* State Reduction and Assignment :-

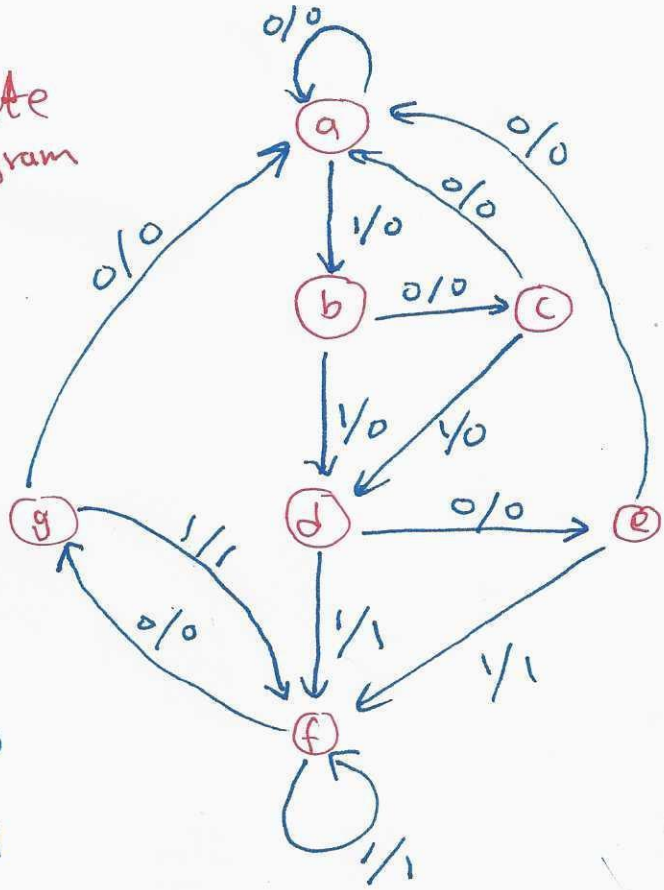
- Design starts with state table or state diagram.

- Reduction on the number of flip-flops and the number of gates

- Two states are said to be equivalent :-
  - If they have the same (1) next state and (2) same output.

Example :-

state diagram



state table :-

Present state	Next state		output	
	x=0	x=1	x=0	x=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

same output
same next state

e is equivalent to g

So we draw the state table and replace g by e

Present state	Next state		output	
	x=0	x=1	x=0	x=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

same output
same next state
∴ d is equivalent to f
d = f

So we delete f and replace each f in the table by d

Present state	Next state		output	
	x=0	x=1	x=0	x=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1

No equivalent states

* Implication Chart

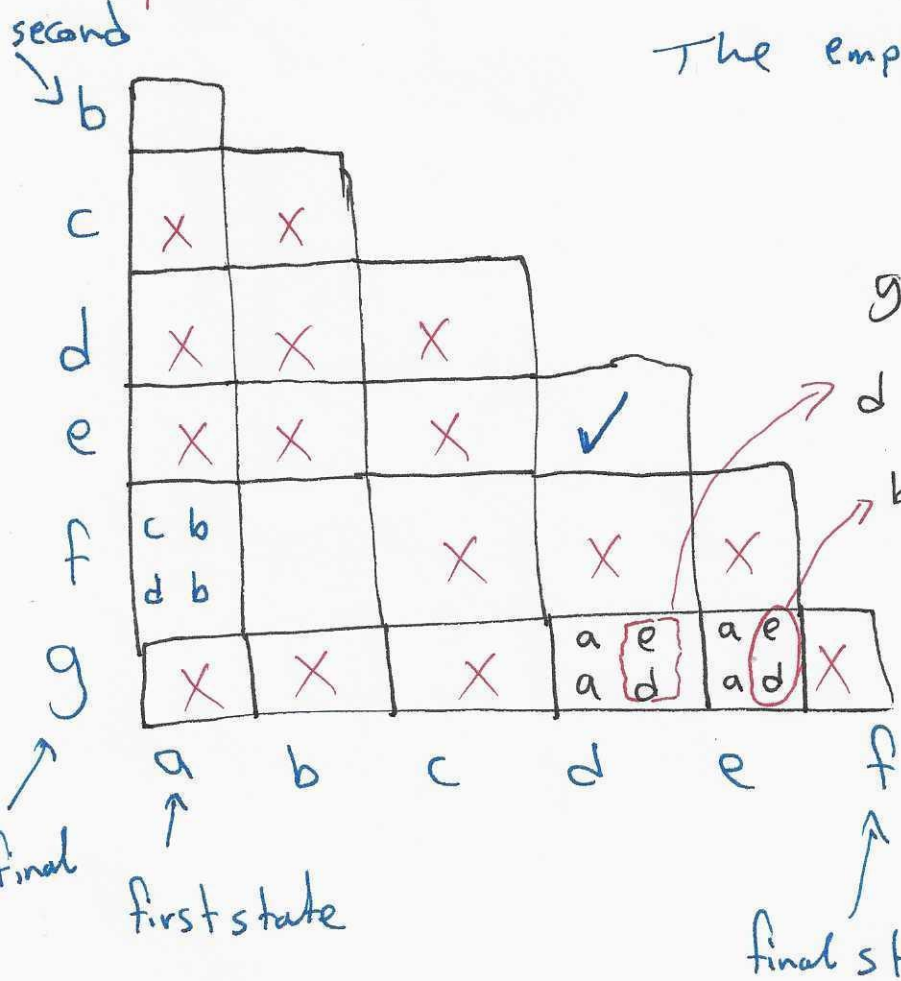
- To check possible equivalent states in state table with large number of states.

