

SUNBEAM

Institute of Information Technology

Digital Electronics

Contents

❖ Introduction

Signal, Analog Signal, Digital Signal

❖ Number System

Decimal number , Binary number, Octal number, Hexadecimal number

Converting from Another Base to Decimal

Converting from Decimal to Another Base

Converting from a base Other than 10 to Another Base Other than 10

Binary to octal and Octal to Binary

Binary to Hex and Hex to Binary

Binary Coded Decimal

Gray code, Excess-3 code

Laws of Boolean Algebra, Boolean Algebra

K-Map, Logic Gates

Binary Addition, Binary Subtraction, Multiplication, Division

1's complement, 2's complement

❖ Combinational Circuit

Half Adder, Full Adder

Half Subtractor, Full Subtractor

Multiplex, Demultiplexer

Decoder, Encoder

❖ Sequential Circuit

RS Flip-flop D Flip-flop, JK Flip-flop, - T Flip-flop

❖ Counter

Johnson Counter

Ring Counter

❖ Shift Register

SISO, SIPO, PISO, PIPO

❖ Logic Family

RTL, DTL, TTL, CMOS, ECL

Introduction

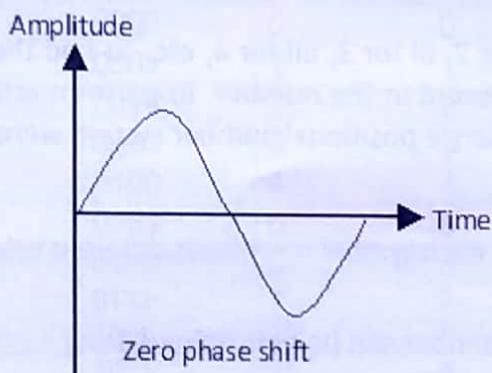
Signal

Signal can be defined as a physical quantity, which contains some information. Signals are of two types.

- Analog Signal
- Digital Signal

Analog Signal

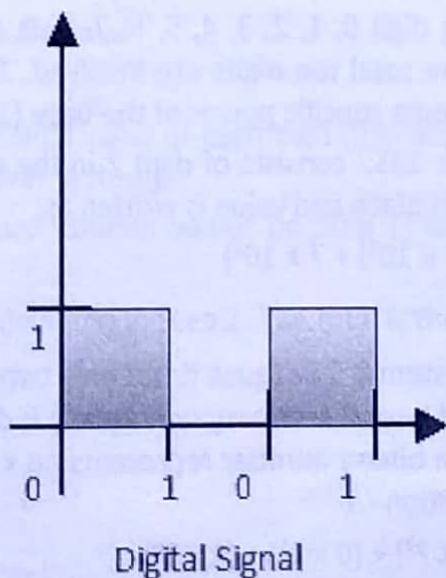
An analog signal is a continuous wave that changes over time period represented by sine wave and vary in signal strength(amplitude) or frequency(time). Example of analog signal is voice, temperature, pressure etc.



Digital Signal

A digital signal is a discrete wave that carries information in binary form represented by square wave and vary in bit rate and bit intervals.

Example of digital signal is binary signal, octal signal, hexadecimal signal.



Advantages of digital signal over Analog Signal

1. Accurate measurement
2. Less distortion
3. Less Noise effects
4. Picture quality is good (pixel)
5. Communication is easy
6. Information storage is easy

Number System

In digital electronics, the number system is used for representing the information in digits. Number System are of two types

- Non-positional number system
- Positional number system

Non-positional Number System

In Non-positional number system, each symbol represents the same value regardless of its position.

Symbols such as I for 1, II for 2, III for 3, IIII for 4, etc. To find the value of number, one has to count the number of symbols present in the number. To perform arithmetic operation with such a number system is very difficult, hence positional number system were developed.

Positional Number System

In positional number system, each symbol represents different values, depending on the position they occupy in the number.

The value of each digit in a number can be determined using

- The digit
- The position of the digit in the number
- The base of the number system (where base is defined as the total number of digits available in the number system).

Decimal Number System

The number system is having digit 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, this number system is known as a decimal number system because total ten digits are involved. The base of the decimal number system is 10. Each position represents specific power of the base (10).

For example, decimal number 2357 consists of digit 7 in the unit's place, 5 in tens place, 3 in hundreds place and 2 in thousands place and value is written as:

$$2357 = (2 \times 10^3) + (3 \times 10^2) + (5 \times 10^1) + 7 \times 10^0$$

Binary Number System

The base of binary number system is 2 because it has only two digits 0 and 1. Each position in a binary number represent power of base 2. In binary unit's place is denoted as (2^0) , tens place as (2^1) , hundreds as (2^2) , Last position in a binary number represents an x power of the base (2). Example: 2^x where x represents the last position - 1

$$10101 = (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$$

Octal Number System

The octal system has the base of eight as it uses eight digits 0, 1, 2, 3, 4, 5, 6, 7.

Each position in an octal number represents a 0 power of the base 8. Example: 8^0

Last position in an octal number represents an x power of the base 8. Example 8^x where x represents the last position-1.

$$12570 = (1 \times 8^4) + (2 \times 8^3) + (5 \times 8^2) + (7 \times 8^1) + (0 \times 8^0)$$

Hexadecimal Number System

The hexadecimal number system has a base of 16, and hence it consists of the following sixteen number of digits. 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Each position in a hexadecimal number represents a 0 power of the base (16). Example 16^0 . Last position in a hexadecimal number represents an x power of the base (16). Example 16^x where x represents the last position - 1.

$$19FDE = (1 \times 16^4) + (9 \times 16^3) + (F \times 16^2) + (D \times 16^1) + (E \times 16^0)$$

The table is shown below are decimal, binary, octal, and hexadecimal numbers from 0 to 15 and their equivalent binary number.

Decimal	Binary	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Converting from Another Base to Decimal

Steps :

Step 1 – Determine the positional value of each digit (this depends on the position of the digit and the base of the number system).

Step 2 – Multiply the obtained column values (in Step 1) by the digits in the corresponding positional value.

Step 3 – Sum the products calculated in Step 2. The total is the equivalent value in decimal.

Binary to Decimal Number

For Example

$$\begin{aligned}(101101)_2 &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\&= 1 \times 32 + 0 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 \\&= 32 + 8 + 4 + 1 \\&= (45)_{10}\end{aligned}$$

Octal to Decimal Number

For Example

$$\begin{aligned}(4072)_8 &= 4 \times 8^3 + 0 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 \\&= 4 \times 512 + 0 \times 64 + 7 \times 8 + 2 \times 1 \\&= 2048 + 56 + 2 \\&= (2106)_{10}\end{aligned}$$

Hexadecimal to Decimal Number

For example

$$\begin{aligned}(27A)_{16} &= 2 \times 16^2 + 7 \times 16^1 + A \times 16^0 \\&= 2 \times 256 + 7 \times 16 + 10 \times 1 \\&= 512 + 112 + 10 \\&= (634)_{10}\end{aligned}$$

Converting from Decimal to Another Base

Steps

Step 1 – Divide the decimal number to be converted by the value of the new base.

Step 2 – Get the remainder from Step 1 as the rightmost digit (least significant digit) of new base number.

Step 3 – Divide the quotient of the previous divide by the new base.

Step 4 – Record the remainder from Step 3 as the next digit (to the left) of the new base number.

Repeat Steps 3 and 4, getting remainders from right to left, until the quotient becomes zero in Step 3.

Decimal to Binary Number

For example

$(37)_{10}$

$$37 / 2 = 18 \text{ remainder } 1 \text{ (least significant digit)}$$

$$18 / 2 = 9 \text{ remainder } 0$$

$$9 / 2 = 4 \text{ remainder } 1$$

$$4 / 2 = 2 \text{ remainder } 0$$

$$2 / 2 = 1 \text{ remainder } 0$$

$$1 / 2 = 0 \text{ remainder } 1 \text{ (most significant digit)}$$

Resulting binary number is $(100101)_2$

Decimal to Octal Number

For Example

$(574)_{10}$

$$574 / 8 = 71 \text{ remainder } 6$$

$$71 / 8 = 8 \text{ remainder } 7$$

$$8 / 8 = 1 \text{ remainder } 0$$

$$1 / 8 = 0 \text{ remainder } 1$$

Resulting Octal no is $(1076)_8$

Decimal to Hexadecimal Number

For Example

$(256)_{10}$

$$256 / 16 = 16 \text{ remainder } 0$$

$$16 / 16 = 1 \text{ remainder } 0$$

$$1 / 16 = 0 \text{ remainder } 1$$

Resulting Octal no is $(100)_{16}$

Converting from a base Other than 10 to Another Base Other than 10

Steps

Step 1: First Convert given base to Base 10 (decimal)number

Step 2: Then convert decimal number to the whichever base is asked

For Example

$(425)_6 = ?_4$

$$= 4 \times 6^2 + 2 \times 6^1 + 5 \times 6^0$$

$$= 4 \times 36 + 2 \times 6 + 5$$

$$= 144 + 12 + 5$$

$$= (161)_{10}$$

$(161)_{10}$

$161/4 = 40$	remainder 1
$40/4 = 10$	remainder 0
$10/4 = 2$	remainder 2
$2/4 = 0$	remainder 2
Result is $(2201)_4$	

Binary to Octal

Steps

- Step 1** – Combine the binary digits into groups of three (starting from the right).
Step 2 – Convert each group of three binary digits to one octal digit.

For Example

Convert $(101111010)_2$ to Octal

A +

101	111	010
↓	↓	↓
5	7	2

Result is $(572)_8$

Octal to Binary

Steps

- Step 1** – Convert each octal digit to a 3-digit binary number.

For Example

Convert $(356)_8$ to binary

3	5	6
↓	↓	↓
011	101	110

Result is $(011101110)_2$

Binary to Hexadecimal

Steps

- Step 1** – Divide the binary digits into groups of four (starting from the right).
Step 2 – Convert each group of four binary digits to one hexadecimal symbol.

For Example

Convert $(110110010111)_2$ to Hexadecimal

1101	1001	0111
↓	↓	↓
D	9	7

Result is $(D97)_{16}$

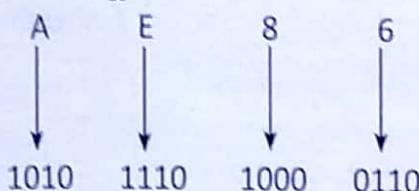
Hexadecimal to Binary

Steps

Step 1 – Convert each hexadecimal digit to a 4-digit binary number.

For Example

Convert $(AE86)_{16}$ to binary



Result is $(1010111010000110)_2$

WEIGHTED AND NON-WEIGHTED CODES

There are two types of binary codes

- 1) Weighted binary codes
- 2) Non-weighted binary codes

In weighted codes, for each position, there is specific weight attached. For example, in binary number, each bit is assigned particular weight 2^n where 'n' is the bit number for $n = 0, 1, 2, 3, 4$ the weights are 1, 2, 4, 8, 16 respectively.

Example: BCD

Non-weighted codes are codes which are not assigned with any weight to each digit position, i.e., each digit position within the number is not assigned fixed value.

Example: Excess – 3 (XS -3) code and Gray codes

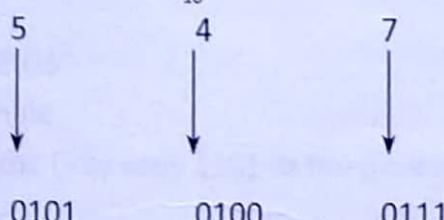
BCD

The binary coded decimal (BCD), each decimal digit is represented by a 4-bit binary number. BCD is a way to express each of the decimal digits with a binary code.

BCD is a weighted code its weight are 8421.BCD code are used only till 9(0000 to 1001).

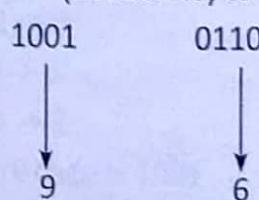
For Example

Convert decimal $(547)_{10}$ to BCD



Result is $(010101000111)_2$

Convert BCD $(10010110)_2$ to decimal



Result is $(96)_{10}$

EXCESS-3 (XS-3) CODE

The Excess-3 code, also called XS-3, is a non-weighted BCD code. This derives its name from the fact that each binary code word is the corresponding 8421 code word plus 0011(3). It is a sequential code. It is a self-complementing code.

For Example

Convert $(25)_{10}$ into Excess-3 code

Step 1 Convert decimal to BCD

2	5
↓	↓
0010	0101

Step 2 Add +3 to each BCD number

$$\begin{array}{r}
 0010 \quad 0101 \\
 +0011 \quad 0011 \\
 \hline
 0101 \quad 1000
 \end{array}$$

Result is $(01011000)_2 = 58$ in Excess-3 code

GRAY CODE

The gray code is a non-weighted code. It is cyclic code because successive words in this differ in one bit position only i.e. it is a unit distance code (at a time one bit changes).

Gray code is used in instrumentation and data acquisition systems where linear or angular displacement is measured. They are also used in shaft encoders, I/O devices, A/D converters and other peripheral equipment.

Gray Code Decimal Equivalent

0000	0
0001	1
0011	2
0010	3
0110	4
0111	5
0101	6
0100	7
1100	8
1101	9
1111	10
1110	11
1010	12
1011	13
1001	14
1000	15

BINARY TO GRAY CONVERSION

If an n-bit binary number is represented by $B_n B_{n-1} \dots B_1$ and its gray code equivalent by $G_n G_{n-1} \dots G_1$,

where B_n and G_n are the MSB, then gray code bits are obtained from the binary code as follows

$$G_n = B_n$$

$$G_{n-1} = B_n \oplus B_{n-1}$$

$$G_1 = B_2 \oplus B_1$$

Where the symbol stands for Exclusive OR (X-OR)

For example

Convert the binary 1001 to the Gray code.

Solution

$$\text{Binary } \rightarrow 10000001$$



$$\text{Gray } \rightarrow 1101$$

The gray code is 1101

GRAY-TO-BINARY CONVERSION

If an n-bit gray number is represented by $G_n G_{n-1} \dots G_1$ and its binary equivalent by $B_n B_{n-1} \dots B_1$, then binary bits are obtained from Gray bits as follows:

$$B_n = G_n$$

$$B_{n-1} = B_n \oplus G_{n-1}$$

$$B_1 = B_2 \oplus G_1$$

For example

Convert the Gray code 1101 to the binary.

