**Practical Workbook**

**Microprocessor & Assembly Language**

****

|  |  |
| --- | --- |
| STUDENT NAME: | Muhammad Ahsan Khan |
| STUDENT ID: | **FA17-BSCS-0061** |
| SECTION: | **AM** |

**Mohammad Ali Jinnah University**

**Department of Computer Science**

**Lab # 08**

**EXERCISE:**

1. **Create labels which perform the following tasks:**
2. Loads 10h into Lower Byte of the Accumulator and 50h into the higher byte of the Accumulator.

L1:

mov al,10h

mov ah,50h

1. Adds values of Accumulator with the contents of Base Register (16 bit mode)

L2:

Add ax,bx

1. Multiplies the contents of Base register by 5

L3:

Mov ax, 5h

Mul bx

1. **From the labels created in Question 1, use Jump instructions to perform the following sequence of operations.**

* Compare the contents of AX and BX, if AX is greater than BX, Load DL with 100

Ans

.model small

.stack 100h

.data

msg db "Enter Number1$ "

msg1 db "Enter Number2$"

.code

main proc

mov ax,@data

mov ds,ax

mov ah,9h

mov dx,offset msg

int 21h

mov ah,1h

int 21h

mov bl,al

sub bl,30h

mov ah,9h

mov dx,offset msg1

int 21h

mov ah,1h

int 21h

mov bl,al

sub bl,30h

cmp al,bl

je greater

greater:

mov dl,100

hlt

* Perform Subtract AX from BX, if the result is negative, Load DL with 50h

Ans.

.model small

.stack 100h

.data

msg db "Enter Number1 Ax $ "

msg1 db "Enter Number2 BX $"

.code

main proc

mov ax,@data

mov ds,ax

mov ah,9h

lea dx,msg

int 21h

mov ah,1h

int 21h

mov bl,al

sub bl,30h

mov ah,9h

lea dx,msg1

int 21h

mov ah,1h

int 21h

mov bl,al

sub bl,30h

sub al,bl

cmp al,bl

je negative

negative:

mov dl,50h

hlt

1. **Implement an IF-Else code given below in EMU 8086 by making use of Labels and suitable Jump instructions:**

**IF (AL < BL)**

*{*

*Add contents of AL and BL*

*}*

**ELSE**

*{*

*Subtract AL from BL*

*}*

**(ii)**

**IF ( (AX + BX ) == 10)**

*{*

*Set All bits in DL*

*}*

**ELSE**

*{*

*Clear all bits in DL*

*}*

1. **Which of the flags are checked when following Jump instructions are executed:**

**JE \_\_\_\_\_ ZF\_\_\_\_\_\_\_**

**JNE \_\_\_\_ ZF\_\_\_\_\_**

**JNG \_\_\_\_CF , SF , AF \_\_\_\_\_\_\_\_\_**

**JZ \_\_\_\_ZF, PF\_\_\_\_\_**

**JC \_\_\_\_\_\_AF,CF\_\_\_\_\_\_\_**

**JNL \_\_\_\_\_SF=OF\_\_\_\_**

**JMP \_\_\_\_\_PF, AF\_\_\_\_\_\_\_\_**

**JB \_\_\_\_\_\_\_CF, SF, PF\_\_**

**Question 3**

cmp al,bl

jg l1

jl l2

ret

l1:

Add al,bl

l2

Sub al,bl

**Question 3**

add ax,bx

mov cx,10

cmp ax,cx

je l1

jne l2

l1:

mov dl,0FFh

exit

l2:

mov dl,0