

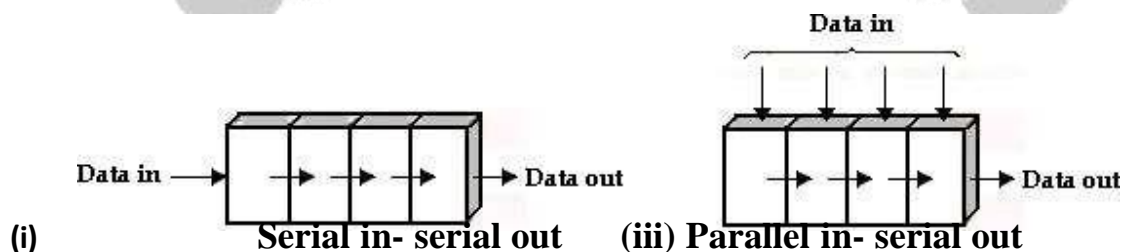
SHIFT REGISTERS:

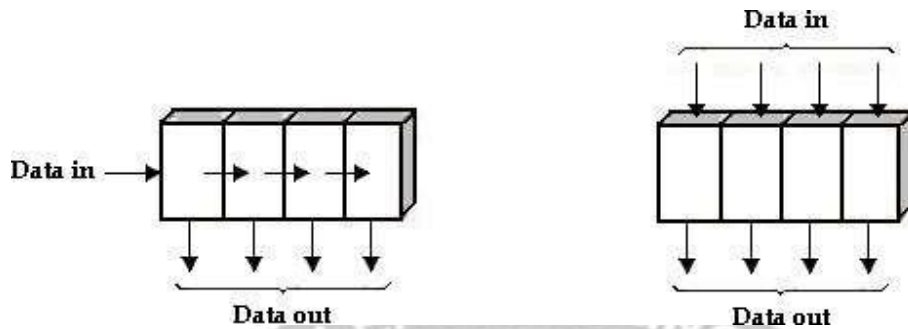
A register is simply a group of Flip-Flops that can be used to store a binary number. There must be one Flip-Flop for each bit in the binary number. For instance, a register used to store an 8-bit binary number must have 8 Flip-Flops.

The Flip-Flops must be connected such that the binary number can be entered (shifted) into the register and possibly shifted out. A group of Flip-Flops connected to provide either or both of these functions is called a *shift register*.

The bits in a binary number (data) can be removed from one place to another in either of two ways. The first method involves shifting the data one bit at a time in a serial fashion, beginning with either the most significant bit (MSB) or the least significant bit (LSB). This technique is referred to as *serial shifting*. The second method involves shifting all the data bits simultaneously and is referred to as *parallel shifting*. There are two ways to shift into a register (serial or parallel) and similarly two ways to shift the data out of the register. This leads to the construction of four basic register types—

- i. Serial in- serial out,
- ii. Serial in- parallel out,
- iii. Parallel in- serial out, iv. Parallel in- parallel out.





(iii) Serial in- parallel out (iv) Parallel in- parallel out

Fig : 3.43 – Various types of Shift Registers

Serial-In Serial-Out Shift Register:

The serial in/serial out shift register accepts data serially, i.e., one bit at a time on a single line. It produces the stored information on its output also in serial form.

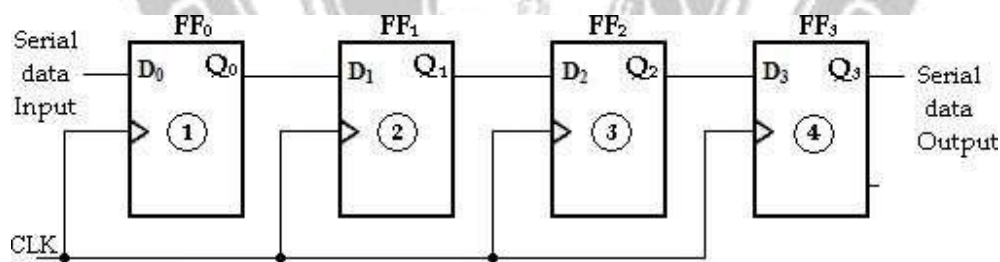


Fig : 3.44 - Serial-In Serial-Out Shift Register

The entry of the four bits 1010 into the register is illustrated below, beginning with the right-most bit. The register is initially clear. The 0 is put onto the data input line, making $D=0$ for FF_0 . When the first clock pulse is applied, FF_0 is reset, thus storing the 0.

