RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY

"Heaven's Light is Our Guide"

Assignment

Course Title : Microprocessors and Assembly Language

Course No : CSE 3109

Submitted To:

Sadia Zaman Mishu

Assistant Professor,

Department of Computer Science and Engineering.

Rajshahi University of Engineering & Technology.

Submitted By:

A. B. M. Siddiqur Rahman

Roll : **1703054**

Section : A

Department : Computer Science and Engineering

Submission Date : 21-08-2021

Chapter: 05
The processor status and the flags register

Exercise: 01
Problem Title: For each of the following instructions, give destination contents and the new settings of CF, SF, ZF, PF and OF. Suppose that th flags are initially 0 in each parts of this question.

(a) ADD AX,BX ;Where AX=7FFFh and BX=0001h

AX= 7FFFh= 0111 1111 1111 1111
BX= 0001h= 0000 0000 0000 0001 +
AX= 8000h= 1000 0000 0000 0000

CF=0, SF=1, ZF=0, PF=1, OF=1

(b) SUB AL,BL ;Where AL=01h and BL=FFH

AL= 01h= 0000 0001
BL= FFh= 1111 1111 -
AL= 02h= 0000 0010

CF=0, SF=1, ZF=0, PF=0, OF=0

(c) DEC AL ;Where AL=00h

AL= 00h= 0000 0000
= 01h= 0000 0001 -
AL=FFh= 1111 1111

CF=0, SF=1, ZF=0, PF=1, OF=0

(d) NEG AL ;Where AL=7FH

AL=7Fh = 0111 1111
1's com.= 1000 0000
= 0000 0001 +
AL=81h= 1000 0001

CF=1, SF=1, ZF=0, PF=1, OF=0

(e) XCHG AX,BX ;Where AX=1ABCh and BX=712Ah

CF=0, SF=0, ZF=0, PF=0, OF=0
None of the flags are affected by XCHG

(f) ADD AL,BL ;Where AL=80h and BL=FFh

```
AL= 80h=   1000  0000
BL= FFh=   1111  1111 +
AL= 7Fh= 10111  1111
```

CF=1, SF=0, ZF=0, PF=0, OF=1

(g) SUB AX,BX ;Where AX=0000h and BX=8000h

```
AX= 0000h=  0000 0000 0000 0000
BX= 8000h=  1000 0000 0000 0000 -
AX= 8000h=  1000 0000 0000 0000
```

CF=1, SF=1, ZF=0, PF=1, OF=1

(h) NEG AX ;Where AX=0001h

```
AX= 0001h=  0000  0001
1's comp.   =  1111 1110
            =  0000  0001 +
AX= FFh     =  1111  1111
```

CF=1, SF=1, ZF=0, PF=1, OF=0

Exercise: 02

(a) Suppose that AX and BX both contain positive numbers and ADD AX, BX is executed. Show that there is a carry into the msb but no carry out of the msb if and only if signed overflow occurs.

```
AX= 0101 0101
BX= 0101 0101 +
  = 1010 1010
```

Ci=1 and Co=0
Ci XOR Co=1 XOR 0=1=OF(Signed Overflow)

(b) Suppose that AX and BX both contain negative numbers and ADD AX, BX is executed. Show that there is a carry out of the msb but no carry into the msb if and only if signed overflow occurs.

AX= 1001 0101
BX= 1010 0101 +
=10011 1010
Ci=0 and Co=1
Ci XOR Co=0 XOR 1=1=OF(Signed Overflow)

Exercise: 03
Problem Title: Suppose ADD AX, BX is executed. Give the resulting value of AX and tell whether signed or unsigned overflow occurred.

