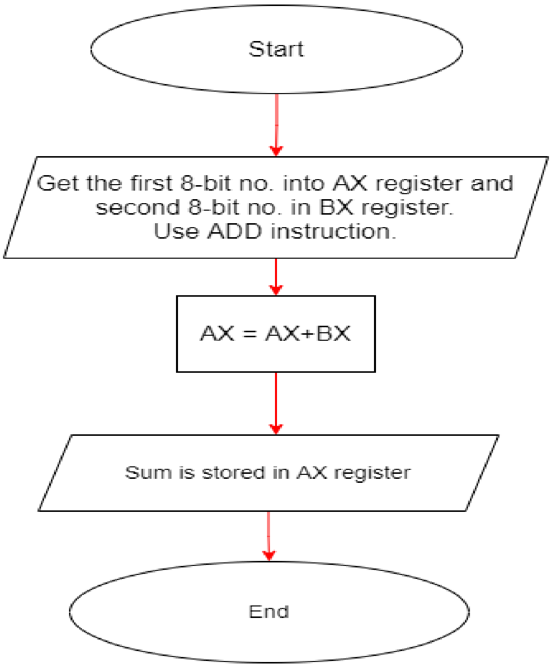
Aim: Write a 8086 program to add two 8-bit numbers.

Tool Used: DOSBOX, MASM

Program:

```
ASSUME CS: CODE  DS: DATA
DATA SEGMENT
        DATA1 DB 24H
        DATA2 DB 12H
DATA ENDS
CODE SEGMENT
BEGIN: MOV AX, DATA
         MOV DS, AX
         MOV AL, DATA1
         MOV BL, DATA2
         ADD AL, BL
         HLT
CODE ENDS
        END BEGIN
```

Flowchart:


Start
Get the first 8-bit no. into AX register and
second 8-bit no. in BX register.
Use ADD instruction.
AX = AX+BX
Sum is stored in AX register
End

Result: Two 8-bit numbers were added and the result is stored in the destination operand i.e., in AL.

Experiment 7

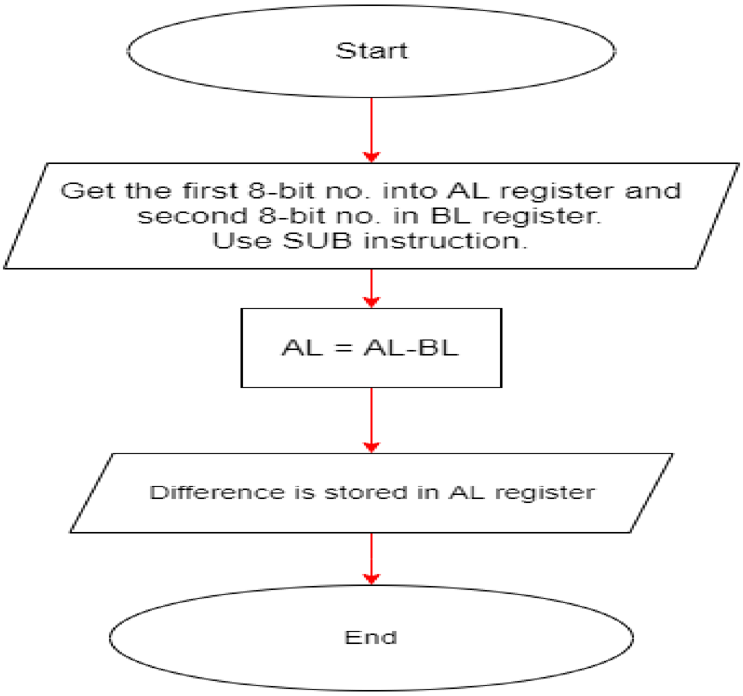
Aim: Write a 8086 program to subtract two 8-bit numbers.

