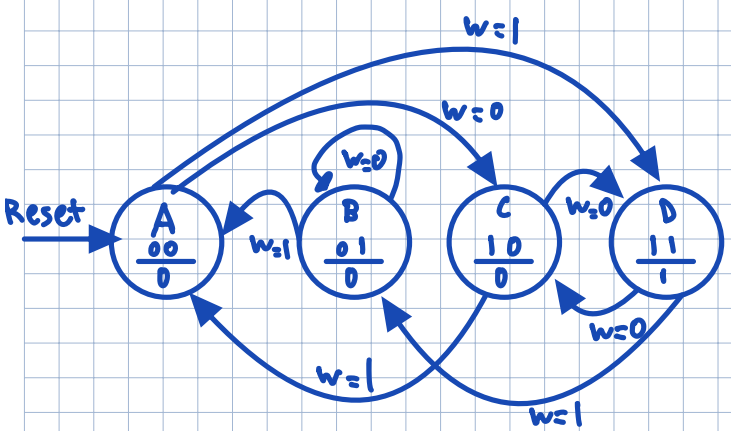
Synchronous Sequential Circuits.
HW Problems: 1, 2, 3(Mealy and Moore), 4(V-Moore), 5, 15, 23, 24.

***6.1** An FSM is defined by the state-assigned table in Figure P6.1. Derive a circuit that realizes this FSM using D flip-flops.



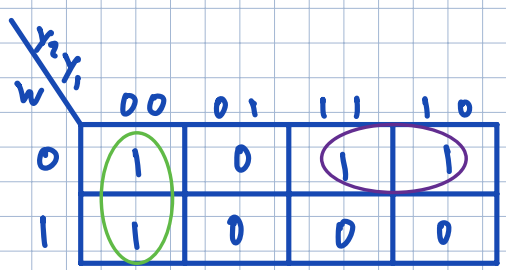
Present state y_2y_1	Next state $w = 0$ Y_2Y_1	Next state $w = 1$ Y_2Y_1	Output z
0 0	1 0	1 1	0
0 1	0 1	0 0	0
1 0	1 1	0 0	0
1 1	1 0	0 1	1

Figure P6.1

blems 6.1 and

Y_2 Y_1	PS	w	NS	Y_2 Y_1	D_2 D_1	z
0 0	A	0	C	1 0	1 0	0
0 0	A	1	D	1 1	1 1	0
0 1	B	0	B	0 1	0 1	0
0 1	B	1	A	0 0	0 0	0
1 0	C	0	D	1 1	1 1	0
1 0	C	1	A	0 0	0 0	0
1 1	D	0	C	1 0	1 0	1
1 1	D	1	B	0 1	0 1	1

D_2 SOP:



$$D_2 = \bar{Y}_2\bar{Y}_1 + Y_2\bar{w}$$

D_1 SOP:

w \ Y_2Y_1	00	01	11	10
0	0	1	0	1
1	1	0	1	0

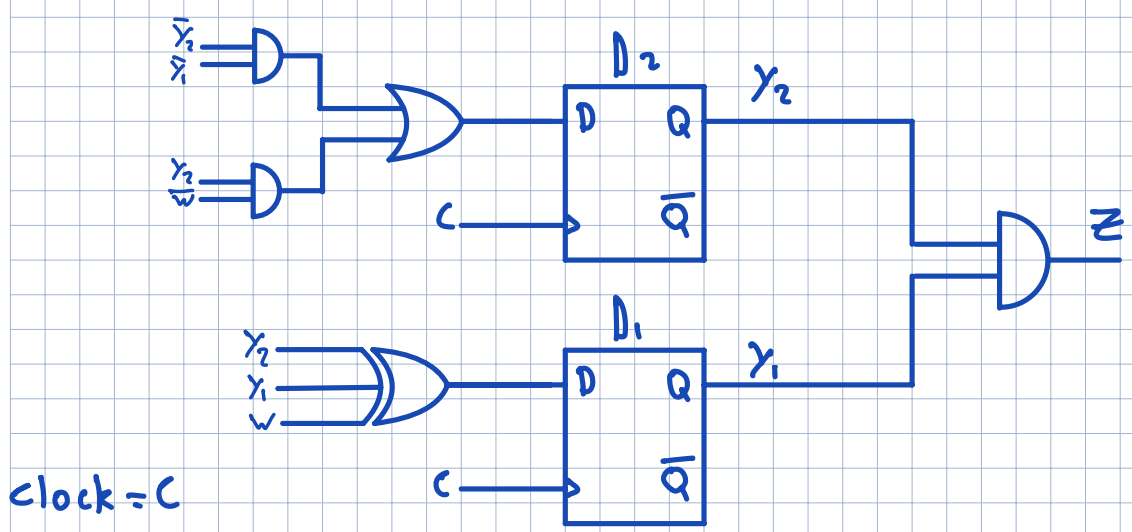
by observation
this is XOR gate

$D_1 = Y_2 \oplus Y_1 \oplus w$

z SOP:

