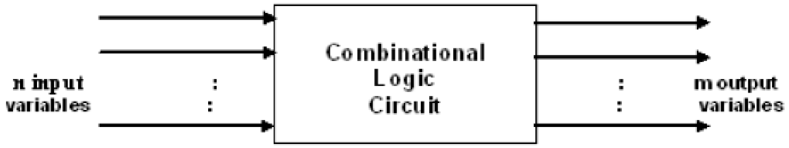
# UNIT III

## 3.1 COMBINATIONAL & SEQUENTIAL CIRCUITS

- Digital logical circiuts are basically categorized into two types
  1. Combinational circuits
  2. Sequential circuits
- A combinational circiuts transforms binay information from the given input data to required output data
- It is represent by truth table with n inputs and m outputs.
- The design of combinational circuits starts from the verbal outline of the problem and ends in a logic circuit diagram
  1. The procedure involves the following steps
  2. The problem is stated
  3. The input and output variables are assigned letter symbols
  4. The truth table that defines the relationship between inputs and outputs is derived.
  5. The simplified boolean functions for each output are obtained.
  6. The logic diagram is drawn

### 3.1.1 BLOCK DIAGRAM OF A COMBINATIONAL CIRCUITS

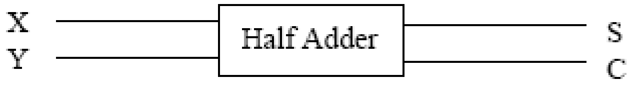


### 3.1.2 HALF ADDER

- A basic module used in binary arithmetic elements is the half-adder.
- The function of the half-adder is to add two binary digits, producing a sum
- according to the binary addition rules shown in the table.

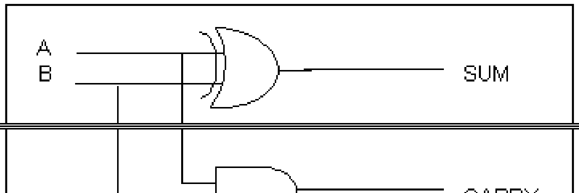
| INPUT | SUM OF BITS |
|---|---|
| 0+0 | 0 |
| 0+1 | 1 |
| 1+0 | 1 |
| 1+1 | 0 With a carry of 1 |

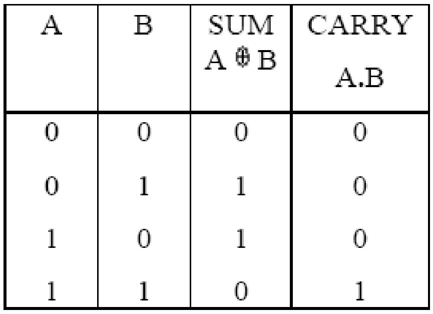
Addition Table



- Two inputs are designated as X and Y and two outputs, designated as S and C.
- The half-adder perform binary addition operation for two binary inputs as shown in table.
- This is arithmetic addition, not logical or Boolean algebra addition.

Half Adder – Block Diagram





| A | B | SUM<br>A ⊕ B | CARRY<br>A.B |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

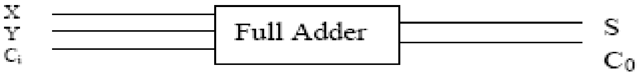
ble

- In the half-adder diagram there are two inputs to the half-adder and two outputs.
- If either of the inputs is a 1 but not both, then the output on the S line will be a 1.
- If both inputs are 1s, the output on the C line will be a 1.
- For all other states, there be a 0 output on the carry line. These relationships may be written in Boolean form as follows.

