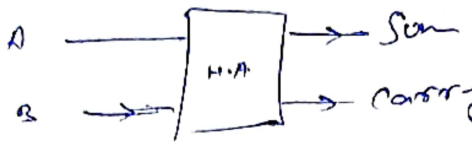
$\Sigma m(4,6) = TTM(0,1,2,3,5,7)$

Half Adder XOR AND

↳ 2 bit adder.

```
  0        1
 +0       +1
 --       --
 00       10
 ↑↑       ↑↑
Carry Sum  Sum
           Carry.
```

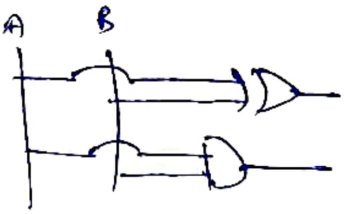


A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Sum = $A \oplus B$

Carry = AB

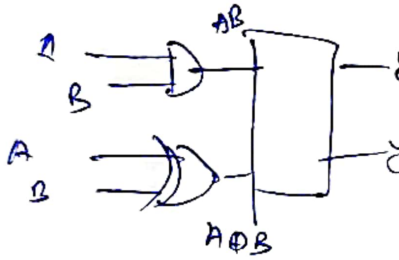
Already minimize d



Total 5 NAND required

5 NOR.

#1

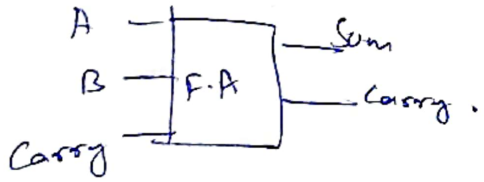


y_0 $(A \oplus B) \oplus AB \rightarrow \bar{A}B + A \rightarrow A + B$

y_1 $(A \oplus B)\,AB \rightarrow 0$.

Full adder

↳ 3bit Add.



Total 9 NAND or NOR.

```
  0        1
  0        1
 +0       +1
 --       --
 00       11
```

A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$= \Sigma m(1,2,4,7)$ = $A \oplus B \oplus C$

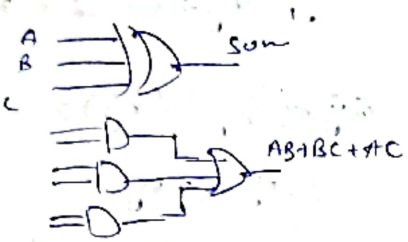
$Sum = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

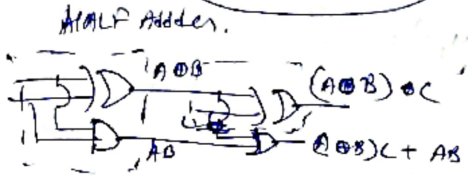
$Carry = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$

$= \Sigma m(3,5,6,7)$

$\star = (A \oplus B)C + AB$ → semi minimised

$AB + BC + AC$





2 HALF Adder and 1 ~~or~~ or gate

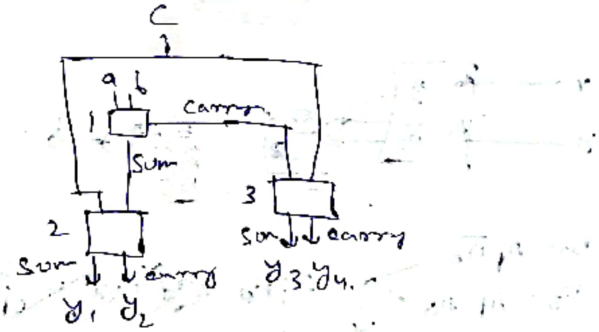
* 3 HA HA_1, HA_2, HA_3 are inter-coupled, as shown below. The 4 output functions y_1, y_2, y_3, y_4 are expressed in terms of input a, b, c.