① - draw the implication chart
- place [X] in any square of a pair of states whose outputs are not equivalent.
- place [✓] for equivalent states (same outputs, same next state)

state table

present state	Next state x=0	Next state x=1	output x=0	output x=1
a	d	b	0	0
	e	a	0	0
b				
c	g	f	0	1
d	a	d	1	0
e	a	d	1	0
f	c	b	0	0
g	a	e	1	0

implication Chart



The empty squares ∴ Two states have same outputs but different next states

g: next states are a e

d: next states are a d

but e = d so g = d

e = d so g = e

e = d = g

(2) For empty squares, enter the implied states

e = d = g

for a and f

	nex stat	
a	d	b
f	c	b

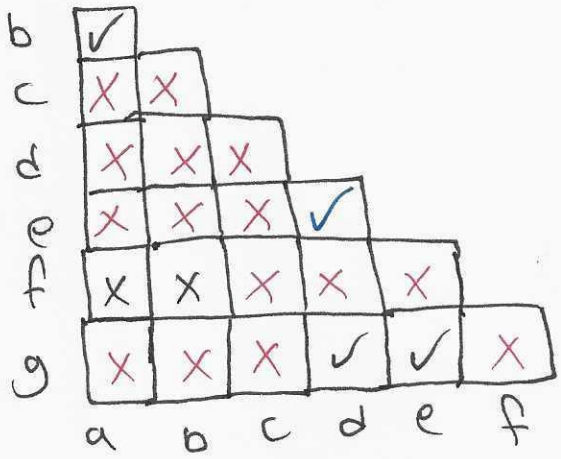
b = b but c is not equivalent to d so a ≠ f

b e a so b ≠ f

a → d, b
b → e, a

↳ equivalent so $a = b$

(3) place ✓ for equivalent states and X for not equivalent states



(4) List equivalent states from squares with ✓

(a,b), (d,e), (d,g), (e,g)
↓ ↓ ↓ ↓
a d d d

(5) Combine pairs of states into large gro

(a,b), (d,e,g)

(6) The final states are the equivalent states and all remaining states in state table

(a,b) ⇒ a f ⇒ f

from f to 4 states

state table

Present state	next state x=0	next state x=1	output x=0	output x=1
a	d	a	0	0
c	d	f	0	1
d	a	d	1	0
f	c	a	0	0

* Design of sequential Logic

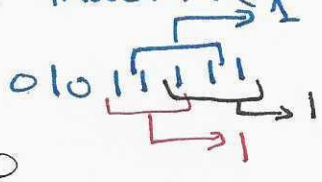
Procedures:-

1. From the word description, derive a state diagram of the circuit
2. Reduce the number of states if necessary
3. Assign binary values to the state
4. obtain the binary-coded state table
5. choose the type of flip-flops to be used
6. Derive the simplified flip-flop input equations and output equations
7. Draw the logic diagram

Example:- Design a sequential circuit that detects three or more consecutive one's?

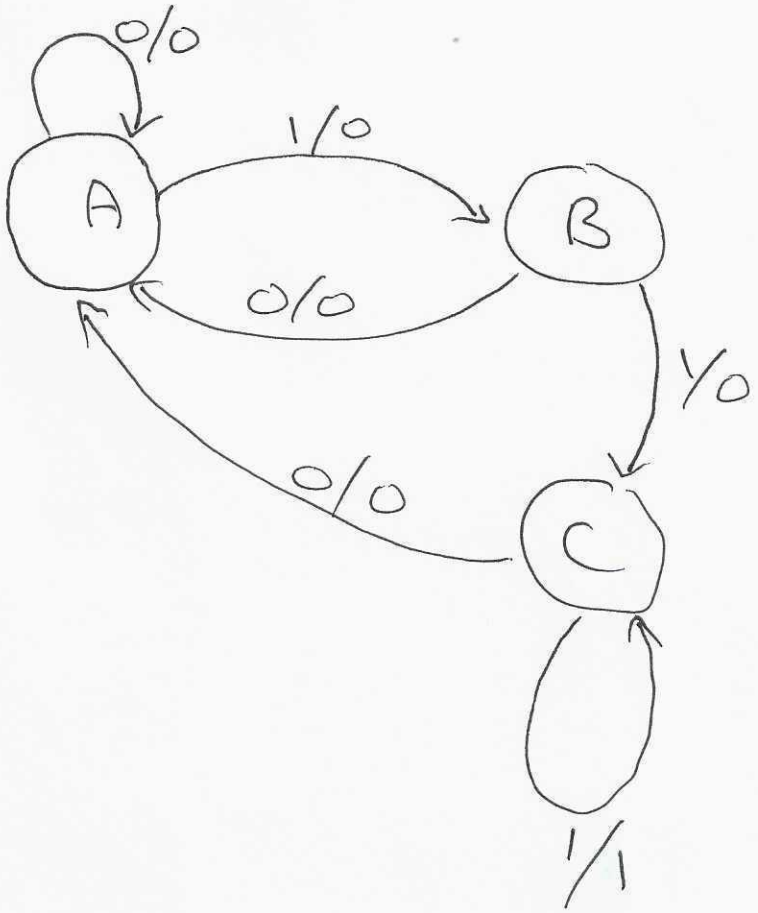
(1) Derive the state diagram

from the beginning we can design the same circuit using mealy machine or moore machine