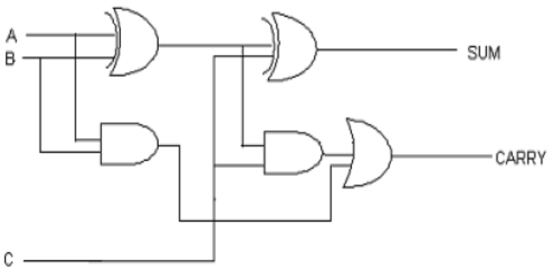
$$S = XY' + X'Y \quad C = XY$$

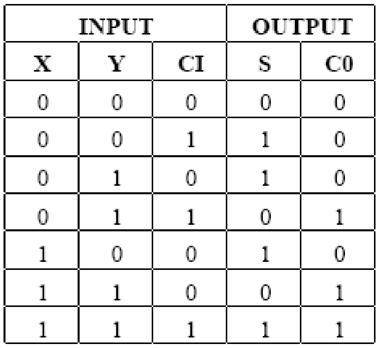
### 3.1.3 FULL ADDER

- The adder circuit is capable of adding the content of two registers.
- It must include provision for handling carries as well as an addend and augends bits.
- So there must be three inputs to each stage of a multi digit adder,
- except the stage for the least significant bits. One for each input from the numbers being added, one for any carry that might have been generated or propagated by the previous stage.
- There are three inputs to the full-adder X and Y inputs from the respective digits of the
- registers to be added, the Ci input, which is for any carry generated by the previous stage.
- The two outputs are S, which is the output value for that stage of the addition,
- and C0, which produces the carry to be added into the next stage.
- The Boolean expressions for the input output relationships for each of the 2 outputs are as follows:



Block diagram





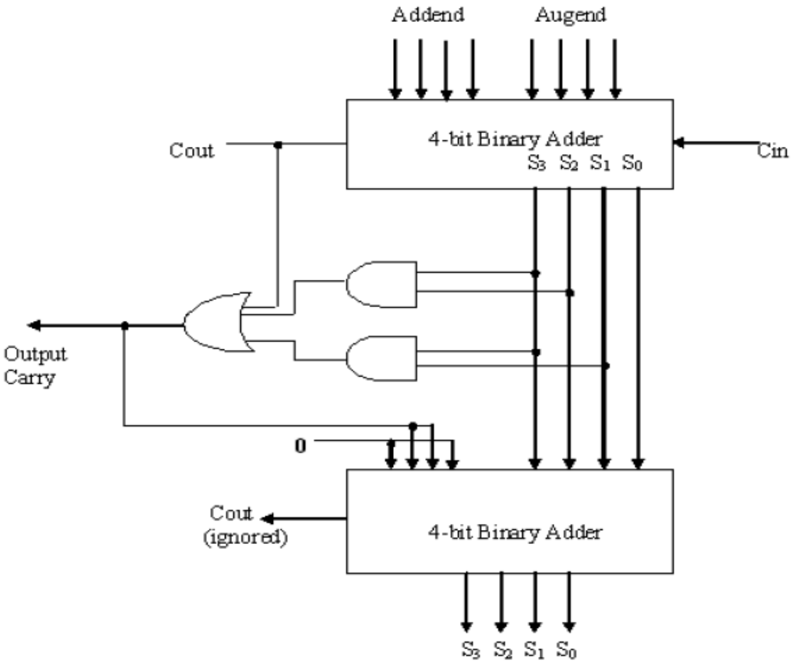
| INPUT | | | OUTPUT | |
|---|---|---|---|---|
| X | Y | CI | S | C0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Logic Diagram

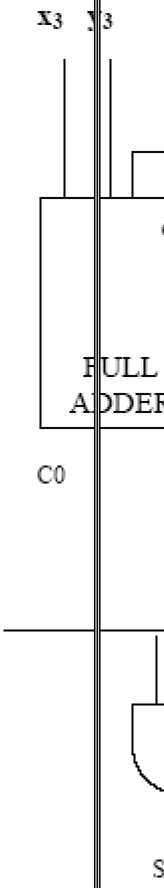
Truth Table

### 3.1.4 BINARY CODED DECIMAL ADDER

- Arithmetic units which perform operations on numbers stored in BCD form must have
- the ability to add 4-bit representations of decimal digits.
- To do this BCD adder is used. A block diagram symbol for an adder is shown.



**BCD Adder**

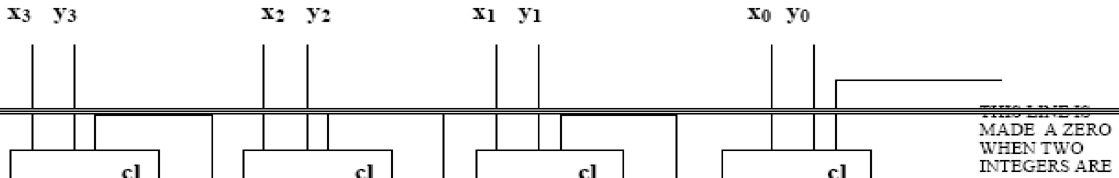


- The adder has an augends digit input consisting of four lines, an addend digit input for four lines, a carry- in and a carryout, and a sum digit with four output lines.
- The augend digit, addend digit and sum digit with four-output line.
- The augend digit, addend digit and sum digit are each represented 8, 4, 2, 1 BCD form.
- The purpose of the BCD adder in figure is to add the augend and addend digits and the carry- in and produce a sum digit and carry out.

```
(i)8+7=15        1000+0111
(ii) 9+5=14            1001
                       0101
                       1110
                      +0110
 Carry Generated      10100          = 14
```