① $y_1 = (a \oplus b)c$

② $y_2 = (a \oplus b) \oplus c$

③ $y_3 = ab \oplus c$

④ $y_4 = a(b \oplus c)$



① Sum = $a \oplus b$

~~Carry~~ y_4 = ab

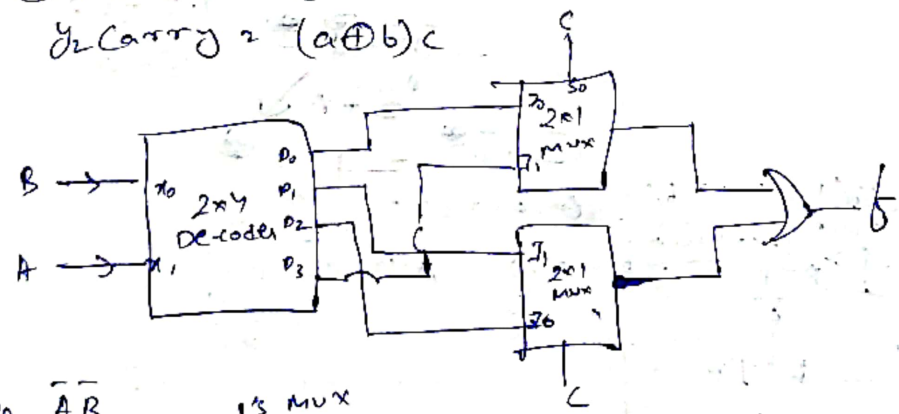
② y_1 Sum = $(a \oplus b) \oplus c$

y_2 Carry = $(a \oplus b)c$

③ y_3 Sum = $(ab) \oplus c$ ✓

y_4 carry = $(ab)c$

#



D_0 $\bar{A}\bar{B}$

D_1 $\bar{A}B$

D_2 $A\bar{B}$

D_3 AB

1's MUX

$I_0 = \bar{A}\bar{B} \rightarrow \bar{A}\bar{B}\bar{C}$ > $\bar{A}\bar{B}\bar{C} + AB\bar{C} \Rightarrow \Sigma m(0,3,4,7)$.

$I_1 = AB \rightarrow AB\bar{C}$

2nd MUX

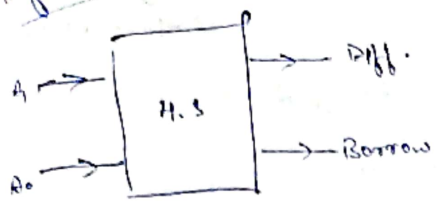
$I_1 \rightarrow \bar{A}BC \rightarrow \bar{A}BC + A\bar{B}\bar{C}$

$I_0 = A\bar{B}\bar{C}$

$f \; (\overline{\bar{A}BC})(\overline{A\bar{B}\bar{C}})$

$(A+\bar{B}+C)(\bar{A}+B+\bar{C})$

Half Subtractor



A_1	B_0	Diff.	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$10 - 1 = 11$

$10 - 1 = 11$; $1 - 1 = 0$

$1 - 0 = 01$; $1 - 1 = 0$; $1 - 0 = 1$

SNAND or NOR.

$Diff = A_1 \oplus A_0$

Borrow = $\bar{A} B$.

Full Subtractor.

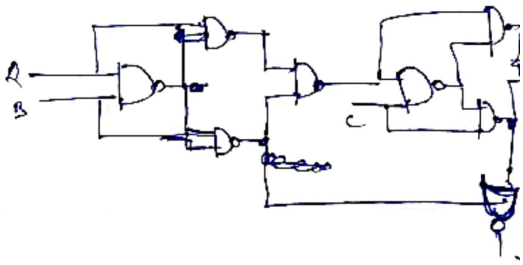
$Diff = A_2 \oplus A_1 \oplus A_2$ $\quad (A - B) - C$

Borrow = $\bar{A}B + BC + \bar{A}C$

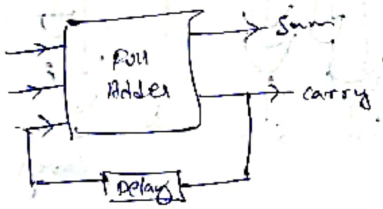
$= (\overline{A \oplus B})C + \bar{A}B$

QNAND or NOR.

2HS + 1 OR



Serial Adder:-
↳ Slowest adder.

