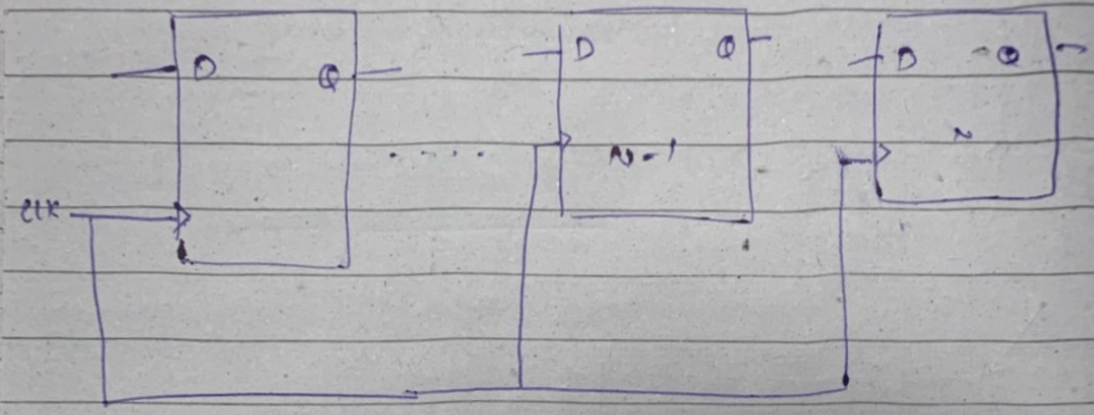
chapter:

It is a group of flip flops which stores n bit of data

Registers.

① as we know a single flipflop stores only 1 bit information. So to store multiple bit information need multiple flipflops.

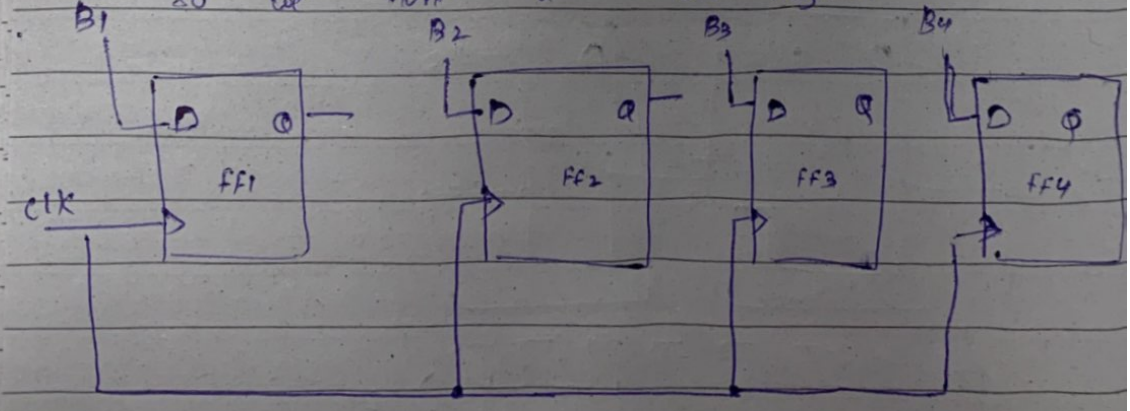
↳ n-bit information = n-flipflops



② Each flipflop shares a common clock.