Next the second bit, which is a 1, is applied to the data input, making $D=1$ for FF_0 and $D=0$ for FF_1 because the D input of FF_1 is connected to the Q_0 output. When the second clock pulse occurs, the 1 on the data input is shifted into FF_0 , causing FF_0 to set; and the 0 that was in FF_0 is shifted into FF_1 .

The third bit, a 0, is now put onto the data-input line, and a clock pulse is applied. The 0 is entered into FF_0 , the 1 stored in FF_0 is shifted into FF_1 , and the 0 stored in FF_1 is shifted into FF_2 .

The last bit, a 1, is now applied to the data input, and a clock pulse is applied.

This time the 1 is entered into FF0, the 0 stored in FF0 is shifted into FF1, the 1 stored in FF1 is shifted into FF2, and the 0 stored in FF2 is shifted into FF3. This completes the serial entry of the four bits into the shift register, where they can be stored for any length of time as long as the Flip-Flops have dc power.

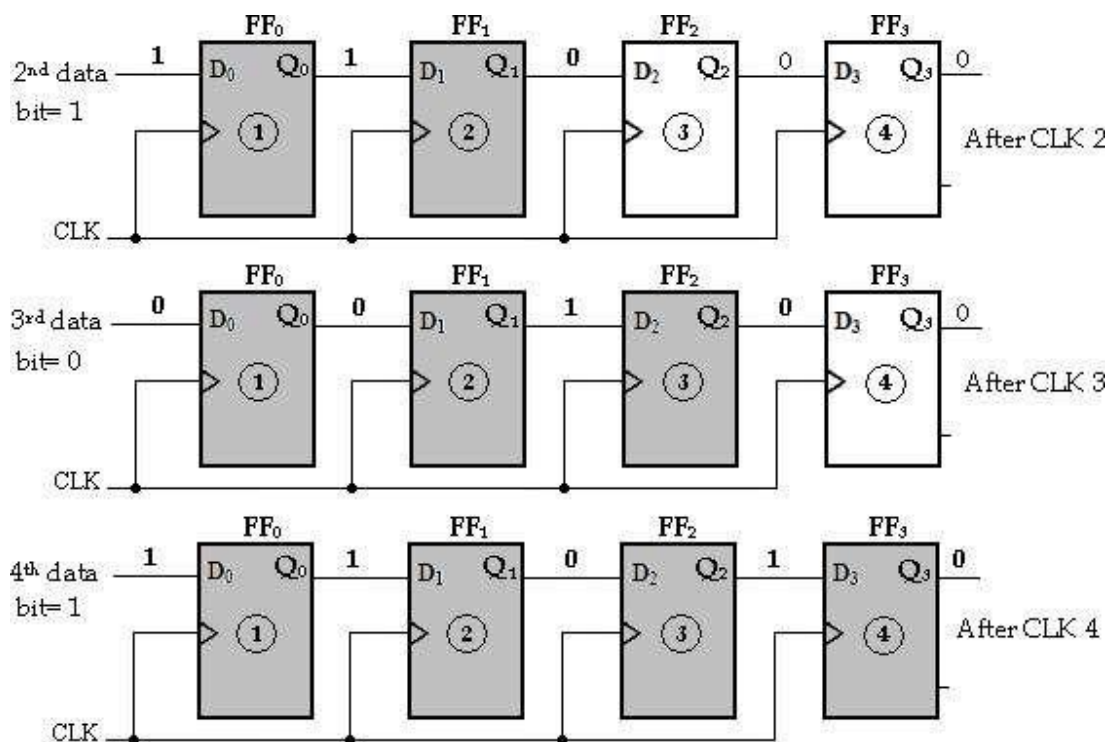


Fig : 3.45 - Four bits (1010) being entered serially into the register

To get the data out of the register, the bits must be shifted out serially and taken off the Q3 output. After CLK4, the right-most bit, 0, appears on the Q3 output.

