// Lighting Automation - Combined with FSM

module lighting\_automation(

input wire clk, // System clock

input wire reset, // System reset

input wire motion\_sensor, // Motion sensor input

input wire light\_sensor, // Light sensor input (e.g., for daylight detection)

input wire manual\_override, // Manual override switch

output reg light // Output control for the light

);

// State definitions

typedef enum logic [1:0] {

OFF = 2'b00,

ON = 2'b01,

AUTO = 2'b10

} state\_t;

state\_t current\_state, next\_state;

// Parameters

parameter DARK\_THRESHOLD = 1'b0; // Light sensor threshold, e.g., 0 for dark

// Combinational block for next state logic

always @(\*) begin

case (current\_state)

OFF: begin

if (manual\_override)

next\_state = ON;

else if (!light\_sensor && motion\_sensor)

next\_state = AUTO;

else

next\_state = OFF;

end

ON: begin

if (!manual\_override)

next\_state = OFF;

else

next\_state = ON;

end

AUTO: begin

if (manual\_override)

next\_state = ON;

else if (light\_sensor || !motion\_sensor)

next\_state = OFF;

else

next\_state = AUTO;

end

default: next\_state = OFF;

endcase

end

// Sequential block for state transitions

always @(posedge clk or posedge reset) begin

if (reset) begin

current\_state <= OFF;

end else begin

current\_state <= next\_state;

end

end

// Output logic

always @(\*) begin

case (current\_state)

OFF: light = 1'b0;

ON: light = 1'b1;

AUTO: light = (!light\_sensor && motion\_sensor);

default: light = 1'b0;

endcase

end

endmodule

// Testbench for Lighting Automation with FSM

module tb\_lighting\_automation\_fsm();

reg clk;

reg reset;

reg motion\_sensor;

reg light\_sensor;

reg manual\_override;

wire light;

// Instantiate the FSM-based module

lighting\_automation uut (

.clk(clk),

.reset(reset),

.motion\_sensor(motion\_sensor),

.light\_sensor(light\_sensor),

.manual\_override(manual\_override),

.light(light)

);

initial begin

clk = 0;

forever #5 clk = ~clk; // 10 ns clock period

end

initial begin

$monitor($time, " Motion: %b, Light: %b, Manual: %b, Reset: %b -> Light Output: %b",

motion\_sensor, light\_sensor, manual\_override, reset, light);

// Initialize inputs

reset = 1; motion\_sensor = 0; light\_sensor = 1; manual\_override = 0; #10;

reset = 0; #10;

// Test cases

motion\_sensor = 1; light\_sensor = 0; manual\_override = 0; #20; // AUTO mode activates

motion\_sensor = 0; light\_sensor = 1; manual\_override = 0; #20; // AUTO mode turns off

manual\_override = 1; #20; // MANUAL ON mode

manual\_override = 0; motion\_sensor = 1; light\_sensor = 0; #20; // Back to AUTO

reset = 1; #10; // Reset

$finish;

end

endmodule