

UNIT-5

Registers:

A Register is a digital circuit that both stores data and moves data.

Shift Register:

Shift registers are a type of sequential logic circuit, mainly for storage of digital data. They are a group of flip-flops connected in a chain so that the output from one flip-flop becomes the input of the next flip-flop. Most of the registers possess no characteristic internal sequence of states. All flip-flop is driven by a common clock, and all are set or reset simultaneously.

Buffer Register:

Buffer registers are a type of registers used to store a binary word. These can be constructed using a series of **flip-flops** as each flip-flop can store a single bit. This means that in order to store an n-bit binary word one should design an array of n flip-flops. Figure 1 shows a 4 bit synchronous buffer register formed by cascading four positive edge triggered **D flip-flops**. Here the entire input data word B1B2B3B4 is loaded onto the register at a single clock tick. This means that at every leading edge of the clock the values of flip-flop outputs follow their input bits i.e. $Q_1 = B_1$, $Q_2 = B_2$, $Q_3 = B_3$ and $Q_4 = B_4$ as shown by Figure 2.

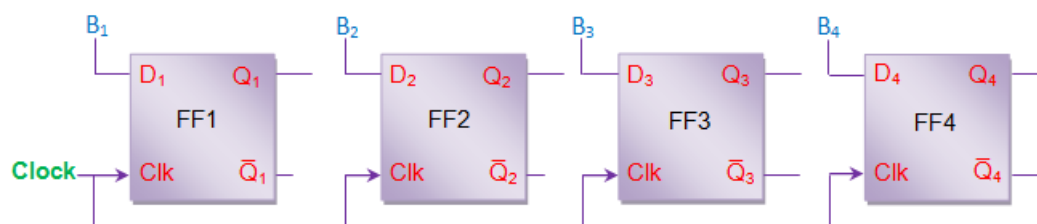


Figure 1 4-bit Buffer Register

Buffer registers offer no means of control over the inputs which in turn leads to uncontrolled outputs. In order to overcome this drawback one can resort to controlled buffer registers as shown by Figure 3.

In this design, tri-state switches are used to control the operation of loading and/or retrieval of the data to/from the buffer register. Here one has to pull the

\overline{LD}

or

\overline{WR}

Control line(blue line) low in order to store the data into the register, while

\overline{RD}

Control line(red line) should be made low to read the data.

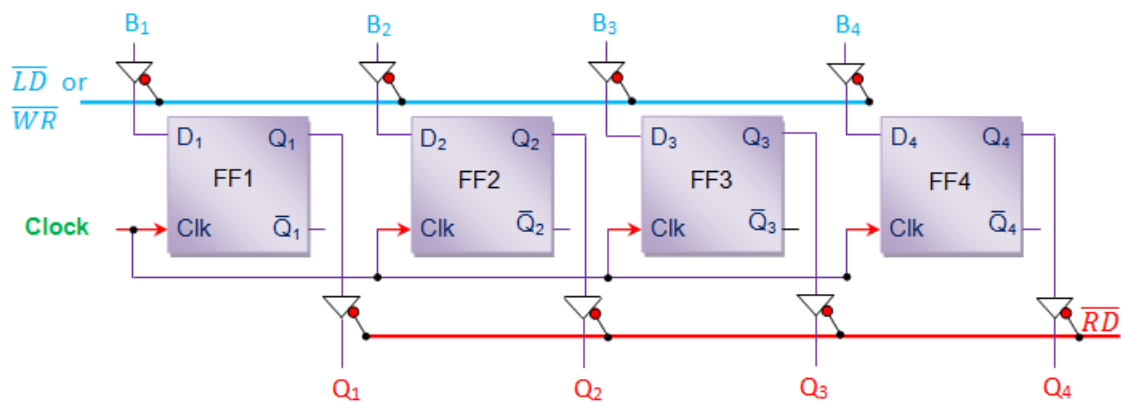


Figure 3 4-bit Controlled Buffer Register

Data transmission in shift register:

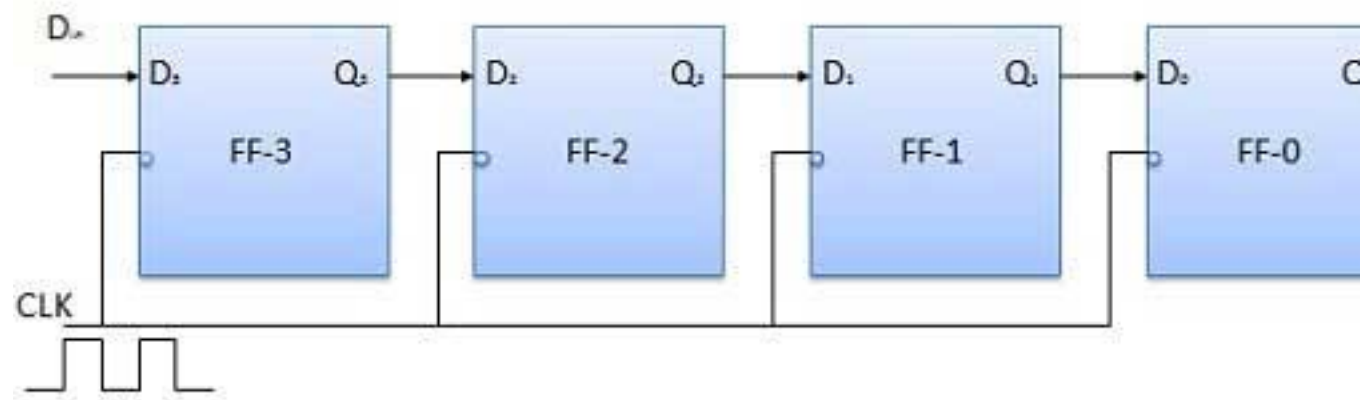
The binary data in a register can be moved within the register from one flip-flop to another. The registers that allow such data transfers are called as **shift registers**. There are four mode of operations of a shift register.

