Name: Saeed Ahmad Asad

ID 14273

Dep BS CS 5th

Subject Microprocessor and Assembly language

Assignment 04

Assignment No 04

Question 01

Ans:- inc val 2

Question 02

sub eax, val 3

Question 03

Ans:- mov ax, val 4
sub val 2, ax

Question 04

Ans:- CF = 0 , SE = 1

Question 05

Ans:- OF = 1 , SE = 1

Question 06

Ans:-

mov ax, 7FF0h
add al, 10h ; a : CF = 1 SE = 0
2F = 1 OF = 0
add ah, 1 ; b : CF = 0 SE = 1 2F = 0 OF = 1
add ax, 2 ; C : CF = 0 SE = 0 2F = 0 OF = 0

Question 07

Ans:-

```
mov esi , OFFSET my Bytes
mov al , [esi] ; a. AL = 10h
mov al , [esi+3] ; b. AL = 40h
mov esi , OFFSET my Word + 2 ; C. AX = 003Bh
mov ax , [esi]
mov edi , 8
mov edx , [myDoubles+edi] ; d. EDX = 3
mov edx , myDoubles [edi] ; e. EDX = 3
mov eax , [ebx+4] ; f. EAX = 2
```

Question 08

Ans:-

```
mov esi , OFFSET my Bytes
mov ax , [esi] ; a. Ax = 2010h
mov eax , DWORD PTR mywords ; b. EAx
= 003B 008Ah
xmov ax , [esi] x
xmov esi , myWORD x
mov esi , my pointer ;
mov ax , [esi+2] ; C. Ax = 0000
mov ax , [esi+6] ; d. Ax = 0000
mov ax , [esi-4] ; e. Ax = 0044h
```

Question 09

Ans:-

The program does not ~~support~~ stops, because the first loop instruction decrement ECX to Zero. The second loop instruction decrements ECX to FFFFFFFFh, causing the outer loop to repeat.

Question 10

Ans:-

```
DATA
        count DWORD ?
CODE
mov eax, 0
mov ecx, 10; outer loop counter
L1:
mov count, ecx
mov eax, 3
mov ecx, 5 inner loop counter
L2:
add eax, 5
~~mov ecx, 5~~
loop L2; repeat inner loop
mov ecx, count
loop L1 ; repeat outer loop
```

Question 11

Ans:-

```
mov ax, word ptr three
mov bx, word ptr three+2
mov three, bx
mov word ptr three+2, ax
```

Question 12

Ans:-

```
Xchg A, B
Xchg A, C
Xchg A, D
```

Question 13

Ans:- Parity flag (PF) will be set if there is an even number of 1 bit in the message byte.

* Parity flag (PF) will be zero for the message byte having an odd number of bits.

Code

```
mov al, 01110101b
add al, 00000000 ; AL = 01110101, PF=0
```

After the execution of the ADD instruction Al contain the value of the message byte since, there are five (5) odd number of ones in the AL register. Thus, PF = 0

Question 14

Ans:-

Any non-zero operand causes the carry flag to be set.

Example:-

* Data

```
val B BYTE 1,0
val C SBYTE -128
```

* Code

```
neg val B ; CF = 1, OF = 0
neg [val B+1] ; CF = 0, OF = 0
neg val C ; CF = 1, OF = 1
```

Question 15

```
mov al, 0F2h
add al, 1
```

Question 16

Ans:-

```
mov al, 0FFh
add al, 1 ; CF=1, AL = 00;
Try to go below zero:
mov al, 0
sub al, 1 : CF=1, AL=FF
```

Question 17

Ans:-

```
INCLUDE Irvine32.inc
.data
val1 SDWORD 8
val2 SDWORD -15
val3 SDWORD 20

Final val SDWORD ?

CODE
main PROC
mov eax val2
neg eax ; eax = -15
add eax, 7 ; -val2 +7
mov ebx, val3
add ebx, val ; val3 + val1
```

```
sub eax, ebx
mov finalval, eax
call DumpRegs ; display the registers
exit
main ENDP
END main
```

