

FPGA assignment: reference and simulation test

Copy each code listing into a plain-text file with the filename shown. In LabsLand, use **leds_mirror.v** for the module source.

leds_mirror.v

```
// Reference solution: E = SW[2], A = SW[1], B = SW[0].
// DE1-SoC Verilog Teach profile: keep file/module name leds_mirror.
module leds_mirror(input [9:0] SW, output [9:0] LEDR);
    assign LEDR[0] = SW[2] & (SW[1] | SW[0]);
    assign LEDR[3:1] = SW[2:0]; // E, A, B in switch order
    assign LEDR[9:4] = 6'b000000;
endmodule
```

check_assignment_tb.v

```
`timescale 1ns/1ps
module check_assignment_tb;
    reg [9:0] SW;
    wire [9:0] LEDR;
    reg expected;
    integer i;
    leds_mirror dut(.SW(SW), .LEDR(LEDR));
    initial begin
        for (i = 0; i < 1024; i = i + 1) begin
            SW = i; #1;
            // Independent requirement oracle: allow only EAB = 101, 110 or 111.
            case (i % 8)
                5,6,7: expected = 1'b1;
                default: expected = 1'b0;
            endcase
            if (LEDR[0] !== expected || LEDR[3] !== SW[2] ||
                LEDR[2] !== SW[1] || LEDR[1] !== SW[0] ||
                LEDR[9:4] !== 6'b000000) begin
                $display("FAIL SW=%b LEDR=%b expected Y=%b",SW,LEDR,expected);
                $fatal(1);
            end
        end
        $display("PASS: all 1024 switch settings; access rule, input mirrors and unused LEDs.");
        $finish;
    end
endmodule
```