- 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.

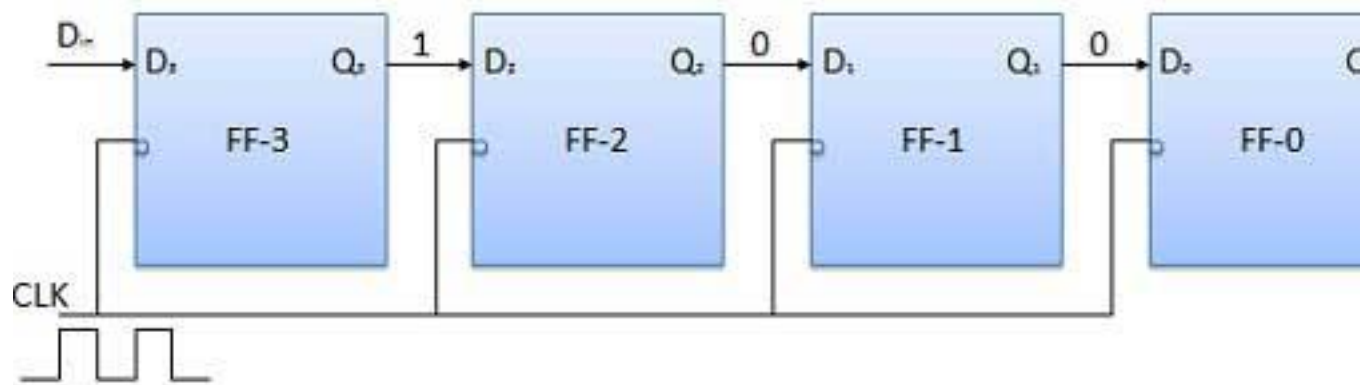
Block Diagram



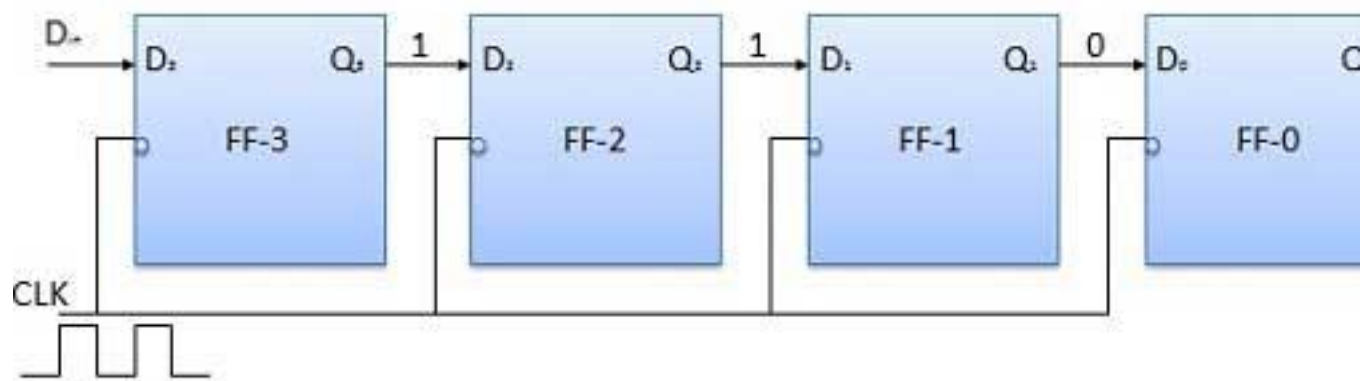
Operation

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 Din. So $D_{in} = D_3 = 1$. Apply the clock. On the first