(a)

AX= 512Ch= 0101 0001 0010 1100
BX= 4185h= 0100 0001 1000 0101 +
AX= 92B1h= 1001 0010 1011 0001

Signed OF=1 and Unsigned CF=0

(b)

AX= FE12h = 1111 1110 0001 0010
BX= 1ACBh= 0001 1010 1100 1011 +
AX= 18DDh= 10001 1000 1101 1101

Signed OF=0 and Unsigned CF=1

(c)

AX= E1E4h= 1110 0001 1110 0100
BX= DAB3h= 1101 1010 1011 0011 +
AX= BC97h= 11011 1100 1001 0111

Signed OF=0 and Unsigned CF=1

(d)

AX= 7132h= 0111 0001 0011 0010
BX= 7000h= 0111 0000 0000 0000 +
AX= E132h= 1110 0001 0011 0010

Signed OF=1 and Unsigned CF=0

(e)

AX= 6389h= 0110 0011 1000 1001
BX= 1176h= 0001 0001 0111 0110 +
AX= 74FFh= 0111 0100 1111 1111

Signed OF=0 and Unsigned CF=0

Exercise: 04
Problem Title: Suppose SUB AX, BX is executed. Give the resulting value of AX and tell whether signed or unsigned overflow occurred.

(a)

AX= 2143h= 0010 0001 0100 0011
BX= 1986h= 0001 1001 1000 0110 -
AX=07BDh= 0000 0111 1011 1101

Signed OF=0 Unsigned CF=0

(b)

AX= 81FEh= 1000 0001 1111 1110
BX= 1986h= 0001 1001 1000 0110 -
AX= 6878h= 0110 1000 0111 1000

Signed OF=1 Unsigned CF=0

(c)

AX= 19BCh= 0001 1001 1011 1100
BX= 81FEh= 1000 0001 1111 1110 -
AX= 97BEh= 1001 0111 1011 1110

Signed OF=1 Unsigned CF=1

(d)

AX= 0002h= 0000 0000 0000 0010
BX= FE0Fh= 1111 1110 0000 1111 -
AX= 01F3h= 0000 0001 1111 0011

Signed OF=0 Unsigned CF=1

(e)

AX= 8BCDh= 1000 1011 1100 1101
BX= 71ABh = 0111 0001 1010 1011 -
AX= 1A22h = 0001 1010 0010 0010

Signed OF=1 Unsigned CF=0

Chapter: 06
Flow control instruction

Exercise: 01
Problem Title: Write assembly code for each of the following decision structure.

(a)

```
CMP AX,0
JGE END_IF
MOV BX,-1

END_IF:
```

(b)

```
CMP AL,0
JNL ELSE_
MOV AH,0FFh
JMP END_IF

ELSE_:
        MOV AH,0
END_IF:
```

(c)

```
CMP DL,'A'
JL END_IF
CMP DL,'Z'
JG END_IF
MOV AH,2
INT 21H

END_IF:
```

(d)

```
CMP AX,BX
JGE END_IF
CMP BX,CX
JGE ELSE_
MOV AX,0
JMP END_IF

ELSE_:
        MOV BX,0
END_IF:
```

(e)

```
CMP AX,BX
JL THEN_
CMP BX,CX
JL THEN_
MOV DX,1
JMP END_IF

THEN:
        MOV DX,0
END_IF:
```

(f)

```
CMP AX,BX
JNL ELSE_
MOV AX,0
JMP END_IF

ELSE_:
        CMP BX,CX
        JNL ELSE_2
        MOV BX,0
        JMP END_IF
ELSE_2:
        MOV CX,0
END_IF:
```

<u>Exercise:</u> 02
<u>Problem Title:</u> Use a CASE structure to code.

```
MOV AH,1
INT 21H

CMP AL,'A'
JE EXE_CR
CMP AL,'B'
JE EXE_LF
MOV AH,4CH
INT 21H
JMP END_CASE

EXE_CR:
        MOV AH,2
        MOV DL,0DH
        INT 21H
        JMP END_CASE
```

```
EXE_LF:
        MOV AH,2
        MOV DL,0AH
        INT 21H
END_CASE
```

Exercise: 03

(a) Put the sum 1+4+7+…+148 in

(last–first)/ difference

=(148 – 1)/3 = 49 loops

```
MOV CX,49
MOV AX,1
MOV BX,1

L1:
        ADD BX,3
        ADD AX,BX
LOOP L1
```

(b) Put the sum 100+95+90+…+5 in AX.