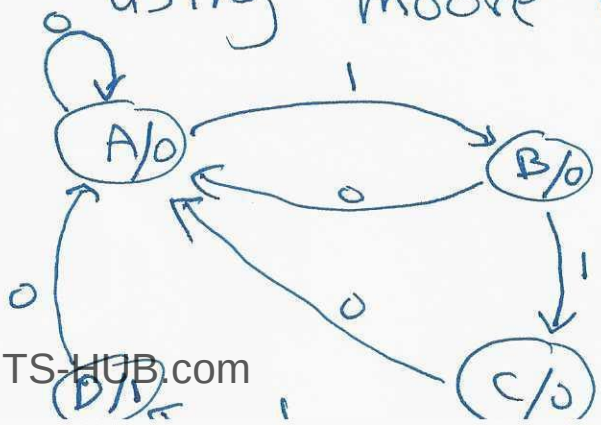


0 0 0 0 0 1 1 1 output

using mealy machine

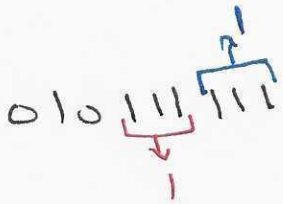


using moore machine

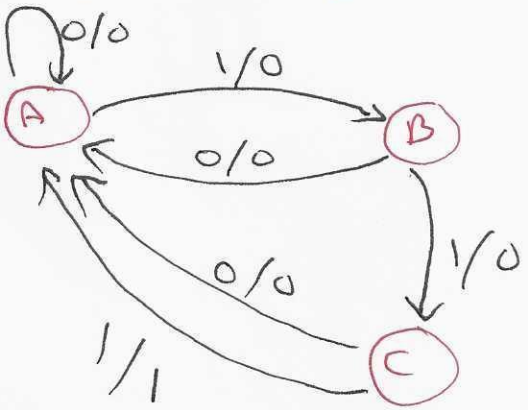


If the problem is to detect three consecutive one's

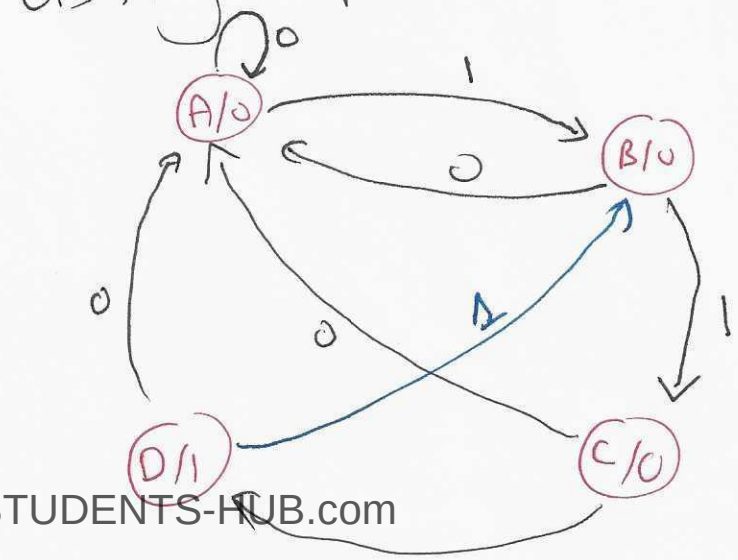
the word mave is not mentioned



Using mealy machine

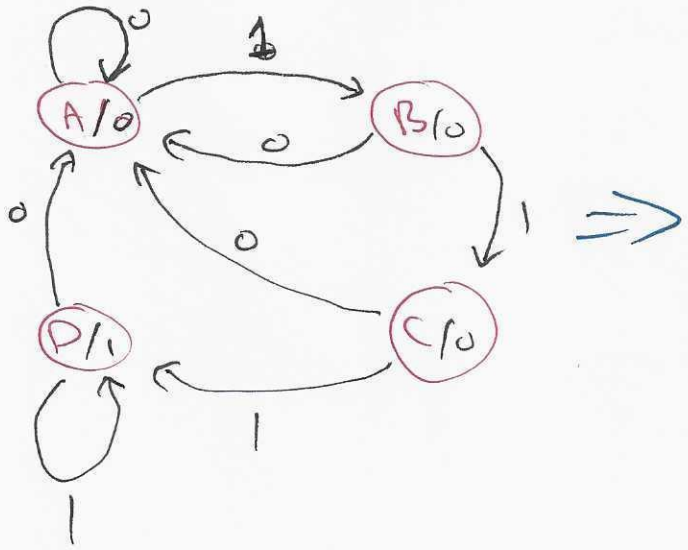


Using moore machine



Once we have the state diagram, we can generate the state table :-

① Derive the state diagram



⇒

Present state	Next state x=0	Next state x=1	output x=0	output x=1
A	A	B	0	0
B	A	C	0	0
C	A	D	0	0
D	A	D	1	1

② Reduce the number of states if necessary

∴ we don't have any equivalent states

③ Assign binary values to the state

we have 4 states so we can use 2-bit code

$\log_2(4) = 2$ # of bits

A 0 0

B 0 1

C 1 0

D 1 1

(4) obtain the binary-Coded state table

state table

Present state A(t)	B(t)	input x	Next state A(t+1)	B(t+1)	output y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	1	1	1

