National University of Computer & Emerging Sciences, Karachi.
FAST School of Computing,
Mid Term II, Fall 2022.
November 3, 2021, 10:00 am – 11:00 am

Course Code: EE 2003	**Course Name:** Computer Organization and Assembly Language
Instructors: Dr. Nouman M Durrani, Syed Asim Mehmood, Kashan Hussain, Atiya Jokhio, Rabia Ahmed, Aashir Mahboob, Aamir Ali, Zakir Hussain, Rukhsar Ali	
SOLUTION	

Time Allowed: 60 minutes. **Maximum Points**: 30 points

==========

Q. No. 1 Briefly answer each of the following: CLO: 2 [7 x 2 = 14 points]

(i) Discuss the difference between CMP and SUB instructions with one example instruction each.

CMP is non-destructive subtraction while SUB is destructive one.

(ii) Using shift and sub instructions to multiply a number Y_{10} by 57_{10}.

```
Mov eax,Y
SHL eax , 6   ; 64
Mov ebx, Y
Shl ebx, 2
Sub eax,ebx ;  60
Mov ebx, Y
Shl ebx, 1
Sub eax,ebx ;  58
Mov ebx, Y
Sub eax,ebx ;  57
Mov y,eax
```

(iii) What happens to the stack when (a) a call e.g. call Factorial and (b) a ret instruction is encountered? Also, elaborate your answer through a stack diagram.

a. Return address of next instruction is pushed on STACK.

b. Value POP from STACK is moved to EIP.

(iv) Which of the following instructions are illegal (if any)?

```
(A)   PUSH AL    ;  8 bit push not allowed
(B)   MOVZX AX, BX ;MOVZX is used for destination size > Source size
(C)   MOV AL, WORD PTR [EBX] ;16 bit value can't be moved in AL.
(D)   INC 1Ah  ; Immediate value can't be increased.
```

(v) List any four important uses of the runtime stacks in programs.

Local variables

Arguments passed to procedure.

Return addresses when call is used.

Temporary memory for value of nested loops.

(vi) Consider the following code:

```
MOV AX, 0H
MOV ECX, 0AH
DOLOOP:
DEC AX
LOOP DOLOOP
```

What is the value of the AX register after completion of the DOLOOP?

FFF6 H

(vii) In the following code sequence, show the value of AL (in hexadecimal format only) after each shift or rotate instruction has executed:

```
MOV AL,0D4H
SHL AL,3                    ; a. A0 h
MOV AL,0D4H
SAL AL,3                    ; b. A0 h
STC
MOV AL,0D4H
ROL AL,1                    ; c. A9 h
STC
MOV AL,0D4H
RCR AL,3                    ; d. 3A h
```

Q. No. 2 (i) Suppose the following data is received from a wireless sensor node operating in a smart building and is stored in EAX register, as shown in the below Figure. You are required to write an assembly language program with the corresponding data definition directives that would extract the data items and store them at memory locations Sequence_Number, Revision_Count, Status, and Sensor_Data. CLO: 2 [6 points]

a) Bits 0 to 11 reflect an integer Sequence_Number of the packet being sent.
b) Bits 12 – 14 show an integer Revision_Count of the packet.
c) Bit 15 is the Status of the sensor flag (0 – Forwarded Data and 1 – Sensed Data)
d) Bits 16 – 31 contain the Sensor_Data.

16 bits	1 bit	3 bits	12 bits
Sensor_Data	Status	Revision_Count	Sequence_Number

```
.Data

Sequence_Number  DWORD ?
Revision_Count DWORD ?
Status DWORD ?
 Sensor_Data. DWORD ?

.code
MOV EBX,EAX
AND EBX, 0FFFH
MOV Sequence_Number , EBX
MOV EBX,EAX
SHL EBX,17
SHR EBX,29
MOV Revision_Count ,EBX
MOV EBX,EAX
SHL EBX,16
SHR EBX,31
       MOV Status,EBX
MOV EBX,EAX
SHR EBX,16
MOV Sensor_Data, EBX
```

(ii) Convert the following High Level Language code to equivalent assembly language: CLO: 2 [6 points]

```
While (op1 <= op2)
{
    if (op1 > x || op1 > y)
    {
        z += 10;
    }
    else
    {
        z -= 10;
    }
    op1++;
}
```

```
.code
_While:
MOV EAX, op1
CMP EAX,op2
JNBE _EndWhile

CMP EAX, x
JA _if
CMP EAX, y
JNA _else
_if:
ADD z,10
JMP _endif
_else:
SUB z,10
_endif:

INC op1
JMP _While
_EndWhile:

exit
```

Q. No. 3 Give the contents of the following registers, along with the run-time stack, when the following instructions are executed. Initially, ESP = 00001FF8h. CLO: 4 [4 points]

```
                X1 DWORD 25H, 27H
                X2 DWORD 0
                MAIN PROC
                CALL Addition
11500000H       MOV X2, EAX             ; a) ESP:   00001FF8 h
                MAIN ENDP
                ADDITION PROC
115000A4H       MOV EBX, OFFSET X1
                PUSH DWORD PTR [EBX]
                ADD EBX, 4
                PUSH DWORD PTR [EBX+4]
                POP EAX
                POP EDX                 ; b) ESP:   00001FF4 h
                ADD EAX, EDX            ; c) EAX:   00000025 h
                RET                     ; d) EIP:   11500000 h
                ADDITION ENDP
```

ESP (Address)	Value	Comments
1FE7		
1FE8		
1FE9	00	X2
1FEA	00	
1FEB	00	
1FEC	00	
1FF1	25	X1[0*type X1]
1FF2	00	
1FF3	00	
1FF4	00	
1FF5	00	Return Address
1FF6	00	
1FF7	50	
1FF8	11	
1FE9		
1FEA		

Best of Luck