### 3.1.5 PARALLEL BINARY ADDER

- The purpose of this adder is to add two 4-bit binary integers.
- The adder inputs are named X0 through X3 and the augend bits are represented by Y0 through Y3.
- A 4-bit parallel binary adder is illustrated in the figure.



Consider the additionFigure. following Binary binary numbers:

0111 where X3=0, X2=1, X1=1 and X0=1

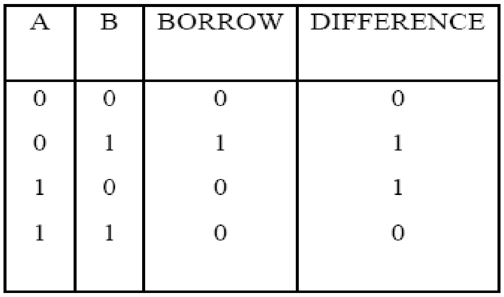
0011 where Y3=0, Y2=0, Y1=1 and Y0=1 Sum = 1010

- The sum should therefore be S3=1, S2=0, S1=1 and S0=0.
- The operation of the adder may be checked as follows.
- Since X0 and Y0 are the least significant digits, they cannot receive a carry from a previous stage.
- In the problem above X0 and Y0 are both 1s, their sum therefore 0 and a carry is generated and added into the full-adder for bits X1 and Y1.
- Bits X1 and Y1 are also both 1s, as is the carry input to this stage.
- Therefore the sum output line S1 carries a 1 and the carry line to the next stage also carries a 1.
- Since X2 is a 1, Y2 is a 0 and the carry input is 1.
- The sum output line S2 will carry a 0, and the carry to the next stage will be a 1.
- Both inputs X3 and Y3 are equal to 0, and the carry
- input line to this adder stage is equal to 1.
- Therefore, the sum output line S3 will represent a 1 and the carry output line designated as "overflow" will have a 0 output.

## 3.1.6 SUBTRACTOR

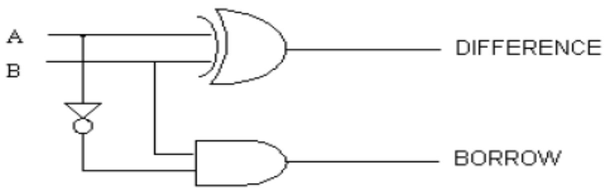
### HALF SUBTRACTOR

- A half subtractor subtracts a bit from another.
- The subtraction table (or truth table) of a half subtractor is shown below.
- The half subtractor has two input bits A and B two output bits, a difference DIFF = (A-B) and a Borrow.



| A | B | BORROW | DIFFERENCE |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 |

Truth table



**Half Subtractor**

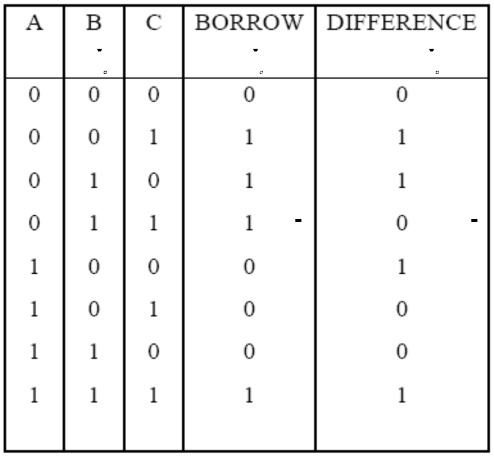
- From the truth table shown in the table , it can be seen that DIFF = A-B and borrow which has been implemented on the logic circuit for half subtractor.

