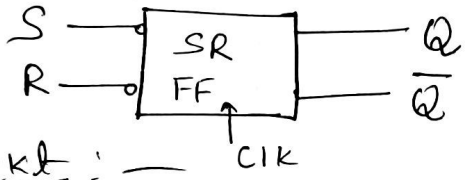
S-R Flip-Flop

SR Flip Flop → gated Set - Reset Flip-Flop

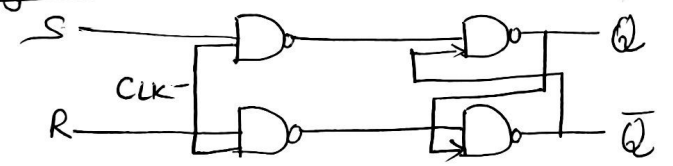
Flip Flop ⇒ FF is a circuit that maintain a state until directed by input to change the state. In sequential logic circuit the flip-flop is basic storage element.

S.R. Flip Flop ⇒ → Set — 1
→ Reset — 0

Block dia. :—



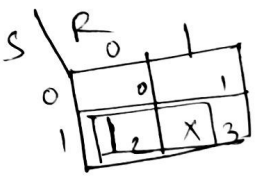
logic ckt :—



Q Qn
0 = 0
1 = 1

Truth Table :—

Clk	S	R	Q. Next
⎍	0	0	Q. Previous state
⎍	0	1	0
⎍	1	0	1
⎍	1	1	Initially forbidden X



Note :—

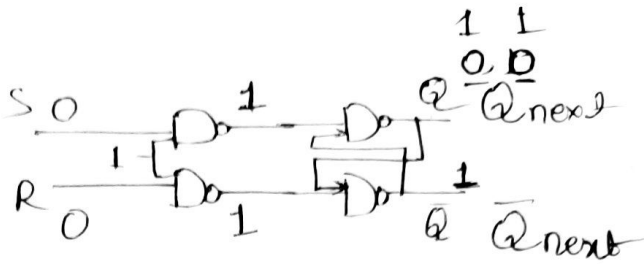
NAND Gate T.T.

0	0	1
0	1	1
1	0	1
1	1	0

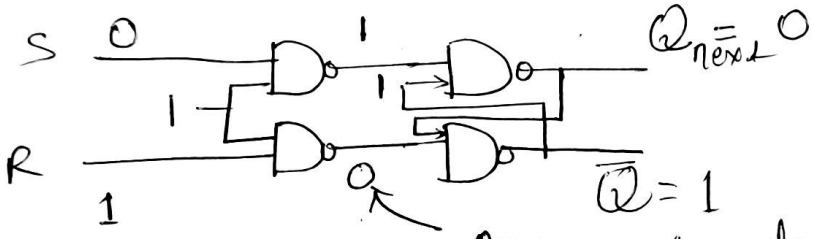
Case ① S R
0 0 Q Previous value

Nand gate

0	0	1
0	1	1
1	0	1
1	1	0

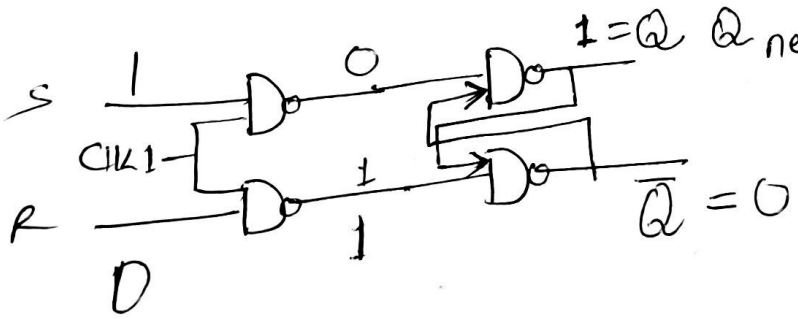


Case ② S = 0 R = 1

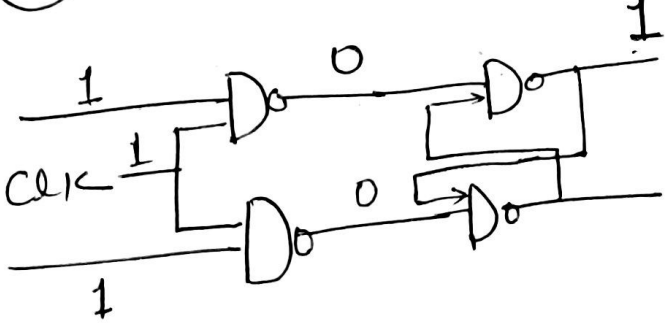


Any one input '0' output 1

Case ③ S = 1 R = 0



Case ④ S = 1, R = 1



Q	$\bar{Q}$	
1	1	X

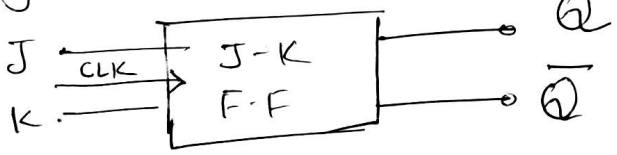
Forbidden

J-K Flip Flop (Jack Kilby)