Solution

$$\text{Gray } \rightarrow 1101$$

$$B_0 \rightarrow 1$$

$$B_1 \rightarrow 1 \oplus 1 \rightarrow 0$$

$$B_2 \rightarrow 0 \oplus 0 \rightarrow 0$$

$$B_3 \rightarrow 0 \oplus 1 \rightarrow 1$$

The binary code is 1001

Laws of Boolean Algebra

1) Commutative Law

$$(a) A+B=B+A$$

$$(b) AB=BA$$

2) Associative Law

$$(a) (A+B)+C=A+(B+C)$$

$$(b) (A B) C = A (B C)$$

3) Distributive Law

$$(a) A (B+C) = AB+AC$$

$$(b) A + (B C) = (A + B) (A + C)$$

4) Identity Law

$$(a) A+A=A$$

$$(b) A A = A$$

5) Involution Law

$$\bar{\bar{A}} = A$$

6) Absorption Law

$$(a) A + A B = A$$

$$(b) A (A + B) = A$$

7) De Morgan's Theorem

$$(a) \overline{A+B} = \bar{A} \cdot \bar{B}$$

$$(b) \overline{AB} = \bar{A} + \bar{B}$$

Some Basic Laws

$$0 + A = A 1 + A = 1$$

$$0.A = 0 1.A = A$$

$$A + \bar{A} = 1 \cdot A \quad \bar{A} = 0$$

Boolean Functions can be done by using two methods

- Algebraic method
- Using K-maps

Algebraic method

Using the theorems of Boolean algebra, prove the following

$$\begin{aligned}(1) A &+ (A \cdot B) \\&= A \cdot (1 + B) \\&= A \cdot 1 \\&= A\end{aligned}$$

$$\begin{aligned}(2) A \cdot &(A + B) \\&= A \cdot A + A \cdot B \\&= A + A \cdot B \\&= A \cdot (1 + B) \\&= A \cdot 1 \\&= A\end{aligned}$$

(3) $(A + B). (A + C)$

$$\begin{aligned} &= A \cdot A + A \cdot C + A \cdot B + B \cdot C && - \text{Distributive law} \\ &= A + A \cdot C + A \cdot B + B \cdot C && - \text{Idempotent AND law } (A \cdot A = A) \\ &= A(1 + C) + A \cdot B + B \cdot C && - \text{Distributive law} \\ &= A \cdot 1 + A \cdot B + B \cdot C && - \text{Identity OR law } (1 + C = 1) \\ &= A(1 + B) + B \cdot C && - \text{Distributive law} \\ &= A \cdot 1 + B \cdot C && - \text{Identity OR law } (1 + B = 1) \\ &= A + (B \cdot C) && - \text{Identity AND law } (A \cdot 1 = A) \end{aligned}$$

Canonical & Standard Forms

We will get four Boolean product terms by combining two variables x and y with logical AND operation. These Boolean product terms are called as min terms or standard product terms. The min terms are $x'y'$, $x'y$, xy' and xy .

Similarly, we will get four Boolean sum terms by combining two variables x and y with logical OR operation. These Boolean sum terms are called as Max terms or standard sum terms. The Max terms are $x+y$, $x+y'$, $x'+y$ and $x'+y'$.

SOP

Canonical SOP form means Canonical Sum of Products form. In this form, each product term contains all literals. So, these product terms are nothing but the min terms. Hence, canonical SOP form is also called as sum of min terms form.

Boolean function of output is, $f = p'qr + pq'r + pqr' + pqr$. This is the canonical SOP form of output, f. We can also represent this function in following two notations.

$$f = m_3 + m_5 + m_6 + m_7$$

$$f = \sum m(3, 5, 6, 7)$$

POS

Canonical POS form means Canonical Product of Sums form. In this form, each sum term contains all literals. So, these sum terms are nothing but the Max terms. Hence, canonical POS form is also called as product of Max terms form.

Boolean function of output is, $f = (p+q+r).(p+q+r').(p+q'+r).(p'+q+r)$. This is the canonical POS form of output, f. We can also represent this function in following two notations.

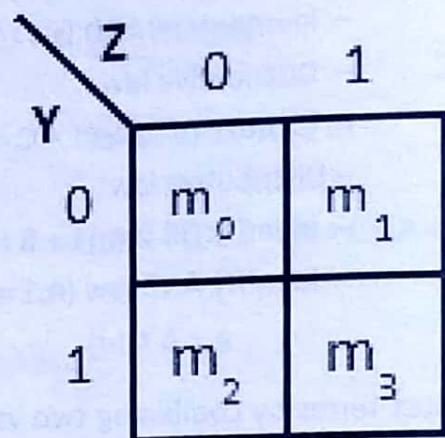
$$f = M_0.M_1.M_2.M_4$$

$$f = \prod M(0, 1, 2, 4)$$

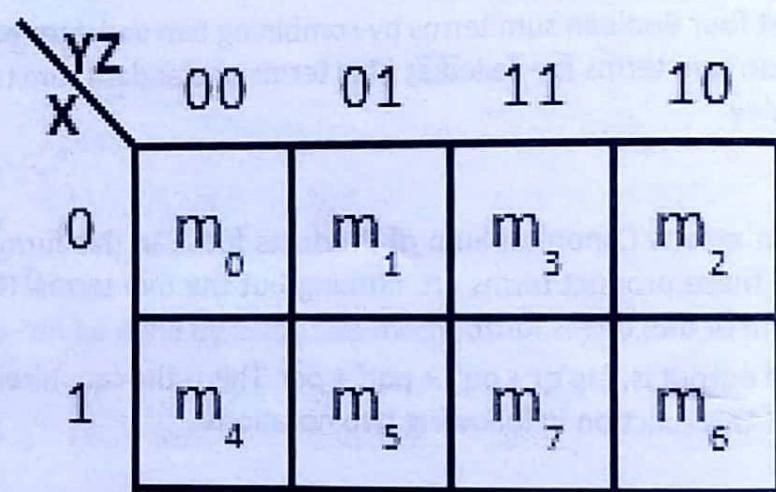
K-Map

Karnaugh introduced a simplification of Boolean functions in an easy way. This method is known as Karnaugh map method or K-map method. It is a pictorial representation of graphical method, which consists of 2^n cells for 'n' variables. The adjacent cells are differed only in single bit position.

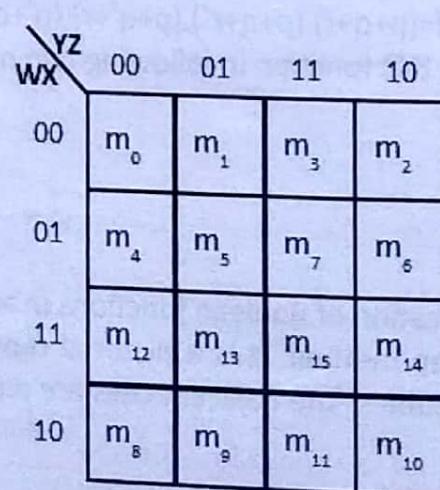
2 Variable K-Map



3 Variable K-Map

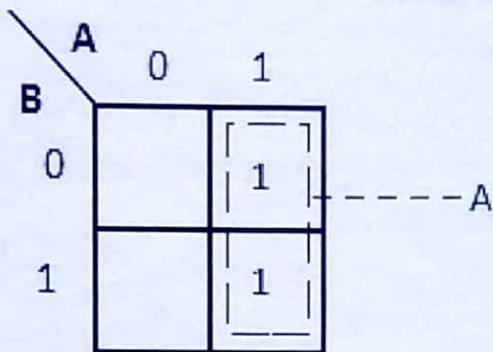


4 Variable K-Map



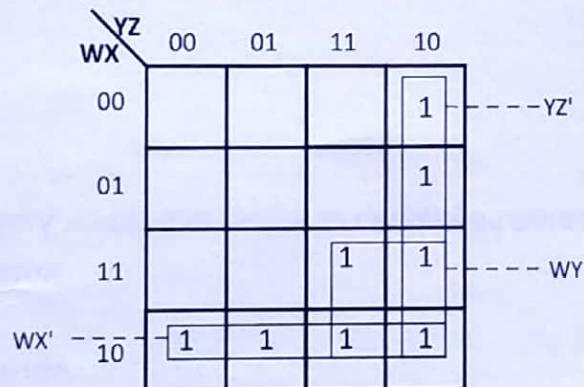
Example

1) $Z = f(A, B) = AB' + AB$



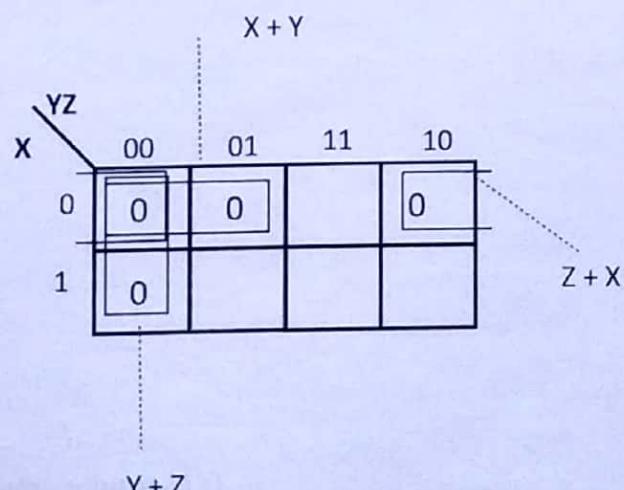
Answer = A (Which value remains constant that value should be taken)

- 2) simplify the following Boolean function, $f(W, X, Y, Z) = WX'Y' + WY + W'YZ'$ using K-map in standard SOP form.



$$f = WX' + WY + YZ'$$

- 3) simplify the following Boolean function, $f(X, Y, Z) = \prod M(0, 1, 2, 4)$ using K-map in standard POS form.



$$f = (X+Y)(Y+Z)(Z+X)$$

BINARY ARITHMETIC OPERATION

1. BINARY ADDITION

The binary addition rules are as follows

$$0 + 0 = 0;$$

$$0 + 1 = 1;$$

$$1 + 0 = 1;$$

$$1 + 1 = 0, \text{ with a carry of } 1$$

For example

Add $(100101)_2$ and $(101111)_2$.

Solution

$$\begin{array}{r} 100101 \\ + 101111 \\ \hline \end{array}$$

$$1010100$$

Result is $(1010100)_2$

2. BINARY SUBTRACTION

The binary subtraction rules are as follows

$$0 - 0 = 0;$$

$$1 - 1 = 0;$$

$$1 - 0 = 1;$$

$$0 - 1 = 1, \text{ with a borrow of } 1$$

For example

Subtract $(111111)_2$ from $(101001)_2$.

Solution

$$\begin{array}{r} 111111 \\ - 101001 \\ \hline \end{array}$$

$$010110$$

Result is $(010110)_2$

3. BINARY MULTIPLICATION

The binary multiplication rules are as follows

$$0 \times 0 = 0;$$

$$1 \times 1 = 1;$$

$$1 \times 0 = 0;$$

$$0 \times 1 = 0$$

For example

Multiply $(1101)_2$ by $(111)_2$.

Solution

$$\begin{array}{r} 1101 \\ \times 111 \\ \hline 1101 \\ 1101x \\ 1101xx \\ \hline 1011011 \end{array}$$

Result is $(1011011)_2$

BINARY DIVISION

The binary division is very simple and similar to decimal number system.

$$0 \div 0 = \text{divide by zero error}$$

$$0 \div 1 = 0$$

$$1 \div 0 = \text{divide by zero error}$$

$$1 \div 1 = 1$$

For Example

Divide $(10110)_2$ by $(110)_2$

Solution

$$\begin{array}{r)10110(0111 \\ 110 \\ \hline 1011 \\ 110 \\ \hline 1001 \\ 110 \\ \hline 0011 \end{array}$$

Answer is $(0111)_2$ and remainder is $(11)_2$

1's COMPLEMENT

The 1's complement of a binary number is obtained by replacing 0 with 1 and 1 with 0.

For example

Find $(1100)_2$ 1's complement.

Solution

Given value 1 1 0 0

1's complement is 0 0 1 1

Result is $(0011)_2$

2's COMPLEMENT

The 2's complement of a binary number is a binary number which is obtained by adding 1 to the 1's

complement of a number

i.e. 2's complement = 1's complement + 1

For example

Find $(1010)_2$ 2's complement.

Solution

Given value 1 0 1 0

1's complement is 0 1 0 1

 + 1

2's complement 0 1 1 0

Result is $(0110)_2$

SIGNED NUMBER

In sign – magnitude form, additional bit called the sign bit is placed in front of the number. If the sign bit is 0, the number is positive. If it is a 1, the number is negative.

For example

0 1 0 1 0 0 1 = +41

↑

Sign bit

1 1 0 1 0 0 1 = -41

↑

Sign bit

LOGIC GATES

- Logic gates are the fundamental building blocks of digital systems.
- There are 3 basic types of gates AND, OR and NOT. Logic gates are electronic circuits because they are made up of a number of electronic devices and components.
- Inputs and outputs of logic gates can occur only in 2 levels. These two levels are termed HIGH and LOW, or TRUE and FALSE, or ON and OFF or simply 1 and 0
- The table which lists all the possible combinations of input variables and the corresponding Output is called a truth table.

DIFFERENT TYPES OF LOGIC GATES

NOT GATE (INVERTER)

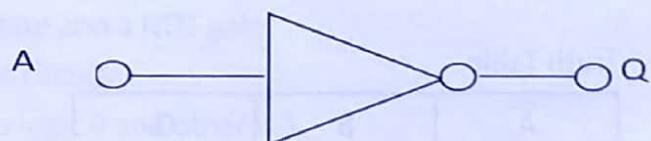
A NOT gate, also called and inverter, has only one input and one output. It is a device whose output is always the complement of its input.

The output of a NOT gate is at logic 1 state when its input is 0 and the logic 0 state when its input is 1.

Truth table

A	Q
0	1
1	0

symbol of NOT



$$Q = \bar{A}$$

AND GATE

An AND gate has two or more inputs but only one output.

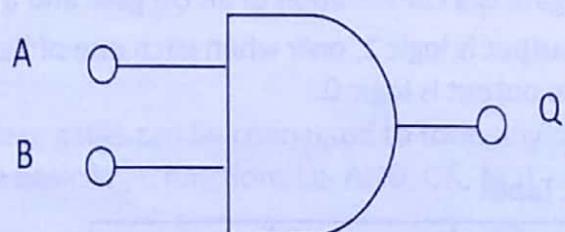
The output is logic 1 only when each one of its inputs is at logic 1.

The output is logic 0 even if one of its inputs is at logic 0.

Truth Table

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

symbol of AND



$$Q = A \cdot B$$

OR GATE

An OR gate may have two or more inputs but only one output.

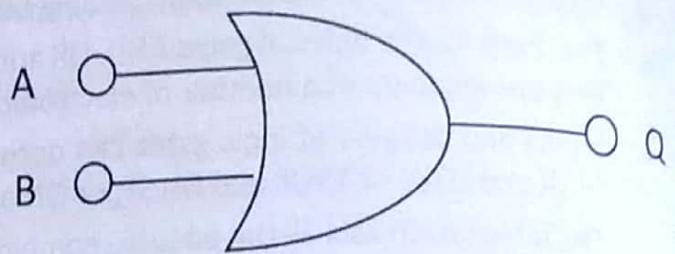
The output is logic 1, When any of the input at logic 1 .