When clock pulse CLK5 is applied, the second bit appears on the Q3 output. Clock pulse CLK6 shifts the third bit to the output, and CLK7 shifts the fourth bit to the output. While the original four bits are being shifted out, more bits can be shifted in. All zeros are shown being shifted out, more bits can be shifted in.

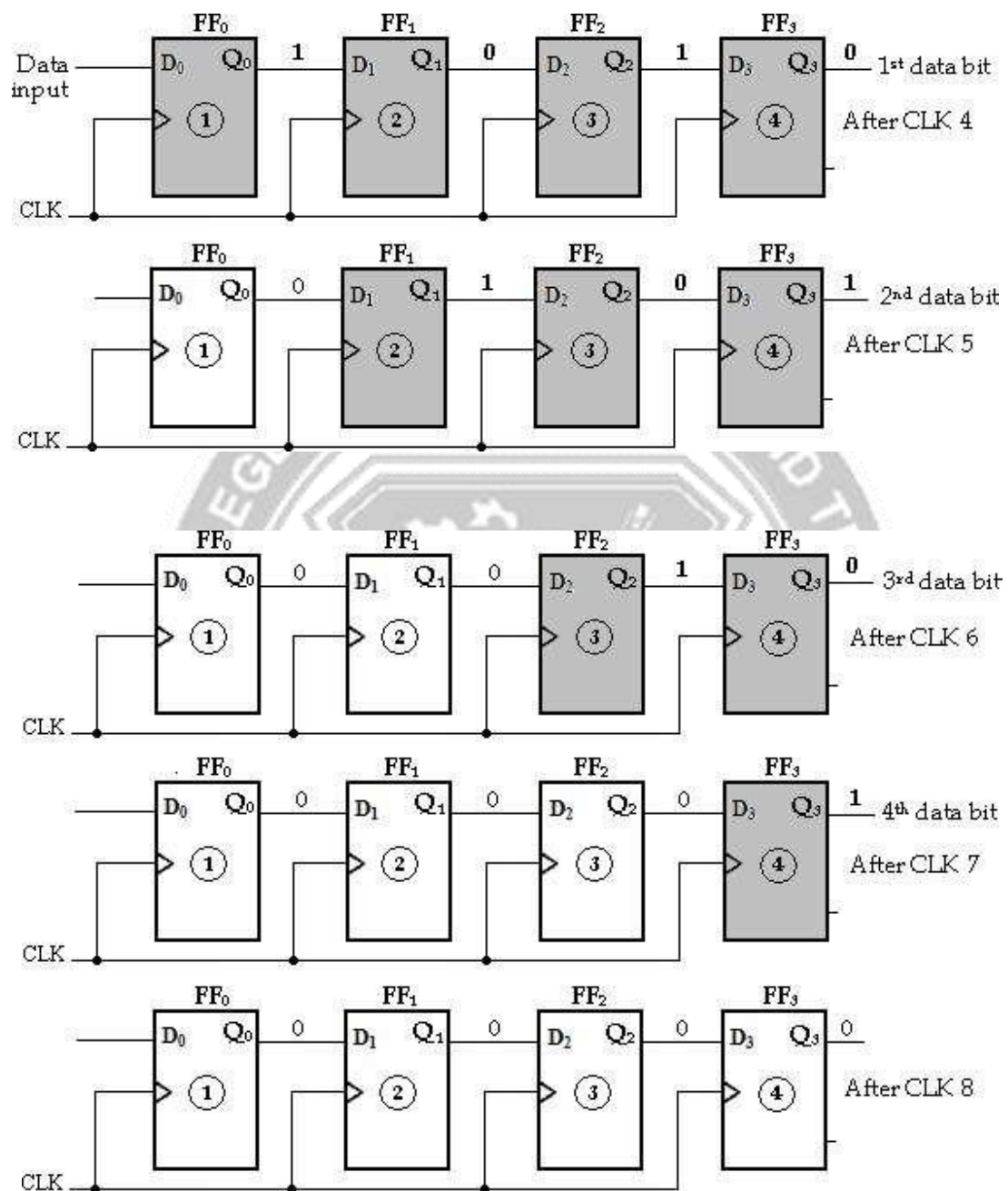


Fig : 3.46 - Four bits (1010) being entered serially-shifted out of the register and replaced by all zeros

Serial-In Parallel-Out Shift Register:

In this shift register, data bits are entered into the register in the same as serial-in serial-out shift register. But the output is taken in parallel. Once the data are stored, each bit appears on its respective output line and all bits are available simultaneously instead of on a bit-by-bit.

