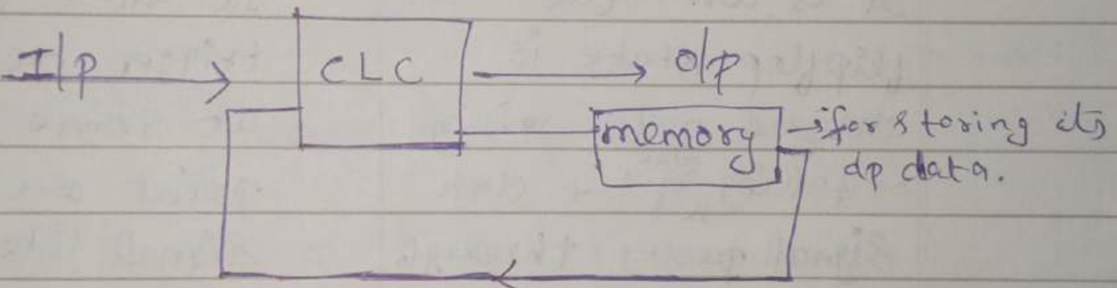
Module - II

sequencial Logical circuit

* output of a sequencial l. circuit not only depend on it's present I/p bt also depend on it's ~~prsnt I/p~~ o/p state / previous state

(i.e) CLC + M/y = SLC.
(combi Log. circuit) (memory)

*



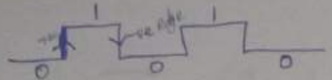
* There is a feedback connection from ~~memory~~ o/p to I/p.

~~* 2 categories of SLC:~~

* sequential circuit are classified into 2 categories -

1) Asy.chronous S.C =
  * A S.circuit whose behaviour depends upon sequence in which the I/p signal change
  → A.S.C. [no common clock signal].

2) synchoronous S.C =

Clock Signal →


A S.C whose behaviour can be defined fom the knowledge of the signal at discrete instance of time.
(Common clock signal [illegible])

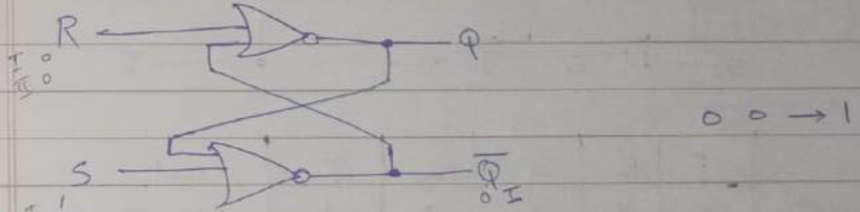
* 2 types of sequencial logic circuit =

a) Edge triggered	(b) level triggered
It is, whn the flipflop state is changed as the rising or falling edge of the clock signal passes through a threshold voltage. [edge-[illegible] work [illegible] (+ve/-ve)]	It allow you to trigger measurmnt at some defined point on the input signal like whn the signal crosses 0 volt or whn it reaches the mid point of the +ve/-ve peak amplitude [0 level / 1 level [illegible] work [illegible]].

→ Latches = (eg for Asynchronous S.C)
Latches & flipflop are bistable

bi-stable {0 - off state, 1 - on state}

memory elemnts (0 or 1) that can store 1 bit of data for as long as the device is powered.
Types :
1) RS Latches = (R→Reset & S→set)
[It is repsnted by NAND gate (active high)]



	R	S	Q	$\overline{Q}$	Comment
I	0	1	1	0	set
II	0	0	1	0	~~Reset~~ nochange
	1	0	0	1	Reset
	1	1	X	X	~~Reset~~ Avoided

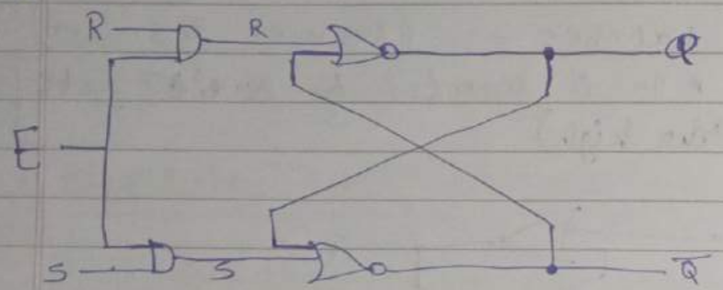
* Raise condi~ occurs whn both I/p of RS Latches. is 1.
* R & S (Reset & set) is avoided state whn it is R = 1 & S = 1.
* Here, the I/p may varied, their o/p will also be varied. Hence it is asynchronous

enable { ON-1
OFF-0

→ RS Latches with enable :

* Enable means whn you are giving an ON & off switch to RS Latches.