**FULL SUBTRACTOR**

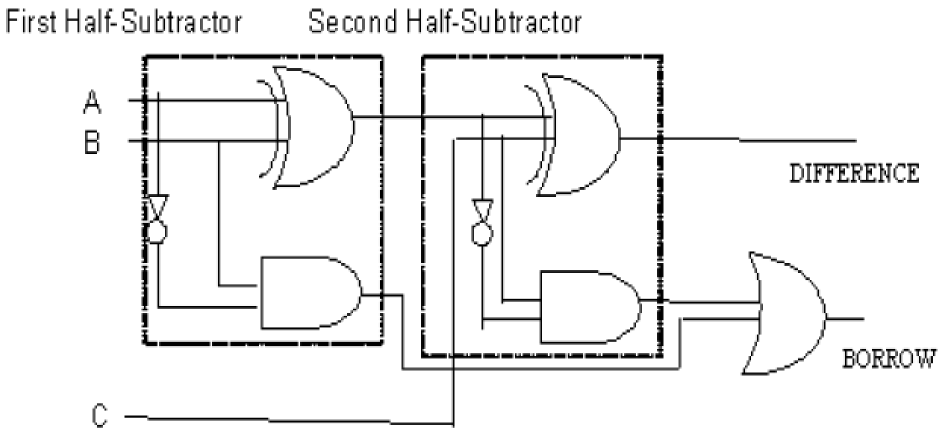
- A full subtractor subtracts with three bits (A-B-C).
- The third bit C is the borrow from previous stage.
- The truth table of a full subtractor is given in table .

From the truth table of the full subtractor it can be seen that DIFF = (A B C) and borrow = A’B + BC + CA’. This logic has been implemented.

Truth Table



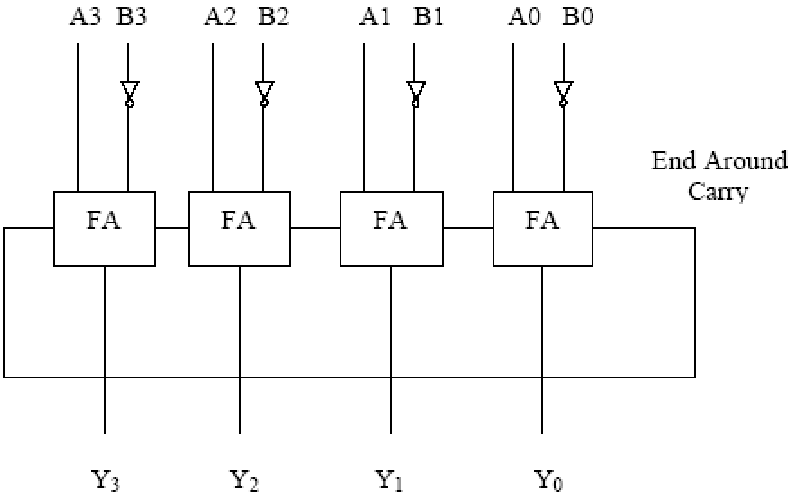
| A | B | C | BORROW | DIFFERENCE |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |



**FULL SUBTRACTOR**

## PARALLEL BINARY SUBTRACTOR

- When a binary number is to be subtracted from another using 1’s complement method, the following circuit can be used.
- The number to be subtracted is first complemented using inverters.
- The complemented number is then added to the minuend using full adders.
- The carry resulting from the addition is added to the least significant bit as shown in Figure.



**Parallel Binary Subtractor**

## PARALLEL BINARY ADDER - SUBTRACTOR

- For the 1’s complement system, if the storage register is composed of flip flops,the 1’s complement can be formed by simply connecting the complement of each input to the adder.

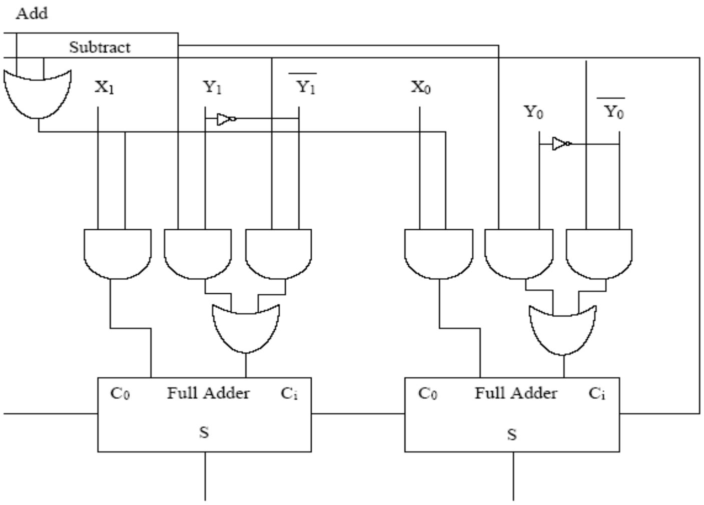
## PARALLEL ARITHMETIC EL EMENT

- A complete logical circuit capable of adding or subtracting two signed 2's complement number is shown below.

