United International University
Department of Computer Science and Engineering

CSE-3313: Computer Architecture
Final Examination: Summer 2023
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 so that it can execute the following instruction 'pw'. Also, write the control unit values for this instruction. **pw $s0, $s1,$s2**. This instruction adds the values of the two source registers(**rs, rt**) to find the address. Then it reads the value of the address and divides by 4 and places that result in the destination register(**rd**). The machine code format for the instruction is given below table-1, where rs, rt, and rd will contain the index numbers of $s1, $s2, and $s0 registers, respectively. [5]

op	rs	rt	rd
6 bits	5 bits	5 bits	16 bits

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

(b) You have to modify the block diagram for the single-cycle data path and write the control unit values for this instruction **'ta'**. The job of the given instruction is to read the value of the source registers (**$s0, $s1**) and subtract then multiply it by 4. It will produce an address where we have to store the value of the register **$t0**. **ta $t0, $s0, $s1. The machine code format for the given instruction is given below Table-2.** [5]

op	rs	rt	rn
0	$s0	$s1	$to
6 bits	5 bits	5 bits	16 bits

Table 2: machine code format of instruction **ta**

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

Instruction Fetch (IF)	250
Instruction Decode(ID)	100
Register Read(RR)	100
Execution(EXE)	150
Memory Write(MEMW)	200
Memory Read(MEMR)	200
Write Back(WB)	100

Table 3: Seven stages pipeline

(a) Find out the total time for the given code below in a single cycle method. [2]

```
1            LOOP:
2            addi $t0,$zero, 10
3            beq $t0,20,EXIT
4            or $s0, $s1,$s2
5            addi $t0,$t0,1
6            j LOOP
7            EXIT:
8            add $s0, $s1,$s2
9            and $s1,$s2,$zero
```

(b) Using the pipeline method please find out the total time using a timing diagram. And if possible reduce the number of cycles for the below given code. [5]

```
addi $s0, $zero,2000
sw, $t0,20($s0)
add $t9,$t0,$t3
lw $t3, 40($s2)
or $t7,$t6,$t3
sub $s4,$s4, $zero
sw $t5, 12($s2)
nor $s6, $s6, $s7
```

(c) Now for the code snippet given in 2b please find out an optimal solution(By re-ordering the code) where stalls will be minimum for the given scenario in the table-4. Find out the total time. No need to show a timing diagram. [5]

Instruction	R-Type	lw
Stalls	2 cycle	4 cycles

Table 4: For question 2c

(d) Can you please point out the advantages/ disadvantages these two stages(Memory Read and Memory Write in place of only Memory Stage) will bring out compared to the five stage method of pipelining? Explain. [3]

3. A 32-bit CPU has a clock rate of 2 GHz and a 32 KB direct mapped cache with 128-byte block size. Suppose A is a two-dimensional array of size 64x64 with elements that occupy 8 bytes each. Consider the following C code segments and direct mapped table for your convenience. **Initially, the array A is not in the cache and i,j, and x are in the registers.**

```
for(i=0;i<64;i++){
    for(j=0;j<64;j++){
        x += A[i][j];
    }
}
```

Index	V	Tag	Data [128 bytes]
0	0	?	?
1	0	?	?
...	-	-	-
...	-	-	-
$2^n - 1$	-	-	-

Table 5: Direct mapped table

(a) Calculate the Tag directory size of the direct mapped cache. [3]

(b) Find the cache miss ratio for the above code. [4]

(c) Determine the type of principle of locality used in the above code segments with an explanation. [2]

(d) If hit time requires 4 cycles and miss penalty is 20 cycles, calculate the average memory access time for question 3b. [3]

(e) Find the cache miss ratio in the given scenario, but instead of the above code segments, the code segments will be the following: [3]

```
for(i=0;i<64;i++){
    for(j=0;j<64;j++){
        x += A[j][i];
    }
}
```