

Finite State Machines modeling using Verilog HDL

Mohammed A. Elsha3rawy

Contents

1. Finite State Machines "recap"

2. Mealy vs. Moore Machines

2.1. Sequence detector "11" FSM diagrams

2.2. Sequence detector "11" circuit

2.2. Sequence detector "11" circuit

3. Complex FSM "Seq. Detector"

4. FSM Optimizations

4.1. State encoding

4.2. clk gating

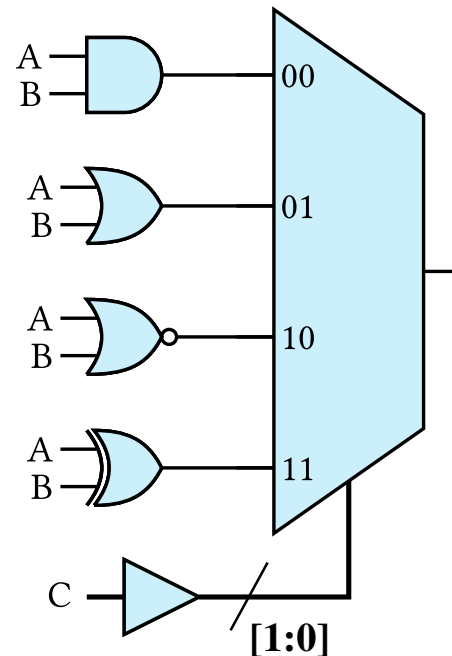
5. Multiplier

5. Tasks

Combinational Circuits

- Consists of (logic gates, Mux, DeMux, encoder, decoder, ...)
- Output depends only on the input
- Don't have any memory cells or latches

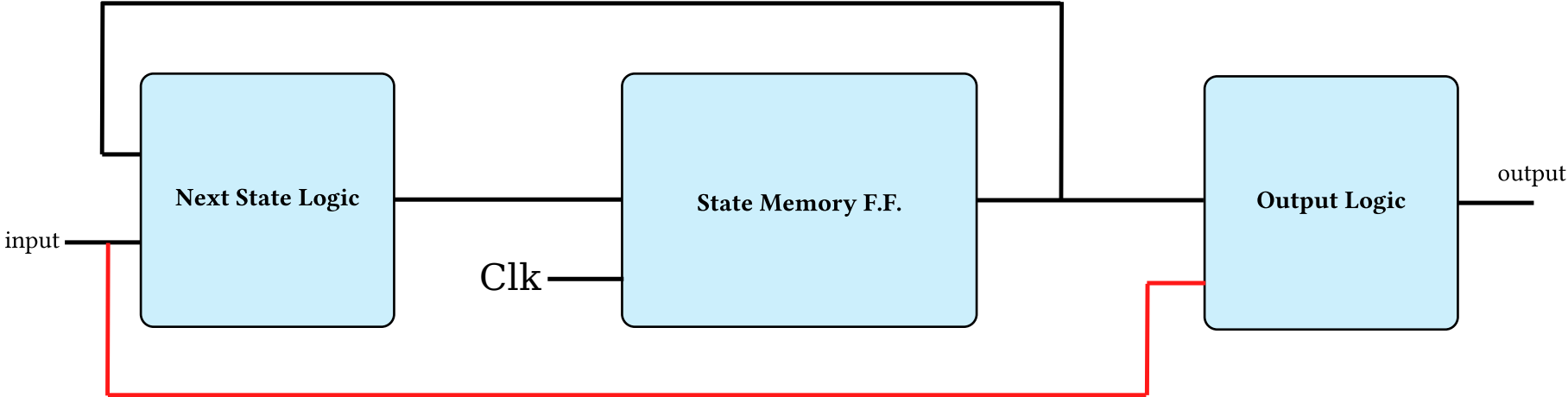
" Don't ever forget The third info. "



