

shift registers

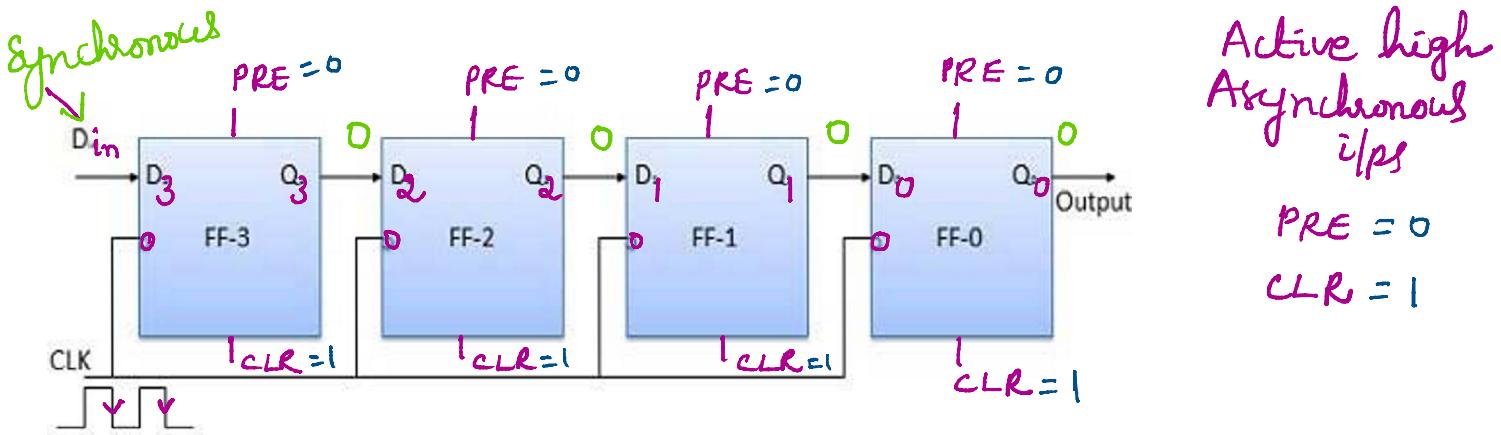
Flip-flop is a 1 bit memory cell which can be used for storing the digital data. To increase the storage capacity in terms of number of bits, we have to use a group of flip-flop. Such a group of flip-flop is known as a Register. The **n-bit** register will consist of **n number** of flip-flop and it is capable of storing an **n-bit word**.

The binary data in a register can be moved within the register from one flip-flop to another. The registers that allow such data transfers are called as shift registers. There are four mode of operations of a shift register.

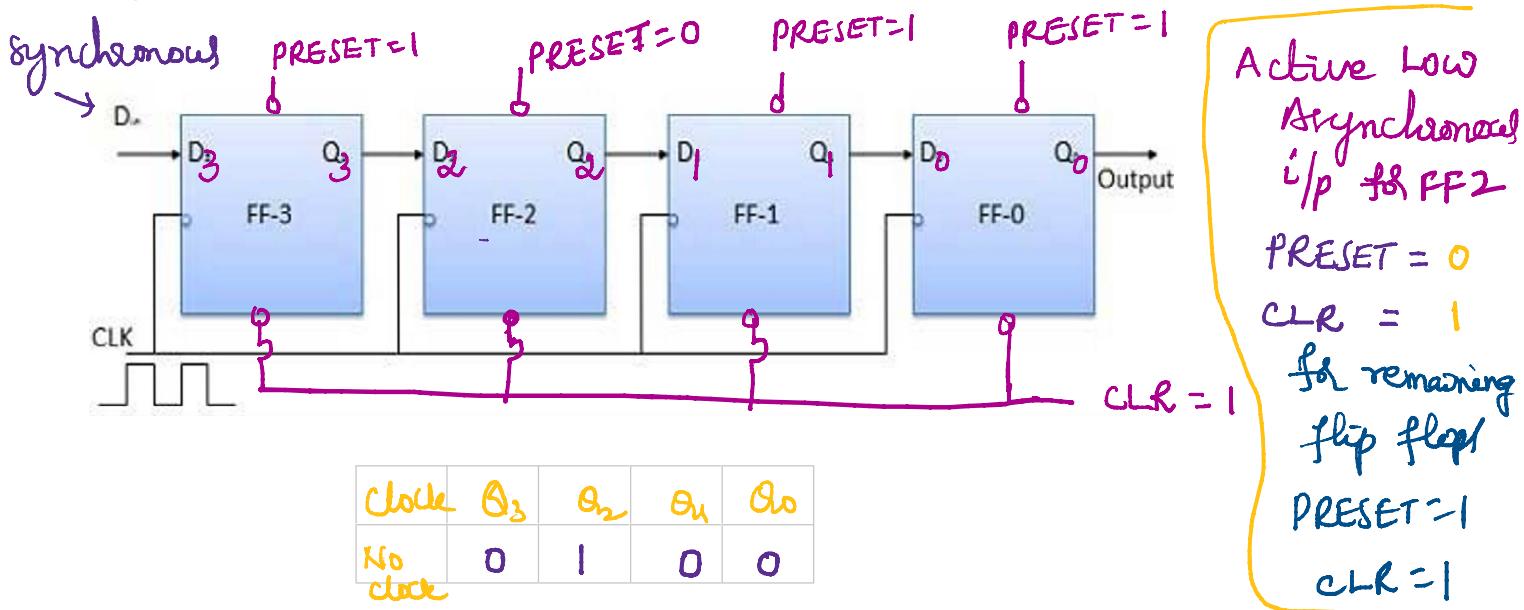
- Serial Input Serial Output
- Serial Input Parallel Output
- Parallel Input Serial Output
- Parallel Input Parallel Output

