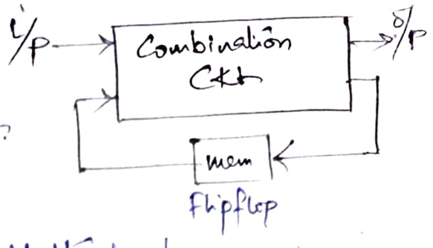
Sequential Circuits

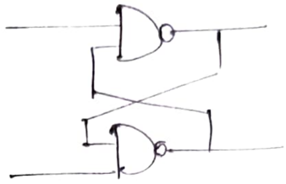


Defn: A circuit in which present output depends not only on present i/ps but also on prev. data is called Seq logic Circuit.

1 bit memory cell: Basic Binary/Digital memory ckt is called Flipflop. it has 2 stable states, also called Bistable Multivibrator. → $Q_n, \overline{Q_n}$

* To store "n" bit data, we require "n" flipflops.

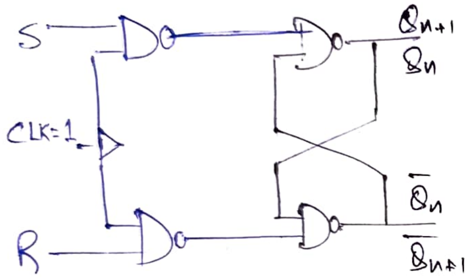
flipflops
- SR FF } Two i/p FFs
- JK FF
- T FF } One i/p FFs
- D FF



Latch or Lock

→ No triggering

SR flipflop:



Simplified TT

* Output Expression:

$$\boxed{Q_{n+1} = S + \overline{R}\cdot Q_n}$$ 4

* Function Table:

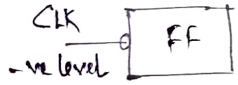
S	R	Q_{n+1}
0	0	Q_n [No change]
0	1	0 [Reset]
1	0	1 [Set]
1	1	ϕ Invalid (or) Not inhibited

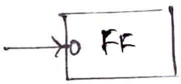
* Excitation table or Transition Table:

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

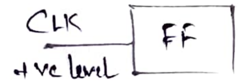
opp.

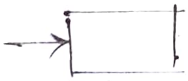
Triggering:



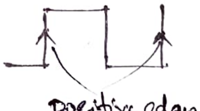


negative edge Triggered

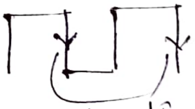




+ve edge Triggered

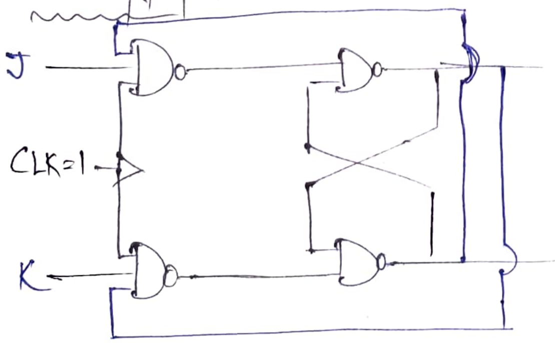


positive edges



negative edges.

JK flipflop:



* Output Expression:

$$\boxed{Q_{n+1} = J\overline{Q_n} + \overline{K}Q_n}$$ 3

* Function Table:

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$ [Complement State]

Same as SR

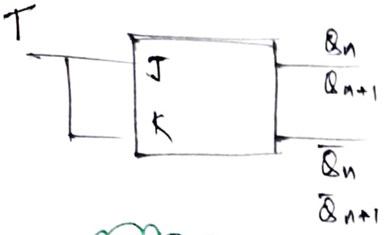
* Excitation Table:

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

opp

same as SR

* T- flipflop :



Toggling

* Output Expression : $\boxed{Q_{n+1} = T \oplus Q_n} \rightarrow$ ②

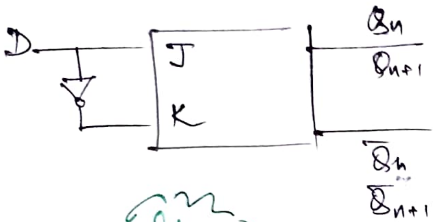
* Function Table :

T	Q_{n+1}
0	Q_n [No change]
1	$\overline{Q}_n$ [Complement]

* Excitation Table :

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

* D flipflop : (Buffer)



Delay → Transparent

* Output Exp : $\boxed{D = Q_{n+1}} \rightarrow$ ①

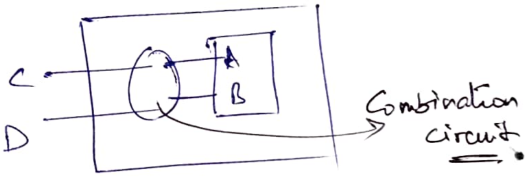
* Fun^n table :

D	Q_{n+1}
0	0
1	1

* Excitation table :

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

* Conversion of one flipflop to another :



Step 1 : Write given FF o/p expression

Step 2 : Write expression from circuit

Step 3 : Obtain expression for the input of given FF and realise them.

