National University of Computer and Emerging Sciences, Lahore Campus



Course: Program: **Duration:**

Section:

COAL BSCS,BSDS,BSR

1 Hour

Paper Date: 28-Sept-2023

All

Midterm - I

Course Code:

Semester:

EE2003 Fall 2023

Total Marks: Page(s):

30 3

Roll No.

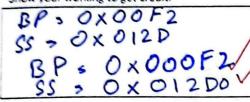
Instruction/Notes:

Exam: 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:

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

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.



BP+SS = D13C2 5013C2

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] imp start num1: dd 0x7E945FA2 num2: dd 0xB2654104 mov ax, [num1+2] 7 E95 mov bx, [num2+1] 654]

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 V		
Mov al, [bx + si]	Valid, V		
Mov ah, [bh]	Invalid		
Mov ax, [bh + bl]	Invalid V		
Mov ax, [0x0200]	Valid V		

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.

FAST School of Computing

Page 1

	Taken/Not Taken	Show your working below
Mov ax, -1 Mov bx, 0xFFFF Cmp ax,bx Je I1	Taken	ans-1 = Ox FFFF bx = Ox FFFF cmp an , bx
Mov ax,0x1924 Mov cx, 0x0123 Sub cx,ax JO I1	Not Takean	ans 0x1924 cn = 0x0123 Sub cn, an = 1801 00011000000000
Mov ax, FFFFh Mov bx, FFFFh Add ax, bx L1: Mov ax, 0 Mov bx, 0 Jnc L1	Not Taken	ax > FFFF bn = FFFF Addition > IFFFE There is a confy

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	OxFF	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
	0xA3, 0xFF, 0x00, 0x00, 0x00 ;odd parity numbers have been removed

FAST School of Computing

Page 2