(5) choose the type of flip-flop

we will solve using 3 types of flip-flops

(A) D - flip-flops

$$\#\text{ of flip-flops} = \#\text{ of bits} = \log_2(\#\text{ of states})$$

$$= \log_2(4) = 2 \text{ flip flops}$$

we know that

Charactristics equation

$$Q(t+1) = D \Rightarrow D = Q(t+1)$$

Truth table

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

$\Rightarrow$

Excitation table

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

state table of D flip-flops

Present state A(t)	Present state B(t)	input x	D_A	D_B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

$D_A = A(t+1)$ from the previous table and excitation table

$D_B = B(t+1)$ from the previous table and excitation table

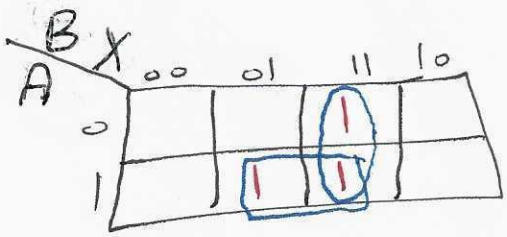
$D_A = \sum(3, 5, 7)$

$D_B = \sum(1, 5, 7)$

$Y = \sum(6, 7)$

(6) Derive the simplified flip-flop input equations and output equations

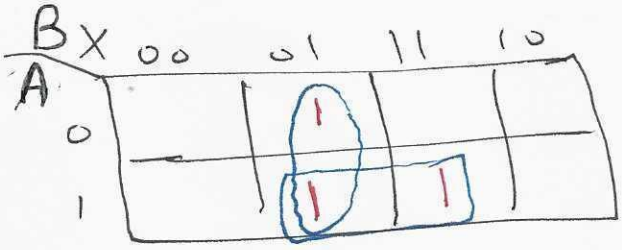
$D_A = \Sigma(3,5,7)$



$= BX + AX$

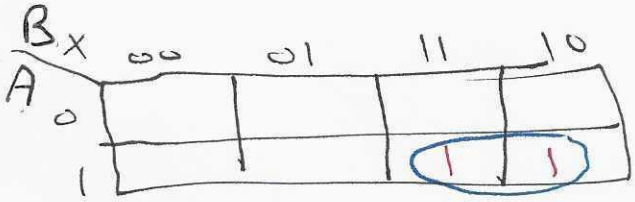
$= B(1)X + A(1)X$

$D_B = \Sigma(1,5,7)$



$= \bar{B}X + AX$

$Y = \Sigma(6,7)$

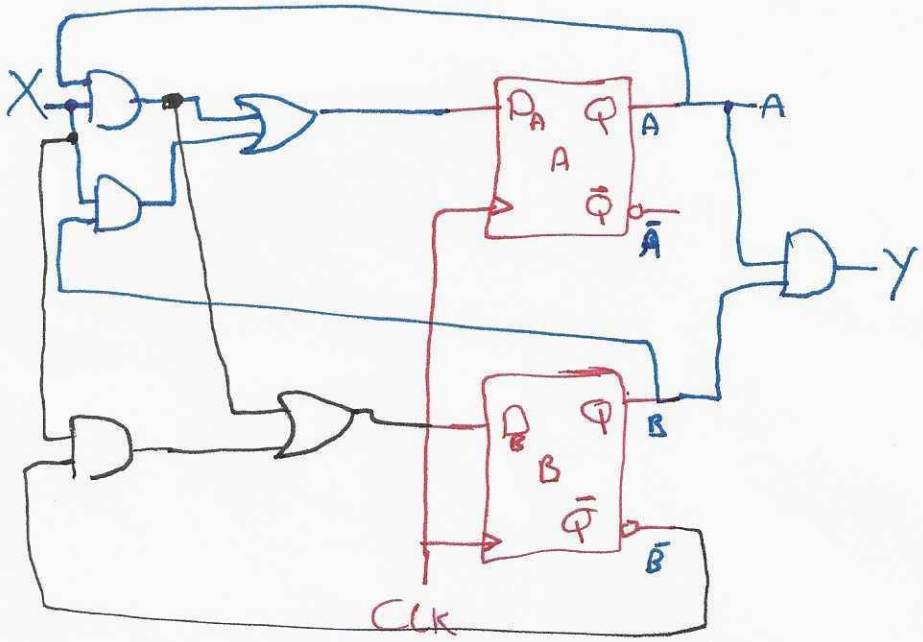


$= AB$

(7) Draw the logic diagram

$D_A = Ax + Bx$

$D_B = Ax + \bar{B}x$



$Y = AB$

(B) using JK flip-flops

we still need 2 flip-flops

we need to convert the state table on page 27

to JK state table using JK excitation table

JK characteristics table

J	K	Q(t+1)	
0	0	Q(t)	no change
0	1	0	Reset
1	0	1	set
1	1	$\bar{Q}(t)$	complement

⇒

JK excitation table

Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

If Q(t)=0 and Q(t+1)=0

It is either J=0 and K=0 (no change)

or J=0 and K=1 (Reset)

J=0 K=X

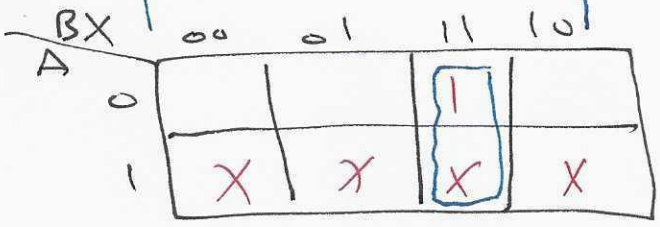
↳ 0 or 1

and so on.....