Finite State Machines

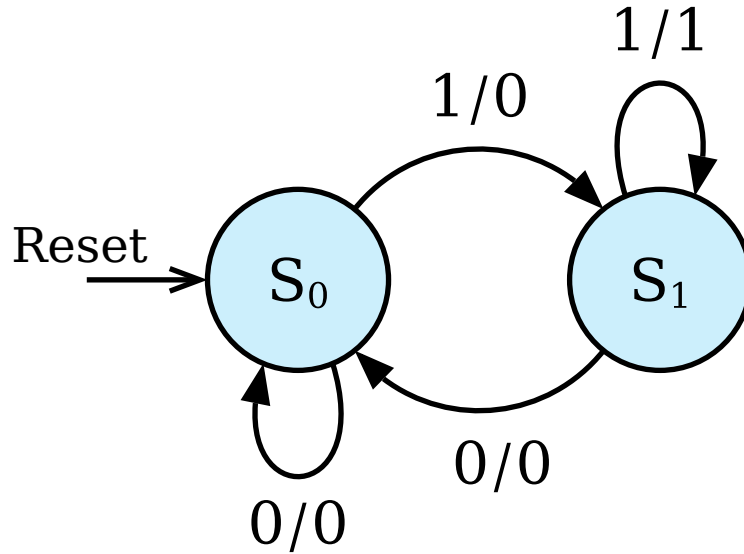
- Don't depend on input (at least not only)
- Consist of combinational parts as well as memory cells (Flip_Flop)
- Most famous applications :
 1. Sequence Detectors "comm. protocol"
 2. Control units

Finite State Machines "Internals"