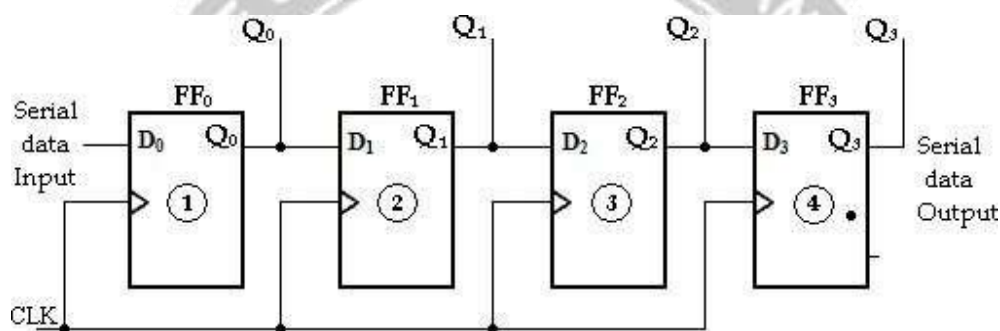
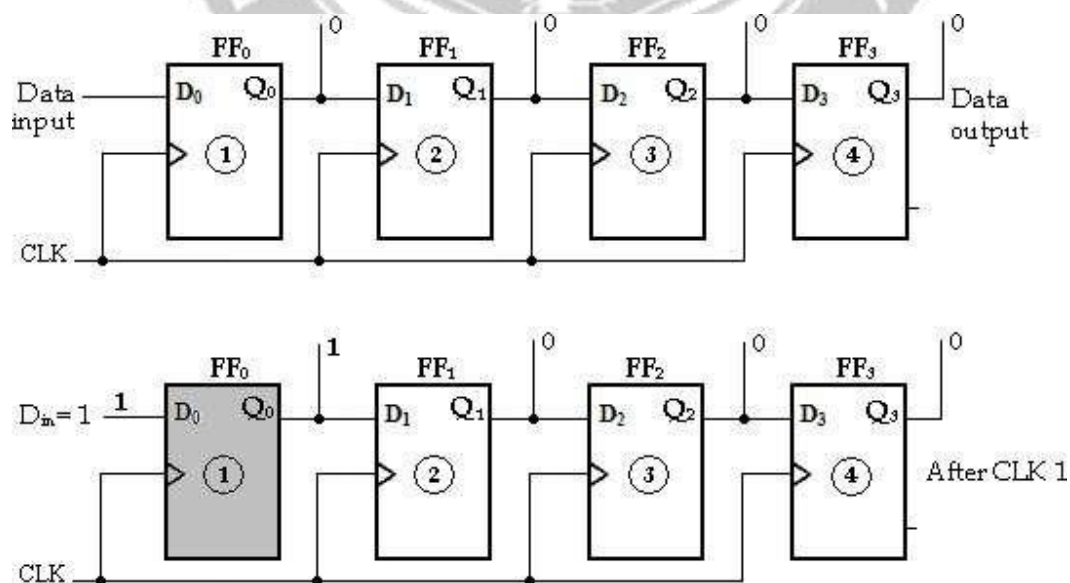