n = 4 bit

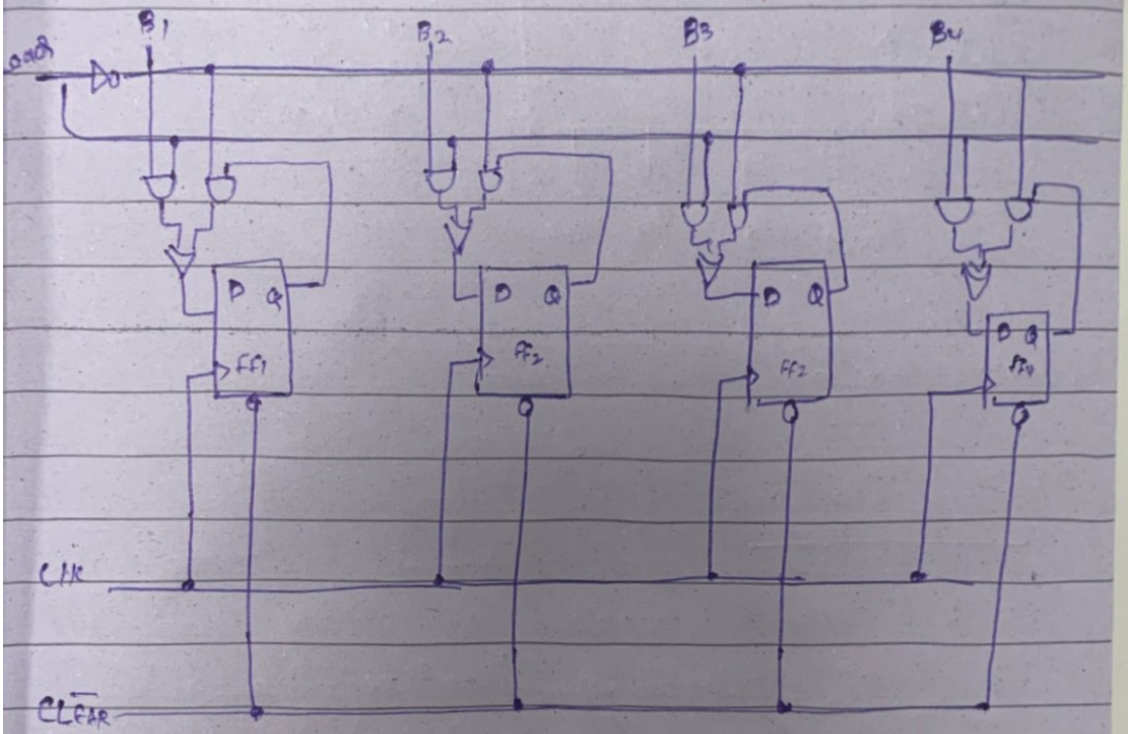
So we have a 4 bit register



① in the synchronous systems, it is preferred that all the synchronous elements receive the clk at the same time.

② Typically in registers D-flipflops are used as memory element. b/c D-flipflop has only one input and whatever input we apply at the input side that same will be stored on the flipflop at the clk transition.

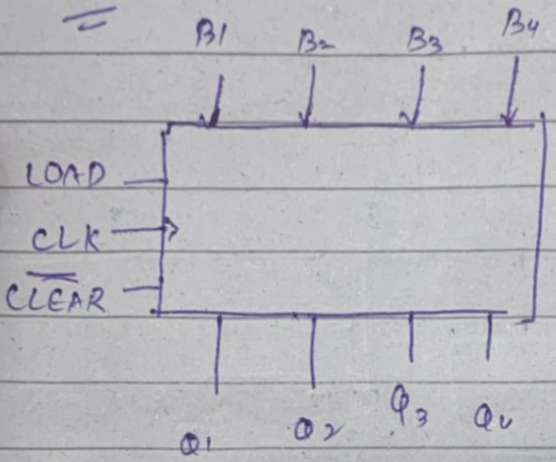
4 bit register..



③ whenever you just want to load new data, then first of all make the data available on the data bus, and then after just enable this load signal and when this load is low then the register will maintain its previous data.

and here the load signal is a synchronous input meaning that once the load signal is high, then the data will get loaded into the register only at the clock transition.

Block diagram



in this type of register we can store the data and retrieve the data.

a) another type :-

Shift Registers.

↳ A register capable of shifting data bits to the neighbouring flip-flops in a selected direction is called shift registers.

↳ The output of one flipflop is connected to the input of the next flipflop

① in shift registers data can be loaded in two ways

① Serially ② parallel.

① Serially → at every clk pulse a new bit will get loaded into the register. So if we have a 4 bit register then in the 4 clock pulse the data will be loaded serially.

② Parallel → in parallel loads all the bit of the register, will get loaded in a single clock pulse

② Similarly there are too ways to retrive the data from the shift register.

① Serial out

↓

in every clk pulse we can access only 1 bit for n-bit data to retrive we need n-clk pulses in serial out

② parallel out

↳ we can access all the bits in a single clock pulse.

④ shift registers are used in
① for providing time delay
② serial to parallel conversion
③ parallel & serial conversion
④ Arithmetic operations (Multiplications & Division)

⑤ on the basis of I/O there are four types of shift register.

Types

Serial IN Serial OUT	Serial IN Parallel OUT	Parallel IN Serial OUT	Parallel IN Parallel OUT
(SISO)	(SIPO)	(PISO)	(PIPO)

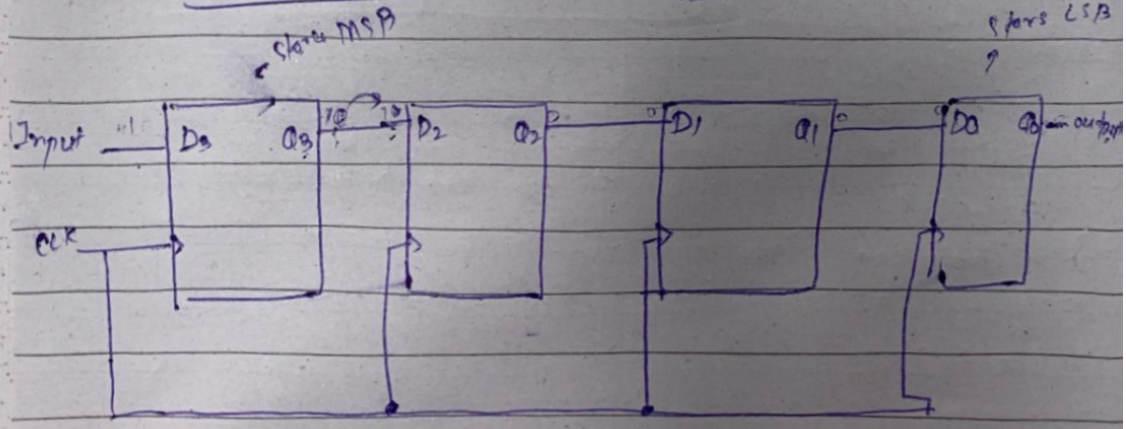
Serial IN Serial OUT (SISO)

Type :- dips on direction in which data is shifted

① ~~Shift~~ right shift register → data entered left, shift to right

② shift left shift register → data entered right, shift to left.