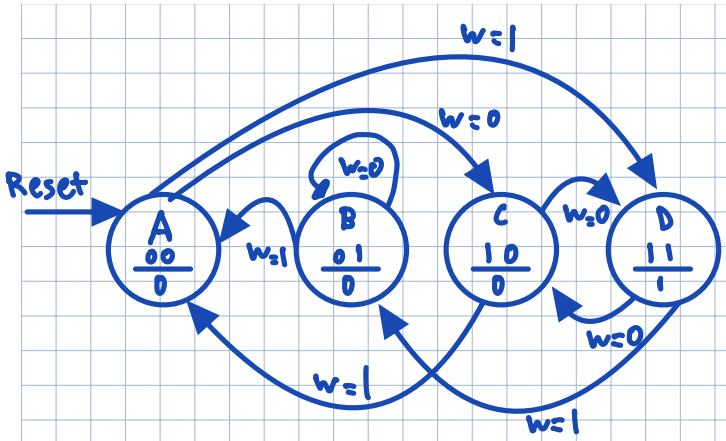
w \ Y_2Y_1	00	01	11	10
0	0	0	1	0
1	0	0	1	0

$z = Y_2 Y_1$



clock = C

***6.2** Derive a circuit that realizes the FSM defined by the state-assigned table in Figure P6.1 using JK flip-flops.



Present state y_2y_1	Next state $w = 0$ Y_2Y_1	Next state $w = 1$ Y_2Y_1	Output z
0 0	1 0	1 1	0
0 1	0 1	0 0	0
1 0	1 1	0 0	0
1 1	1 0	0 1	1

Figure P6.1

Problems 6.1 a

y_2	y_1	PS	w	NS	Y_2	Y_1	J_2	K_2	J_1	K_1	z
0	0	A	0	C	1	0	1	d	0	d	0
0	0	A	1	D	1	1	1	d	1	d	0
0	1	B	0	B	0	1	0	d	d	0	0
0	1	B	1	A	0	0	0	d	d	1	0
1	0	C	0	D	1	1	d	0	1	d	0
1	0	C	1	A	0	0	d	1	0	d	0
1	1	D	0	C	1	0	d	0	d	1	1
1	1	D	1	B	0	1	d	1	d	0	1

J_2 SOP:

y_2y_1 / w	00	01	11	10
0	1	0	d	d
1	1	0	d	d

$J_2 = \overline{Y_1}$

K_2 SOP:

y_2y_1 / w	00	01	11	10
0	d	d	0	0
1	d	d	1	0

$K_2 = w$

J_1 SOP:

w \ Y_2Y_1	00	01	11	10
0	0	d	d	1
1	1	d	d	0

$J_1 = Y_2\,\bar{w} + \bar{Y}_2 w$

$= Y_2 \oplus w$

k_1 SOP:

w \ Y_2Y_1	00	01	11	10
0	d	0	1	d
1	d	1	0	d

$k_1 = Y_2\,\bar{w} + \bar{Y}_2 w$

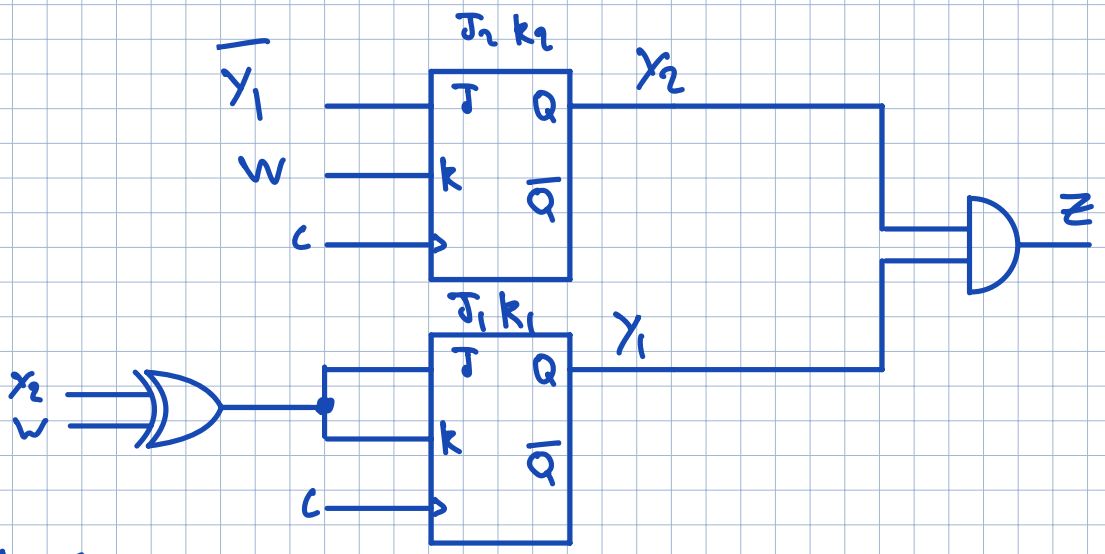
$= Y_2 \oplus w$

$= J_1$

z SOP:

w \ Y_2Y_1	00	01	11	10
0	0	0	1	0
1	0	0	1	0

$z = Y_2 Y_1$



clock = C

6.3 Derive the state diagram for an FSM that has an input w and an output z. The machine has to generate $z = 1$ when the previous four values of w were 1001 or 1111; otherwise, $z = 0$. Overlapping input patterns are allowed. An example of the desired behavior is

$$w : 0101111001100111111$$
$$z : 0000001001000100111$$

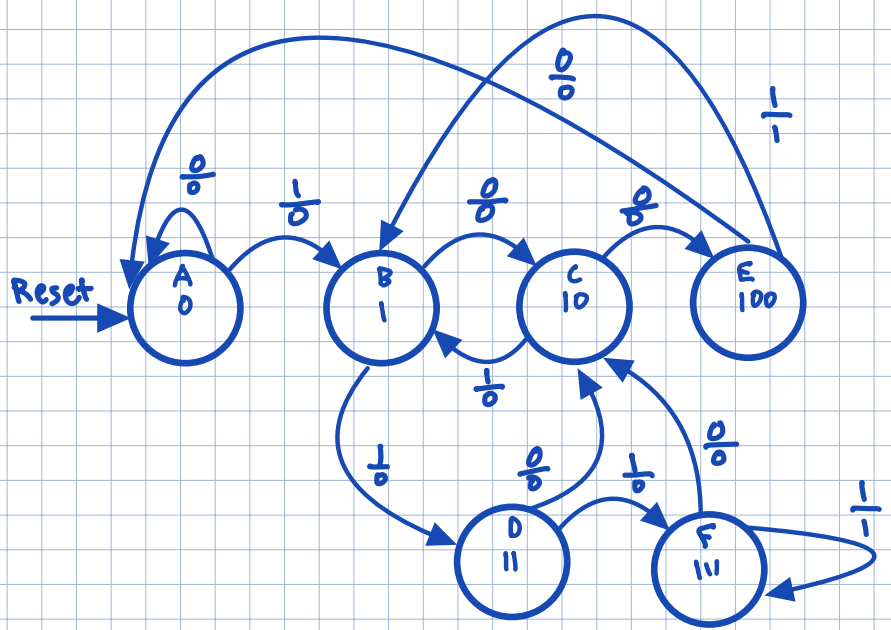
Derive using Mealy then Moore

Mealy:

Let

whenever input breaks pattern go back

PS	NS $w=0$	NS $w=1$	output $w=0$	output $w=1$
A: 0	A 0	B 1	0	0
B: 1	C 10	D 11	0	0
C: 10	E 100	B 1	0	0
D: 11	C 10	F 111	0	0
E: 100	A 0	B 1	0	1
F: 111	C 10	F 111	0	1