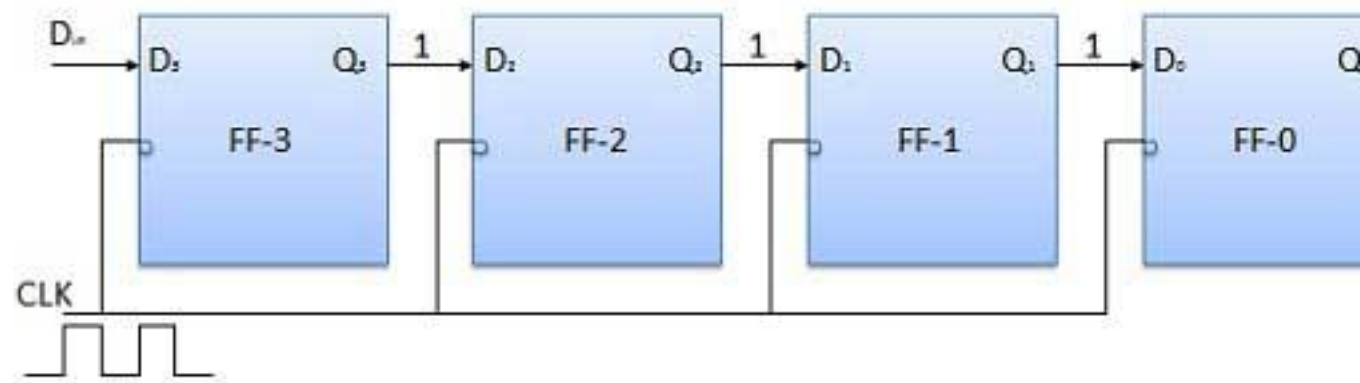
falling edge of clock, the FF-3 is set, and stored word in the register is $Q_3 Q_2 Q_1 Q_0 = 1000$.



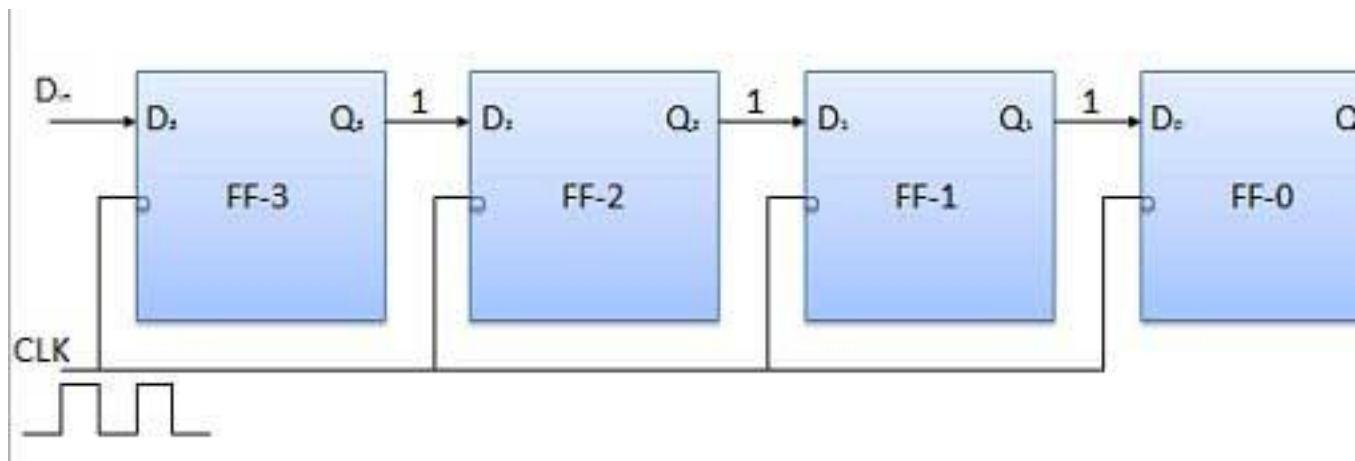
Apply the next bit to D_{in} . So $D_{in} = 1$. As soon as the next negative edge of the clock hits, FF-2 will set and the stored word change to $Q_3 Q_2 Q_1 Q_0 = 1100$.



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 $Q_3 Q_2 Q_1 Q_0 = 1111$.

