3.3.4 Flip-flops

What exactly is memory?

- **A memory should have at least three properties :**

  1. It should be able to hold a value.
  2. You should be able to **read** the value that was saved
  3. You should be able to **change** the value that was saved.

We shall start with the simplest case, a one-bit memory.

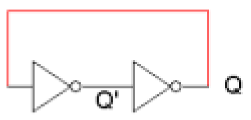
1. It should be able to hold a single bit, 0 or 1.
2. You should be able to read the bit that was saved.
3. You should be able to change the value.
   Since there is only one bit, there are only 2 options:
   - **Set** the bit to 1
   - **Reset**, or **clear**, the bit to 0.

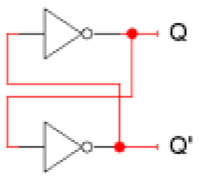
The basic idea of storage

- How can a circuit "remember" anything, when it's just a bunch of gates that produce outputs according to the inputs?

- The basic idea is to make a loop, so that the circuit outputs are also the inputs.

- Here is one initial attempt, shown with two equivalent layouts:





- **Does this satisfy the properties of memory?**

  - These circuits "remember" Q, because its value never changes. (Similarly Q' never changes either.)
  - We can also "***read***" Q, by attaching a light probe or another circuit.
  - But we can't **change** Q! There are no external inputs here, so we can't control whether Q=1 or Q=0.

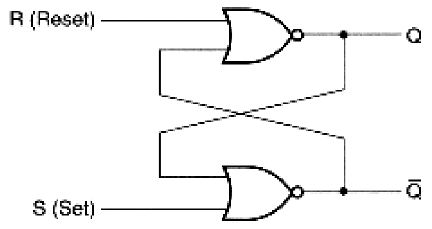
3.3.4 Flip-flops

The Latch:

The most common memory element used is the **latch** which is made up of an assembly of logic gates. Even though a logic gate by itself has no storage capacity, several can be connected together in ways that permit information to be stored.

A storage element maintains binary state indefinitely (as long as power is applied), until directed by an input signal to switch to its alternated state.

- The S-R (Set-Reset) latch is the most basic type. It can be constructed from **NOR** gates or **NAND** gates. With **NOR** gates, the latch responds to active-HIGH inputs; with NAND gates, it responds to active-LOW inputs.

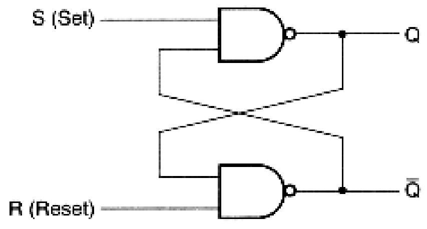


(a) Logic diagram

S	R	Q	$\bar{Q}$	
1	0	1	0	Set state
0	0	1	0	
0	1	0	1	Reset state
0	0	0	1	
1	1	0	0	Undefined

(b) Function table

NOR Active-HIGH Latch



(a) Logic diagram

S	R	Q	$\bar{Q}$	
0	1	1	0	Set state
1	1	1	0	
1	0	0	1	Reset state
1	1	0	1	
0	0	1	1	Undefined

(b) Function table

NAND Active-LOW Latch

- Note that when S and R simultaneously change from their asserted state to their deasserted states, the latch enters an unstable state when its outputs oscillate between two binary states indefinitely

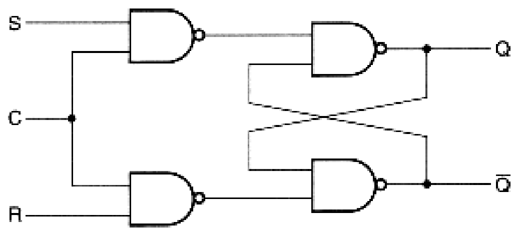
Or in simpler words, the latch cannot **SET** and **RESET** at the same instant.

3.3.4 Flip-flops

Gated S-R Latch

One way to prevent the system from becoming unstable is by means of gating the set and reset inputs using a (**control**) input.

