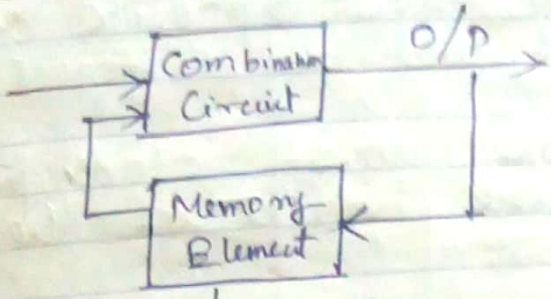
@Orunoyan@

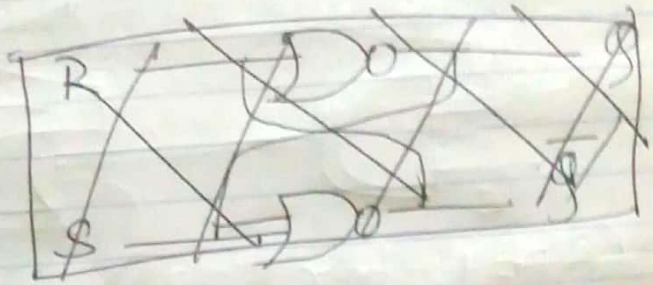
SKG (Extra class)

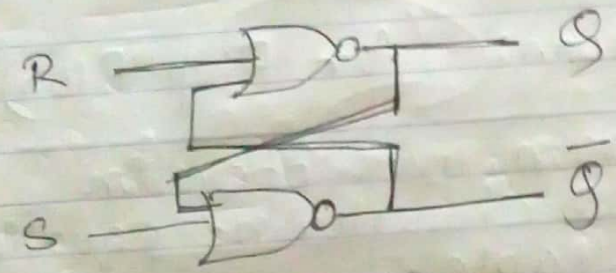
Block Diagram of Sequential Circuit



Flip/Flop → Lowest memory element
→ Can store 1-bit data.

NOR		
A	B	C
0	0	1
0	1	0
1	0	0
1	1	0





Say, $\underline{Q=1}$ & $\underline{\bar{Q}=0}$

1) R=0, S=0 ⇒ NC ⇒ Q=1; $\bar{Q}$=0.

2) R=0, S=1 ⇒ Set (Q_{t+1}=1) ⇒ Q=1, $\bar{Q}$=0.

(when storing =1, Set Condition)

(Flip-Flop output=0, Reset Condition)

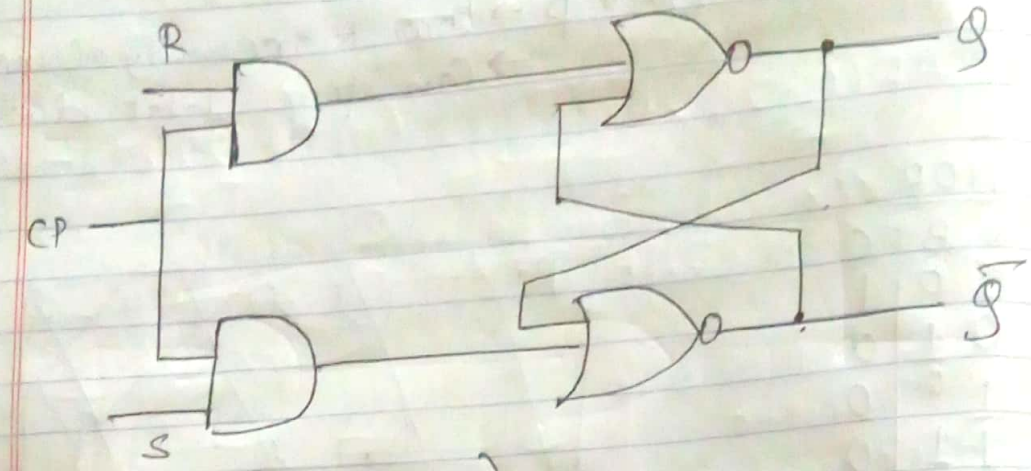
3) R=1, S=0 ⇒ Q=0, $\bar{Q}$=1. ⇒ (Q_{t+1}=0) ⇒ Reset

4) R=1, S=1 ⇒ Q=0, $\bar{Q}$=0 ⇒ Indeterminate Condition.