*

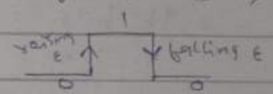


E-1 అయితే circuit work అవుతుంది

E	R	S	Q	$\overline{Q}$	Cmnt
0	x	x	0	1	No change
0	x	x	1	0	No Change.
1	0	1	1	0	set
1	0	0	1	0	No Change
1	1	0	0	1	Reset
1	0	0	0	1	No Change
1	1	1	x	x	Avoided

→ Flip flops :

* It is synchronous logical circuit, bcz it contain clock signal

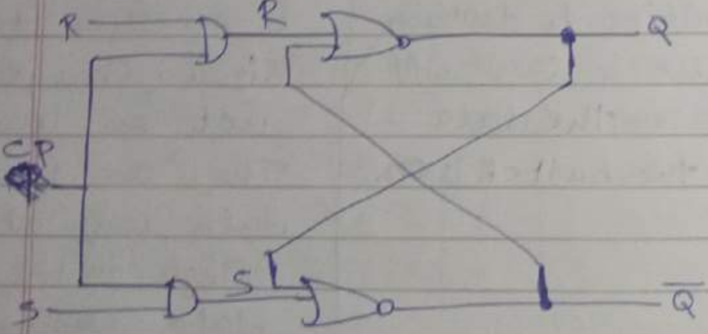
* Latches	Flipflops
* They are asynchronous	They are synchronous.
* It works on the basis of Enable I/p.	It works on basis of clock pulse.
* A Latche sample the I/p continously ~~changes~~ whnevr it is enabled	A flipflop samples the I/ps only at a clock evnt (raising edge / falling edge) 
* Latches are level sensitive (i.e) Latch is sensitive to duration of pulse & can snd / receive the data whn the switch is OFF.	They are edge sensitive (i.e) flipflop is sensitive to signal change & not on level. They can trasfer data only at a single instance & data cannot be changed until nxt signal change.
* Latches also continously checks all its I/p & correspondnly	A flipflop continously checks its I/ps & correspondly changes

changes its o/p, independent of the time determined by clock signal	its o/p only at times determined by clocking signal.

→ ~~Clocked version of RS Latches =~~ ~~of RS Latches~~

Clocked version of RS Latches =

* It has 3 I/p:
  R, S, Cp (clock pulse)
* It has 2 O/p:
  Q & $\overline{Q}$
* Diagram using NOR gate:



* Truth table:
  Q_t → prsnt o/p state of particular flipflop
  S → set & R → Reset.
  $Q_{(t+1)}$ → Q_t + R + S

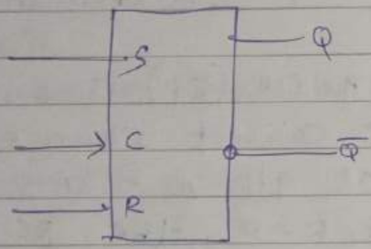
Q_t	S	R	$Q_{(t+1)}$
0	0	0	0 (no change)
0	0	1	0 (reset)
0	1	0	1 (set state)
0	1	1	Indeterminate
1	0	0	1 (No change)
1	0	1	0 (reset)
1	1	0	1 (set state)
1	1	1	Indeterminate

* If the clock pulse (CP) is 0, the o/p of the circuit cannot change irrespective of values at I/p R & S.
* when S=1 & R=0, the RS flipflop moves on to set state ($Q=1$ & $\overline{Q}=0$) on the triggering edge of the CP.
* when S=0 & R=1, then RS flipflop moves on to reset state ($Q=0$ & $\overline{Q}=1$) on the triggering edge of the CP.
* when both S & R at 0, the o/p doesnot change from its previous state.
* An invalid condi- exist when both S & R at 1.
* K map —

Q_t \ SR	00 $\bar{S}\bar{R}$	01 $\bar{S}R$	11 SR	10 $S\bar{R}$
0 $\bar{Q_t}$			X	1
1 Q_t	1		X	1

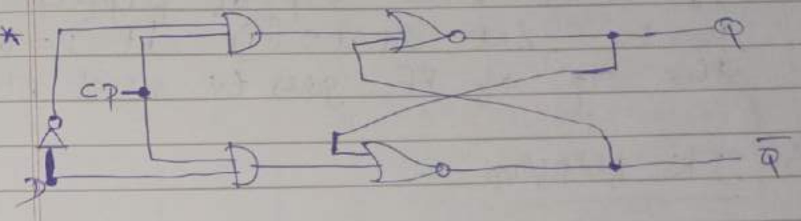
~~logical~~ symbol of clocked

logical symbol of flipflop:



~~Tooo~~ D Flipflop =

* D (data) flipflop is a single I/p version of RS flipflop.
* with clocked RS flipflop, the I/p combination R = S = 1 is forbidden. [S=1 & R=1 - [illegible] solve [illegible] flipflop)
* It can be ensured that $S \neq R$. by connection an invertor b/w R & S I/ps. Now the flipflop has only 1 I/p. This I/p is → ~~D flt~~ D input.