Serial Input Serial Output

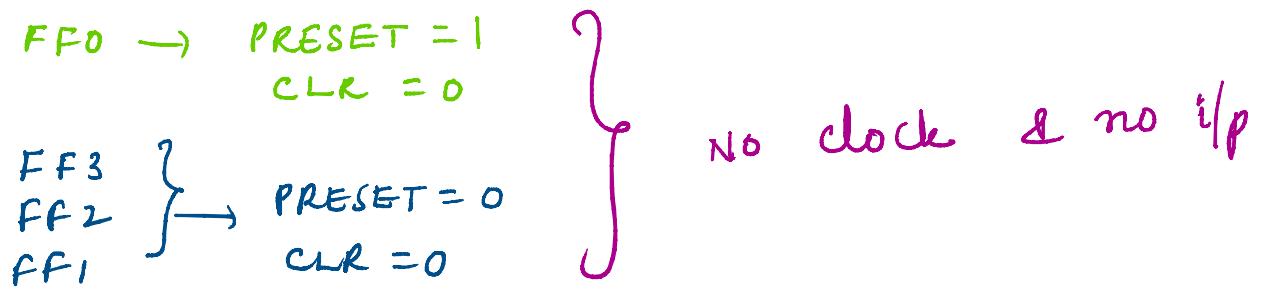
Let all the flip-flop be initially in the reset condition i.e. $Q_3 = Q_2 = Q_1 = Q_0 = 0$. If an entry of a four bit binary number **1 1 1 1** is made into the register, this number should be applied to **Din bit with the LSB** bit applied first. The D input of FF-3 i.e. D3 is connected to serial data input Din. Output of FF-3 i.e. Q3 is connected to the input of the next flip-flop i.e. D2 and so on.



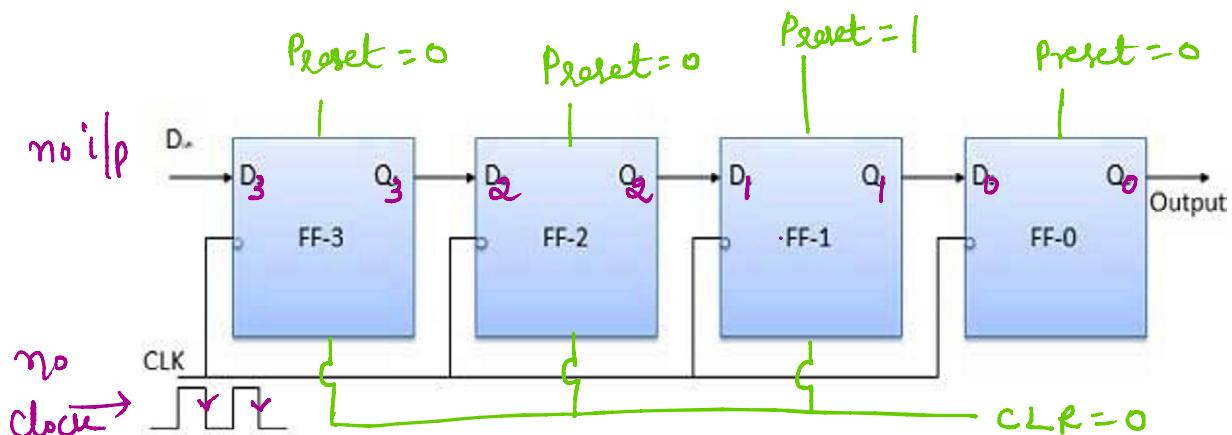
We want to make $\text{FF2} (\text{Q}_2=1)$ by using Active Low Asynchronous inputs