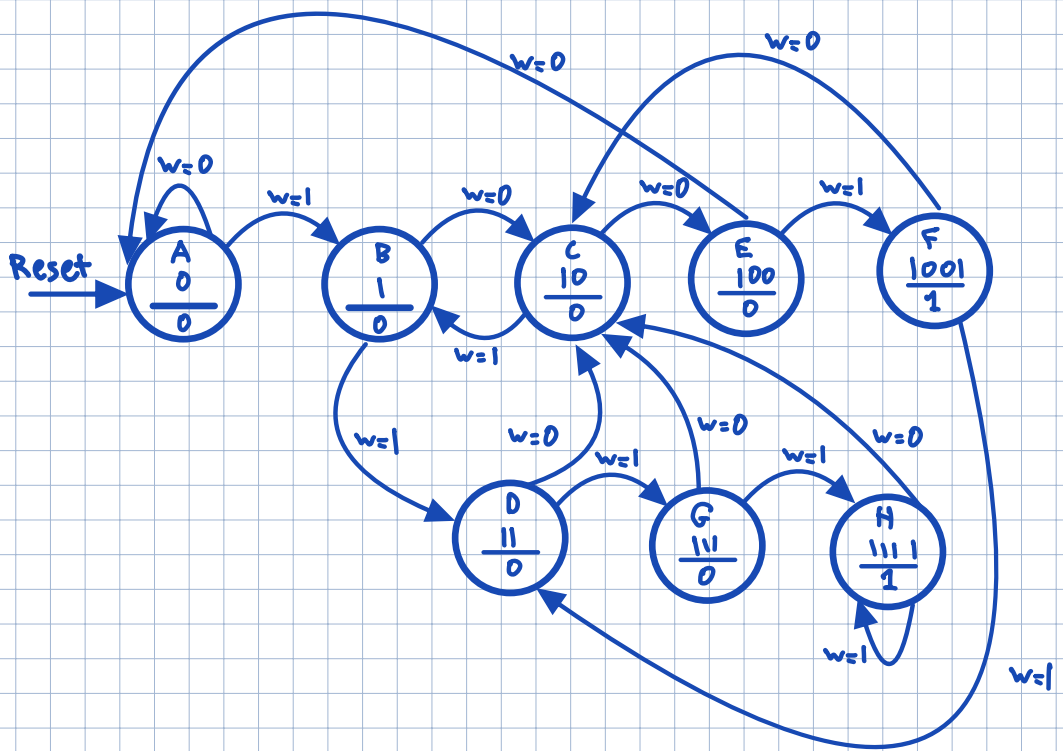


Moore:

Let

whenever input breaks pattern go back

PS	NS w=0	NS w=1	z
A: 0	A 0	B 1	0
B: 1	C 10	D 11	0
C: 10	E 100	B 1	0
D: 11	C 10	G 111	0
E: 100	A 0	F 1001	0
F: 1001	C 10	D 11	1
G: 111	C 10	H 1111	0
H: 1111	C 10	H 1111	1



6.4 Write Verilog code for the FSM described in Problem 6.3.

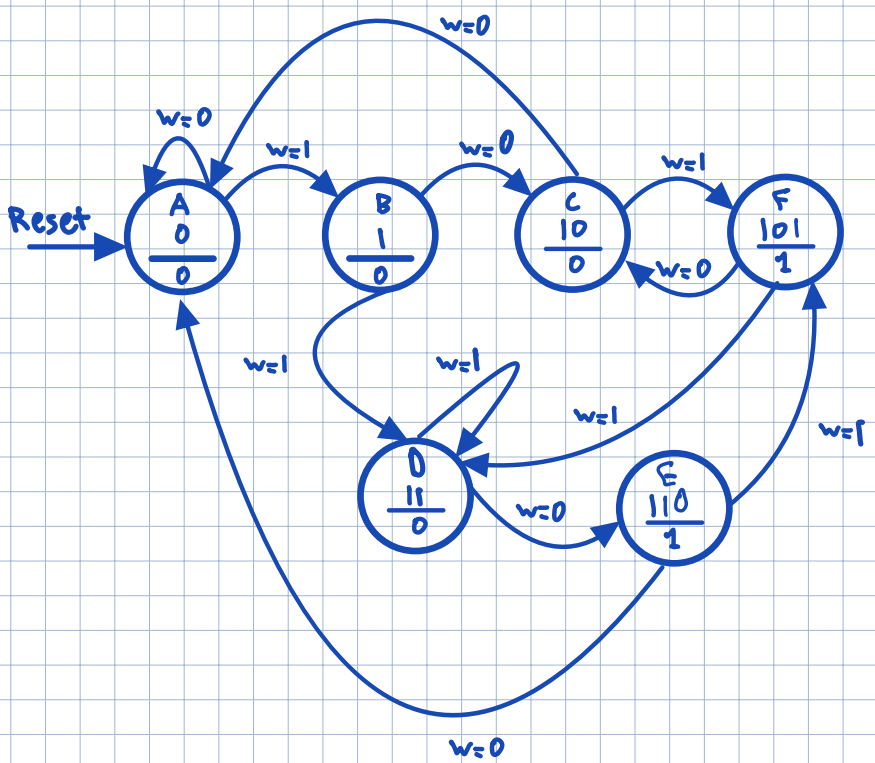
Write code for Moore only

```
module moore(clock, reset, w, z);
input clock, reset, w;
output reg z;

parameter A = 4'b0000;
parameter B = 4'b0001;
parameter C = 4'b0010;
parameter D = 4'b0011;
parameter E = 4'b0100;
parameter F = 4'b1001;
parameter G = 4'b0111;
parameter H = 4'b1111;

reg [3:0] y, Y;

always @(y or w)
   begin
      case (y)
         A: Y <= (w ? B : A);
         B: Y <= (w ? D : C);
         C: Y <= (w ? B : E);
         D: Y <= (w ? G : C);
         E: Y <= (w ? F : A);
         F: Y <= (w ? D : C);
         G: Y <= (w ? H : C);
         H: Y <= (w ? H : C);
         default: Y <= 4'bxxxx;
      endcase
   end
always @(negedge reset or posedge clock)
   begin
      if (!reset)
         y <= A;
      else
         y <= Y;

      z = (y == F) | (y == H);
   end
endmodule
```

