Digital Design

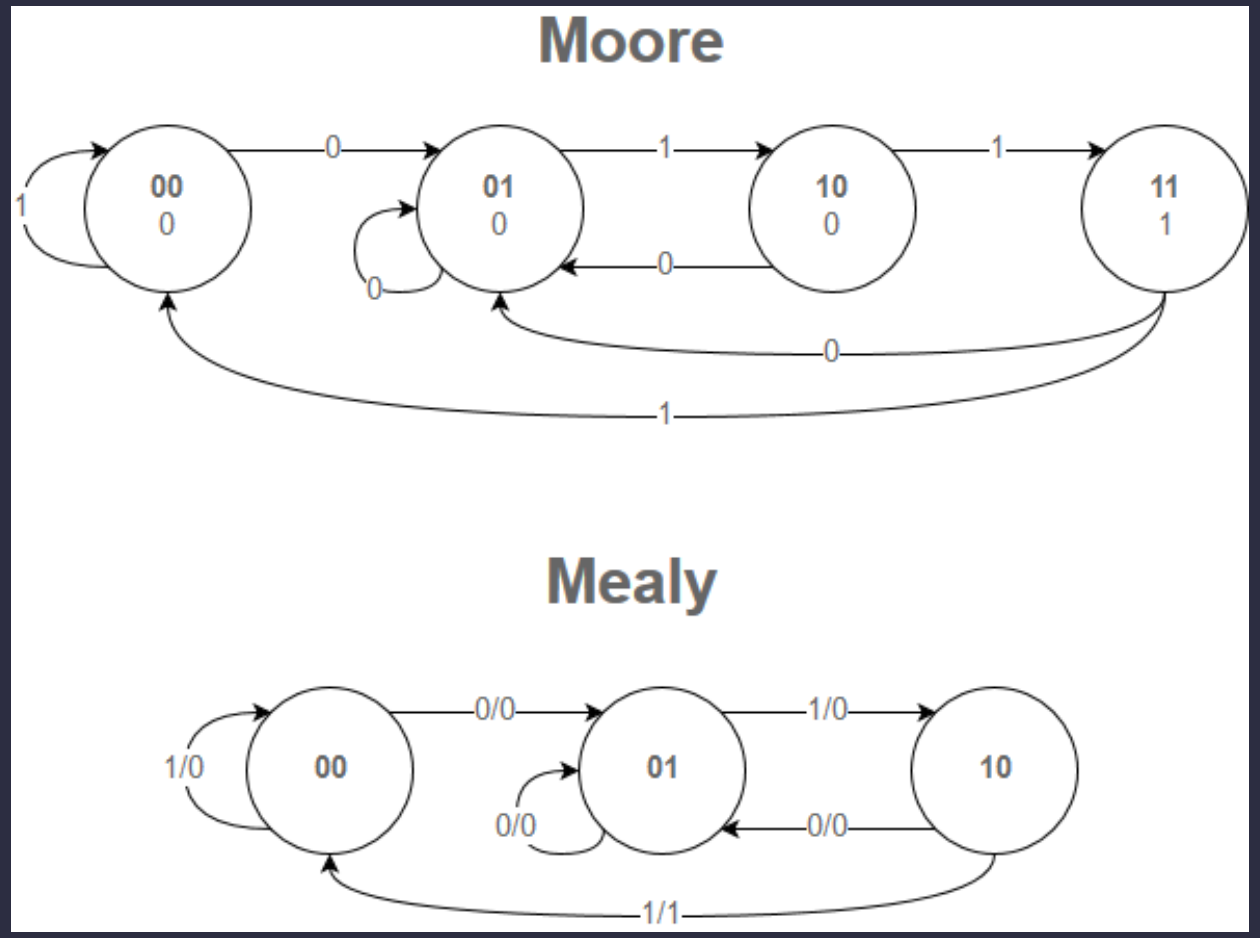
Week 6: State Machines
Part II

Fenerbahce University

Course

- State Machines
  - State Machines Verilog Representations

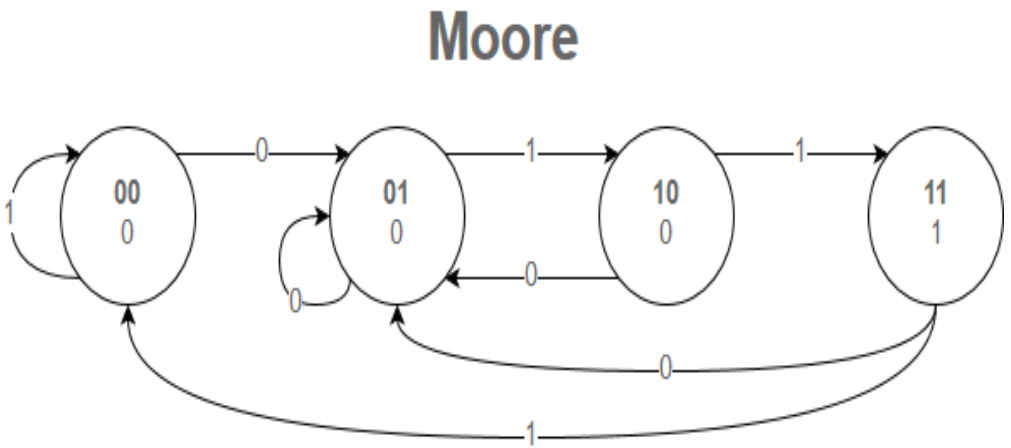
Durum Makinaları



Moore State Machine Examples

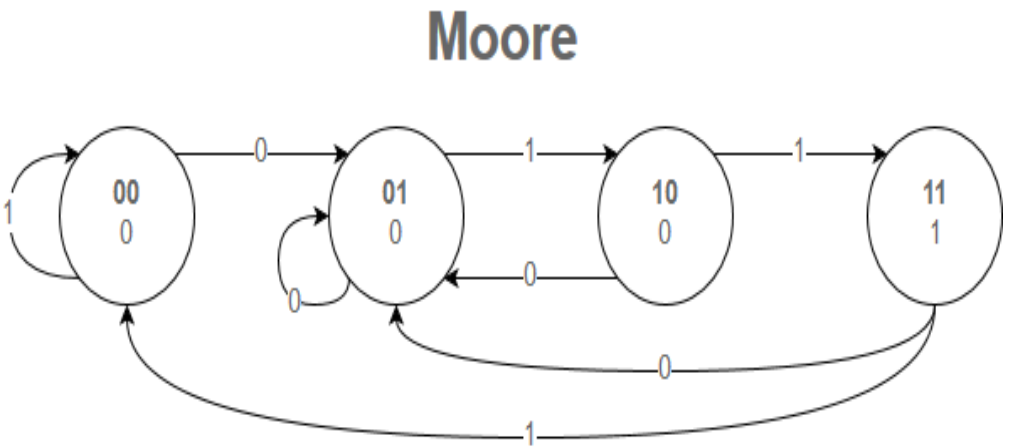
State Machines

```
module stateMachineTest (input clk, input in, output reg out)

reg [1:0] state, stateNext;
reg outNext;

initial begin
        state = 0;
        stateNext = 0;
        out = 0;