We want to make FF1 ($Q_1=1$) by using Active high Asynchronous inputs

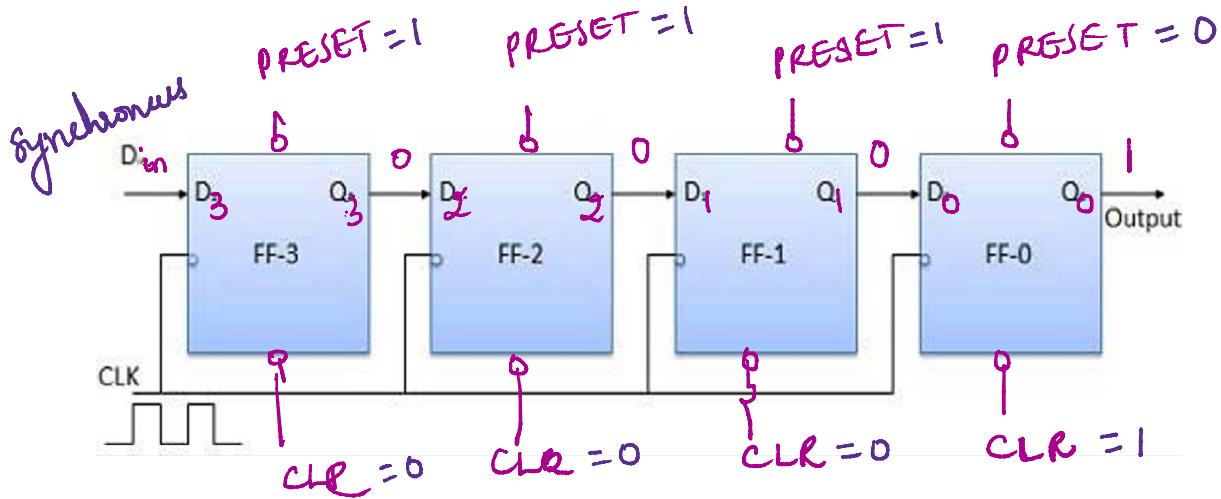


We want to make FF1 ($Q_1=1$) by using Active high Asynchronous inputs



clock	Q_3	Q_2	Q_1	Q_0
No clock	0	0	1	0

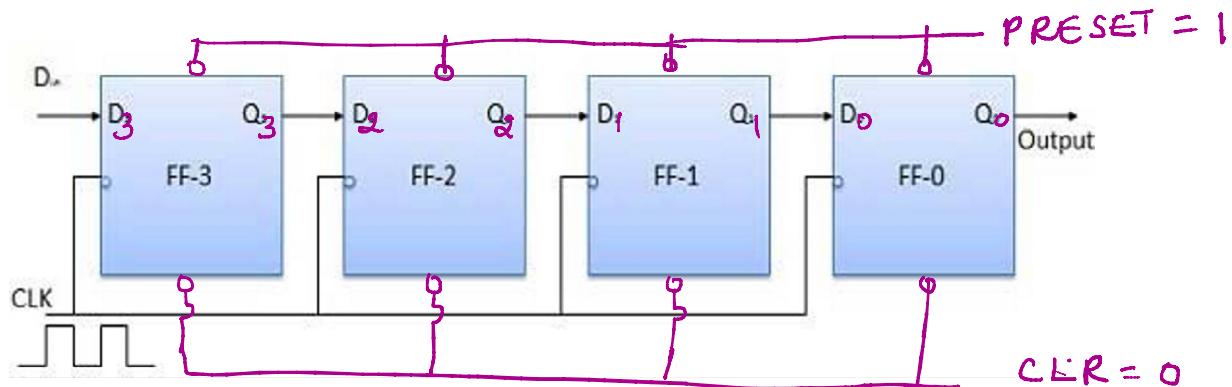
We want to make FF0 ($Q_0=1$) by using Active Low Asynchronous inputs