***6.5** Derive a minimal state table for a single-input and single-output Moore-type FSM that produces an output of 1 if in the input sequence it detects either 110 or 101 patterns. Overlapping sequences should be detected.

whenever input breaks pattern go back

PS	NS w=0	NS w=1	z
A: 0	A 0	B 1	0
B: 1	C 10	D 11	0
C: 10	A 0	F 101	0
D: 11	E 110	D 11	0
E: 110	A 0	F 101	1
F: 101	C 10	D 11	1



***6.15** Show a state table for the state-assigned table in Figure P6.1, using A, B, C, D for the four rows in the table. Give a new state-assigned table using a one-hot encoding. For A use the code $y_4y_3y_2y_1 = 0001$. For states B, C, D use the codes 0010, 0100, and 1000, respectively. Synthesize a circuit using D flip-flops.

Present state	Next state		Output
	$w = 0$	$w = 1$	
			z
A	C	D	0
B	B	A	0
C	D	A	0
D	C	B	1

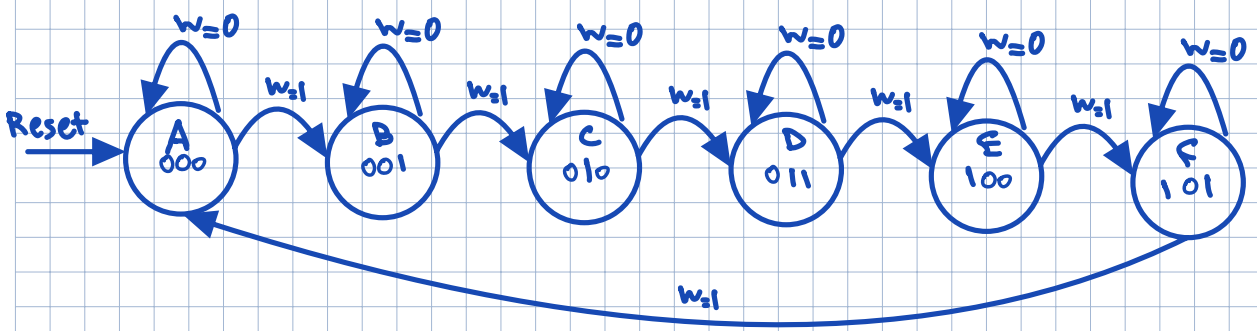
Present state	Next state		Output
	$w = 0$	$w = 1$	
y_2y_1	Y_2Y_1	Y_2Y_1	z
0 0	1 0	1 1	0
0 1	0 1	0 0	0
1 0	1 1	0 0	0
1 1	1 0	0 1	1

Figure P6.1

blems 6.1 and

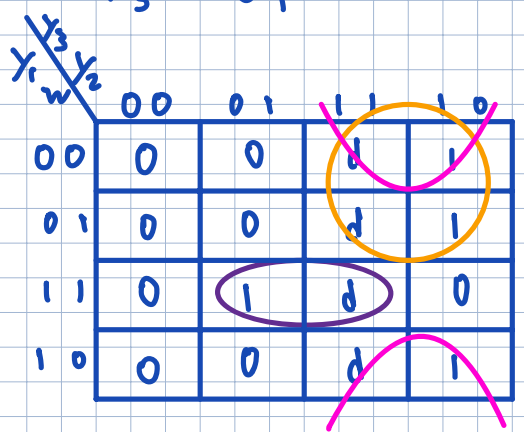
Present state	Next state		Output
	$w = 0$	$w = 1$	
$Y_4Y_3Y_2Y_1$	$Y_4Y_3Y_2Y_1$	$Y_4Y_3Y_2Y_1$	z
A 0001	0100	1000	0
B 0010	0010	0001	0
C 0100	1000	0001	0
D 1000	0100	0010	1

6.23 Design a modulo-6 counter, which counts in the sequence 0, 1, 2, 3, 4, 5, 0, 1, ... The counter counts the clock pulses if its enable input, w, is equal to 1. Use D flip-flops in your circuit.