        outNext = 0;
end

always@(posedge clk) begin
        state <= stateNext;
        out <= outNext;
end
```



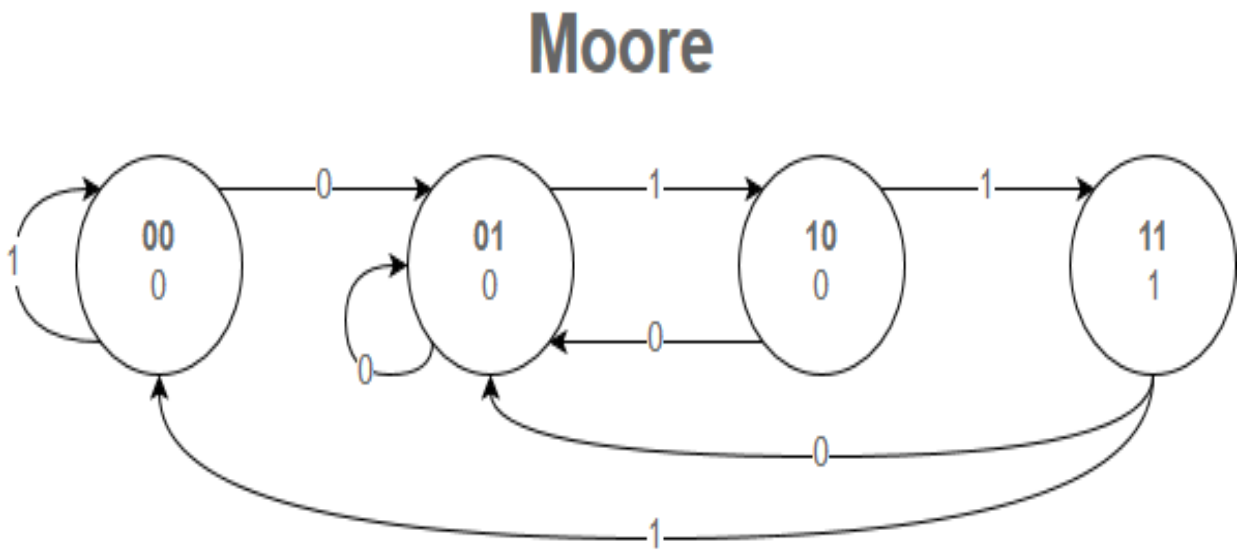
State Machines

```
always@(*) begin
        stateNext = state;
        outNext = out;
        case(state)
                0: begin
                        if(in == 0)begin
                                stateNext = 1;
                                outNext = 0;
                        end
                end
        endcase
end

endmodule
```