Fig : 3.47 - Serial-In parallel-Out Shift Register



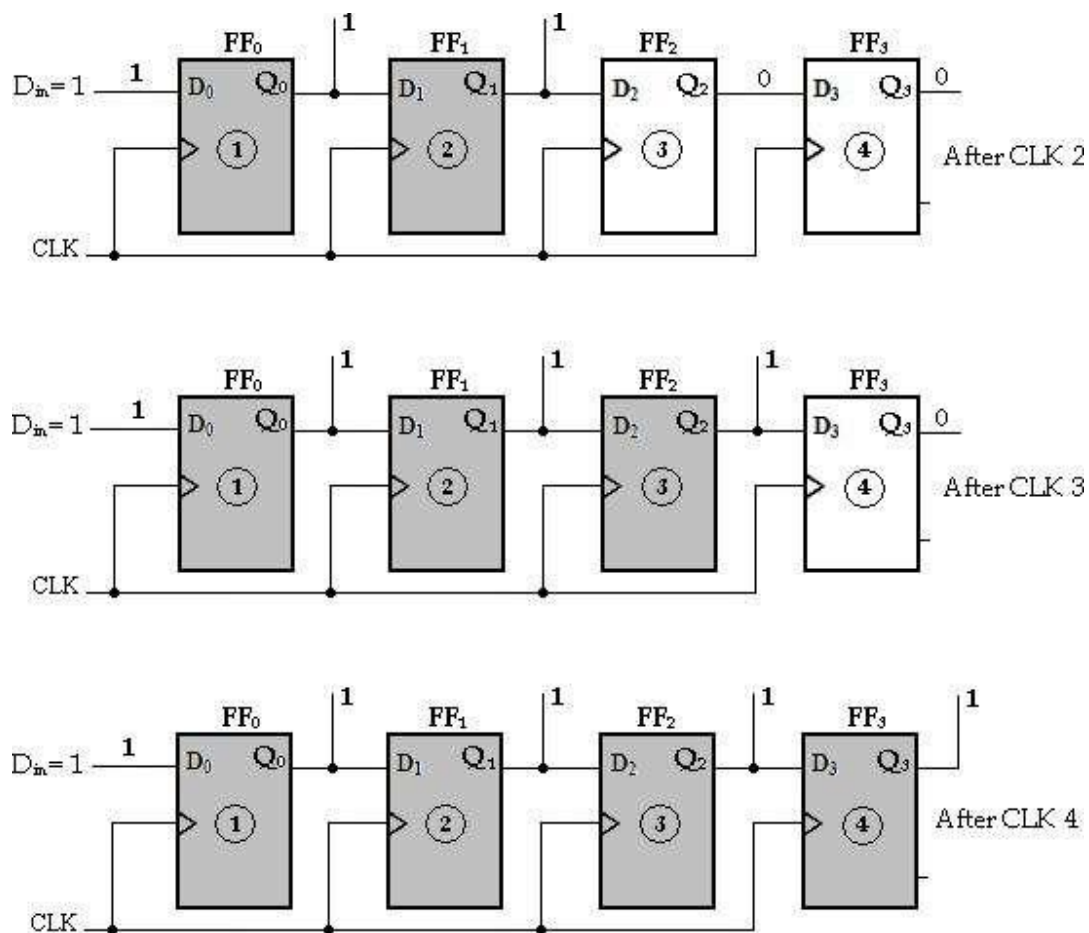


Fig : 3.47 - Four bits (1111) being serially entered into the register

Parallel-In Serial-Out Shift Register:

In this type, the bits are entered in parallel i.e., simultaneously into their respective stages on parallel lines.

A 4-bit parallel-in serial-out shift register is illustrated below. There are four data input lines, X_0 , X_1 , X_2 and X_3 for entering data in parallel into the register. SHIFT/ LOAD input is the control input, which allows four bits of data to **load** in parallel into the register.

When SHIFT/LOAD is LOW, gates G_1 , G_2 , G_3 and G_4 are enabled, allowing each data bit to be applied to the D input of its respective Flip-Flop. When a clock pulse is

applied, the Flip-Flops with $D = 1$ will **set** and those with $D = 0$ will **reset**, thereby storing all four bits simultaneously.

