(#) Preset & Clear in SR Flip Flop

They are asynchronous i/p i.e. they do not require any clock They can be operated in any application. In many applications it is desired that before starting any application or operation the initial cond" of the flipflop should be known That can be set with the help of the two buttons

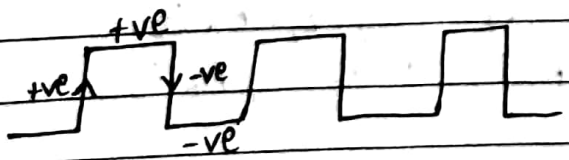
⇒ Preset = 0 It sets the ff
Clear = 1

when $P_r = 1$ it resets the ff
$C_r = 0$

$P_r = C_r = 0$ uncertain
$P_r = C_r = 1$ normal ff

(#) Clock

Periodic train of pulses



Level triggering — +ve, -ve

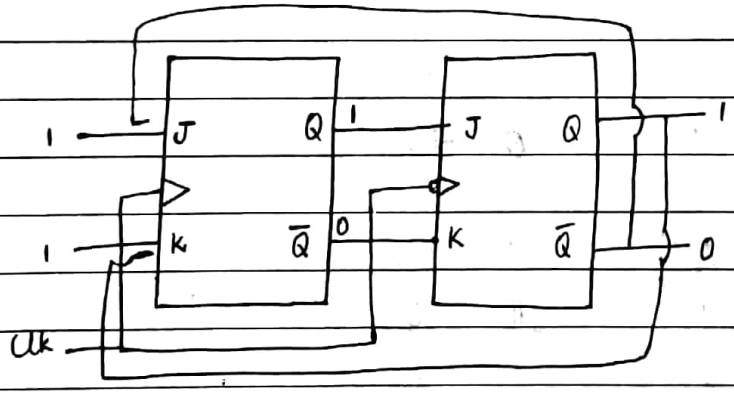
Edge triggering — +ve, -ve

Race around condition (JK ff)

$S = R$

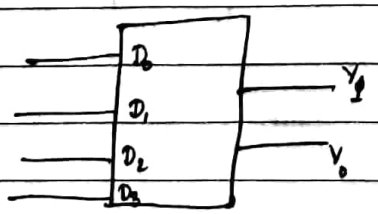
When clock pulse of width t_p is applied an i/p $J = K = 1$ with $Q_n = 0$ is applied then after time interval Δt we get o/p $Q = 1$. Now again after time period of Δt we get the $Q = 0$
So for whole t_p, o/p keeps on oscillating b/w 0 & 1 and this cond^n is k/a race around cond^n.

To overcome this problem there are 2 methods:
i) $t_p < \Delta t$
ii) Master slave JK ff



The master slave ff is made up of 2 JK ff connected in serial manner. Both are collected with clock. Master is one which is driven by +ve clock & slave is driven by -ve

Priority Encoder



D_0	D_1	D_2	D_3	Y_1	Y_0
1	0	0	0	0	0
*	1	0	0	0	1
*	*	1	0	1	0
0	0	0	1	1	1

It is giving priority to highest order valid bit

Excitation Table of Flipflop

Sometimes we know the present state & the desired next state. We just have to find out what i/p should be given to the ff which can give the desired o/p.
The tabulation of all these cond's is called excitation table of ff