Tool Used: DOSBOX, MASM

Program:

```
ASSUME CS: CODE DS: DATA
DATA SEGMENT
        DATA1 DW 24H
        DATA2 DW 13H
DATA ENDS
CODE SEGMENT
BEGIN: MOV AX, DATA
         MOV DS, AX
         MOV AL, DATA1
         MOV BL, DATA2
         SUB AL, BL
         HLT
CODE ENDS
        END BEGIN
```

Flowchart:


Start
Get the first 8-bit no. into AL register and second 8-bit no. in BL register. Use SUB instruction.
AL = AL-BL
Difference is stored in AL register
End

Result: Two 16-bit numbers were added and the result is stored in the destination operand i.e., in AX.

Experiment 8

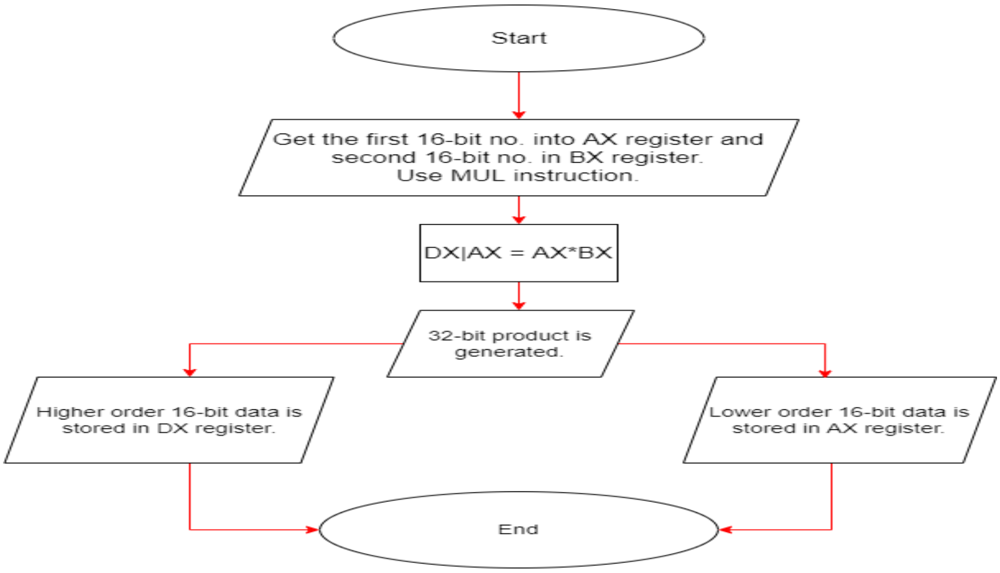
Aim: Write a 8086 program to multiply two 16-bit numbers.

Tool Used: DOSBOX, MASM

Program:

```
ASSUME CS: CODE  DS: DATA
DATA SEGMENT
        DATA1 DW 1111H
        DATA2 DW FFFFH
        RESULT1 DW 0000H
        RESULT2 DW 0000H
DATA ENDS
CODE SEGMENT
BEGIN: MOV AX, DATA
         MOV DS, AX
         MOV AX, DATA1
         MOV BX, DATA2
         MUL BX
         MOV RESULT1, DX
         MOV RESULT2, AX
         HLT
CODE ENDS
        END BEGIN
```

Flowchart:


Start
Get the first 16-bit no. into AX register and second 16-bit no. in BX register. Use MUL instruction.
DX|AX = AX*BX
32-bit product is generated.
Higher order 16-bit data is stored in DX register.
Lower order 16-bit data is stored in AX register.
End

Result: Two 16-bit numbers were multiplied and the 32-bit product is generated and stored in DX and AX register.

