Assignment 8

Due: 9:00am, Tue Apr 3rd, 2018

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.

1. Consider an in-order 5-stage pipeline similar to the one discussed in class, e.g., see slides 4-6 of lecture 18. First assume that the pipeline does not support bypassing (forwarding). What are the stall cycles introduced between the following pairs of back-to-back instructions? Then, solve the same problem while assuming support for bypassing. Clearly show your work, i.e., show how each instruction goes through the 5 stages, indicate the point of production and point of consumption, show how the consuming instruction is held back in the D/R stage when there are stalls. Recall that a register read is performed in the second half of the D/R stage and a register write is performed in the first half of the RW stage. **(30 points)**

    1. lw $1, 8($2)
       add $4, $1, $3
    2. lw $1, 8($2)
       sw $3, 8($1)

2. Consider an in-order pipeline that has the following stages. Unlike our 5-stage pipeline, a register read takes an entire cycle and a register write takes an entire cycle (not a half cycle).

Fetch	Decode	Regread	IntALU	Regwrite		
			IntALU	Datamem	Datamem	Regwrite

    After instruction fetch, the instruction goes through a separate Decode stage where dependences are analyzed, then a separate Regread stage where input operands are read from the register file. After this, an instruction takes one of two possible paths. Int-adds go through the stages labeled "IntALU" and "Regwrite". Loads/stores go through the stages labeled "IntALU", "Datamem", "Datamem", and "Regwrite", i.e., it takes two cycles to retrieve data from the data memory unit. How many stall cycles are introduced between the following pairs of successive instructions (i) for a processor with no register bypassing and (ii) for a processor with full bypassing? **(40 points)**

    1. add $1, $2, $3
       add $4, $1, $5
    2. lw $1, 8($2)
       lw $3, 8($1)

3. Consider a program that executes a large number of instructions. Assume that the program does not suffer from stalls from data hazards or structural hazards. Assume that 20% of all instructions are branch instructions, and 75% of these branch instructions are Taken. What is the average CPI for this program when it executes on each of the processors listed below? All of these processors implement an 10-stage pipeline and resolve a branch outcome at the end of the 4th stage. The 1st stage fetches an instruction, the 2nd stage does decode, the 3rd stage does register read, and the 4th stage does the computations for the branch. **(30 points)**
    1. The processor pauses instruction fetch as soon as it fetches a branch. Instruction fetch is resumed after the branch outcome has been resolved.
    2. The processor always fetches instructions sequentially. If a branch is resolved as Taken, the incorrectly fetched instructions after the branch are squashed.
    3. The processor implements three branch delay slots. The compiler fills the branch delay slots with three instructions that come before the branch in the original code (option A in the videos).
    4. The processor does not implement branch delay slots. Instead, it implements a hardware branch predictor that makes correct predictions for 90% of all branches. When an incorrect prediction is discovered, the incorrectly fetched instructions after the branch are squashed.