* RACE Around Conditions :

During one clock period outputs should change only one time but in JK flipflop when J=1 , K=1 and if the clock pulse width is more, then the o/p continuously keep on changing [toggles (or) racing] 0 to 1 and 1 to 0. This is called as RACE Around Condition.

Note :
Racing is uncontrolled o/p where as toggling is controlled o/p.

RAC occurs in any FF if the following condition satisfies

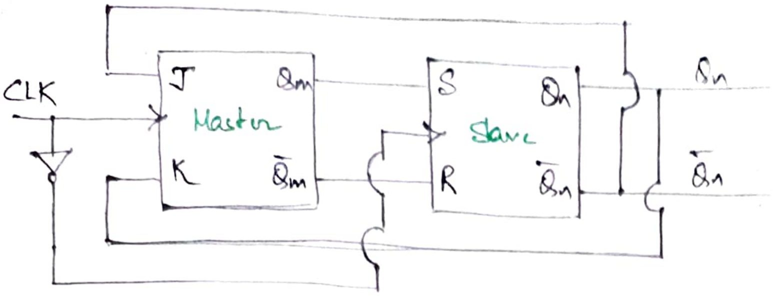
1. If the FFs are operated in level triggering
2. if $t_{pd} < T_{clk\ on}$
3. If the FFs are operated in Toggle mode.

If the above 3 conditions satisfies simultaneously then there is a continuous race in the output of the FF between 0 and 1 to reach the next state, who will be the winner of the race is not certain, than depends on tpd and $(T_{clk})_{on}$.

* Condition to avoid RAC :

1. $t_{pw} < t_{pd} < T_{clock}$
2. Using Master - Slave
3. Using edge-Triggered.

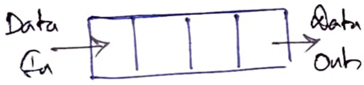
Master - Slave flip flop :-



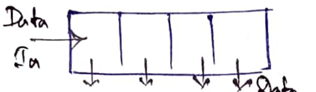
1. Master is connectd using JK flipflop where as slave is constructd using SR flipflop.
2. Master connectd with clock whereas slave is connectd with inverted clock.
3. When master is Enabled [functional] then slave is disabled when slave is enabled then master is disabled. Slave copies Action of master.
4. Master, slave outputs never change at the same time.
5. It is used to store only one bit of data.

Registers.

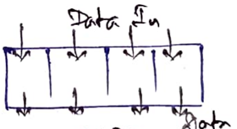
- Group of flipflops are called as Registers.
- n-bit Register consists of n flipflops and used to store n-bit data.



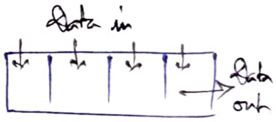
SISO



SIPO



PIPO



PISO

* SISO configuration has Only - 1 - input
1 - output

* SIPO confn only
1 - input
4 - output

* PIPO confn only
4 - i/p
4 - o/p

* PISO confn only
4 - i/p
1 - o/p

* For SISO configuration
for storing = (n) CP
for retrieving = (n-1) CP
Total no of clock pulse = (2n-1)

* For storing = (n) CP
For retriving = 0 CP
Total no of CP = n

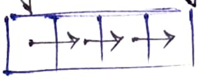
for storing = 1 CP
for retriving = 0 CP
Total CP = 1

for storing = 1 CP
for retriving = (n-1) CP
Total CP = n

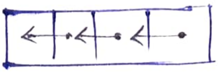
* Shift Registers :- A register in which shifting of data is possible in one or both direction is called as Shift register.

Type of Shift Reg :-

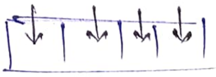
① Right Shift Reg



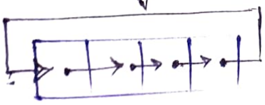
③ Left Shift Reg



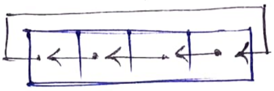
⑤ Parallel Shift in



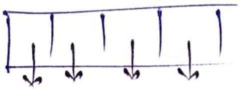
② Rotate Right Shift Reg



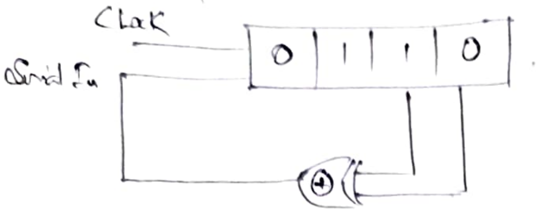
④ Rotate Left Shift Reg



⑥ Parallel Shift out



Q) The initial content of the 4-bit serial-in-parallel-out, right-shift, shift is shown in fig



$SI = \text{Serial In} = Q_1 \oplus Q_0$

After 3 clock pulses, the contents of the shift reg will be
(a) 0110 (b) 0101 (c) 1010 (d) 1011

Q_1	Q_0	SI $Q_1 \oplus Q_0$
0	0	0
0	1	1
1	0	1
1	1	0

Clock	Serial Input $SI = Q_1 \oplus Q_0$		Q_3	Q_2	Q_1	Q_0
0	-		0	1	1	0
1	1	⟶	1	0	1	1
2	0	⟶	0	1	0	1
3	1	⟶	1	0	1	0

* **Counters:** A counter is a register that goes through a Predetermined Sequence of States. Any possible output of a counter is known as its state, for a n-bit counter the maximum possible states is 2^n

* **Modulus of a Counter:** Modulus (MOD) of a counter indicates Number of different output states.

