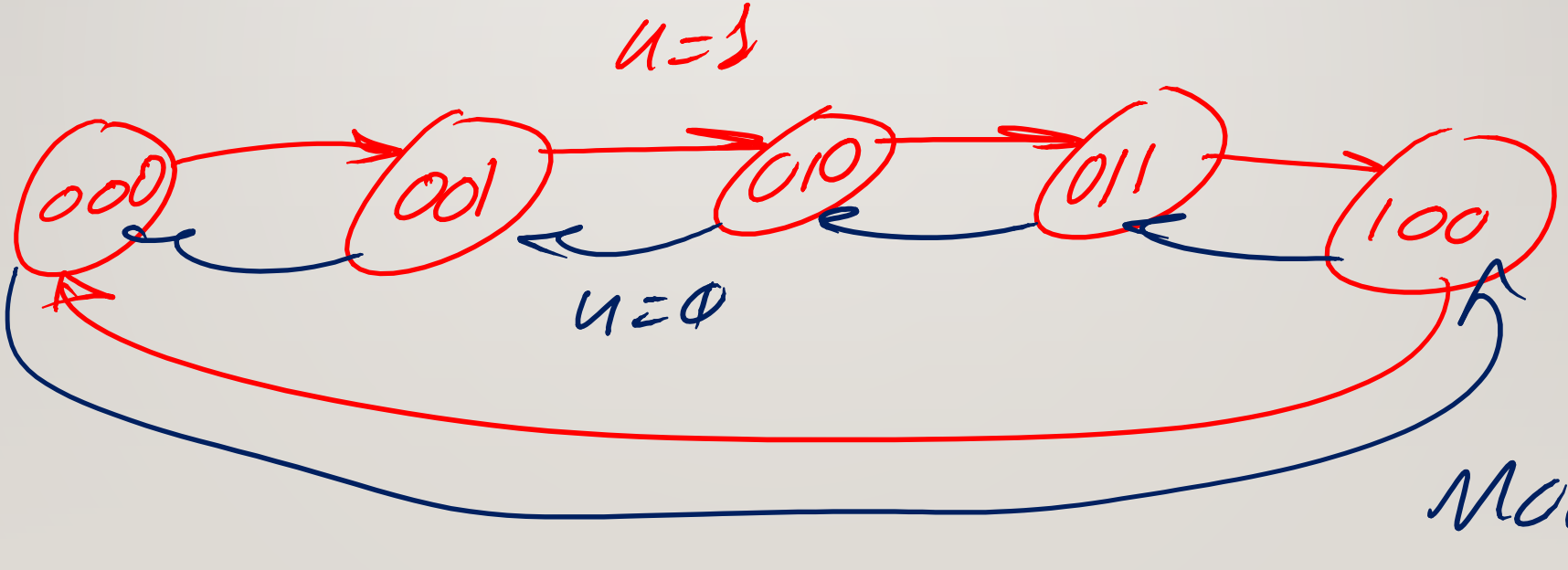
Digital Logic Design

Lecture 9

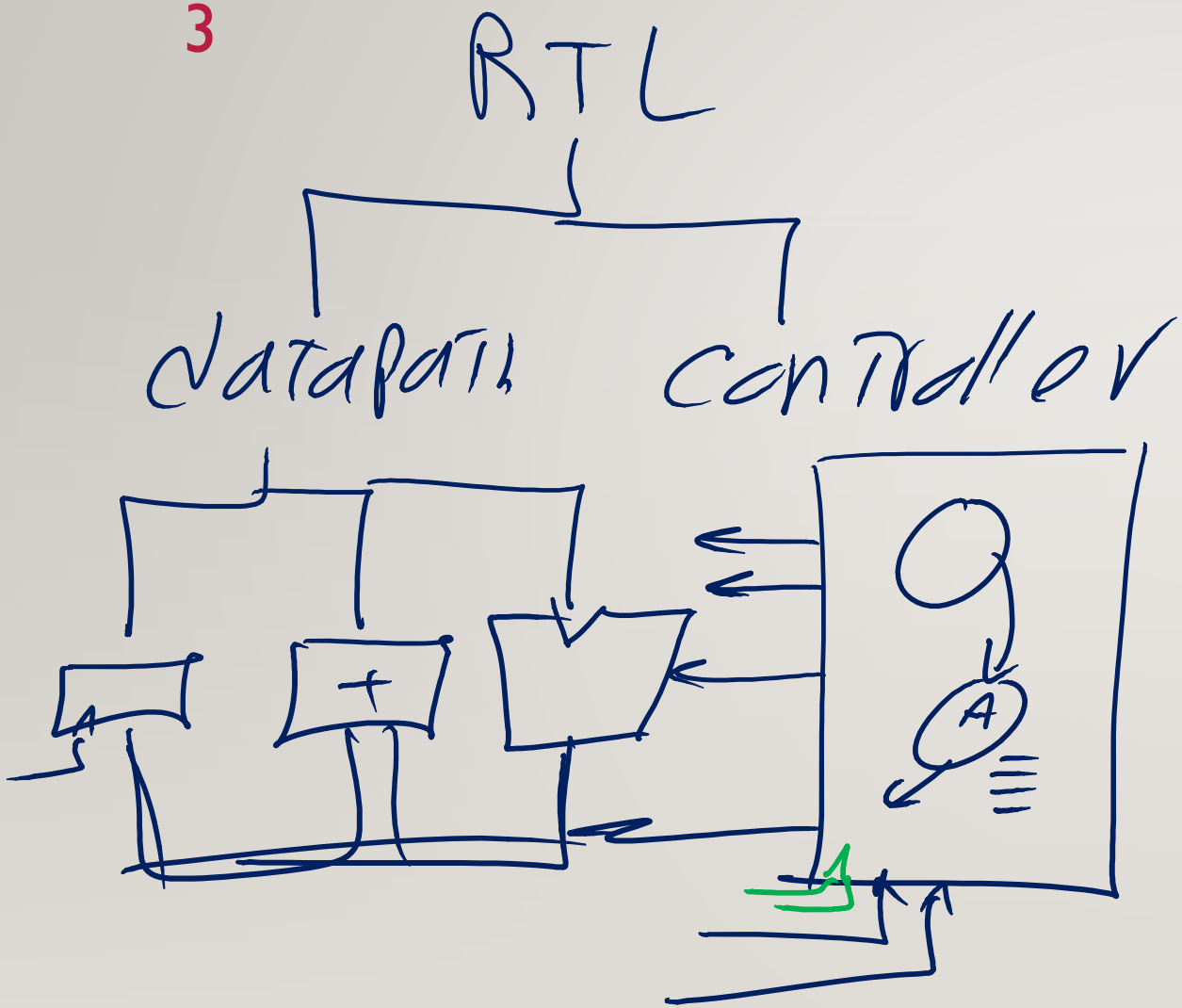
Dr. Navabi

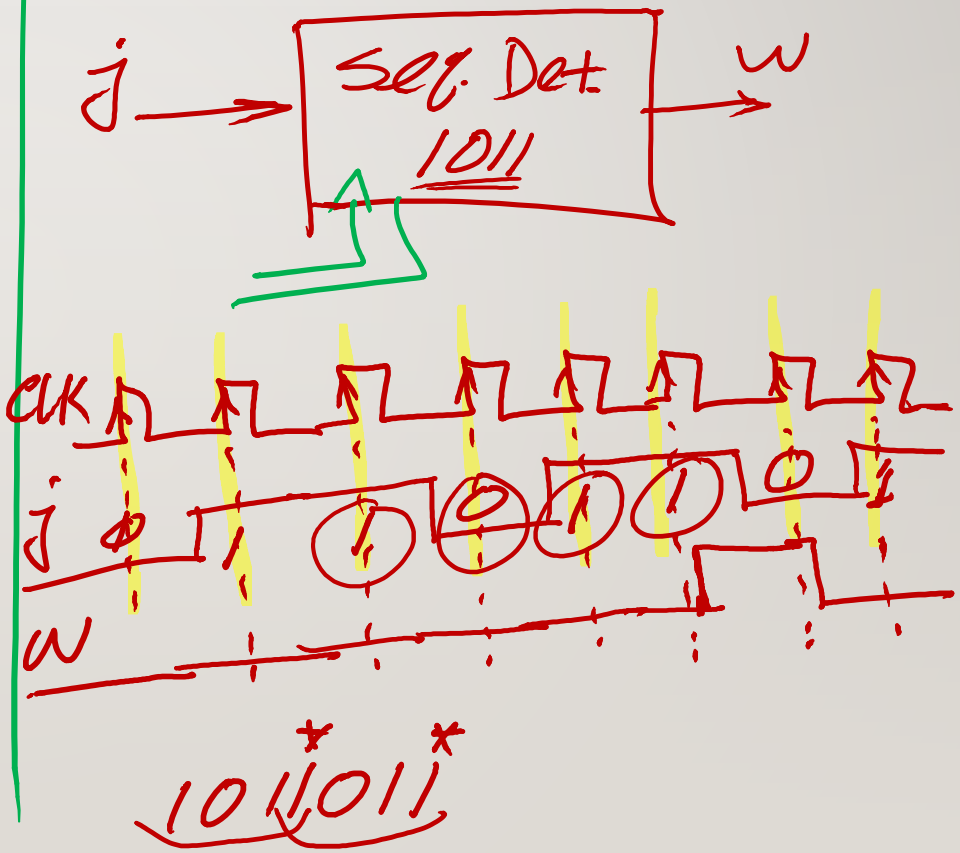
2 FSM & State machine