Shift right shift register



we want to store MSB 1010 LSB

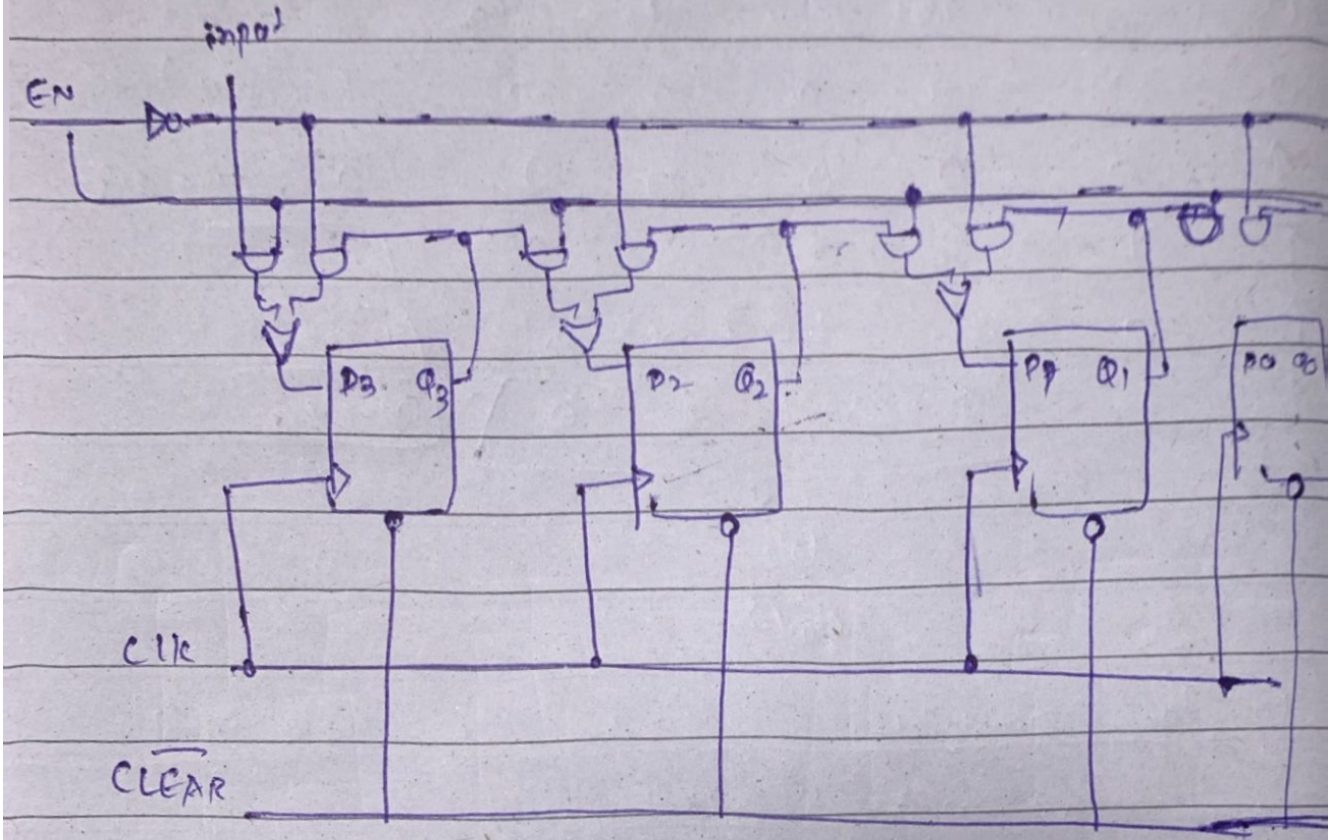
clk	input	Q_3	Q_2	Q_1	Q_0	
0	X	0	0	0	0	(initially)
↑	0	0	0	0	0	discarded.
↑	1	1	0	0	0	
↑	0	0	1	0	0	
↑	1	1	0	1	0	

data stored.

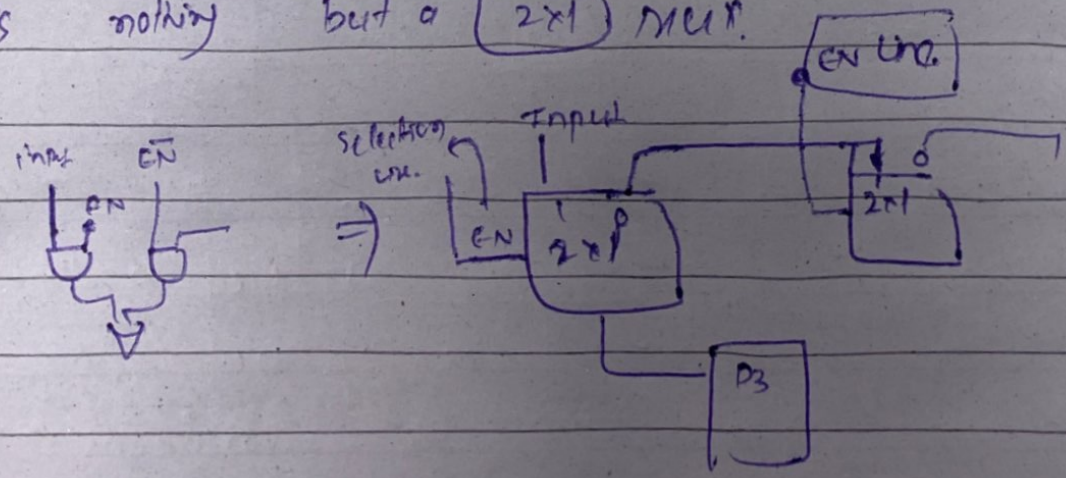
② To completely retrive the data stored we also need 4 clk pulses.