  To add: Add line is made 1.

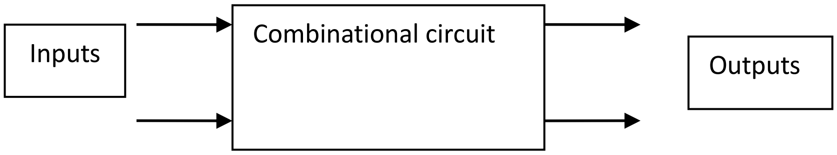
  To Subtract: Subtract line is made 1.

- One number is represented by X1 X0 and the other number is represented by Y1Y0.
- There are two control signals. ADD and SUBTRACT.
- If the add control line is made a 1 then the sum of the number X and Y will appear as S1 S0.
- If the subtract line is made a 1, then the difference between X and Y will appear S1 S0.
- The AND to OR gate network connected to each input selects either Y or Y.

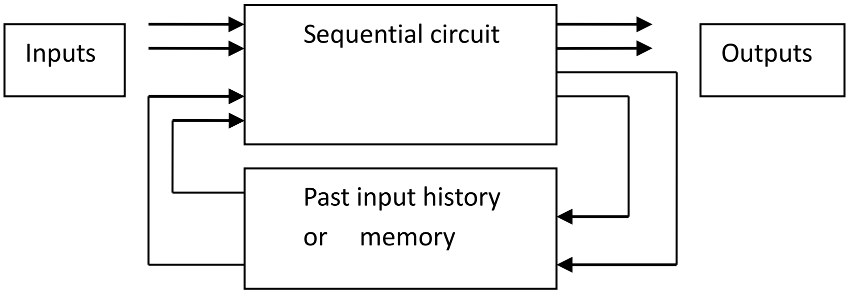


## 3.2 SEQUENTIAL CIRCUITS

- Two types of circuits - Combinational circuit and Sequential circuit.
- Combinational circuit doesn’t has any memory elements and its output depends only on the combined set of input applied to it simultaneously at that instant of time.



- Sequential circuit must contain atleast one memory element and its output depends on both the present input and previous history of inputs.



- Two types of sequential circuits – i) clocked or synchronous
  ii) unclocked or asynchronous

**Clocked or synchronous sequential circuits:**

- A master oscillator provides the regular timing pulses.
- Events are allowed to occur during the timing pulses.
- Has a propagation delay.
- Example : Magnetic tape reader – reading takes place only when the portion of tape to be read is under the read head.

**Unclocked or asynchronous sequential circuits:**

- Don’t wait for the timing pulses.
- Event takes place or occurs after the completion of a previous event.
- Has internal logic circuit delay.

- Example : Telephone dial system – user dials the next number as soon as the dialing of the
  previous number is completed.

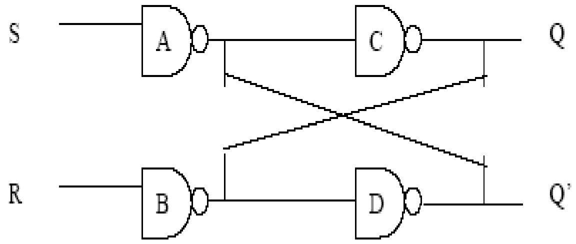
### 3.2.1 INTRODUCTION

### FLIP-FLOPS

- Synchronous sequential circuits employ signals that affect the storage elements onl at a discrete instants of time.
- Synchronization is achieved by a timing device called a clock pulse generator that produces a periodic train of clock pulse.
- The storage element employed in clocked sequential circuits are called flip-flops.
- A flip-flop is a basic memory element used to store one bit of information.
- This is abbreviated as FF or F/F.
- It has 2 output signals, one for the normal value and one for the complement value of the bit stored.

### SR FLIP FLOP

- The RS (SR) flip flop has three inputs, labeled S (for Set), R(for Reset) and C(for Clock).
- It has an output Q.
- The complemented output is indicated with a small circle at the other output terminal.