Present state A(t)	B(t)	input X	Next state A(t+1)	B(t+1)	Flip-flop inputs JA	KA	JB	KB	output Y
0	0	0	0	0	0	X	0	X	0
0	0	1	0	1	0	X	1	X	0
0	1	0	0	0	0	X	X	1	0
0	1	1	1	0	1	X	X	1	0
1	0	0	0	0	X	1	0	X	0
1	0	1	1	1	X	0	1	X	0
1	1	0	0	0	X	1	X	1	1
1	1	1	1	1	X	0	X	0	1

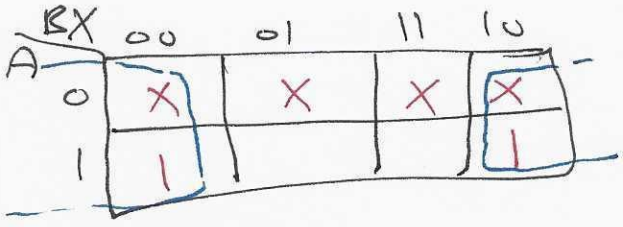
$$JA = \sum_m(3) + \sum_d(4,5,6,7)$$

$$= BX$$



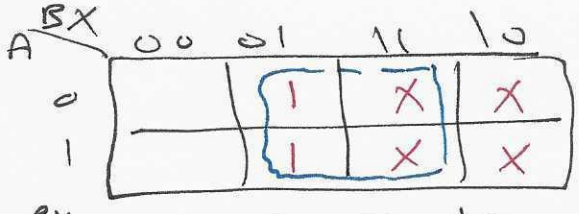
$$KA = \sum_m(4,6) + \sum_d(0,1,2,3)$$

$$= \bar{X}$$



$$JB = \sum_m(1,5) + \sum_d(2,3,6,7)$$

$$= X$$



$$KB = \sum_m(2,3,6,7) + \sum_d(0,1,4,5)$$

$J_A = BX$

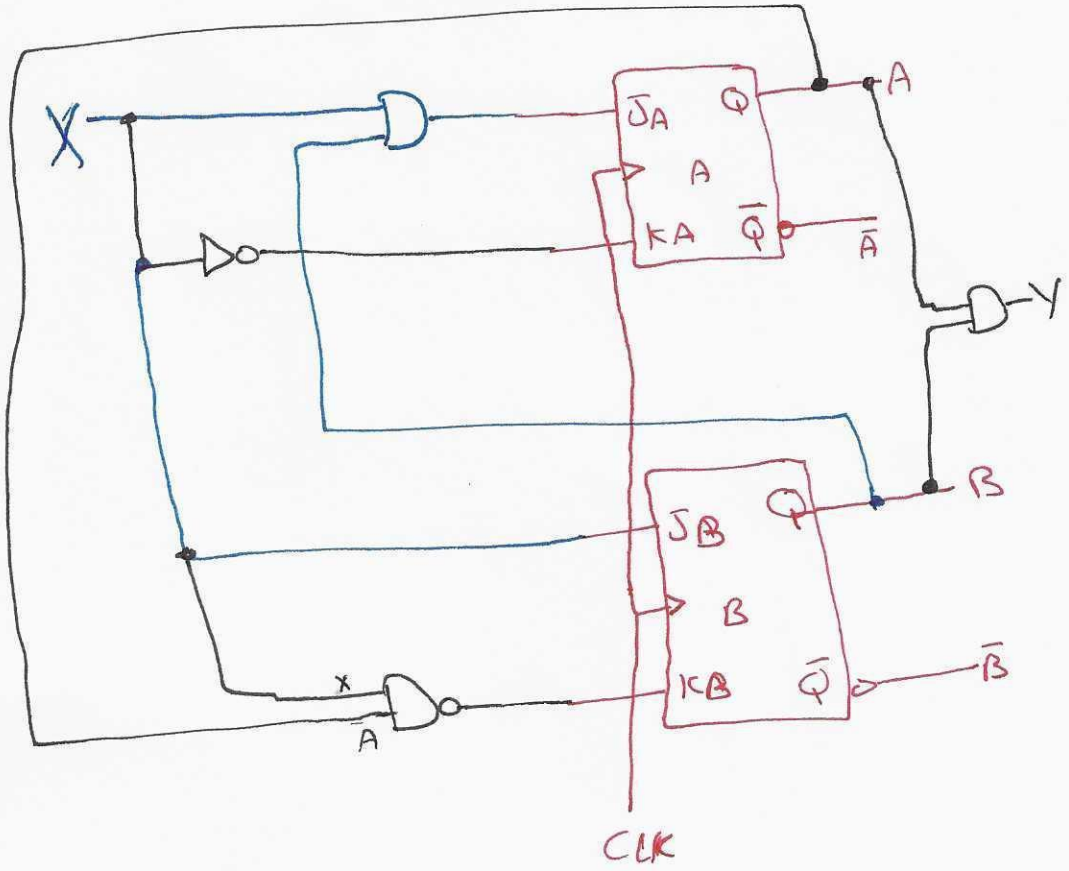
$K_A = \bar{X}$

$J_B = X$

$K_B = \bar{A} + \bar{X}$

$= \overline{(A \cdot X)}$

$Y = AB$



(c) Using T flip-flops

we need to convert the state table on page 27 to T-state table using T excitation table