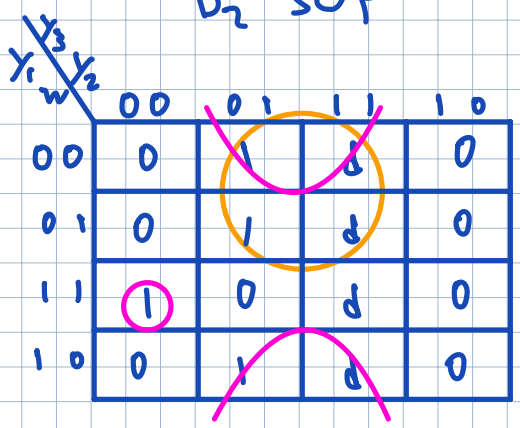
Y_3	Y_2	Y_1	PS	w	NS	Y_3	Y_2	Y_1	D_3	D_2	D_1
0	0	0	A	0	A	0	0	0	0	0	0
0	0	0	A	1	B	0	0	1	0	0	1
0	0	1	B	0	B	0	0	1	0	0	1
0	0	1	B	1	C	0	1	0	0	1	0
0	1	0	C	0	C	0	1	0	0	1	0
0	1	0	C	1	D	0	1	1	0	1	1
0	1	1	D	0	D	0	1	1	0	1	1
0	1	1	D	1	E	1	0	0	1	0	0
1	0	0	E	0	E	1	0	0	1	0	0
1	0	0	E	1	F	1	0	1	1	0	1
1	0	1	F	0	F	1	0	1	1	0	1
1	0	1	F	1	A	0	0	0	0	0	0

D_3 SOP



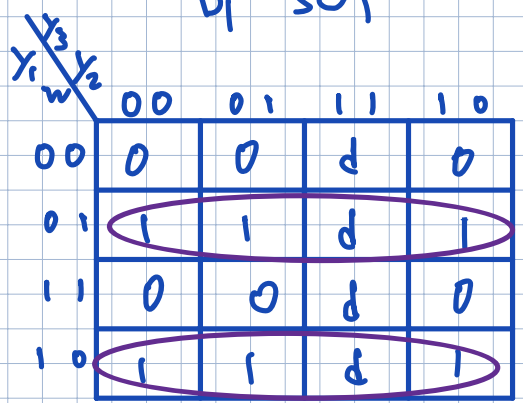
$$D_3 = Y_3 \bar{Y}_1 + Y_3 \bar{w} + Y_2 Y_1 w$$

D_2 SOP

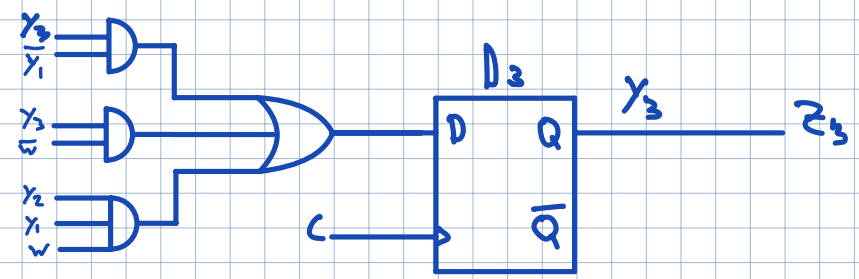


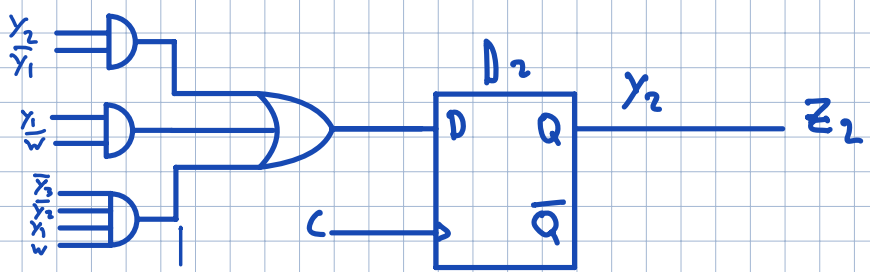
$$D_2 = Y_2\bar{Y_1} + Y_1\bar{w} + \bar{Y_3}\bar{Y_2}Y_1w$$

