Roll Number: _____ Section: _____

National University of Computer and Emerging Sciences, Lahore Campus

Course:	**COAL**	**Course Code:**	
Program:	**BS(Computer Science)**	**Semester:**	**Fall 2020**
Duration:	**3 Hours**	**Total Marks:**	**110**
Paper Date:	**22-02-2021**	**Weight**	**45%**
Section:	**All (Your section _____)**	**Page(s):**	**11**
Exam:	**Final**	**Roll. No**	

Instruction/Notes: This is an open note/book exam. 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, take reasonable assumption. Questions during exam are not allowed. **ATTEMPT ALL QUESTIONS UNLESS GIVEN EXPLICIT INSTRUCTIONS FOR YOUR SECTION.**

Question 1: Short Questions [10 x 5 = 50 Marks]

I. What will be the content of memory (in HEX) before and after the execution of the following code?

```
[org 0x0100]
jmp start
num1: db 0xA
dw 0x1234
dd 0xABCDEF09

start:        mov ax, [num1+5]
              add ax, [num1+2]
              mov [num1], ax
              mov ax, 0x4c00 ; terminate program
              int 0x21
```

Memory Configuration BEFORE Execution:

Address	Num1+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
Content										

Memory Configuration AFTER Execution:

Address	Num1+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
Content										

II. Write Assembly language code for the following

if dx <= cx, ax = 1, else ax = 2

Roll Number: _____ Section: _____

III. Find the values of Sign, Carry, Overflow, and Zero flag given the values of registers and operations.

	CF	OF	SF	ZF

MOV AL , 10
MOV BL, 20
ADD AL, 10
SUB AL, BL				

MOV AL, 66H
MOV BL, 1A
ADD BL, AL				

IV. A function takes three parameters P1, P2, P3, returns two values Output1, Output2, declares one local variable (local1), and saves AX, BX and CX registers. **What will be the configuration of the stack for this function after pushing all these variables and registers? Fill in the given stack. Also specify where BP and SP should be pointing?**

Solution of Part (IV)

Stack

V. Fill in the blanks

a. If SP = 1735h and SS:SP = 37B35h,
SS = 0x_____

b. If SS = FE07h and SS:BP = FF565h,
BP = 0x_____

c. A parallel port is mapped to interrupt number 0x0F, **address of its service routine (handler) can be found at**
_____

d. Timer tick comes _____ times per second.

Roll Number: _____ Section: _____

VI. The following code is trying to copy the arrays 0th to 9th element to 1st to 10th elements. For example if array initially 1,2,3,4,5,6,7,8,9,10,11,12 after code runs it should be 1,1,2,3,4,5,6,7,8,9,10,12 But there is a mistake in this code. **Identify and correct the mistake**. (Hint: Source and Destination are overlapping)

Code with Mistakes	Corrections (Write Correct Lines Only)
00 push ds 01 pop es 02 Mov cx, 10 03 Mov si, array 04 Mov di, array+1 05 cld 06 Rep movsb 07 mov ax, 4c00h 08 int 21h 09 array: db 1,2,3,4,5,6,7,8,9,10,11,12	

VII. Dry run the following code and find the values of given registers and arrays. Also, write in one line what this code is doing. Assume address of Array is 103h

```
01 Jmp start:
02    array: db 5,6,3,7,8,9,1
03    len_of_array: db 7
04 start:
05    mov cx, [len_of_array]
06    push ds
07    pop es
08    mov si, array
09    mov di, array
10    cld
11    loop1:
12       lodsb
13       add al, 30
14       stosb
15       loop
```

Array:_____

SI:_____

DI:_____

What is this code doing?

_____

VIII. The following keyboard custom ISR is trying to block the numbers from 0 to 9 from the keyboard, the rest of the keys should work as before. **Complete the code of KBSIR.** (Only write the lines that are to be added in kbISR, don't write the whole code again. Assume that kbsir has been hooked by start and oldisr variable saves the values of old keyboard ISR. DON'T WRITE START CODE)

Code	Additions in Code
01 oldisr: dd ; stores old isr 02 kbisr: 03 push ax 04 ;read a char from keyboard 05 in ax, 0x60; asci in ah and scan code in al 06 cmp ah, 30h; asci of 0 07 JL exit 08 cmp ah,39h ; asci of 9 09 JG exit 10 exit: 11 mov al, 0x20	

```
12        out 0x20, al
   13       pop ax
```

IX. **[For All Sections Except Section E]** Consider 5 cache levels. It takes 0.001us to read from cache L1, 0.0015µs to read from L2, 0.01µs from L3, 0.05 µs from L4 and 0.3 µs from L5. Data is found in L1 70% of time, 15% of the time in L2, 10% of the time in L3, 3% of the time in L4 and 2% of the time in L5. What is the average access time?