Latch → closing it, or, [illegible]
↳ Externally changing the value if u change so no change.

→ Flip-Flop must have opposite results q and $\bar{q}$ should be different.

→ Counter counts no. of clock pulse arriving at the input of the circuit

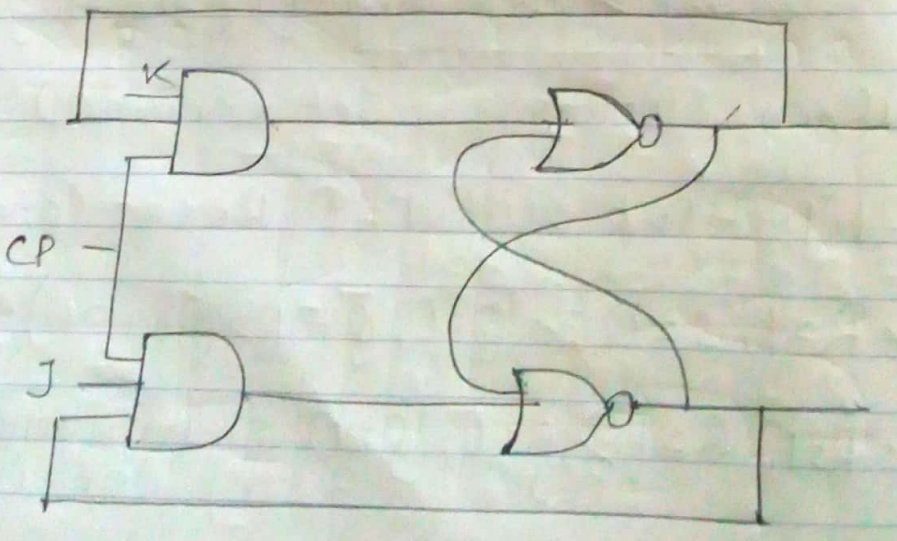


(CP=1, for SR).

Points to learn:- (for Flip-Flop)

(i) Circuit diagram, with operation
(ii) Truth table & characterising equation.
(iii) Excitation Table.

JK-Flip Flop



For, $Q_t = 1$ and CP = 1.

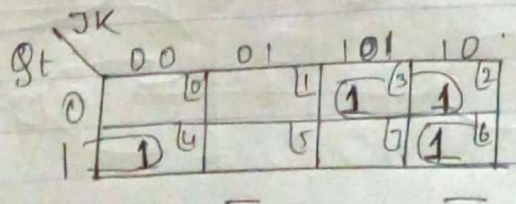
J	K	Q_{t+1}	
0	0	1	NC.
0	1	0	→ Reset.
1	0	1	→ Set.
1	1	0	→ Toggle (change in State).

For, $Q_t = 0$ and CP = 1.

4th case, $Q_{t+1} = 1$.