Sequence detector

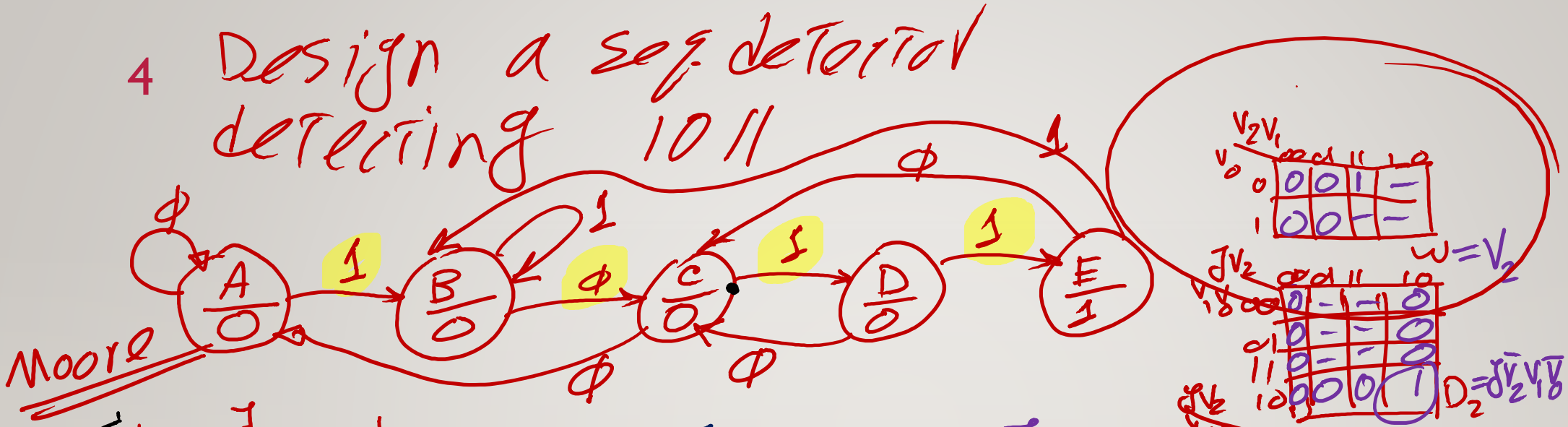


3





4 Design a seq detector detecting 1011



S	0	J	1	
000	A	A	B	0
001	B	C	B	0
011	C	A	D	0
010	D	C	E	0
