FSMs Verilog Behavioral Modeling CheatSheet

```
module fsm_template(reset_n, x_in, clk, mealy, moore);
    input  reset_n, x_in, clk;
    output reg mealy, moore;
```

variable declations for state manipulations

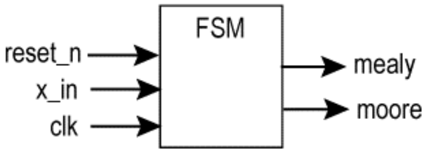
```
    //- next state & present state variables
    reg [1:0] NS, PS;
```

state bit assignments

```
    //- bit-level state representations
    parameter [1:0] st_A=2'b00, st_B=2'b01, st_C=2'b11;
```

state register definition

```
    //- model the state registers
    always @ (negedge reset_n, posedge clk)
       if (reset_n == 0)
          PS <= st_A;
       else
          PS <= NS;
```



```
    //- model the next-state and output decoders
    always @ (x_in,PS)
    begin
```

assign all outputs to avoid latches

```
       mealy = 0; moore = 0;
       case(PS)
          st_A:
          begin
```

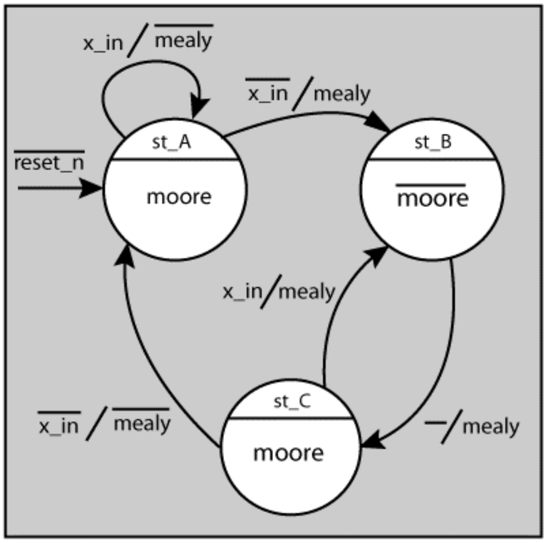
Moore outputs are function of state only

```
             moore = 1;
             if (x_in == 1)
             begin
                mealy = 0;
                NS = st_A;
             end
             else
             begin
                mealy = 1;
                NS = st_B;
             end
          end
```

Mealy outputs are function of state and external input

```
          st_B:
             begin
                moore = 0;
                mealy = 1;
                NS = st_C;
             end

          st_C:
             begin
                 moore = 1;
                 if (x_in == 1)
                 begin
                    mealy = 1;
                    NS = st_B;
                 end
                 else
                 begin
                    mealy = 0;
                    NS = st_A;
                 end
             end
```


x_in / $\overline{\text{mealy}}$
$\overline{\text{x\_in}}$ / mealy
$\overline{\text{reset\_n}}$
st_A
moore
st_B
$\overline{\text{moore}}$
x_in/mealy
$\overline{\text{x\_in}}$ / $\overline{\text{mealy}}$
st_C
moore
—/mealy

illegal state recovery

```
          default: NS = st_A;

          endcase
      end
endmodule
```