(last–first)/ difference

=(100 – 5)/5 = 19 loops

```
MOV CX,19
MOV AX,100
MOV BX,100

L1:
        SUB BX,5
        ADD AX,BX
LOOP L1
```

Exercise: 04

(a) Put the sum of the fisrt 50 terms of the arithmetic sequence 1,5,9,13,…in DX.

```
MOV CX,50
MOV DX,1
MOV
AX,1

L1:
        ADD AX,4
LOOP L1
```

(b) Read a character and display it 80 times on the next line.

```
MOV AH,1
INT 21H
MOV AH,2
MOV DL,0AH
INT 21H
MOV DL,0DH
INT 21H
MOV DL,AL
MOV CX,80

DISPLAY:
        INT 21H
LOOP DISPLAY
```

(c) Read a five-character password and overprint it by executing a carriage return and displaying five X's.

```
MOV CX,5
MOV AH,7
L1:
        INT 21H
LOOP L1

MOV DL,'X'
MOV CX,5
MOV AH,2
L2:
        INT 21H
LOOP L2
```

Exercise: 05

Write a sequence of instructions to divide AX by BX and put the quotient in CX.

```
MOV AX,0
WHILE_:
        CMP CX,BX
        JL END_WHILE
        INC AX
        SUB CX,BX
        JMP WHILE_
END_WHILE:
```

Exercise: 06

Write a sequence of instructions to multiply AX by BX and put the product in CX.

```
XOR CX,CX
L1:
        ADD CX,AX
        DEC BX
JNZ L1
```

Exercise: 07

(a) Write instructions to read characters until either a nonblank character is typed, or 80 characters have been typed. Use LOOPE.

```
MOV AH,1
MOV CX,80
L1:
        INT 21H
        CMP AL,20H
LOOPE L1
```

(b) Write instructions to read characters until either a carriage return is typed or 80 characters have been types. Use LOOPNE.

```
MOV AH,1
MOV CX,80
L1:
        INT 21H
        CMP AL,0DH
LOOPNE L1
```

Exercise: 08

Program Name: Write a program to display a “?”, read two capital letters and display them on the next line in alphabetical order.

Source Code:

```
.MODEL SMALL
.STACK 100H

.DATA
    CR_LF  DB  0DH,0AH,"$"

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

        MOV AH,2
        MOV DL,"?"
        INT 21H

        MOV AH,1
        INT 21H

        MOV BL,AL

        MOV AH,1
        INT 21H

        MOV BH,AL

        MOV AH,9
        LEA DX,CR_LF
        INT 21H

        MOV AH,2

        CMP BL,BH

        JAE FIRST
        MOV DL,BL
        INT 21H

        MOV DL,BH
        INT 21H

        JMP FINISH
        FIRST:
            MOV DL,BH
            INT 21H

            MOV DL,BL
            INT 21H
        FINISH:
```

```
            MOV AH,4CH
            INT 21H
    MAIN ENDP
END MAIN
```

Output:

emulator screen (80x25 chars)

```
?BA
AB
```

Exercise: 09

Program Name: Write a program to display the extended ASCII characters(ASCII codes 80h to FFh).

Source Code:

```
.MODEL SMALL
.STACK 100H

.DATA
    CR_LF   DB   0DH,0AH,"$"

.CODE
MAIN PROC

    MOV AX,@DATA
    MOV DS,AX

    MOV BL,80H
    MOV CL,0
    TOP:
        CMP CL,10
        JE NEWLINE

        INC CL

        MOV AH,2
        MOV DL,BL
        INT 21H

        MOV DL," "
        INT 21H
        INC BL

        CMP BL,0FFH
        JE BOTTOM
        JMP TOP
    NEWLINE:
```

```
        MOV AH,9
        LEA DX,CR_LF
        INT 21H

        MOV CL,0
        JMP TOP

    BOTTOM:
        MOV AH,4CH
        INT 21H
    MAIN ENDP
END MAIN
```

Output:

emulator screen (80x25 chars)

Exercise: 10

Program Name: Write a program that will prompt the user to enter a hex digit("0" to "9" and "A" to "F") and display it on the next line in decimal.

