

Assignment 5

Due: 1:25pm, Wed Oct 26th, 2022

Note: Make reasonable assumptions where necessary and clearly state them. Feel free to discuss problems with classmates, but the only written material that you may consult while writing your solutions are the textbook and lecture slides/videos. Solutions should be uploaded as a single pdf file on Canvas. **Show your solution steps** so you receive partial credit for incorrect answers and we know you have understood the material. Don't just show us the final answer.

Every homework has an automatic penalty-free **1.5 day extension** to accommodate any covid/family-related disruptions. In other words, try to finish your homework by Wednesday 1:25pm to keep up with the lecture content, but if necessary, you may take until **Thursday 11:59pm**.

1. Out-of-order processing (100 points)

Consider an out-of-order processor similar to the one described in class. The architecture has 32 logical registers (also known as architected registers or program-defined registers and indicated as LR*) and 38 physical registers (indicated as PR*). On power up, the following program starts executing. To simplify the problem, some of the initialization code is not shown and you can ignore that code. The loop in the program is executed for at least two iterations.

```
line1: L.D LR1 0(LR2)
DADD LR1, LR1, LR3
DADD LR1, LR1, LR1
ST.D LR1, 0(LR2)
DADD LR2, LR2, 8
BNE LR2, LR4, line1
```

The processor has a width of 3, i.e., every pipeline stage can move up to 3 instructions through in every cycle. Show the renamed code for the first 12 instructions of this program. In what cycle will the 12th instruction get committed? Show your work, similar to the table in the class slides.

Assumptions:

Assume that branch prediction is perfect for a simple program like this. With the help of a trace cache, even fetch is perfect. Assume that caches are perfect as well. Assume that the dependent of a DADD instruction can leave the issue queue in the cycle right after the DADD. Assume that the dependent of an L.D cannot leave in the next cycle, but the cycle after that. Assume a ROB, an issue queue, and an LSQ with 20 entries each. When the thread starts executing, its logical register LR1 is mapped to physical register PR1, LR2 is mapped to PR2, and so on. An instruction goes through 5 pipeline stages before it gets placed in the issue queue and an additional 5 pipeline stages (6 for a LD/ST) after it leaves the issue queue (in other words, an instruction will take a minimum of 11 cycles to go through the pipeline). When determining if a L.D can issue, you need not check to see if previous store addresses have been resolved (just to make the problem simpler). As a further simplification, assume that stores leave the issue queue when their register dependences have been fulfilled (recall that a real processor will issue a store only when the store is the oldest instruction in the ROB).