CSE3666

Homework 5 (140 points)

Due Date: By the end of Saturday, 3/23/2024.

Submit your work for the first 6 questions in a PDF in HuskyCT. The deadline for the first six questions is 3/23/20024.

Question 7 is mandatory. Read the document and take a test in HuskyCT. The firm deadline for Question 7 is Monday, 4/8/204.

1. Find the normalized (binary) representation of the following single-precision floating point numbers. Then, write the number in a) in decimal, and write the significand in b) in decimal.

   a. `0x45652000`
   b. `0x00070000`

2. Find the single-precision representation of the following values/numbers.

   a. -15.82×2^{-10}
   b. – 831.9

   We may need to do rounding, which is another complicated issue. In this question, we use the rule "Round to nearest, ties away from zero". For example, if keep only two bits after the binary points, `0.1110` is rounded to `1.00`. `-0.1010` is rounded to `-0.11`

   Write the normalized (binary) representation of each number first. Then find out the bits in each field.

Example:

`2.32`
`= 0b10.0101_0001_1110_1011_1000_0101…`
`= 0b1.00101_0001_1110_1011_1000_01` … $\times 2^1$

	S	Exponent	Fraction	Single-precision
`2.32`	`0`	`1000 0000`	`001 0100 0111 1010 1110 0001`	`0x40147AE1`
`-15.82*2^-10`				
`-831.9`				

3. Find the largest odd integer that can be represented in single-precision format. Write the number in decimal and 8 hexadecimal digits for its single-precision floating-point representation. Justify your answer.

4. Implement the following C code with RISC-V assembly code. Assume F-extension is available. Follow RISC-V calling conventions and use symbolic register names (e.g., ft0 and fa0) in your code. Skeleton code is provided. Only include this function in the submission.

```
float dot_product(float x[], float y[], int n)
{
    float sum = 0.0;

    for (int i = 0; i < n; i += 1)
        sum += x[i] * y[i];
    return sum;
}
```

5. Consider two processors P1 and P2 that have the same ISA but different implementations. The ISA has four classes of instructions: class A, B, C, and D. The clock rate of two processors and the number of clock cycles required for each class on the processors are listed in the following table.

Processor	Clock Rate	Class A	Class B	Class C	Class D
P1	2 GHz	1	2	3	3
P2	3 GHz	1	1	4	5

Suppose the breakdown of instructions executed in a program is as follows:

10% class A, 20% class B, 50% class C, and 20% class D.

Round answers to the nearest hundredth if necessary. For example, 1/2 to 0.5, 2/3 to 0.67.

a. What is the overall CPI of the program on P1?
b. What is the overall CPI of the program on P2?
c. How many times faster is the program on P2 than on P1? Note that the clock rate is different on P1 and P2.
d. Suppose a compiler can optimize the program, replacing all class D instructions with class A instructions. Each class D instruction requires two class A instructions. What is the average CPI of the program on P2 after the optimization?
e. What is the speedup the compiler in d) can achieve on processor P2?

6. Suppose you have two different methods to accelerate a program. Method 1 can accelerate 20% of the program 100 times. Method 2 can accelerate 20% of the program 10 times and 15% of the program 6 times. The part of code enhanced by each method does not overlap.

   Round answers to the nearest hundredth if necessary. For example, 1/2 to 0.5, 2/3 to 0.67.

   a. What is the speedup Method 1 can achieve on the entire application?
   b. What is the speedup Method 2 can achieve on the entire application?
   c. What is the speedup if both methods are applied?
   d. After both methods are applied, what is the best speedup (or the upper bound) one can achieve if they continue to optimize the code that is already enhanced by the two methods? Note that the reference is the code after both methods are applied.

7. (20 points) Mandatory. Some applications need floating-point numbers, but do not require the accuracy of single-precision floating-point numbers. IEEE 754-2008 specifies half-precision floating point numbers, a standard that represents floating-point numbers with 16 bits. It is similar to single-precision/double-precision numbers, but the field sizes are different. Read the following Wikipedia page and then complete the hw5-7 test in HuskyCT. This question is an exercise for learning by reading technical documents.

   https://en.wikipedia.org/wiki/Half-precision_floating-point_format