National University of Computer and Emerging Sciences, Lahore Campus

Course:	COAL	Course Code:	EE2003
Program:	BSCS,BSDS,BSR	Semester:	Fall 2023
Duration:	3 Hour	Total Marks:	90
Paper Date:	18-Dec-2023	Page(s):	13
Section:	All	Roll No.	Solution
Exam:	Final	Your Section:	

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] [1+1+1+2+5 = 10 Marks]: Answer following questions:

(i) **Which of the following bus is unidirectional?**

a. Data Bus (b.) Address Bus c. Control Bus

(ii) **Which of the following instruction can change the value of IP register?**

a. Mov (b.) Call c. push

(iii) **Which of the following instruction does not change the flag register?**

a. cli b. std (c.) jc

(iv) **What is the last physical address of the segment 0xB432?** 0xC431F

Show your working to get credit:

B4320
FFFF
C431F

(v) **Content of Memory starting from offset 0x~~139D~~ 319D is given below. Write the updated memory content after execution of the following code. There is not syntax error in this code.**

```
mov bx, 0x319E
mov si, 4
mov cx, 0xAB01
mov [bx], cx
mov al, [0x31A4]
mov [bx+3], al
mov word [bx+si], 10
```

Memory Content before Execution of Code:

12	34	2D	1F	F5	89	9A	BC	A0

Memory Content after Execution of Code:

12	01	AB	1F	BC	0A	00	BC	A0
	0	0		0	0	0		

0286

Question 2 [CLO 2] [2+3+5+5+5 = 20 Marks]:

(i)

Following Code is trying to print '*' on top left cell of display memory with black background and white foreground (0x07). **Complete the code.** (Ascii of '*' is 0x2A.)

```
[ORG 0x0100]
mov ax, 0xb800
mov es, ax

mov byte[es:0], 0x 2A

mov byte[es:1], 0x 07

MOV  AX, 0x4C00          ; Terminate Program
INT  0x21
```

(ii)

Following code is trying to highlight the first five characters from top left of the video memory. Originally the whole screen is white foreground on blue background, without blinking. After highlight the specified characters should be blinking with red background, rest should remain the same. **Write the mask instruction to accomplish this task.**

```
[ORG 0x0100]
 mov ax, 0xb800
mov es, ax
mov si, 0
mov cx,5

label1:
mov ax, [es:si]

xor ah, 0xD0 ; Apply required mask in one instruction
mov [es:si], ax
add si,2
loop label1  ;runs the loop cx times

MOV  AX, 0x4C00          ; Terminate Program
INT  0x21
```

(iii)

Suppose when the following program loads, the Buffer is at offset 0x0103 in memory. CS=DS=ES=SS=0x19F5

What will be the content of 'buffer' after execution of code? Explain in one line only.

Nothing is changed.

At what Logical Address (Base:Offset) the above code is starting copying the data?

0x0103 : 0

```
[ORG 0x0100]
jmp start
buffer: times 80 dw 0 ;reserves 80 words in memory by placing zeros.

start:
mov ax, 0xb800
mov ds, ax
mov si,0

mov bx, buffer
mov es, bx
mov di, 0

mov cx, 80
cld

rep movsw

MOV  AX, 0x4C00          ; Terminate Program
INT  0x21
```

(iv) Suppose that AX= 1234h, BX= 5678h, CX = 9ABCh, and SP=0100h. **Write the contents of AX, BX, CX, and SP after executing the following instructions:**

AX: 0x9ABC

BX: 0x9ABC

CX: 0x5678

SP: 0xFE

```
PUSH AX
PUSH BX
XCHG AX, CX
POP CX
PUSH AX
POP BX
```

(v) We are required to write a program that disables space key on Dosbox. After running our program if user tries to write anything on dosbox, he should not be able to add space, rest of the keys should run normal with command prompt. Partial code is given; **complete the following program such that it fulfills the required functionality**

```
[org 0x0100]
jmp start
oldisr: dd 0

kbisr:      push ax
            in al, 0x60
;Add your code here (if required)

            cmp al, 0x39
            jne nomatch
            jmp exit

nomatch:    pop ax
            jmp far [cs:oldisr]

exit:       mov al, 0x20
            out 0x20, al
            pop ax
            iret

nomatch:    pop ax
            jmp far [cs:oldisr]
```

```
start:      xor ax, ax
            mov es, ax

            mov ax, [es:9*4]
            mov [oldisr], ax
            mov ax, [es:9*4+2]
            mov [oldisr+2], ax

            cli
            mov word [es:9*4], kbisr
            mov [es:9*4+2], cs
            sti
;Add your code here (if required)

            jmp TSR

Unhook:
            mov ax, [oldisr]
            mov bx, [oldisr+2]
            cli
            mov [es:9*4], ax
            mov [es:9*4+2], bx
            sti

            mov ax, 0x4c00
            int 0x21
TSR:
            mov dx, start
            add dx, 15
            mov cl, 4
            shr dx, cl

            mov ax, 0x3100
            int 0x21
```

