```
`timescale 1ns / 1ps

module digital_clock_tb;

    // Inputs
    reg clk;
    reg rst;
    reg btn_h;
    reg btn_m;

    // Outputs
    wire [6:0] seg;
    wire [3:0] an;

    // Instantiate the Unit Under Test (UUT)
    digital_clock_top uut (
        .clk(clk),
        .rst(rst),
        .btn_h(btn_h),
        .btn_m(btn_m),
        .seg(seg),
        .an(an)
    );

    // Override parameters for simulation speedup
    // Temporarily reduce the counter max values so we can see the
time change quickly in simulation.
    defparam uut.dividers.COUNTER_1HZ_MAX = 10;
    defparam uut.dividers.COUNTER_REFRESH_MAX = 4;

    // Clock generation (10ns period -> 100MHz)
    initial begin
        clk = 0;
        forever #5 clk = ~clk;
    end

    initial begin
        // Initialize Inputs
        rst = 1;
        btn_h = 0;
        btn_m = 0;

        $display("--- Start Simulation ---");

        // Reset Phase
        #100;
        rst = 0; // Release reset
        // Phase 1: Observe normal counting (wait for 20 simulated
seconds)
        $display("Phase 1: Observing counters increment normally (fast
speed)...");
        #2000;

        // Phase 2: Test Minute Increment Button
        $display("Phase 2: Testing Minute Button press...");
        btn_m = 1; // Press button
        #200; // Hold briefly (simulating debounce period)
        btn_m = 0; // Release
        #500; // Observe change

        // Phase 3: Test Hour Increment Button
        $display("Phase 3: Testing Hour Button press...");
        btn_h = 1; // Press button
        #200;
        btn_h = 0;
        #500;

        // Phase 4: Test Reset
        $display("Phase 4: Testing Reset...");
        rst = 1;
        #100;
        rst = 0;
        #500;

        // End simulation
        $display("--- Simulation Complete ---");
        $stop;
    end

endmodule
```