United International University

Department of Computer Science and Engineering

CSE-3313: Computer Architecture

Final Examination: Fall 2024

Total Marks: 40 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 and write the control unit values for this instruction: **'cat $s0, $s1, $s2'**. The job of the given instruction is to read the value of the source registers **(rs, rt)** and add them. After that, the sum will be **multiplied by 64** to get an **address**. We need to read the **value of that address** and write it to the destination register **rd**. The machine code format for the given instruction is given below, where **rs, rt, and rd** will contain the index numbers of **$s1, $s2, and $s0** registers, respectively. [5]

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 **cat**

(b) Modify the block diagram for a single-cycle data path so that it can execute the following instruction **'fa $s1, $s0'**. The instruction reads the value of the source register **rs** and **resets(makes 0) all the odd bits of the value**. Then the result is written into the destination register **rd**. Write down the control unit values as well. The machine code format for the given instruction is given below, where **rs and rd** will contain the index numbers of **$s0 and $s1** registers, respectively. [5]

op	rs	rd	blank(not used)
6 bits	5 bits	5 bits	16 bits

Table 2: Machine code format of instruction **fa**

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

Instruction Fetch (IF)	120 ns
Instruction Decode (ID)	70 ns
Register Read (RR)	150 ns
Execution (EXE)	180 ns
Memory Read (MEMR)	190 ns
Memory Write (MEMW)	200 ns
Write Back (WB)	170 ns

Table 3: Seven stages pipeline

(a) Find out the total time for the given code below in a single cycle method. (Assume $s0 = 8 and 0($s0) holds the value 0) [3]

Listing 1: Code for 2(a)

```
lw $t0, 0($s0)
add $s3, $zero, $t0
addi $s1, $zero, 20
addi $s2, $zero, 1
LOOP:
    slt $t1, $s2, $s1
    beq $t1, $zero, END
    sub $t2, $s2, $s1
    add $s3, $s3, $t2
    sll $s2, $s2, 1
    srl $s1, $s1, 1
    j LOOP
END:
```

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

```
sw $t0, 0($s0)
lw $t1, 4($s0)
lw $t2, 8($s0)
add $t2, $t1, $t2
subi $t2, $t2, 10
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? Justify your answer with a timing diagram. [5]

(d) Suppose the memory cannot be accessed for reading or writing while another instruction is using it. Specifically, if any instruction is in the MEMR (Memory Read) stage, no other instruction can perform a memory read or write at the same time. Similarly, if an instruction is in the MEMW (Memory Write) stage, no other instruction can access the memory for reading or writing during that cycle. Based on this restriction, draw the timing diagram for the first three instructions of the given code and find the total time required for this.[2]

3. A 16-bit 2GHz processor has a cache memory of 128 bytes, which is byte-addressable, and a line size of 2 words. Each word contains 2 bytes of data. Note that cache memory contains two flag bits: a valid bit and a dirty bit. Based on the information, answer the following questions:

(a) **Find** out index bits and tag bits. [2]

(b) **Calculate** the actual size of the cache (in Bytes). [3]

(c) **Find** the cache miss ratio for the given MIPS code. Assume that, initially cache is empty. [8]

Listing 3: Code for 3(c)

```
    addi $s0, $zero, 1
    addi $s1, $zero, 20
    addi $s2, $zero, 97
    LOOP:
        slt $t0, $s0, $s1
        beq $t0, $zero, END
        lw $t1, 0($s2)
        addi $s2, $s2, 2
        lw $t2, 0($s2)
        add $t3, $t1, $t2
        sw $t3, 4($s2)
        addi $s2, $s2, 1
        sll $s0, $s0, 1
        j LOOP
    END:
```

(d) The hit time is 20 cycles and a miss penalty is 500 cycles. **Calculate** Average Memory Access Time. [2]