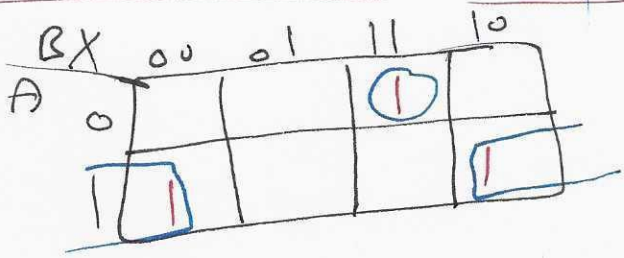
T- flip-flop Characteristics table

T	Q(t+1)	
0	Q(t)	no change
	$\bar{Q}(t)$	Complement

T- flip-flop excitation table

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

Present state A(t)	B(t)	input X	Next state A(t+1)	B(t+1)	flip flop inputs TA	TB	output Y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	0	0	1	0
0	1	1	1	0	1	1	0
1	0	0	0	0	1	0	0
1	0	1	1	1	0	1	0
1	1	0	0	0	1	1	1
1	1	1	1	1	0	0	1



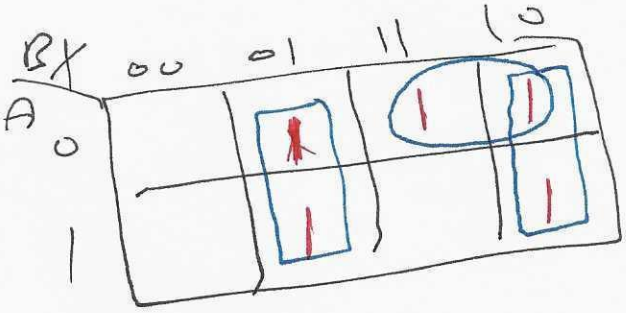
$$T_A = \Sigma(3,4,6)$$

$$= A\bar{X} + \bar{A}BX$$

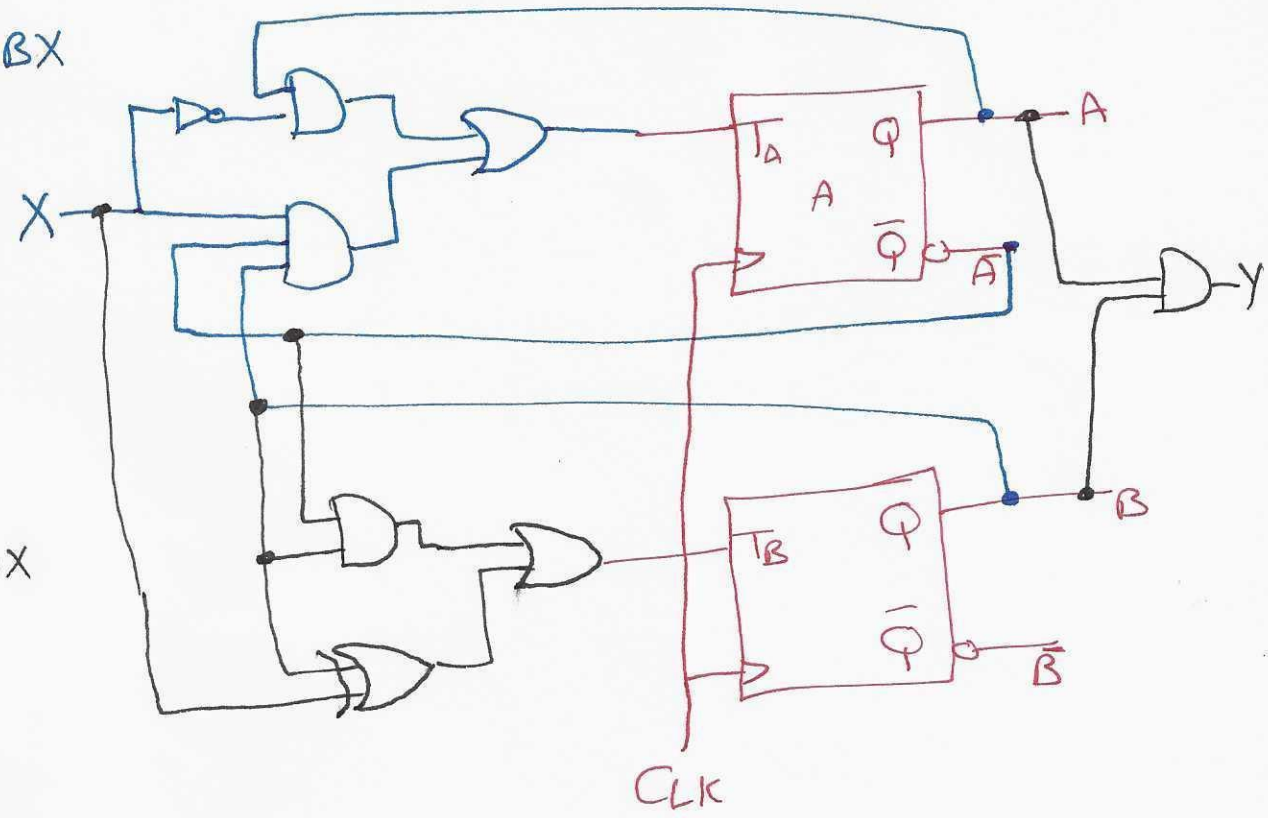
$$T_B = \Sigma(1,2,3,5,6)$$

$$= \bar{B}X + B\bar{X} + \bar{A}B$$

$$= B \oplus X + \bar{A}B$$



$T_A = A\bar{X} + \bar{A}BX$

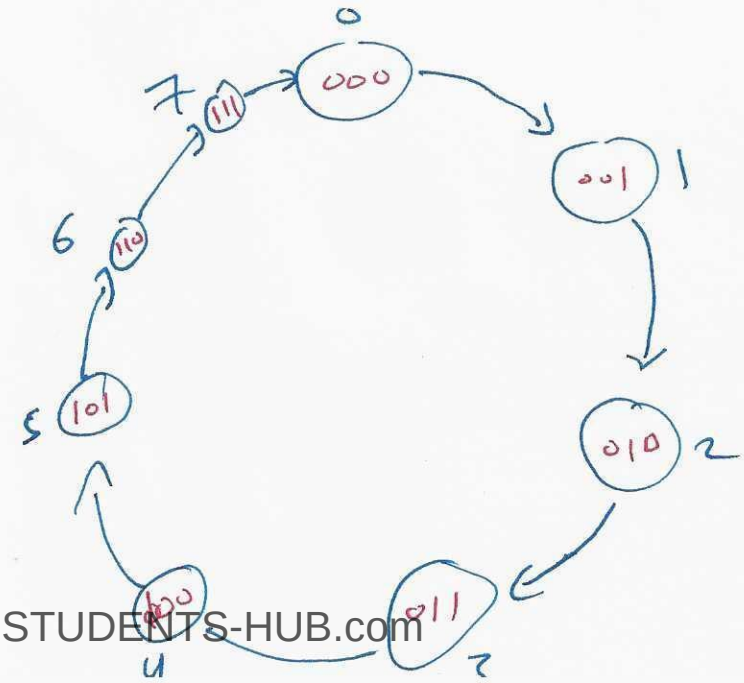


$T_B = \bar{A}B + B \oplus X$

$Y = AB$

* Design of Binary Counter

Example: Design a circuit counts up from 0 to 7 then back to 0



present state			Next state		
Q_2	Q_1	Q_0	$Q_2(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

using D-flip-flops

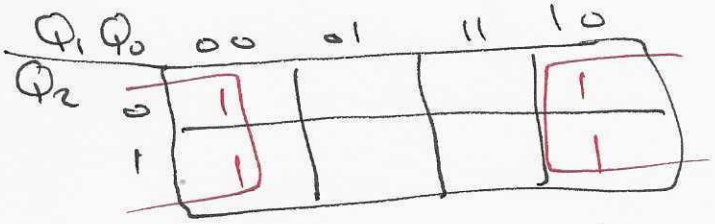
we have 8 states ∴ # of flip-flops = $\log_2(8) = 3$