clock	Q_3	Q_2	Q_1	Q_0
No clock	0	0	0	1

Operation

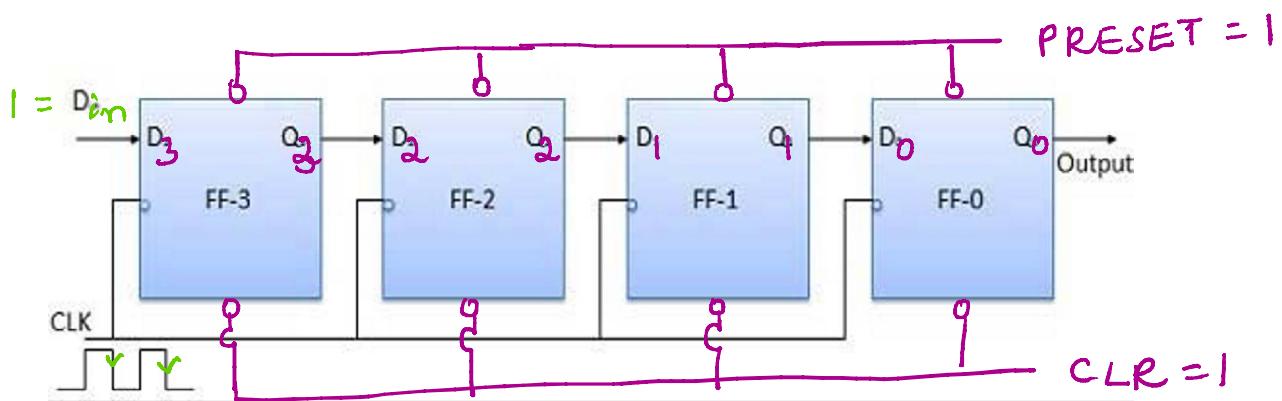
Before application of clock signal, let $Q_3 Q_2 Q_1 Q_0 = 0000$ and apply LSB bit of the number to be entered to D_{in} .



Without applying clock and data input, using active low asynchronous inputs ($\text{PRESET} = 1$ & $\text{CLR} = 0$) will make Q₀ of all D flip flops $Q_3 = Q_2 = Q_1 = Q_0 = 0$

Apply LSB bit of the number to be entered to Din. So $D_{in} = D_3 = 1$.
Apply the clock. On the first falling edge of clock, the FF-3 is set, and stored word in the register is $Q_3 Q_2 Q_1 Q_0 = 1000$.

With $\text{PRESET} = 1$ & $\text{CLR} = 1$ for active low asynchronous inputs, all flip flops perform clocked operation.



D _{in}	clock	Q ₃	Q ₂	Q ₁	Q ₀
1	✓	1	0	0	0

for the first -ve triggered clock pulse

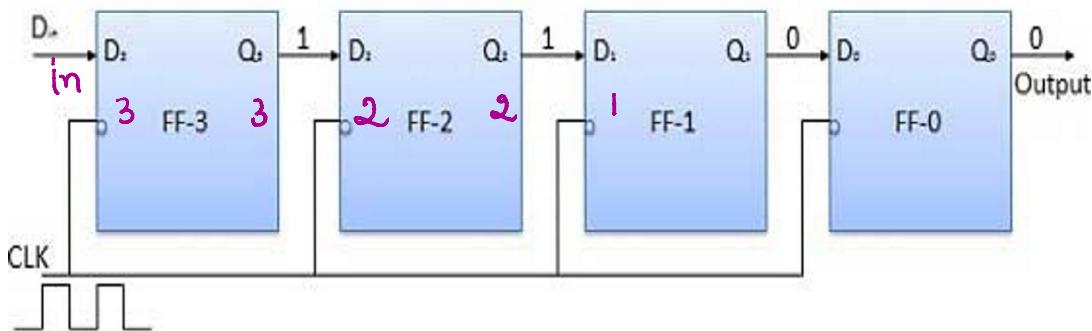
$D_{in} = 1$ is given to D_3 then $Q_3 = 1$

During second -ve triggered clock pulse

$D_{in} = 1$ is given to D_3 then $Q_3 = 1$

Here $Q_3 = D_2$ then $Q_2 = 1$

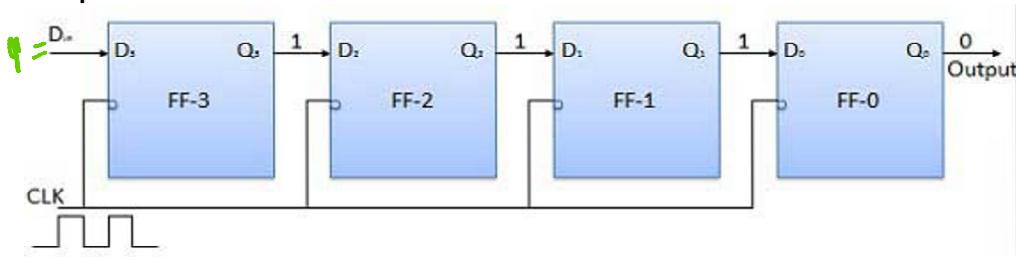
edge of the clock hits, FF-2 will set and the stored word change to
 $Q_3 Q_2 Q_1 Q_0 = 1100$.