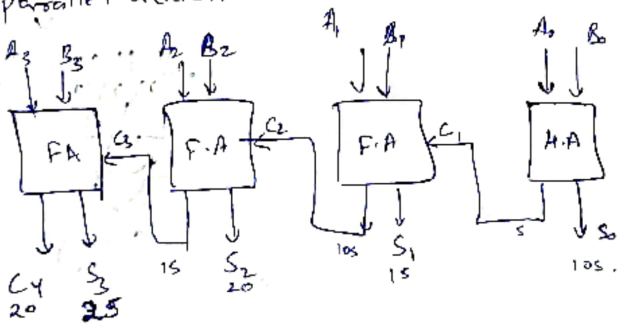


$T = n T_{FA}$

↑ no. of bits

Parallel adder / Ripple carry adder.

4 bit parallel adder.



★ Can be made with

(i) (N-1) F.A + 1 H.A

(ii) N FA

(iii) (N-1) (2H.A + 1OR) + 1HA

↳ (2N-1) HA + (N-1) OR

$T_{sum} = 10s$

$T_{carry} = 5s.$

25 sec.

Worst case delay:

★ $T = (N-1) T_{carry} +$ Max of T_{sum}, T_{carry}

$T = (4-1) \times 5 + \max(10s, 5s)$

$= 3 \times 5 + 10$

$= 25$ sec.

In a 4 bit ripple adder, sum delay and carry delay of each full adder is $t_s = 1$ nsec and $t_c = 5$ nsec respectively. No. of additions per sec. by adder is N_1. If delays are interchanged then it is N_2. Then N_2 / N_1 ——

Case 1

$n = 4 \quad t_s = 1\text{ns} \quad t_c = 5\text{ns}$

$T = 3 \times 5 + 5$

$= 20\text{ns}$

$\therefore N_1 = \frac{1}{20}$

Case 2.

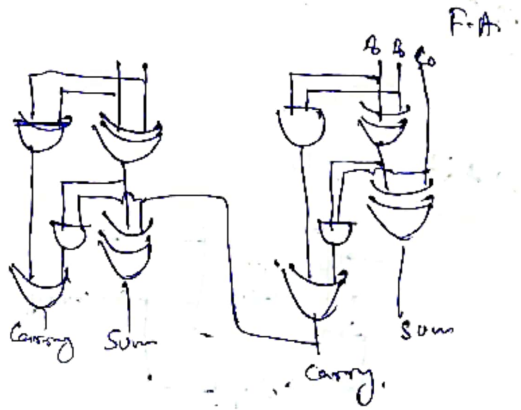
$T = 3 \times 1 + 5$

$= 8$

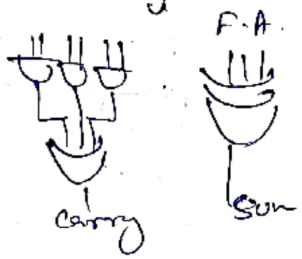
$N_2 = 1/8$

$\therefore \frac{N_2}{N_1} = \frac{20\text{ns}}{8\text{ns}}$

Internal diagram

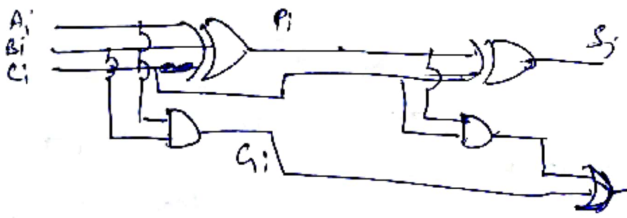


$$T = (n-1)\{T_{AND} + T_{OR}\} + 2 \times T_{XOR}.$$



Look ahead carry adder.

↳ Fastest adder.



P_i = Carry propagating term

G_i = Carry generating term

$P_i = A_i \oplus B_i \qquad S_i = P_i \oplus C_i$

$G_i = A_i B_i \qquad C_{i+1} = G_i + P_i C_i$

Expressions -

$C_1 = G_0 + P_0 C_0$

$C_2 = G_1 + P_1 C_1$

$C_2 = G_1 + P_1 G_0 + P_1 P_0 C_0$

$C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$

$C_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$

* Total delay.

$T = T_{XOR} + T_{AND} + T_{OR} + T_{XOR}$

If $T_{XOR} = T_{AND} = T_{OR} = T_{pd}$.

$T = 4T_{pd}$.

Carry block delay.

$T = T_{AND} + T_{OR}$.

Nor of add gate in Carry Block.

$= \frac{n(n+1)}{2}$

" or " " " .

$\approx n$.

Multiplier

2 bit Multiplier.