* 

S = R = 1

*

Q_t	D	$Q(t+1)$
0	0	0 (reset)
0	1	1 (set)
1	0	0 (reset)
1	1	1 (set)

*

Q_t \ D	0 $\bar{D}$	1 D
0 $\bar{Q}$		1
1 Q_t		1

$\therefore Q_{t+1} = D$

* If the clock pulse is 0, the o/p of circuit cannot change irrespective of the values at ~~the~~ D I/p.

FF → flipflop

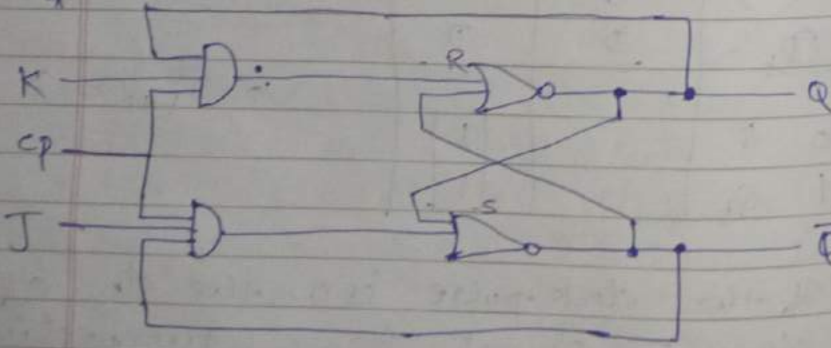
DFF	JK FF
1 I/P	2 I/P

* D I/p is sample ~~doing~~ during the occurance of a clock transition from 0 to 1.
* If D = 1, the o/p of flipflop goes to set state (1), but D = 0 the o/p of FF goes to reset state

II JK flipflop =

* It is a refinement of RS FF in that the avoided state of FF is defined in the JK FF.
* I/p J & K behave like I/p R & S, to set & reset the FF.
In JK FF the letter J is cause set & the letter K is for clear/reset.

*



Q	J	K	Q(k+1)
0	0	0	0 (no change)
0	0	1	0 (reset)
0	1	0	1 (set)
0	1	1	1 (complmnt).
1	0	0	1 (no change)
1	0	1	0 (reset)
1	1	0	1 (set)
1	1	1	0 (Complemnt)

* kmap for JK FF =

Q_t \ JK	00 $\overline{J}\overline{K}$	01 $\overline{J}K$	11 JK	10 $J\overline{K}$
0 $\overline{Q}_t$			1	1
1 Q_t	1			1

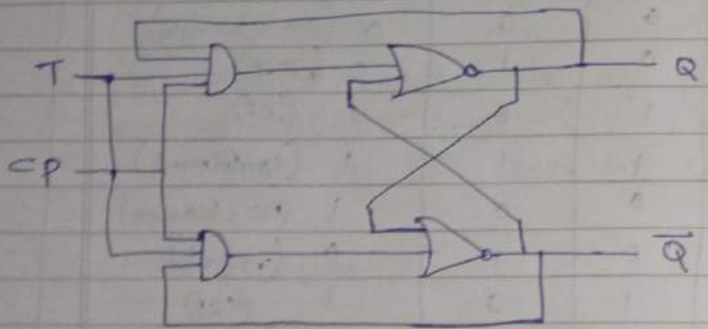
$Q_{(k+1)}$ ൽ 1 വരുന്നത് നോക്കി [illegible] 1 ഇടുക.

$$\therefore Q_{t+1} = \underbrace{\overline{Q}_t J}_{J\,U\ \text{common}} + \overline{K} Q_t$$

III T Flipflop =

* It is a single I/p version of JK FF.
* It is obtained from the JK FF, if

both I/ps are tied togthr as shown -



Q_t	T	Q_{t+1}
0	0	0 (no change)
0	1	1 (complmnt)
1	0	1 (no change)
1	1	1 (complmnt)

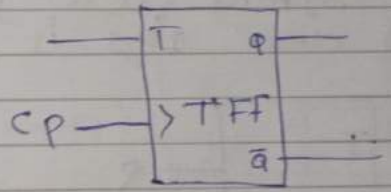
* The designation 'T' comes frm the ability of the FF. to 'toggle' / change state.
* Regardless of prsnt state of the FF, it assumes the complemnt state when the CP occurs, by I/p T is at logic 1

* k map -

$Q_t \backslash T$	0 $\overline{T}$	1 T
0 $\overline{Q_t}$		$\boxed{1}$
1 Q_t	$\boxed{1}$	-

$$\therefore, Q_{t+1} = \overline{Q_t}\,T + Q_t\,\overline{T} = T \oplus Q$$

logic symbol -



→ Asynchronous set & clear I/ps =

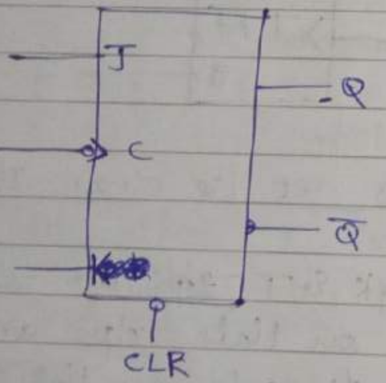
* The S/R, D, JK & T. I/ps are → Synchronous I/ps, bcz data on this I/ps are transfered to the FF o/p only on the trigering edge of CP. (i.e) the data are transfered synchronously with clock.