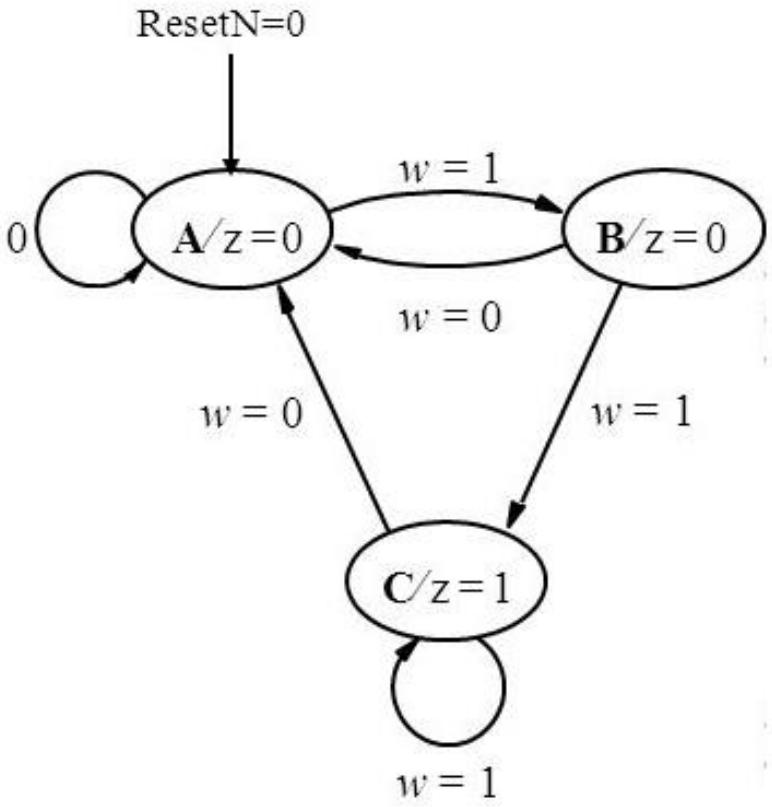
State Machines

```
1: begin
        if(in == 1)begin
                stateNext = 2;
                outNext = 0;
        end
end
2: begin
        if(in == 1)begin
                stateNext = 3;
                outNext = 1;
        end else begin
                stateNext = 1;
                outNext = 0;
        end
end
3: begin
        if(in == 0)begin
                stateNext = 1;
                outNext = 0;
        end else begin
                stateNext = 0;
                outNext = 0;
        end
end
```



