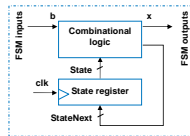
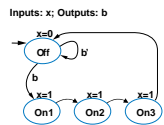


Lecture 4 Implementing FSMs in Verilog

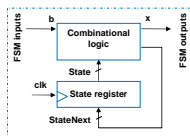
Finite-State Machines (FSMs)—Sequential Behavior

- Finite-state machine (FSM) is a common model of sequential behavior
 - Example: Laser Timer
 - If B=1, hold X=1 for 3 clock cycles
 - Remember, transitions implicitly ANDed with rising clock edge
- Implementation model has two parts:
 - State register
 - Combinational logic
- HDL model will reflect those two parts



Finite-State Machines (FSMs)—Sequential Behavior

- Modules has two procedures
 - One procedure for combinational logic
 - One procedure for state register
- But it's still a behavioral description



```

`timescale 1 ns/1 ns
module LaserTimer(B, X, Clk, Rst);
    input B;
    output reg X;
    input Clk, Rst;

    parameter S_Off = 0, S_On1 = 1,
              S_On2 = 2, S_On3 = 3;

    reg [1:0] State, StateNext;

    // CombLogic
    always @(State, B) begin
        ...
    end

    // StateReg
    always @(posedge Clk) begin
        ...
    end
endmodule
    
```

Describing Safe FSMs in VHDL

- Explicitly describing a safe FSM
 - Include case(s) to describe illegal states
 - Can use default: case item
 - Executes if State value equals anything other than S_Off, S_On1, or S_On2

```
...
reg [1:0] State, stateNext;
always @(State, B) begin
  case (State)
    S_Off: begin
      X <= 0;
      if (B == 0)
        stateNext <= S_Off;
      else
        stateNext <= S_On1;
      end
    S_On1: begin
      X <= 1;
      stateNext <= S_On2;
      end
    S_On2: begin
      X <= 1;
      stateNext <= S_Off;
      end
    default: begin
      X <= 0;
      stateNext <= S_Off;
      end
  endcase
end
...
```