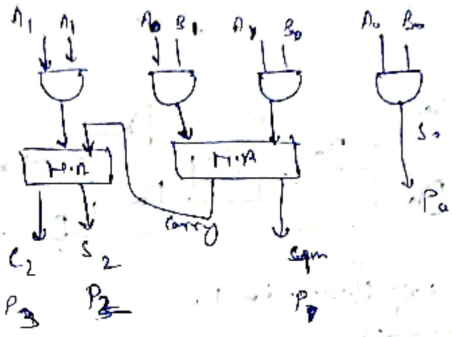
$A_1\ A_0$

$B_1\ B_0$

C_1	A_1B_0	A_0B_0
A_1B_1	A_0B_1	x
		A_1B_0

$C_2\ S_2\ S_1\ S_0$

$P_3\ P_2\ P_1\ P_0$



How many half adder, required for adding K bit no.

* $(K-1)$ FA + 1 HA.

$(K-1)$ (2 HA + 1 OR) + 1 HA.

$2K-2$ HA + 1 HA $= (2K-1)$ HA.

Design a combinational logic circuit with input x y z and output A B C is attempted using HS HA FS FA.

In Binary input 0 1 2 3 → same output

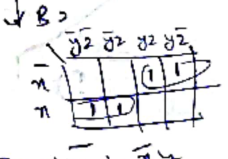
" " 4 5 6 7 → 2 less than binary input

x	y	z	A	B	C
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	0	1

$A = xy\bar{z} + xyz$

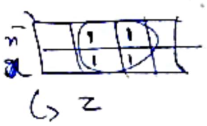
$B = \bar{x}y\bar{z} + \bar{x}yz + x\bar{y}\bar{z} + x\bar{y}z$

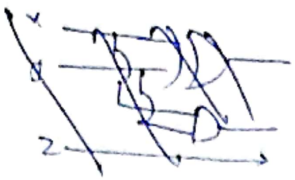
$C = \bar{x}\bar{y}z + \bar{x}yz + x\bar{y}z + xyz$

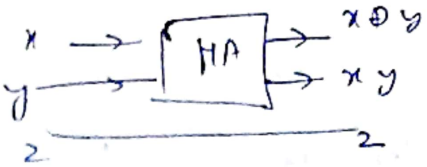


$B = x\bar{y} + \bar{x}y$

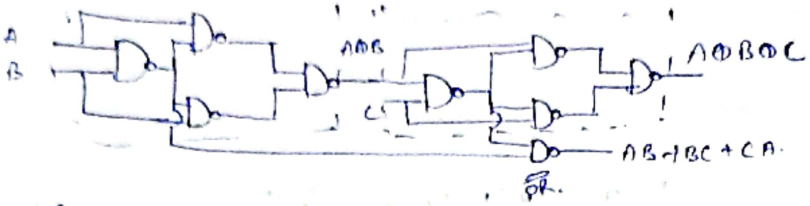
$= x \oplus y$



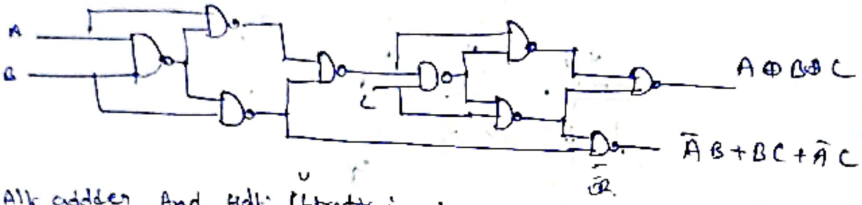




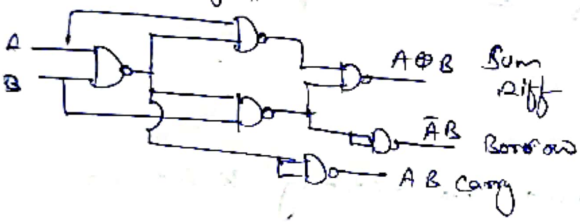
★ Full adder



Full Subtractor.



Half adder And Half Subtractor in 1.