110	E	C	B	1
	S^+			W

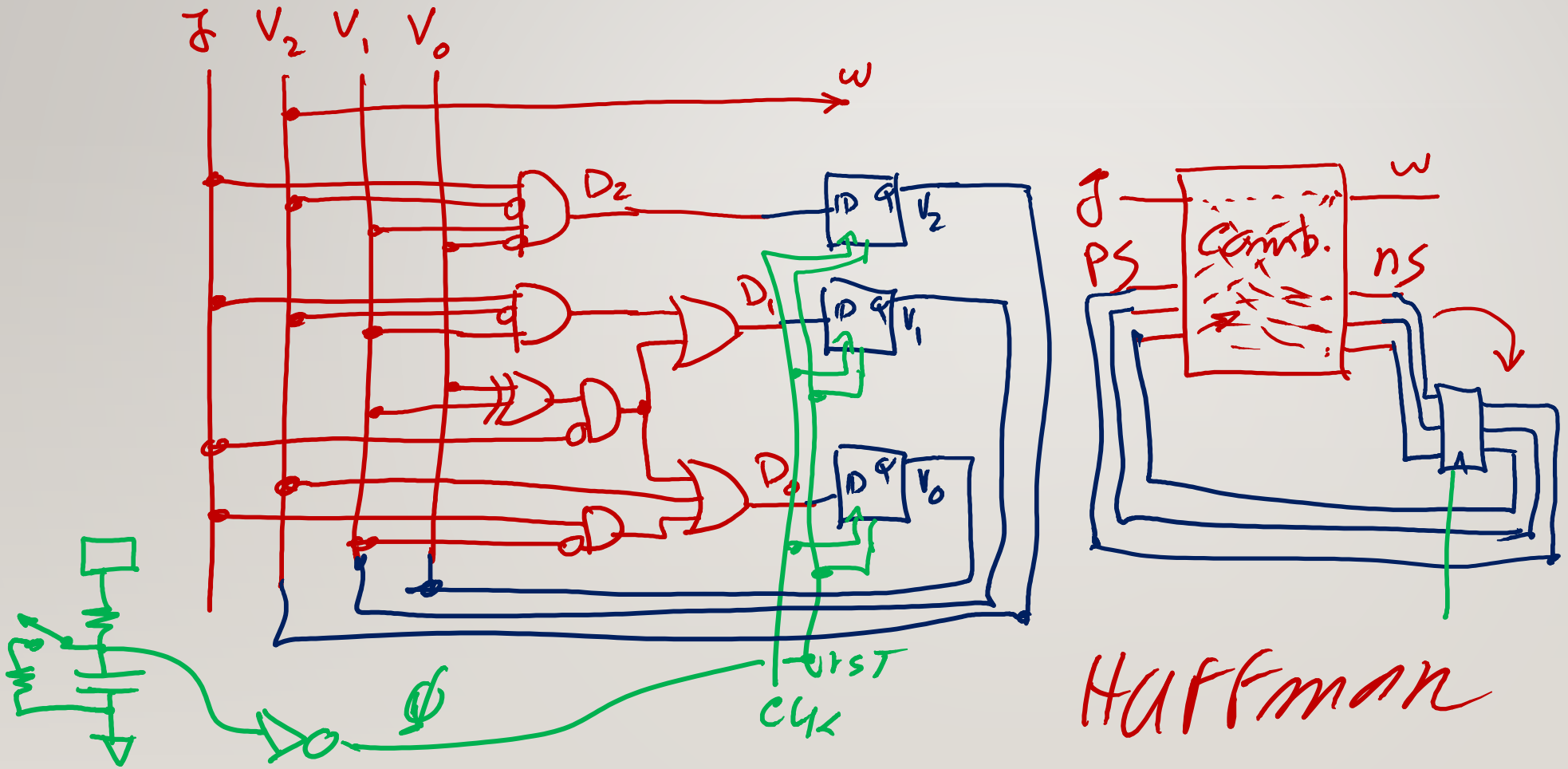
$V_2V_1V_0$	0	J	1	
000	000		001	0
001	011		001	0
011	000		010	0
010	011		110	0
110	011		001	1
	$V_2^+V_1^+V_0^+$			W