The output is logic 0, only when each one of its inputs is in logic state.

Truth Table

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

symbol of OR



$$Q = A + B$$

NAND GATE

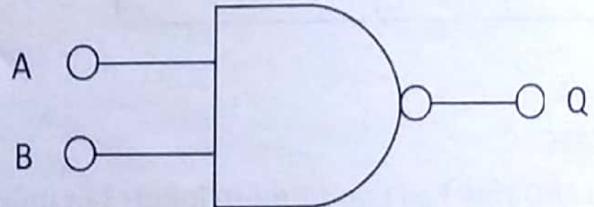
NAND gate is a combination of an AND gate and a NOT gate.

The output is logic 0 when each of the input is logic 1 and for any other combination of inputs, the output is logic 1.

Truth Table

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

symbol of NAND



$$Q = \overline{A \cdot B}$$

NOR GATE

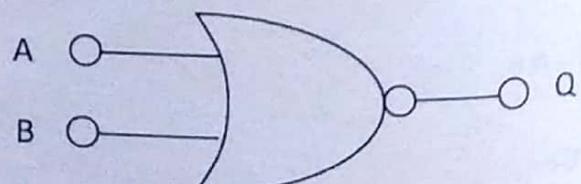
NOR gate is a combination of an OR gate and a NOT gate.

The output is logic 1, only when each one of its input is logic 0 and for any other combination of inputs, the output is logic 0.

Truth Table

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

symbol of NOR



$$Q = \overline{A+B}$$

Exclusive-OR

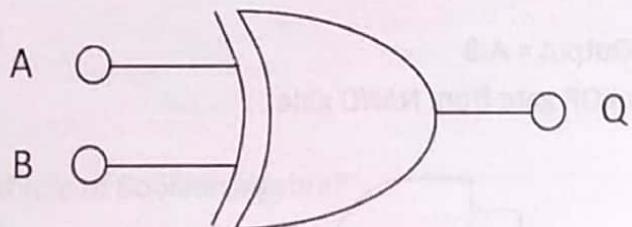
An X-OR gate is a two input, one output logic circuit.

The output is logic 1 when one and only one of its two inputs is logic 1. When both the inputs are logic 0 or when both the inputs are logic 1, the output is logic 0.

Truth Table

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

symbol of Ex-OR



$$Q = A \oplus B = \bar{A}B + A\bar{B}$$

EXCLUSIVE-NOR (X-NOR) GATE

An X-NOR gate is the combination of an XOR gate and a NOT gate

An X-NOR gate is a two input, one output logic circuit.

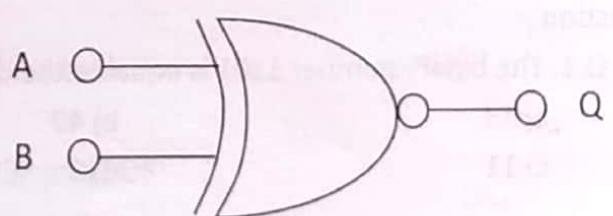
The output is logic 1 when one of the inputs is logic 0 and other is 1.

The output is logic 0 when both the inputs are equals.

Truth Table

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

symbol of Ex-NOR



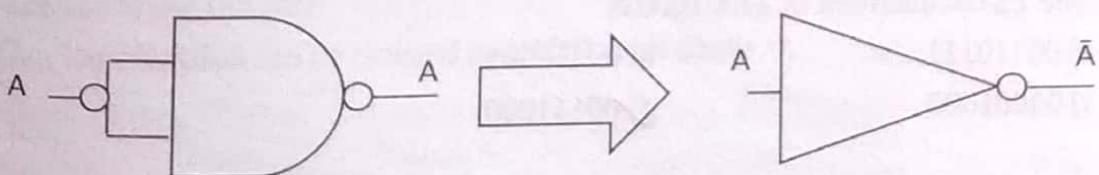
$$Q = \overline{A \oplus B} = AB + \bar{A}\bar{B}$$

Universal Gates

NAND and NOR are Universal Gates since these gates can be connected to form any other logic gates. Both NAND and NOR gates can perform all basic logic functions i.e. AND, OR, NOT

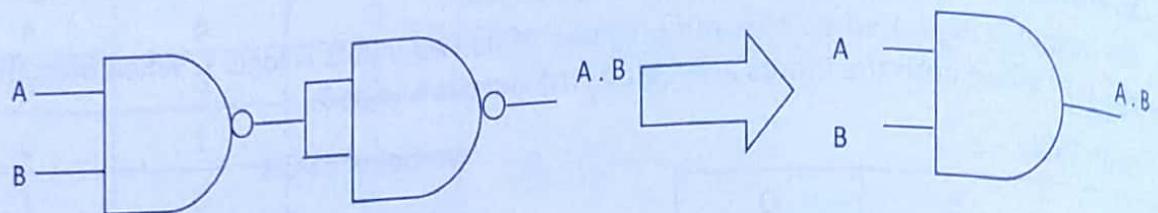
NAND GATE

a) Inverter from NAND gate



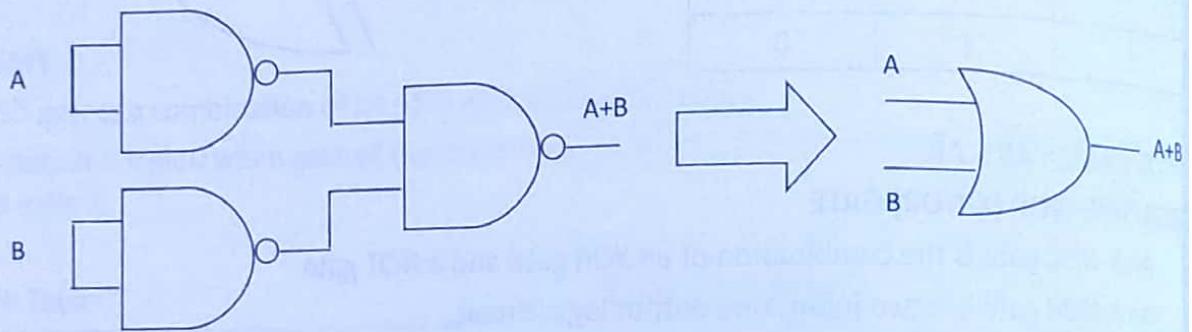
$$\text{Output } Q = \bar{A}$$

b) AND gate from NAND gate



Output = A.B

c) OR gate from NAND gate



Output = A+B

Similarly, we can see NOT, AND, OR gate by Using NOR as a Universal gate.

Question

Q.1. The binary number 1101 is equal to the decimal number

- a) 13 b) 49
c) 11 d) 3

Q.2. The decimal 17 is equal to the binary number

- a) 10010 b) 11000
c) 10001 d) 01001

Q.3. The 1's complement of 10111001 is

- a) 01000111 b) 01000110
c) 11000110 d) 10101010

Q.4. The 2's complement of 11001000 is

- a) 00110111 b) 00110001
c) 01001000 d) 00111000

Q.5. The output of gate is LOW when at least one of its inputs is HIGH. It is true for

- a) AND
- b) NAND
- c) OR
- d) NOR

Q.6. The output of gate is HIGH when at least one of its inputs is LOW. It is true for

- a) AND
- b) OR
- c) NAND
- d) NOR

Q.7. Which one of the following is not a valid rule of Boolean algebra?

- a) $A + 1 = 1$
- b) $A = \bar{A}$
- c) $A \cdot A = A$
- d) $A + 0 = A$

Q.8. The NAND gate can perform the invert function if the inputs are

- a) Connected together
- b) Left open
- c) Either (a) or (b)
- d) None of these

Q.9. If a three-input AND gate has eight input possibilities, how many of those possibilities will result in a HIGH output?

- a) 1
- b) 2
- c) 7
- d) 8

Q.10. In which function is each term known as minterm

- a) SOP
- b) POS
- c) Hybrid
- d) both SOP and POS

Q.11. For a certain two-input logic gate, the output is '1' for like inputs and '0' for unlike inputs.

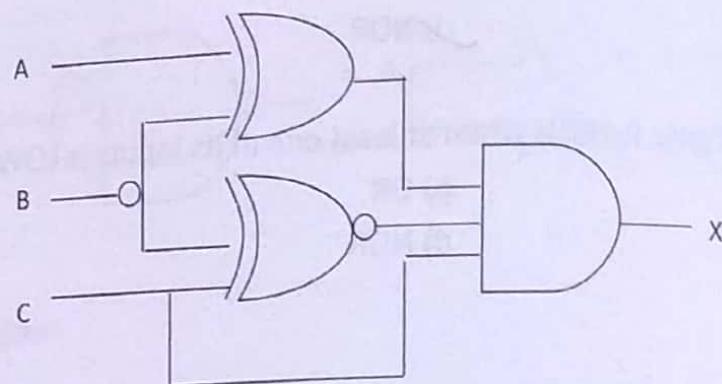
The logic gate is

- a) Ex-OR
- b) NAND
- c) NOR
- d) Ex-NOR

Q.12. A NAND gate is called a universal logic element because

- a) All digital computers use NAND gates
- b) All the minimization techniques are applicable for optimum NAND gate realization
- c) Everybody use this gate
- d) Any logic function can be realized by NAND gates alone

Q.13. In circuit shown below, for what input at the terminal A the output is $X = 1$?

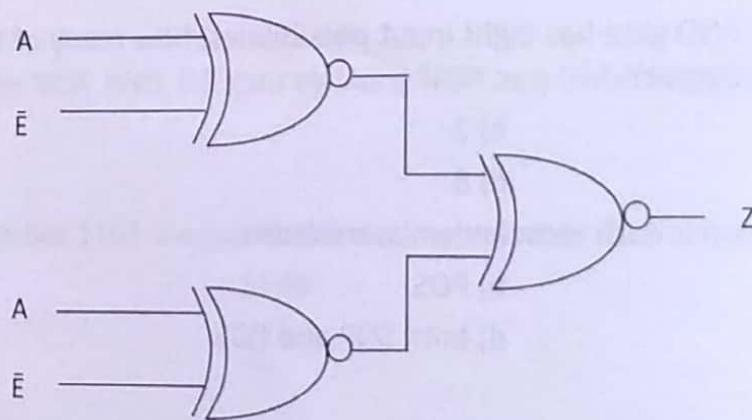


- a) $A = 1$
b) $A = B$
~~c) $A = 0$~~
d) $A = C$

Q.14 Complement of complement of $A' \cdot B + A \cdot B'$

- a) $A \cdot B + A' \cdot B'$
b) $(A' + B) \cdot (A + B')$
~~c) $A' \cdot B + A \cdot B'$~~
d) None of these

Q.15 In the following circuit the output Z is _____



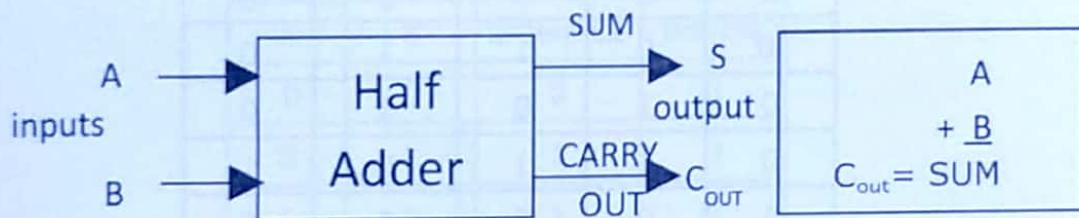
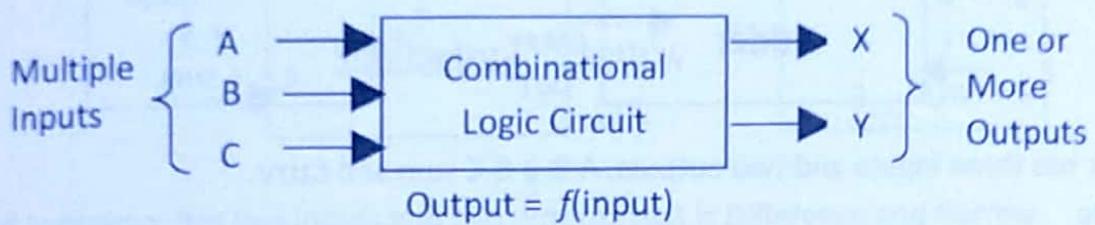
- a) $Z = A$
b) $Z = 1$
c) $Z = A'$
~~d) $Z = 0$~~

Answer

- | | | |
|------|-------|-------|
| 1) A | 6) C | 11) D |
| 2) C | 7) B | 12) D |
| 3) B | 8) A | 13) C |
| 4) D | 9) A | 14) C |
| 5) D | 10) A | 15) D |

Combinational Logic Circuit

Combinational Logic Circuits are made up from basic logic NAND, NOR or NOT gates that are "combined" or connected together to produce specified output for certain specific combination of input variables, with no storage. Its output is dependent on present input.



Half Adder

Half Adder has two inputs and two outputs that is sum and carry.

