Review Problem 26

- An ambulance company wants a flashing yellow light that, when a button is held, will instead hold a solid red. Design this machine.

Set clock at 1 Hz

I write down the true outputs
An input/output system seems to work. What am I outputting now? I know this clock edge will probably not work in the future.

Read this is an input to this.
Vending Machine Example (cont.)

\[ \text{Open} = D + \overline{P_S}N \]

\[ N_S = \overline{P_S}N + P_S\overline{N}D \]
Vending Machine Example (cont.)

- Implementation:
FSMs in Verilog - Declarations

module simple (clk, reset, w, out);
  input logic clk, reset, w;
  output logic out;

enum { A, B, C} ps, ns; // Present state, next state
FSMs in Verilog – Combinational Logic

// Next State Logic
always_comb begin
    case (ps)
        A: if (w) ns = B;
            else ns = A;
        B: if (w) ns = C;
            else ns = A;
        C: if (w) ns = C;
            else ns = A;
    endcase
end

// Output Logic – could also be “always”,
// or part of next-state logic.
assign out = (ps == C);
FSMs in Verilog – DFFs

// Sequential Logic (DFFs)
always_ff @(posedge clk) begin
    if (reset)
        ps <= A;
    else
        ps <= ns;
end

endmodule
Circuit Diagram of FSM