FAST- National University of Computer & Emerging Sciences, Karachi.
FAST School of Computing,
Mid Term II Examination, Fall 2021.
10th December, 2021, 10:00 am – 11:00 am

Course Code: EE 229	**Course Name:** Computer Organization and Assembly Language
Instructors:	
Student's Roll No:	**Section:**

Instructions:

- Except your Roll No and Section, DO NOT SOLVE anything on this paper.
- Return the question paper.
- Read each question completely before answering it. There are **3 questions on 3 pages.**
- In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper.
- All the answers must be solved according to the SEQUENCE given in the question paper, otherwise points will be deducted.
- This paper is subjective.
- Where asked for values, only provide the **hex-decimal** values.
- Problems needing iterations should be coded using iterative instructions. No points will be awarded otherwise.

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

==========

Q No. 1 Briefly answer each of the following: [6 x 2 + 1 x 4 = 16 points]

(i) Differentiate between the arithmetic shifts and logical shifts with one example instruction each.

(ii) Using shift and add instructions multiply a number X_{10} by 23_{10} . Assume that the result does not exceed the range of a16-bit register.

(iii) Why an assembly procedure cannot be implemented using JMP instructions?

(iv) Why stack parameters are considered more convenient than register parameters. Justify your answer with example.

(v) Write an assembly language program that adds two 128-bit numbers. You may define the data as a byte, word or doubleword array.

(vi) Provide the contents of registers where indicated, after execution of the following instructions:

```
MOV AL, 41h
MOV BL, 56h
TEST AL, 56h                    ; AL = _________
CMP AL,  BL
JNE L1
ADD AL, 32h                     ; AL = _________
L1:
MOV DX,0
MOV AX, 8003H
MOV CX,100H
DIV CX                          ; AX = _________ , DX = _________
```

(vii) Give the contents of the following registers, along with the run-time stack, when the following instructions are executed. Initially, ESP = 00001FF8h.

```
X1 DWORD 25H, 27H
MAIN PROC
                PUSH 6
                PUSH 5
                CALL P1
11500000H       MOV RESULT, EAX              ; ESP:      _____________
MAIN ENDP
    P1 PROC
     115000A4H  PUSH EBP
                MOV EBP, ESP                 ; EBP:      _____________
                MOV EAX, [EBP+8]
                ADD EAX, [EBP+12]            ; EAX:      _____________
                PUSH OFFSET X1
                PUSH OFFSET [X1+4]           ; ESP:      _____________
                POP ESI
                ADD [ESI], EAX               ; [ESI]:    _____________
                MOV ESP, EBP
                POP EBP
                RET 8                        ; EIP:      _____________
    P1 ENDP
```

Q. No. 3 (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 figure 1. 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 Data.

a) Bits 0 to 11 reflect the integer Sequence_Number of the packet being sent.
b) Bits 12 – 14 together form 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 contains the original data. [04 points]

16 bits	1 bit	3 bits	12 bits
Data	Status	Revision_ Count	Sequence_Number

Figure 1

(ii) Write an assembly language program that copy characters of the source string to a target string. The characters are stored in such a way that only a single instance of any character is stored. Initialize a source string to: "This is the test string. Do you want to continue?". Show the target string being stored in the memory. [04 points]

Q. No. 3 (i) Write an assembly language procedure MINIMUM that is called from the MAIN procedure to find the minimum MIN among X, Y and Z. The arguments are passed by value to the procedure MINIMUM using registers. The result is also returned in a register. Also, write the corresponding data definition directives. [04 points]

(ii) Implement the following pseudo-code in assembly language. Also, give the corresponding data definition directives: [04 points]

```
; All values are
; 32-bit signed integers


while (OP1 < OP2)
    {
       OP1++;
      if (OP3 == OP2)
          X = Y + 2;
      else
          X = Y + 10;
    }
```

STAY BRIGHT