CS2100 Computer Organization
AY2023/24 Semester 2
Assignment 1

INSTRUCTIONS

1. There are FIVE (5) questions in this assignment, with a total of FORTY (40) marks. Please do all parts of every question. Marks are indicated against each part.

2. This assignment is due on **Monday, 19 February 2024, 1 pm.**

3. This is the question file for your reference. You have to submit the answers ONLY into the Canvas\Quizzes\Assignment1. Please follow all the necessary instructions specified in the canvas assignment before you submit the answers. Please follow the instructions specified in canvas assignment correctly for the autograder to work.

4. Late submission will NOT be acceptable and will receive ZERO marks. We are absolutely strict about this.

5. You should do these assignments **on your own**. You should neither discuss the assignment questions with others nor use any form of AI tools to complete this assignment. Please refer to these resources on plagiarism and using AI tools.

6. Please use the QnA Topic "Assignments" if you have any queries on the assignment.

7. If you encounter any technical issue regarding canvas assignment, please post it to the QnA Topic in the forum.

1. Tertiary (Base-3) Number System (Total: 8 marks)

(a) **Tertiary to Decimal Conversion** **(3 marks)**

You are given this unsigned tertiary (base-3) value in a tertiary number system with 8 digits for the integer part and 3 digits for the fraction part: $N_3=21012202.111_3$.
Convert N_3 to its decimal equivalent, correct to 4 decimal places.

(b) **Conversion to binary** **(3 marks)**

Convert the tertiary number $N_3=02210001.121_3$ to its binary equivalent, correct to 4 binary places.

(c) **Range of Representable Numbers** **(2 marks)**

Determine the range of numbers representable in a 4-digit tertiary number system in unsigned format.

2. Base-4 Number System (Total: 4 marks)

(a) **3's complement and 4's complement:** **(2 marks)**

Calculate the 3's complement and 4's complement for a given 8-digit base-4 number on signed numbers, N_4=32103203.

(b) **Range of Representable Numbers** **(2 marks)**

Determine the range of numbers representable in a 4-digit base-4 number system in (i) unsigned and (ii) signed (4's complement) format.

3. Excess-N Number Representation (Total: 6 marks)

(a) Convert the decimal number 25 to 8-bit excess-128 form. Explain the steps involved in the conversion process. **(2 marks)**

(b) Given an 8-bit binary number in excess-128 format, $10010110_{\text{Excess-128}}$, convert it back to its original decimal value. Describe the process used for conversion. **(2 marks)**

(c) Range of Representable Numbers: Determine the range of decimal numbers that can be represented in an 8-bit excess-128 system. Explain how the excess-N system affects the representable range of numbers compared to the standard unsigned binary representation. **(2 marks)**

4. IEEE 754 Format (Total: 6 marks)

Consider the single-precision IEEE 754 format for this question.

(a) **Decimal to IEEE 754 Conversion:** Convert the decimal number −118.625 to its IEEE 754 single-precision floating-point representation. Write your answer in hexadecimal. Outline the steps involved, including normalization, binary conversion, exponent adjustment, and final encoding. **(3 marks)**

(b) **IEEE 754 to Decimal Conversion:** Given the IEEE 754 single-precision floating-point number represented by the binary string 11000010111010000000000000000000, convert this binary string back into its decimal form. Describe the decoding process, including how to interpret the sign, exponent, and mantissa (fraction) fields. **(3 marks)**

5. MIPS (Total: 16 marks)

Study the MIPS program below. Mem[x] and Mem[y] are non-negative integers less than 10000. Assume that registers 4 – 15 are set to 0 prior to the execution of this program.

```
main:
    la    $8, x
    la    $9, y
    lw    $4, 0($8)
    lw    $5, 0($9)
    move  $11, $0
    beq   $5, $0, exit
loop:
    andi  $10, $5, 1
    neg   $10, $10
    and   $10, $10, $4
    add   $11, $10, $11
    srl   $5, $5, 1
    sll   $4, $4, 1
    bgei  $5, 1, loop
exit:
```

(a) There are four pseudo-instructions present in this MIPS program:

- `la` (load address, see lab 3)
- `move` (move value of one register to another)
- `bgei` (branch if greater than or equal to)
- `neg` (negate, in 2s complement).

Provide equivalent MIPS instruction(s) for the following pseudo-instructions:

- `move $dst, $src` (1 instruction)
- `bgei $src, imm, label` (2 consecutive instructions)
- `neg $dst, $src` (1 instruction)

Assume that there will be no overflow/underflow. Your equivalent MIPS instruction(s) should contain exactly the number of instructions specified. Use $t0 if you need a temporary register, and use $zero for the zero register. **(3 marks)**

(b) If Mem[x] = 2100 and Mem[y] = 24 at the start of the program, what is the value of register $11 at the end of the program? Write your answer in hexadecimal. **(2 marks)**

(c) In one sentence, explain the relationship between Mem[x], Mem[y], and the value of register $11 at the end of the program. **(1 mark)**

(d) Given Mem[x] = 2024 and Mem[y] = 2100 at the start of the program, determine the total number of times the `beq`/`bgei` instructions result in a branch during the execution of the program. **(2 marks)**

(e) What is the minimum and maximum number of instructions which could be executed in this program? Assume that any of the pseudo-instructions used counts as only one MIPS instruction. **(2 marks)**

(f) Encode the instructions on lines 4, 7, 10, 12, 13, and 14 in hexadecimal. Assume that each pseudo-instruction used above counts as only one MIPS instruction. **(6 marks)**