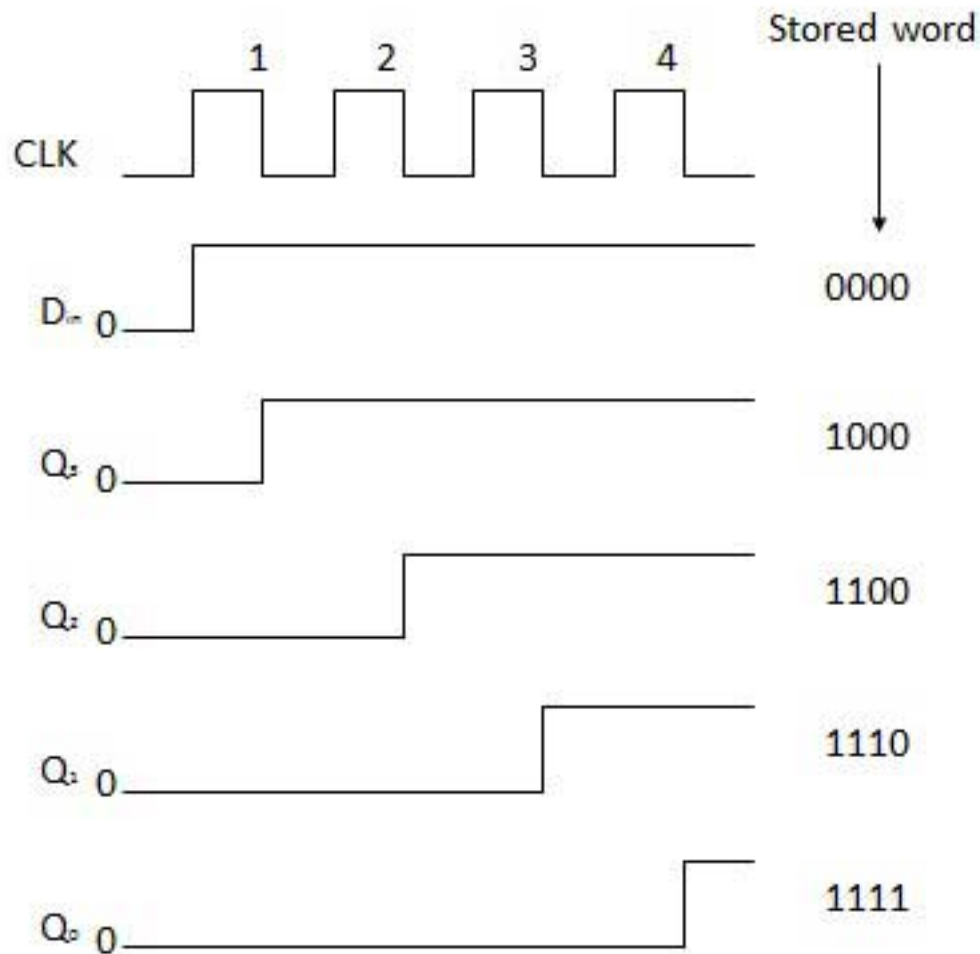


Truth Table

	CLK	$D_n = Q_3$	$Q_3 = D_2$	$Q_2 = D_1$	$Q_1 = D_0$	Q_0
Initially			0	0	0	0
(i)	↓	1	1	0	0	0
(ii)	↓	1	1	1	0	0
(iii)	↓	1	1	1	1	0
(iv)	↓	1	1	1	1	1

→ Direction of data travel

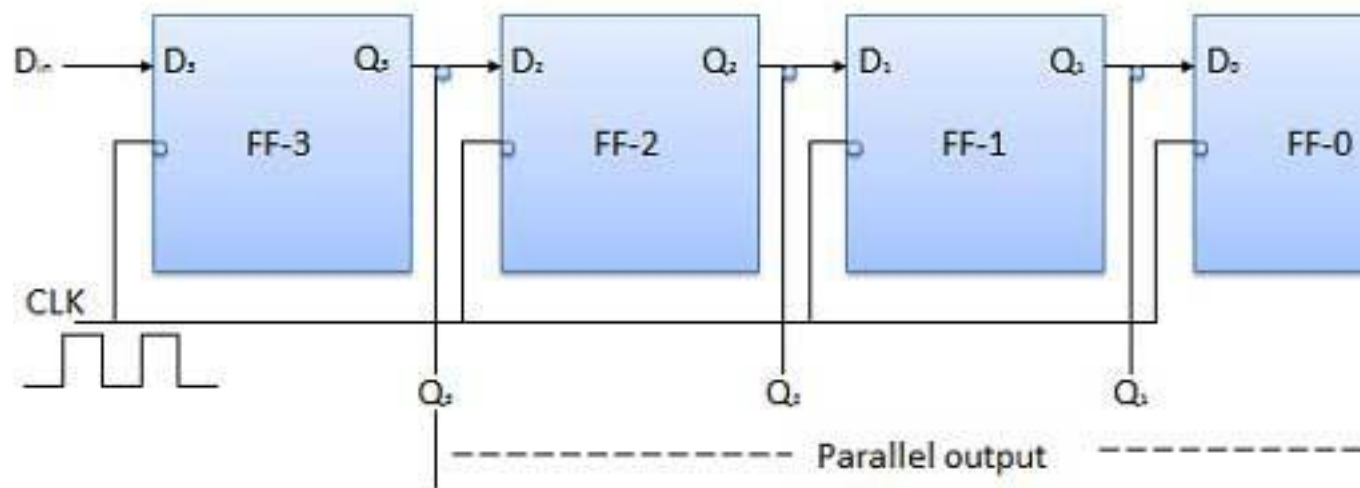
Waveforms



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



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 B0, B1, B2, B3 is applied through the same combinational circuit.
- There are two modes in which this circuit can work namely - shift mode or load mode.