Question 3 [CLO 4] [10+10+10 = 30 Marks]:
Q3 Part (A) [2x5 = 10 Marks]

Consider the cache has 4 blocks/indexes and we have following memory references:

5, 12, 13, 17, 4, 12, 13, 6, 4

What is the hit ratio for the following caches? Show your working below to get credit.

i.	**Fully/Simple Associative with FIFO Replacement Algorithm**	3/9
ii.	**Fully/Simple Associative with LRU Replacement Algorithm**	3/9
iii.	**Direct mapping**	0
iv.	**2-Way Set Associative using LRU Replacement Algorithm**	3/9

v. Consider a pipelined and a non-pipelined architecture. In both, one instruction completes in 6 stages and each stage takes 2 µs.

Fill-in following information:

	Clock Cycle Time	**Frequency**
Pipelined Architecture	2 µs	$1/2\times10^{-6} = 5\times10^{5}$ Hz
Non-Pipelined Architecture	12 µs	$1/12\times10^{-6} = 83,333$ Hz

Show your working here:

Q3 Part (B) [2+2+6 = 10 Marks]

Assume we have declared an array in memory of 200 bytes and we have an (fully) associative cache of size 8 bytes and each cache index stores 1 byte of data. **For the following code, answer the questions below:**

```
mov ax, 0
mov bx, 0
loop1: add al, [array1+bx]
add al, [array1+bx+1]
add bx, 1
cmp bx, 100
jne loop1
```

i. **What is the total number of memory references in the code?** 200

ii. **If we have total memory size of 1Mb, how many bits are required for the tag in the cache?** 20

iii. **What is the hit rate for the cache for the above code? Use FIFO replacement algorithm. Assume that the cache is initially empty.** 99/200

SHOW YOUR WORKING HERE TO GET CREDIT:

Q3 Part (C) - Pipelining Question *FOR ALL THE SECTIONS EXCEPT BCS-3A, BCS-3B, BCS-3C*:

Let us consider the following decomposition of the instruction processing. All stages take equal amount of time i.e 1μs

Fetch Instruction (FI): Read the next expected instruction into a buffer.
Decode Instruction (DI): Determine the opcode and the operand specifiers.
Fetch Operands (FO): Calculate the effective address of each source operand and fetch each operand from the memory. Operand in registers need not to be fetched.
Execute Instruction (EI): Perform the indicated operation and store the result if any, in the specified destination operand location.
Write Operand (WO): Store the result in memory.
Following is a set of instructions. Their implementation through pipelining has some control hazards. You have to solve those hazards by using Branch prediction (Predict taken) method. Also calculate latency and throughput.
Note: Normal execution will be done in case of simple instructions. 'Jump instructions calculation is done in the execution phase.

Set of instructions

I1: jmp start
I2: n1: db 10,20
I3: start: mov ax, 1
I4: mov cx, 8
I5: jcxz I1
I6: add ax, cx
I7: mov bx, 8
I8: l1: add cx, bx
I9: add ax, 4
I10: add si, 4

Solution:

Instruction No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
I1	FI	DI	FO	EX	WO															
I3		FI	DI	FO	EX	WO														
I4		FI	~~FI~~ DI	~~DI FO~~ FO	~~EO EX~~ EX	WO														
I5				FI	DI	FO	EX	~~WO~~ WO												
I8					FI	DI	FO	~~EX WO~~												
I9						FI	DI	~~FO EX WO~~												
I10							FI													
I6								FI	DI	FO	EX	WO								
I7									FI	DI	FO	EX	WO							

Latency: 5 ,15

Throughput: 10/16

I8 FI DI FO EX WO
I9 FI DI FO EX WO
I10 FI DI FO EX WO