$V_2V_1V_0$	0	J	1	
000	000		001	0
001	011		001	0
011	000		010	0
010	011		110	0
110	011		001	1
	$D_2D_1D_0$			W

- Prob desc.
- State diag
- State Tab
- State Assign
- Transition Tab
- FF Type
- Excitation

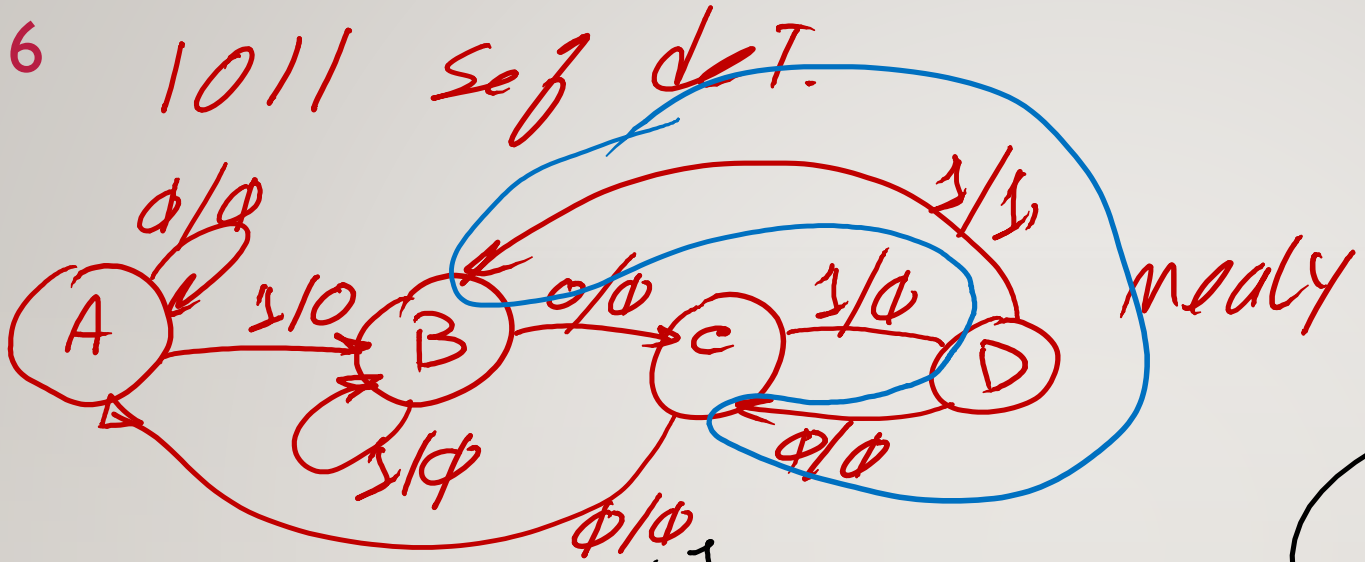
$D_2 = V_2^+$ for a DFF

5



Huffman

6 1011 Seq det.