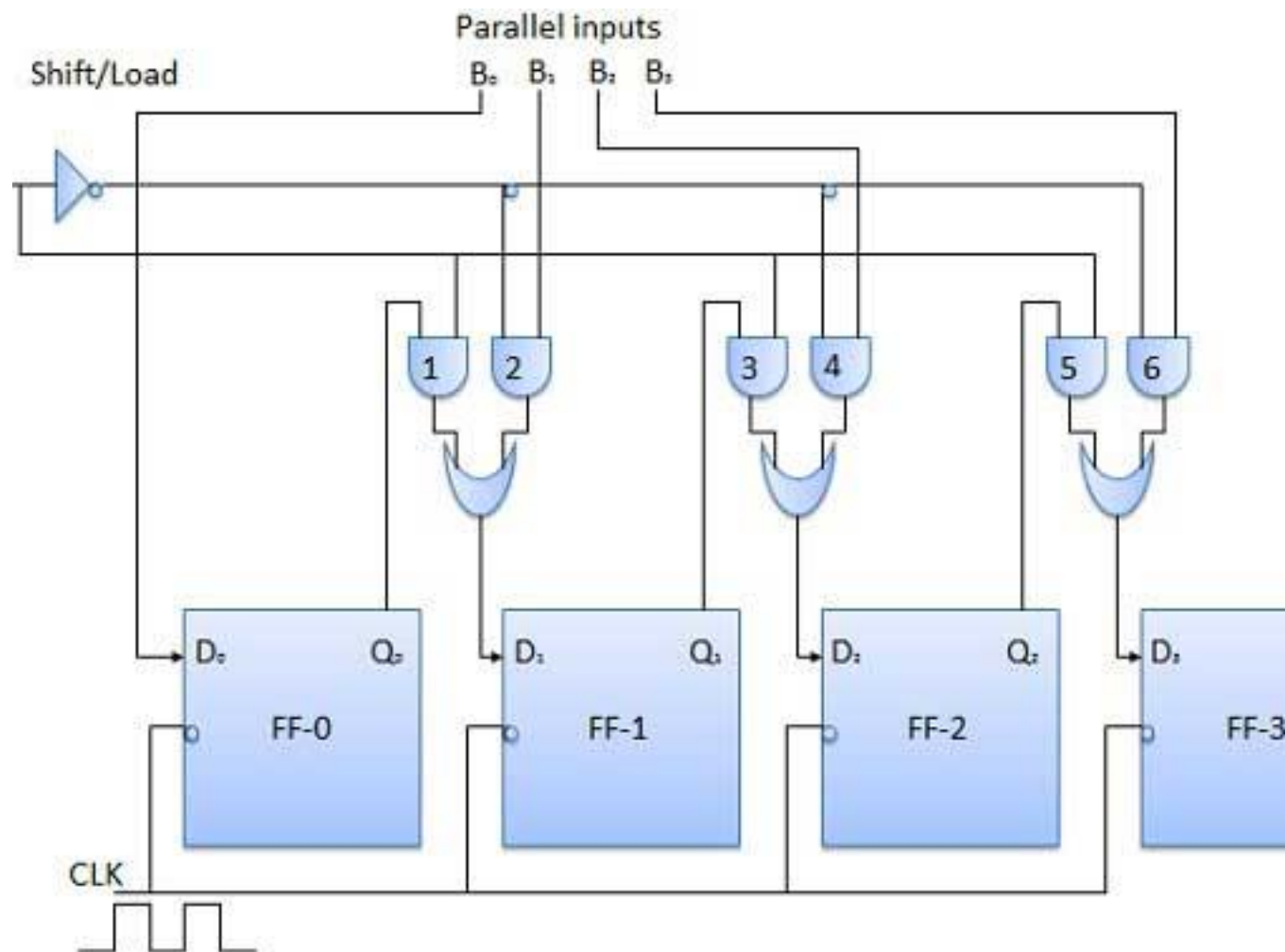
Load mode

When the shift/load bar line is low (0), the AND gate 2, 4 and 6 become active they will pass B1, B2, B3 bits to the corresponding flip-flops. On the low going edge of clock, the binary input B0, B1, B2, B3 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.