- The S and R input control the state to which the latch will go when a **HIGH** level is applied to the C input.
- The latch will not change until C is **HIGH**, but as long as it remains **HIGH**, the output is controlled by the state of the S and R inputs.
- In this circuit, the invalid state occurs when both S and R are simultaneously **HIGH** while the latch is enabled (Control input is **HIGH**)



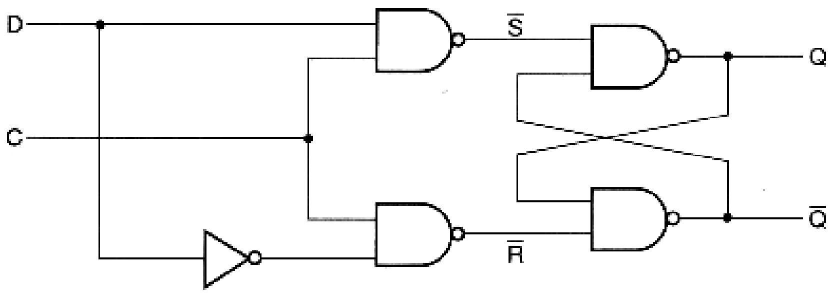
(a) Logic diagram

C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	Q = 0; Reset state
1	1	0	Q = 1; Set state
1	1	1	Undefined

(b) Function table

The gated D Latch

- Another way of eliminating the undesirable unstable state (Undefined state) is by means of making sure that both **set** and **reset** signals are never active at the same instant. Giving rise to what is known as the **D latch**.



(a) Logic diagram

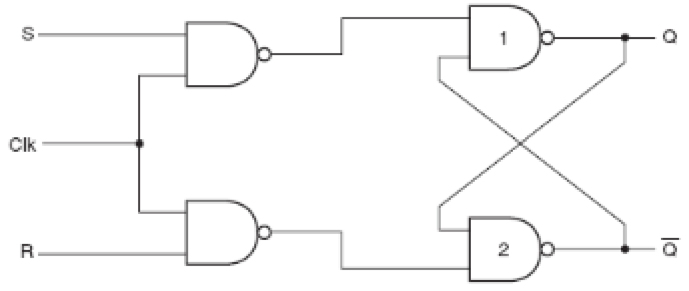
C	D	Next state of Q
0	X	No change
1	0	Q = 0; Reset state
1	1	Q = 1; Set state

(b) Function table

3.3.4 Flip-flops

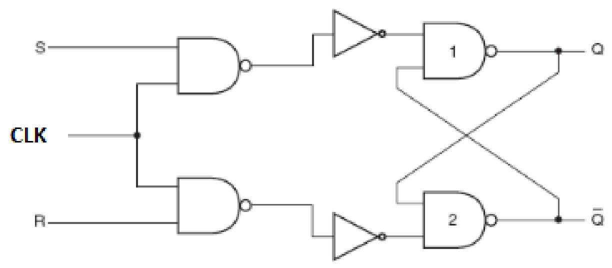
Clocked Latch (flip-flop):

- Flip flops are actually an application of logic gates. They can be considered as the most basic idea of a Random Access Memory (RAM).
- A flip-flop differs from a latch in the manner that it changes states constantly. It is a clocked device, in which only the clock state determines when a new bit is entered
- A clock is simply a varying Enable signal that alternates between “0” and “1”



S	R	Clk	Q OUTPUT
0	0	0	No Change
0	0	1	No Change
0	1	0	No Change
0	1	1	0
1	0	0	No Change
1	0	1	1
1	1	0	No Change
1	1	1	Invalid

CLOCKED R-S FLIP-FLOP WITH ACTIVE HIGH INPUT



S	R	Clk	Q_{n+1}
0	0	0	Q_n
0	0	1	Invalid
0	1	0	Q_n
0	1	1	1
1	0	0	Q_n
1	0	1	0
1	1	0	Q_n
1	1	1	Q_n

Next State

CLOCKED R-S FLIP-FLOP WITH ACTIVE LOW INPUT

- When the clock signal is HIGH, the two NAND gates are enabled and the S and R inputs are passed on to flip-flop inputs with their statuses complemented. The outputs can now change states as per the status if R and S at the flip-flop inputs.

