United International University
Department of Computer Science and Engineering

CSE-3313: Computer Architecture

Midterm Examination: Spring 2024

Total Marks: 30   Time: 1 hour and 30 minutes

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. Consider the following MIPS code:

```
        block:
            add $sp, $sp, -8
            sw $ra, $zero($sp)
            sw $s0, 4($sp)

            addi $v0, 0, 0
            addi $s0, $zero, 1

        loop:
            slti $t0, $a0, s0
            bne $t0, $zero, end_loop
            add $v0, $v0, $s0
            addi $s0, $s0, 1
            jump loop

        end_loop:
            lw $ra, 4($sp)
            lw $s0, 8($sp)
            addi $sp, $sp, 12
            jr $ra
```

[N.B. **You should assume the value of $a0** = 3]

(a) **Find** the errors and fix them in the above code. [4]

(b) As you get error-free code from 1a, **Write** the value of $vo and $s0 registers after executing the program.[2]

(c) Consider a CPU with a $4GHz$ clock rate and CPI in the following table-1. **Calculate** the CPU time for executing the program 1a. [4]

(d) Consider program 1a running on another computer that requires $160ns$, with $40ns$ spent executing FP instructions, $90ns$ executed L/S instructions, and $30ns$ spent executing branch instructions. What is the improvement factor using Amdahl's law if we only improve the performance of L/S instructions using a better ALU to get the program completion time improved by 2x? [4]

Instruction	CPI
add	2
addi	3
sub	2
lw	4
sw	4
sll	1
srl	1
slt	4
stli	8
beq	4
bne	4
j	8
jr	4
jal	4

Table 1: CPI for each instruction for program 1a

2. Consider the following C function. Compiler will assign a (base address) into \$s0, b(base address) into \$s1, x into \$s2 and i into \$s3.

```
    int main() {
    int x=1,i=32,a[10],b[10];
    do{
        if(a[2*i]==b[2*(i+1)]){
            x += a[2*i] - b[2*(i+1)];
            break;
        }
        else if(a[2*i] & b[2*(i+1)]){
            i = i+x;
            continue;
        }
        else{
            i = i - 3;
        }
        i = i+1;
    }while(i%4);
    return 0;
}
```

(a) **Convert** the code to the corresponding MIPS assembly instructions. [6]

(b) **Convert** the first 8 lines of MIPS assembly instructions to the corresponding machine code after entering the loop body. No need to convert it to binary. [4]

(c) Assume that the processor has 64 registers. The size of MIPS instruction is 32 bits and 6 bits are reserved for opcode. The structure for addi instruction is given in the table-2. **Find** out the maximum constant value for **addi** instruction in MIPS. Note that, MIPS supports negative constants. [2]

opcode	rs	rt	constant

Table 2: Structure of addi instruction

3. Using the division algorithm, **show** each step of the division of 15 by 4. [4]

Instruction	Opcode	Function Code
add	0	32
sub	0	34
lw	35	-
sw	43	-
and	0	36
or	0	37
nor	0	39
andi	12	-
ori	13	-
sll	0	0
srl	0	2
beq	4	-
bne	5	-
slt	0	42
j	2	-
jr	0	8
jal	3	-
addi	8	-

(a) MIPS Machine Codes

Name	Register Number
\$zero	0
\$at	1
\$v0-\$v1	2-3
\$a0-\$a3	4-7
\$t0-\$t7	8-15
\$s0-\$s7	16-23
\$t8-\$t9	24-25
\$k0-\$k1	26-27
\$gp	28
\$sp	29
\$fp	30
\$ra	31

(b) MIPS Registers