```
module seq_det(input signal_in,
               input      clk,
               input      reset,
               output reg signal_out);
    reg out_int;
    reg [1:0] curr_state, next_state;
    parameter s0 = 2'b00, s1 = 2'b01, s2 = 2'b10;

    always@(posedge clk or posedge reset) begin
        if(reset) begin
            curr_state <= s0;
            signal_out <= 0;
        end else begin
            curr_state <= next_state;
            signal_out <= out_int;
        end
    end

    always@(*) begin
        case(curr_state)
            s0: begin
                next_state = (~signal_in)?s1:s0;
                out_int = 1'b0;
                end
            s1: begin
                next_state = (~signal_in)?s2:s0;
                out_int = 1'b0;
                end
            s2: begin
                next_state = (~signal_in)?s0:s0;
                out_int <= 1'b1;
                end
            default: begin
                next_state = s0;
                out_int = 1'b0;
                end
       endcase
    end
endmodule
```