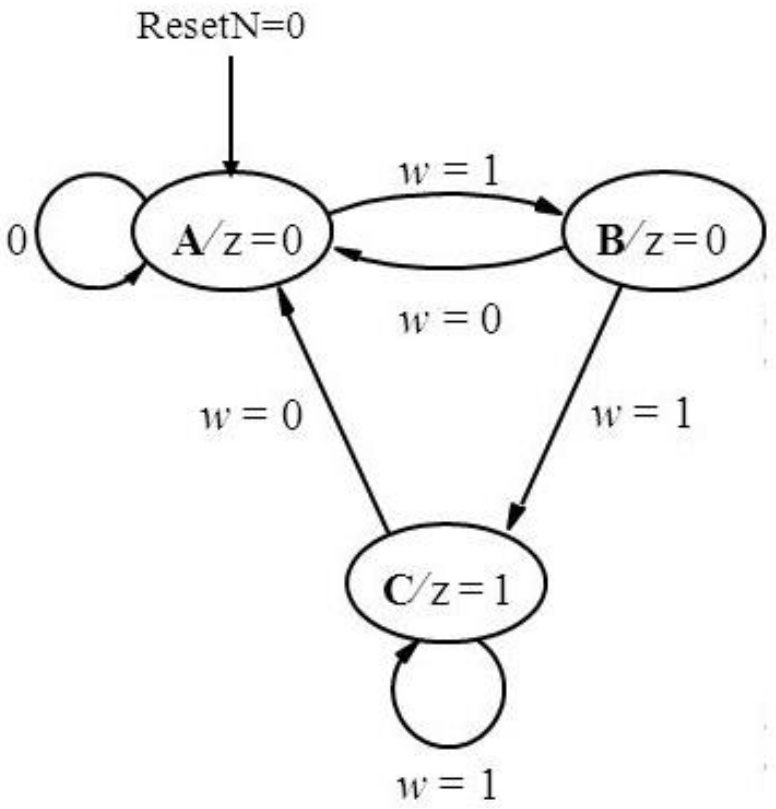
State Machines

```
module mooreMachine2 (input clk, input ResetN, input w, output reg z)

reg [1:0] state, stateNext;
reg zNext;

initial begin
        state = 0;
        stateNext = 0;
        z = 0;
        zNext = 0;
end

always@(posedge clk) begin
        state <= stateNext;
        z <= zNext;
end
```



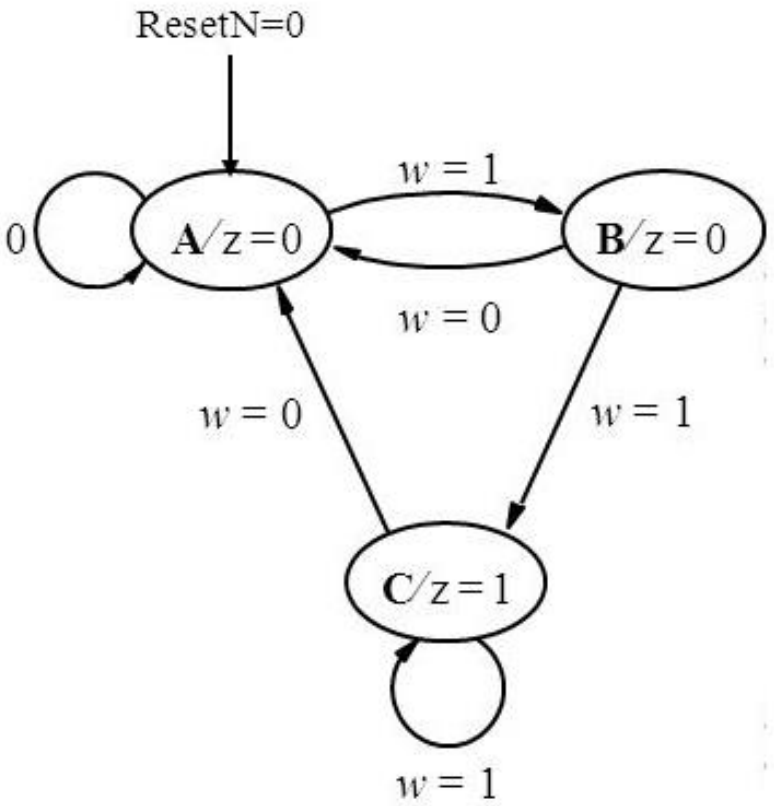
State Machines

```
always@(*) begin
        stateNext = state;
        zNext = z;
        if(ResetN) begin
                stateNext = 0;
                zNext = 0;
        end else begin
                case(state)
                        0: begin
                                if(w == 1)begin
                                        stateNext = 1;
                                        zNext = 0;
                                end
                        end
                endcase
        end
end
endmodule
```