J	K	Q_{t+1}	
0	0	Q_t	
0	1	0	→ Reset
1	0	1	→ Set
1	1	$\bar{Q}_t$	→ Toggle

Q_t	J	K	Q_{t+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

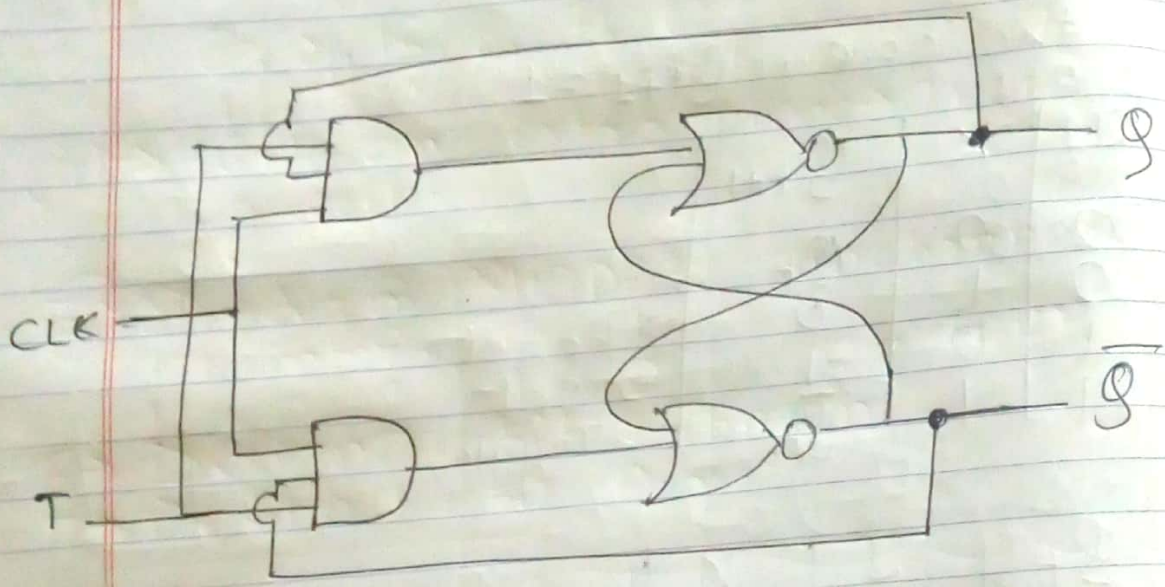


$Q_{t+1} = \bar{Q}_t J + Q_t \bar{K}$ (Characteristic Equation)

Excitation table :—

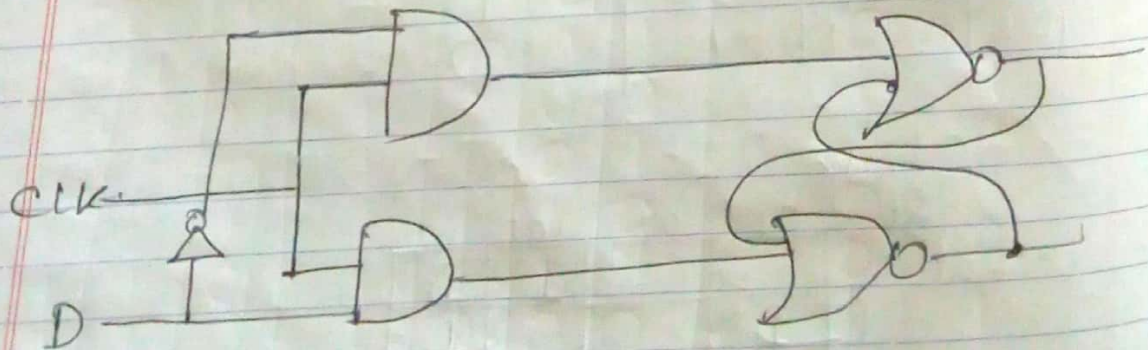
Q_t	Q_{t+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

T-flip flop (Toggle).



J = K = 0 }
J = K = 1 }

D-FF (Delay) (Works as a Buffer)

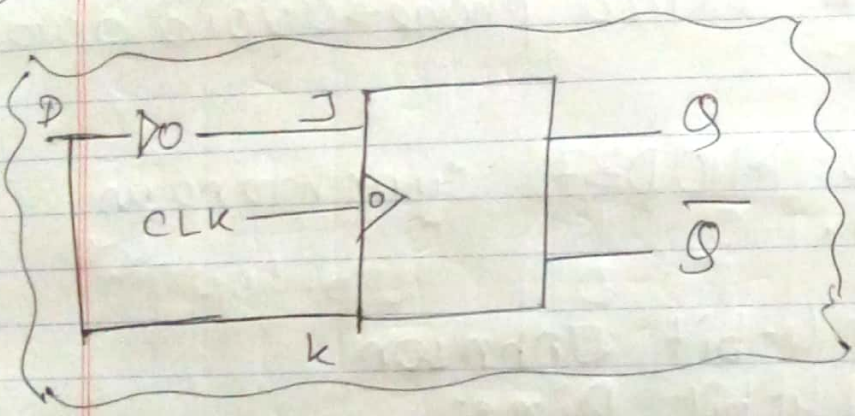


$D = 1, CLK = 1$ } → Set.
i.e., $S = 1, R = 0$ }

$D = 0, CLK = 1$ } → Reset.
i.e., $S = 0, R = 1$ }

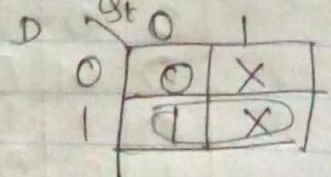
for, $D = 1$, or $D = 0$, with $CLK = 0$.
→ then No Change condition will come.

(*) → Convert JK to D-f/f.