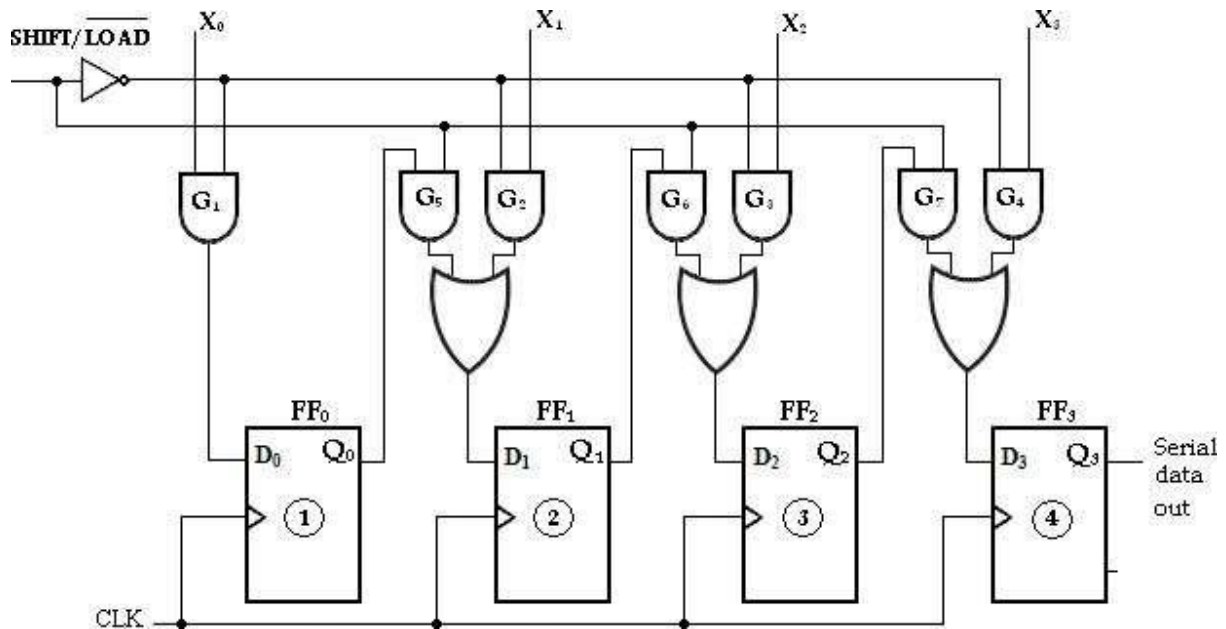


Fig : 3.48 - Parallel-In Serial-Out Shift Register

When SHIFT/LOAD is HIGH, gates G₁, G₂, G₃ and G₄ are disabled and gates G₅, G₆ and G₇ are enabled, allowing the data bits to shift right from one stage to the next. The OR gates allow either the normal shifting operation or the parallel data-entry operation, depending on which AND gates are enabled by the level on the SHIFT/LOAD input.

Parallel-In Parallel-Out Shift Register:

In this type, there is simultaneous entry of all data bits and the bits appear on parallel outputs simultaneously.

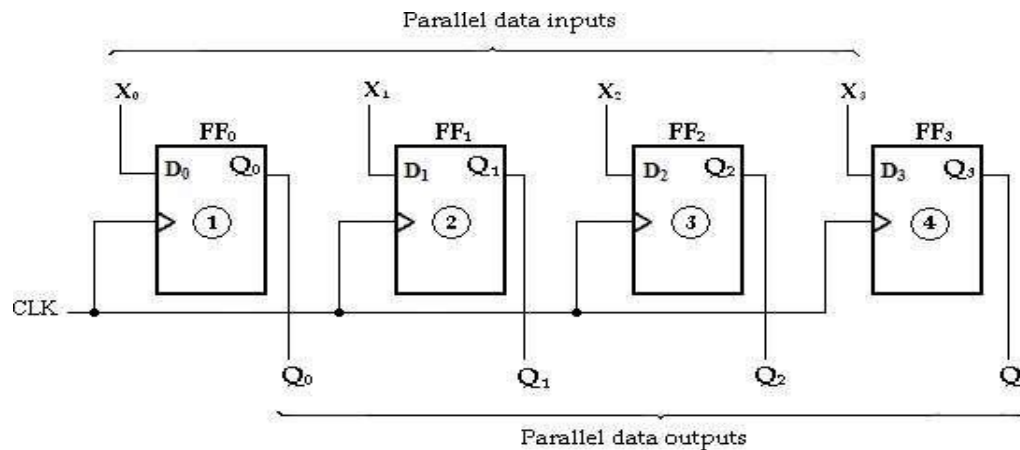


Fig : 3.49 - Parallel-In Parallel-Out Shift Register

UNIVERSAL SHIFT REGISTERS

If the register has shift and parallel load capabilities, then it is called a shift register with parallel load or ***universal shift register***. Shift register can be used for converting serial data to parallel data, and vice-versa. If a parallel load capability is added to a shift register, the data entered in parallel can be taken out in serial fashion by shifting the data stored in the register.