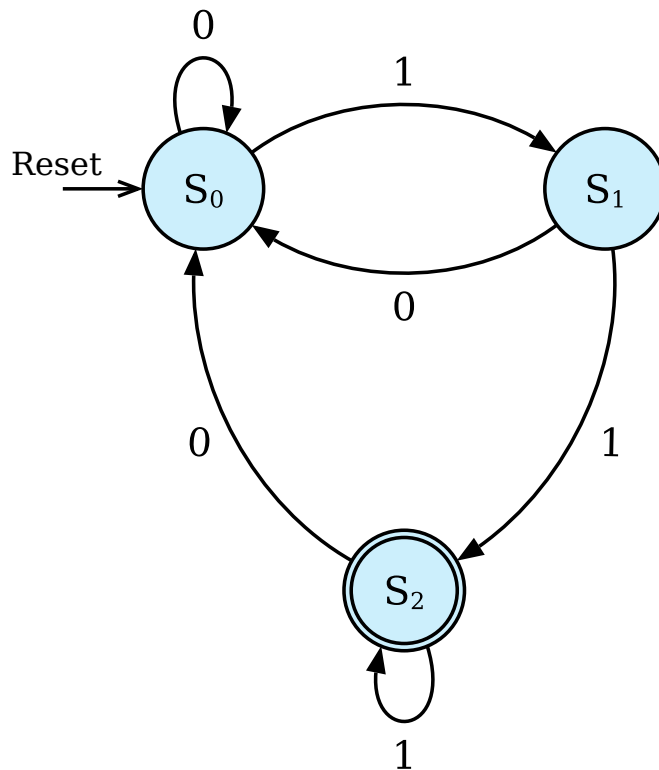


internal structure of FSM

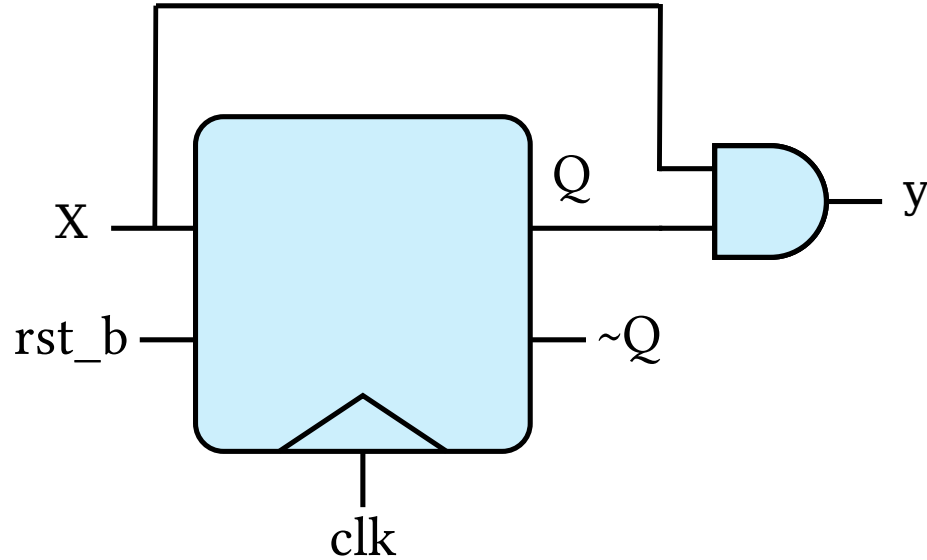
"11" Sequence detector - Mealy



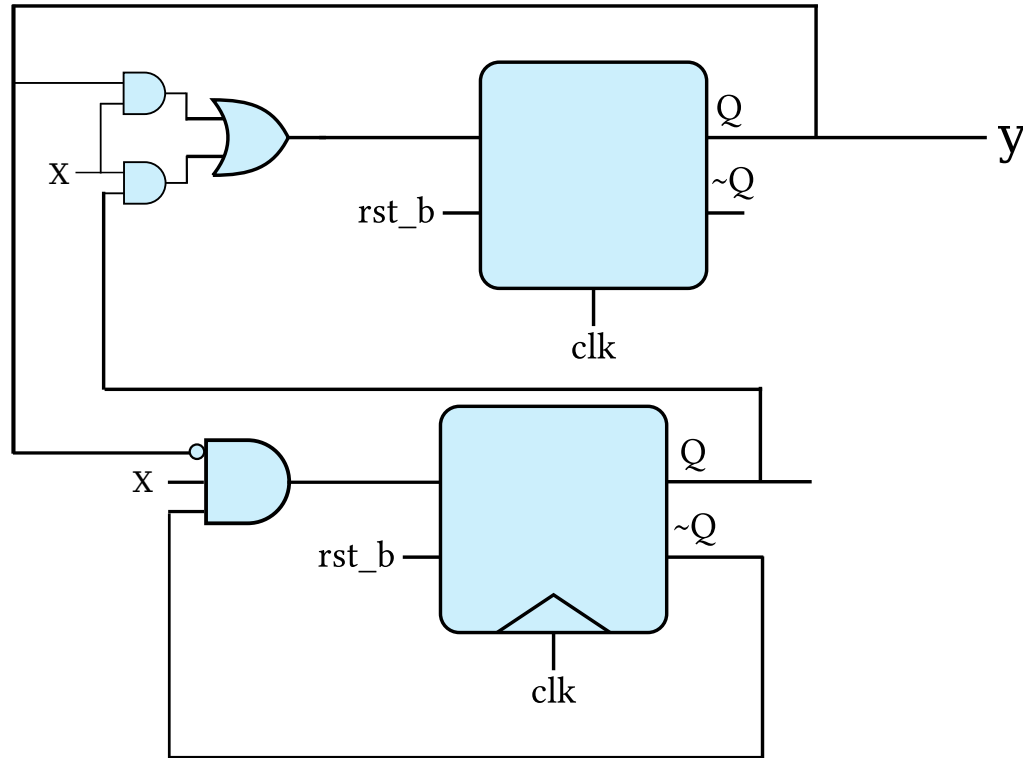
"11" Sequence detector - Moore



"11" Sequence detector - Mealy

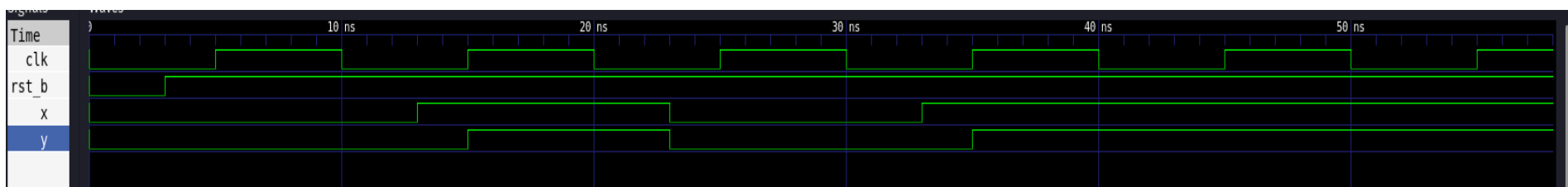


"11" Sequence detector - Moore

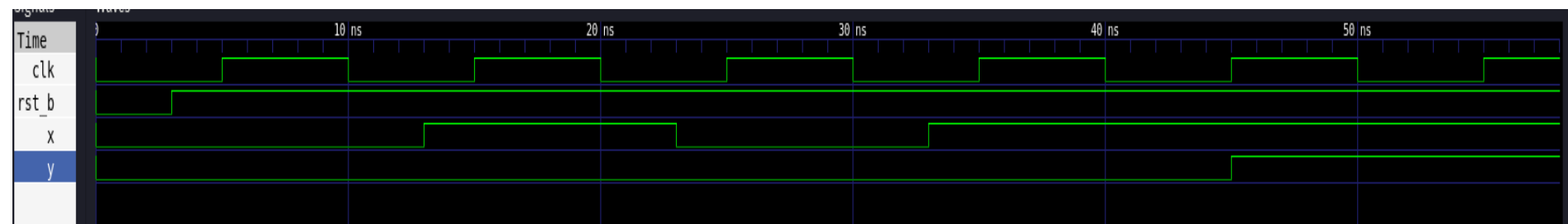


```
1 module moore_seq(  
2     input x,  
3     input clk,  
4     input rst_b,  
5     output y  
6 );  
7  
8     localparam S0= 2'b00;  
9     localparam S1= 2'b01;  
10    localparam S2= 2'b10;  
11  
12    reg [1:0] NextState, PresentState;  
13  
14    always @(posedge clk or negedge rst_b) begin  
15        if(!rst_b) PresentState <= S0;  
16        else PresentState <= NextState;  
17    end  
18  
19    always @(x,PresentState) begin  
20        case(PresentState)  
21            S0: NextState = (x)? S1:S0;  
22            S1: NextState = (x)? S2:S0;  
23            S2: NextState = (x)? S2:S0;  
24        endcase  
25    end  
26  
27    assign y=(PresentState==S2);  
28  
29 endmodule
```

```
1 module mealy_seq(  
2     input x,  
3     input clk,  
4     input rst_b,  
5     output y  
6 );  
7  
8     localparam S0= 1'b0;  
9     localparam S1= 1'b1;  
10  
11     reg NextState, PresentState;  
12  
13     always @(posedge clk or negedge rst_b) begin  
14         if(!rst_b) PresentState <= S0;  
15         else PresentState <= NextState;  
16     end  
17  
18     always @(x,PresentState) begin  
19         case(PresentState)  
20             S0: NextState = (x)? S1:S0;  
21             S1: NextState = (x)? S1:S0;  
22         endcase  
23     end  
24  
25     assign y = (PresentState == S1) & (x==1);  
26  
27 endmodule
```