[change anusarich o/p → change nrob. → Syn &
I/p changes, at same time o/p also changes → Asyn.]

* most integrated circuit FF also hue

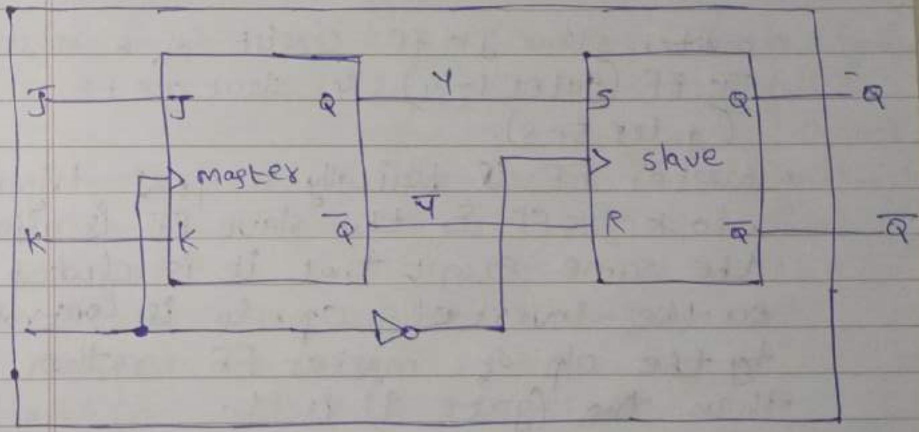
Asynchronous I/p. This are I/p that affect the state of the FF independent of the clock. They are normally labelled preset (PRE) & clear (CLR) or direct set (SD) & direct reset (RD) by some manufacture.

* An active level on the preset I/p will set the FF. & an active level on clear ~~to~~ I/p will reset it.
* A logical symbol for a JK FF with preset & clear I/ps is shown blow-



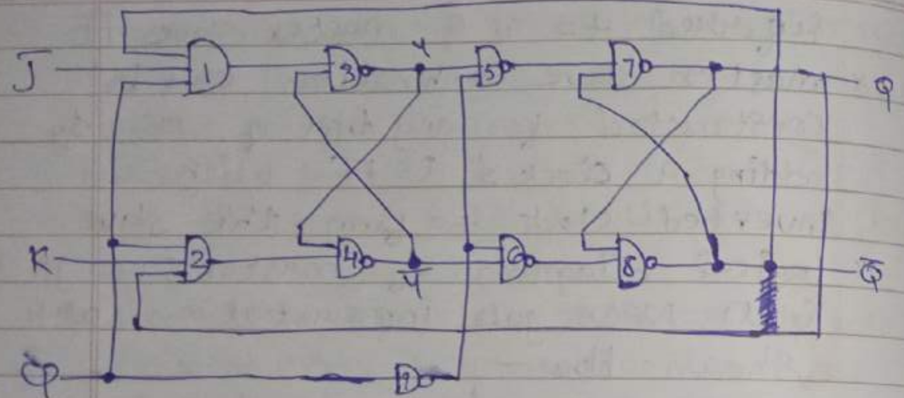
→ Master~~'s~~ slave FF =

* It is constructed from 2 seperate FF, 1 FF serves as a master & othr as a slave & overall circuit is referred to as a master slave FF.
* master slave combination can be constructed for any type of FF by adding a clocked RS FF with an inverted clock to form the slave.
* logical diagram of master slave JK FF & its NAND gate implementation, are shown blow-



• NAND gate implementation-

master wrk നടക്കുമ്പോൾ slave wrk ആകുന്നില്ല.



* master slave JK FF consist of a master JK FF (gates 1–4) & slave JK FF. (gates 5–8).
* master FF is basically a pulse trigered clock JK FF & the slave FF is also the same except that it is clocked on the inverted Cp. & is controlled by the o/p of master FF rather than (gate 9) is 1.
* when the CP is 0, the o/p of the invertor (gate 9) is 1, since the clock I/p of slave is 1 the slave FF is enabled & o/p 'Q' = y & $\bar{Q} = \bar{y}$
The master FF is disabled since cp=0
* ~~[illegible]~~ when the CP = 1, the info. at the External I/p J & K is transmitted to the master FF. & is held there until the -ve edge of cp occures. ~~at this~~

⇒ <u>Counters = (c)</u>

* A counter is a device which stores (sometimes displays) the no. of times a particular event / process has occured, often in relationship to a clock signal.
* (c) are used in digital electronic device. for counting purpose, they can count specific event happening in the circuit.
* It is a grp of FF with a clock signal
* Here, state of FF changes with respect to Cp.
* Major categories of (C):

1) <u>Asynchronous (c)</u> & (2) Synchronous (C).