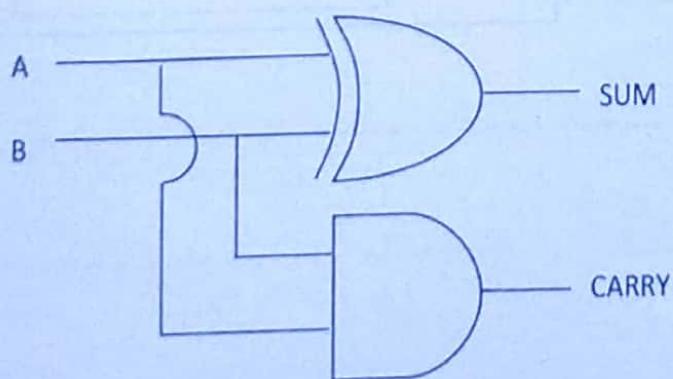
Truth table

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

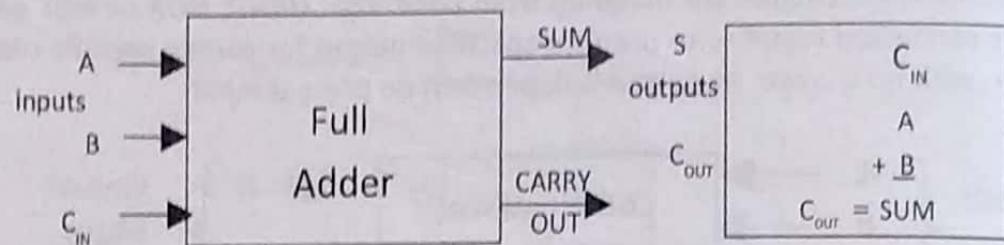
by using K map

$$\text{Sum} = A \oplus B$$

$$\text{Carry} = A \cdot B$$



Full Adder

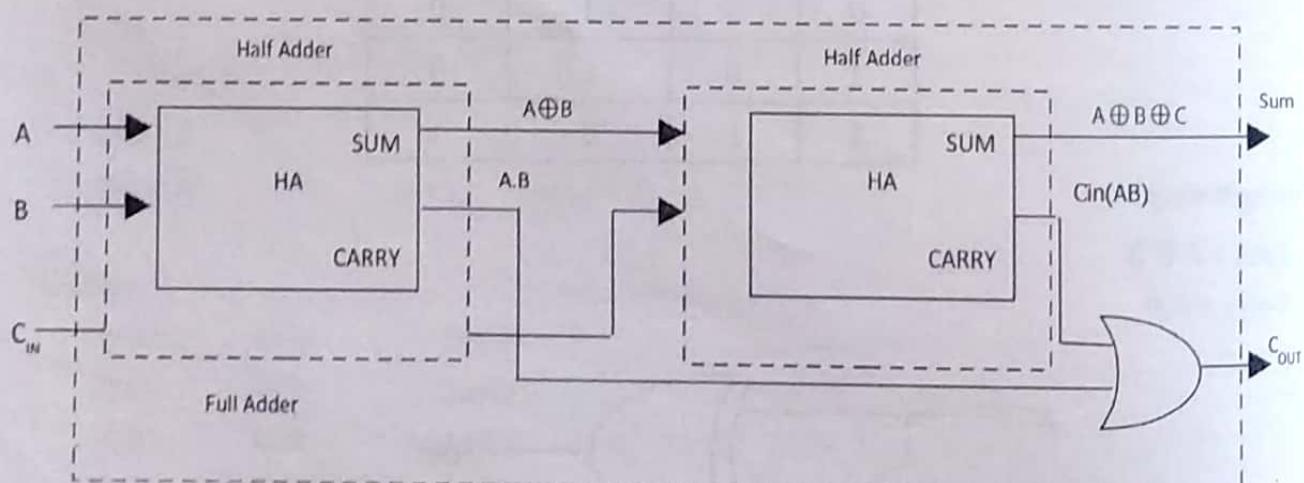


Full Adder has three inputs and two outputs, A ⊕ B ⊕ C sum and carry
Truth table

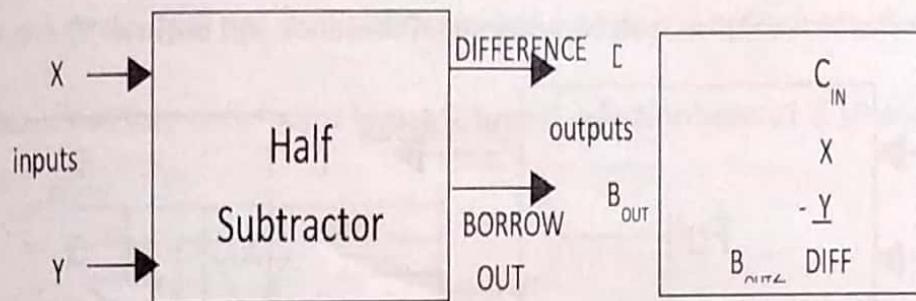
A	B	C _{in}	S	C _{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$\text{Sum} = A \oplus B \oplus C$$

$$\text{Carry} = AB + BC + AC$$



Half Subtractor



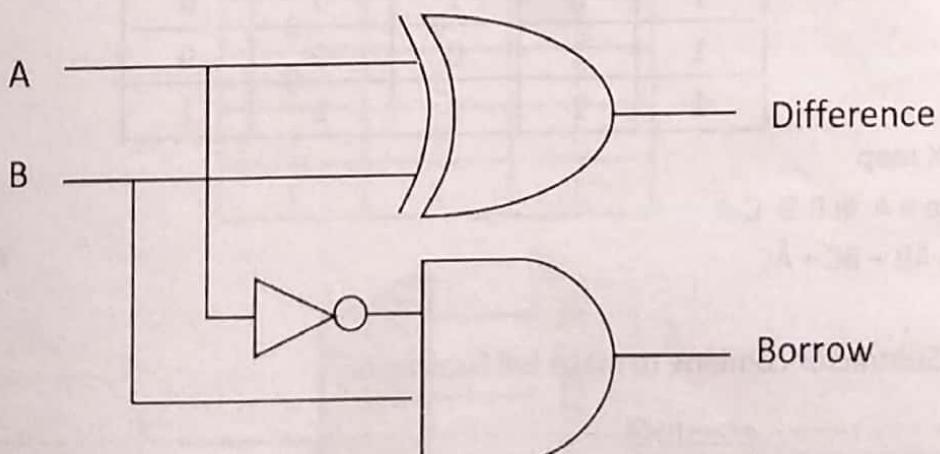
Half Subtractor has two inputs and two outputs that is Difference and Borrow.

Truth table

A	B	Difference	Borrow
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

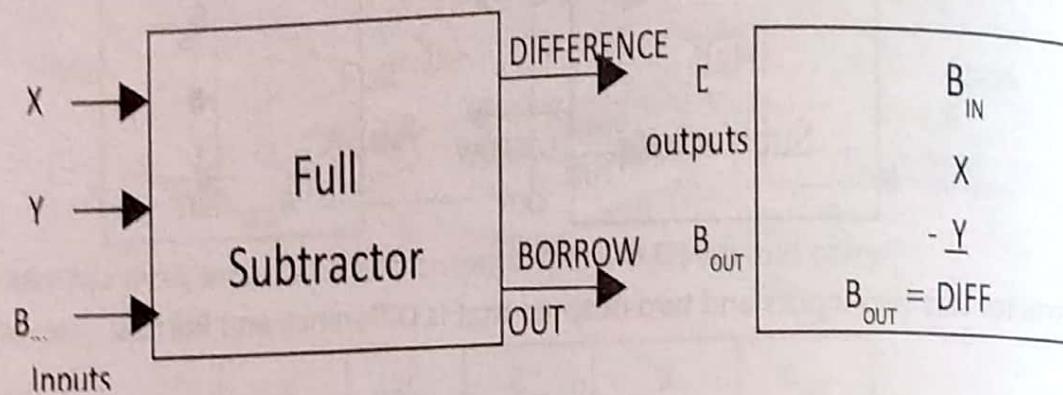
$$\text{Difference} = A \oplus B$$

$$\text{Borrow} = \bar{A} \cdot B$$



Full Subtractor

Full Subtractor has three inputs and two outputs, Difference and Borrow



Truth table

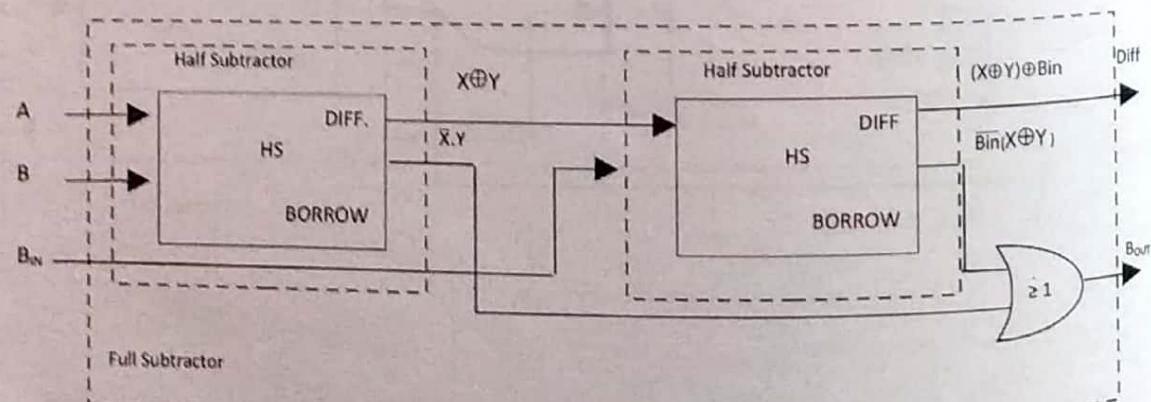
A	B	C _{in}	S	C _{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

by using K map

$$\text{Difference} = A \oplus B \oplus C$$

$$\text{Borrow} = \bar{A}B + BC + \bar{A}C$$

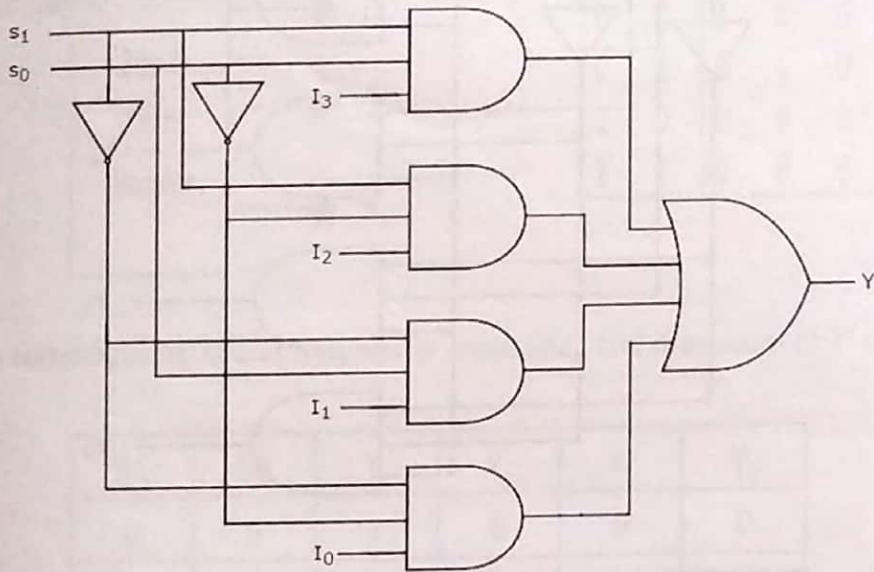
Two Half Subtractor combine to make full Subtractor



Multiplexer

Multiplexer is a device that has multiple inputs and a single line output. Multiplexer means many into one.

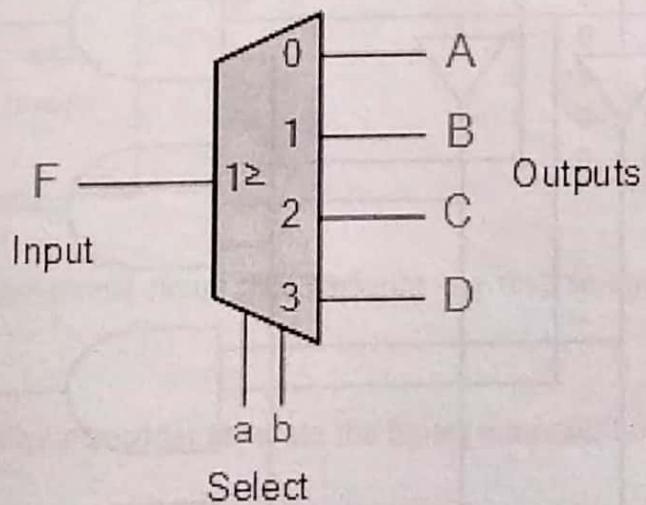
4x1 Multiplexer has four data inputs $I_3, I_2, I_1 \& I_0$, two selection lines s_1 & s_0 and one output Y .



Truth Table

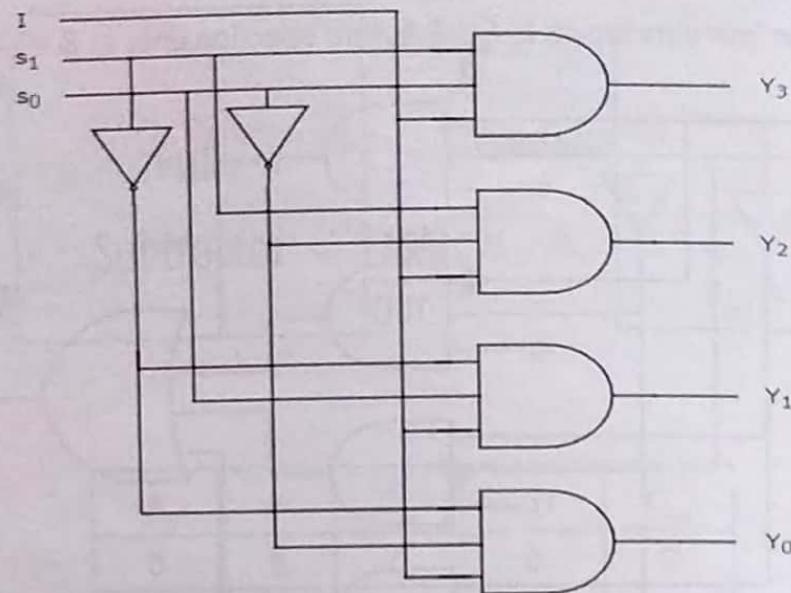
S_1	S_2	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Demultiplexer



Demultiplexer means one to many.

