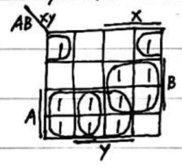
Assignment 3

1、 The state equations are

$$A(t+1) = J_A A' + K_A' A$$
$$= A'(Bx + B'y') + A(B'xy')'$$
$$= A'Bx + A'B'y' + AB + Ax' + Ay$$

Using K-map,



$$A(t+1) = Ax' + Ay + Bx + A'B'y'$$

$$B(t+1) = J_B B' + K_B' B$$
$$= A'B'x + (A + xy')'B$$
$$= A'B'x + A'Bx' + A'By \text{ (simplest form)}$$

$\cancel{= A'B}$

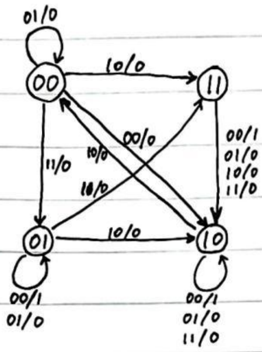
State table:

Present state A	Present state B	Input x	Input y	Next State A	Next State B	Output Z	FF Inputs J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	1	0	1	1	1	1
0	0	1	1	0	1	0	0	0	1	0
0	1	0	0	0	1	1	0	0	0	0
0	1	0	1	0	1	0	0	0	0	0
0	1	1	0	1	0	0	1	0	1	1
0	1	1	1	1	1	0	1	0	1	0
1	0	0	0	1	0	1	1	0	0	1
1	0	0	1	1	0	0	0	0	0	1
1	0	1	0	0	0	0	1	1	0	1
1	0	1	1	1	0	0	0	0	0	1
1	1	0	0	1	0	1	0	0	0	1
1	1	0	1	1	0	0	0	0	0	1
1	1	1	0	1	0	0	1	0	0	1
1	1	1	1	1	0	0	1	0	0	1

↓ FSM next page

Date

So we can get the FSM.



2. From the diagram, we can get input equations

$T_A = A + B$

$T_B = A' + B$

Then, state equation:

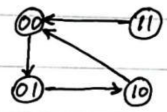
$$A(t+1) = T_A \oplus A = T_A'A + T_A A'$$
$$= AA'B' + AA' + A'B$$
$$= 0 + 0 + A'B$$
$$= A'B$$

$$B(t+1) = T_B \oplus B = T_B'B + T_B B'$$
$$= AB'B + A'B' + BB'$$
$$= 0 + A'B' + 0$$
$$= A'B'$$

Since there aren't any input, so ~~thx~~ truth table only has 4 lines. (next page)

Present State A	Present State B	Next State A	Next State B
0	0	0	1
0	1	1	0
1	0	0	0
1	1	0	0

At last, we can draw FSM.



3. (a) Input equation: $J_1 = X$, $K_1 = \cancel{XQ_2 + X'Q_2}\ (XQ_2')' = X' + Q_2$

$J_2 = X$, $K_2 = (XQ_1)' = X' + Q_1'$

State equation: $Q_1(t+1) = J_1Q_1' + K_1'Q_1$

$= XQ_1' + XQ_2'Q_1 = X(Q_1' + Q_2')$

$Q_2(t+1) = J_2Q_2' + K_2'Q_2$

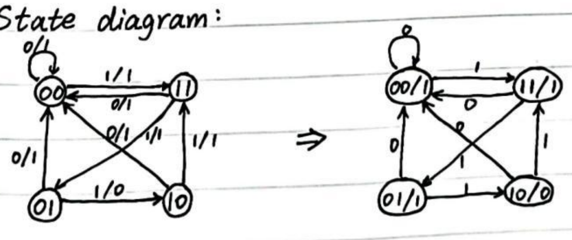
$= XQ_2' + XQ_1Q_2$

$= X(Q_1 + Q_2')$

State table:

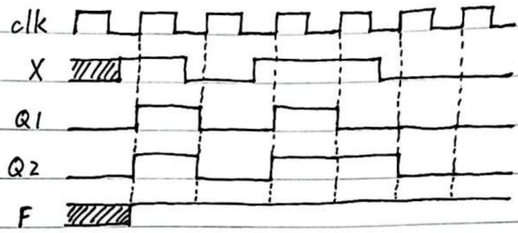
Present State Q_1	Present State Q_2	Input X	Next State Q_1	Next State Q_2	Output F
0	0	0	0	0	1
0	0	1	1	1	1
0	1	0	0	0	1
0	1	1	1	0	0
1	0	0	0	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	0	1	1

State diagram:



No.
Date

(b) Moore Machine. Because output only depend on state, output doesn't depend on input.

(c)


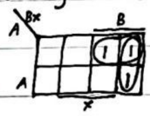
4. We first get state table.

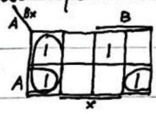
Present State A	Present State B	Input x	Next State A	Next State B	TFF Input T_A	TFF Input T_B
0	0	0	0	1	0	1
0	0	1	0	0	0	0
0	1	0	1	1	1	0
0	1	1	1	0	1	1
1	0	0	1	1	0	1
1	0	1	1	0	0	0
1	1	0	0	0	1	1
1	1	1	1	1	0	0

TFF:

Q(t)	Q(t+1)	T
0	0	0
0	1	1
1	0	1
1	1	0

Using K-map get input equation:



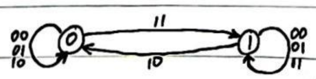


$T_A = A'B + Bx'$

$T_B = B'x' + Ax' + A'Bx$

5. When positive edge reach, the state change.

Suppose input is {En, D}, we can get state diagram.



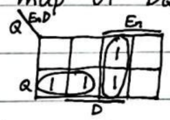
Then list state table.