These counters which donot operate on simultanous clock.

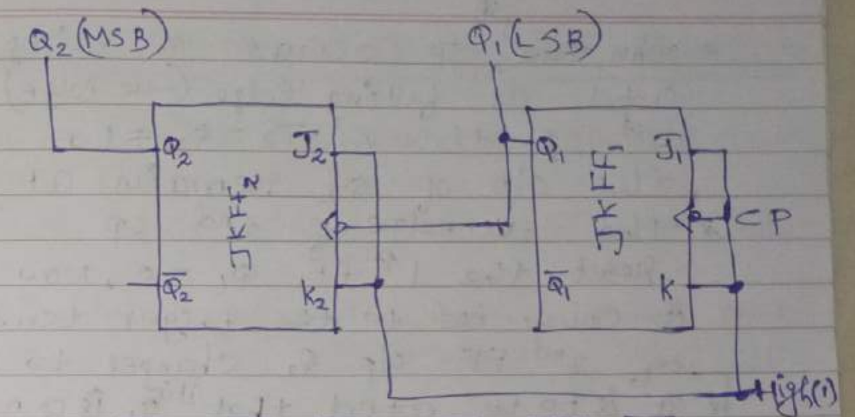
Here, only the 1st FF is externally clock using cp, while the I/p for the successive FF will be the o/p from a previous FF.

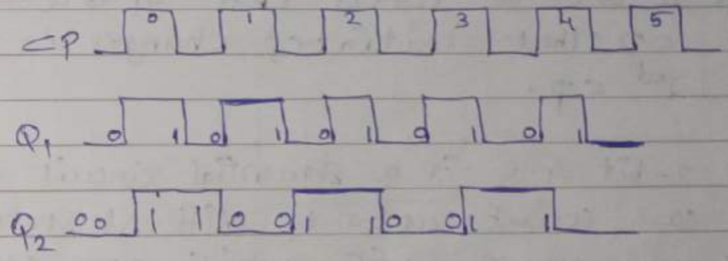
* This means that only a single cp is not driving all the FF in the arrangmnt of (C)

I A 2-bit Asynchronous counter :

* A binary asynchronous counter consist of a series connections of complementing FF. (eg → JKFF with J=K=1 or T FF) with o/p of each FF connected to the cp I/p of the higher order FF.
* The FF holding the LSB receives the incoming count pulse.
* Logical diagram of 2 bit A.C :-

Q2 ← | → Q1
(MSB) (LSB)





* A small circle in the CP I/p indicates that the FF complements during a -ve going transition / when the o/p to which it is connected goes from 1 to 0.
* Let initially $Q_1 = Q_2 = 0$, it is seen that $J_1 = K_1 = 1$, hence -ve edge of 1st CP of edge CP sets the o/p of the 1st FF to 1.
It is seen that $J_2 = K_2 = 1$

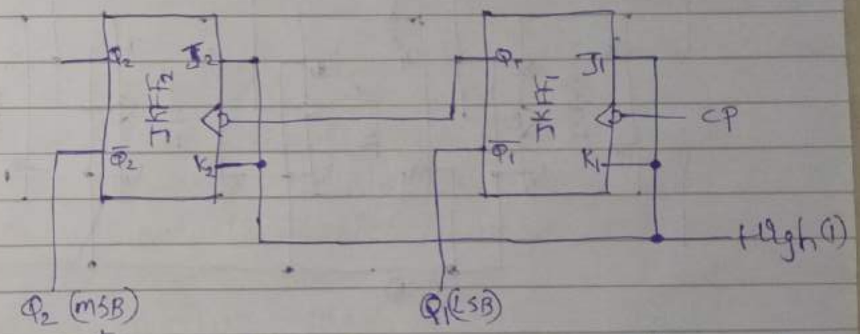
* whn the 1st CP occurs Q_1 raises frm 0 to 1 at falling edge (-ve edge) of the 1st cp. though $J_2 = K_2 = 1$, the o/p of Q_2 remain at 0
* The -ve edge of 2nd CP reset the 1st FF $Q_1 = 0$, Now as Q_1 is connected to the triger terminal of 2nd FF o/p Q_2 changes to 1.
* It is to be noted that its Q_1 & not the cp that initiates changes in the 2nd cp.

①
* 2-bit A.C is a sequential circuit that can count upto 4 diffrnt states.
* made up of 2 FF, which are connected togethr in a chain.
* CP is applied to 1st FF & o/p of 1st FF is used as the I/p to 2nd FF.
* 4 States are —

00 → Both FF are reset
01 → 1st FF is set & 2nd is R
10 → 1st FF is R & 2nd is S
11 → Both FF are S.