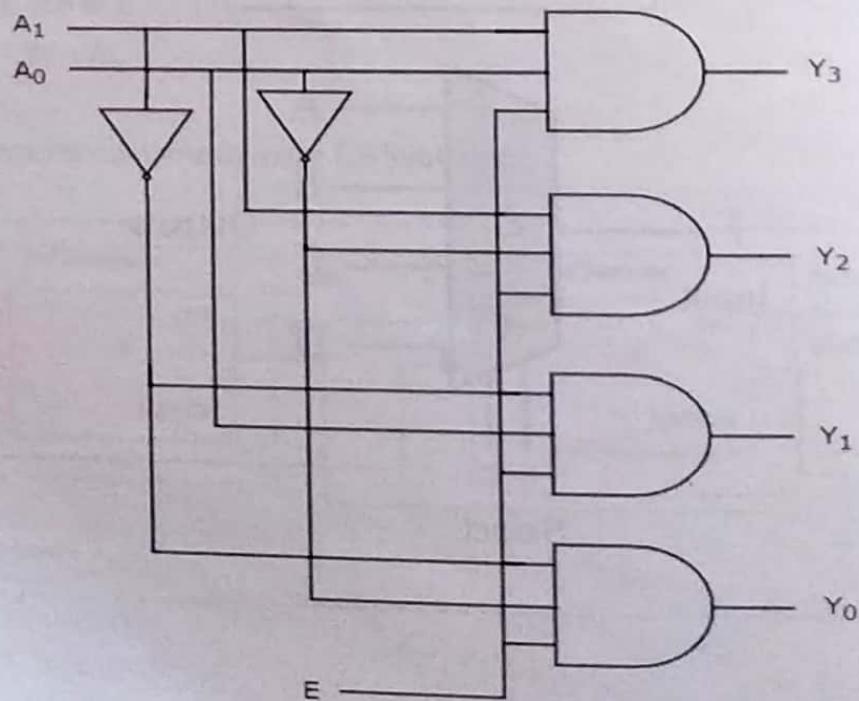
1x4 De-Multiplexer has one input I, two selection lines, s1 & s0 and four outputs Y3, Y2, Y1 & Y0

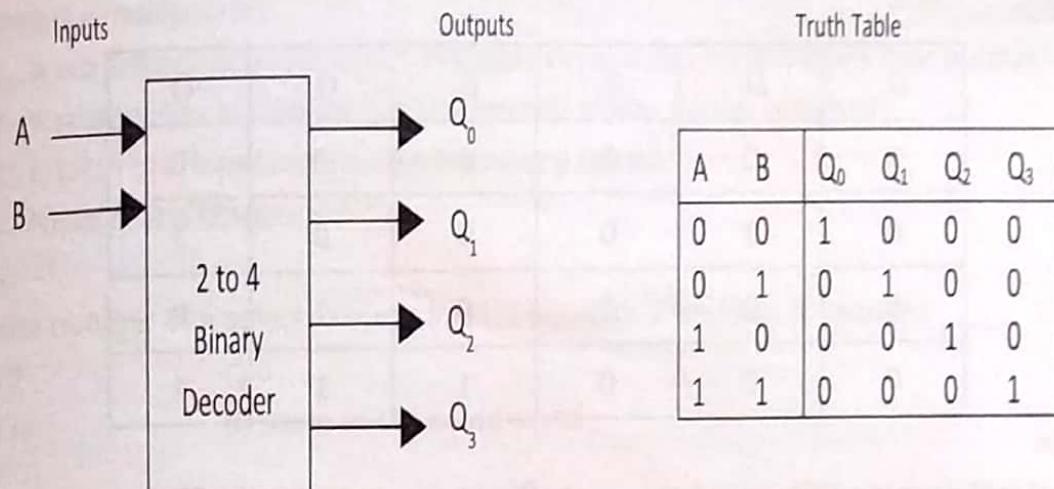


Truth Table

S1	S0	Y_4	Y_3	Y_2	Y_1
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

Decoder



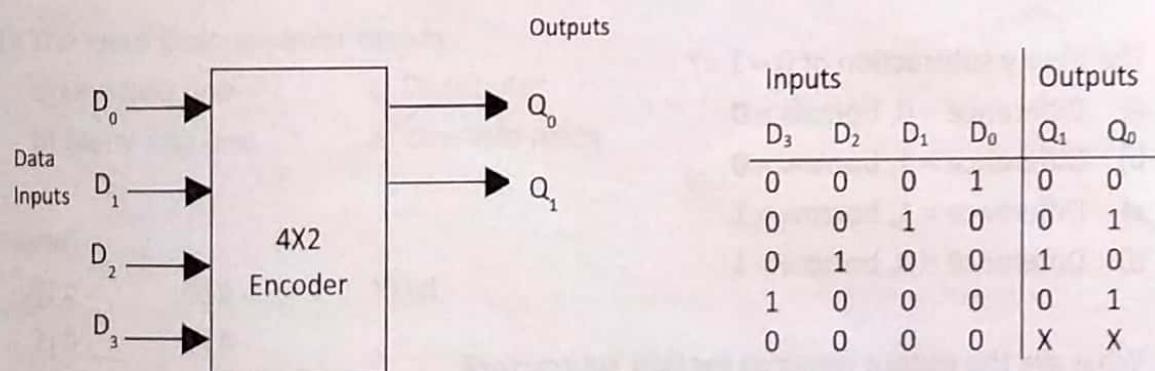


Decoder is a combinational circuit that has 'n' input lines and maximum of 2^n output lines.

Truth Table

S1	S0	Y_4	Y_3	Y_2	Y_1
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Encoder



An Encoder is a combinational circuit that performs the reverse operation of Decoder. It has maximum of $2n$ inputs.

The output lines of a digital encoder generate the binary equivalent of the input line

Truth table

D ₀	D ₁	D ₂	D ₃	Q ₀	Q ₁
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Question

- 1) What is the major difference between half-adders and full-adders?
 - a) Full-adders are made up of two half-adders
 - b) Full adders can handle double-digit numbers
 - c) Full adders have a carry input capability
 - d) Half adders can handle only single-digit numbers
- 2) What are the two types of basic adder circuits?
 - a) Sum and carry
 - b) Half-adder and full-adder
 - c) Asynchronous and synchronous
 - d) One and two's-complement
- 3) The binary subtraction of $0 - 1 = ?$
 - a) Difference = 0, borrow = 0
 - b) Difference = 1, borrow = 0
 - c) Difference = 1, borrow = 1
 - d) Difference = 0, borrow = 1
- 4) What are the output required for Half Subtractor?
 - a) Difference, Adder
 - b) Difference, Subtract
 - c) Sum, Carry
 - d) Difference, Borrow
- 5) For Half Subtractor the equation for Difference is
 - a) $A'B$
 - b) AB
 - c) $A'B + AB'$
 - d) AB'

- 6) What is a multiplexer?
a) It is a type of decoder which decodes several inputs and gives one output
 b) A multiplexer is a device which converts many signals into one
c) It takes one input and results into many output
d) None of the Mentioned

7) If the number of n selected input lines is equal to 2^m then it requires _____ select lines.
a) 2 c) n
 b) m d) None of the Mentioned

8) How many select lines would be required for an 8-line-to-1-line multiplexer?
a) 2 c) 8
b) 4 d) 3

9) The enable input is also known as
a) Select input c) Strobe
b) Decoded input d) Sink

10) Which of the following circuit can be used as parallel to serial converter?
 a) Multiplexer c) Decoder
b) Demultiplexer d) Digital counter

11) The word Demultiplexer means
a) One into one c) Distributor
b) Many into one d) One into many

Answer

- | | | |
|------|-------|-------|
| 1) c | 6) b | 11) d |
| 2) b | 7) b | |
| 3) c | 8) d | |
| 4) d | 9) c | |
| 5) c | 10) a | |

Sequential Circuit

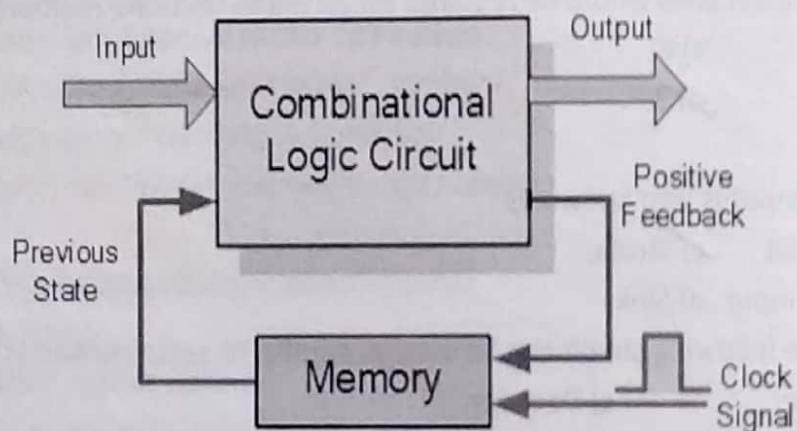
Sequential circuit is a circuit whose output depends upon the present input, previous output and the sequence in which the inputs are applied.

Difference Between Combinational Circuit and Sequential Circuit

In combinational circuit output depends upon present input at any instant of time and do not use memory. Hence previous input does not have any effect on the circuit. But sequential circuit has memory and depends upon present input and previous output.

Sequential circuits are slower than combinational circuits and these sequential circuits are harder to design.

The data stored by the memory element at any given instant of time is called the present state of sequential circuit.



Sequential logic circuits (SLC) are classified as

- (i) Synchronous SLC
- (ii) Asynchronous SLC

Sequential logic circuit that are controlled by clock are called synchronous SLC and those which are not controlled by clock are called asynchronous SLC.

Flip-Flop and Latch

A flip-flop or latch is a circuit that has two stable states and can be used to store information.

A flip-flop is a binary storage device capable of storing one bit of information. In a stable state, the output of a flip-flop is either 0 or 1.

Latch is an un-coded flip-flop, so output changes at any instant of time doesn't depend on clock. Flip-flop operate with clock, so output changes only at the clock signal.

Clock signals may be positive-edge triggered or negative-edge triggered.

Positive-edge triggered flip-flops are those in which state transitions take place only at positive going edge of the clock pulse.

Negative-edge triggered flip-flops are those in which state transition take place only at negative-going edge of the clock pulse.

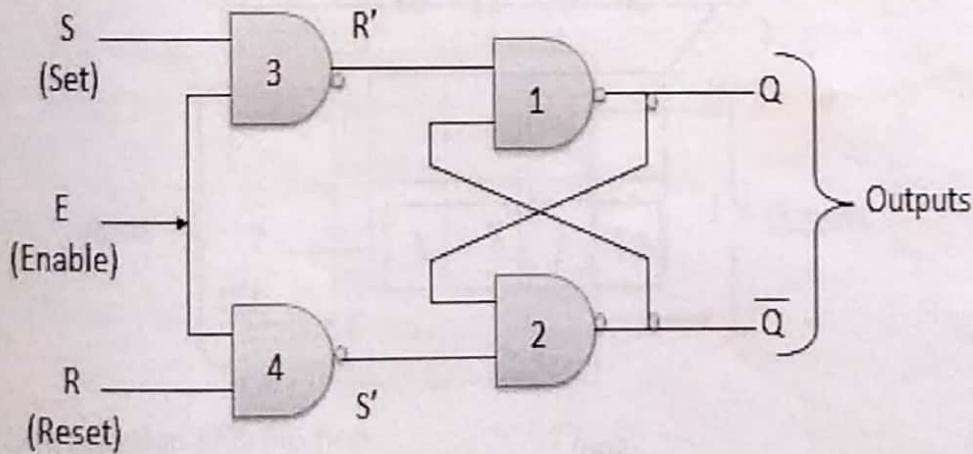
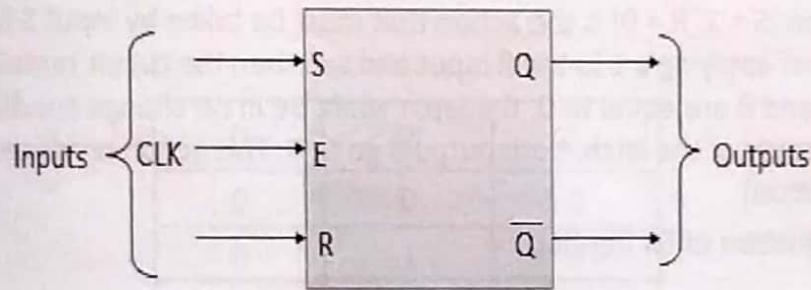
Type of flip-flops

- a) SR(set-reset) Flip-Flop
- b) D(data or delay) Flip-Flop
- c) T(toggle) Flip-Flop and
- d) JK Flip-Flop

SR Flip-flop

The SR latch is a circuit with two cross-coupled NOR gates or two cross-coupled NAND gates.

It has two outputs labeled Q and Q'. Two inputs are there labeled S for set and R for reset.



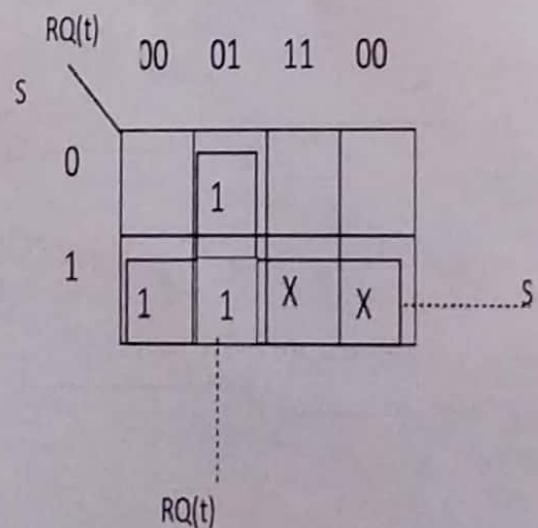
SR latch with two cross-coupled NAND gates. The circuit has NAND gates and as we know if any one of the input for NAND gate is LOW then its output will be HIGH and if both the inputs are HIGH then only the output will be LOW.

Truth table

Input		Previous input	Present output	State
S	R	Q_t	Q_{t+1}	
0	0	0	0	No Change
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	x	Indeterminate
1	1	1	x	

The first condition ($S = 1, R = 0$) is the action that must be taken by input S to bring the circuit to the set state. After that applying a 1 to the R input and $s=0$ then the circuit remains in the reset state. When both inputs S and R are equal to 0, the latch while be in no change condition. If a 1 is applied to both the S and R inputs of the latch, both outputs go to 0. This action produces an undefined state (Indeterminate condition).

Characteristic equation of SR flip-flop

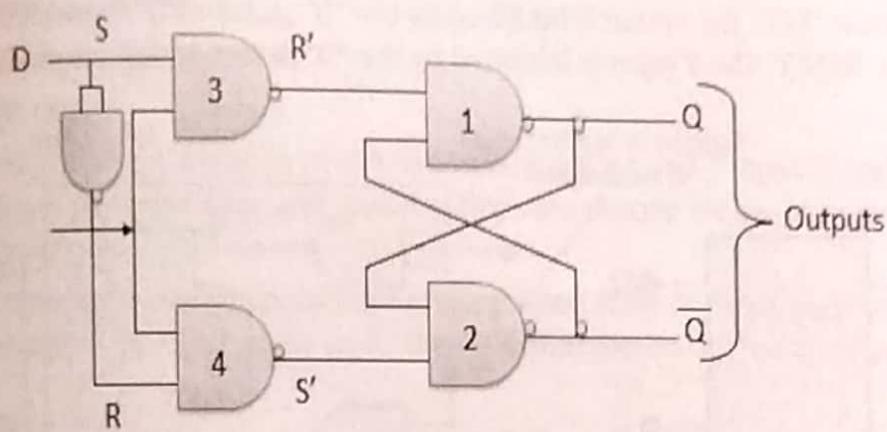


By using K-map we get the equation as

$$Q(t+1) = S + R'Q(t)$$

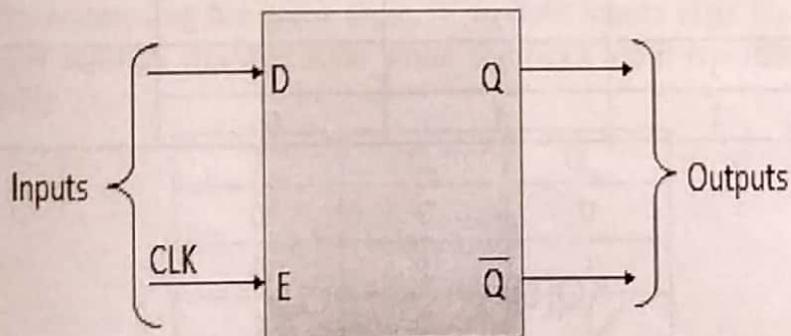
D Flip-flop

SR flip-flop can be converted into D just by placing NOT gate.