Part (IX) [for Section E ONLY] Following program is trying to print the right aligned "This is COAL Final" in the last row of video memory using BIOS service 0x10. **Modify the code to produce required output (sample output is shown below). Clearly highlight the lines/segment having errors and write only modified (neat) code in the same space.**

Sample Output:

```
                                                                    This is COAL Final
```

```
[org 0x0100]
jmp start
message: db 'Hello World'
message2: db 'This is COAL Final'
start:        mov ah, 0x13
              mov al, 1
              mov bh, 0
              mov bl, 7
              mov cx, 11
              mov dx, 0x0A03
              push cs
              pop ds
              mov si, message
              INT 0x10
              mov ax, 0x4c00 ; terminate program
              int 0x21
```

X. **[For All Sections Except Section E]** For each of the following questions, you are given some information and you are required to find one value. In case the given information is not enough to find the values, write "Info not enough" in answer and specify which info is missing.

	Question	Answer
i.	Given the clock cycle of a pipelined processor is1µs, what is the freq?	

ii. Given the freq of a non-pipelined processor is x what is the throughput of this processor for n instructions?	
iii. Given the freq of a pipelined processor is y what the throughput of this processor for n instructions is?	
iv. A pipelined processor with 5 stages, stage 1 to 5 takes 2 µs, 2.5µs, 2.6µs, and 0.5µs, Latch time is 0.1µs. What is the Clock Cycle?	
v. Consider the following instructions in a 6 stage pipeline. Assume that there are no data hazards or resource hazards. If the 1st instruction is a conditional branching to I6, how many cycles will be wasted before the branch?	

I1	FI	DI	CO	FO	EI	WO					
I2		FI	DI	CO	FO	EI	WO				
I3			FI	DI	CO	FO	EI	WO			
I4				FI	DI	CO	FO	EI	WO		
I5					FI	DI	CO	FO	EI	WO	
I6						FI	DI	CO	FO	EI	WO

Part (X) [<u>For Section E ONLY</u>] Consider two different processors P1 and P2 executing the same instruction set. P1 has a 3 GHz clock rate and a CPI of 1.5. P2 has a 2.5 GHz clock rate and a CPI of 1.0.

a. Which processor has better performance expressed in instructions per second? Show your calculations to get credit.

b. If each processor executes a program in 10 seconds, find the number of cycles and the number of instructions.

Solution:

Question 2 [12 Marks]: It takes 10µs to complete one instruction in a non-pipelined processor. We were able to convert the circuit to a 5 stage pipeline processor. Stage 1 to 5 take 2µs, 1.5µs, 3µs, 2µs, 1.5µs resp. Latch time is 1µs. **Calculate the following values for pipeline and non-pipelined processor** (Write the answer in the given table) **Note for Section E:** Ignore Latch Time.

Value	Non-Pipeline	Pipeline
Clock Cycle		
Frequency (clock speed)		
Latency (Time it takes to complete one instruction)		
Throughput for 100 instructions **[For all Section except E]**		
Time required to complete 100 instructions **[For Section E ONLY]**		
Speedup of pipeline processor for 1 instruction		
Speedup of pipeline processor for 100 instructions		

Question 3 [for all sections except Section E] [8 Marks]

I1	FI	DI	CO	FO	EI	WO					
I2		FI	DI	CO	FO	EI	WO				
I3			FI	DI	CO	FO	EI	WO			
I4				FI	DI	CO	FO	EI	WO		
I5					FI	DI	CO	FO	EI	WO	
I6						FI	DI	CO	FO	EI	WO

Consider the following instructions in the 6 stage pipeline. Assume that there are no data hazards or control hazards. Given the following memory reads are required by different stages of each instruction add the stalls to omit resources hazards. Fill the table given below

- FI of all instructions read from memory.
- FO of only I1, I3, and I6 from memory.
- WO of only I2 and I3, 14 is from memory.

Rest of the FO and WO are from registers and have no conflict.

Roll Number: _____ Section: _____

**Note the FI stands for Fetch Instructions, DI for Decode Instruction CO for Calculate operands, FO for Fetch operands, EI for Execute instruction, and WO for write operands.*

Question 3 [<u>For Section E ONLY</u>] [8 Marks] For the code segment given below, fill-in the following pipeline diagram for optimized pipelined MIPS Architecture (with all the hazards control implementation) as we have studied in class. Properly indicate Forwarding or Stalls (if required). In case of forwarding, clearly mention the name of the operand that needs forwarding.