State Machines

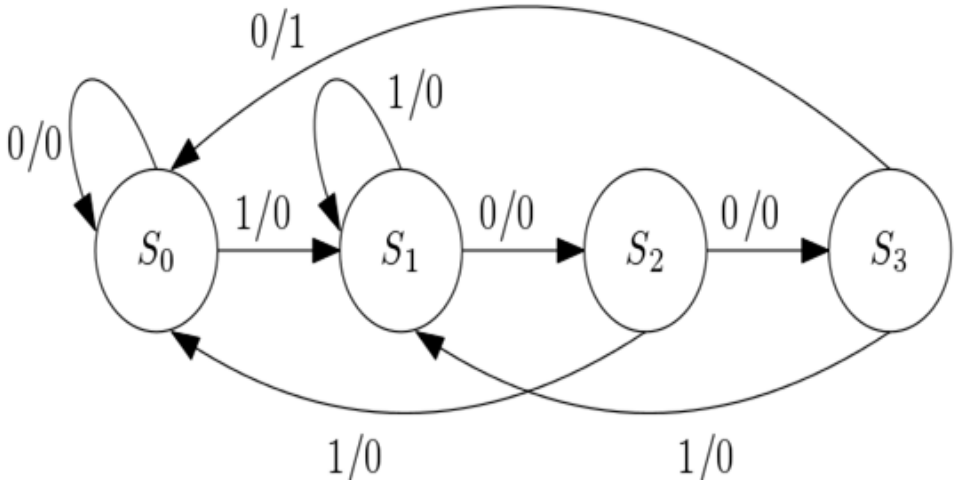
```
1: begin
        if(w == 0)begin
                stateNext = 0;
                zNext = 0;
        end else begin
                stateNext = 2;
                zNext = 1;
        end
end
2: begin
        if(w == 0)begin
                stateNext = 0;
                zNext = 0;
        end
end
```



Mealy State Machine Examples

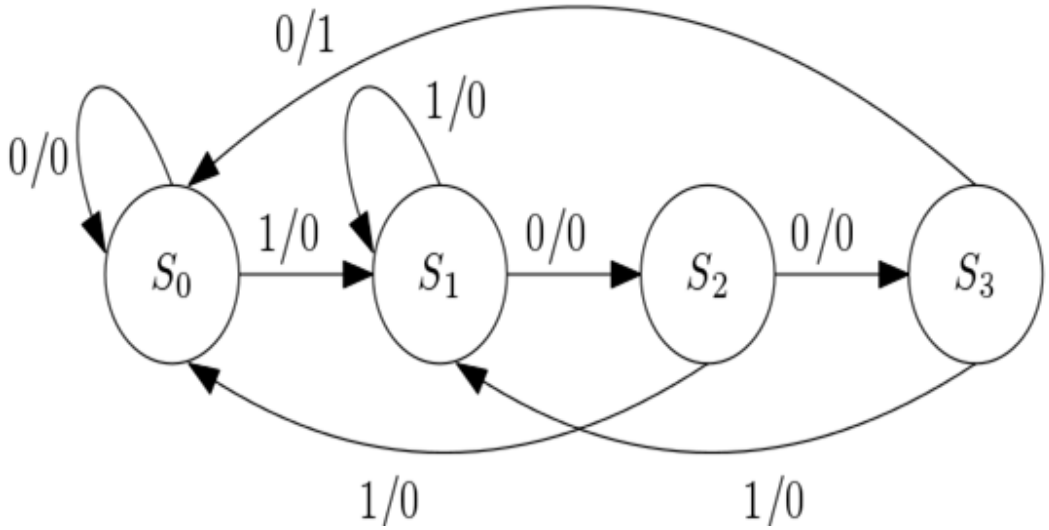
State Machines

```
module mooreMachine1 (input clk, input in, output reg out)

reg [1:0] state, stateNext;

initial begin
        state = 0;
        stateNext = 0;
        out = 0;
end

always@(posedge clk) begin
        state <= stateNext;
end
```