③ once we move the data inside the shift register, then we need to make sure that this input line will get disabled, bcz if the input line is not disabled, then the random data will be available at input side, and due to that in the subsequent clock edges, any arbitrary value will get stored inside the shift register. So to avoid that we need to control input line.

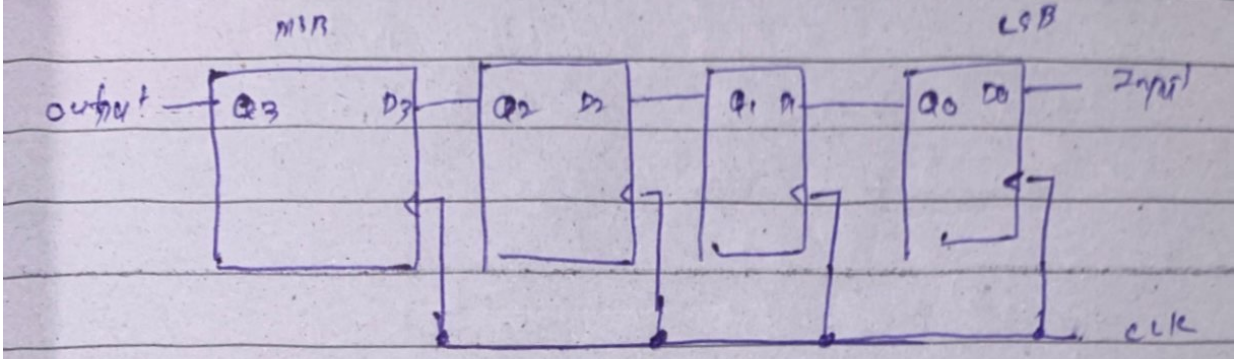
(*) EN → High circuit will work at the CLK transitions → when EN is low → then the circuit will retain its previous state



This two AND & ONE OR gate combination is nothing but a 2×1 MUX.



shift left shift Register.



store MSB 1100 LSB

CLK	input	Q0	Q1	Q2	Q3
0	x	0	0	0	0
↑	1	1	0	0	0
↑	1	1	1	0	0
↑	0	0	1	1	0
↑	0	0	0	1	1