The functions of universal shift register are:

1. A clear control to clear the register to 0.
2. A clock input to synchronize the operations.
3. A shift-right control to enable the shift right operation and the serial input and output lines associated with the shift right.
4. A shift-left control to enable the shift left operation and the serial input and output lines associated with the shift left.
5. A parallel-load control to enable a parallel transfer and the n input lines associated with the parallel transfer.
6. 'n' parallel output lines.

7. A control line that leaves the information in the register unchanged even though the clock pulses are continuously applied.

It consists of four D-Flip-Flops and four 4 input multiplexers (MUX). S_0 and S_1 are the two selection inputs connected to all the four multiplexers. These two selection inputs are used to select one of the four inputs of each multiplexer.

The input 0 in each MUX is selected when $S_1S_0 = 00$ and input 1 is selected when $S_1S_0 = 01$. Similarly inputs 2 and 3 are selected when $S_1S_0 = 10$ and $S_1S_0 = 11$ respectively. The inputs S_1 and S_0 control the mode of the operation of the register.

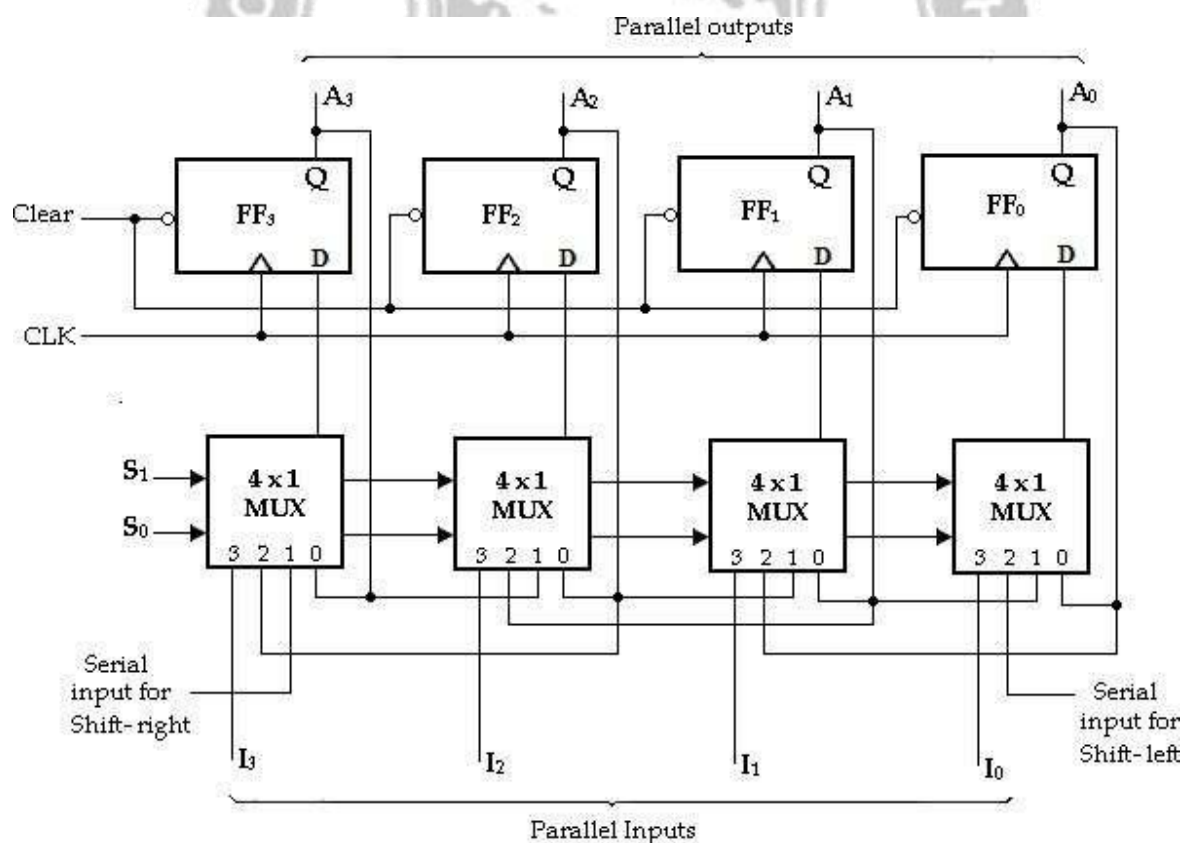


Fig : 3.50 - 4-Bit Universal Shift Register

When $S_1S_0 = 00$, the present value of the register is applied to the D-inputs of the Flip-Flops. This is done by connecting the output of each Flip-Flop to the 0 input of the respective multiplexer. The next clock pulse transfers into each Flip-Flop, the binary value is held previously, and hence no change of state occurs.