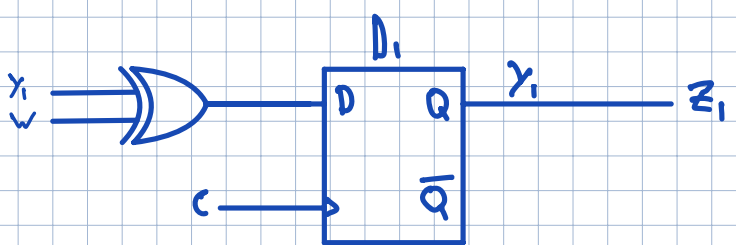
D_1 SOP



$$D_1 = \bar{Y_1}w + Y_1\bar{w}$$
$$= Y_1 \oplus w$$

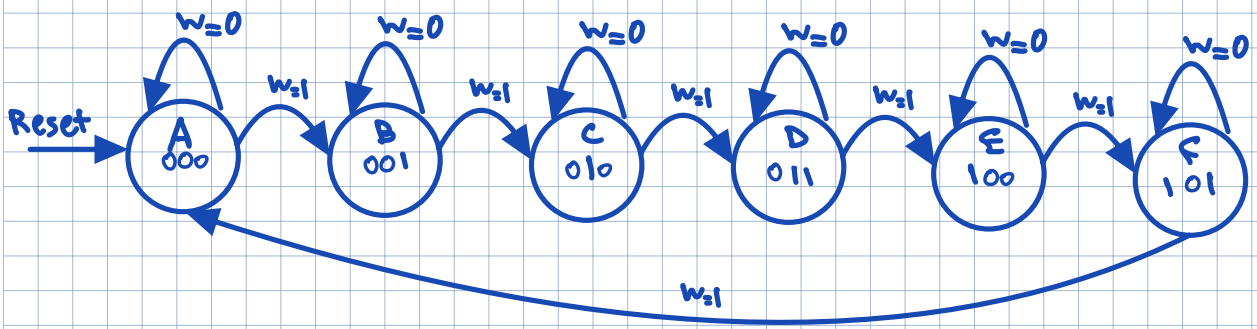

Y_3
$\overline{Y_1}$
Y_3
w
Y_2
Y_1
w
D_3
D
Q
$\overline{Q}$
C
Y_3
Z_3


Y_2
$\overline{Y_1}$
Y_1
w
$\overline{Y_2}$
Y_1
Y_2
w
D_2
D
Q
$\overline{Q}$
C
Y_2
Z_2


Y_1
w
D_1
D
Q
$\overline{Q}$
C
Y_1
Z_1

clock = C

6.24 Repeat Problem 6.23 using JK flip-flops.



Y_3	Y_2	Y_1	PS	w	NS	Y_3	Y_2	Y_1	J_3	K_3	J_2	K_2	J_1	K_1
0	0	0	A	0	A	0	0	0	0	d	0	d	0	d
0	0	0	A	1	B	0	0	1	0	d	0	d	1	d
0	0	1	B	0	B	0	0	1	0	d	0	d	d	0
0	0	1	B	1	C	0	1	0	0	d	1	d	d	1
0	1	0	C	0	C	0	1	0	0	d	d	0	0	d
0	1	0	C	1	D	0	1	1	0	d	d	0	1	d
0	1	1	D	0	D	0	1	1	0	d	d	0	d	0
0	1	1	D	1	E	1	0	0	1	d	d	1	d	1
1	0	0	E	0	E	1	0	0	d	0	0	d	0	d
1	0	0	E	1	F	1	0	1	d	0	0	d	1	d
1	0	1	F	0	F	1	0	1	d	0	0	d	d	0
1	0	1	F	1	A	0	0	0	d	1	0	d	d	1