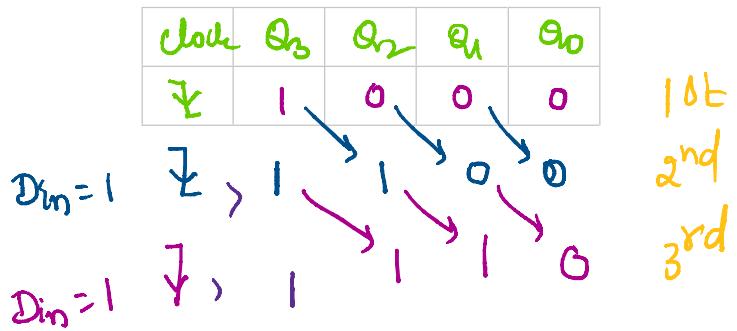


$$D_{in} \rightarrow D_3 = 1 \Rightarrow Q_3 = 1$$

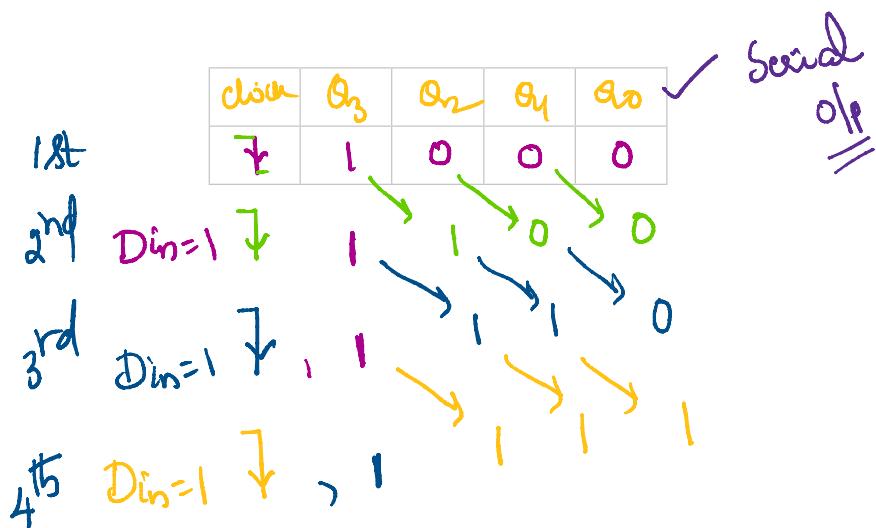
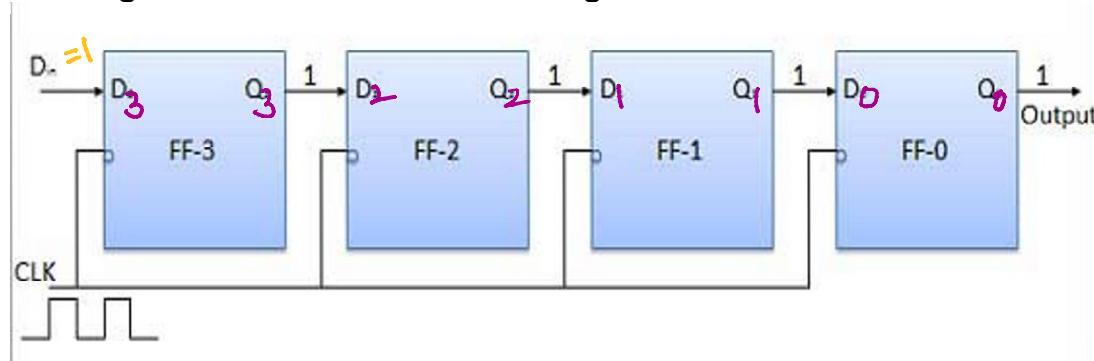
clock	Q_3	Q_2	Q_1	Q_0
1st	1	0	0	0
2nd	1	1	0	0

Apply the next bit to be stored i.e. 1 to D_{in} . Apply the clock pulse.
As soon as the third negative clock edge hits, FF-1 will be set and output will be modified to $Q_3 Q_2 Q_1 Q_0 = 1110$.





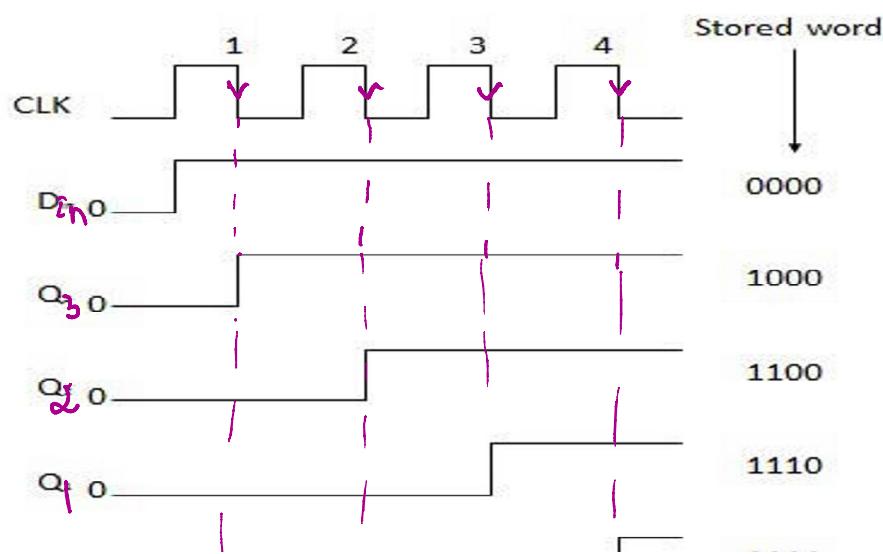
Similarly with $D_{in} = 1$ and with the fourth negative clock edge arriving, the stored word in the register is $Q_3 Q_2 Q_1 Q_0 = 1111$.