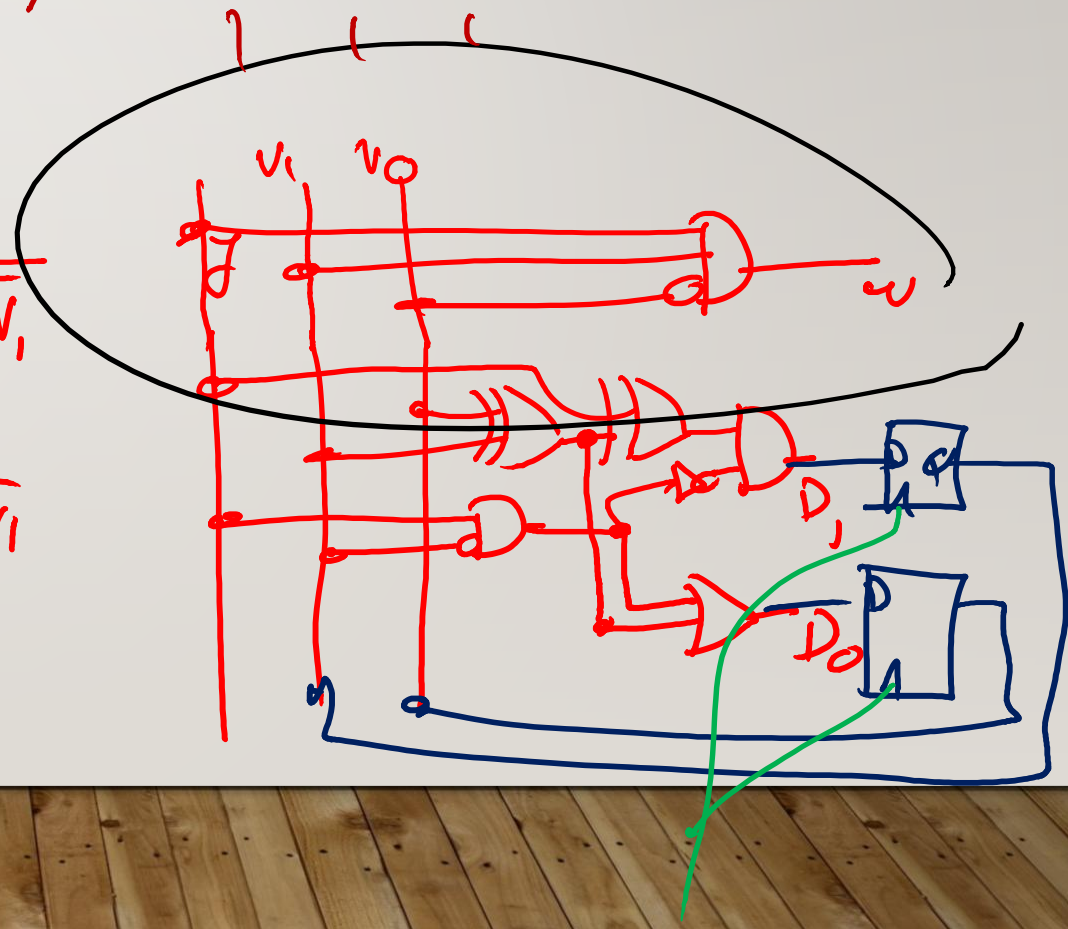


	v_1	v_0	J	v_1	J	v_0	w
00	A	A	00	B	01	0	0
01	B	C	11	B	01	0	0
11	C	A	00	D	10	0	0
10	D	C	11	B	01	0	1