If $D = 1$, the Q output goes to 1, placing the circuit in the set state. If $D = 0$, output Q goes to 0, placing the circuit in the reset state. Whatever is the D input we get the same output hence; the circuit is called TRANSPARENT latch.

D	Q_t	Q_{t+1}
0	0	0
0	1	0
1	0	1
1	1	1



Characteristic equation of D flip-flop

$$Q(t+1) = D$$

JK Flip-Flop

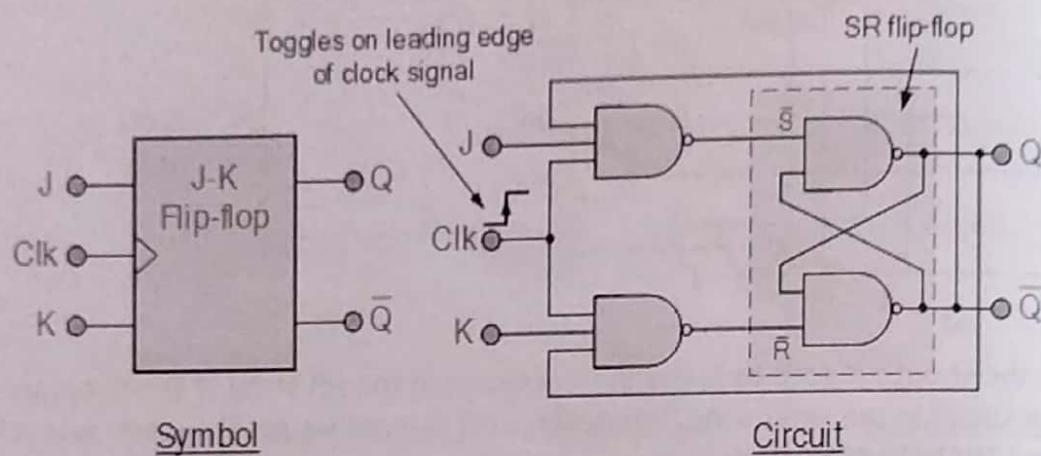
The JK flip-flop can be constructed by using basic SR latch and a clock. In this case the outputs Q and Q' are returned back and connected to the inputs of NAND gates.

Both the S and the R inputs of the SR bi-stable have been replaced by two inputs called J and K inputs, respectively after its inventor Jack and Kilby. Then this equates to: $J = S$ and $K = R$.

The two 2-input NAND gates of the gated SR bi-stable have now been replaced by two 3-input NAND gates with the third input of each gate connected to the outputs at Q and Q' .

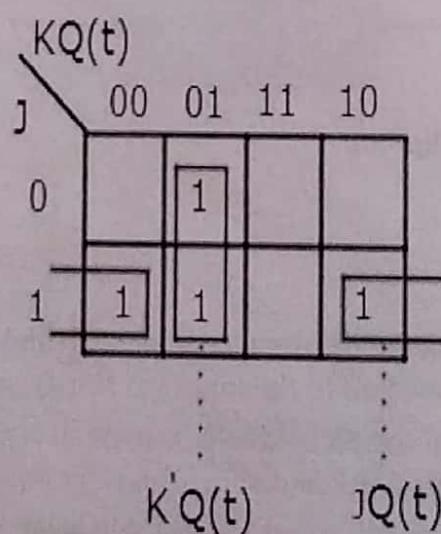
This cross coupling of the SR flip-flop allows the previously invalid condition of $S = 1$ and $R = 1$ to be used to produce a "toggle action" as the two inputs are now interlocked.

If the circuit is now "SET" the J input is inhibited by the "0" status of 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.



Truth table

Input		Previous input	Present output	State
J	K	Q	Q_{t+1}	
0	0	0	0	No Change
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	1	Toggle
1	1	1	0	



Characteristic equation of JK

By using K-map we get Equation

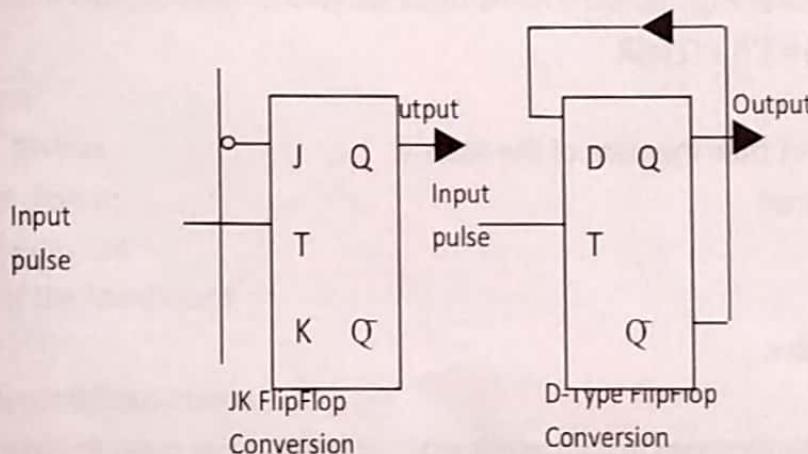
$$Q(t+1) = JQ(t)' + K'Q(t)$$

Race around condition occur in JK flip-flop when both J and K input is high. If you keep this condition for longer period of time and then flip-flops are disable we cannot predict the output this is race around condition.

We can remove race around condition by Master and Slave JK flip-flop. Where "Master" works on, the leading edge of the clock pulse while the "Slave" works on the falling edge of the clock pulse.

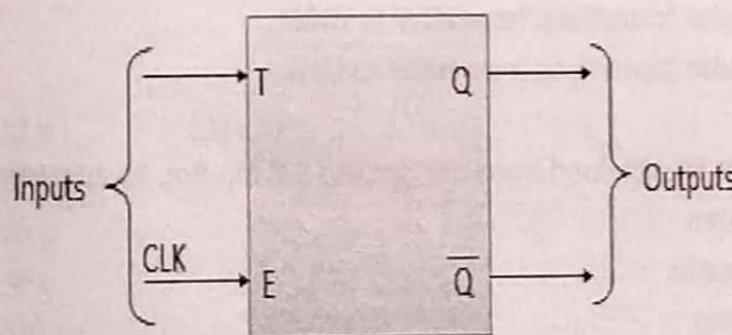
T Flip-flop

The T-type (toggle) flip-flop is a single input bistable



It is obtained by connecting the same input 'T' to both inputs of JK flip-flop (by shorting both the inputs). The toggle flip-flop changes state when the clock input is applied, $T = 1$ and remains unchanged when $T = 0$.

T	Q	Q_{t+1}
0	0	0
0	1	1
1	0	1
1	1	0



Characteristic Equation of T Flip-flop

$$Q(t+1) = T'Q(t) + TQ'(t)$$

Questions

- 1) When both inputs of a J-K flip-flop are one, the output will
 - a) Be invalid
 - b) Change
 - c) Not change
 - d) Toggle
- 2) The characteristic equation of S-R latch is
 - a) $Q(n+1) = S + Q(n)R'$
 - b) $Q(n+1) = SR + Q(n)R$
 - c) $Q(n+1) = S' + Q(n)R$
 - d) $Q(n+1) = S'R + Q'(n)R$
- 3) If $S'=1$ & $R'=1$ then the state of the latch is:
 - a) No change
 - b) Set
 - c) Reset
 - d) Forbidden
- 4) When is a flip-flop said to be transparent?
 - a) When the Q output is opposite the input
 - b) When the Q output follows the input
 - c) When you can see through the IC packaging
 - d) None of the Mentioned
- 5) On a positive edge-triggered S-R flip-flop, the outputs reflect the input condition when
 - a) The clock pulse is LOW
 - b) The clock pulse is HIGH
 - c) The clock pulse transitions from LOW to HIGH
 - d) The clock pulse transitions from HIGH to LOW
- 6) A J-K flip-flop can be obtained from the clocked S-R flip-flop by augmenting
 - a) Two AND gates
 - b) Two NAND gates
 - c) Two NOT gates
 - d) None of the Mentioned

7) A D flip-flop can be constructed from an _____ flip-flop.

- a) S-R
- b) J-K
- c) T
- d) None of the Mentioned

8) Which of the following is correct for a D latch?

- a) The output toggles if one of the inputs is held HIGH
- b) Q output follows the input D when the enable is HIGH
- c) Only one of the inputs can be HIGH at a time
- d) The output complement follows the input when enabled

9) Flip-flops are

- a) Stable devices
- b) Astable devices
- c) Bistable devices
- d) None of the Mentioned

10) The term synchronous means

- a) The output changes state only when any of the input is triggered
- b) The output changes state only when the clock input is triggered
- c) The output changes state only when the input is reversed
- d) None of the Mentioned

11) Which of the following flip-flops is free from race around problem?

- a) T flip-flop
- b) SR flip-flop
- c) Master-Slave Flip-flop
- d) None of the Mentioned

Answer

- | | | |
|------|-------|-------|
| 1) d | 6) a | 11) c |
| 2) a | 7) a | |
| 3) a | 8) b | |
| 4) b | 9) c | |
| 5) c | 10) b | |

Shift Registers

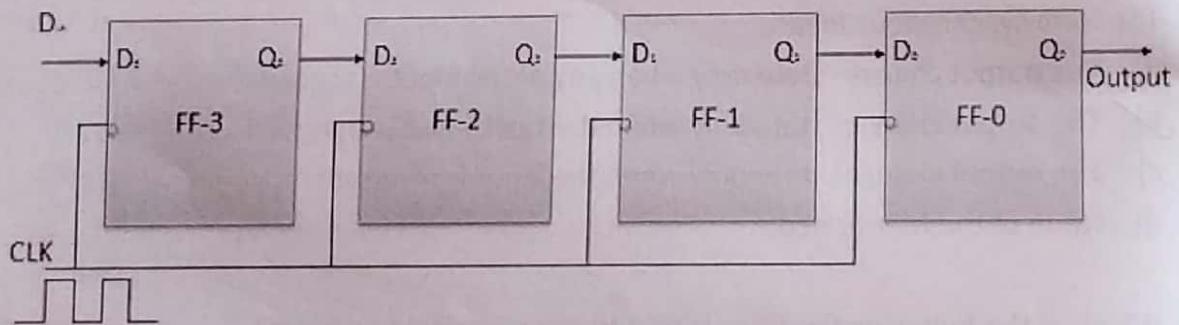
Flip-flop is a 1 bit memory cell which can be used for storing the digital data. To increase the storage capacity in terms of number of bits, we have to use a group of flip-flop. Such a group of flip-flop is known as a Register. The n-bit register will consist of n number of flip-flop and it is capable of storing an n-bit word.

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 modes of operations of a shift register.

- Serial Input Serial Output
- Serial Input Parallel Output
- Parallel Input Serial Output
- Parallel Input Parallel Output

Serial In Serial Out

Let all the flip-flop be initially in the reset condition i.e. $Q_3 = Q_2 = Q_1 = Q_0 = 0$. Four-bit binary number 1 1 1 1 is waiting to enter 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.



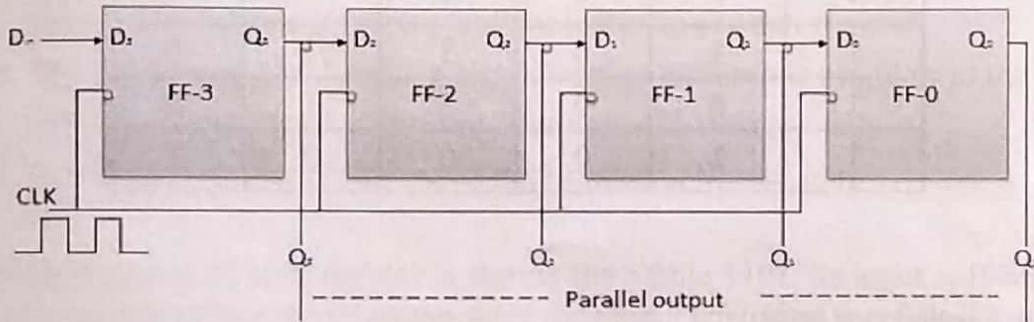
Operation

- 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} . 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 $Q_3 Q_2 Q_1 Q_0 = 1111$.

	CLK	$D_{in} = Q_3$	$Q_3 = D_3$	$Q_2 = D_2$	$Q_1 = D_1$	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

Serial In Parallel Out



The data is entered serially and taken out in parallel way. Data is loaded bit by bit.

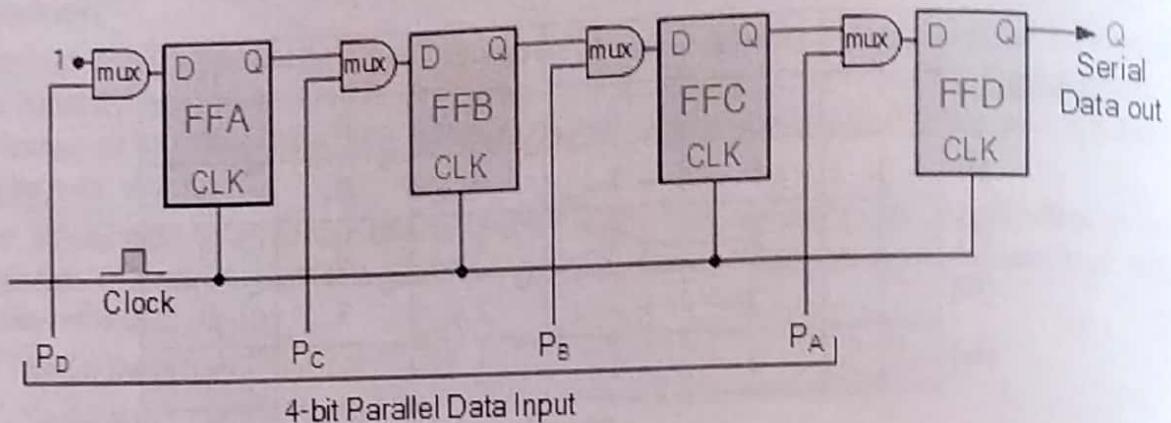
As soon as the data loading gets completed, all the flip-flops contain their required data, the outputs are enabled so that all the loaded data is made available over all the output lines at the same time.

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.