Sequential Circuit

A Circuit with feedback and memory

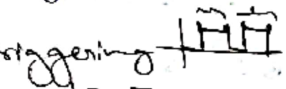
① Latch. ② Flip-flops ③ Registers ④ Counters

↓

Basic memory element

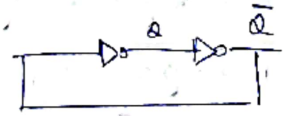
Latches are level triggering

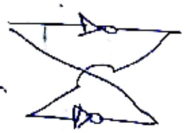
Flip flop are edge "

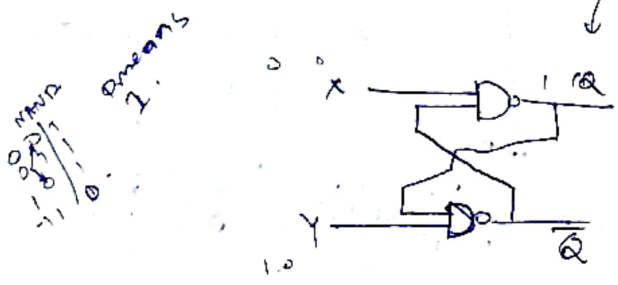


Latch

- It are level triggered.
- It has two output which is complement to each other :





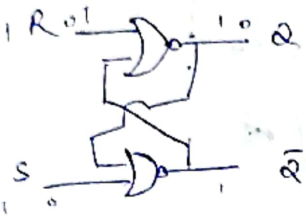


X	Y	Q	Q̄	
0	0	1	1	✕ Invalid
0	1	1	0	set
1	0	0	1	reset
1	1	Q	Q̄	→ HOLD

Racing Problem. / Raising Prob.

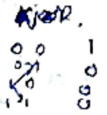
After invalid condition, 1 1 is provided. the one having propagation delay faster will change its value. And the other will not change.

S-R latch



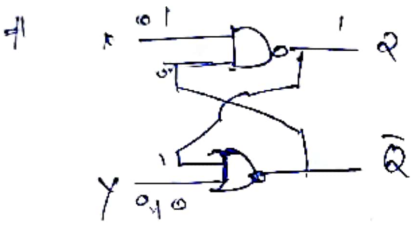
S	R	Q	$\bar{Q}$	
0	0	Q	$\bar{Q}$	HOLD
0	1	0	1	Reset
1	0	1	0	Set
1	1	X	X	

NOR



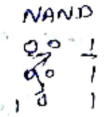
Timing

0.

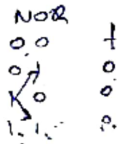


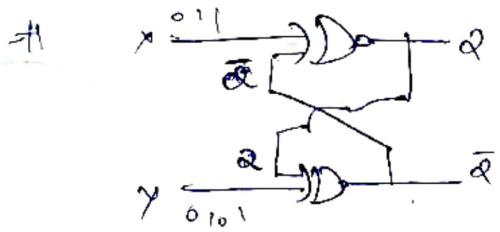
X	Y	Q	$\bar{Q}$
0	0	1	0
0	1	1	0
1	0	Q	$\bar{Q}$
1	1	1	0

NAND



NOR





X	Y	Q	$\bar{Q}$	
0	0	X	X	
0	1	Q	$\bar{Q}$	→ HOLD
1	0	$\bar{Q}$	Q	→ Toggle
1	1	X	X	

XOR

X	Y	
0	0	0
0	1	1
1	0	1
1	1	0

XNOR

X	Y	
0	0	1
0	1	0
1	0	0
1	1	1

$Q \oplus 0 = Q$

$Q \oplus 1 = \bar{Q}$

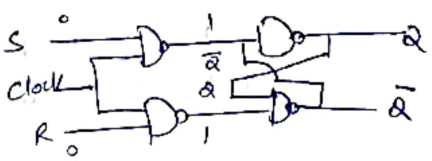
$Q \odot 0 = \bar{Q}$

$Q \odot 1 = Q$

Flip Flop

(i) Circuit Diagram
(ii) Truth table
(iii) Characteristic Table
(iv) " Equation
(v) Excitation Table
(vi) State Diagram

SR Flip Flop



Clock = 1

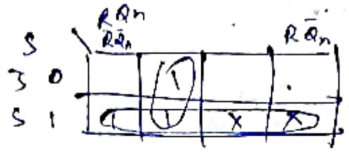
S	R	Q_{n+1}	$\bar{Q}_{n+1}$	
0	0	Q_n	$\bar{Q}_n$	
0	1	0	1	Reset
1	0	1	0	set
1	1	X	X	

Characteristics Table

S	R	Q_n	Q_{n+1}	
0	0	0	0	=
0	0	1	1	
Reset 0	1	0	0	=
Reset 0	1	1	0	
1	0	0	1	
Set 1	0	1	1	
1	1	0	x	
1	1	1	x	

eq.

