module topLevel;

reg clk, reset;

wire [7:0] instruction, PC;

Imem I_mem (clk, PC, instruction);
cpu cpuI (clk, reset, instruction, PC);

initial begin
  clk = 1'b1;
  reset = 1'b1;
  #110 reset = 1'b0;
  #2000 $finish(2);
end

always begin
  #50 clk = ~clk;
end

// This statement is used for testing purposes.
// It outputs various pipeline register contents
// at the beginning and at the end of each cycle.
// Also, it indicates how to refer to variables defined
// at lower-level models.
always @(posedge clk) begin
  #10 $strobe (%t, PC = %h instr = %h IR = %h opA = %h opB = %h wr = %h", $time, PC, instruction, cpuI.instr, cpuI.rd0, cpuI.opB, cpuI.wr);
  #80 $strobe (%t, PC = %h instr = %h IR = %h opA = %h opB = %h wr = %h", $time, PC, instruction, cpuI.instr, cpuI.opA, cpuI.opB, cpuI.wr);
end

endmodule