SR

P.S.	N.S.	S	R
0	1	1	0
0	0	0	x
1	1	x	0
1	0	0	1

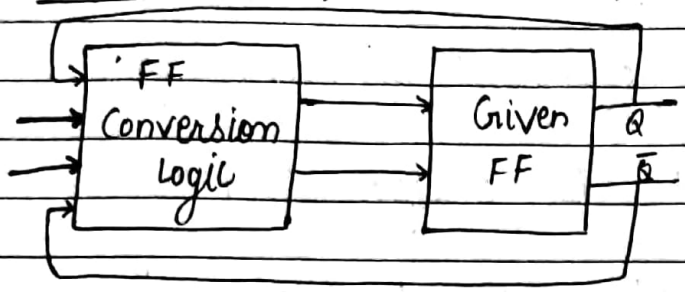
JK

PS	NS	J	K
0	1	1	x
0	0	0	x
1	1	x	0
1	0	x	1

PS	NS	T
0	0	0
0	1	1
1	0	1
1	1	0

PS	NS	D
0	0	0
0	1	1
1	0	0
1	1	1

Conversion from one flipflop to another flipflop



To design a conversion logic we need to combine the excitation table of both FF then make the truth table with data i/p & Q as i/p and i/p of given ff as o/p.

Q. Make D flipflop using SR ff.

D	Q_n	Q_{n+1}	S	R
0	0	0	0	x
0	1	0	0	1
1	0	1	1	0
1	1	1	x	0

Q_n \ D	0	1
0		1
1		x

$S = D$

Q_n \ D	0	1
0	x	
1	1	

$R = \bar{D}$

Counter Designing

MOD-7 It can count 7 different stages

JK FF or T FF to design counter

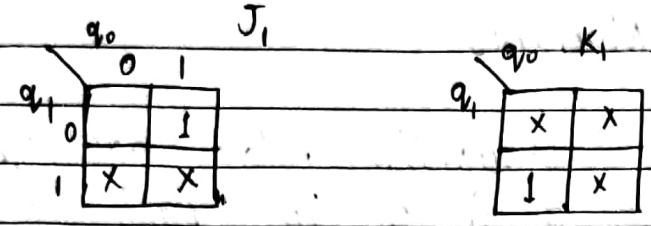
No. of FF = 2^n No. of stages = $2^n - 1$

PS	NS
0	1
1	2
2	3
3	4
4	5
5	0

PS	NS
000	001
001	010
010	011
011	100
100	101
101	000
~~110~~	

Design Mod-3 counter with JK FF

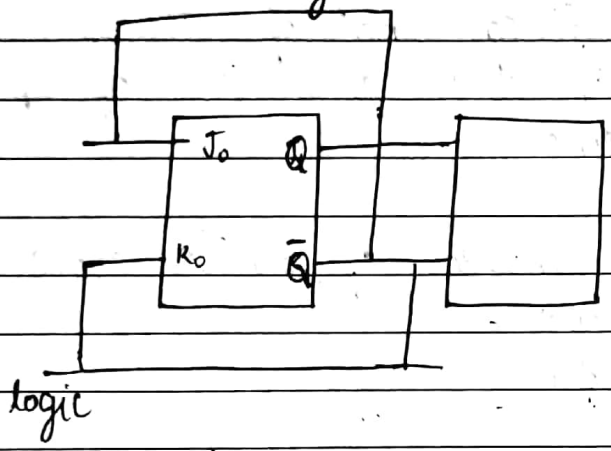
PS $q_1 q_0$	NS $Q_1 Q_0$	J_1 K_1	J_0 K_0
00	01	0 x	1 x
01	10	1 x	x 1
10	00	x 1	0 x



$J_1 = q_0$ $\quad$ $J_0 = \bar{q}_0$

$K_1 = 1$ $\quad$ $K_0 = 1$

Draw ckt diagram



Lockout cond^n

Sometimes during working of the counter it goes into some unused stage & keeps on counting those unused stages and never come back to the demanding stage is called lockout cond^n.

To overcome this problem there are 2 methods.

① Self starting counters
② Self correcting counters

Finite State Machines

As we know sequential m/c's are those in which o/p is a fn of i/p, past i/p's & o/p's & internal states of flipflop

A finite state m/c is an abstract model of sequential m/c's b'coz they have a limited storage capacity with fixed no. of states that are actually used in the calculation of o/p depend on i/p conditions.