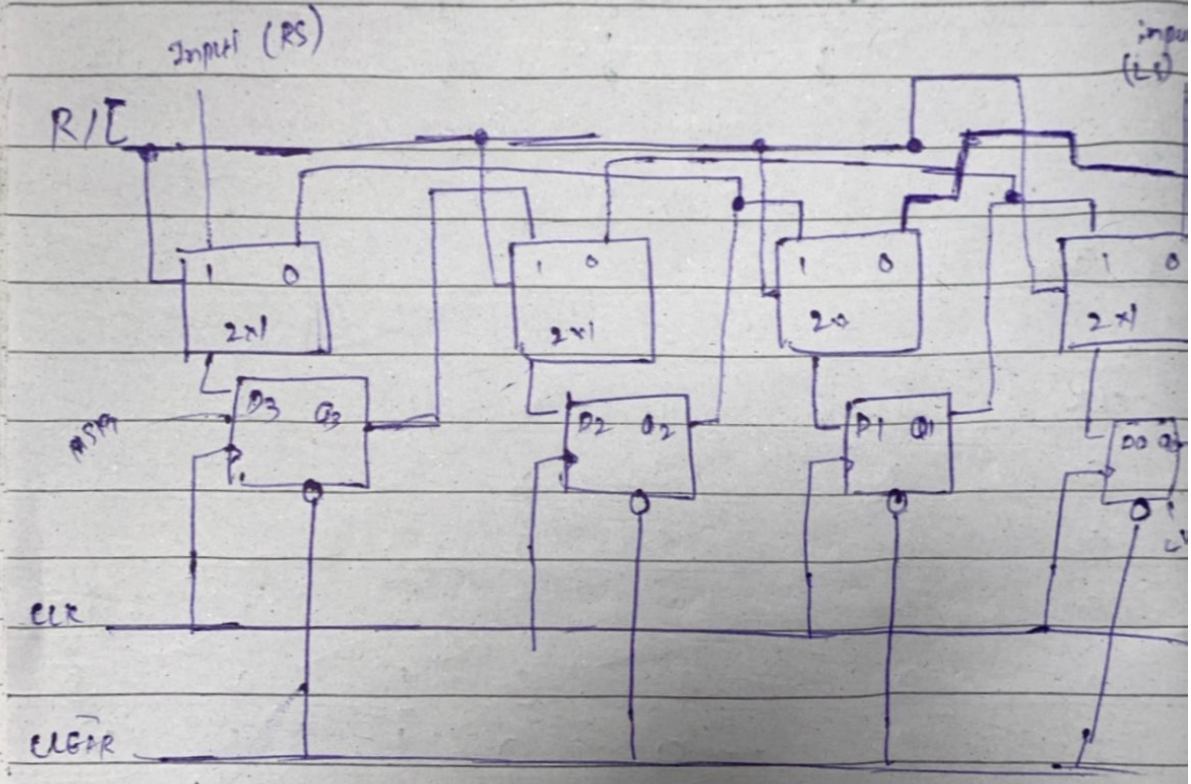
discarded

0 0 1 1 ⇒ 1100

LSB MSB

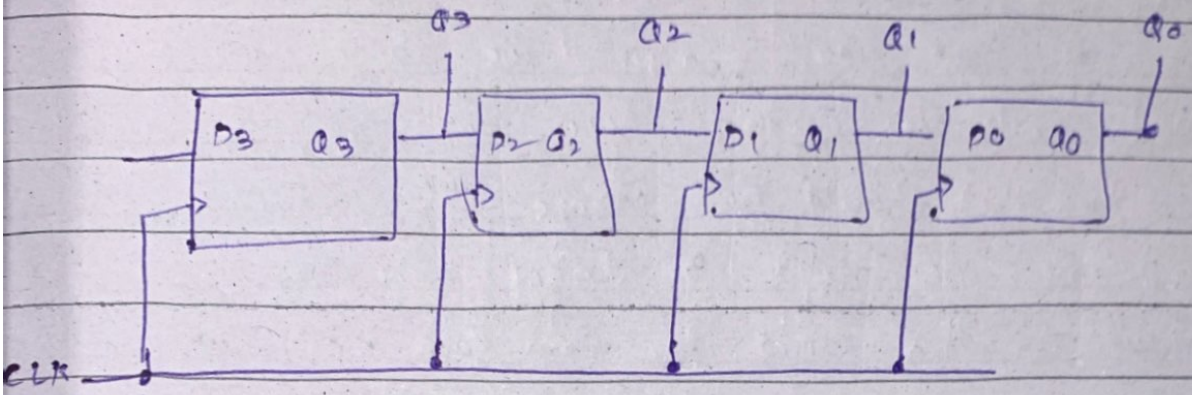
stored

Bidirectional shift registers — shifts in both direction either left or right in SISO

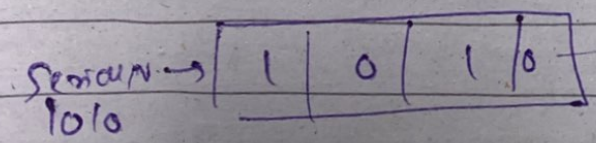


When R/L line = 1 then shift right operation is performed and when R/L line = 0 then shift left operation is performed.

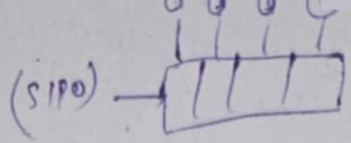
Serial IN Parallel OUT (SIPO)



① for a n-bit SIPO register, we will require N-clock cycle to load data into the register



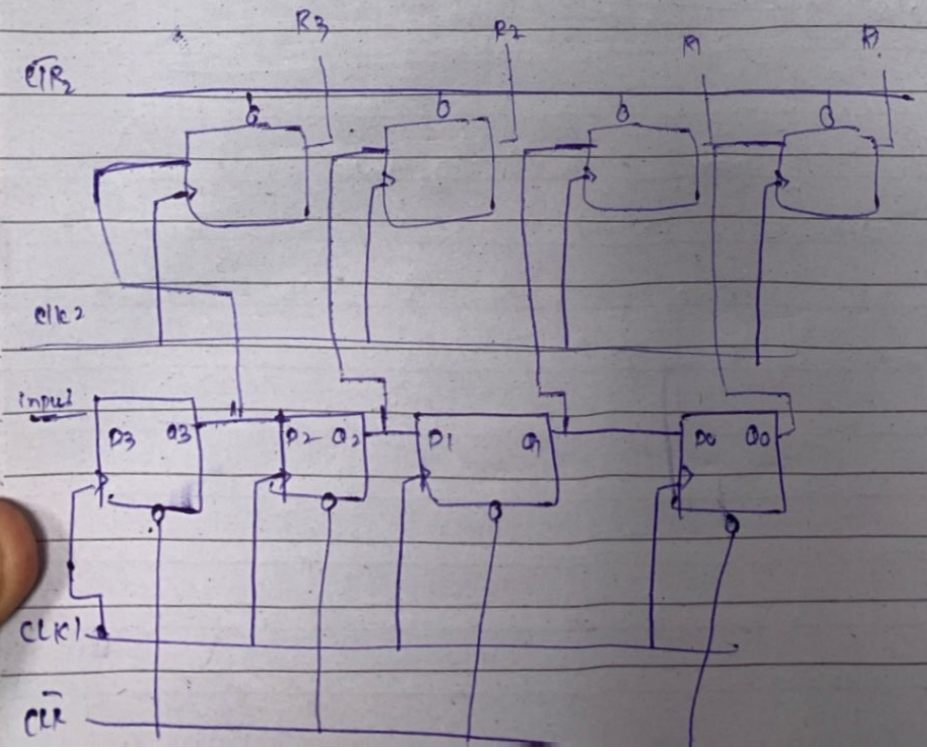
② Now many time, it is required that once the data set shifted into the register, it should not change, until the new data set loaded into the shift register.