II ⇛ 2-Bit Asynchronous Binary Down Counter =

* It is a digital circuit that counts down frm 11 to 00 in Binary.
* made up of 2 FF each of which can hve 2 states.
* 1st FF is clocked by an external clock signal & 2nd FF is clocked by the o/p of 1st FF.
* circuit wrk as follows—



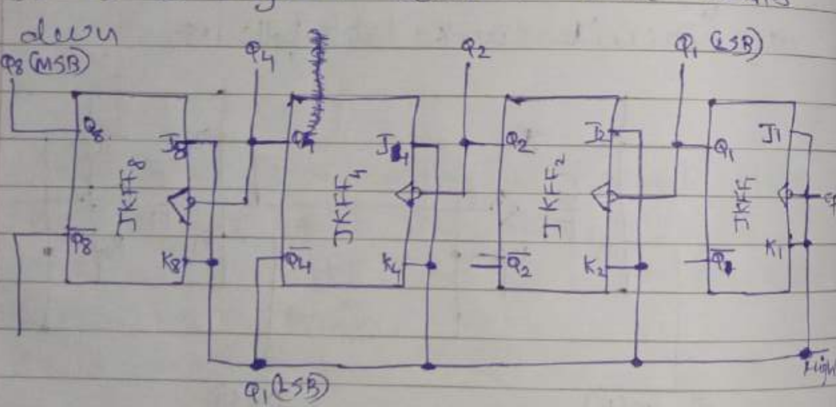
1) whn 1st clock signal is high, the 1st FF is set to 1.
2) whn clock signal goes low, the 1st FF is reset to 0.
3) o/p of 1st FF is then used to clocked the 2nd FF.
4) 2nd FF will change state if and only if

the o/p of 1st FF is 0.

5) This process continous until the 2nd FF ~~reaches~~ reaches the state 00. At this point the counter will reset & start counting $\overset{\text{down}}{\wedge}$ from 11 again.

III 4-bit Asynchronous counter =

* It is a digital circuit that counts down



- It is a sequencial circuit that uses 4 DFF to count from 0 to 15.
- Clock I/p of all FF are cascaded, & D I/p of each FF is connected to the state. to the o/p of FF.
- when clock signal is high, 1st FF changes its state.
- o/p of 1st FF is then connected to the D I/p of 2nd FF
- when the clock signal is high again the 2nd FF changes its state.
- This process repeat for all FF, so the countr counts from 0 to 15

IV Asynchronous Decade counter =
(to represent 10 no — (0–9))

* It is a digital circuit that counts from 0–9 in binary & then resets to 0

* made up of 4 FF, each of which can hue 2 states — 0 or 1

* The clock inputs of all FF are cascaded & D I/p of each FF is connected to logic 1 that means the FF will toggle (0 → 1 → 0 & vise versa) at each active edge of clock signal.

* clock I/p is connected to the 1st FF. The other FF in countr receives the clock signal I/p from Q o/p of previous FF rather than Q' o/p.

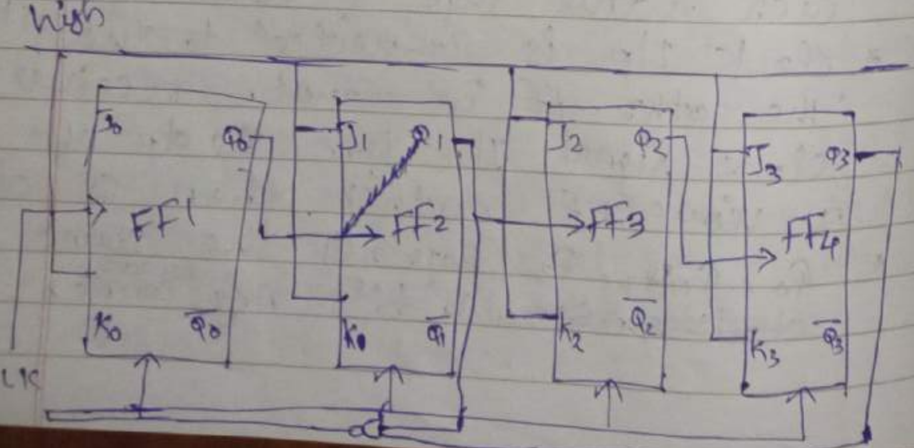
* Q_0, Q_1, Q_2, Q_3 represents the count of ~~clock pulse~~. 4 bits Asyn $\overset{\text{decade}}{\wedge}$ countr

...

CLK → Clock Signal

	8	4	2	1
	Q0	Q1	Q2	Q3
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

As you can see the countr starts in the state 0000 & then counts upto 1001. At this point the countr will reset to 0000 & start counting again.



→ Prbms of Asyn counters =

1) Propagation Delay of each FF in a Asy counter adds up, so overall speed of the counter is limited.

