National University of Computer and Emerging Sciences, Lahore Campus

0495

Course:	COAL	Course Code:	EE2003
Program:	BS(CS,SE)	Semester:	Fall 2024
Duration:	1 Hour	Total Marks:	40
Paper Date:	24-09-2024	Page(s):	4
Section:	All	Roll No.	23L-0700
Exam:	Midterm-I	Your Section:	BCS-3K

40

Instruction/Notes: This is an open book exam. Sharing book or calculators is NOT ALLOWED. PAPERS/NOTES ATTACHED WITH BOOK ARE NOT ALLOWED. 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] [10 Marks]: Short questions

i) **[1 Mark]** To address 1-MB of memory we need 20 bits of addressing, 1024 MB can be accessed using 30 bits of addressing.

ii) **[2 Mark]** For each of the instructions given below, **identify whether the instruction is valid or invalid:**

Instruction	Valid/Invalid	Instruction	Valid/Invalid
mov [bx+si+di],10	Invalid	mov word [mem1],al	Invalid
add word[mem1], word[mem2]	Invalid	mov word[bx],3	Valid

iii) **[4 Mark]** For following write the required code, assume that least significant bit is bit number 0:

a) Using one instruction, clear bits 3 and 4:
MOV AL, 0xFF
Assuming bit count starts from 0 and from LSB
AND AL, 0xE7

b) Using one instruction, toggle bits 2 and 6:
MOV AL, 0xAA
XOR AL, 0x44

c) Jump to "Label1" if bit 7 is set, without changing the value of destination operand:
MOV AL, 0x80
TEST AL, 0x80
JNZ Label1

d) Using one instruction, Take 1's complement of AX:
Mov ax, 0x0A0F
NOT AX

iv) **[3 Mark]** Given that AX is initialized with **0x4567** and BX is initialized with **0x789A**, what are the values of the flags after executing each of the following instructions and tell whether jump will be taken or not? **Show each and every step** to get marks.

Instruction	ZF	CF	SF	OF	Taken/Not-Taken	Working
add ax, 0x8765 jg l1	0	0	1	0	Not-Taken	JG checks OF = SF 0100 0101 0110 0111 + 1000 0111 0110 0101 1100 1100 1100 1100
sub ax, bx jbe l2	0	1	1	0	Taken	Assuming previous value of AX is not carried over 0100 0101 0110 0111 + 1000 0111 0110 0110 1100 1100 1100 1101

10

Question 2 [CLO 2] [10 Marks]:

i) **[5 Mark]** You are given memory addresses, their content and values of registers in hexadecimal. What would be the value of ax register after executing the instruction given below? Show your working to get full marks.

CS: 0x1D3F, DS: 0x1D3D, SS: 0x1D3E, BP: 0x011F, SI: 0x013F, DI: 0x0112, BX: 0x012D

Instruction: Mov ax, [cs:bx+si+12]

Value of AX after Execution of above instruction: 0x5566

Physical Memory Address	Memory Content (Hex)
0x1D666	02EF
0x1D658	035F
0x1D558	1111
0x1D668	6655
0x1D648	3344

Show Your working here:

$BX + SI + 12 = 0x012D + 0x013F + C = 0x0278$

~~CS: Effective address~~

$CS : offset = 0x1D3F0 + 0x00278 = 0x1D668$

ii) **[5 Mark]** The following is an incomplete listing file, answer the following questions. Show your working to get credit.

```
1                                       [org 0x100]
2  00000000                             jmp start
3
4  00000003 0500                        num1: dw 5
5  00000005 0000                        square: dw 0
6
7  00000007 A1[0300]                    start: mov ax, [num1]
8  0000000A 8B0E[0300]                  mov cx, [num1]
9
10 0000000E 0106[0500]                  loop1:add [square], ax
11 00000012 49                          dec cx
12 00000013 81F90000                    cmp cx, 0
13 00000017                             jg loop1
14
15 00000019 B8004C                      mov ax, 0x4C00
16 0000001C CD21                        int 0x21
```

Working:

$0x0000 + 0x0003 + 0x0004$ ← offset

$= 0x0007$

$0x0017 + 0x0002 = 0x0019$

$0x0019 - 0x000B$ ← offset

$= 0x000E$

Taking 2's complement of offset

$0xFFF5$

(a) What is the instruction in hexadecimal for the assembly instruction jmp start in the above code, given the opcode of the jmp is 0xE9? 0xE90400

(b) What is the instruction in hexadecimal for the assembly instruction jg loop1 in the above code, given the opcode of the jg is 0x7F? 0x7FF5

Considering Conditional jumps are short jumps

10

Question 3 [CLO 3] [20 Marks]: Write assembly language program to find Mode of an array i.e. the most frequent number in the array. Safely assume that input array will be sorted in ascending order and it will always have exactly one mode. Sample run is shown below:

Sample Run:

Example 1	Arr: 1,1,2,2,2,2,4,4,4,4,4	Mode = 4
Example 2	Arr: 1,1,2,2,2,2,3,4,4,4,5	Mode = 2

Solution:

```
[org 0x0100]
jmp start
arr: db 1,1,2,2,2,2,4,4,4,4,4
count: dw 11 ; There are total 11 elements in array
; add extra variables here, if required
mode: dw 0
modeCount: dw 0
start:
        MOV SI, 1
        MOV AL, [arr]
        MOV CX, 1

iter:
        CMP [arr+SI], AL
        JNZ change

        INC CX
        JMP update

change:
        MOV AL, [arr+SI]
        CMP CX, [modeCount]
        JNA nochangemode

        MOV [mode], AL
        MOV [modeCount], CX
```

; one is actual mode the other is temporary count

; starting from first index so I have track for the ; previous element
; AX is used for keeping track of mode
; CX keeps track of count for mode

; iterates through array and increments mode count ; which is in CX

; changes value of mode based on its ; count stored in CX

```
nochangemode:.                  ; changes to new element as mode in AX
        MOV  AL, [arr+SI]       ; and starts counting in CX again
        MOV  CX, 01

update:
        INC  SI                 ; loop break condition
        CMP  SI, [count]
        JNE  iter

        CMP  CX, [modeCount]    ; edge case to deal with count of
        JNA  terminate          ; last variable since I start with
        MOV  [mode], AL         ; index 1
        MOV  [modeCount], CX

terminate:
        MOV  AX, 0x4C00
        INT  0x21
```

20