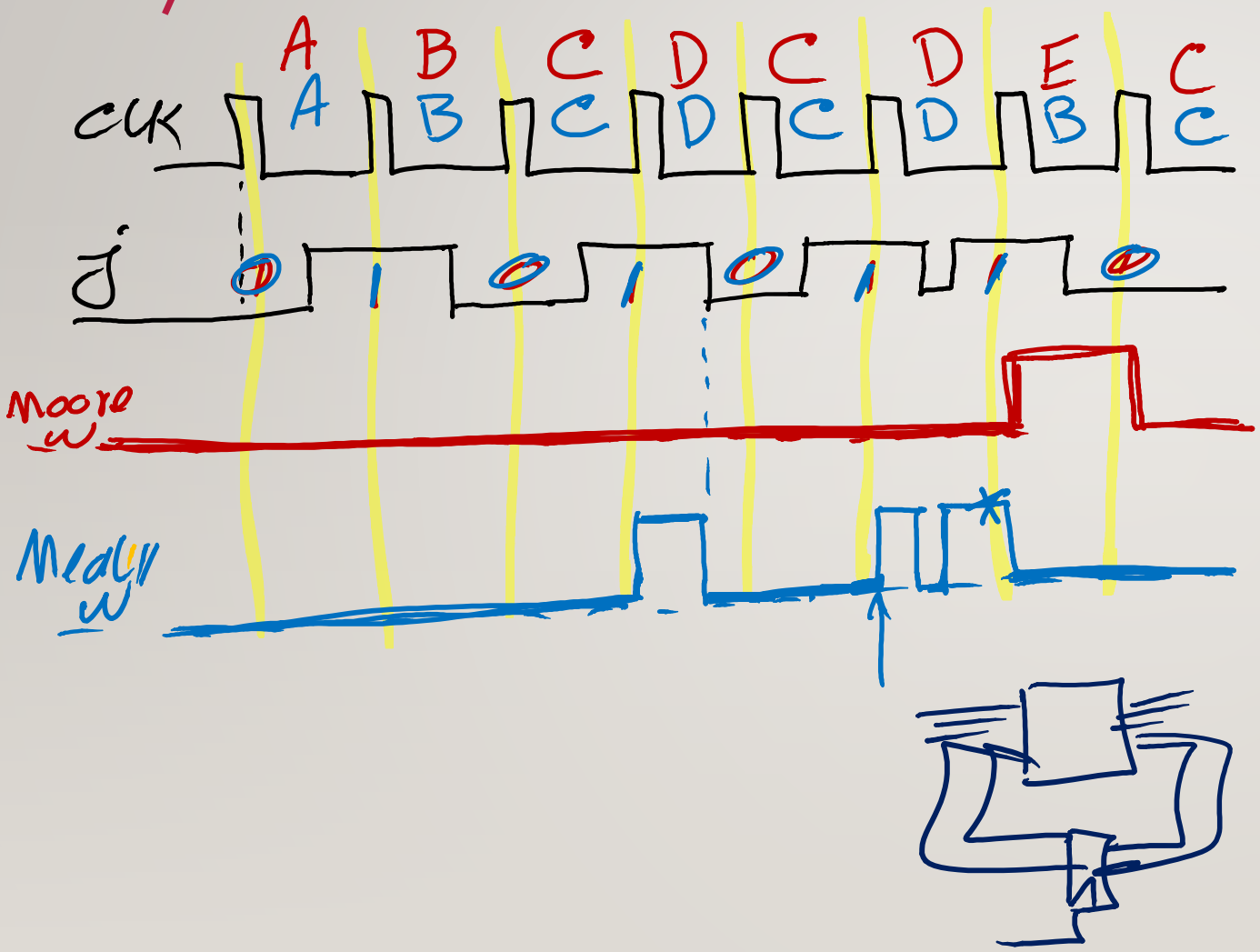
$D_1 = (J \oplus v_1 \oplus v_0) \cdot J\bar{v}_1$