J_3 SOP

$Y_1 w$ \ $Y_3 Y_2$	00	01	11	10
00	0	0	d	d
01	0	0	d	d
11	0	1	d	d
10	0	0	d	d

$J_3 = Y_2 Y_1 w$

K_3 SOP

$Y_1 w$ \ $Y_3 Y_2$	00	01	11	10
00	d	d	d	0
01	d	d	d	0
11	d	d	d	1
10	d	d	d	0

$K_3 = Y_1 w$

J_2 SOP

$Y_1 w$ \ $Y_3 Y_2$	00	01	11	10
00	0	d	d	0
01	0	d	d	0
11	1	d	d	0
10	0	d	d	0

$J_2 = \overline{Y_3}\, Y_1\, w$

K_2 SOP

$Y_1 w$ \ $Y_3 Y_2$	00	01	11	10
00	d	0	d	d
01	d	0	d	d
11	d	1	d	d
10	d	0	d	d

$K_2 = Y_1\, w$

J_1 SOP

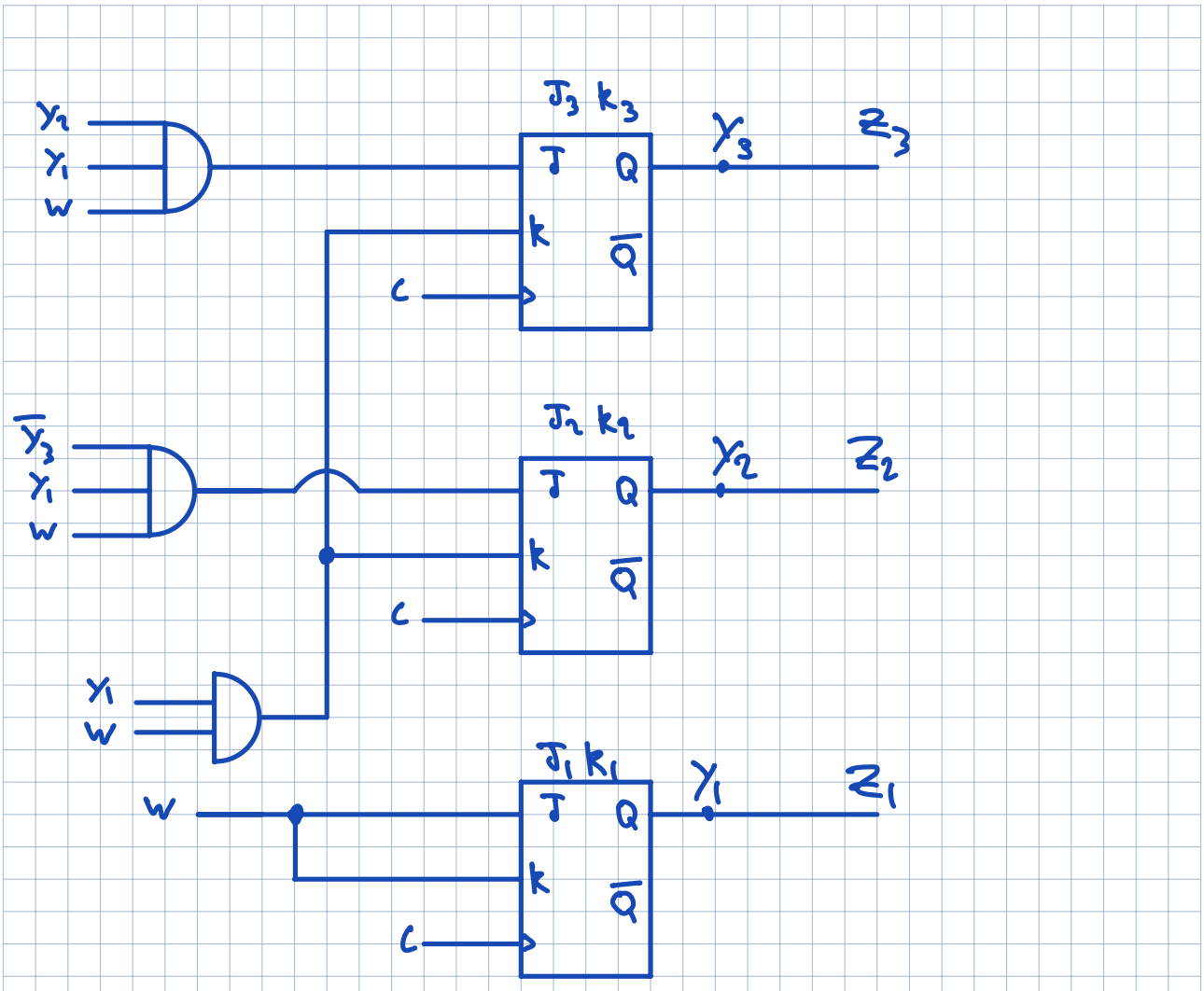
$Y_1 w$ \ $Y_3 Y_2$	00	01	11	10
00	0	0	d	0
01	1	1	d	1
11	d	d	d	d
10	d	d	d	d

$J_1 = w$

K_1 SOP

$Y_1 w$ \ $Y_3 Y_2$	00	01	11	10
00	d	d	d	d
01	d	d	d	d
11	1	1	d	1
10	0	0	d	0

$K_1 = w$


Y_2
Y_1
w
J_3 k_3
J
Q
K
$\overline{Q}$
C
Y_3
Z_3
$\overline{Y_3}$
Y_1
w
J_2 k_2
J
Q
K
$\overline{Q}$
C
Y_2
Z_2
Y_1
w
w
J_1 k_1
J
Q
K
$\overline{Q}$
C
Y_1
Z_1

clock = C