	CLK	D_{in}	Q_3	Q_2	Q_1	Q_0
Initially	1		0	0	0	0
(i)	1	1	1	0	0	0
(ii)	1	1	1	1	0	0
(iii)	1	1	1	1	1	0
(iv)	1	0	1	1	1	0

Parallel In Serial Out (PISO)

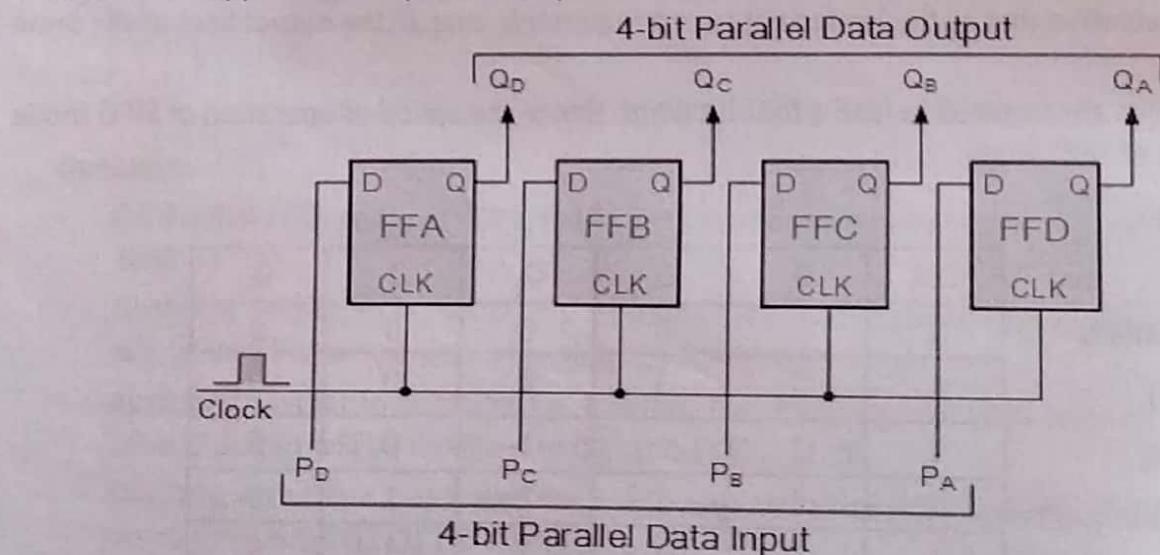
For parallel in, serial out shift register the data bits are entered simultaneously into their respective stages on parallel lines, rather than on bit by bit basis on one line as with serial data inputs, but the data bits are transferred out of the register serially, i.e., on a bit by bit basis over a single line.



	CLK	D _{in}	Q ₃	Q ₂	Q ₁	Q ₀
Initially	1		0	0	0	0
(i)	1	1	1	1	0	1
(ii)	1	1	0	1	1	0
(iii)	1	0	0	0	1	0
(iv)	1	1	0	0	0	1

Parallel In Parallel Out (PIPO)

In a parallel in, parallel out shift register, the data entered into the register in parallel form and also the data taken out of the register in parallel form. Immediately following the simultaneous entry of all data bits appear on the parallel outputs.



BIDIRECTIONAL SHIFT REGISTER

In bidirectional shift register is one in which the data bits can be shifted from left to right or from right to left.

The figure shown below the logic diagram of a 4-bit serial in, serial out, bidirectional (shift-left, shift-right) shift register.

Right /Left is the mode signal. When Right /Left is a 1, the logic circuit works as a shift right shift register. When Right /Left is a 0, the logic circuit works as a shift left shift register.

UNIVERSAL SHIFT REGISTERS

Universal shift register is a bidirectional register, whose input can be either in serial form or in parallel form and whose output also can be either in serial form or parallel form. The universal shift register can be realized using multiplexers

Question

- 1) A shift register is defined as
 - a) The register capable of shifting an information to another register
 - b) The register capable of shifting an information either to the right or to the left
 - c) The register capable of shifting an information to the right only
 - d) The register capable of shifting an information to the left only

- 2) A bidirectional 4-bit shift register is storing the nibble 1101. Its input is HIGH. The nibble 1011 is waiting to be entered on the serial data-input line. After three clock pulses, the shift register is storing _____
 - a) 1101
 - b) 0111
 - c) 0001
 - d) 1110

- 3) What type of register would have a complete binary number shifted in one bit at a time and have all the stored bits shifted out at a time?
 - a) Parallel-in Parallel-out
 - b) Parallel-in Serial-out
 - c) Serial-in Parallel-out: *Serial-in - serial out*
 - d) Serial-in Serial-out

- 4) The group of bits 10110111 is serially shifted (right-most bit first) into an 8-bit parallel output shift register with an initial state 11110000. After two clock pulses, the register contains
 - a) 10111000
 - b) 10110111
 - c) 11110000
 - d) 11111100

- 5) The bit sequence 0010 is serially entered (right-most bit first) into a 4-bit parallel out shift register that is initially clear. What are the Q outputs after two clock pulses?
- a. 0000 b. 0010 c. 1000 d. 1111

Answer

- 1) b
- 2) b
- 3) c
- 4) d
- 5) a

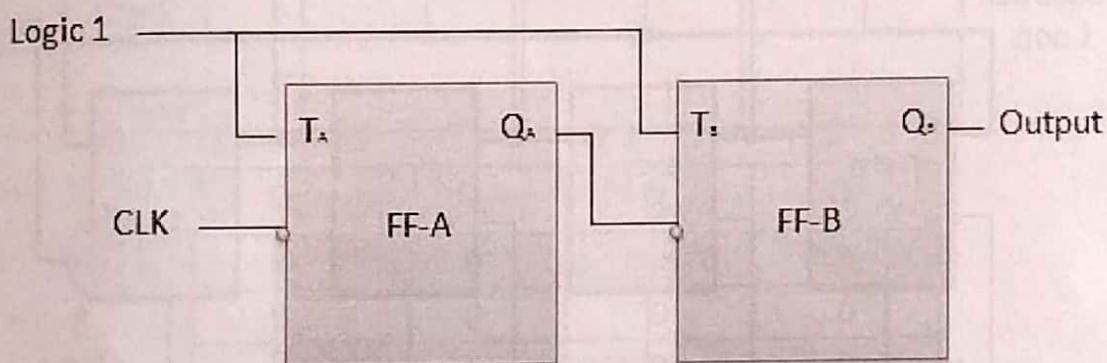
Counter

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. Counters are of two types.

- Asynchronous or ripple counters.
- Synchronous counters.

Asynchronous or ripple counters

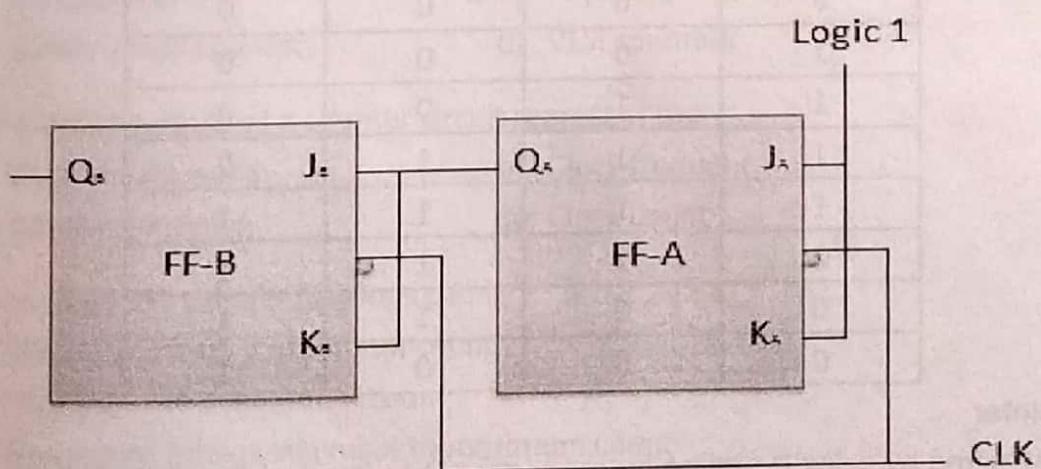
The logic diagram of a 2-bit ripple up counter. The toggle (T) flip-flop are being used. But we can use the JK flip-flop also with J and K connected permanently to logic 1. External clock is applied to the clock input of flip-flop A and QA output is applied to the clock input of the next flip-flop i.e.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



Synchronous counters

If the "clock" pulses are applied to all the flip-flops in a counter simultaneously, then such a counter is called as synchronous counter.

A 2-bit synchronous counter has J_A and K_A inputs of FF-A tied to logic 1. So FF-A will work as a toggle flip-flop. The J_B and K_B inputs are connected to Q_A.



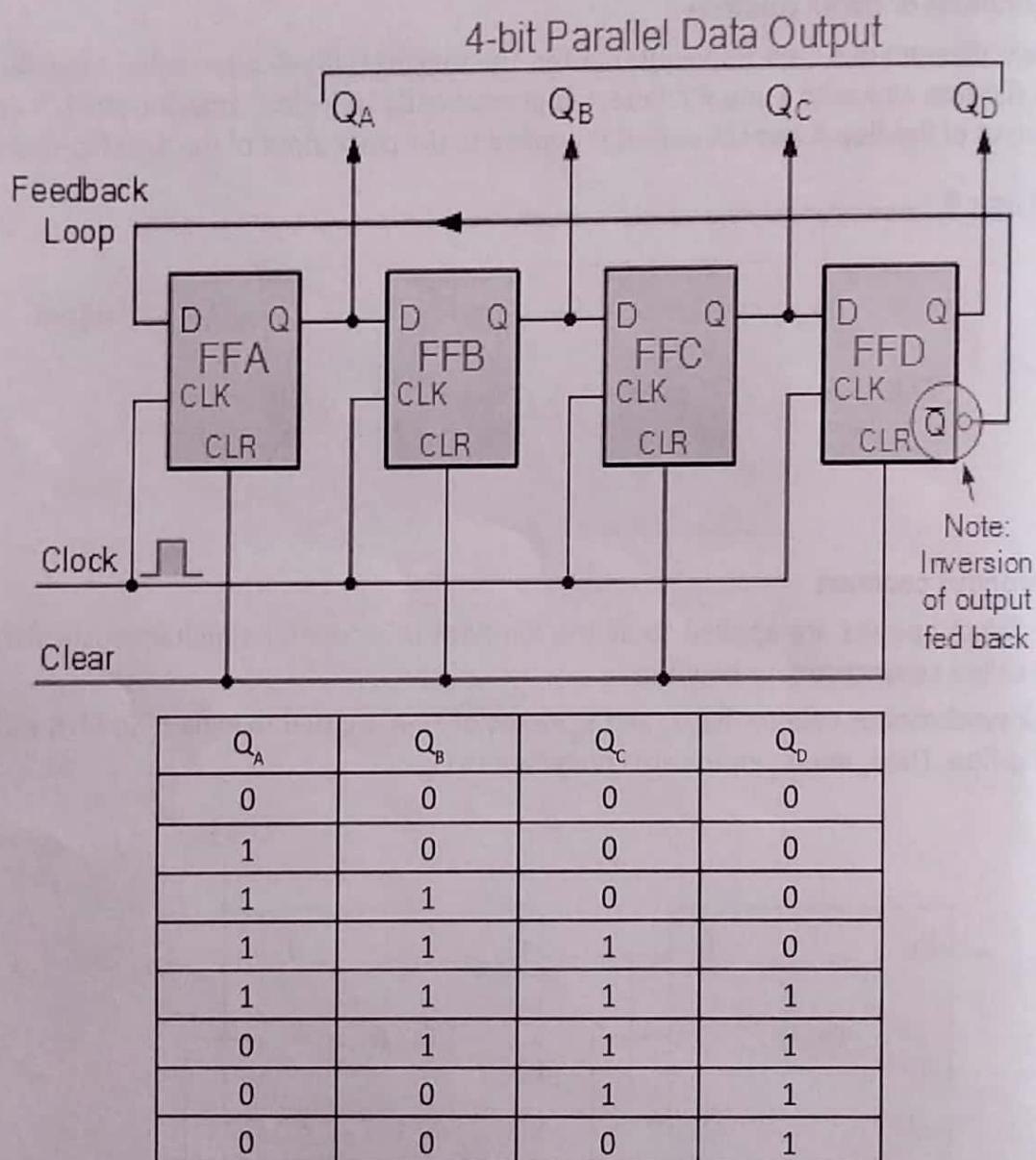
Up/Down Counter

Up/Down counter is used to control the direction of the counter through a certain sequence.

An 'N' bit binary counter consists of 'N' T flip-flops. If the counter counts from 0 to $2^n - 1$, then it is called as binary up counter. Similarly, if the counter counts down from $2^n - 1$ to 0, then it is called as binary down counter.

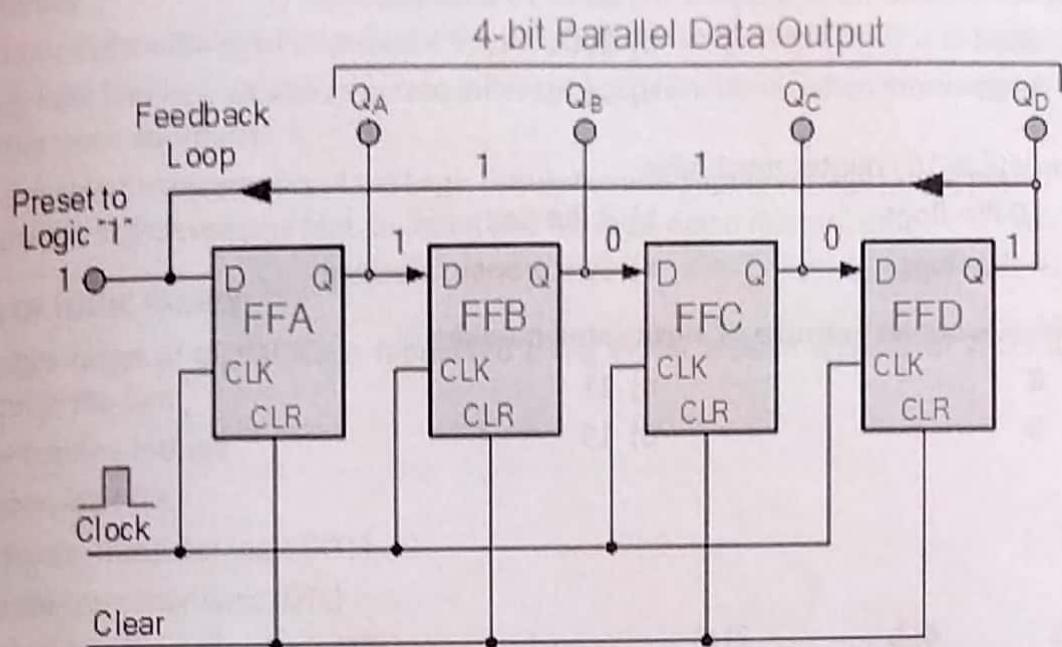
Johnson's counter

It is constructed by using D-flip-flop by connecting Q' to input of first D flip-flop. The complement output of final flip-flop is connected to the input of first flip-flop. Initially, all the flip-flops are reset. After the first clock pulse FFO is set and the remaining flip-flops are reset. After fourth clock pulse, all FF's are set. After fifth clock pulse FFO is reset and the remaining flip-flops are set. There are 8 different output conditions creating a mod-8 Johnson counter. Johnson counter is also called Twisted ring counter or divide by 2N counter.