2) Glitch prbms :
The propagation delay can also leads to glitch prbm, which are brief pulses of the incrt o/p signals.

3) Limited clock freq :
Propogation delay & Glitch prbms limits the max clock freq that an Asy counter can operate at.
This can be a prbm in applin where high speed is required.

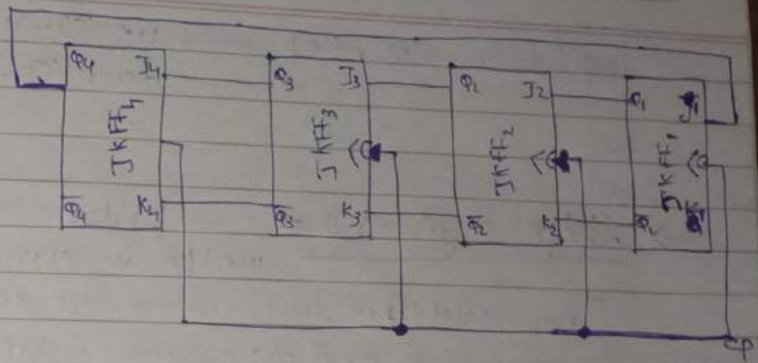
4) Race Condi = (2 o/p for 1 [illegible] — R. condi)
It is a situation where 2/more FF in a counter are trying to change state at the same time (S = R = 1).
So ~~at~~ this can cause the counter to count incrtly.

⇒ Synchronous Countr =

* It is a type of counter in which all of the FF change state at the same time, in response to a common clock signal.
* Typically faster & more reliable than Asy counters.
This is bcz the propogation delay of clock signal is shared by all of the FF, so there is no cumilative delay :
* Additionally the prbm of glitches & race condi is lowr in Syn countr.
* Adv : Dis adv :
reliable, faster, Highcost, complexity
versatility.

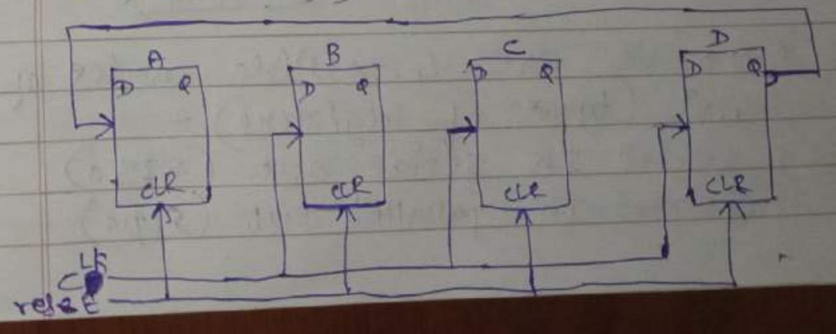
I Ring counter : [1ast FF 1st FF-മായി നേരിട്ടും connected ആയിരിക്കും]

* It is a type of countr Composed of FF connected into a shift register with the o/p of last FF fed to the I/p of the 1st, making a ~~[illegible]~~ circular / ring structure



* There are 2 types of Ring countrs :

a) Johnson Counter :
[modified version of Ring co.]
* Also → modified ring counter
* Designed with a grp of FF where the inverted o/p from the last FF is connected to the I/p of 1st FF.
* Generally it is implemnted by using D FF / JKFF.
* Also → An inverse feedback counter / twisted ring counter.



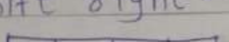
up/down. counter. ⟶ up (C) → 00 – 11
down (C) → 11 → 00

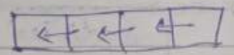
→ Shift Registers = Registrs used for temporary data storag

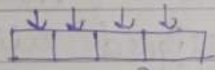
ഒരു registerൽ store ചെയ്താൽ left-ലേക്ക് move ചെയ്യുന്നതിനെ ഷിഫ്റ്റ് opr ചെയ്യുന്നു ie for right mvmt

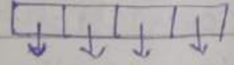
6 mvmnts :

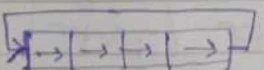
1) serial shift right =

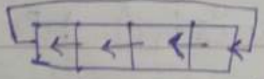

data പുറത്ത് പോകും.

