

# OC EXPERIMENT LAB 4

---

**TITLE:** Writing assembly language programs

**NAME:** Yash Gupta

**ROLL NO:** S20200010234

**OBSERVATION:** In this lab I learnt how to write 8086 assembly language programs of doing basic arithmetic and logical operations on two numbers using different addressing modes.

## QUESTION:-

Write assembly language program to perform the following operations (given A=AFH and B=0FH)

### 1. Subtract A-B and check the CF, AC, PF and SF flags

```

01 CODE SEGMENT
02     ASSUME CS: CODE
03
04 START:  MOV AX,2000H;Initializing Data Register
05
06         MOV DS,AX
07
08         MOV AX,0000H;clearing AX Register
09
10         MOV AL,0AFH;storing value 0AFH to AL Register
11
12         MOV [600H],0FH
13
14         MOV BL,[600H];storing vlaue 0FH in BL Register
15
16         SUB AL,BL;Result of subtraction stored at AL
17
18         MOV BX,0500H
19
20         MOV [BX],AL;storing the result on memory location 500H
21
22 STOP:   HLT
23
24         CODE ENDS
25         ENDS START
26
27
28
29
30

```

flags

CF	0
ZF	0
SF	1
OF	0
PF	1
AF	0
IF	1
DF	0

analyse

Random Access Memory

2000:500

update

table list

2000:0500	A0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	á.....
2000:0510	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
2000:0520	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
2000:0530	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
2000:0540	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
2000:0550	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
2000:0560	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
2000:0570	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....

[illegible]

## 2. Multiplication A\*B and check the CF, AC, PF and SF flags

```

01 CODE SEGMENT
02     ASSUME CS: CODE
03
04 START: MOV AX,2000H;Initializing Data Register
05
06     MOV DS,AX
07
08     MOV AX,0000H;clearing the AX Register
09
10     MOV AL,0AFH;storing the value 0AFH in AL Register
11
12     MOV [600H],0FH
13
14     MOV BL,[600H];storing vlaue 0FH in BL Register
15
16     MUL BL;multiplying content of BL With AL and storing at AX register
17
18     MOV BX,0500H
19
20     MOV [BX],AX;storing the result in 500H memory location
21
22 STOP: HLT
23
24 CODE ENDS
25 END START
26
27

```

[illegible][illegible]

flags

CF	1
ZF	0
SF	0
OF	1
PF	0
AF	0
IF	1
DF	0

analyse

### 3. Division A/B and check the CF, AC, PF and SF flags

```
01 CODE SEGMENT
02     ASSUME CS: CODE
03
04 START:  MOV AX,2000H;Initializing Data Register
05
06         MOV DS,AX
07
08         MOV AX,0000H;clearing AX Register
09
10         MOV AL,0AFH;storing value 0AFH in AL Register
11
12         MOV [600H],0FH
13
14         MOV BL,[600H];storing vlaue 0FH in BL Register
15
16         DIV BL;Quotient will be in AL and remainder will be at AH
17
18         MOV BX,0500H
19
20         MOV [BX],AX;storing result in 500H memory location
21
22 STOP:   HLT
23
24         CODE ENDS
25         END START
26
27
28
29
```

Random Access Memory

2000:500    update    table    list

2000:0500	0B 0A 00 00 00 00 00 00 00-00 00 00 00 00 00 00	δ.....
2000:0510	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
2000:0520	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
2000:0530	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
2000:0540	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
2000:0550	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
2000:0560	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
2000:0570	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....

flags

CF 0

ZF 0

SF 0

OF 0

PF 0

AF 0

IF 1

DF 0

analyse

emulator: division lab 4.exe\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L		
AX	0A	0B	07118: 89 137 E	
BX	05	00	07119: 07 007 BEEP	
CX	00	1B	0711A: F4 244 f	
DX	00	00	0711B: 90 144 E	
CS	0710		0711C: 90 144 E	
IP	001A		0711D: 90 144 E	
SS	0710		0711E: 90 144 E	
SP	0000		0711F: 90 144 E	
BP	0000		07120: 90 144 E	
SI	0000		07121: 90 144 E	
DI	0000		07122: 90 144 E	
DS	2000		07123: 90 144 E	
ES	0700		07124: 90 144 E	