Experiment 9

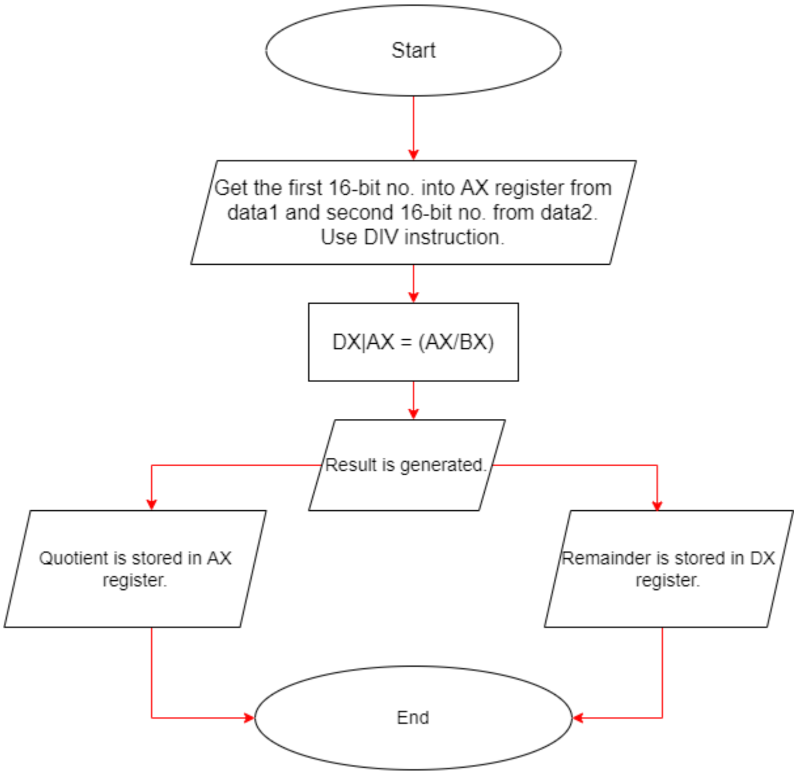
Aim: Write a 8086 program for 16-bit division.

Tool Used: DOSBOX, MASM

Program:

```
ASSUME CS: CODE  DS: DATA
DATA SEGMENT
        DATA1 DW 24FEH
        DATA2 DW 0010H
DATA ENDS
CODE SEGMENT
BEGIN: MOV AX, DATA
         MOV DS, AX
         MOV AX, DATA1
         DIV DATA2
         HLT
CODE ENDS
        END BEGIN
```

Flowchart:


Start
Get the first 16-bit no. into AX register from data1 and second 16-bit no. from data2. Use DIV instruction.
DX|AX = (AX/BX)
Result is generated.
Quotient is stored in AX register.
Remainder is stored in DX register.
End

Result: 16-bit number was divided by 8-bit number and the quotient is stored into AX register and remainder is stored in DX register.

