CS2100 Computer Organization
AY2025/26 Semester 2
Assignment 1
Total Marks: 40

Instructions for Students:

1. There are **4** questions in this assignment, with a total of 40 marks.

2. This assignment is due on **Monday, 16th February 2026, 1 pm**. Late submissions will incur penalties as spelt out on Canvas > Pages: "a 10% penalty for submissions up to 3 hours late (i.e. submitting between 1.01pm and 4:00pm for a 1pm deadline), a 20% penalty for submissions up to 6 hours late, a 30% penalty for submissions up to 9 hours late, and no marks will be given for assignments submitted more than 9 hours late."

3. Enter your answers into Canvas > Assignments > Assignment 1. You are allowed [3] attempts. Only your last submission (not your best) will be graded. You are not to request us to ignore your last submission (including a late submission), nor can you choose the version you want us to grade.

4. You could refer to the document assign1_25s2_qns.pdf at Canvas > Files > Assignments > Assignment 1 in which all the questions are presented in a single file for your convenience. You will look at the questions in this file and answer them in the Canvas.

5. Please read and follow the instructions in each question on how to format your answers. This is important as your answers will be auto-graded, so any answer that departs from the specified format will be graded as incorrect.

6. You should do these assignments on your own. Do not discuss the assignment questions with others.

7. Please post on QnA "Assignment " topic if you have any queries on this assignment.

Question 1: Number System Conversions and Precision (12 marks)

(a) Base Conversion from Binary (6 marks)
Convert the following binary number into the bases specified below:
1010110.101_2
(i) Convert the number to decimal.
(ii) Convert the number to hexadecimal.
(iii) Convert the number to base-5.
For fractional parts, your final answers should be rounded to the nearest 3 digits after the radix point.

(b) Base Conversion from Decimal (6 marks)
Convert the following decimal number into the bases specified below:
142.6875_{10}
(i) Convert the number to binary.
(ii) Convert the number to ternary (base-3).
(iii) Convert the number to hexadecimal.
For fractional parts, your final answers should be rounded to the nearest 3 digits after the radix point.

Question 2: Custom Number System Design and Comparison (8 marks)

A computer architect proposes a custom signed number system with the following properties:

- 6 digits and Base-5
- The most significant digit (MSD) is used as a sign digit: 0 is positive and 1 is negative
- The remaining digits represent the magnitude in base-5

(a) Range of the Number System (3 marks)
(i) The largest positive number representable in this system (in decimal) is [______]
(ii) The most negative number representable in this system (in decimal) is [______]
(iii) The total number of distinct numerical values representable in this system is [______]

(b) Representation of a Given Number (2 marks)
The representation of the decimal number −224 in this custom number system is [______].

(c) Comparison with Two's Complement (3 marks)
(i) The minimum value representable using 5-bit two's complement (in decimal) is [______]
(ii) Is the number -224_{10} is representable in 5-bit two's complement? [Yes/No]
(iii) One advantage of two's complement over the custom base-5 system above is: [______]
(Answer in one short phrase)

Question 3: Floating-Point Representation and Precision (4 marks)

This question focuses on how real numbers are represented (approximately) using the IEEE 754 single-precision floating-point format.

(a) Decimal to IEEE 754 (2 marks)
Consider the decimal number: +0.1875
(i) The normalized binary form of this number is $[\ \_\_\_\_\_\_\ ]_2 \times 2^\wedge[\ \_\_\ ]$
(ii) The exponent field (8 bits, after applying bias) is $[\ \_\_\_\_\_\_\ ]_2$

(iii) The final IEEE 754 single-precision representation (hexadecimal) is 0x[_______]

(b) IEEE 754 to Decimal (2 marks)
Given the IEEE 754 single-precision floating-point number: 0xC1A00000
(i) The sign of the number is [positive / negative]
(ii) The actual exponent value (after removing bias) is [_______]
(iii) The binary mantissa (including the implicit leading 1) is [_______]$_2$
(iv) The decimal value of the number is [_______]$_{10}$

Question 4: Analyse the given MIPS Code and answer the following questions (16 marks)

```
  # $s0 - address of memory that stores base address of the array
  # $s1 - address of memory that holds the array size

  lw $a0, 0($s0)
  lw $a1, 0($s1)
  addi $v0, $0, 0
  addi $t0, $0, 0
label3:
  beq $t0, $a1, label1
  lw $t1, 0($a0)
  andi $t2, $t1, 1
  bne $t2, $zero, label2
  add $v0, $v0, $t1
label2:
  addi $a0, $a0, 4
  addi $t0, $t0, 1
  j label3
label1:
  add $t4, $v0, $zero
exit:
```

(a) Explain the purpose of the given MIPS code when `$s0 = 0x2000` and `$s1 = 0x2004`. Assume that the address `0x2000` has the base address of an integer array, and the address `0x2004` stores the size of the array as 7. The elements of the integer array can take values from 0 to 99. What operation does the given MIPS code perform on the integer array?
(1 mark)

(b) What does register `$t4` contain at the end of execution?
(1 mark)

(c) Given the following memory contents, calculate the **total number of instructions executed** by the MIPS code. **(3 marks)**

Memory Address	**Values**
0x2000	0x3000
0x2004	7
…	…
0x3000	4
0x3004	7
0x3008	10
0x300C	3
0x3010	8
0x3014	6
0x3018	9

(d) Assuming that the array size is *n* (where *n* is a positive integer). What are the maximum and minimum possible number of times the branch instruction `beq $t0, $a1, label1` (line 9) is taken? A branch instruction is taken if its branch condition is true, otherwise it is not taken. **(2 marks)**

(e) Assuming that the array size is *n* (where *n* is a positive integer). What are the maximum and minimum possible number of times the branch instruction `bne $t2, $zero, label2` (line 12) is taken? A branch instruction is taken if its branch condition is true, otherwise it is not taken. **(2 marks)**

(f) Modify the given MIPS code so that it **terminates the loop immediately** when the value **99** is encountered in the array. The program should then exit and store the current accumulated result in register $t4. You should make a **minimal number of changes** to the original code and at the same time keep your code efficient. Clearly indicate the number of instruction(s) that must be added and/or removed with explicit reference to the line numbers in the original code. Example: *remove 2 instructions in line 1 and 2 and add the following 2 instructions between lines 3 and 4.* **(3 marks)**

(g) Instruction Encoding: Encode the instructions in line 9 and line 10 from the original code into 32-bit binary machine code. Clearly show the opcode and relevant fields used in the encoding. You need to represent the final machine code in hexadecimal. **(4 marks)**
(i) Instruction at line 9: `beq $t0, $a1, label1`
(ii) Instruction at line 10: `lw $t1, 0($a0)`