Lw $1, 100 ($2)
Lw $3, 200 ($2)
Add $4, $1, $3
Sub $5, $4, $6

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10	CC11

Roll Number: _____ Section: _____

Roll Number: _____ Section: _____

Question 4 - Cache [10+10 = 20 Marks]: Consider a sequence of memory address references given below. In the sequence, each word address is provided in both the decimal and binary formats. Below each address, the relative time at which these references occur is also listed. Memory contents and addresses are shown in the second table.

Data Access Sequence		
Time	**Address Decimal**	**Aaddress Binary**
1	28	00 01 11 00
2	40	00 10 10 00
3	36	00 10 01 00
4	16	00 01 00 00
5	52	00 11 01 00
6	36	00 10 01 00
7	8	00 00 10 00
8	12	00 00 11 00
9	36	00 10 01 00
10	40	00 10 10 00
11	36	00 10 01 00
12	40	00 10 10 00

Memory		
Address Decimal	**Address Binary**	**Data**
8	00 00 10 00	25
12	00 00 11 00	45
16	00 01 00 00	4
20	00 01 01 00	83
24	00 01 10 00	12
28	00 01 11 00	39
32	00 10 00 00	53
36	00 10 01 00	52
40	00 10 10 00	96
44	00 10 11 00	63
48	00 11 00 00	57
52	00 11 01 00	236
56	00 11 10 00	263
60	00 11 11 00	55

Now consider two different 8-word caches shown below. Assume that each of the caches was used independently to facilitate memory access for the sequence above. For each cache type, assume that the cache is initially empty. Assume that the least-recently used (LRU) scheme is used where appropriate. Also, when inserting an element into the cache, if there are multiple empty slots for one index, you should insert the new element into the left-most slot (first available slot).

Part (A): Use the **direct-mapped cache** to facilitate memory access for the memory sequence above. You should fill in the binary form of the Tag values. **Show the final contents of the cache in the table below, and compute the hit rate and miss rate.**

Index	**Cache**		
	V	**Tag**	**Data**
0			
1			
2			
3			
4			
5			
6			

Direct-Mapped Cache Summary

Hit Rate: _____

Miss Rate: _____

7				

Part (B): Use the **2-way set associative cache** to facilitate memory access for the memory sequence above. You should fill in the binary form of the Tag values. **Show the final contents of the cache in the table below, and compute the hit rate and miss rate.**

Set	2-way Set Associative Cache					
	V	Tag	Data	V	Tag	Data
0						
1						
2						
3						

2-way Set Associative Cache Summary

Hit Rate: _____

Miss Rate: _____

Question 5 – Programming [20 Marks]: You are required to implement a game CollectCoins with following requirements:

- A star '*' will keep moving on the screen in Up, Down, Left or Right direction until the game is over, the '*' changes its position after every clock/timer tick.
- Initially '*' will be moving towards right i.e., the '*' will start from a starting position (0, 0) and it will keep moving from the first to last column of first row until the user changes its direction.
- Direction of the '*' can be changed by using Up, Down, Left or Right arrow keys (you may assume any scan codes for these keys).
- Downward movement means the '*' will keep moving from first to last row of same column until user changes its direction.
- Due to time constraint, our game is not supporting Left and Up movement, so **you are NOT required to handle Left and Up movement.**
- Assume there are randomly placed Green and Red cells (coins) on the screen at the start of your game (you are NOT required to place green and red cells on screen). If the '*' crosses a green cell, one point is added to your score and the cell is cleared (i.e. it becomes black). Displaying score on screen is also NOT required.
- If the '*' hits a Red cell, the game is over and it terminates successfully i.e. your program will terminate and DOSBOX and command prompt will run normally.

Important Note: Assume that you are already given a subroutine '*CalculateOffset*' that takes row and col as parameter, calculates position offset and saves it in 'DI' register. You may call it where required, do not write this function again. Instead of writing code to save state of all registers just comment "; pushing all registers here" and similarly just comment for restoring

registers. If required, just call functions given in book, do not write those functions again. **Variables row, col and direction are already given to handle position and directions, do not use any other variable for this purpose. Comment your code properly.**

```
[org 0x0100]
jmp start
; Use only following variables for position and
; direction
row: dw 0    ; initial position row = 0
col: dw 0    ; initial position col = 0
dir: dw 'R'  ; initially direction is Right
; declare other variables here (if reqd.)



; Write your start functionality here
start:
```

```
KBISR:
;Write your KBISR here
```


`; Write your Timer routine here` `Timer:`	`;Space for your code`

Roll Number: _____ Section: _____