Experiment 10

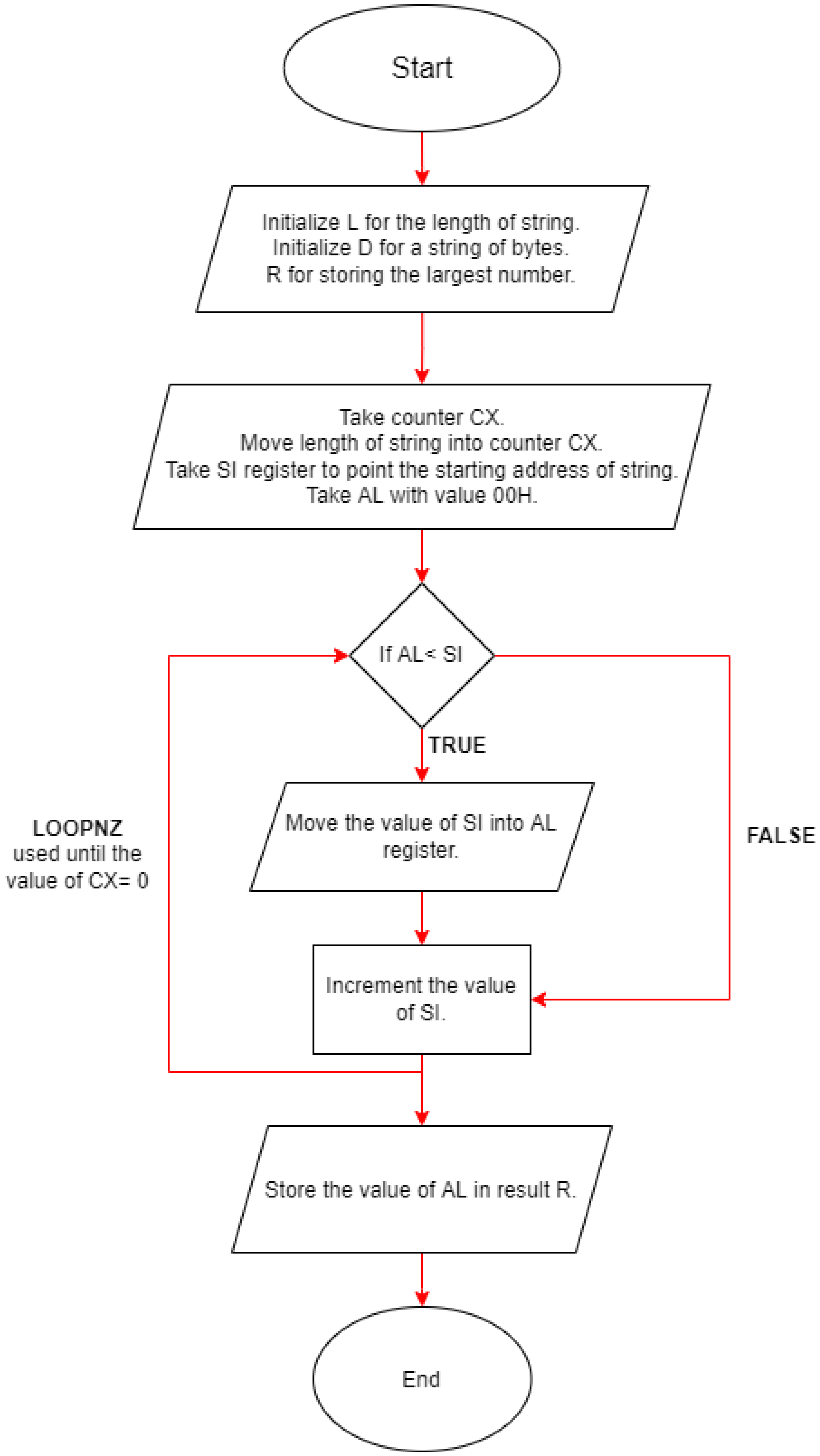
Aim: Write a 8086 program to find the largest number from a string of bytes and assume the length of string is 8.

Tool Used: DOSBOX, MASM

Program:

```
ASSUME CS: CODE  DS: DATA
DATA SEGMENT
          L DW 0008H
          D DB 38H,75H,26H,17H,30H,67H,99H,54H
          R DB 00H
DATA ENDS
CODE SEGMENT
BEGIN:MOV AX,DATA
           MOV DS,AX
           MOV CX,L
           LEA SI,D
           MOV AL,00H
AGAIN:CMP AL,[SI]
            JAE NEXT
            MOV AL,[SI]
 NEXT:INC SI
           LOOPNZ AGAIN
           MOV R,AL
           HLT
CODE ENDS
          END BEGIN
```

Flowchart:


Start
Initialize L for the length of string.
Initialize D for a string of bytes.
R for storing the largest number.
Take counter CX.
Move length of string into counter CX.
Take SI register to point the starting address of string.
Take AL with value 00H.
If AL< SI
TRUE
FALSE
LOOPNZ
used until the
value of CX= 0
Move the value of SI into AL
register.
Increment the value
of SI.
Store the value of AL in result R.
End

Result: Largest number from a string of bytes of length 8 is stored in result R.