$D_2 = Q_0(t+1) = \Sigma(3,4,5,6)$

$D_1 = Q_1(t+1) = \Sigma(1,2,5,6)$

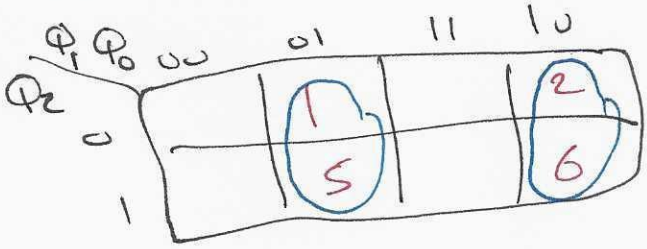
$D_0 = Q_2(t+1) = \Sigma(0,2,4,6)$

$D_0 = \bar{Q}_0$



$D_1 = \bar{Q}_1 Q_0 + Q_1 \bar{Q}_0$

$= Q_0 \oplus Q_1$

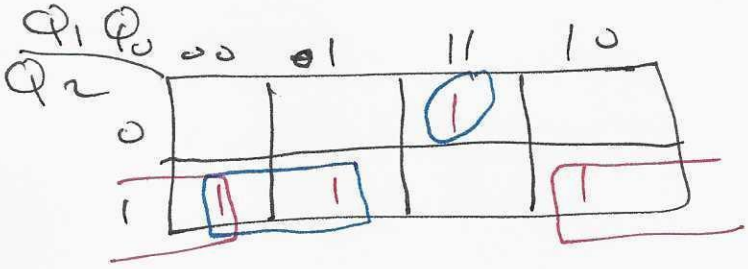


$D_2 = Q_2\bar{Q}_1 + Q_2\bar{Q}_0 + \bar{Q}_2 Q_1 Q_0$

$= Q_2(\bar{Q}_1 + \bar{Q}_0) + \bar{Q}_2 Q_1 Q_0$

$= Q_2(Q_1 Q_0)' + \bar{Q}_2(Q_1 Q_0)$

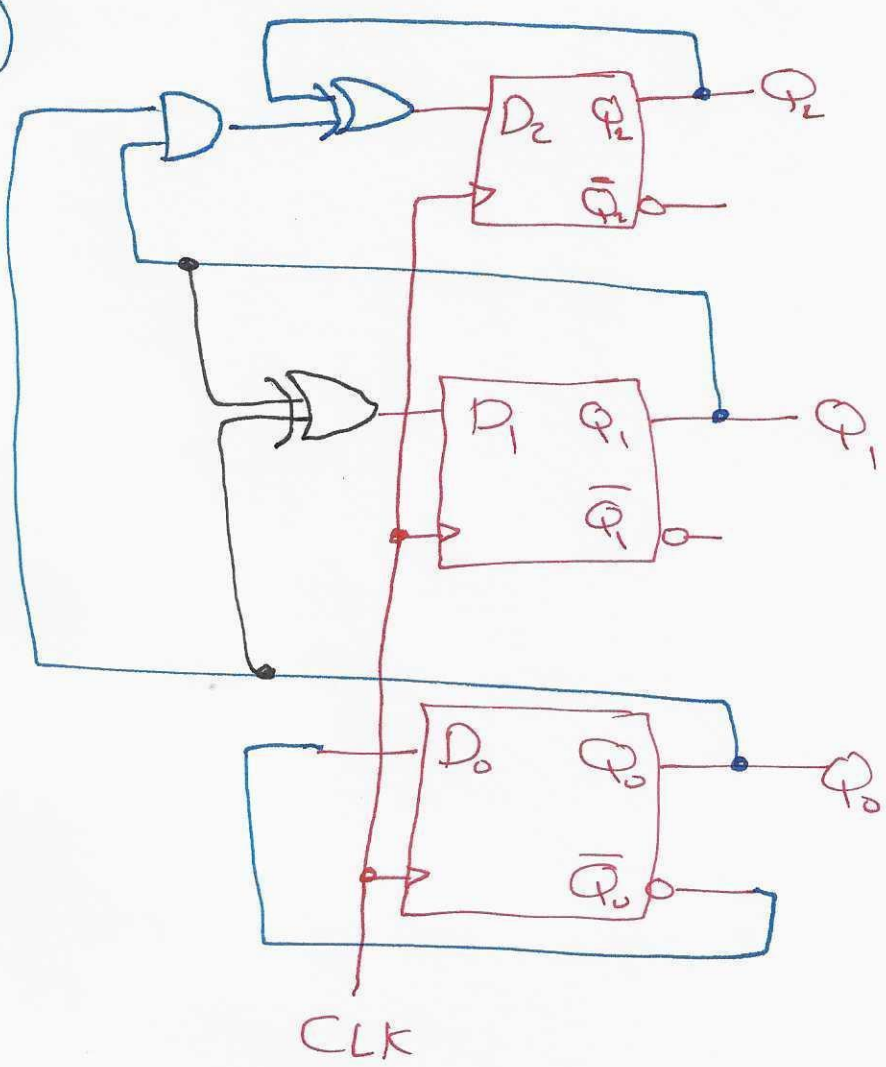
$= Q_2 \oplus (Q_1 Q_0)$



$D_2 = Q_2 \oplus (Q_1 Q_0)$

$D_1 = Q \oplus Q_0$

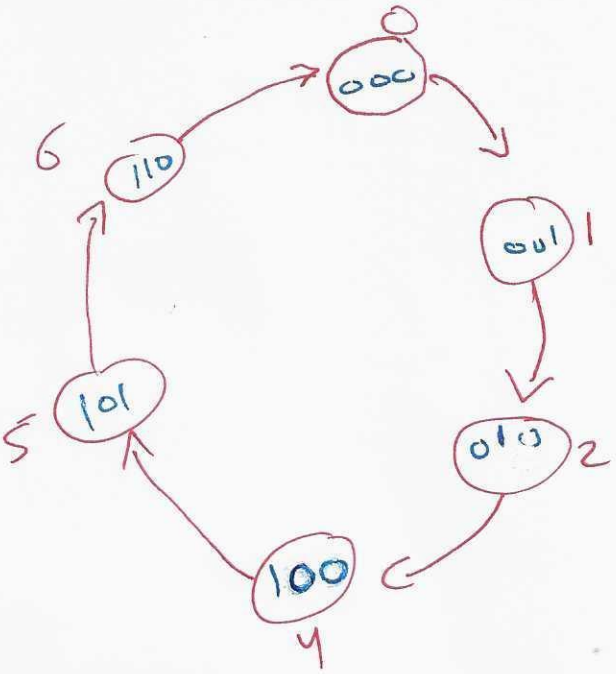
$D_0 = \overline{Q_0}$

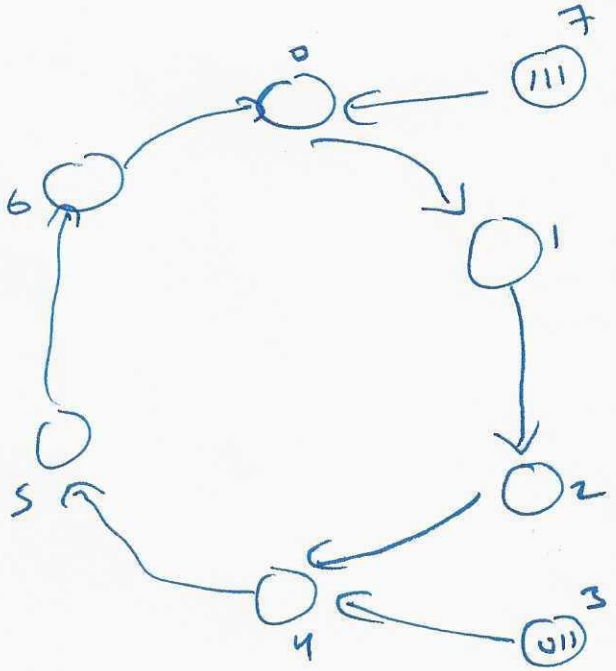


* Dealing with unused states

- in some cases we wish to use less than the total number of states available
- The unused state may be treated as "don't care" or assigned to specific next state
- we should ensure that a Counter enter one of its unused states, eventually goes into one of the valid states after one clock pulse.

Example:-





states 3 and 7 are unused

Don't Care

or

self Correcting Counter

Q_2	Q_1	Q_0	$Q_2(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	X	X	X
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	X	X	X

Q_2	Q_1	Q_0	$Q_2(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	0	0	0