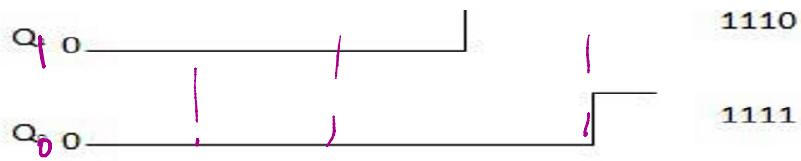


Truth Table

clock	Q_3	Q_2	Q_1	Q_0
No clock	0	0	0	0
$D_3=1$	0	0	0	0
$D_1=1$	1	0	0	0
$D_1=1$	1	1	0	0
$D_1=1$	1	1	1	0
$D_1=1$	1	1	1	1
$D_0=0$	0	1	1	1
$D_1=1$	1	0	1	1

Waveforms

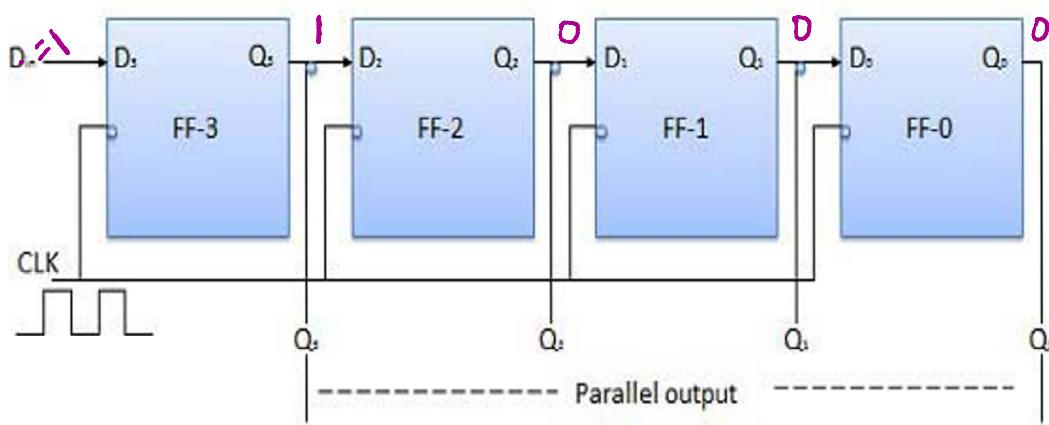




Serial Input Parallel Output

- In such types of operations, the data is entered serially and taken out in parallel fashion.
- Data is loaded bit by bit. The outputs are disabled as long as the data is loading.
- As soon as the data loading gets completed, all the flip-flops contain their required data, the outputs are enabled so that all the loaded data is made available over all the output lines at the same time.
- 4 clock cycles are required to load a four bit word. Hence the speed of operation of SIPO mode is same as that of SISO mode.

Block Diagram



for first clock $\rightarrow D_{in} \rightarrow Q_3 \quad Q_2 \quad Q_1 \quad Q_0$

$D_{in}=1 \quad 1 \quad 0 \quad 0 \quad 0$

2nd clock $D_{in}=0 \quad 0 \quad 1 \quad 0 \quad 0$

Parallel Input Serial Output (PISO)

- Data bits are entered in parallel fashion.
- The circuit shown below is a four bit parallel input serial output register.
- Output of previous Flip Flop is connected to the input of the next one via a combinational circuit.
- The binary input word B_0, B_1, B_2, B_3 is applied through the same combinational circuit.
- There are two modes in which this circuit can work namely - shift mode or load mode.