③ commercially built in SIPO register also have an additional register built

④ for eg: itch say currently 1010 is stored inside the shift register. and let say we want to store 0011, so until we shift this data inside the shift register, these four outputs (1,2,3,4 pin value) should not change b/c these four pins are used for driving some other circuit or it is used for some data manipulation so this means during the shifting operation this data should not change (1,2,3,4)

but this will not happen in the upper design b/c after four clk pulse the new data will get stored and the content of (1,2,3,4 pins) will change, so to avoid that we need to store this data into some other register.



So first when we store our data into the lower register then we just need to apply [clk] signal then the lower register data will be stored in the upper one and the lower register will be available for the new data to be entered.

Parallel IN Serial OUT (PISO).

load data — 1 CLK Pulse

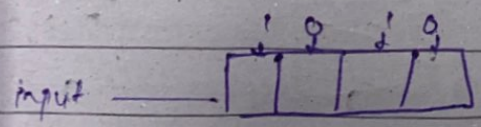
completely shifted → 4 - CLK Pulses retrieve

PISO register also has one serial input ↓

if the input pin is not connected to anyother pin then any random data will get stored inside the shift register so typically this pin is connected to logic "0" or "1"

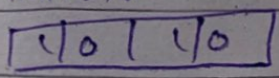
for e.g.

we want to store 1010.

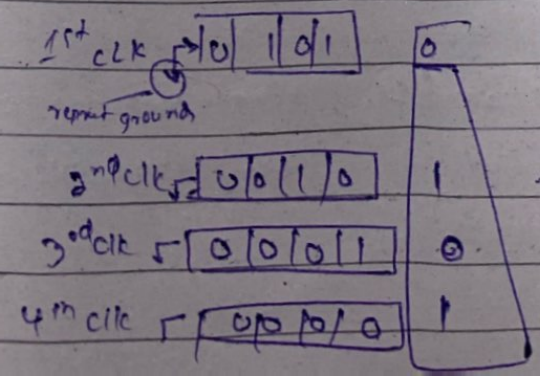


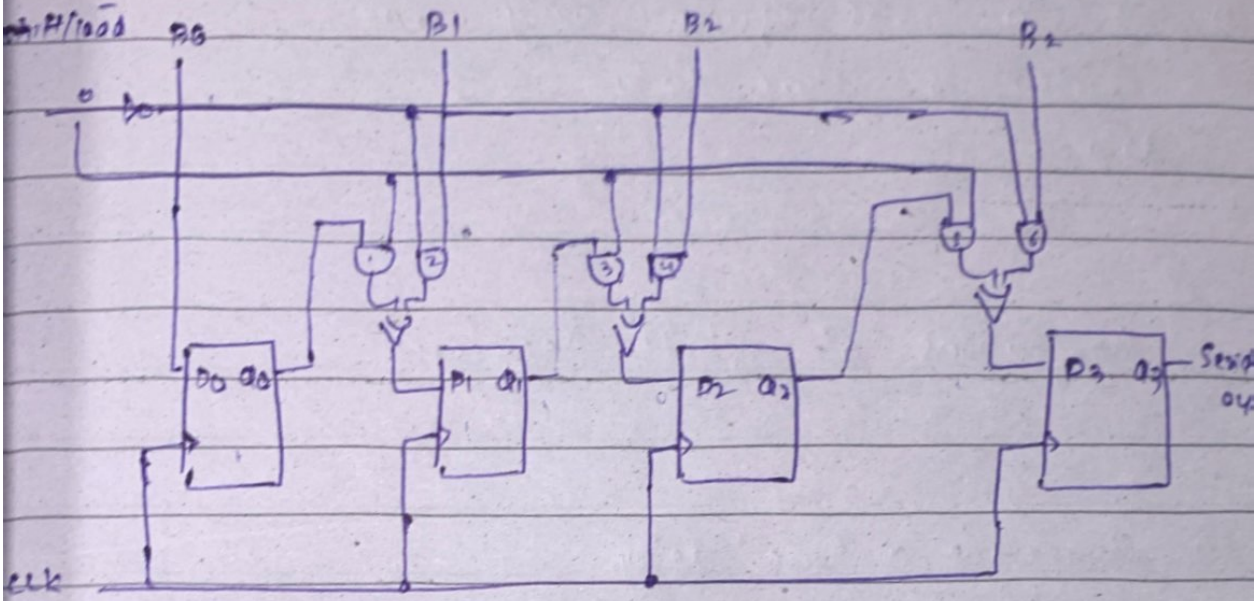
1st CLK transition — data stored.

