

## EECS470 Homework 3 Answers

- 1) False (WAR and WAW) dependencies.
- 2) Many examples would work. Most commonly, making sure the code sequence has a WAW or WAR hazard that stalls out the Tomasulo machine.
- 3) A machine that implements *precise exceptions* ensures that trap instructions (and typically also external interrupts) are handled such that all effects of instructions prior to the trap are complete upon entering the trap handler, but none of the effects of instructions after the trap have occurred. In other words, the trap (or interrupt) appears to occur precisely between two instructions in the instruction sequence. Precise exceptions allow the system to resume execution where it left off after the trap is handled.
- 4) Complete could stall if multiple instructions finish execution in the same cycle, and there are not enough CDBs. Note that some designs try to prevent this by allocating CDBs at issue, in which case complete never stalls.
- 5) With an ROB, the result of the instruction is copied from the ROB entry to its destination register in the register file, and the ROB head pointer is advanced. With a history buffer, just the ROB head pointer is advanced. The ROB requires more work on retirement, while the history buffer requires more work when recovering from an exception.
- 6)  $32 \text{ GPRs} + 32 \text{ ROB entries} = 64 \text{ physical registers}$
- 7)  $32 \text{ GPRs for thread 1} + 32 \text{ GPRs for thread 2} + 32 \text{ ROB entries} = 96 \text{ physical registers}$
- 8) The previous physical register associated with an ROB entry (i.e., the physical register containing the value that this instruction overwrites) is returned to the free list when that ROB entry retires.