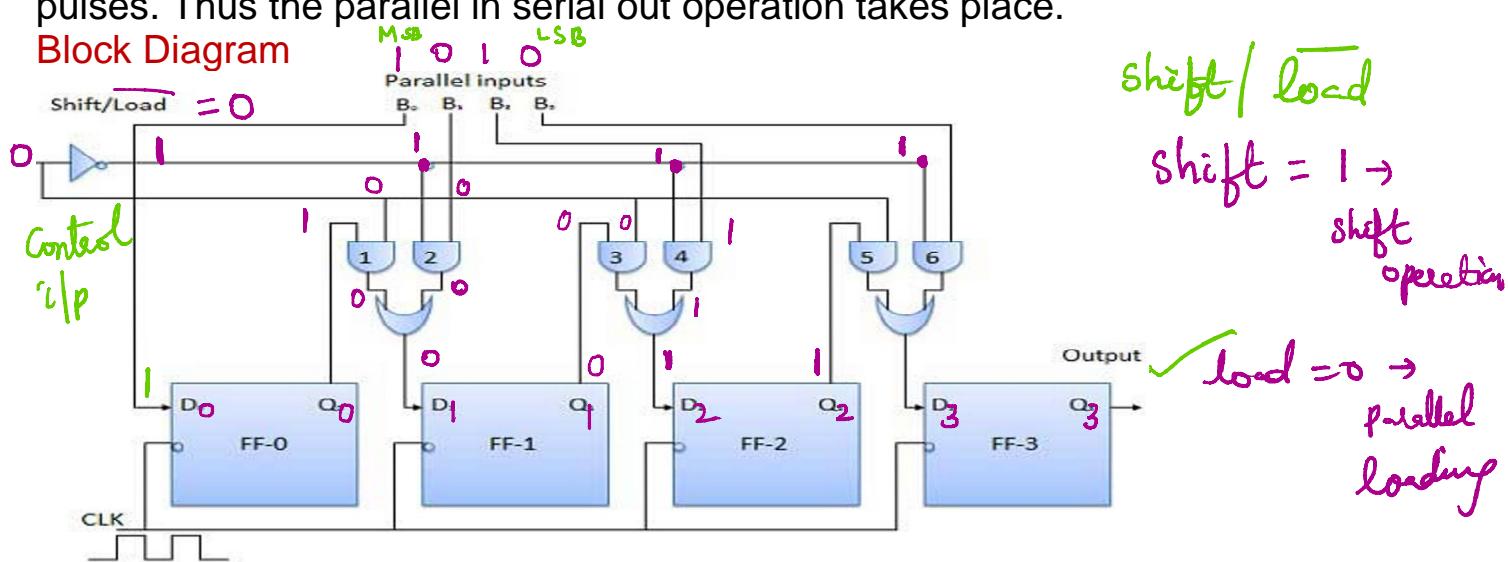
Load mode

When the shift/load bar line is low (0), the AND gate 2, 4 and 6 become active they will pass B_1, B_2, B_3 bits to the corresponding flip-flops. On the low going edge of clock, the binary input B_0, B_1, B_2, B_3 will get loaded into the corresponding flip-flops. Thus parallel loading takes place.

Shift mode

When the **shift/load** bar line is high (1), the AND gate 2, 4 and 6 become inactive. Hence the parallel loading of the data becomes impossible. But the AND gate 1,3 and 5 become active. Therefore the shifting of data from left to right bit by bit on application of clock pulses. Thus the parallel in serial out operation takes place.

Block Diagram



shift / $\overline{\text{load}}$ = low \rightarrow Gates G_2, G_4 & G_6 enabled
allowing each data bit to be applied to D input of respective flip flop

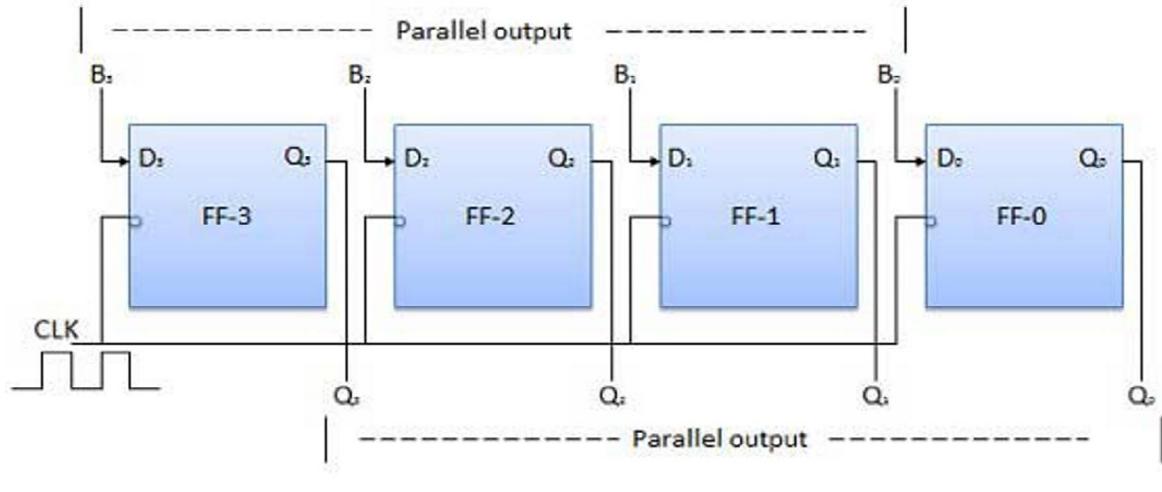
shift / $\overline{\text{load}}$ = High \rightarrow Gates G_1, G_3 & G_5 enabled.
and G_2, G_4 & G_6 disabled.
This allows the data bits to shift from one stage to next