2) serial shift left = 

3) parallel shift In = 
(ഒരുമിച്ച് data fill ആകും)

4) parallel shift out = 

5) Ratate right = 

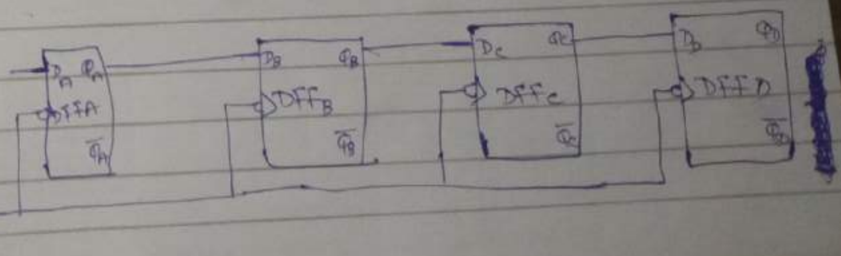
6) Rotate left = 

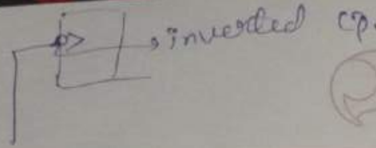
* There are 4 possible modes of opr (types of registers) –
1) serial In serial out (SISO)
2) serial In parallel out (SIPO)
3) parallel In parallel out (PIPO)
4) parallel In serial out (PISO)

* Register that permits the movemnt of data from stage to stage within the registr ⟶ shift register.

* 4 possible modes of opr –
1) SISO :- queue പോലെ serial ആയി ~~[illegible]~~ വരും.
" " " പുറത്ത് പോകും.
2) PIPO :- ഒരുമിച്ച് വരും ഒരുമിച്ച് പോകും.
3) SIPO :- queue പോലെ കയറും then ഒരുമിച്ച് പുറത്ത് പോകും.

I SISO :
This kind of shift registers ~~except~~ accept data serially that is, 1 bit at a time on a single time.
It produces the stored info on it's o/p also in serial form.
A 4-bit SISO register implemnted with DFF –



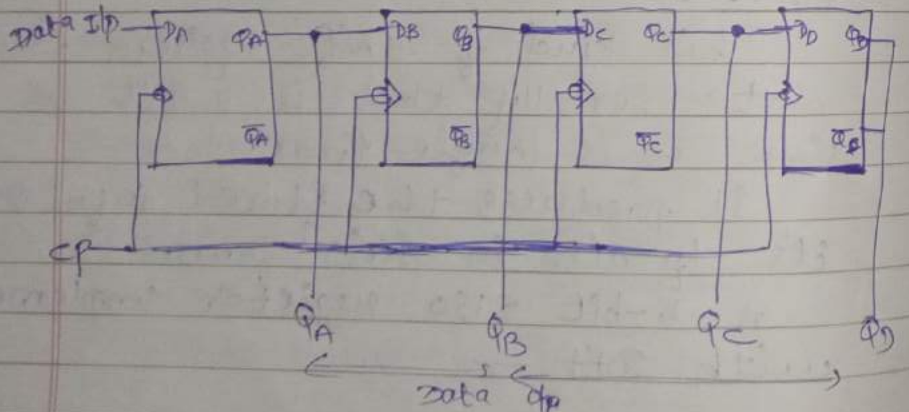


2) SIPO :

This kind of registrs also accepts data serially (i.e) 1 bit at a time on a single line.

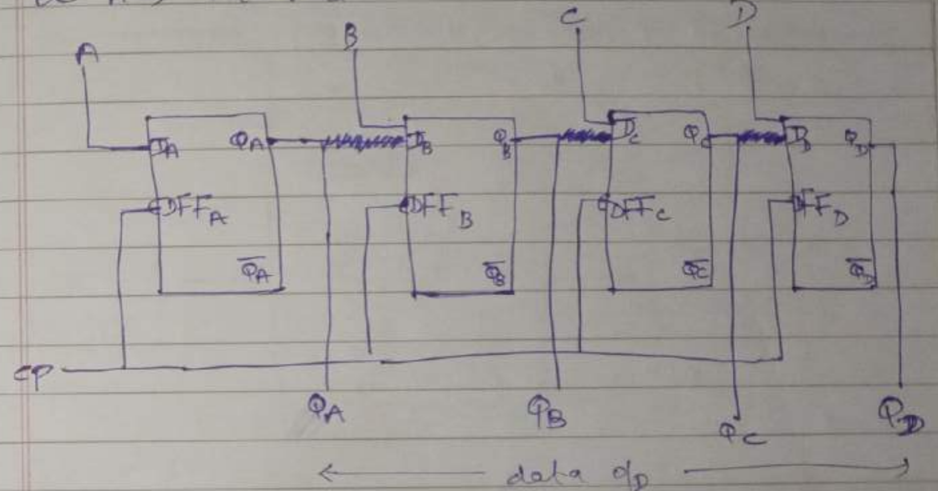
But the o/p is availalele in parallel, (i.e) the o/p is each stage is availalele.

once the data is stored, each bit appears on it's respective outline & all bits are availalele simultanously.



3) PIPO :

Here is 1 in which the data are loaded in parallel & the register can simultanously o/p all the bits it has stored.



Data I/p → A, B, C, D.

4) PISO :

For a registr with parallel data I/p the bits are entered simultanoasly into their respective stages on parallel lines rathr than on a bit by bit basis on 1 line as with serial I/p.

once the data are completely stored in the registr, the serial

out is executed inthe same manner
as in PISO shift register