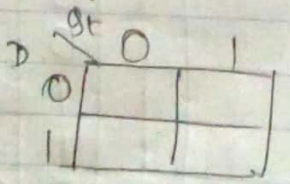
D	Q_t	Q_{t+1}	J	K
0	0	0	0	X
0	1	0	X	1
1	0	1	1	X
1	1	1	X	0

$J = D\overline{Q_t}$
$K = \overline{D}$



for J.

$J = D$
$K = \overline{D}$



For K

(*) no. of bit = no. of flip flops.

(1) Synchronous
(2) Asynchronous

1 → Design 2-bit up-counter.
2 → Design 8-bit down-counter.
3 Design 2-bit up-down synchronous with external M=1 and M=0.
4→(i) Design a counter which counts the following sequence

0, 2, 5, 6, 7, 0.

(ii) → With an extra phone → avoid data log out.

5→ Design MOD-7 synchronous counter.

6→ Design 4-bit Johnson
7→ Design 4-bit Ring.

→ 3-bit up-counter.

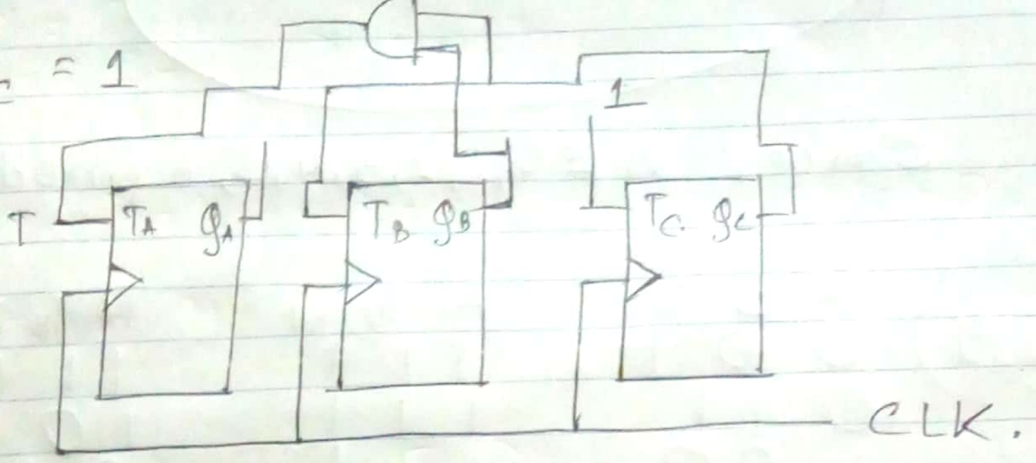
P S			N S			T_A	T_B	T_C
A	B	C	A_1	B_1	C_1			
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

With ABC in the K-Map →

$T_A = BC$.

$T_B = C$

$T_C = 1$



4 → (i) 0, 2, 5, 6, 7, 0.

0 → 2 → 5 → 6 → 7

A	B	C	A_1	B_1	C_1	T_A	T_B	T_C
0	0	0	0	1	0	0	1	0 - 0
0	1	0	1	0	1	1	1	1 - 2
1	0	1	1	1	0	0	1	1 - 5
1	1	0	1	1	1	0	0	1 - 6
1	1	1	0	0	0	1	1	1 - 7

$T_A = BC + A'B$.

A \ BC	00	01	11	10
0	0	X 1	X 3	1 2
1	X 4	5	1 7	6

$T_B =$

$T_C =$

(ii) 0 → 2 → 5 → 6 → 7 ← 1, 3, 4

Unused sequence forcefully brought to sequence.

(5) MOD-7.

$0 \to 1 \to 2 \to 3 \to 4 \to 5 \to 6$ (back to 0)

Since, $n \equiv 0, 1, 2, 3, 4, 5, 6 \pmod 7$.

(4)(ii)

A	B	C	A_1	B_1	C_1	T_A	T_B	T_C
0	0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0	0
1	0	0	1	1	1	0	1	1

(5)

M	A	B	A_1	B_1	
1	0	0	0	1	up-counter
1	0	1	1	0	
1	1	0	1	1	
1	1	1	0	0	
0	0	0	1	1	down counter
0	0	1	0	0	
0	1	0	0	1	
0	1	1	0	0	

T_A	T_B
0	1
1	1
0	1
1	1
0	1
1	1
0	1
1	1