Source Code:

```
.MODEL SMALL
.STACK 100H

.DATA
    CMD DB 'ENTER A HEX DIGIT: $'
    DECIMAL DB 0DH,0AH,'IN DECIMAL, IT IS: $'
    REPEAT DB 0DH,0AH,'DO YOU WANT TO DO IT AGAIN? $'
    INVALID DB 0DH,0AH,'ILLEGAL CHARACTER - ENTER 0..9 or A..F: $'
    CR_LF DB 0DH,0AH,"$"

.CODE
    MAIN PROC
        MOV AX,@DATA
        MOV DS,AX
        FIRST:
            MOV AH,9
            LEA DX,CMD
            INT 21H
```

```
SECOND:
    MOV AH,1
    INT 21H

    MOV BL,AL

    CMP BL,"A"
    JB THIRD

    CMP BL,"F"
    JA ILLEGAL

    JMP LETTER_DIGIT
THIRD:
    CMP BL,"0"
    JB ILLEGAL

    CMP BL,"9"
    JA ILLEGAL

    JMP NUMERIC_DIGIT

ILLEGAL:
    MOV AH,9
    LEA DX,INVALID
    INT 21H

    JMP SECOND

NUMERIC_DIGIT:
    MOV AH,9
    LEA DX,DECIMAL
    INT 21H

    MOV AH,2
    MOV DL,BL
    INT 21H

    JMP CONTINUE

LETTER_DIGIT:
    MOV AH, 9
    LEA DX,DECIMAL
    INT 21H

    MOV AH,2
    MOV DL,31H
    INT 21H

    SUB BL,11H

    MOV DL,BL
    INT 21H
CONTINUE:
    MOV AH,9
```

```
            LEA DX,CR_LF
            INT 21H

            LEA DX,REPEAT
            INT 21H

            MOV AH,1
            INT 21H

            CMP AL,"y"
            JE JUMP

            CMP AL,"Y"
            JE JUMP

            JMP FINISH

        JUMP:
            LEA DX,CR_LF
            MOV AH,9
            INT 21H
            INT 21H

            JMP FIRST

        FINISH:
            MOV AH,4CH
            INT 21H
    MAIN ENDP
END MAIN
```

Output:

emulator screen (80x25 chars)

```
ENTER A HEX DIGIT: 9
IN DECIMAL, IT IS: 9

DO YOU WANT TO DO IT AGAIN? Y

ENTER A HEX DIGIT: X
ILLEGAL CHARACTER - ENTER 0..9 or A..F: C
IN DECIMAL, IT IS: 12

DO YOU WANT TO DO IT AGAIN? N
```

Exercise: 11

Program Name: Do exercise 10, except that if the user fails to enter a hex digit character in three tries, display a message and terminate the program.

Source Code:

```
.MODEL SMALL
.STACK 100H

.DATA
    CMD DB 'ENTER A HEX DIGIT: $'
    DECIMAL DB 0DH,0AH,'IN DECIMAL, IT IS: $'
    REPEAT DB 0DH,0AH,'DO YOU WANT TO DO IT AGAIN? $'
    INVALID DB 0DH,0AH,'ILLEGAL CHARACTER - ENTER 0..9 or A..F: $'
    BANNED DB 0DH,0AH,'YOU HAVE ENTERED WRONG INPUT THREE TIMES. TRY
AGIAN LATER...$'
    CR_LF DB 0DH,0AH,'$'

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

        MOV CL,0
        FIRST:
            MOV AH,9
            LEA DX,CMD
            INT 21H
        SECOND:
            MOV AH,1
            INT 21H

            MOV BL,AL

            CMP BL,"A"
            JB THIRD

            CMP BL,"F"
            JA ILLEGAL

            JMP LETTER_DIGIT
        THIRD:
            CMP BL,"0"
            JB ILLEGAL

            CMP BL,"9"
            JA ILLEGAL

            JMP NUMERIC_DIGIT

        ILLEGAL:
            INC CL
            CMP CL,3
            JE LIMIT_REACHED
```

```
    MOV AH,9
    LEA DX,INVALID
    INT 21H

    JMP SECOND

NUMERIC_DIGIT:
    MOV AH,9
    LEA DX,DECIMAL
    INT 21H

    MOV AH,2
    MOV DL,BL
    INT 21H

    JMP CONTINUE

LETTER_DIGIT:
    MOV AH, 9
    LEA DX,DECIMAL
    INT 21H

    MOV AH,2
    MOV DL,31H
    INT 21H

    SUB BL,11H

    MOV DL,BL
    INT 21H

CONTINUE:
    MOV CL,0

    MOV AH,9
    LEA DX,CR_LF
    INT 21H

    LEA DX,REPEAT
    INT 21H

    MOV AH,1
    INT 21H

    CMP AL,"y"
    JE JUMP

    CMP AL,"Y"
    JE JUMP

    JMP FINISH
JUMP:
    LEA DX,CR_LF
    MOV AH,9
```

```
            INT 21H
            INT 21H

            JMP FIRST

        LIMIT_REACHED:
            LEA DX,CR_LF
            MOV AH,9
            INT 21H

            LEA DX,BANNED
            MOV AH,9
            INT 21H
        FINISH:
            MOV AH,4CH
            INT 21H
    MAIN ENDP
END MAIN
```