3.3.4 Flip-flops

The JK Flip Flop

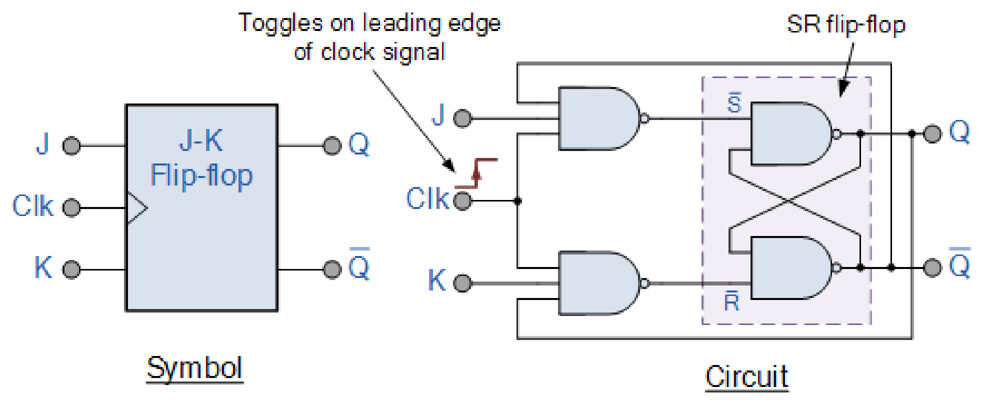
- A J-K flip-flop behaves in the same fashion as a R-S flip-flop except for one of the entries in the function table.
- In the case of an R-S flip-flop, the input combination S=R=1 is prohibited.
- In the case of a J-K flip-flop with active HIGH inputs, the output of the flip-flop **"toggles"**.
- Thus, a J-K flip-flop overcomes the problem of a forbidden input combination of the R-S flip-flop.
- This is achieved by a minor addition in the R-S flip-flop circuit.

The simple JK flip-flop is the most widely used of all the flip-flop designs and is considered to be a universal flip-flop circuit. The sequential operation of the JK flip flop is exactly the same as for the R-S flip flop. The difference this time is that the JK flip flop has no invalid or forbidden states even when the "J" and "K" are both logic "1"

The **JK flip flop** is basically a gated SR Flip-flop with the addition of a clock input that prevents the illegal output condition from occurring. The JK flip flop has 4 possible input combinations "logic 1", "logic 0", "no change" and "toggle".

3.3.4 Flip-flops

The Basic JK Flip-flop



Both the S and the R inputs of the previous R-S flip flop have now been replaced by 2 inputs called the "J" and "K" inputs, respectively after its inventor **Jack Kilby**. Then this equates to J=S and K=R.

The two 2-input AND gates of the R-S flip flop has now been replaced by two 3-input NAND gates with the third input of each gate connected to the outputs at Q and Q'. This cross coupling of the R-S flip flop allows the invalid condition of "S=1" and "R=1" to produce a "toggle action" as the two inputs are now interlocked.

If the circuit is now "SET" and the J input is inhibited by the "0" status of the Q' through the lower NAND gate. If the circuit is "RESET" the K input is inhibited by the "0" status of Q through the upper NAND gate. As Q and Q' are always different, we can use them to control the input. When both inputs J and K are equal to logic "1", the JK flip flop toggles as shown in the following truth table.

