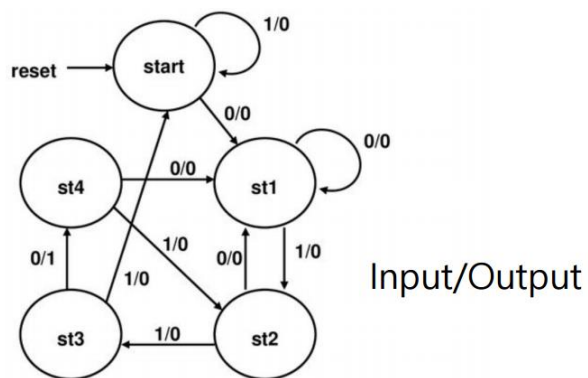


1. Mealy FSM 에 대한 간단한 설명

Mealy machine 은 output 이 현재 state 와 입력에 의해 결정됩니다. 이번 과제의 경우 0110 이 입력되면 output 이 1 이 나와야 하므로 그림 같은 state 들을 가지고 있고 st3 에서 input 이 0 이라면 output 이 1 이 나와야 합니다.



2. 코드, 시뮬레이션 waveform 스크린샷 첨부

```
module meely_FSM_bychangmin (
    input clk,      // clk
    input rstn,     // 리셋
    input din_bit,  //input, 0110 이어야함

    output reg detect_out // output
);

// ST state 를 나타내는 파라미터 정의, START 를 state0 으로 정의
parameter ST0 = 0, ST1 = 1, ST2 = 2, ST3 = 3, ST4 = 4;

// curr_state 정의, 초기는 ST0
reg [2:0] curr_state = ST0;

// 다음 state 정하기
always @(posedge clk , negedge rstn) begin
    case (curr_state)
        ST0: if (rstn) curr_state = ST0; else if(!din_bit)
            curr_state=ST1; else curr_state = ST0;
```

```

        //리셋이 1 이면 start, input 0 이면 다음 state, input 이 1 이면
start 에서 유지
        ST1: if (rstn) curr_state = ST0; else if(din_bit)
curr_state=ST2; else curr_state = ST1;
        //리셋이 1 이면 start, input 1 이면 다음 state, input 이 0 이면
st1 에서 유지, st1 인 이유는 다음 0110 의 시작이 될 수 있기 때문
        ST2: if (rstn) curr_state = ST0; else if(din_bit)
curr_state=ST3; else curr_state = ST1;
        //리셋이 1 이면 start, input 1 이면 다음 state, input 이 0 이면
st1 로 돌아감, st1 인 이유는 다음 0110 의 시작이 될 수 있기 때문
        ST3: if (rstn) curr_state = ST0; else if(!din_bit)
curr_state=ST4; else curr_state = ST0;
        //리셋이 1 이면 start, input 0 이면 다음 state, input 이 1 이면
start 로 돌아감
        ST4: if (rstn) curr_state = ST0; else if(din_bit)
curr_state=ST2; else curr_state = ST0;
        //리셋이 1 이면 start, input 1 이면 state2, input 이 0 이면
start 로 돌아감
    endcase
end

// detect_out 정의
always @(curr_state) begin
    case (curr_state)
        ST0: detect_out = 0;
        ST1: detect_out = 0;
        ST2: detect_out = 0;
        ST3: detect_out = 0;
        ST4: detect_out = 1;
    endcase
end

endmodule

```

```

module tb;
    reg clk, rstn, din_bit;
    wire detect_out;

    initial begin
        clk=0;
        forever begin
            #5 clk=!clk;
        end
    end

    meely_FSM_bychangmin meely_fsm_bychangmin(clk, rstn, din_bit,
detect_out);

```

```

initial begin
    din_bit=0;
    #10 din_bit=1;
    #10 din_bit=0;
    #10 din_bit=1;
    #10 din_bit=1;
    #10 din_bit=0;
    #10 din_bit=0;
    #10 din_bit=1;
    #10 din_bit=0;
    #10 din_bit=0;
    #10 din_bit=1;
    #10 din_bit=1;
    #10 din_bit=0;
    #10 din_bit=1;
    #10 din_bit=1;
    #10 din_bit=0;
    #10 $finish;
end

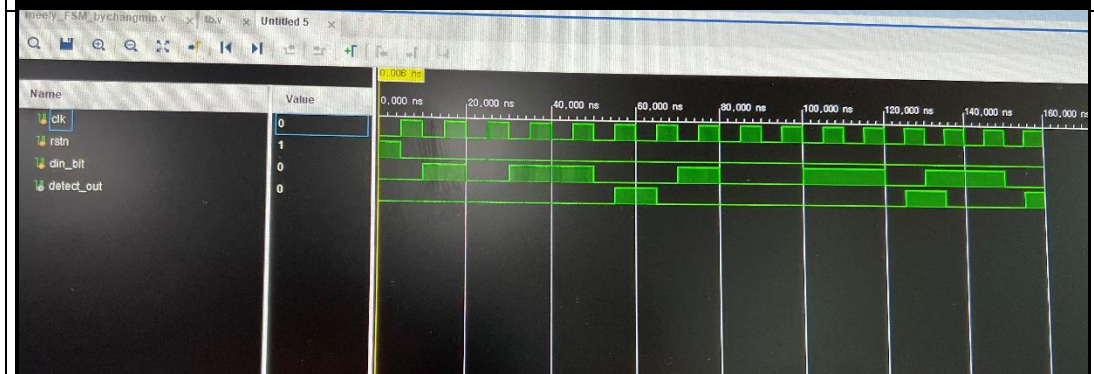
```

```

initial begin
    rstn = 1;
    #5 rstn=0;
end

```

endmodule



3. Moore FSM 과의 차이점에 대한 설명 (실습 때 했던 것과 과제와 비교 설명)

Mealy FSM 은 output이 현재 state와 입력에 의해 결정됩니다. Moore FSM은 output이 현재 state에 의해서만 결정됩니다. 실습 때 했던 Moore FSM은 ST0일 때 0, ST1일 때 1, 이런 식으로 했다고 하면 과제의 Mealy FSM은 ST3일 때 입력이 0이면 출력이 1이도록 했다는 차이가 있습니다. 제 코드에서는 ST4일 때 1을 출력하도록 하였지만 자세히 보면 state가 바뀔 때 체크를 하도록 하였기에 ST3에서 ST4로 넘어갈 때 1이 나오도록 하였고 ST3에서 0이 나와야 ST4로 넘어가도록 하였기에 'ST3일 때 입력이 0이면 출력이 1'과 등가라고 할 수 있습니다. 또한 ST4에서 다음 클럭에 반드시 start나 ST1으로 넘어가 유지되지도 않기에 Mealy FSM이라고 볼 수 있습니다.