UNIT: 2 – Memory Elements

- **Latches**
- **Flip Flops – D (Clocked and Unclocked) and RS (Clocked and Unclocked)**
- **Registers – Controlled Buffer, Shift – Left, Shift – Right**

Gates are decision-making elements. They can perform binary addition and subtraction. But decision-making elements are not enough. A computer also needs memory elements, devices that can store a binary digit. This unit is about memory elements called latches, flip-flops and registers.

Latch

- A latch is a digital electronic circuit that has two stable states and can store a digital signal.
- These two states are known as ***set*** and ***reset*** state or ***high*** or ***low*** state.
- It stores the data using the feedback lane (loop). The feedback loops are created in circuit diagrams so that output values depend on themselves indirectly.
- Latch in digital electronics can store only 1 bit of data at a time so it can either store set or reset at a particular time interval until the signal is changed.
- Input of a latch works as an electronic switch to control the state of device. Latch circuits have two output stable states, LOW & HIGH.
- A latch's output depends on its ***current and previous inputs***, and its state can change at any time when input changes.
- **Use:** As Latches store data, it is used as memory elements. Latches are used in building flip-flop which is used for designing a sequential logical circuit.

Flip-Flop:

Flip-flop is a basic digital memory circuit, which stores one bit of information. Flip-flops are the fundamental blocks of most sequential circuits. It is also known as a bistable multivibrator or a binary or one-bit memory. Flip-flops are used as memory elements in sequential circuit. The output is obtained in a sequential circuit from combinational circuit or flip-flop or both. The state of flip-flop changes at active state of clock pulses and remains unaffected when the clock pulse is not active. In particular, clocked flip flops serve as memory elements in synchronous sequential Circuits and unclocked flip-flops (i.e., latches) serve as memory elements in asynchronous sequential circuits.

Difference between Latch and Flip-Flop:

Flip – Flop	Latch
Flip-flop utilizes an edge triggering (only changes state when a control signal goes from high to low or low to high) approach.	Latch follows a level triggering (outputs can change as soon as the inputs change) approach.
The clock signal is present.	The clock signal is absent.
You can design it using Latches along with a clock.	You can design it using Logic gates.
Flip-flop is sensitive to the applied input and the clock signal.	Latches are sensitive to the applied input signal-only when enabled.
It has a slow operating speed.	It has comparatively fast operating speed.
You can classify a flip-flop into a synchronous or asynchronous flip-flop.	A user cannot classify the Latch this way.
Flip-Flops work using the binary input and the clock signal.	Latches operate only using binary inputs.
It requires more power.	It requires comparatively less power.
It is quite easy to perform circuit analysis.	Analyzing the circuit is quite complex.

RS Flip-Flop:

R stands for **reset** add **S** stands for ***set***. There are two inputs **R** is used for reset and **S** is used for set. The output of RS depends on current as well as previous state. And its state changes as soon as input change. The RS flip-flop can either be designed either using NOR latch or using NAND latch.

UnClocked Flip-Flop:

The working of RS flip-flop depends on inputs S – Set & R – Reset.

The two useful states are as follows:

If output Q = 1 and $\overline{Q}$ = 0 then latch is in set state.

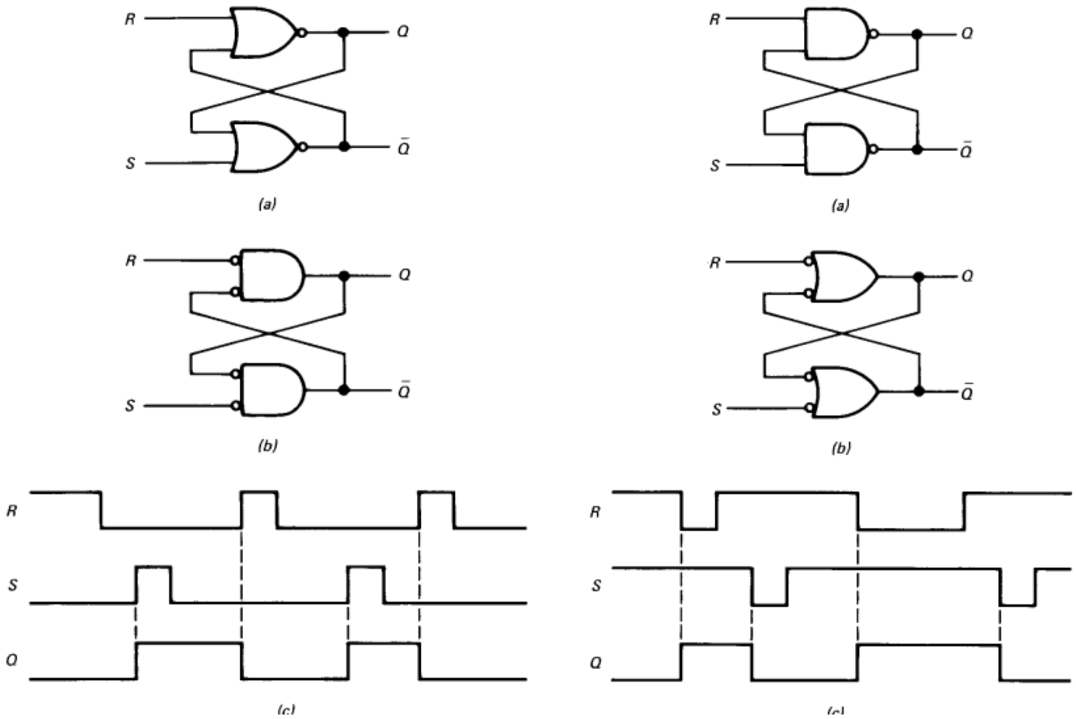
If output Q = 0 and $\overline{Q}$ = 1 then latch is in reset state.

