National University of Computer and Emerging Sciences, Lahore Campus

Course:	COAL	Course Code:	EE2003
Program:	BSCS,BSDS,BSR	Semester:	Fall 2023
Duration:	1 Hour	Total Marks:	30
Paper Date:	28-Sept-2023	Page(s):	3
Section:	All	Roll No.	Solution
Exam:	Midterm - I		

Instruction/Notes: This is an open notes/book exam. Sharing notes and calculators is NOT ALLOWED. All the answers should be written in provided space on this paper. Rough sheets can be used but will not be collected and checked. In case of any ambiguity, make reasonable assumptions. Questions during exams are not allowed.

Question 1 [CLO 1, 2] [15 Marks]: Answer following short questions:

i. **[1 Mark]** How many number of address lines (no. of bits) are required to access 2GB memory? 31

ii. **[2 Marks]** SS = 0x012D, DS = 0x3F22 and BP = 0x00F2. Calculate the physical memory address of the destination operand for following statement: **Mov word [bp], 7**
Show your working to get credit.

P.A = SS * 10h + BP
= 0x013C2

```
  012D0
+ 00F2
  013C2
```

iii. **[3 Marks]** What will be the values of AX and BX registers in hex after the execution of the following piece of code?

```
[ORG 0x0100]
jmp start
num1: dd 0x7E945FA2
num2: dd 0xB2654104

start:
mov ax, [num1+2]
mov bx, [num2+1]
```

AX = 0x7E94

BX = 0x6541

iv. **[3 Marks]** Identify whether the following combinations for addressing are valid or not. Each part is independent of others.

	Valid/Invalid
Mov ax , [bx - si]	Invalid
Mov ax, [bx+ di + 0x0300]	Valid
Mov al, [bx + si]	Valid
Mov ah, [bh]	Invalid
Mov ax, [bh + bl]	Invalid
Mov ax, [0x0200]	Valid

v. **[3 Marks]** Write assembly language code that calculates 2's complement of a number in the AX register. Your code should not exceed 2 instructions. No credit will be given if code exceeds 2 instructions.

```
not ax
add ax,1
```

0612

vi. [3 Marks] Identify whether the following your working to get credit.

	Taken/Not Taken	Show your working below
Mov ax, -1 Mov bx, 0xFFFF Cmp ax,bx Je l1	Taken	-1 in 16-bits is 0xFFFF
Mov ax,0x1924 Mov cx, 0x0123 Sub cx,ax JO l1	Not Taken	0123 - 1924 E7FF Subtracting larger number from smaller number gives negative answer so no overflow.
Mov ax, FFFFh Mov bx, FFFFh Add ax, bx L1: Mov ax, 0 Mov bx, 0 Jnc L1	Not Taken	FFFF + FFFF 1FFFE carry generated move instructions does not affect flags so Carry Flag remains 1 so no jump taken.

Question 2 [CLO 3] [15 Marks]: Parity of a number is odd if the total number of 1s in its binary is odd. Following examples show different numbers, their binary and parity.

Number	0xA7	0xA3	0x94	0xFF	0x00
Binary	1010 0111	1010 0011	1001 0100	1111 1111	0000 0000
Total No of 1s	5	4	3	8	0
Parity	Odd	Even	Odd	Even	Even

Write a program that removes odd parity numbers from an array and keeps even parity numbers in start. A sample array before and after execution of required program is shown below:

Array Before Execution:	0xA7, 0xA3, 0x94, 0xFF, 0x00
Array After Execution:	0xA3, 0xFF, 0x00, 0x00, 0x00 ;odd parity numbers have [illegible] removed