$Q_{n+1}(S,R,Q_n) = \Sigma m(1,4,5) + \Sigma d(6,7)$

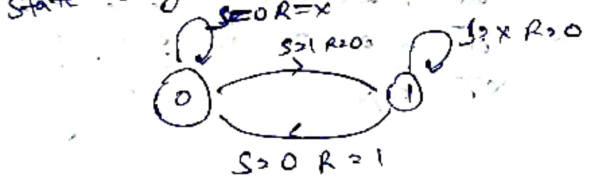


$$Q_{n+1} = \bar{R}Q_n + S$$ ★

Excitation Table

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

State Diagram



JK Flip Flop

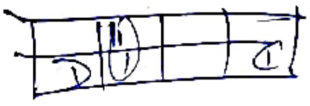
Truth table

J	K	Q_{n+1}	
0	0	Q_n	
0	1	0	
1	0	1	
1	1	$\bar{Q}_n$	→ Toggle

Characteristic table

J	K	Q_n	Q_{n+1}	
0	0	0	0	same as SR
0	0	1	1	
1				
	1			
1	1	0	1	
1	1	1	0	

equation



$$Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$$ ★

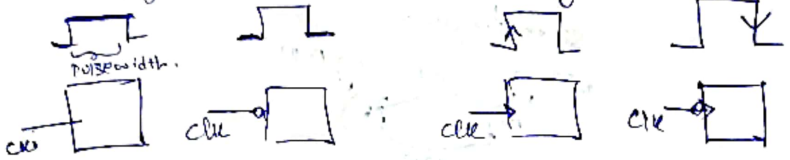
Excitation table

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

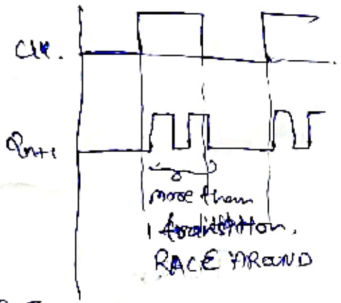
Triggering

Level triggered
- +ve level triggered
- -ve

Edge Triggering
- +ve edge triggered
- -ve



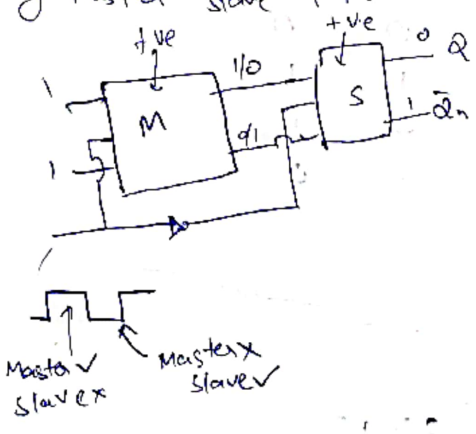
Level triggered JK FF suffers from problem of RACE AROUND.



when J = 1 K = 1

① To avoid RACE AROUND problem. Increase Propagation delay > pulse width.

② By Master Slave FF.

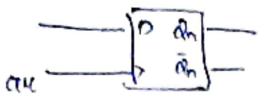


1 time at a time.

at slave J=1, K=1 not possible

- Inverted clock is applied to the slave as compared to master.
- M-S FF is used to store single bit because output is taken only from the slave.

D Flip Flop. Data ff.



D	Q_{n+1}
0	0
1	1

Q.T.

Characteristic T.

↓ No deal with previous state.

D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

Eq: $Q_{n+1} = D$

Excitation table

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

State diagram

T Flip Flop. (Toggle)

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

dependent

→ T=0 Hold's Q_n
T=1 Toggle Q_n

$Q_{n+1} = \bar{T}Q_n + T\bar{Q}_n = T \oplus Q_n$

Excitation T.

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

J-K Flip flop. → Universal ff.

SR.

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\bar{Q}_n$

D FF

T-FF