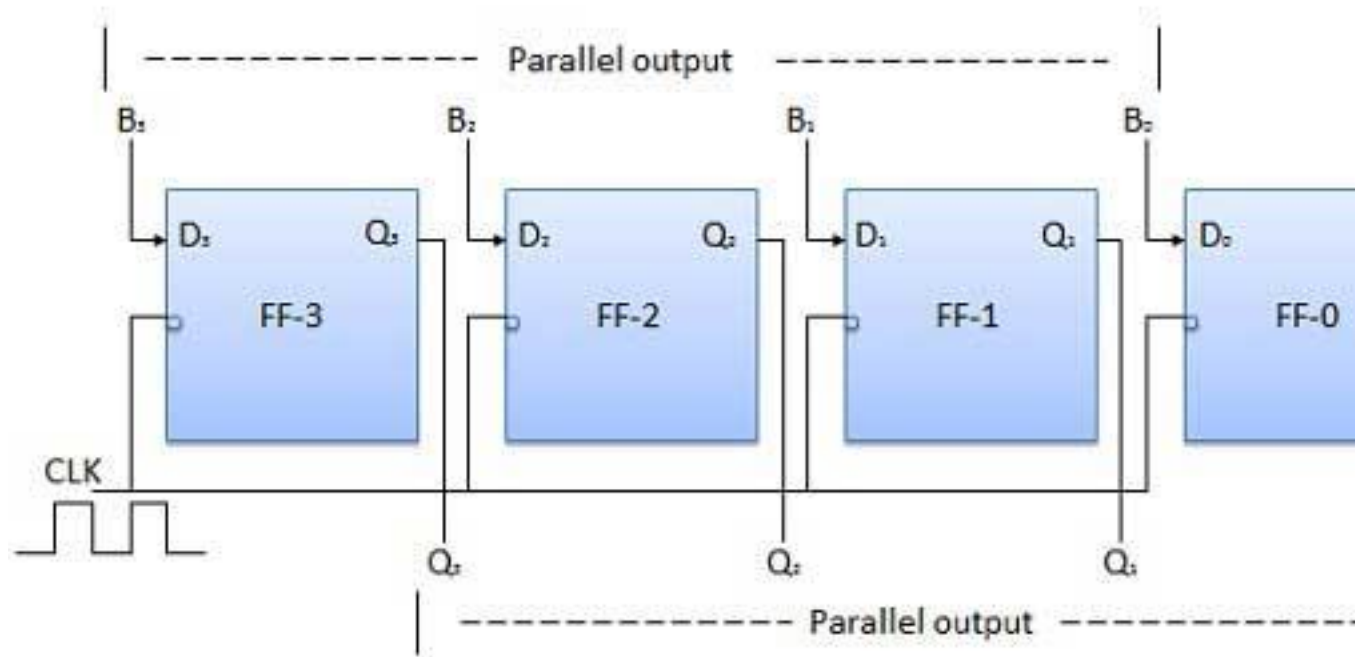
Block Diagram



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.

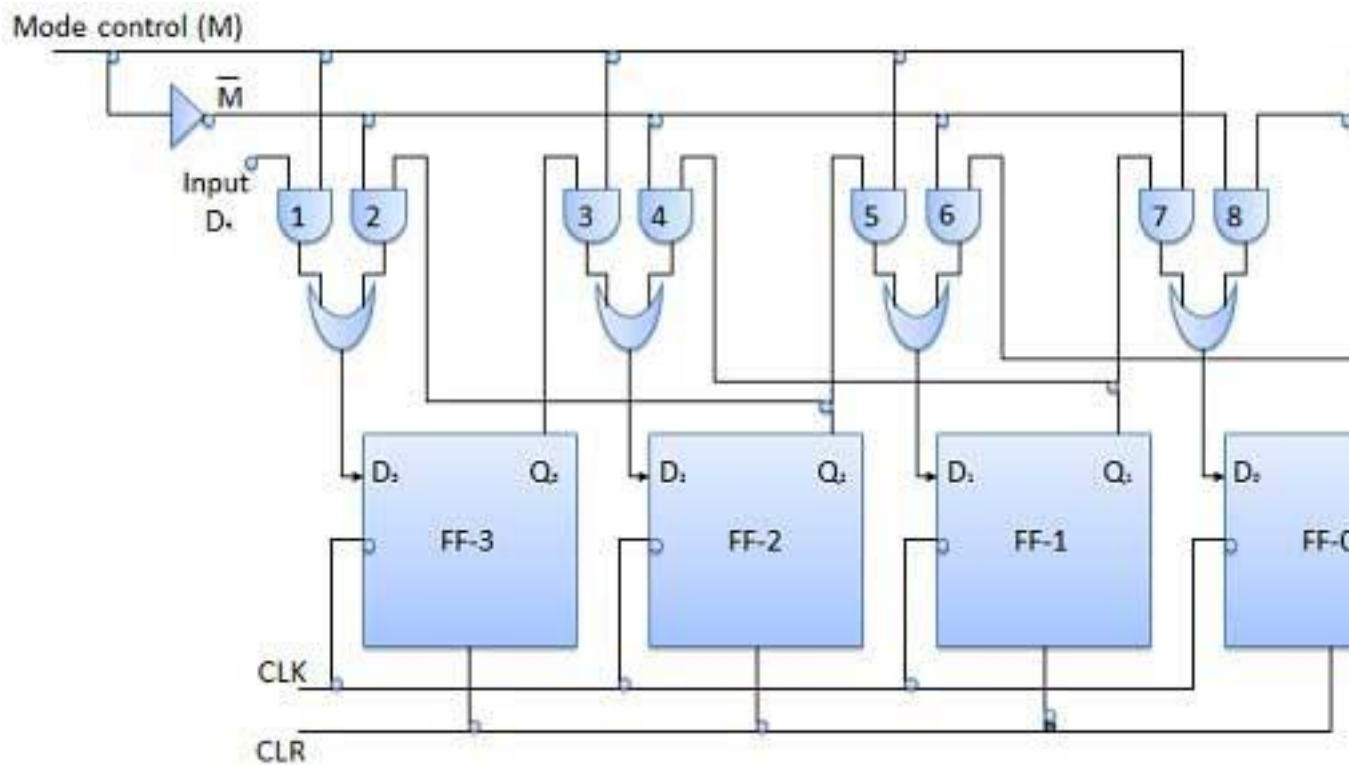
Block Diagram



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 DR 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 DL 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 right shift operation.