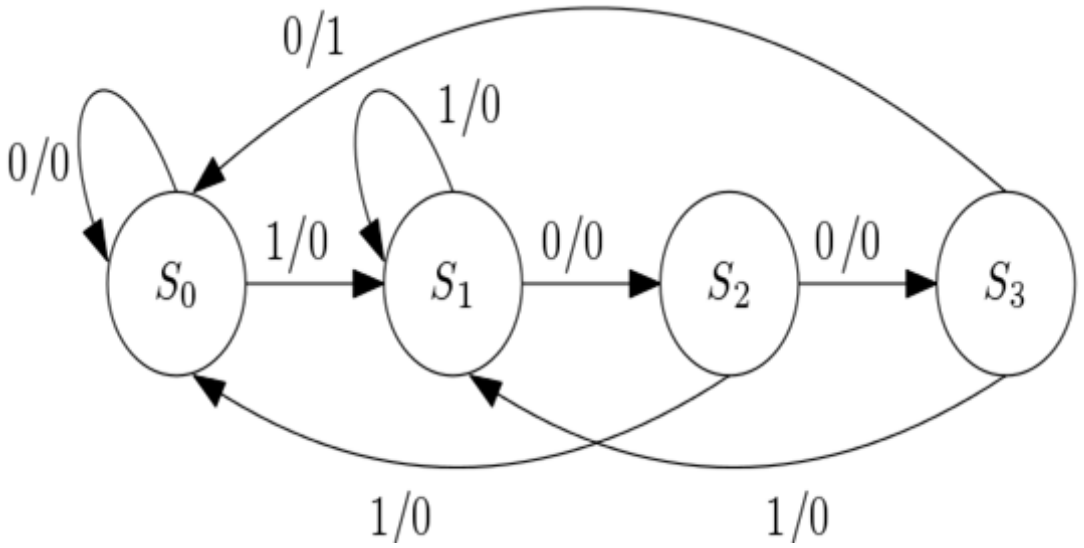
State Machines

```
always@(*) begin
        stateNext = state;
        out = 0;
        case(state)
                0: begin
                        if(in == 0)begin
                                out = 0;
                        end else begin
                                out = 1;
                                stateNext = 1;
                        end
                end
        endcase
end
```



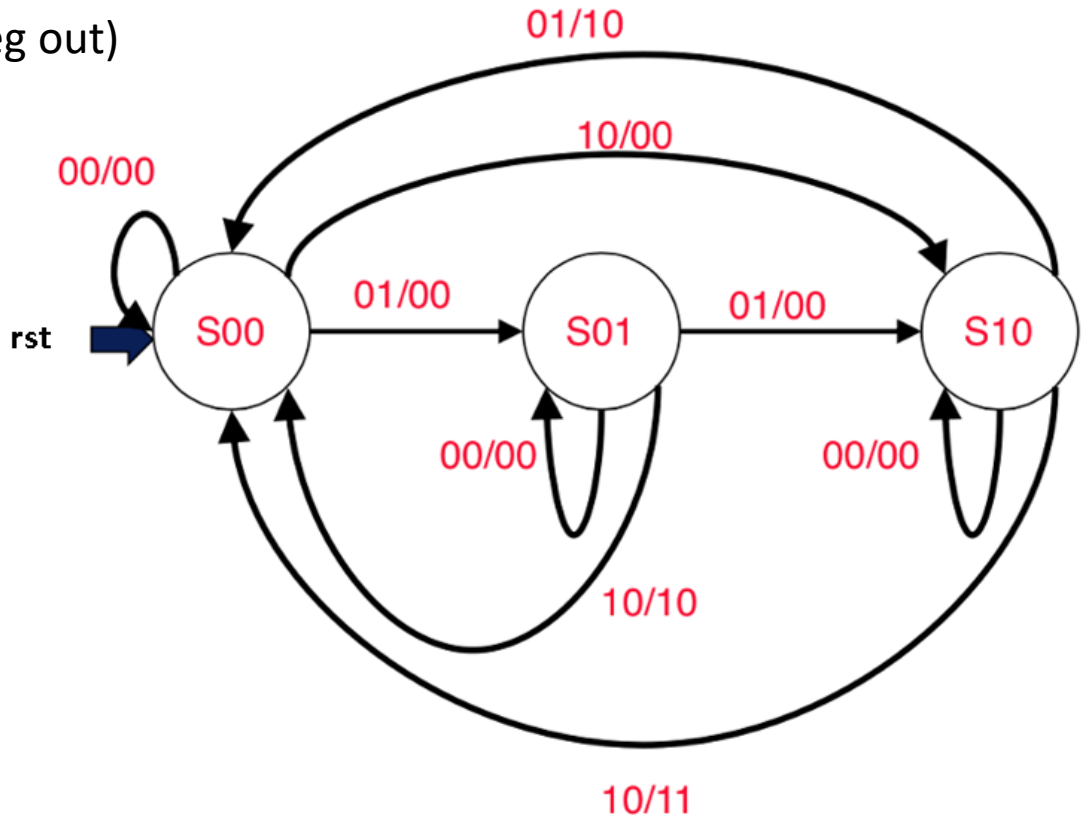
State Machines

```
1: begin
        if(in == 0)begin
                out = 0;
                stateNext = 2;
        end else begin
                out = 0;
        end
end
2: begin
        if(in == 0)begin
                out = 0;
                stateNext = 3;
        end else begin
                out = 0;
                stateNext = 0;
        end
end
3: begin
        if(in == 0)begin
                out = 1;
                stateNext = 0;
        end else begin
                out = 0;
                stateNext = 1;
        end
end
```