0710:001A

MOV AX, 02000h  
MOV DS, AX  
MOV AX, 00000h  
MOV AL, 0AFh  
MOV BL, [00600h], 0Fh  
DIV BL  
MOV BX, 00500h  
MOV [BX], AX  
HLT  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
...

screen source reset aux vars debug stack flags

#### 4. AND A with B and check the CF, AC, PF and SF flags

```

01 CODE SEGMENT
02     ASSUME CS: CODE
03
04 START:  MOV AX,2000H;Initializing Data Register
05
06         MOV DS,AX
07
08         MOV AX,0000H;clearing the AX Register
09
10         MOV AL,0AFH;storing value 0AFH in AL Register
11
12         MOV [600H],0FH
13
14         MOV BL,[600H];storing vlaue 0FH in BL Register
15
16         AND AL,BL;performing AND operation for values at AL and BL register
17
18         MOV BX,0500H
19
20         MOV [BX],AL;storing result in 500H memory location
21
22 STOP:   HLT
23
24         CODE ENDS
25     END START
26
27
28
29
30

```

Address	Value	...
2000:0500	0F 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	*
2000:0510	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	.
2000:0520	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	.
2000:0530	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	.
2000:0540	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	.
2000:0550	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	.
2000:0560	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	.
2000:0570	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	.



5. Average A and B i.e.  $(A+B)/2$  and check the CF, AC, PF and SF flags

```

01 CODE SEGMENT
02     ASSUME CS: CODE
03
04 START:  MOV AX,2000H;Intializing Data Register
05
06         MOV DS,AX
07
08         MOV AX,0000H;clearing the AX Register
09
10         MOV AL,0AFH;storing value 0AFH in AL Register
11
12         MOV [600H],0FH
13
14         MOV BL,[600H];storing vlaue 0FH in BL Register
15
16         ADD AL,BL;storing sum at AL register
17
18         MOV BL,02H;Giving 02H Value to BH
19
20         DIV BL;dividing it with Sum at AL register
21
22         MOV BX,0500H
23
24         MOV [BX],AX;storing the result in 500H memory location
25
26 STOP:   HLT
27
28         CODE ENDS
29         END START
30
31
32

```

[illegible]

flags	
CF	0
ZF	0
SF	1
OF	0
PF	1
AF	1
IF	1
DF	0

analyse

The screenshot shows the emulab 4.exe emulator window. The title bar indicates "emulator: average lab 4.exe\_". The menu bar includes file, math, debug, view, external, virtual devices, virtual drive, and help. Below the menu are several control buttons: Load, reload, step back, single step, run, and a slider for step delay ms: 0.

The main area displays registers, memory, and instructions:

- Registers:** A table with columns H and L. The values are: AX: 00 5F, BX: 05 00, CX: 00 1F, DX: 00 00, CS: 0710, IP: 001E, SS: 0710, SP: 0000, BP: 0000, SI: 0000, DI: 0000, DS: 2000, ES: 0700.
- Memory:** Address 0710:001E contains F4 244. Other addresses shown include 07117:F6 246, 07118:F3 243, 07119:BB 187, 0711A:00 000, 0711B:05 005, 0711C:89 137, 0711D:07 007, 0711F:90 144, 07120:90 144, 07121:90 144, 07122:90 144, 07123:90 144, 07124:90 144, 07125:90 144, 07126:90 144, 07127:90 144, 07128:90 144, 07129:90 144, 0712A:90 144, 0712B:90 144, 0712C:90 144.
- Instructions:** MOV AX, 02000h; MOV DS, AX; MOV AX, 00000h; MOV AL, 0AFh; MOV b.[00600h], 0Fh; MOV BL, [00600h]; ADD AL, BL; MOV BL, 02h; DIV BL; MOV BX, 00500h; MOV [BX], AX; HLT.