If the circuit is designed using NOR gate then for both S = R = 0 then Q and $\overline{Q}$ will retain the previous state represented as $\overline{Q}$ which is said to be No change and S = R= 1, then the output states are said to be invalid or undefined or race condition.

If circuit is designed using NAND gate, then for both S = R = 1 and S = R = 0, then the output states will be reverse of the NOR gate.

R	S	Q	Comment
0	0	NC	No change
0	1	1	Set
1	0	0	Reset
1	1	*	Race

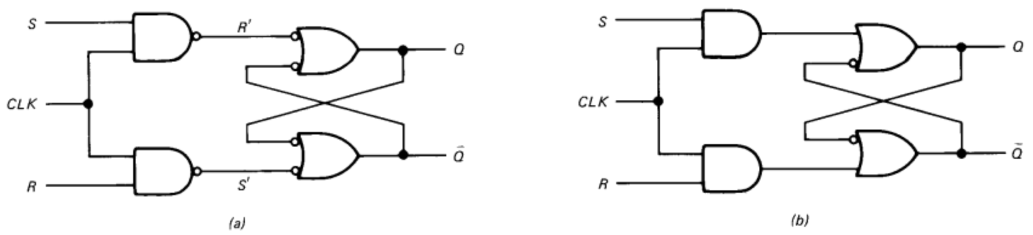
R	S	Q	Comment
0	0	*	Race
0	1	1	Set
1	0	0	Reset
1	1	NC	No change

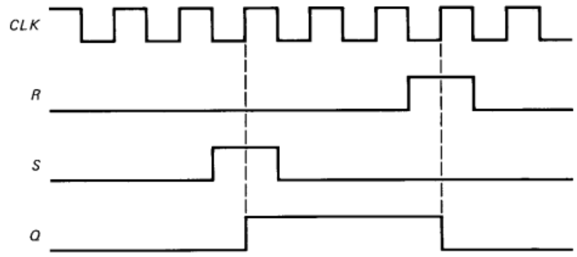


UnClocked RS Flip–Flop

Clocked Flip-Flop:

In the clocked RS flip flop the appropriate levels applied to their inputs are blocked till the receipt of a pulse from another source called a clock. The flip-flop changes state only when the clock pulse is applied depending upon the inputs. The basic circuit is shown in Figure 2. This circuit is formed by adding two AND gates at inputs to the R-S flip flop. In addition to control inputs Set (S) and Reset (R), there is a clock input (CLK) also.

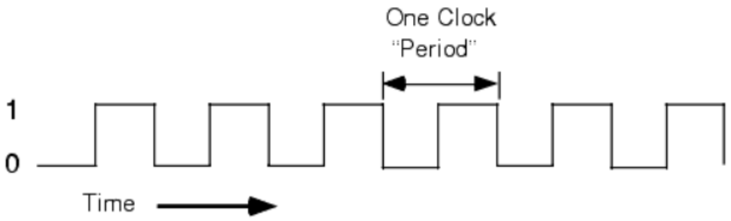




What is Clock Signal?

A clock signal shown in figure is a particular type of signal that oscillates between a high and low state. With the signal acting as a metronome, the digital circuit follows in time to coordinate its sequence of actions. Digital circuits rely on clock signals to know when and how to execute the functions that are programmed.

