ITSC 3181 Introduction to Computer Architecture, Spring 2023

Homework #2, Due on 02/09

Covered topics: ISA and RISC-V, CPU performance analysis and CPI

Question	1	2	3	4	5	6	7	8	9	10	11	12	Total
Points	6	6	6	6	6	10	10	10	10	10	10	10	100

Count for total 5% of the final grade.

Submission:

1. **Only electronic submissions on canvas are accepted.**
2. **All your solutions should be included in a SINGLE PDF file. Include your full name in the PDF file.**
3. **Number your solutions in the same way and in the same order as the questions are numbered in this document.**
4. **You must show the necessary steps when you solve each question. 0 point will be given to the question if the answer only includes the final answer with no necessary steps.**
5. **Scanned copy of handwritten work are accepted and graded only if it is written clearly and readable.**

1. What is the average CPI of a 1.4 GHz machine that executes 12.5 million instructions in 12 seconds?

2. What is the CPI of a program execution that consists of the following instruction types (classes) and CPI:

Instruction class	CPI	Percentage
A	1.4	25%
B	2.4	70%
C	2	5%

3. Suppose one machine, A, executes a program with an average CPI of 1.9. Suppose another machine, B (with the same instruction set and an enhanced compiler), executes the same program with 20% less instructions and with a CPI of 1.1 at 800MHz. In order for the two machines to have the same performance, what does the clock rate of the first machine need to be?

4. Intel is considering two different possible enhancements to their newest i7 line of processors. They only have enough time to implement one of these enhancements before the scheduled release date of the processor.

   The instruction set of their original processor can be divided into three different types of instructions having the following CPIs:

Instruction type	CPI
A	1.4
B	2.4
C	2

   The benchmarks that are used to evaluate processor performance consist of the following instruction mixture:

Instruction type	Percentage
A	25%
B	70%

C	5%

Intel must choose one of the following options:

1. decrease the CPI of instruction type A by a factor of 5 (divide the CPI by 5), or
2. decrease the CPI of instruction type B by a factor of 2 (divide the CPI by 2).

These enhancements will not affect the number of instructions or the clock rate. Which enhancement is preferable? You must justify your answer numerically (with calculations).

5. Consider two different implementations, M1 and M2, of the same instruction set. There are three classes of instructions (A, B, and C) in the instruction set. M1 has a clock rate of 80 MHz and M2 has a clock rate of 100 MHz. The average number of cycles for each instruction class and their frequencies (for a typical program) are as follows. Calculate the average CPI for each machine, M1, and M2.

Instruction class	**Machine M1 CPI**	**Machine M2 CPI**	**Frequency**
A	1	2	60%
B	2	3	30%
C	4	4	10%

6. Translate the following high-level code to RISC-V assembly. **`vals`** is an array of **words (4 bytes)** whose base address is in x22**.** i and sum variables are in x5 and x6 registers, respectively:

```
int i=0;
int sum=0;
while (vals[i]!=0) {
      sum=sum+vals[i];
      i++;
}
```

7. Translate the following high-level code to RISC-V assembly:

```
for (i=0;i<n;i++) a[i]+=i;
```

Before your translation begins, assume variable **i** and **n** is stored in register **x5 and x6 respectively,** and that array **a** is an array of **words (4 bytes) and its base address is in x7**. Note: in the a[i] += i; statement, += operator makes a[i] a both left-val and righ-val. It needs to be load and stored.

8. Compile the following high-level code into RISC-V pseudo assembly language. Assume variable **a** and **val** are stored in memory as words. Make sure you load these values into registers before you use them and store back any changes to memory. You can use “lw x5, a” and “sw x5, a” as pseudoinstructions for load and store a word for variable a, considering a is a symbol for the address of the memory that store the value of a. This question is for you to practice using shift-logic instruction, logic (AND, OR, etc) instructions as well as branch instructions.

```
if (a < 20) {
      val = (val & 0xFFFF) * 16;
} else {
      val = (val >> 16) + 10;
}
```

9. Compile the following high-level code into RISC-V pseudo assembly language. Assume **a**, **b**, and **c** are stored in memory as words. Use x5, x6 and x7 for a, b, and c. When the computation completes, the variable whose value changes should be stored back to memory. You can use “lw x5, a” and “sw x5, a” as pseudoinstructions for load and store a word for variable a.

```
if (a < b) {
     c=1; //true1
} else if (a == b) {
     c=2;//true2
} else {
     c=3;
}
```

10. Representing the following decimal number using 8-bit two’s complement binary representation:
    a. 31
    b. -1
    c. -27
    d. 69

11. Compute the followings in binary. Use 8-bit two’s complement representation for binary numbers unless the numbers are already in binary.

    1. 15_{10}- 16_{10}

    2. 0011_{2} + 1101_{2}

    3. -15_{10}– 16_{10}

    4. 0011_{2} - 1101_{2}

12. Let A = 1001_2 and B = 0101_2. What is
    a. A & B (bitwise AND)

    b. ~A (bitwise NOT/Invert)

    c. -B (negative B, use 2’s complement)

    d. A >> 1 (right shift)

    e. A | B (bitwise OR)

    f. A ^ B (bitwise XOR)

    g. A << 2 (left shift)