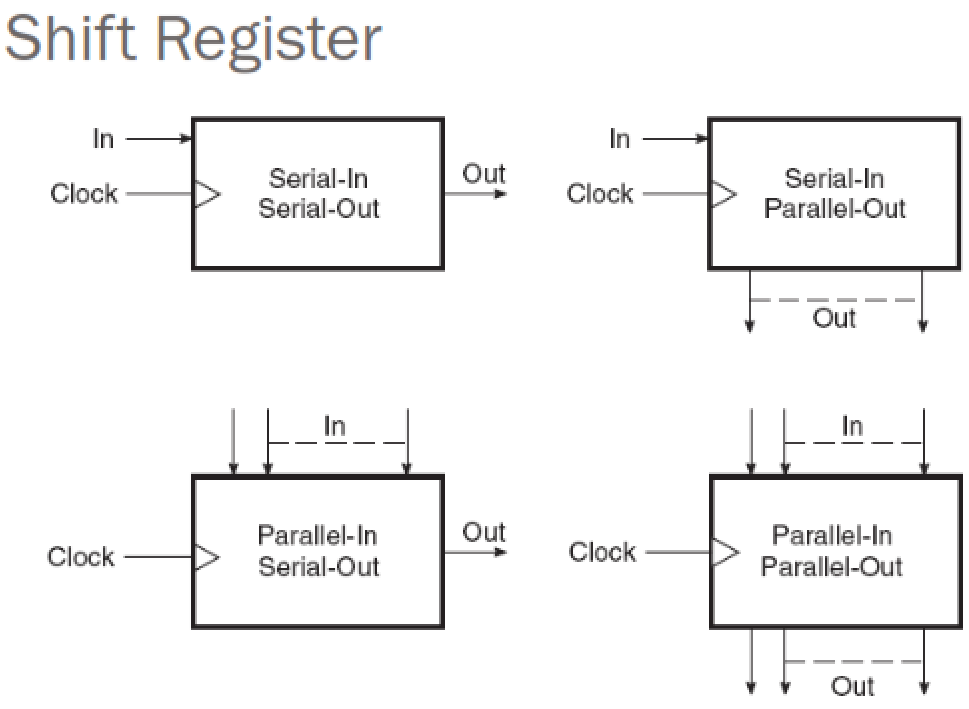
The Truth Table for the JK Function

	Input		Output		Description
	J	K	Q	$\overline{Q}$	
same as for the SR Latch	0	0	0	0	Memory no change
	0	0	0	1	
	0	1	1	0	Reset Q » 0
	0	1	0	1	
	1	0	0	1	Set Q » 1
	1	0	1	0	
toggle action	1	1	0	1	Toggle
	1	1	1	0	

3.3.4 Flip-flops

Flip-flops as data storage elements

- A flip flop stores one bit at a time in a digital circuit. In order to store several bits, flip flops can be connected together in series or in parallel to form a structure called **shift registers**.
- A **shift register** is a digital device used for storage and transfer of data. The shift register forms an important link between the main digital system and the input/output channels.
- Since each flip-flop can store only 1 bit of data, the storage capacity of the shift register depends on the number of flip-flops used.
- Based on the method used to load data onto and read data from shift registers, they are classified as :
  - Serial-in serial-out (SISO) shift registers
  - Serial-in parallel-out (SIPO) shift registers
  - Parallel-in serial-out (PISO) shift registers
  - Parallel-in parallel-out (PIPO) shift registers



3.3.4 Flip-flops

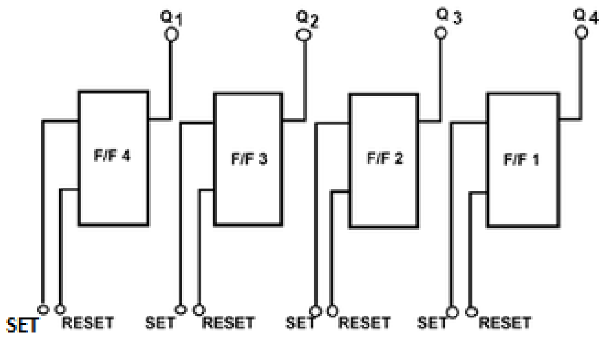
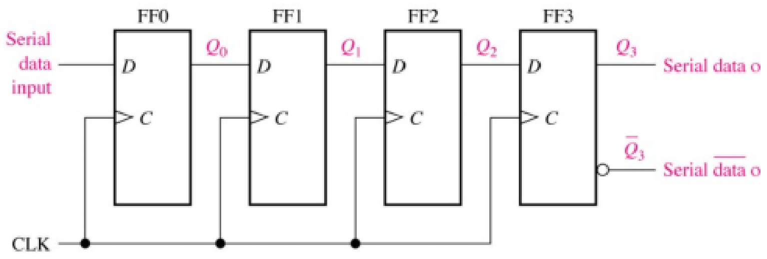


