CSC 462 Final Exam
Due: Monday, May 6 by 2 pm

This exam counts as 2 homeworks toward your final grade. Answer questions #1 and #9 and any four other questions (6 total).

1. What specific topics that you learned in this course will help you be a better software developer? For each topic, provide an example or explanation for how the topic will help you. If you feel that none of the topics will help you, explain why you feel this way.

2. The ideal CPI of a pipeline is 1.0. We have two processors, one is pipelined and the other is not. The pipelined processor uses an optimizing compiler to schedule code to remove as many stalls and penalties as possible. The only sources of stalls/penalties are loads followed by using the loaded value and branch penalties. 50% of all loads cause a RAW hazard of 1 cycle but 60% of these are removed by scheduling. All taken branches have a 2 cycle branch penalty but 20% of these are completely removed by scheduling and a further 30% are reduced to 1 cycle penalty by scheduling. For the non-pipelined machine, the CPI is given below. The non-pipelined machine's clock is 25% faster than the pipelined machine and the code running on the non-pipelined machine is 30% shorter (IC is 30% less than that of the pipelined machine). How much faster is the pipelined machine on a benchmark of the given instruction mix.
   Loads 38% (CPI = 4)
   Stores 10% (CPI = 3)
   ALU 40% (CPI = 3)
   Branches taken 9% (CPI = 3)
   Branches not taken 3% (CPI = 2)

3. Provide the pipeline timing diagram for the following RISC-V code on the 5-stage RISC-V pipeline (with forwarding, and branches determined in the ID stage). Show two full iterations of the loop where the first bne is taken the first iteration and not taken the second. The second bne is taken in both iterations.

```
Loop:   lw      x2, 0(x1)
        lw      x4, 0(x3)
        addw    x5, x2, x4
        lw      x6, 0(x5)
        bne     x6, x7, Foo
        lw      x8, 0(x6)
        addiw   x8, x8, 1
        sw      x8, 0(x6)
        j       Out
Foo:    subiw   x5, x2, x4
        sw      x8, 0(x6)
Out:    addiw   x1, x1, 4
        addiw   x3, x3, 4
        bne     x1, x10, Loop
```

4. The effective access time formula, `hit time + miss rate * miss penalty`, shows us how memory references can impact the performance of our processor. First, explain how an EAT > 1 cycle impacts the processor. Second, select 3 cache optimizations as covered in class that have the greatest positive impact on EAT. Explain your choices.

5. Which one innovation from the list below has the greatest impact on processor performance? Justify your answer by both explaining the positive impact of your choice and comparing it to some of the other choices that you feel have had a lesser impact. You do not need to compare your choice to all of those listed below, but you should compare it to those you feel are the closest competitors.

Pipelines	Superscalar pipelines	Reservation stations/reg. renaming
Caches	Pipelined functional units	Compiler optimizations
GPUs	Branch speculation	Multi-core processors
VLIW	Dynamic scheduling	Pipelining microcode

6. Given the following C code, where a is an array of doubles and c is a double
    a. rewrite it in RISC-V and show where the stalls/penalties arise
    b. unroll and schedule the loop to remove as many stalls as possible.
    c. unroll and schedule the loop on a 5-issue VLIW superscalar as covered in section 3.7 of the textbook (which can accommodate up to 2 loads/stores, 2 FP operations and 1 int operation per cycle)
    d. provide the CPI for all three of your sets of code (original, unrolled and scheduled, VLIW).

```
for(i=0;i<1000;i++) {
      a[i] = a[i] + a[i] * c;
```

   Assume x1 stores the starting address of array a, x2 stores 8000 (used to compare against x1 to exit the loop), and f1 stores c. For stalls, assume multiplication takes 4 cycles to compute, addition takes 3 cycles, branches are determined in the ID stage, forwarding is available and an FP instruction can enter the MEM stage at the same time as a store.

7. Explain the following concepts from chapters 4 in your own words.
    a. How does a vector (SIMD) processor differ from a GPU?
    b. Convoy and chime
    c. Maximum vector length and strip mining
    d. Predicate registers
    e. Lanes
    f. The gather-scatter algorithm
    g. CUDA

8. What is cache coherence and why is it a problem? Would it be a problem on a computer with one single-core processor? In order to resolve cache coherence, two approaches are snoopy caches and directories. Explain how both work. In your explanation, make sure you explain what the various states are and mean (e.g., modified, owner, exclusive, shared, invalid, uncached).

9. Provide a critique of the course:
    a. What material did you find most interesting (if any)?
    b. What material did you find least interesting?
    c. What material should be covered in more detail and more slowly (if any)?
    d. What material should be dropped from the course (if any)?
    e. Did you like only having homework assignments? Would you have preferred more shorter assignments or fewer longer assignments, or as it was? Would you have preferred open-book, open-note exams in place of homeworks?
    f. Knowing what you now know of the course, would you have taken it or would you have preferred a different CSC elective?
    g. Anything else you want to add? Please add it.