United International University

Department of Computer Science and Engineering

CSE-3313: Computer Architecture

Final Examination: Summer 2024

Total Marks: 50   Time: 2 hours

Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.

Answer all questions. Numbers to the right of the questions denote their marks.

1. (a) You have to modify the block diagram for the single-cycle data path so that it can execute the following instruction **'jumpReg'**. Also, write the control unit values for this instruction: **jumpReg $s2**. This instruction will read the values of the source register(**rs**), multiply the value by 4, and store the result to the PC register. The machine code format for the instruction is given below table-1, where rs will contain the index numbers of $s2 registers. [8]

op	rs	rt	rd	shamt	funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

Table 1: machine code format of instruction **jumpReg**

(b) Modify the block diagram for a single-cycle data path so that it can execute the following j-type instruction **'jp 2024'**. Note that, the job of the given instruction is to jump to the given address if the value of the given address is greater than the current PC values. Write down the control unit values as well. [8]

2. Consider a processor that goes through the following six stages while executing an instruction.

Instruction Fetch (IF)	120 ns
Instruction Decode (ID)	100 ns
Execution (EXE)	180 ns
Memory Read (MEMR)	200 ns
Memory Write (MEMW)	230 ns
Write Back (WB)	170 ns

Table 2: Six stages pipeline

(a) Find out the total time for the given code below in a single cycle method. (Assume $s1 = 3) [3]

Listing 1: Code for 2(a)

```
addi $s3, $zero, 0
addi $s0, $zero, 0
LOOP:
    slt $t0, $s0, $s1
    beq $t0, $zero, END
    sll $t0, $s0, 2
    add $t0, $t0, $s2
    lw $t1, 0 ($t0)
    add $s3, $s3, $t1
    addi $s0, $s0, 1
    j LOOP
END:
```

Listing 2: Code for 2(b), 2(c), and 2(d)

```
sw $t0, 0($s0)
lw $t1, 4($s0)
add $t2, $t0, $t1
add $t3, $t1, $t2
slt $t4, $t3, $zero
sw $t4, 8($s0)
```

(b) Implement the basic pipeline method and find out the total time using a timing diagram. [5]

(c) What type of pipelining hazards occurred in the given code segment? What can you do to mitigate these hazards? Can you reduce the total time found in 2 (b)? Justify your answer with a timing diagram. [5]

(d) Alex thinks he can find the optimal solution for the given code. He also boasts that his solution doesn't have any stalls. Do you agree with Alex? Why or why not? (No need to show any timing diagram) [5]

3. Suppose you are to design a **512kb** direct-mapped cache for a CPU having **32-bit** architecture. The cache should store **4 words** in each data block. Memory is byte-addressable.

Consider the cache is initially empty and looks as follows (here, "..." represents imaginary values):

index	**V**	**D**	**Tag**	**Data**
0	0	0	...	...
1	0	1	...	...
2	0	0	...	...
3	0	1	...	...
4	0	0	...	...
5	0	1	...	...
6	0	0	...	...
7	0	1	...	...
8	0	0	...	...
. . .	. . .	. . .	. . .	. . .
$2^n - 1$	0	0	...	...

Now, based on this information, answer questions 3(a)-3(c).

(a) Find out the actual size of the cache. [4]

(b) Find out if it's a cache hit or miss if the CPU needs to access the following addresses sequentially.
32, 64, 67, 88, 92, 7, 19, 14.
From that calculate the cache hit and miss rate. No need to consider the dirty bits in this calculation. [8]

(c) State the principle of locality for cache. Explain the concept with appropriate examples.
Then, differentiate between "write-through" and "write-back" policies of caching. [2+2]