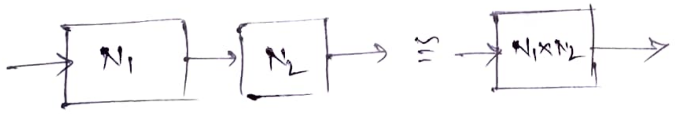
In MOD-N counter, Number of different output states = N.

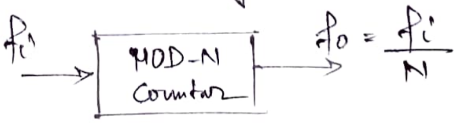
Design equ of a counter,

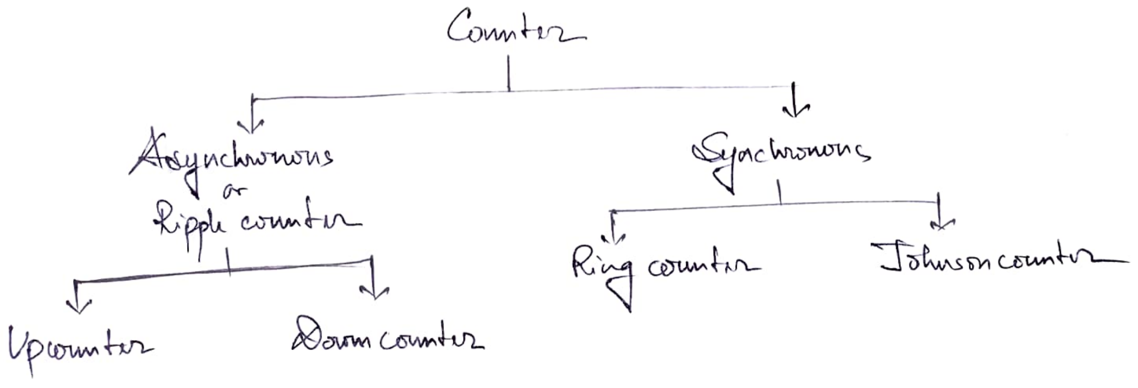
$$2^n \geq N \qquad n \rightarrow \text{no of flipflops}$$
$$n \geq \log_2 N \qquad N \rightarrow \text{MOD no of a counter or Number of States}$$

Application: Frequency Division:

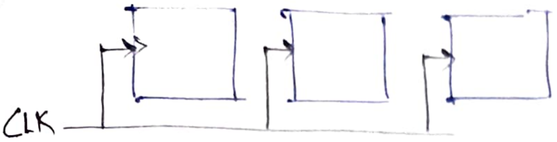
MOD-N counter is also called as Divided by N-counter







Synchronous counter



1) All FF's o/p changes at the same time.

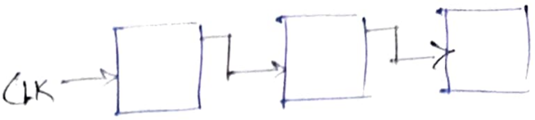
2) If one FF propagation delay = t_{pd}
"n" FF propagation delay = t_{pd}

Propagation Delay

$T_{pdsyn} = T_{FF} + T_{combinational}$

$T_{pdasyn} = N \times T_{FF} + T_{combinational}$

Asynchronous counter



1) Here All FF's o/p changes at different times.

2) If one FF propagation delay = t_{pd}
then "n" FF propagation delay = $n \times t_{pd}$

* frequency in n-bit Asynchronous counter

$$f = \frac{1}{n \times t_{pd}}$$

f = clock frequency
t_{pd} = Propagation delay of each flipflop.
n = number of flipflops.

Summary

Counter Name	Repeats same state after	MOD (N)	Output frequency $\left(f_o = \frac{f_i}{N}\right)$
1) n-bit Asynchronous [For both up and down]	2^n clock pulses	$N = 2^n$	$f_o = \frac{f_i}{2^n}$
2) n-bit Ring counter	n clock pulses	$N = n$	$f_o = \frac{f_i}{n}$
3) n-bit johnson counter	2n clock pulses	$N = 2n$	$f_o = \frac{f_i}{2n}$

* Procedure to solve Counter Problems

Step 1: Write the given Flipflop truth table

Step 2: Write the relations from the given circuit

Step 3: Write Input & output relation table