

Modelarea unui circuit secvențial în Verilog

Finite State Machines – FSMs

- Mealy machine – valorile de ieșire sunt determinate atât de starea curentă cât și de valorile input-urilor

Out = f(current_state, input) => rapid

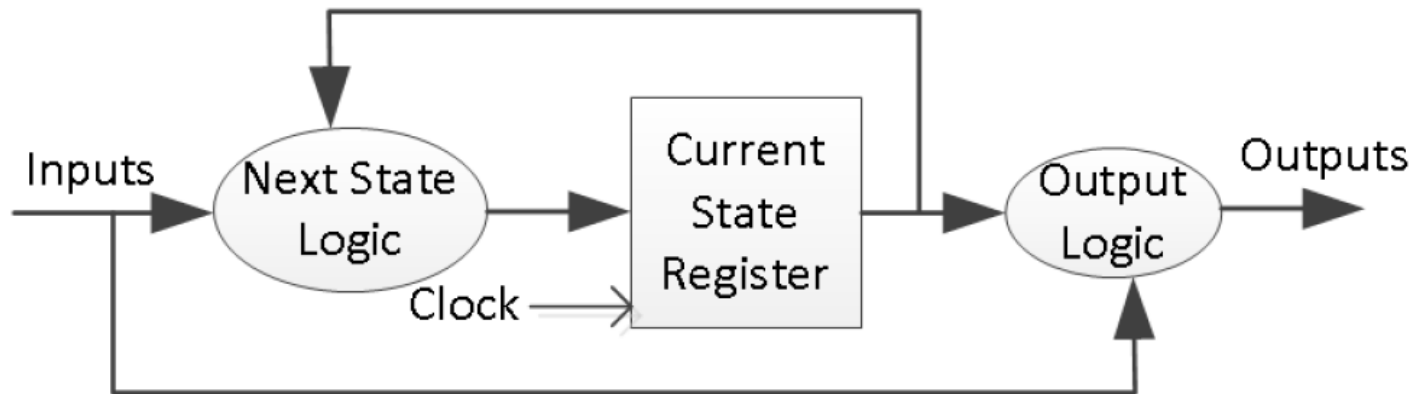
- Moore machine – valorile de ieșire sunt determinate doar de starea curentă

Out = f(current_state) => simplu

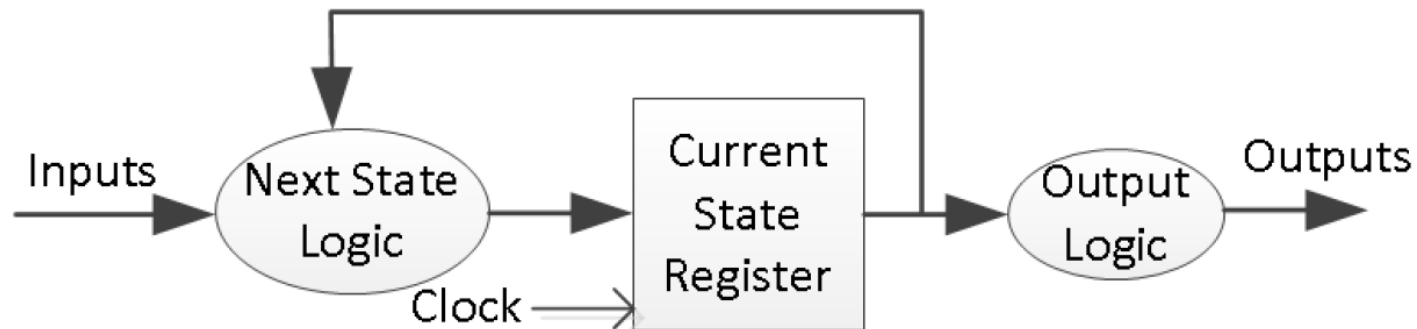
Fiecare mașină de stări constă din 3 părți:

- Logica pentru starea următoare
- Logica registrului de stare
- Logica ieșirii

Mealy machine

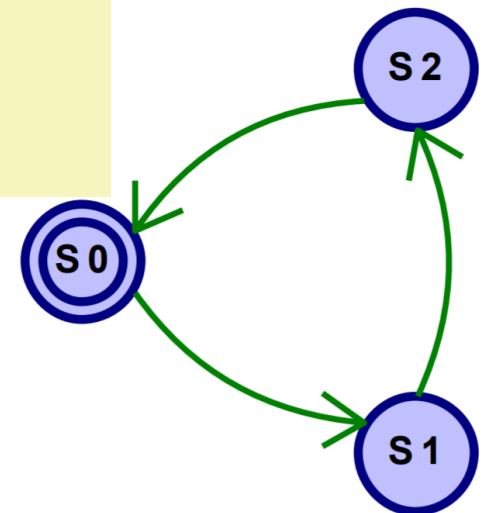


Moore machine



Exemplu Moore machine:

```
module divideby3FSM (input clk, input reset, output q);  
    reg [1:0] state, nextstate;  
  
    parameter S0 = 2'b00;  
    parameter S1 = 2'b01;  
    parameter S2 = 2'b10;  
  
    always @ (posedge clk, posedge reset) // state register  
        if (reset) state <= S0;  
        else      state <= nextstate;  
    always @ (*) // next state logic  
        case (state)  
            S0:      nextstate = S1;  
            S1:      nextstate = S2;  
            S2:      nextstate = S0;  
            default: nextstate = S0;  
        endcase  
    assign q = (state == S0); // output logic  
endmodule
```



Exemplu Mealy machine:

Numărător pe 2 biți care va număra fie crescător, fie descrescător. Circuitul va avea o intrare *op* :

1. *op*="1": 0-1-2-3-0-1-2-3

2. *op*="0": 0-3-2-1-0-3-2-1

```
module counter_2_bit (op, clk, rst, count);
input op, clk, rst;
output [1:0] count;
reg [1:0] count, nextcount;
always @ (posedge clk or negedge rst)
    if (rst)
        count<=0;
    else
        count<=nextcount;
always @ (op or count)
case (count)
0: begin
    if (op) nextcount =1;
    else nextcount =3;
end
```

```
1: begin
    if (op) nextcount =2;
    else nextcount =0;
end
2: begin
    if (op) nextcount =3;
    else nextcount =1;
end
3: begin
    if (op) nextcount =0;
    else nextcount =2;
end
default nextcount=0;

encase
endmodule
```