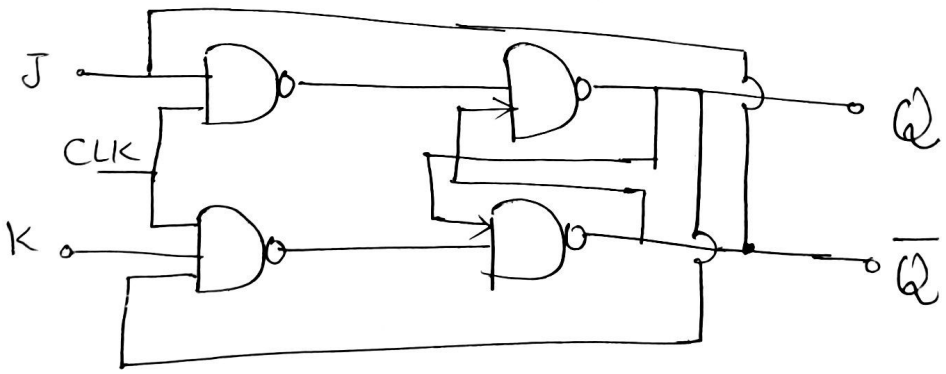
S	R	Q next
0	0	No change
0	1	0
1	0	1
1	1	Invalid

To solve the Invalid ~~position~~ state of S-R flip flop we use J-K flip flop

Block diagram :-



logic circuit :-

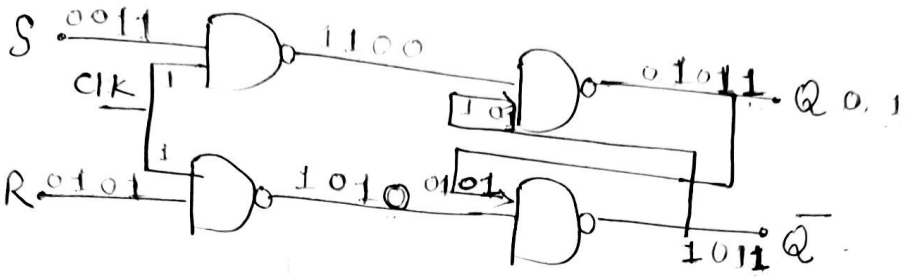


Truth Table :-

J	K	Q next
0	0	No change
0	1	0
1	0	1
1	1	$\overline{Q}$

SR Flip Flop by K-map

Logic Circuit :-

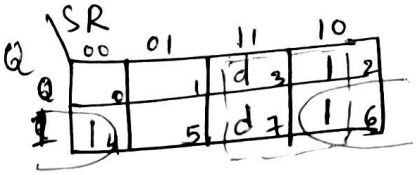


Truth Table :-

	I/p			O/p
Clk	Q	S	R	Q next
↑	0	0	0	0
↑	0	0	1	0
↑	0	1	0	1
↑	0	1	1	x (1 $\bar{Q}$)
↑	1	0	0	1
↑	1	0	1	0
↑	1	1	0	1
↑	1	1	1	x (1 Q)

We are having 3 inputs

$\therefore \quad 2^3 = 8$ K-map



$$F(Q,S,R) = S + Q\bar{R}$$

J K Flip Flop By K-Map

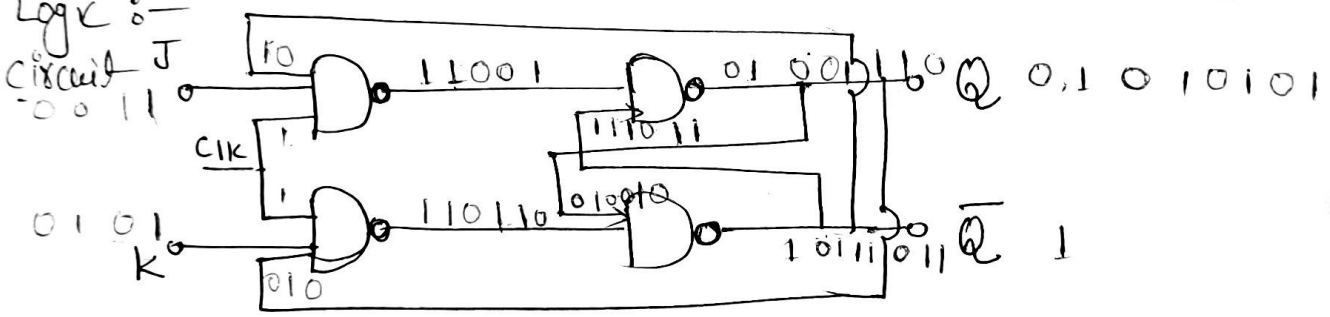
→ J K flip-flop is a re-finement of the S-R flip-flop to solve the problem of Invalid state when both inputs are 1

→ J K flip-flop input J & K is used same as R-S flip flop i/p where J - set, K is Reset

→ J K is named by scientist Jack Kilby

→ Extra feedback required as shown in fig.

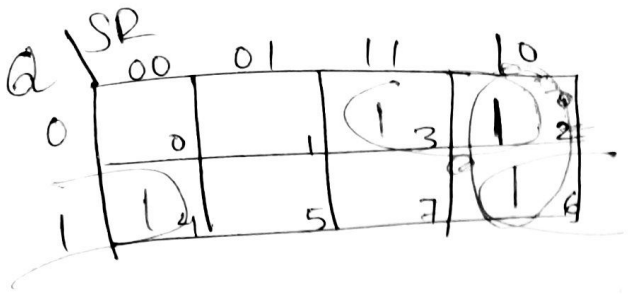
Logic Circuit :—



Truth Table :—

CLK	Q	J	K	Q_{next} or $\overline{Q}$
↑	0	0	0	0
↑	0	0	1	0
↑	0	1	0	1
↑	0	1	1	1 $\overline{Q}$
↑	1	0	0	1
↑	1	0	1	0
↑	1	1	0	1
↑	1	1	1	0 $\overline{Q}$

J	K
0	0
0	1
1	0
1	1



$F = \bar{Q}S + S\bar{K} + Q\bar{R}$

Race Around condition of J-K Flip Flop

For JK Flip-Flop if $J = K = 1$ & if clock is too long then state of flip flop keep on toggle which leads to uncertainity in determining output state of flip flop. This problem is called Race Around condition