load



Shift 1st CLK





two mode

① load mode

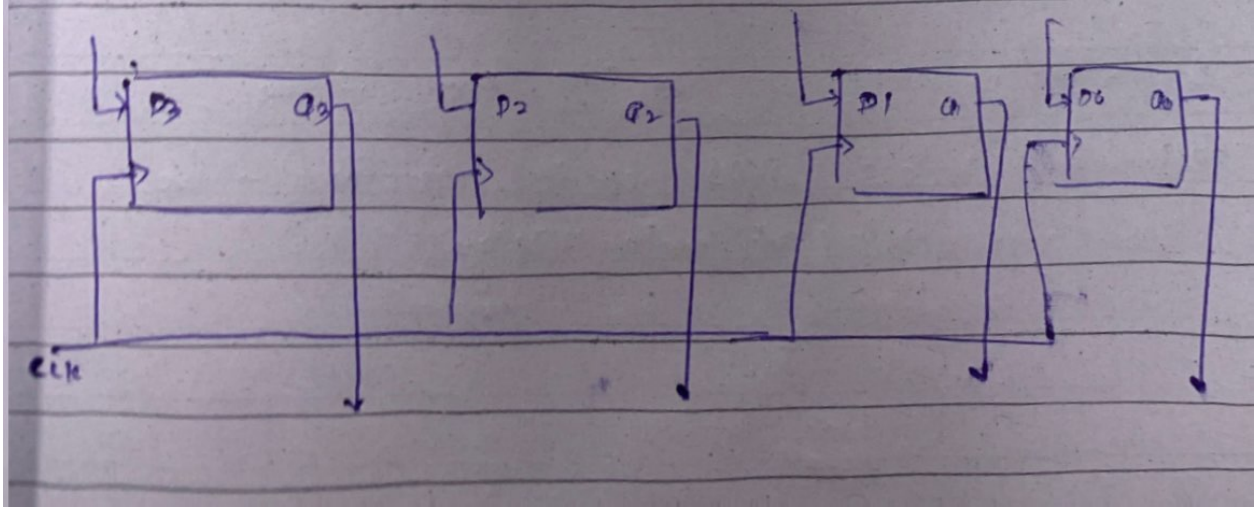
↳ SH/LD = 0 then data get loaded

② for shifte mode.

SH/LD = 1

storage register / Buffer register

PIPO (Parallel in Parallel out)

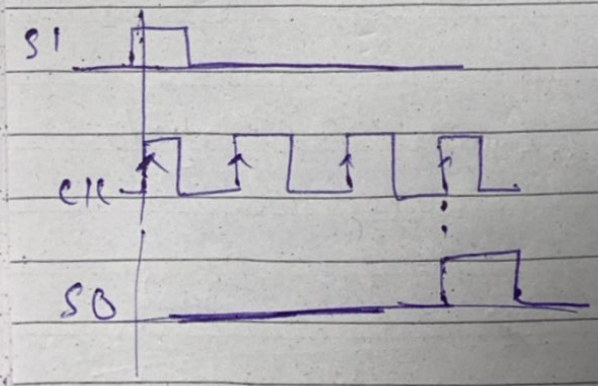


④ In PIPO we require only one clock pulse to store data and to get data.

Applications of Shift Registers.

① Generating a Time delay

as we know in SISO register, output comes after 4 cycles so it provide the delay of 4 clock cycles



② So in general we can say that if we have a N-bit SISO shift register - we can have N - clock cycle Delay

(1) shift register counter

↳ Two types.

① Johnson counter

② Ring counter.

Counter: is a digital circuit which count how many times the spectic event has been occured

↳ The event is noted based on the rising or falling edge of the signal at the clock input.

↳ The clock input could be periodic or Aperiodic (random & event driven).

(2) counters are also made up of flipflops. So the output of the each flipflop in the counter, represent the specific state of the counter.

(3) as the clock signal is applied to the counter, then at the rising or falling edge of the clock, the state of the counter will change in the specific sequence.

→ The number of different states, or the number of different sequence which can be represented by this counter is known as the Modulus or the mode of the counter.

↳ Eg: if the counter is mode 10 counter, then during the counting it will go through 10 diff: state in the specific sequence, after 10 the sequence will get repeated.

Ring Counter.

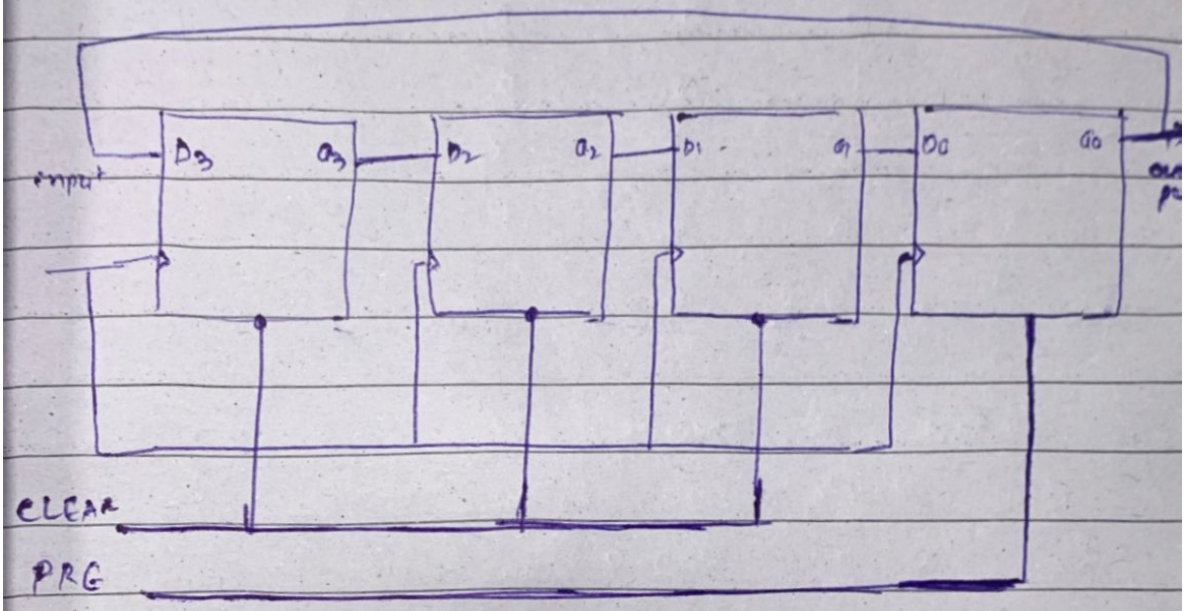
↳ The output of the shift register is connected back to the input, and initially using the preset & clear input of the flipflop the initial value of the register is loaded