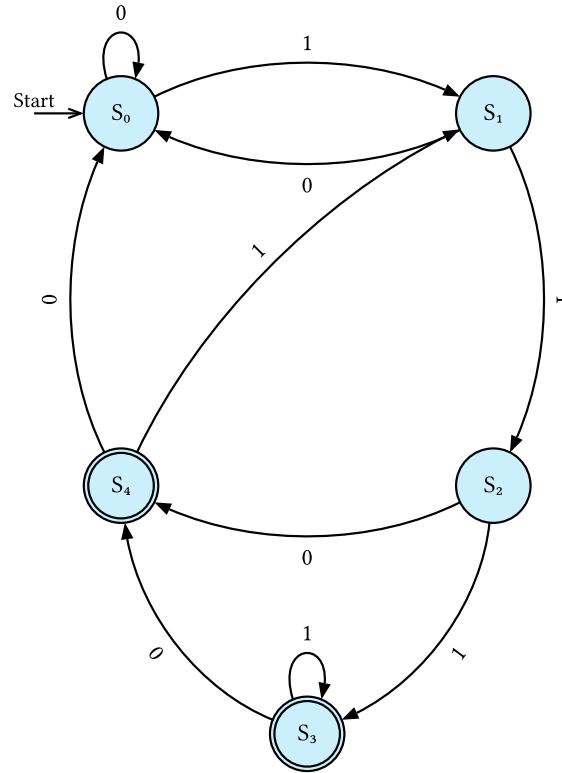


Mealy Output Waveform

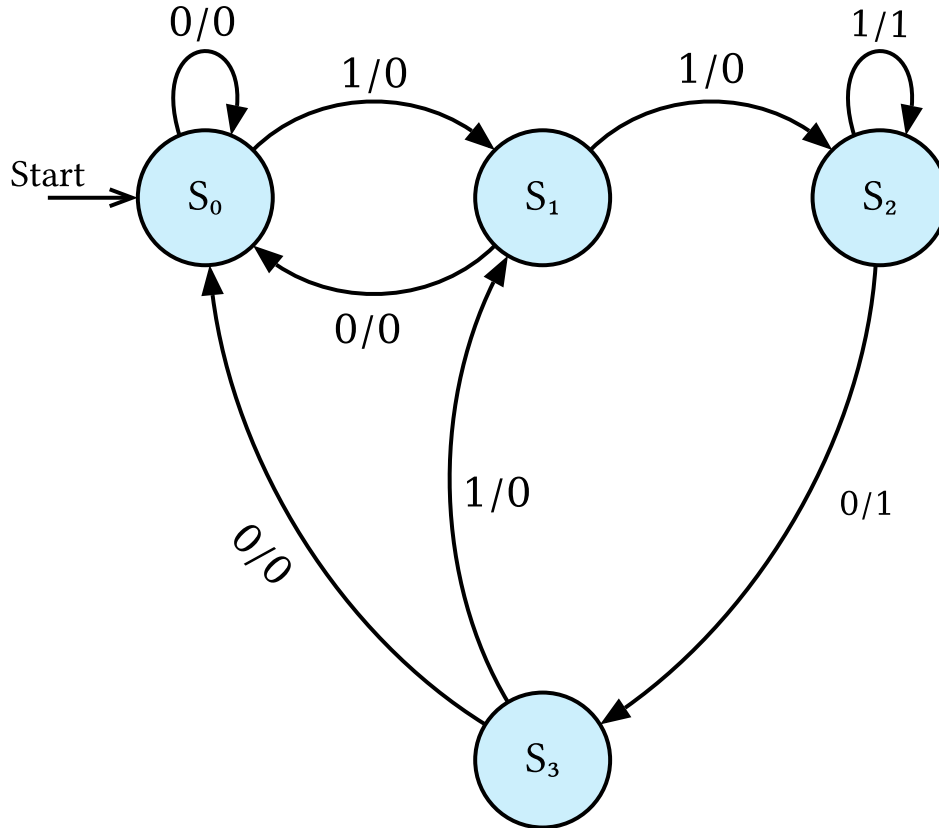


Moore Output Waveform

Complex FSM "110/111" detector - Moore



Complex FSM "110/111" detector -Mealy



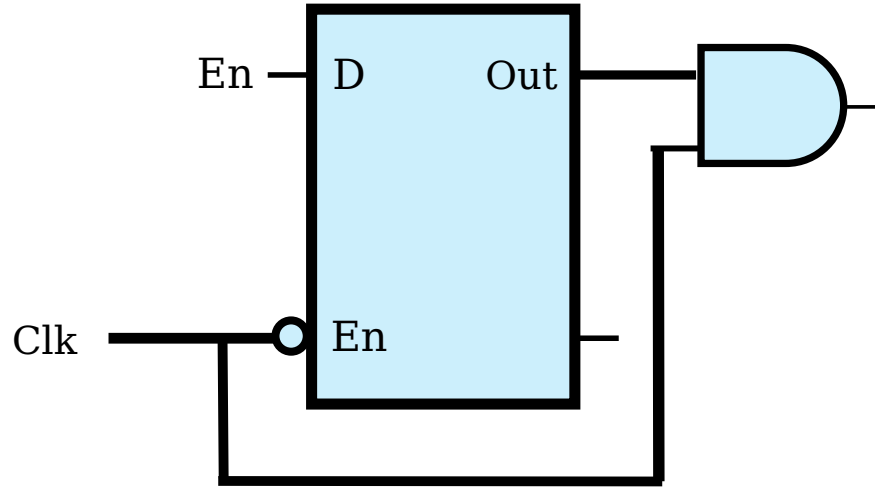
State Encodings

-- Binary Encoding

-- One-Hot Encoding

-- Gray Encoding

Clock Gating using Latches



Clock Gating using Latches

verilog

```
1 module clk_gate(  
2     input clk,  
3     input en,  
4     output clk_gated  
5 );  
6  
7     reg out;  
8  
9     always @(*) begin  
10         if(!clk) out = en;  
11     end  
12  
13     assign clk_gated = (out & clk);  
14  
15 endmodule
```