Output:

emulator screen (80x25 chars)

```
ENTER A HEX DIGIT: A
IN DECIMAL, IT IS: 10

DO YOU WANT TO DO IT AGAIN? Y

ENTER A HEX DIGIT: X
ILLEGAL CHARACTER - ENTER 0..9 or A..F: Y
ILLEGAL CHARACTER - ENTER 0..9 or A..F: Z

YOU HAVE ENTERED WRONG INPUT THREE TIMES. TRY AGIAN LATER...
```

Exercise: 12

Program Name: Write a program that reads a string of capital letters, ending with a carriage return and displays the longest sequence of consecutive alphabetically increasing capital letters read.

Source Code:

```
.MODEL SMALL
.STACK 100H

.DATA
    MSG_1 DB 'ENTER A STRING OF CAPITAL LETTERS: $'
    MSG_2 DB 0DH,0AH,'THE LONGEST CONSECUTIVELY INCREASING STRING
IS: $'
    ILLEGAL DB 0DH,0AH,'INVALID STRING OF CAPITAL LETTERS. TRY
AGAIN: $'

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

    LEA DX,MSG_1
    MOV AH,9
    INT 21H
    JMP INITIATE

    NOT_CAPITAL:
        LEA DX,ILLEGAL
        MOV AH,9
        INT 21H

    INITIATE:
        MOV AH,1
        INT 21H

        CMP AL,0DH
        JE NOT_CAPITAL
        CMP AL,41H
        JB NOT_CAPITAL
        CMP AL,5AH
        JA NOT_CAPITAL

        MOV BL,AL
        MOV BH,AL
        MOV DH,AL
        MOV DL,1
        MOV CL,1

    ENTER_CAPITAL:
        INT 21H

        CMP AL,0DH
        JE TERMINATE_INPUT
```

```
        CMP AL,41H
        JB NOT_CAPITAL
        CMP AL,5AH
        JA NOT_CAPITAL

        INC BL

        CMP AL,BL
        JNE CHECK_REPLACE

        INC CL
        JMP ENTER_CAPITAL

        CHECK_REPLACE:
        CMP CL,DL
        JLE BREAK_UPDATE_1

        MOV DH,BH
        MOV DL,CL

        BREAK_UPDATE_1:
        MOV BH,AL
        MOV BL,AL
        MOV CL,1
        JMP ENTER_CAPITAL

    TERMINATE_INPUT:
        CMP CL,DL
        JLE BREAK_UPDATE_2

        MOV DH,BH
        MOV DL,CL

    BREAK_UPDATE_2:
        MOV BX,DX

        LEA DX,MSG_2
        MOV AH,9
        INT 21H

        XOR CX,CX

        MOV CL,BL
        MOV DL,BH
        MOV AH,2

        DISPLAY:
            INT 21H
            INC DL
        LOOP DISPLAY
        MOV AH,4CH
        INT 21H
    MAIN ENDP
END MAIN
```

Output:

SCR emulator screen (80x25 chars)

```
ENTER A STRING OF CAPITAL LETTERS: FGHADEFGHC
THE LONGEST CONSECUTIVELY INCREASING STRING IS: DEFGH
```