```
// Eugene Ngo
// 4/20/2023
// CSE 469
// Lab 2, Task 1 and 2

/* testbench is a simulation module which simply instantiates the processor system and runs 50 cycles
** of instructions before terminating. At termination, specific register file values are checked to
** verify the processors’ ability to execute the implemented instructions.
*/
module testbench();

    // system signals
    logic clk, rst;

    // generate clock with 100ps clk period
    initial begin
        clk = '1;
        forever #50 clk = ~clk;
    end

    // processor instantion. Within is the processor as well as imem and dmem
    top cpu (.clk(clk), .rst(rst));

    initial begin
        // start with a basic reset
        rst = 1; @(posedge clk);
        rst <= 0; @(posedge clk);

        // repeat for 50 cycles. Not all 50 are necessary, however a loop at the end of the program will keep anything weird from happening
        repeat(50) @(posedge clk);

        // basic checking to ensure the right final answer is achieved. These DO NOT prove your system works. A more careful look at your
        // simulation and code will be made.

        // task 1:
        assert(cpu.processor.u_reg_file.memory[8] == 32'd11) $display("Task 1 Passed");
        else                                                 $display("Task 1 Failed");

        // task 2:
        //assert(cpu.processor.u_reg_file.memory[8] == 32'd1)  $display("Task 2 Passed");
        //else                                                 $display("Task 2 Failed");

        $stop;
    end

endmodule
```