

**CSE341: Microprocessors**  
**Department of Computer Science and Engineering**

**Assignment 01**

**Task 01**

Swap two numbers using **ADD/SUB** instructions only.

```
edit: F:\9TH\341 LAB\LabA1_Task01.asm
file  edit  bookmarks  assembler  emulator  math  ascii codes  help
new  open  examples  save  compile  emulate  calculator  convertor  options  help  about

01  .MODEL SMALL
02
03  .STACK 100H
04
05  .DATA
06      ; DEFINE YOUR VARIABLES HERE
07
08  .CODE
09  MAIN PROC
10
11      MOV AX, @DATA
12      MOV DS, AX
13
14      MOV AL, 3 ;AL=3
15      MOV BL, 8 ;BL=8
16
17      ; Code for Swap two numbers using ADD/SUB instructions only.
18
19      ADD DL,BL ;DL=DL+8=0+8=8 Here, DL is used for storing BL register's value temporarily
20      ADD BL,AL ;BL=BL+AL=8+3=11
21      SUB BL,DL ;BL=BL-DL=11-8=3 So,BL=3
22      SUB AL,AL ;AL=AL-AL=3-3=0
23      ADD AL,DL ;AL=AL+DL=0+8=8
24      ;SWAP complete as now, AL=8
25
26      ; MY CODE ENDS HERE
27
28      MOV AX, 4C00H
29      INT 21H
30
31  MAIN ENDP
32  END MAIN
```

```

.MODEL SMALL
.STACK 100H
.DATA
    ; DEFINE YOUR VARIABLES HERE

.CODE
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX

    ;CODE STARTS HERE

    MOV AL, 3 ;AL=3
    MOV BL, 8 ;BL=8

    ; Code for Swap two numbers using ADD/SUB instructions only.

    ADD DL,BL ;DL=DL+8=0+8=8 Here, DL is used for storing BL register's value temporarily
    ADD BL,AL ;BL=BL+AL=8+3=11
    SUB BL,DL ;BL=BL-DL=11-8=3 So,BL=3
    SUB AL,AL ;AL=AL-AL=3-3=0
    ADD AL,DL ;AL=AL+DL=0+8=8
                ;SWAP complete as now, AL=8

    ; MY CODE ENDS HERE

    MOV AX, 4C00H
    INT 21H

MAIN ENDP
END MAIN

```

## Task 02

If A, B, C and D are 4 byte sized non zero variables, perform the given mathematical operation

$$D = A - (C - A) + D$$

```
edit: F:\9TH\341 LAB\LabA1_Task02.asm
file  edit  bookmarks  assembler  emulator  math  ascii codes  help
new  open  examples  save  compile  emulate  calculator  convertor  options  help  about

01 .STACK 100H
02
03
04 .DATA
05 ; DEFINE YOUR VARIABLES HERE
06 A db 1
07 C db 3
08 D db 4
09
10 .CODE
11 MAIN PROC
12
13     MOV AX, @DATA
14     MOV DS, AX
15
16     ;CODE STARTS HERE
17
18     ; Code for If A, B, C and D are 4 byte sized non zero variables,
19     ; perform the given mathematical operation D = A - (C - A) + D
20
21     MOV AL, A ; AL=A
22     MOV CL, C ; CL=C
23     MOV DL, D ; DL=D
24
25
26     SUB CL, AL ; CL=CL-AL = C-A
27     SUB AL, CL ; AL=AL-CL =>A = A-(C-A)
28     ADD AL, DL ; AL=AL+DL =>AL = A-(C-A)+D
29     MOV D, AL ; mathematical operation is completed and result is stored in D
30
31     ;CODE ENDS HERE
32
33     MOV AX, 4C00H
34     INT 21H
35
36 MAIN ENDP
37 END MAIN
```

```

.MODEL SMALL
.STACK 100H
.DATA
    ; DEFINE YOUR VARIABLES HERE
    A db 1
    C db 3
    D db 4
.CODE
    MAIN PROC

        MOV AX, @DATA
        MOV DS, AX

        ;CODE STARTS HERE

        ; Code for If A, B, C and D are 4 byte sized non zero variables,
        ; perform the given mathematical operation  $D = A - (C - A) + D$ 

        MOV AL, A ; AL=A
        MOV CL, C ; CL=C
        MOV DL, D ; DL=D

        SUB CL,AL ; CL=CL-AL = C-A
        SUB AL,CL ; AL=AL-CL =>A = A-(C-A)
        ADD AL,DL ; AL=AL+DL =>AL = A-(C-A)+D
        MOV D,AL ; mathematical operation is completed and result is stored in D

        ;CODE ENDS HERE
        MOV AX, 4C00H
        INT 21H

    MAIN ENDP
END MAIN

```

## Task 03

If W, X, Y and Z are 4 byte sized non zero variables, then perform the given mathematical operation