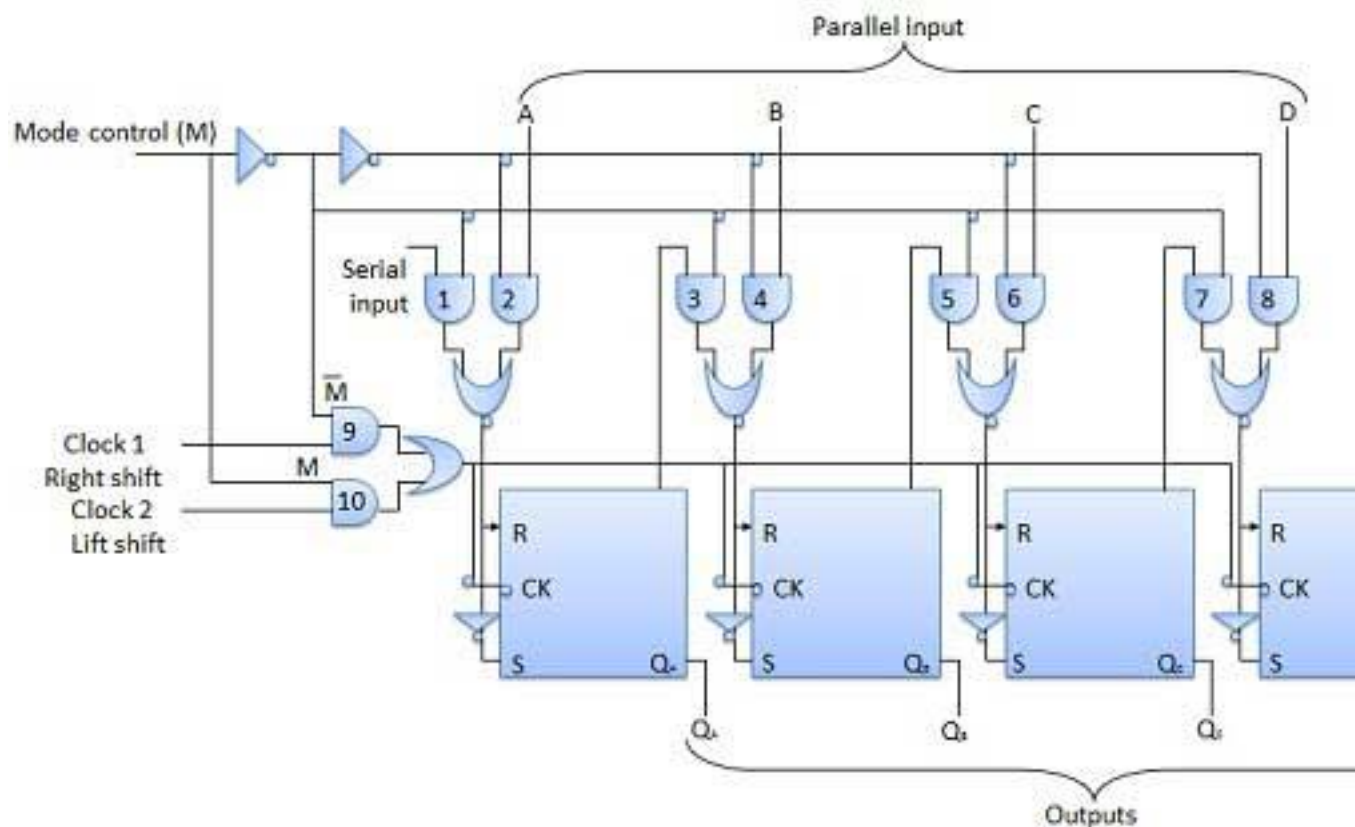
Universal Shift Register

A shift register which can shift the data in only one direction is called a uni-directional shift register. A shift register which can shift the data in both directions is called a bi-directional shift register. Applying the same logic, a shift register which can shift the data in both directions as well as load it parallelly, is known as a universal shift register. The shift register is capable of performing the following operation –

- Parallel loading
- Left Shifting
- Right shifting

The mode control input is connected to logic 1 for parallel loading operation whereas it is connected to 0 for serial shifting. With mode control pin connected to ground, the universal shift register acts as a bi-directional register. For serial left operation, the input is applied to the serial input which goes to AND gate-1 shown in figure. Whereas for the shift right operation, the serial input is applied to D input.

Block Diagram



Counters:

Counter is a sequential circuit. A digital circuit which is used for a counting pulses is known counter. Counter is the widest application of flip-flops. It is a group of flip-flops with a clock signal applied.