There are two models of finite state m/c's:
i) Melay type model
ii) Moore type model

In Melay models we are considering that o/p is a fn of present state & present i/p whereas in Moore mode o/p is a fn of present state on

→ State diagrams

The state diagram is a pictorial representation of the relationship b/w present state i/p, next state and o/p of the sequential ckt.

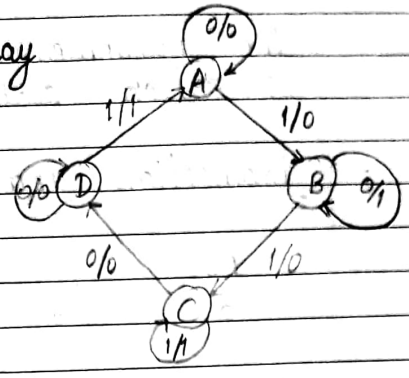
→ State Table

The state table is tabular representation in tabular form of i/p, o/p, present state & next state

Suppose a sequential ckt has four states A, B, C, D

PS	NS, O/p I/p x x = 0	x = 1
A	A, 0	B, 0
B	B, 1	C, 0
C	D, 0	C, 1
D	D, 0	A, 1

Melay

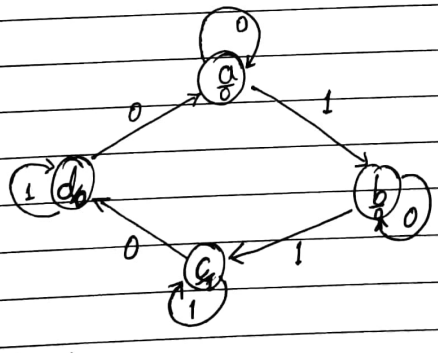


For Moore diagram

The ~~state~~

~~PS~~

PS	NS x = 0 I/p x x = 1		O/p
a	a	b	0
b	b	c	0
c	d	c	1
d	a	d	0



State Reduction

→ State Assignment

The assignment of binary values to th
is called state assignment.

→ Transition and o/p Table

P.S	N.S.		o/p	
	x=0	x=1	x=0	x=1
a	a	b	0	0

When you are separating the column for
& o/p for the state table, the table is k
transition & o/p table.

Minimization of completely specified sequential m/c using partition technique

PS	NS, Z	
	x = 0	x = 1
A	C, 0	F, 0
B	D, 1	F, 0
C	E, 0	B, 0
D	B, 1	E, 0
E	D, 0	B, 0
F	D, 1	B, 0

Step 1:

P_1 = (A, C, E) (B, D, F)

(C, E, D) (D, B, D) — 0 successor

(A, C, E) (B, D, F)

(F, B, B) (F, F, B) — 1 successor

P_2 = (A, C) (E) (B, F) (D)

0 ⇒ (C, E) (D) (D, D) (B)

1 ⇒ (F, B) (B) (F, B) (E)

P_3 = (A) (C) (E) (B, F) (D)

0 ⇒ (C) (E) (D) (D, D) (B)

1 ⇒ (F) (B) (B) (F, B) (E)

P_3 = (A) (C) (E) (B, F) (D)

⇒ B & F are equivalent

For the table given below

PS	NS, Z	
	X = 0	X = 1
A	F, 0	B, 0
B	D, 0	C, 0
C	F, 0	E, 0
D	G, 1	A, 0
E	D, 0	C, 0
F	F, 1	B, 1
G	G, 0	H, 0
H	G, 1	A, 0

$P_1 = (A, B, C, E, G)(D, H)(F)$

0 ⇒ (F, D, F, D, G)(G, G)(F)

1 ⇒ (B, C, E, C, H)(A, A)(B)

$P_2 =$ ~~(A, C)(B, E)(G)(D, H~~

(A, C)(B, E)(G)(D, H)(F)

0 ⇒ (F, F)(D, D)(G)(G, G)(F)

1 ⇒ (B, E)(C, C)(H)(A, A)(B)

A ≡ C B ≡ E D ≡ H