If the clock in a design is like the heart of an animal, then clock signals are the heartbeats that keep the system in motion.



Truth table of Clocked Flip-Flop using NAND Latch and NOR Latch

NAND Latch			
CLK	**R**	**S**	**Q**
0	0	0	NC
0	0	1	NC
0	1	0	NC
0	1	1	NC
1	0	0	NC
1	0	1	1
1	1	0	0
1	1	1	*

NOR Latch			
CLK	**R**	**S**	**Q**
0	0	0	NC
0	0	1	NC
0	1	0	NC
0	1	1	NC
1	0	0	*
1	0	1	1
1	1	0	0
1	1	1	NC

D Flip-Flop:

Since the RS flip-flop is susceptible to a race condition, we will modify the design to eliminate the possibility of a race condition. The result is a new kind of flip-flop known as a D latch.

Unclocked:

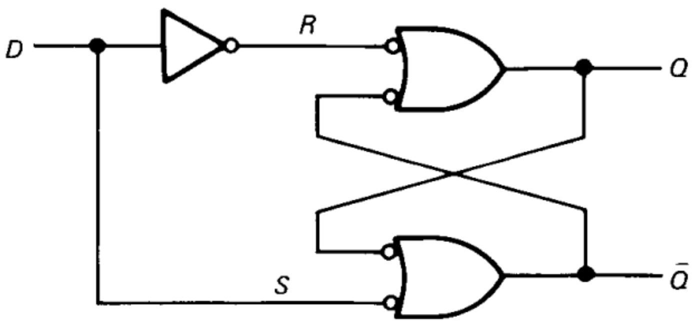
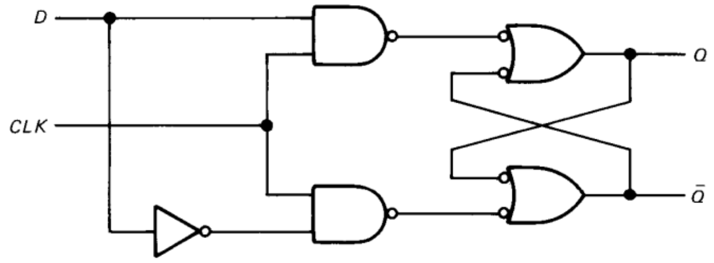


Figure shows one way to build a D latch. Because of the inverter, data bit ***D*** drives the ***S*** input of a NAND latch and the complement D drives the R input. Therefore, a high D sets the latch, and a low D reset it which is shown in below table:

D	Q
0	0
1	1

Clocked:



(a)

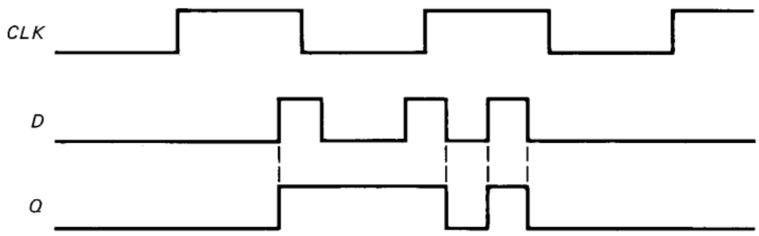


Figure is level-clocked. A low CLK disables the input gates and prevents the latch from changing states. In other words, while CLK is low, the latch is in the inactive state and the circuit stores or remembers. When CLK is high, D controls the output. A high D sets the latch, while a low D reset it.

CLK	D	Q
0	X	NC
1	0	0
1	1	1

Above table summarizes the operation. X represents a don't care condition; it stands for either 0 or 1. While CLK is low, the output cannot change, no matter what D is. When CLK is high, however, the output equals the input summarizes the operation of the D latch. Especially important, there is no race condition in this truth table. The inverter guarantees that Sand R will always be in opposite states; therefore, it's impossible to set up a race condition in the D latch.

$$Q = D$$

In figure shows a timing diagram. If the clock is low, the circuit is latched and the Q output cannot be changed. While the clock is high, however, Q equals D; when D goes high, Q goes high; when D goes low, Q goes low. The latch is transparent, meaning that the output follows the value of D while the clock is high.

Disadvantage: Because the D latch is level-clocked, it has a serious disadvantage. While the clock is high, the output follows the value of D. Transparent latches may be all right in some applications but not in the computer circuits.