Classification:

Counters are of two types.

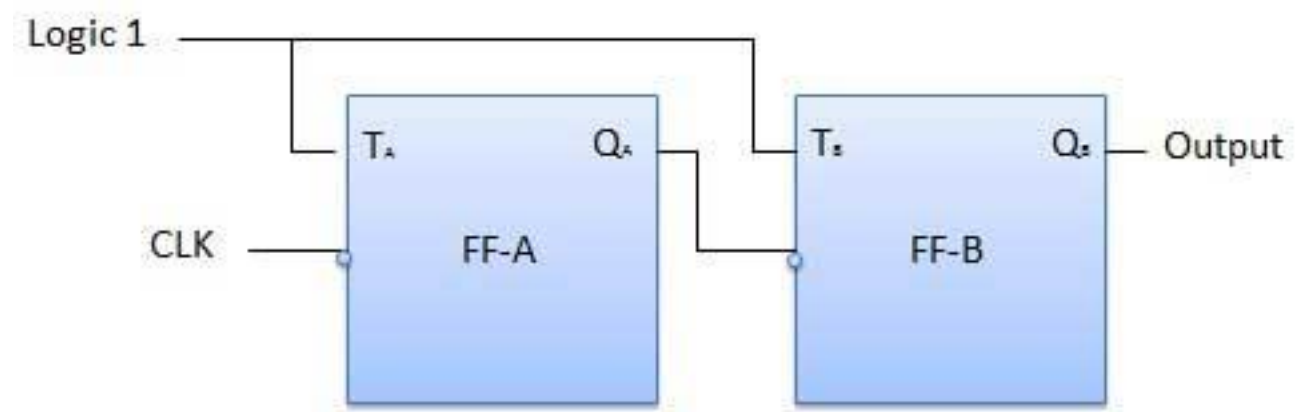
- Asynchronous or ripple counters.
- Synchronous counters.

Asynchronous or ripple counters:

The input signal is applied to the clock input of the first FF, and the output of each FF is connected directly to the clock input of the next.

The logic diagram of a 2-bit ripple up counter is shown in figure. The toggle (T) flip-flop are being used. But we can use the JK flip-flop also with J and K connected permanently to logic 1. External clock is applied to the clock input of flip-flop A and QA output is applied to the clock input of the next flip-flop i.e. FF-B.

Logical Diagram



Operation

S.N.	Condition	Operation
1	Initially let both the FFs be in the reset state	QBQA = 00 initially
2	After 1st negative clock edge	As soon as the first negative clock edge is applied, FF-A will toggle and QA will be equal to 1. QA is connected to clock input of FF-B. Since QA has changed from 0 to 1, it is treated as the positive clock edge by FF-B. There is no change in QB because FF-B is a negative

		edge triggered FF. QBQA = 01 after the first clock pulse.
3	After 2nd negative clock edge	On the arrival of second negative clock edge, FF-A toggles again and QA = 0. The change in QA acts as a negative clock edge for FF-B. So it will also toggle, and QB will be 1. QBQA = 10 after the second clock pulse.
4	After 3rd negative clock edge	On the arrival of 3rd negative clock edge, FF-A toggles again and QA become 1 from 0. Since this is a positive going change, FF-B does not respond to it and remains inactive. So QB does not change and continues to be equal to 1. QBQA = 11 after the third clock pulse.
5	After 4th negative clock edge	On the arrival of 4th negative clock edge, FF-A toggles again and QA becomes 1 from 0. This negative change in QA acts as clock pulse for FF-B. Hence it toggles to change QB from 1 to 0. QBQA = 00 after the fourth clock pulse.

Truth Table

Clock	Counter output		State number	Decimal Counter output
	Q ₂	Q ₁		
Initially	0	0	—	0
1st	0	1	1	1
2nd	1	0	2	2
3rd	1	1	3	3
4th	0	0	4	0

Effect of propagation delay in ripple counters:

A major problem with ripple counters arises from the propagation delay of the flip-flops constituting the counter. The effective propagation delay in a ripple counter is equal to the sum of propagation delays due to different flip-flops. The situation becomes worse with increase in the number of flip-flops used to construct the counter, which is the case in larger bit counters. An increased propagation delay puts a limit on the maximum frequency used as clock input to the counter. We can appreciate that the clock signal time period must be equal to or greater than the total propagation delay. The maximum clock frequency therefore corresponds to a time period that equals the total propagation delay. If t_{pd} is the propagation delay in each flip-flop, then, in a counter with N flip-flops having a modulus of less than or equal to $2N$, the maximum usable clock frequency is given by $f_{max} = 1/(N \times t_{pd})$. Often the propagation delay times are specified in the case of flip-flops, one for LOW-to-HIGH transition (t_{pLH}) and the other for HIGH-to-LOW transition (t_{pHL}) at the output. In such a case, the larger of the two should be considered for computing the maximum clock frequency.