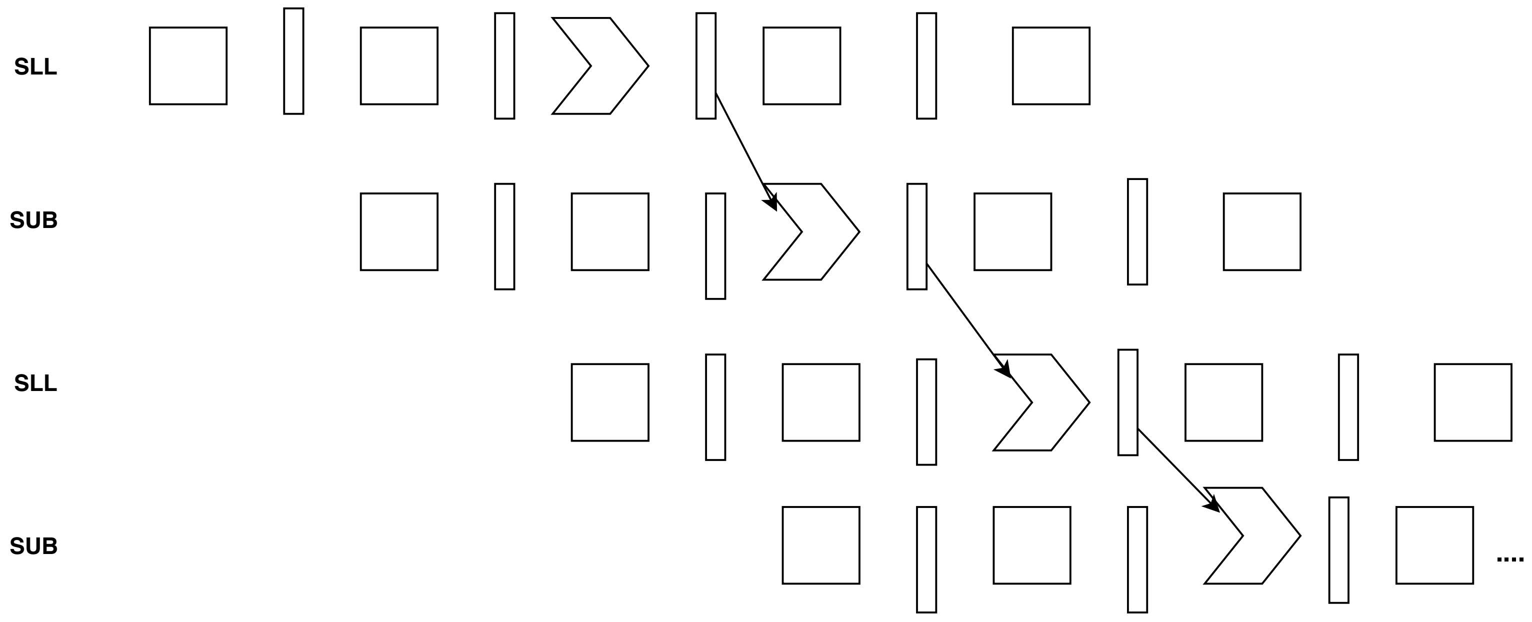
Alexander Miller
Homework 6

1D



2F

CASE 1: NO FORWARDING

In Cycle 4, the SUB operation is in ID phase, where it detects a hazard (needs $s2 from the ALU result of first instruction and $s5 from ALU result of the second instruction. It can only get the values it needs once the lattermost instruction enters WB phase, so it stalls Cycles 4-5. In Cycle 6, SUB in ID can read the value since instruction 2 has already written in its WB; instructions 1 and 2 have finished; the ADDI instruction can now do IF. In Cycle 7, SUB proceeds to EX phase; ADDI enters ID, recognizes a hazard since it needs result from SUB instruction, and so stalls.

CASE 2: FORWARDING

In Cycle 4, instruction 3 notices a potential hazard in ID but doesn't stall since the first 2 instructions (where its dependencies lie) will have finished EX before instruction 3 enters EX. In Cycle 5, Instructions 1 and 2 forward data from MEM/WB and EX/MEM pipeline registers (respectively) to the SUB ALU inputs. Instruction 4 detects a potential hazard in ID, but doesn't stall since instruction 3 (where it's dependency lies) will have finished EX before instruction 4 needs to perform EX. In Cycle 6, instruction 3 forwards from EX/MEM pipeline register to ALU input for instruction 4. In Cycle 7, instruction enters WB and instruction 4 enters MEM.
~ NO STALLING OCCURS, ALL HAZARDS RESOLVED WITH FORWARDING