**Graphic Symbol**

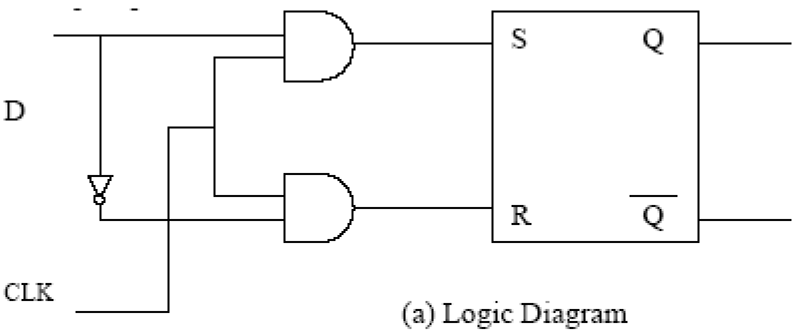
| S | R | Q(t+1) |
|---|---|---|
| 0 | 0 | Q(t) No change |
| 0 | 1 | 0 Clear to 0 |
| 1 | 0 | 1 Set to 1 |
| 1 | 1 | ? Indeterminate |

**Characteristic Table**

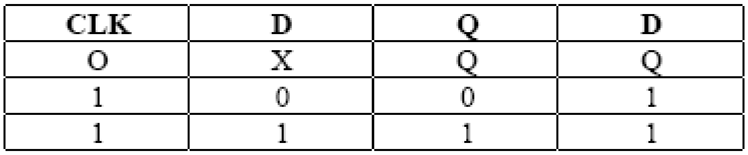
- The arrowhead-shaped symbol in front of the letter C to designate a dynamic input.
- The dynamic indicator symbol denotes that the flip-flop responds to a positive transition (from 0 to 1) of the input clock signal.
- If there is no clock input C, the output of the circuit cannot change.
- Only when the clock signal changes from 0 to 1 the output can be affected according to the values of S and R.
- S and R columns gives the binary value of the two inputs.
- Q(t) is the binary state of the Q output at a given time (present time).
- Q(t+1) is the binary state of the Q output after the occurrence of a clock transition (next state).

### 3.2.2 D FLIP FLOP

- The D(data) flip-flop is a modification of RS (SR) flip-flop.
- An SR flip-flop is converted to a D flip-flop by inserting an inverter between S and R and assigning the symbol D to the single input.



(a) Logic Diagram



| CLK | D | Q | D |
|---|---|---|---|
| O | X | Q | Q |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 |

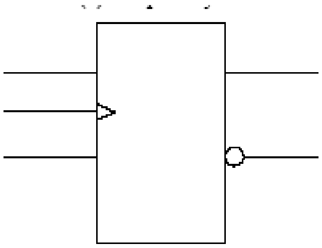
**Characteristic Table**

- If D = 0 and when C changes from 0 to 1, output Q is cleared to 0.
- If D = 1 and when C changes from 0 to 1, output Q is set to 1.
- The output state Q(t+1) is determined by the input D.
- The relationship can be expressed using the characteristic equation
    - Q(t+1) = D
- No input condition will leave the state of D flip-flop unchanged.
- The advantage of D flip-flop is it has only one input(excluding C).
- The disadvantage of D flip-flop is it do not have "no change" condition.
    - Q(t+1)=Q(t)
- This can be done by feeding the output back to the input or by disabling the clock.

### 3.2.3 JK FLIP FLOP

- This is a refinement of SR flip-flop.
- The indeterminate state of SR flip-flop is defined in JK flip-flop.

- The inputs J and K are used to set and clear the flip-flop. (just like S and R).
- If J = 1 and K = 1 and when C changes from 0 to 1, then a complemented output is obtained. Q(t+1)=Q’(t)



| J | K | Q(t+1) |
|---|---|---|
| 0 | 0 | Q(t) No change |
| 0 | 1 | 0 Clear to 0 |
| 1 | 0 | 1 Set to 1 |
| 1 | 1 | Q’(t) Complement |

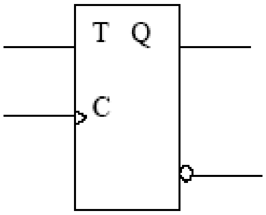
**Graphic Symbol** **Characteristic Table**

### 3.2.4 T FLIP FLOP

- The T (toggle) flip-flop is obtained from a JK flip-flop when inputs J and K are connected to provide a single input designated by T.