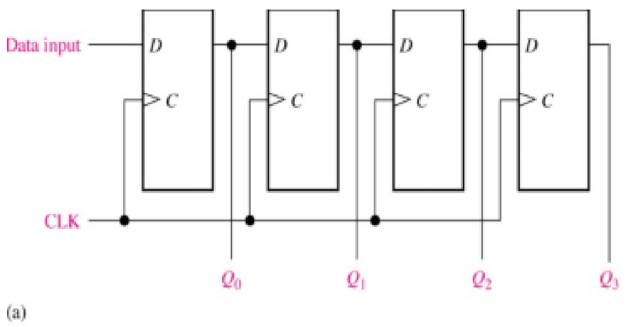
Figure 1: Bit Binary Register

Figure 1 shows a 4 bit register. Any number from (0000)2 to (1111)2 may be stored in it simply by setting or resetting the appropriate flip flops. Let us suppose that flip flop one is SET (1), flip flop two is RESET (0), flip flop three is RESET (0) and flip flop 4 is SET (1), the binary number stored in this register is (1001)2.

Serial in/Serial Out Shift Registers

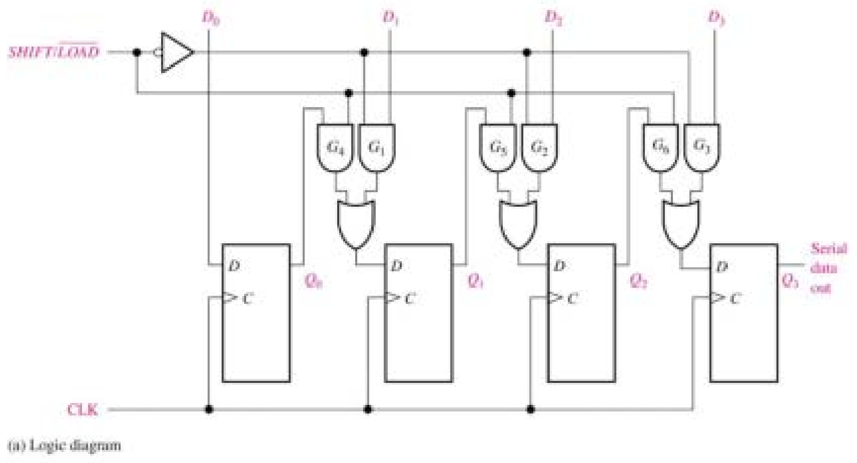


Serial in/Parallel Out Shift Registers

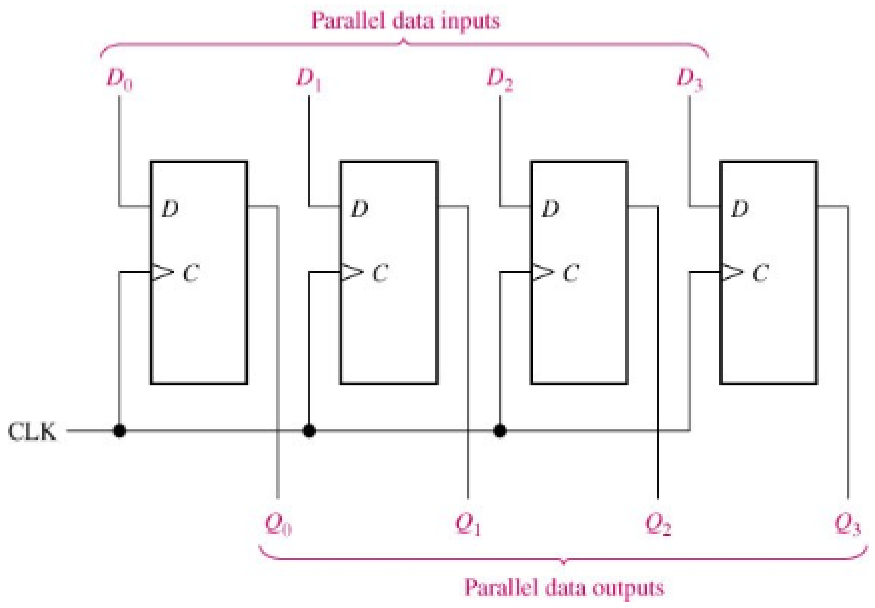


3.3.4 Flip-flops

Parallel in/ Serial out Shift Register



Parallel in/ Parallel out Shift Register:


Parallel data inputs
D_0
D_1
D_2
D_3
D
C
CLK
Q_0
Q_1
Q_2
Q_3
Parallel data outputs