Ⓨ So in ring counter initial value is loaded in such a way that, the output of only one flipflop is high & the remaining flipflop remains low

only flipflop

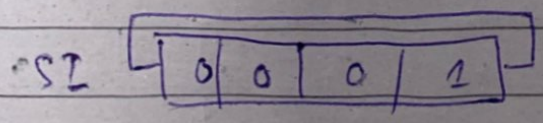
by ~~[illegible]~~

Ring counter.



↳ so in this case preset is only connected to D_0 flipflop while the clear is connected to D_3, D_2 & D_1, with the help of CLEAR D_3, D_2 & D_1 can be set to "0" and with the help of PRE D_0 can be set to "high" "1"

intial state is



to with every clk pulse one set loaded/shifted

clk	Q_3	Q_2	Q_1	Q_0	
x	0	0	0	1	(initial)
1	1	0	0	0	
2	0	1	0	0	
3	0	0	1	0	
4	0	0	0	1	

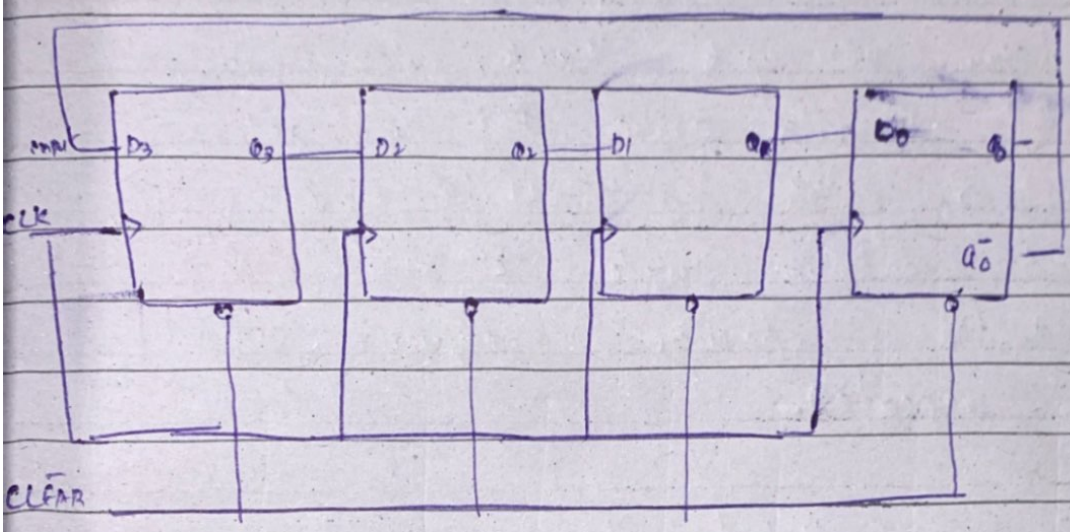
↳ so here a specific sequence repeat after 4 clk pulses, so we can say that this is a MOD-4 counter.

⇒ so in the ring counter the modulus or mode of the counter is same as the number of flip flops in the shift register

~~Ring counter with JK flipflop~~

Reset/CLEAR

Johnson comter. / Twisted Ring counter



- initially all the flipflops are set to zero with the help of clear input.

input is connected to $\bar{Q}$

CLK	Q_3	Q_2	Q_1	Q_0
X	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0

Since input is connected to $\bar{Q}$ so in 1st clk "one" will be passed to input from $\bar{Q}$ bcz $(\bar{Q}=1)$ at that time that flow goes upto 4 clk pulses as the 5th clk pulse arrive FF1 will get input = 0 — bcz [at 4th clk Q=1 so $\bar{Q}=0$] and it continue upto 8th clk pulse and the specific sequence "0000" gets repeated.

(4) So this 4-bit johnson counter is a MOD-8 counter

⇒ In general if we have a N-bit johnson counter then it will have total 2xN states

(5) we can also decode or confirm the 8 clk pulses by putting a AND gate and giving it four input of $\bar{Q_3}$, $\bar{Q_2}$, $\bar{Q_1}$ & $\bar{Q_0}$ so only if all are one then output of the AND gate will be one and it is only possible at a specific state 0000.

or we can just connect $\bar{Q_3}$ & $\bar{Q_0}$