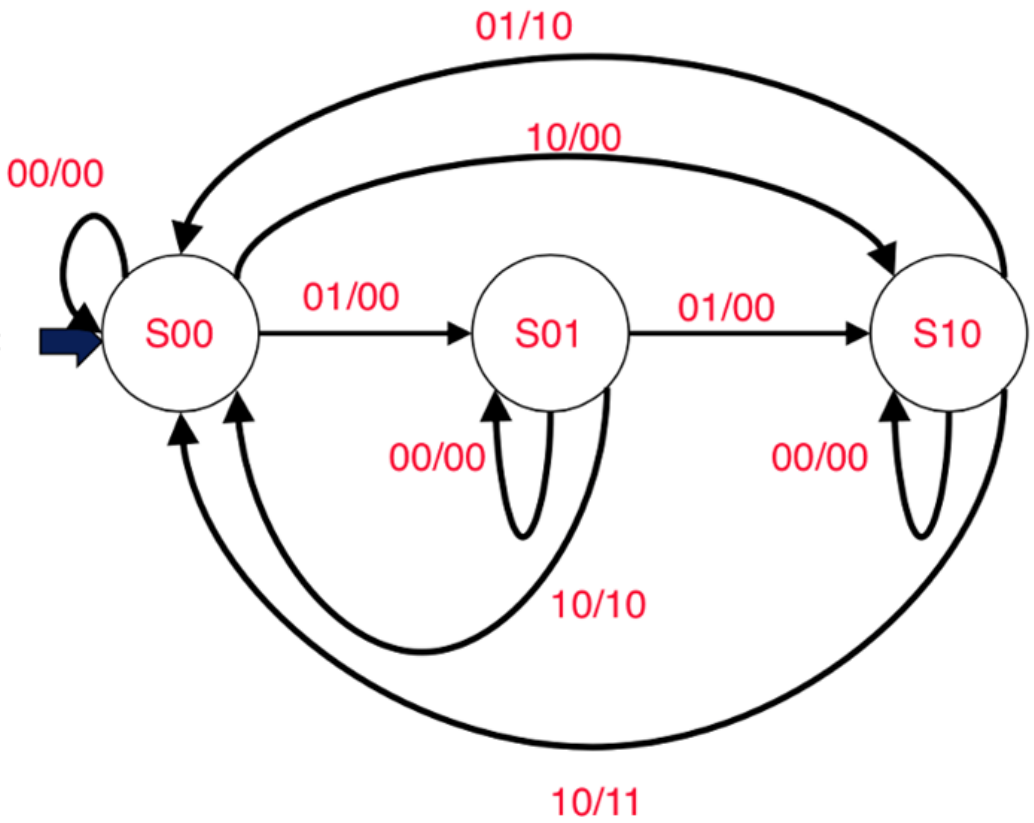
State Machines

```
module mooreMachine2 (input clk, input in, output reg out)

reg [1:0] state, stateNext;

initial begin
        state = 0;
        stateNext = 0;
        out = 0;
end

always@(posedge clk) begin
        state <= stateNext;
end
```



State Machines

```
always@(*) begin
        stateNext = state;
        out = 0;
        if(rst) begin
                stateNext = 0;
                out = 0;
        end else begin
                case(state)
                        0: begin
                                if(in == 0)begin
                                        out = 0;
                                end else if(in == 1) begin
                                        out = 0;
                                        stateNext = 1;
                                end else if(in == 2) begin
                                        out = 0;
                                        stateNext = 2;
                                end
                        end
                                        endcase
        end
end

endmodule
```



State Machines

```
1: begin
            if(in == 0)begin
                        out = 0;
            end else if(in == 1) begin
                        out = 0;
                        stateNext = 2;
            end else if(in == 2) begin
                        out = 2;
                        stateNext = 2;
            end
end
2: begin
            if(in == 0)begin
                        out = 0;
            end else if(in == 1) begin
                        out = 2;
                        stateNext = 0;
            end else if(in == 2) begin
                        out = 3;
                        stateNext = 0;
            end
end
```