When $S_1S_0 = 01$, terminal 1 of the multiplexer inputs has a path to the D inputs of the Flip-Flops. This causes a shift-right operation with the left serial input transferred into Flip-Flop FF_3 .

When $S_1S_0 = 10$, a shift-left operation results with the right serial input going into Flip-Flop FF_1 .

Finally when $S_1S_0 = 11$, the binary information on the parallel input lines (I_1, I_2, I_3 and I_4) are transferred into the register simultaneously during the next clock pulse.

The function table of bi-directional shift register with parallel inputs and parallel outputs is shown below.

Mode Control		Operation
S_1	S_0	
0	0	No change
0	1	Shift-right
1	0	Shift-left
1	1	Parallel load

BI-DIRECTION SHIFT REGISTERS:

A bidirectional shift register is one in which the data can be shifted either left or right. It can be implemented by using gating logic that enables the transfer of a

data bit from one stage to the next stage to the right or to the left depending on the level of a control line.

A 4-bit bidirectional shift register is shown below. A HIGH on the RIGHT/LEFT control input allows data bits inside the register to be shifted to the right, and a LOW enables data bits inside the register to be shifted to the left.

When the RIGHT/LEFT control input is **HIGH**, gates G_1 , G_2 , G_3 and G_4 are enabled, and the state of the Q output of each Flip-Flop is passed through to the D input of the following Flip-Flop. When a clock pulse occurs, the data bits are shifted one place to the right.

When the RIGHT/LEFT control input is **LOW**, gates G_5 , G_6 , G_7 and G_8 are enabled, and the Q output of each Flip-Flop is passed through to the D input of the preceding Flip-Flop. When a clock pulse occurs, the data bits are then shifted one place to the left.

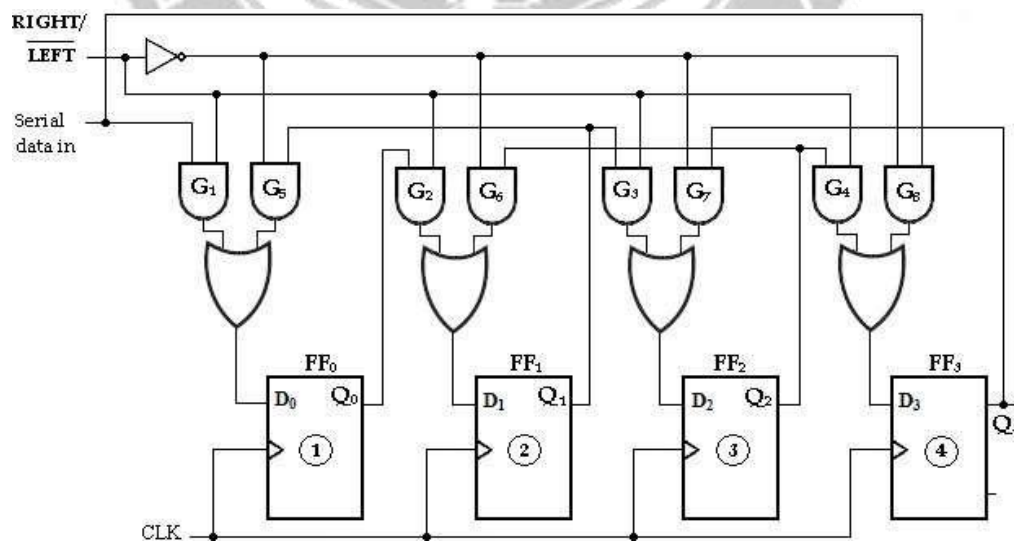


Fig : 3.51 - 4-bit bi-directional shift register