- When T =0 (J = K = 0) and when C changes from 0 to 1, there is no change in the output.
- When T =1 (J = K = 1) and when C changes from 0 to 1, the complemented state of the output is obtained.
- The conditions can be expressed by a characteristic equation

$$Q(t+1) = Q(t) \oplus T$$



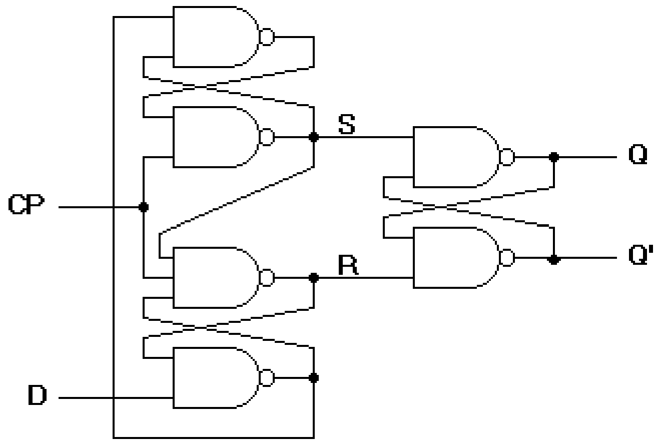
| T | Q(t+1) |
|---|---|
| 0 | Q(t) No change |
| 1 | Q’(t) Complement |

**Graphic Symbol** **Characteristic Table**

### 3.2.5 EDGE TRIGGERED FLIP-FLOP

- Another type of flip-flop that synchronizes the state changes during a clock pulse transition is the edge-triggered flip-flop.
- When the clock pulse input exceeds a specific threshold level, the inputs are locked out and the flip-flop is not affected by further changes in the inputs until the clock pulse returns to 0 and another pulse occurs.

- Some edge-triggered flip-flops cause a transition on the positive edge of the clock pulse (positive-edge-triggered), and others on the negative edge of the pulse (negative-edge-triggered).
- The logic diagram of a D-type positive-edge-triggered flip-flop is shown.

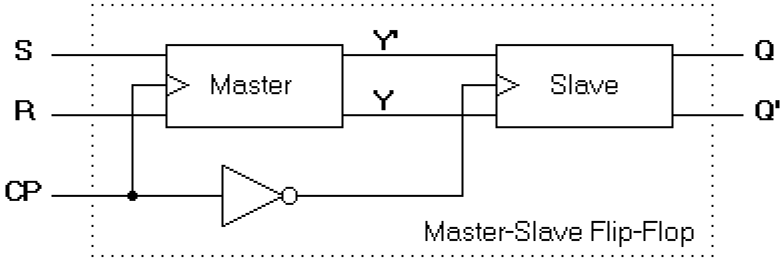


D-type positive-edge triggered flip-flop

When using different types of flip-flops in the same circuit, one must ensure that all flip-flop outputs make their transitions at the same time, ie., during either the negative edge or the positive edge of the clock pulse.

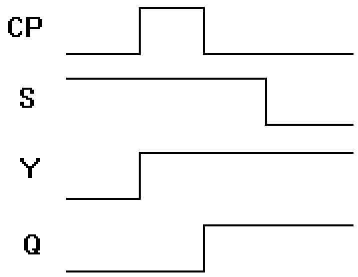
### 3.2.6 MASTER-SLAVE FLIP-FLOP

- A master-slave flip-flop is constructed from two separate flip-flops.
- One circuit serves as a master and the other as a slave.
- The logic diagram of an SR flip-flop is shown.
- The master flip-flop is enabled on the positive edge of the clock pulse CP and the slave flip-flop is disabled by the inverter.
- The information at the external R and S inputs is transmitted to the master flip-flop.
- When the pulse returns to 0, the master flip-flop is disabled and the slave flip-flop is enabled.
- The slave flip-flop then goes to the same state as the master flip-flop.



Logic diagram of a master-slave flip-flop

- The timing relationship is shown and is assumed that the flip-flop is in the clear state prior to the occurrence of the clock pulse.
- The output state of the master-slave flip-flop occurs on the negative transition of the clock pulse.
- Some master-slave flip-flops change output state on the positive transition of the clock pulse by having an additional inverter between the CP terminal and the input of the master.



Timing relationship in a master slave flip-flop