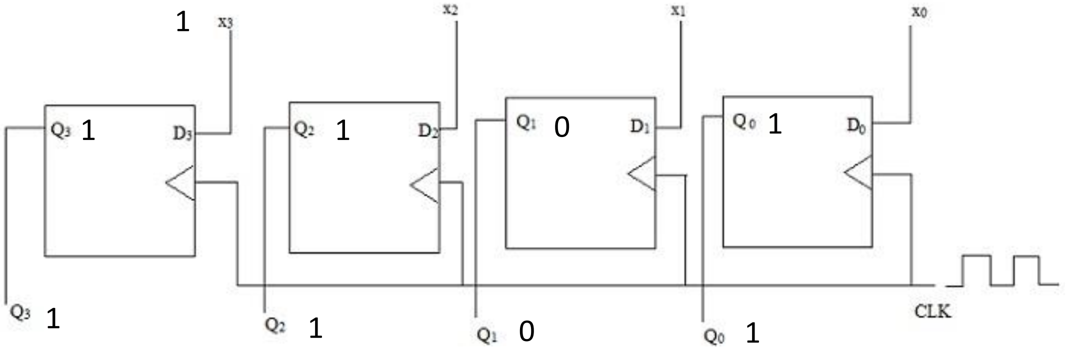
Registers:

A register is a group of memory elements (Flip-Flops) that work together as a unit (Sequential Circuit). The outputs of the sequential circuits completely depend on the present inputs and previous outputs. The simplest registers do nothing more than store a binary word; others modify the stored word by shifting its bits left or right. Its basic function is to hold information within a digital system so as to make it available to the logic units during the computing process.

Buffer Registers:

Purpose of Buffer Registers: It prevents the high-speed processor from being locked to a slow I/O device during a sequence of data transfer or reduces speed mismatch between faster and slower devices.

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 $B_1B_2B_3B_4$ 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. $Q_0 = X_0$, $Q_1 = X_1$, $Q_2 = X_2$ and $Q_3 = X_3$ as shown by Figure:

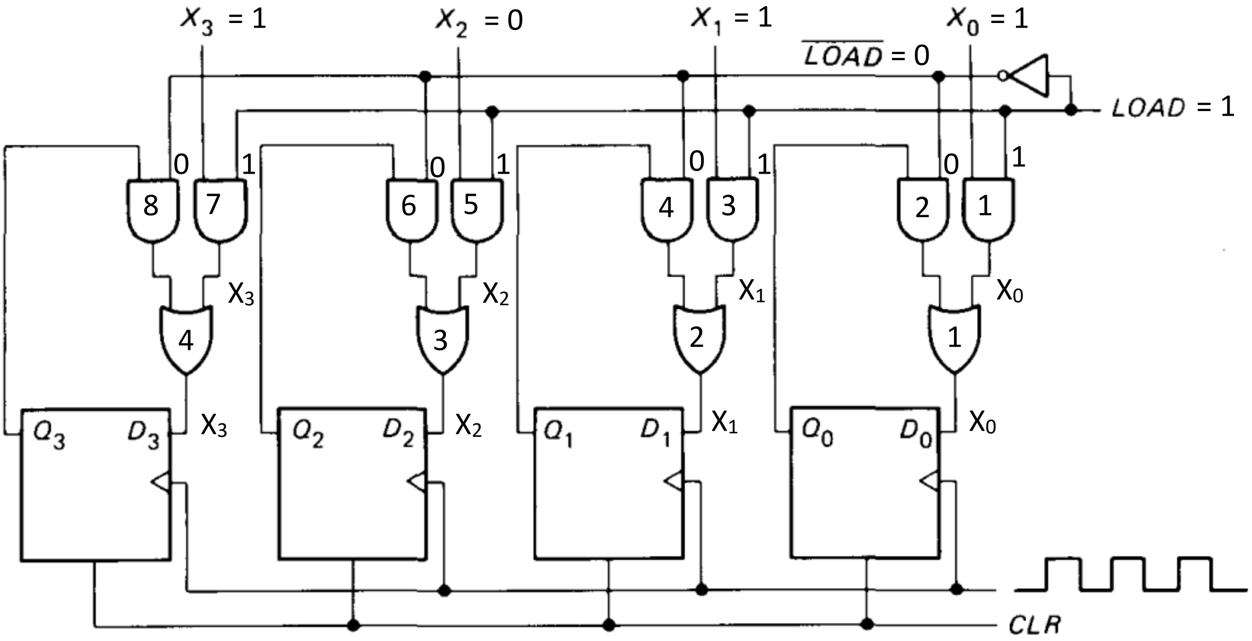


In the simple buffer register whatever the inputs ($X_3X_2X_1X_0$) given to the D-Flip-Flop it will stores in it, so there will be no control on inputs ($X_3X_2X_1X_0$), and there for the controlled buffer register was invented.

