EECS 470 Fall '25
Homework 3

Due Fri. Oct. 3rd at 11:59pm.

Name: _____ unique name: _____

Register Renaming.

1. What kinds of dependencies does register renaming eliminate? [2].

2. Create a very brief code sequence (at most 3 instructions) that finishes faster in a Tomasulo machine than in a Scoreboard machine with the same functional units. Briefly (1-3 sentences) state assumptions about each machine required to understand your example, and explain why your instruction sequence stalls in the Scoreboard machine, and how the Tomasulo machine avoids that stall [3].

Precise Exceptions.

3. In no more than 2 sentences, explain what is meant by *precise exceptions*. [1]

4. In a P6-like machine, under what conditions will the complete stage stall? [3].

5. What happens during the retire (a.k.a commit, complete, graduate) stage in a P6-like machine with a reorder buffer? What about a history buffer? Which machine performs more work on instruction retirement? When recovering from an exception? (Hint: See Smith & Pleszkun paper) [4].

MIPS R10K-style microarchitecture (physical register file).

Consider a MIPS R10K-style machine (ROB + physical register file) for an ISA with 32 general-purpose registers.

6. Suppose the machine has 32 ROB entries. How big must the physical register file be? [2]

7. Many processors support multiple execution threads in hardware (e.g., Intel's Hyperthreading, Sun's Niagara). These machines can switch execution from thread to thread each cycle, or in some cases, execute instructions from more than one thread in the same cycle. To support this, the hardware must keep track of register values for each thread. What would your answer for #6 be for a processor that supports two hardware threads? [2]

8. When is a physical register returned to the free list? [3]

9. Post a screenshot of your work on the in-class P6 Problem (Lecture 9). [10]