Question 18

- .data

```
intarray DWORD 10000h, 20000h
30000h, 40000h
```

- Code

```
main proc
mov edi, OFFSET intarray
mov ecx, LENGTHOF intarray
mov eax, 0
L1
add eax, [edi]
add edi, TYPE intarray
loop L1
invoke ExitProcess, 0
main endp
end main
```

Question 19

Ans:-

```
mov al , 80h
add al , 80h
```

Question 20

Ans:-

```
mov al, 0FFh
inc al
jz INC-overflow

mov b1, 1
dec b1
jz DEC-overflow

INC-overflow
DEC-overflow
```

Question 21

Ans:-

```
mov eax, TYPE myBytes ; a. 1
mov eax, LENGTHOF myBytes ; b. 4
mov eax, SIZEOF myBytes ; c. 4
mov eax, TYPE myWord ; d. 2
mov eax, LENGTHOF myWord ; e. 4
mov eax, SIZEOF myWord ; f. 8
mov eax, SIZEOF myString ; g. 5
```

Question 22

Ans:-

```
mov dx, WORDPTR myBytes
```

Question 23

Ans:-

```
mov al, BYTE PTR myWords +1
```

Question 24

Ans:-

```
mov eax, DWORD PTR myBytes
```

Question 25

Ans:-

```
myWords LABEL DWORD
```

myWords WORD 3 DUP(?), 2000h

• data
mov eax, myWords D

Question 26

Ans:-

• Data
myByte BYTE 10h, 20h, 30h, 40h
myWords WORD 3 DUP(?) 2000h
myWords DLABEL DWORD
mywords WORD 3 DUP(?) 2000h

• Code
mov eax, my words D

Question 27

Big Endian to Little Endian

• 386
model Flat, stdcall
• Stack 4096
Exit process PROTO, dw exit
code : DWORD

• Data
bigEndian BYTE 12h, 34h, 56h, 78h
littleEndian DWORD ?

• Code
main PROC
mov al, [bigEndian+3]
mov BYTE PTR [littleEndian], al
mov al, [bigEndian+2]
mov BYTE PTR [littleEndian+1], al
mov al, [bigEndian+1]
mov BYTE PTR [littleEndian+2], al
mov al [bigEndian]
mov BYTE PTR [littleEndian+3], al

INVOKE ExitProcess, 0
main ENDP
END main

Question 28

Ans:-
• 386
model flat, stdcall
stack 4096
Exit process PROTO, dwExitcode:
DWORD

- Data

```
array WORD 0,2,5,9,10
newArray DWORD LENGTHOF array
DUP(?)
```

- Code

```
main PROC
Mov ecx ,0
Mov Ax, [ESI]
Mov[EDI], EAX
ADD ESI, TYPE array
ADD EDI, TYPE newArray

loop L1

INVoke Exitprocess ,0
main ENDP
END main
```

Question 29

Ans:-

```
.386
.model flat, stdcall
.stack 4096
ExitProcess PROTO, dwExitCode:DWORD

.data
decimalArray DWORD 1,2,3,4,5,6,7,8
.code
main PROC
Mov ESI, OFFSET decimalArray
Mov EDI, OFFSET decimalArray
Mov ecx, LENGHOF decimalArray -1
L1:
ADD EDI, TYPE decimalArray
Loop L1
mov ecx, LENGTHOF decimalArray
L2:
Mov EAX, [ESI]
Mov EBX, [EDI]
XCHG EAX, EBX
Mov [ESI], EAX
Mov [EDI], EBX
```

```
Add ESI, TYPE decimalArray
Sub EdI, TYPE decimalArray
DEC ECX
Loop L2

INVOKE Exit process, 0
main ENDP
END main
```

Question 30

Ans:-

```
.386
.model flat, stdcall
.stack 4096
Exit process PROTO, DWExitCode:DWORD
.Data
Source BYTE "This is sourse string", 0
target BYTE SIZEOF Source DUP

.Code
main PROC
mov esi, 0
mov edi, LENGTHOF Source - 1
mov ecx, SIZEOF Source
L1:
mov eax, 0
```

```
mov al, source [esi]
mov target [edi], al
inc esi
dec edi
loop L1

INVOKE ExitProcess, 0
main ENDP
END main
```