Five-Stage Pipelined Processing Simulator

Implement (in any programming language) a simulator for five-stage pipelined processing (IF, ID, EX, MEM, WB).

Support the following instructions: ADD, SUB, MUL, DIV, LOAD, and STORE (with their standard meanings).

ADD, SUB, LOAD, and STORE require one cycle to execute (EX stage). MUL and DIV require two cycles to execute (EX stage).

LOAD and STORE require three cycles to access memory. Assume that the IF, ID, and WB stages always take one cycle to complete.

Each instruction has a maximum of three register operands (a destination register and one or two source registers).

As input, a sequence of instructions is provided. The output should be presented in a table that shows, for each cycle (column), the stages (or stalls) in which the input instructions (rows) are located.

Allow for specifying whether data forwarding between pipeline stages will be performed or not.

Allow for printing cases where data forwarding occurs, with all relevant information (instruction serial numbers and their execution stages during forwarding).

Implement detection of hazards (RAW dependencies) between instructions in the input sequence.