$$X = (X * Z) / (Y * W) / (Z * X)$$

```
edit: F:\9TH\341 LAB\LabA1_Task03.asm
file  edit  bookmarks  assembler  emulator  math  ascii codes  help

new  open  examples  save  compile  emulate  calculator  convertor  options  help  about

01  .MODEL SMALL
02
03  .STACK 100H
04
05  .DATA
06      ; DEFINE YOUR VARIABLES HERE
07      W db 1
08      X db 2
09      Y db 3
10      Z db 4
11
12  .CODE
13      MAIN PROC
14
15          MOV AX, @DATA
16          MOV DS, AX
17
18          ; YOUR CODE STARTS HERE
19
20          MOV AL, X ; AL=X
21          MOV BL, Z ; BL= Z
22          MUL BL    ; AX=AL*BL =X*Z
23          MOV CX,AX ; CX=AX=X*Z
24
25          MOV AL, Y ; AL=Y
26          MOV BL, W ; BL=W
27          MUL BL    ; AX=AL*BL = AX=Y*W
28          MOV DX,AX ; DX=Y*W
29
30          MOV AX,CX ; AX=X*Y
31          DIV DL    ; AX=AX/DL = <X*Y>/<Y*W>
32
33          MOV CX,AX ; CX=AX = CX=<X*Y>/<Y*W>
34
35          MOV AL, Z ; AL=Z
36          MOV BL, X ; BL=X
37          MUL BL    ; AX=AL*BL = AX=Z*X
38
39          MOV DX,AX ; DX=Z*X
40
41          MOV AX,CX ; AX=CX = <X*Y>/<Y*W>
42          DIV DL    ; AX=AX/DL = <X*Y>/<Y*W>/<Z*X>
43          MOV X,AL  ;mathematical operation is completed and result is stored in X
44
45          ; YOUR CODE ENDS HERE
46
47          MOV AX, 4C00H
48          INT 21H
49
50      MAIN ENDP
51      END MAIN
```

```

.MODEL SMALL
.STACK 100H
.DATA
    ; DEFINE YOUR VARIABLES HERE
    W db 1
    X db 2
    Y db 3
    Z db 4

.CODE
MAIN PROC

    MOV AX, @DATA
    MOV DS, AX
    ; YOUR CODE STARTS HERE

    MOV AL, X ;AL=X
    MOV BL, Z ;BL= Z
    MUL BL    ;AX=AL*BL =X*Z
    MOV CX,AX ;CX=AX=X*Z

    MOV AL, Y ;AL=Y
    MOV BL, W ;BL=W
    MUL BL    ;AX=AL*BL = AX=Y*W
    MOV DX,AX ;DX=Y*W

    MOV AX,CX ;AX=X*Y
    DIV DL    ;AX=AX/DL = (X*Y)/(Y*W)

    MOV CX,AX ;CX=AX    = CX=(X*Y)/(Y*W)

    MOV AL, Z ;AL=Z
    MOV BL, X ;BL=X
    MUL BL    ;AX=AL*BL = AX=Z*X

    MOV DX,AX ;DX=Z*X

    MOV AX,CX ;AX=CX    = (X*Y)/(Y*W)
    DIV DL    ;AX=AX/DL = (X*Y)/(Y*W)/(Z*X)
    MOV X,AL  ;mathematical operation is completed and result is stored in X

    ; YOUR CODE ENDS HERE
    MOV AX, 4C00H
    INT 21H

MAIN ENDP
END MAIN

```

## Task 04

Perform the following arithmetic operation:  $(5 * 2) + (4 - 0) + (5 + 3 + 2) - (6 / 2)$

edit: F:\9TH\341 LAB\LabA1\_Task04.asm

file edit bookmarks assembler emulator math ascii codes help

new	open	examples	save	compile	emulate	calculator	converter	options	help	about
-----	------	----------	------	---------	---------	------------	-----------	---------	------	-------

```
01 .MODEL SMALL
02
03 .STACK 100H
04
05 .DATA
06 ; DEFINE YOUR VARIABLES HERE
07 .CODE
08 MAIN PROC
09
10     MOV AX, @DATA
11     MOV DS, AX
12
13     ; YOUR CODE STARTS HERE
14
15     MOV AX, 5 ;AX=5
16     MOV BX, 2 ;BX=2
17     MUL BX    ;AX=<5*2>
18
19     MOV BX, 4 ;BX=4
20     SUB BX, 0 ;BX=<4-0>
21
22     ADD AX, BX ;AX=AX+BX=<5*2>+<4-0>
23
24     MOV BX, 5 ;BX=5
25     ADD BX, 3 ;BX=5+3
26     ADD BX, 2 ;BX=5+3+2
27
28     ADD AX, BX ;AX=AX+BX=<5*2>+<4-0>+<5+3+2>
29     MOV BX, AX ;BX=AX=<5*2>+<4-0>+<5+3+2>
30
31     MOV AX, 6 ;AX=6
32     MOV CX, 2 ;CX=2
33     DIV CX    ;AX=AX/CX=6/2
34
35     SUB BX, AX ;BX=BX-AX=<5*2>+<4-0>+<5+3+2>-<6/2>
36                 ;arithmetic operation is completed & result stored in BX
37
38     ; YOUR CODE ENDS HERE
39
40     MOV AX, 4C00H
41     INT 21H
42
43 MAIN ENDP
44 END MAIN
45
46
```

```

.MODEL SMALL
.STACK 100H
.DATA
    ; DEFINE YOUR VARIABLES HERE
.CODE
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX
    ; YOUR CODE STARTS HERE

    MOV AX, 5 ;AX=5
    MOV BX, 2 ;BX=2
    MUL BX    ;AX=(5*2)
    MOV BX, 4 ;BX=4
    SUB BX, 0 ;BX=(4-0)
    ADD AX,BX ;AX=AX+BX=(5*2)+(4-0)
    MOV BX, 5 ;BX=5
    ADD BX, 3 ;BX=5+3
    ADD BX, 2 ;BX=5+3+2
    ADD AX,BX ;AX=AX+BX=(5*2)+(4-0)+(5+3+2)
    MOV BX,AX ;BX=AX=(5*2)+(4-0)+(5+3+2)
    MOV AX, 6 ;AX=6
    MOV CX, 2 ;CX=2
    DIV CX    ;AX=AX/CX=6/2

    SUB BX, AX;BX=BX-AX=(5*2)+(4-0)+(5+3+2)-(6/2)
    ;arithmetic operation is completed & result stored in BX
    ; YOUR CODE ENDS HERE
    MOV AX, 4C00H
    INT 21H

MAIN ENDP
END MAIN

```