OR gates at the D input of the flip flop allow either the parallel data entry operation (or) shift operation, depending on AND gates are enabled by level on the shift / $\overline{\text{load}}$ input

Parallel Input Parallel Output (PIPO)

In this mode, the 4 bit binary input B_0, B_1, B_2, B_3 is applied to the data inputs D_0, D_1, D_2, D_3 respectively of the four flip-flops. As soon as a negative clock edge is applied, the input binary bits will be loaded into the flip-flops simultaneously. The loaded bits will appear simultaneously to the output side. Only clock pulse is essential to load all the bits.

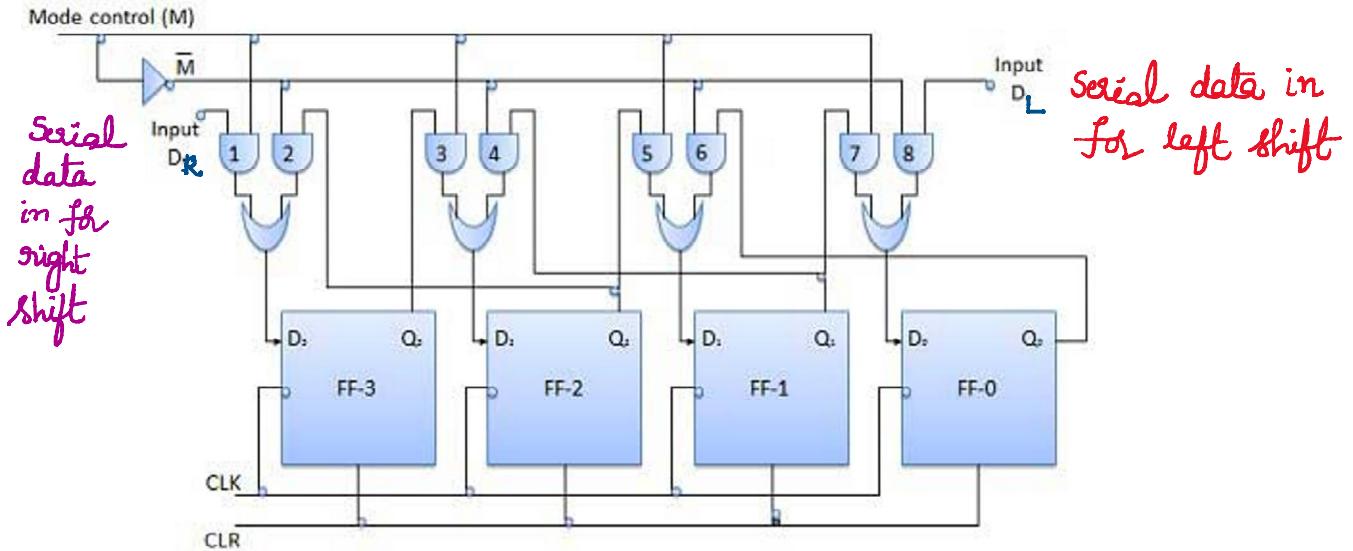
Block Diagram



Bidirectional Shift Register

- If a binary number is shifted left by one position then it is equivalent to multiplying the original number by 2. Similarly if a binary number is shifted right by one position then it is equivalent to dividing the original number by 2.
- Hence if we want to use the shift register to multiply and divide the given binary number, then we should be able to move the data in either left or right direction.
- Such a register is called bi-directional register. A four bit bi-directional shift register is shown in fig.
- There are two serial inputs namely the serial right shift data input DR, and the serial left shift data input DL along with a mode select input (M).

Block Diagram



Operation

S.N.	Condition	Operation
1	With $M = 1$ – Shift right operation	If $M = 1$, then the AND gates 1, 3, 5 and 7 are enabled whereas the remaining AND gates 2, 4, 6 and 8 will be disabled. The data at D_R is shifted to right bit by bit from FF-3 to FF-0 on the application of clock pulses. Thus with $M = 1$ we get the serial right shift operation.
2	With $M = 0$ – Shift left operation	When the mode control M is connected to 0 then the AND gates 2, 4, 6 and 8 are enabled while 1, 3, 5 and 7 are disabled. The data at D_L is shifted left bit by bit from FF-0 to FF-3 on the application of clock pulses. Thus with $M = 0$ we get the serial left shift operation.