Controlled Buffer Register:

Below figure is a controlled buffer register with an active-high CLR. Therefore, when CLR is Enable or goes High, all flip-flops are reset and the stored word becomes.

$$Q_0Q_1Q_2Q_3 = 0000$$

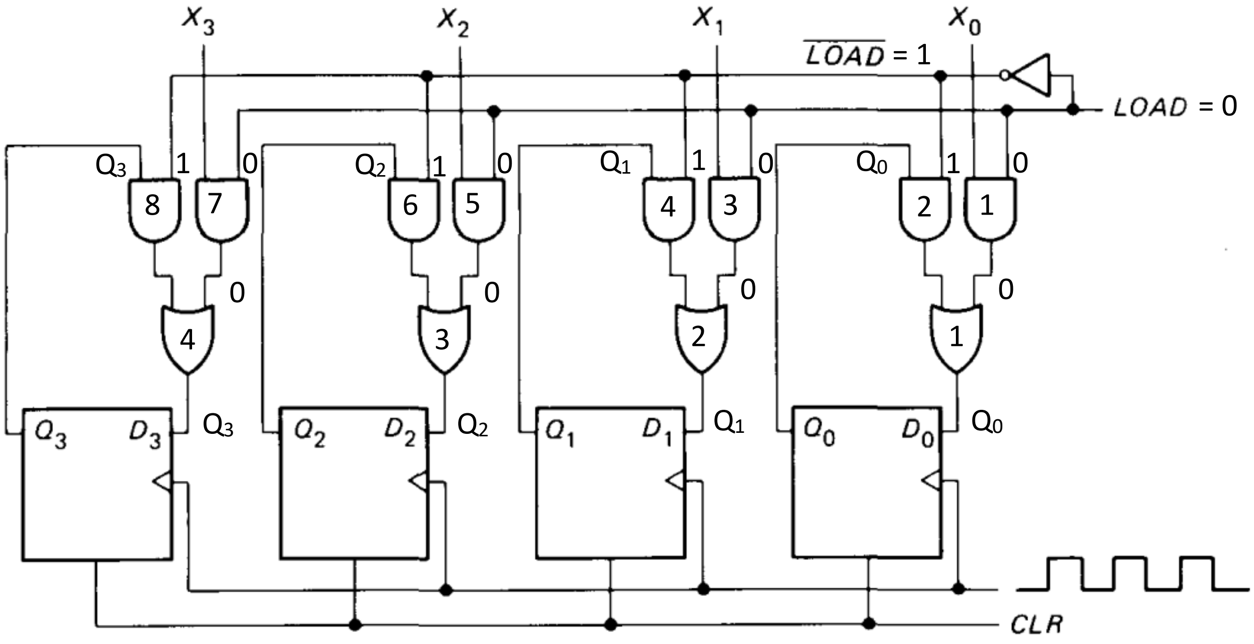


When CLR goes low or disable, the register is ready for action. LOAD is a control input; it determines what the circuit does. When LOAD is low, the X bits cannot reach the flip-flops means no data will be stored in Flip-Flops. At the same time, the inverted signal $\overline{\text{LOAD}}$ is high;

this forces each flip-flop output to feed back to its data input. When each rising clock edge arrives, data is circulated or retained. In other words, the register contents are unchanged when LOAD is low. When LOAD goes high, the X bits are transmitted to the data inputs. After a short setup time, the flip-flops are ready for loading. With the arrival of the positive clock edge, the X bits are loaded and the stored word becomes

$$Q_3Q_2Q_1Q_0 = X_3X_2X_1X_0$$

If LOAD returns to low, the foregoing word is stored indefinitely; this means that the X bits can change without affecting the stored word.