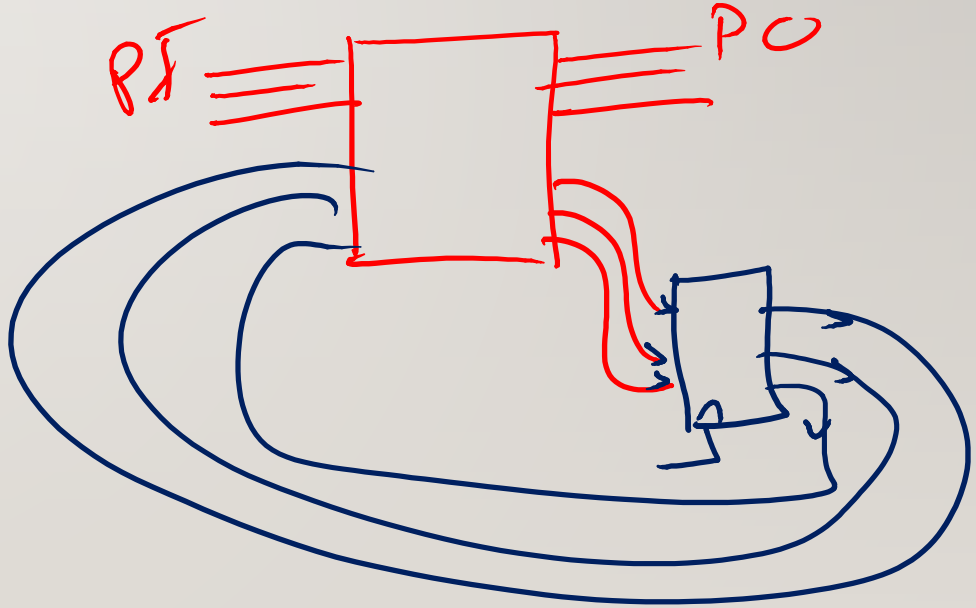
$D_0 = (v_1 \oplus v_0) + J\bar{v}_1$

$w = J v_1 \bar{v}_0$



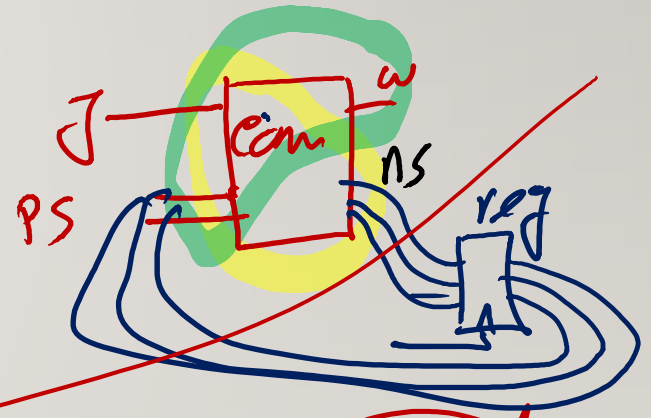
7

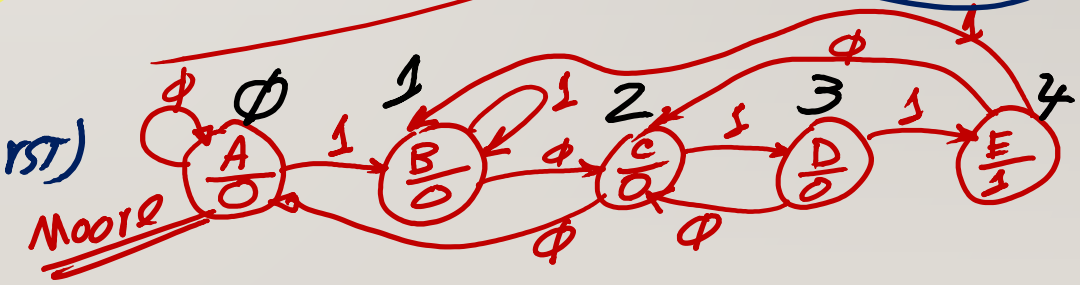




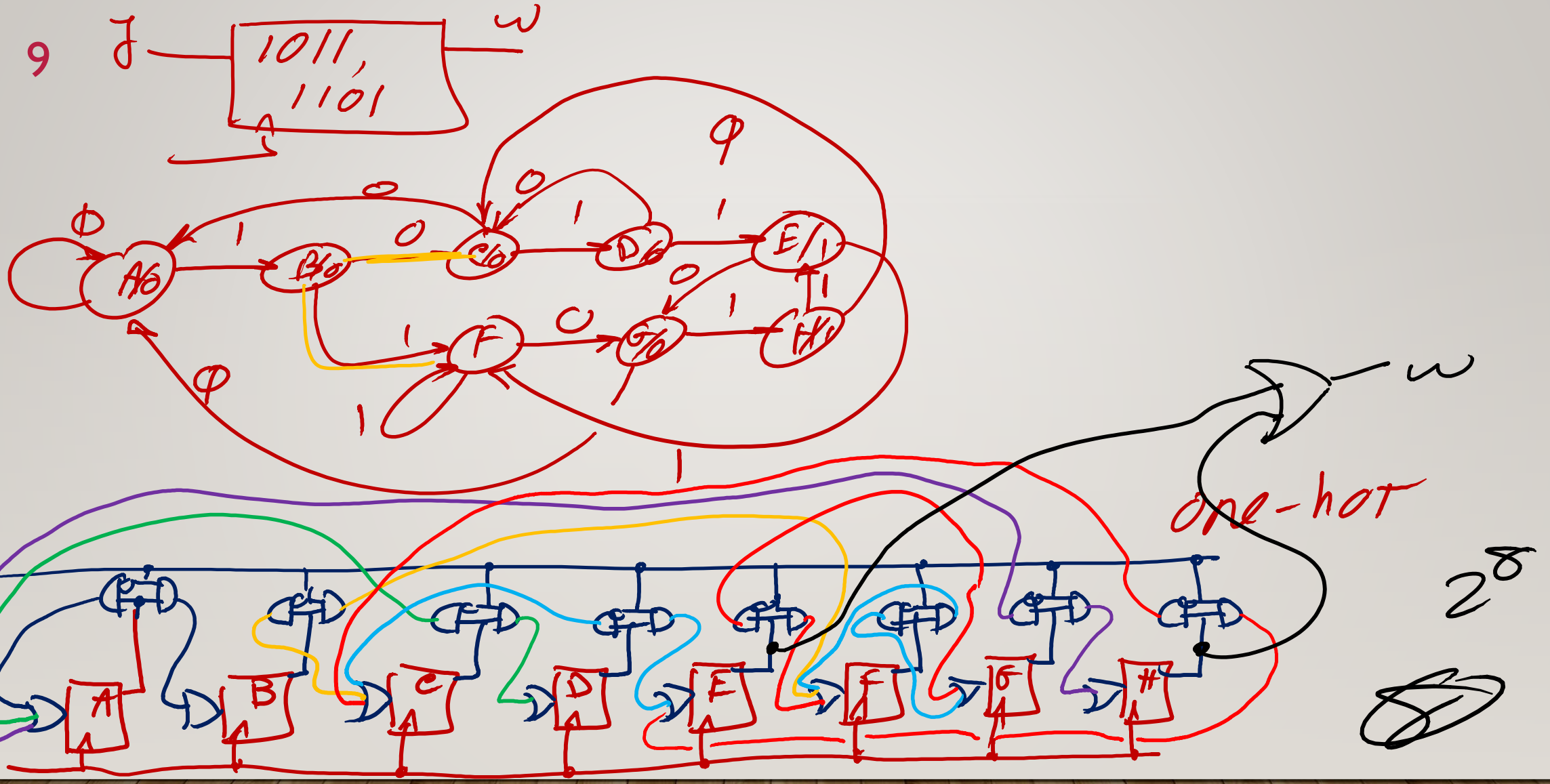
8

```
module moore101(input clk, rst, j, output w);
    reg [2:0] ps, ns;
    assign w = (ps==3'd4)?1'b1:1'b0;
    always @(j,ps) begin
        ns = 3'd0;
        case (ps)
        3'd0: ns = j?3'd1:3'd0;
        3'd1: ns = j?3'd1:3'd2;
        3'd2: if(j) ns=3'd3; else ns=3'd0;
        3'd3: ns = j?3'd4:3'd2;
        3'd4: ns = j?3'd1:3'd2;
        default: ns=3'd0;
        endcase
    end
    always @(posedge clk, posedge rst)
        if(rst) ps<=3'd0; else ps<=ns;
endmodule
```