Present State Q	Input E_n	Input D	Next State Q	DFF Input D_Q
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

DFF:

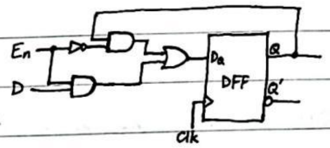
Q(t)	Q(t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

K-map of D_Q



$D_Q = QE_n' + DE_n$

At last, the design is as follow.



6. (a)

A	B	Q(t+1)	Operation
0	0	0	Reset
0	1	Q(t)	No change
1	0	Q'(t)	Complement
1	1	1	Set

(b) Next page

(c) When $Q(t)=0$, $Q(t+1)=0$, AB can be 00, 01, that is 0X.

When $Q(t)=0$, $Q(t+1)=1$, AB can be 10, 11, that is 1X.

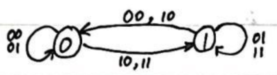
When $Q(t)=1$, $Q(t+1)=0$, AB = 00 or 10, that is X0.

When $Q(t)=1$, $Q(t+1)=1$, AB = 01 or 11, that is X1.

So ~~exi~~ excitation table:

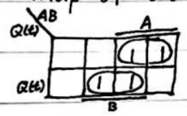
Q(t)	Q(t+1)	A	B
0	0	0	X
0	1	1	X
1	0	X	0
1	1	X	1

(b) Draw state diagram and get state table.



Present State $Q(t)$	Input A	Input B	Next State $Q(t+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

K-map of $Q(t+1)$:

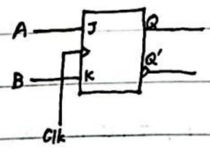


$Q(t+1) = AQ'(t) + BQ(t)$

(c) previous page

(d) From (c), we found the excitation table is the same as JKFF's, so this FF can be converted to JKFF.

The block diagram:



7. Found that h, d are the same, b, e are the same, we reduce them.

Present State	Next State x=0	Next State x=1	Output x=0	Output x=1
a	f	b	0	0
b	d	c	0	0
c	f	b	0	0
d	g	a	1	0
f	f	b	1	1
g	g	d	0	1

Then reduce a, c.

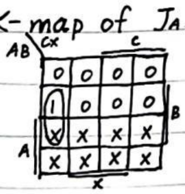
Present State	Next State x=0	Next State x=1	Output x=0	Output x=1
a	f	b	0	0
b	d	a	0	0
d	g	a	1	0
f	f	b	1	1
g	g	d	0	1

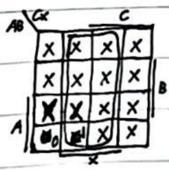
Code the state: a: 000, b: 001, d: 010, f: 011, g: 100

~~Present State~~ We need to use 3 JKFF.

Present State A	B	C	Input x	Next State A	B	C	Output y	JKFF Inputs J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	1	0	0	X	1	X	1	X
0	0	0	1	0	0	1	0	0	X	0	X	1	X
0	0	1	0	0	1	0	0	0	X	1	X	X	0
0	0	1	1	0	0	0	0	0	X	0	X	X	0
0	1	0	0	1	0	0	1	1	X	X	0	0	X
0	1	0	1	0	0	0	0	0	X	X	0	0	X
0	1	1	0	0	1	1	1	0	X	X	1	X	1
0	1	1	1	0	0	1	1	0	X	X	0	X	1
1	0	0	0	1	0	0	0	X	0	0	X	0	X
1	0	0	1	0	1	0	1	X	1	1	X	0	X
1	0	1	0	X	X	X	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X	X	X	X

K-map of J_A, K_A





$J_A = BC'x'$

$K_A = x$

K-map of J_B、K_B:

1	0	0	1
X	X	X	X
X	X	X	X
0	1	X	X

X	X	X	X
0	0	0	1
X	X	X	X
X	X	X	X

$J_B = A'x'$

$K_B = Cx'$

K-map of J_C、K_C:

1	1	X	X
0	0	X	X
X	X	X	X
0	0	X	X

X	X	0	0
X	X	1	1
X	X	X	X
X	X	X	X

$J_C = A'B'$

$K_C = B$

K-map of output:

0	0	0	0
1	0	1	1
X	X	X	X
0	1	X	X

$y = BC + Bx' + Ax$

8、 Suppose state A: detect nothing, B: have 1, C: have 11, D: have 110, E: have 1101

And output of A to E is 0, 0, 0, 0, 1

Code the state: A: 000, B: 001, C: 010, D: 011, E: 100

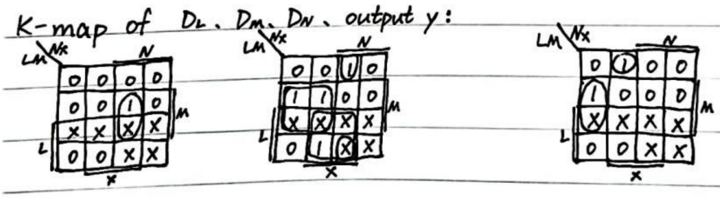
We need to use 3 DFF (Suppose is L、M、N)

Next Page

The state table is as below

Present State L	M	N	Input x	Next State L(D_L)	M(D_M)	N(D_N)	Output y
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0
0	0	1	1	0	1	0	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	0	0
0	1	1	0	0	0	0	0
0	1	1	1	1	0	0	0
1	0	0	0	0	0	0	1
1	0	0	1	0	1	0	1
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

K-map of D_L, D_M, D_N, output y:



$D_L = MNx$ $D_M = MN' + Lx + M'Nx$ $D_N = L'M'N'x + MN'x'$

And without K-map, we can easily see $y = L$