CLR	Enable	High	All Flip-flops are reset	Q = $Q_3Q_2Q_1Q_0$ = 0000
CLR	Disable	Low	All Flip-flops are ready to Operate	
LOAD	High	1	The input ($X_3X_2X_1X_0$) is loaded into register	Q = $Q_3Q_2Q_1Q_0$ = 1011
LOAD	Low	0	The contents ($Q_3Q_2Q_1Q_0$) of the register remains unchanged.	Q = $Q_3Q_2Q_1Q_0$ = 1011

Shift-Left Register:

The register that shifts bits of stored word towards left, known as shift-left register, is shown below. As is clear from circuit,

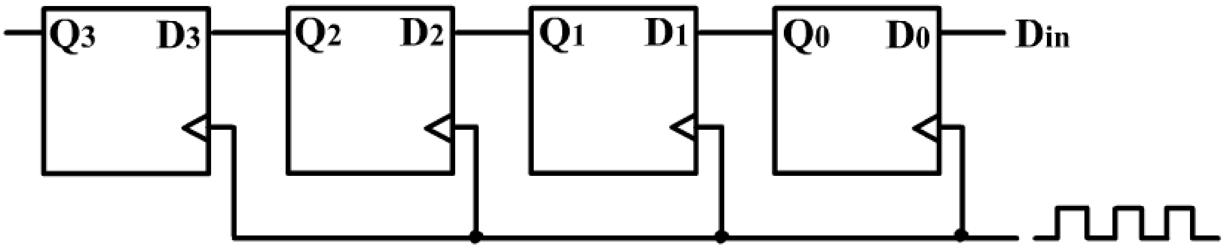
data input D_{in} sets up first flip-flop

Q_0 output of this flip-flop set up second flip-flop

Q_1 sets up third

Q_2 sets up fourth

Data is given to input of first flip-flop, i.e., D_{in} and output is got simultaneously from all flip-flops, circuit is called as serial-in/parallel-out.



Working of shift-left registers can be understood by following example:

Consider that input data D_{in} is 1, i.e., input to flip-flop-1, D_0 = 1 and initial output

Q = 0000

That is, initially inputs to all other three flip-flops are 0. Now with arrival of first rising clock edge sets the right flip-flop, and the output of Q_0 becomes 1, and stored word becomes

Q = 0001

This new word means DI now equals 1, as well as Do. When the next positive clock edge hits, the QI flip-flop sets and the register contents become

Q= 0011

The third positive clock edge results in

Q= 0111

and the fourth rising clock edge gives

Q = 1111

Stored word is therefore 1111 and it remains unchanged so long as D_{in} = 1. Though, if D_{in} = 0, then with successive CLK pulses register output or content becomes

At 1st CLK Q = 1110

At 2nd CLK Q = 1100

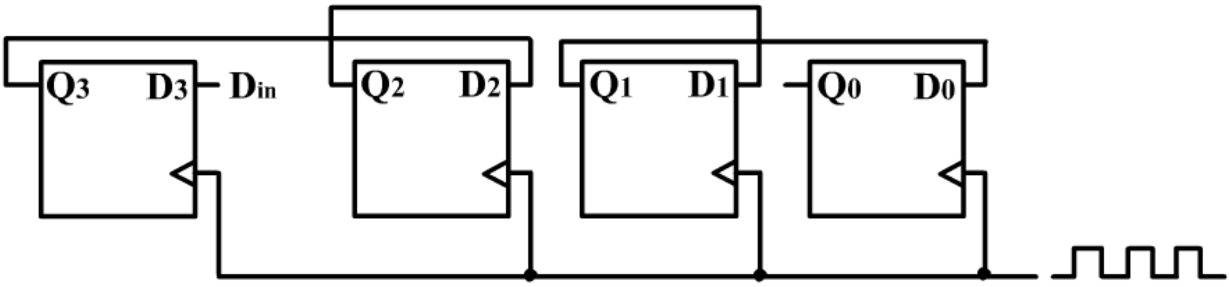
At 3rd CLK Q = 1000

At 4th CLK Q = 0000

This word 0000 remains stored so long as $D_{in} = 0$. Entire operation of shift-left register in terms of its tuning diagram is shown figure.

Shift-Right Register:

Below figure is a shift-right register. As shown, each Q output sets up the D input of the preceding flip-flop. Data input, Din is given to input of fourth flip-flop as D_3. Q output of each flip-flop is fed back to D input of previous flip-flop, i.e. Q_3 is given to D_2, Q_2 is given to D_1 and Q_1 is given to D_0.



Here is an example with Din = 1 and

Q = 0000

All data inputs except the one on the left are 0s. The first positive clock edge sets the left flip-flop and the stored word becomes

Q = 1000

With the appearance of this word, D_3 and D_2 are 1s. The second rising clock edge gives

Q = 1100

The third clock pulse gives

Q = 1110

and the fourth clock pulse gives

Q = 1111

Suppose D_{in} is now changed to 0

Then clock pulse produce

Q = 0111

Q = 0011

Q = 0001

Q = 0000 as long as $D_{in} = 0$