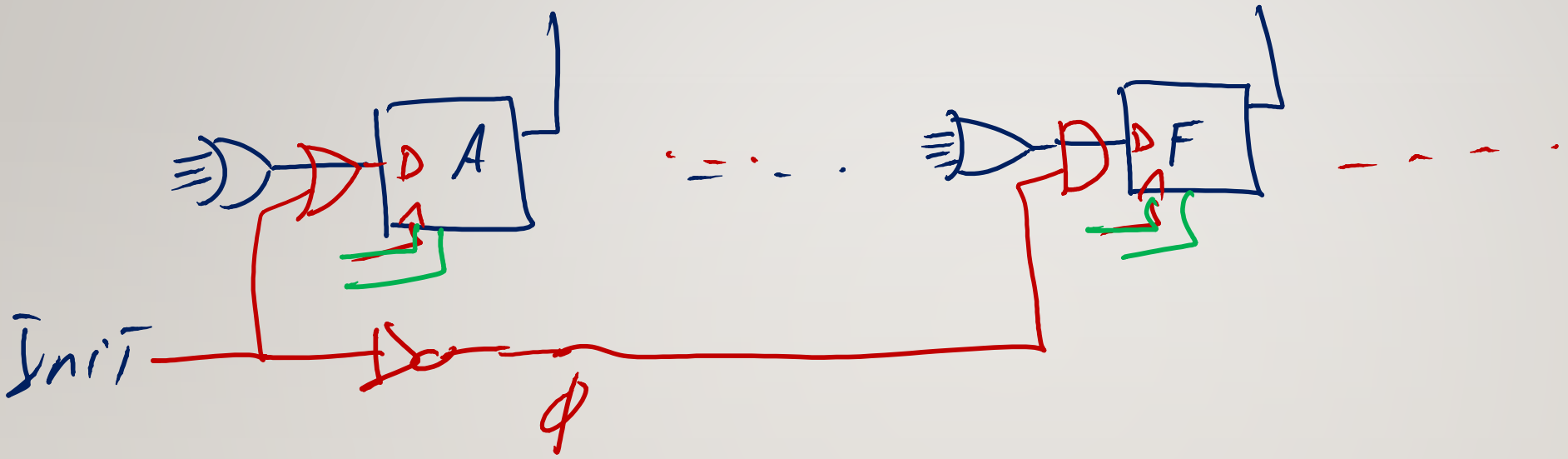




9



10



Initialization to:

1000000000 :A
0100000000 :B
0010000000 :C
0001000000 :D