Ring Counter

It is constructed by using D-flipflop by connecting Q to input of first D flipflop. The outputs of the final flip-flop are connected to the inputs of the first flip-flop. To start the counter, first flip-flop is set using preset facility and the remaining flip-flops are reset using clear input. When "clock" signal arrives, this set condition continues to shift around the ring. Ring counter using D flip-flops are made by connecting the Q output of the last flip-flop to the D input of the first flip-flop. As it can be seen from the truth table, there are four unique output states for this counter, rendering a mod-4 ring counter. Ring counter is called a **divide by N counter**, where N is the number of flip-flops.



Q_A	Q_B	Q_C	Q_D
0	0	0	0
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Question

Q.1. Ripple counters are also called

- a) SSI counters
- b) Asynchronous counters
- c) Synchronous counters
- d) VLSI counters

Q.2. The parallel outputs of a counter circuit represent the

- a) Parallel data word
- b) Clock frequency
- c) Counter modulus
- d) Clock count

Q.3. A down counter using n-flip-flops count

- a) Downward from a maximum count
- b) Upward from a minimum count
- c) Downward from a minimum to maximum count
- d) None of the Mentioned

Q.4. In a 3-bit asynchronous down counter, at the first negative transition of the clock, the counter content becomes

- a) 000
- b) 101
- c) 111
- d) 010

Q.5. The terminal count of a typical modulus-10 binary counter is

- a) 0000
- b) 1001
- c) 1010
- d) 1111

Q.6. A modulus-16 counter must have _____

- a) 10 flip-flops
- b) 2 flip-flops
- c) 4 flip-flops
- d) Synchronous clocking

Q.7. Which is not an example of a truncated modulus?

- a) 8
- b) 11
- c) 9
- d) 15

Answer

- 1) b
- 4) b
- 7) a
- 2) d
- 5) c
- 3) a
- 6) c

Logic Family

A circuit configuration used to produce a type of digital integrated circuit is called Logic Family. By using logic families, we can generate different logic functions, when fabricated in the form of an IC with the same approach.

Some common Characteristics of the Logic Family include Supply voltage range, power dissipation, input and output logic levels, current sourcing and fan-out, noise margin, etc.

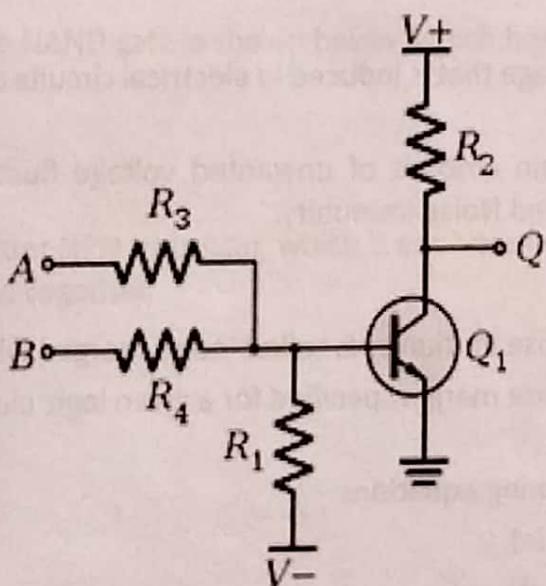
TYPES OF LOGIC FAMILY

The entire range of digital ICs is fabricated using either bipolar devices or MOS devices or a combination of the two.

Bipolar families include

- Diode logic (DL)
- Resistor-Transistor logic (RTL)
- Diode-transistor logic (DTL)
- Transistor- Transistor logic (TTL)
- Emitter Coupled Logic (ECL),
- Integrated Injection logic (I_2L)

Resistor Transistor Logic



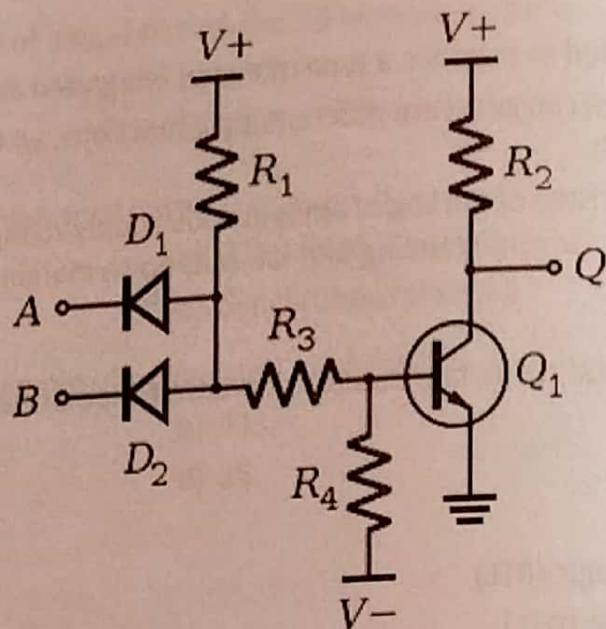
In RTL (resistor transistor logic), all the logic is implemented using resistors and transistors. One basic thing about the transistor (NPN), is that HIGH at input causes output to be LOW (i.e. like an inverter). In the case of PNP transistor, the LOW at input causes output to be HIGH.

Advantage:

- (A) Less number of Transistors

Disadvantage:

- 1) High Power Dissipation
- 2) Low Fan In



Diode Transistor Logic

In DTL (Diode transistor logic), all the logic is implemented using diodes and transistors.

Disadvantage:

Propagation Delay is Larger

Noise Immunity

Noise is the unwanted voltage that is induced in electrical circuits and can present a threat to the poor operation of the circuit.

Ability to tolerate a certain amount of unwanted voltage fluctuation on its inputs without changing its output state is called Noise Immunity.

Noise Margin

A measure of a circuit's noise immunity is called 'noise margin' which is expressed in volts.

There are two values of noise margin specified for a given logic circuit: the HIGH (VNH) and LOW (VNL) noise margins.

These are defined by following equations

$$VNH = VOH(\text{Min}) - VIH(\text{Min})$$

$$VNL = VIL(\text{Max}) - VOL(\text{Max})$$

Power Dissipation

A logic gate draws ICCH current from the supply when the gate is in the HIGH output state, draws ICCL current from the supply in the LOW output state.

Average power is

$$PD = VCC \cdot ICC \text{ where } ICC = (ICCH + ICCL) / 2$$

Propagation Delay time

When a signal passes through a logic circuit, it always experiences a time delay. A change in the output level occurs for a short time, called 'propagation delay time',

Fan Out of Gates

When the output of a logic gate is connected to one or more inputs of other gates, a load on the driving gate is created. There is a limit to the number of load gates that a given gate can drive. This limit is called the 'Fan-Out' of the gate.

TRANSISTOR-TRANSISTOR LOGIC

In Transistor-Transistor logic or TTL, logic gates are built only around transistors. TTL was developed in 1965. Through the years basic TTL has been improved to meet performance requirements. There are many versions or families of TTL.

For example, are Standard TTL, High Speed TTL, Low Power TTL, Schottky TTL etc.

All TTL logic families have three configurations for outputs

1. Totem pole output
2. Open collector output
3. Tristate output

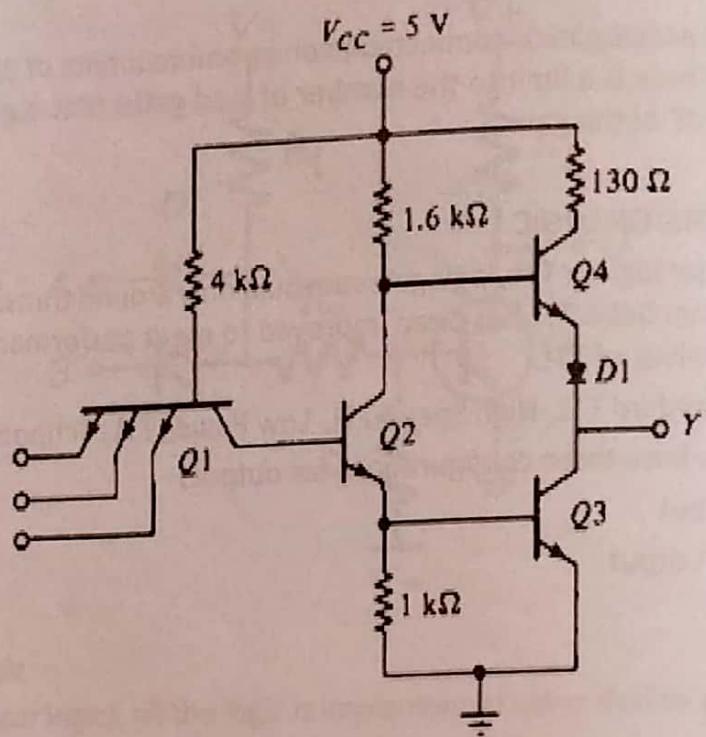
Totem pole output:

Addition of an active pull up circuit in the output of a gate is called totem pole. To increase the switching speed of the gate which is limited due to the parasitic capacitance at the output totem pole is used.

The circuit of a totem-pole NAND gate is shown below, which has got three stages

1. Input Stage
2. Phase Splitter Stage
3. Output Stage

Transistor Q1 is a two-emitter NPN transistor, which is equivalent two NPN transistors with their base and emitter terminals tied together.



The two emitters are the two inputs of the NAND gate

- (A) In TTL technology multiple emitter transistors are used for the input devices

When there is large negative voltage at input, the diode conducts and shorting it to the ground Q2 provides complementary voltages for the output transistors Q3 and Q4.

The combination of Q3 and Q4 forms the output circuit often referred to as a totem pole arrangement (Q4 is stacked on top of Q3). In such an arrangement, either Q3 or Q4 conducts at a time depending upon the logic status of the inputs

Diode D1 ensures that Q4 will turn off when Q2 is on (HIGH input)

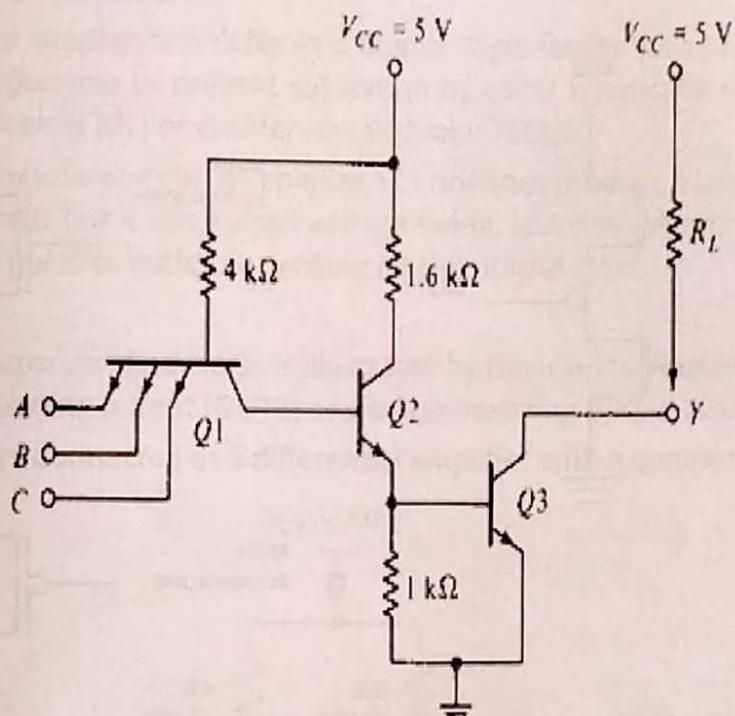
The output Y is taken from the top of Q3

Advantages of Totem Pole Output

The features of this arrangement are

1. Low power consumption
2. Fast switching
3. Low output impedance

OPEN COLLECTOR OUTPUT



In open-collector output we can see that the circuit elements associated with Q4 in the totem-pole circuit are missing and the collector of Q3 is left open-circuited, hence the name open-collector.

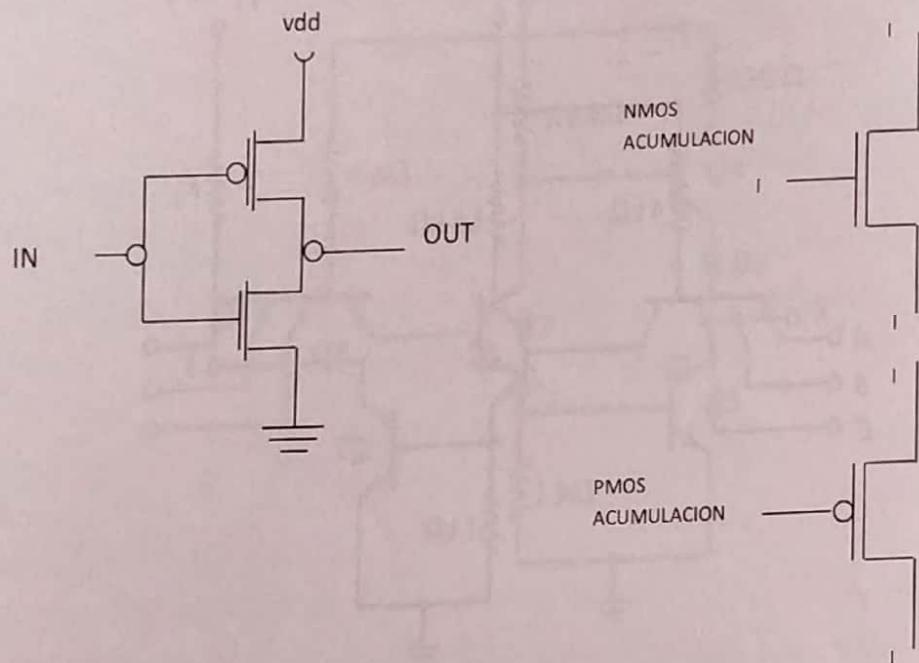
An open-collector output can present a logic LOW output. Since there is no internal path from the output Y to the supply voltage V_{CC} , the circuit cannot present a logic HIGH on its own.

Advantages of Open Collector Outputs

- 1) Open-collector outputs can be tied directly together which results in the logical ANDing of the outputs. Thus, the equivalent of an AND gate can be formed by simply connecting the outputs.
- 2) Increased current levels - Standard TTL gates with totem-pole outputs can only provide a HIGH current output of 0.4 mA and a LOW current of 1.6 mA. Many open-collector gates have increased current ratings.
- 3) Different voltage levels - A wide variety of output HIGH voltages can be achieved using open-collector gates.

Disadvantage of open-collector gates

They have slow switching speed. This is because the value of pull-up resistor, which results in a relatively long time Constants.



MOS stands for Metal Oxide Semiconductor and this technology uses FETs.

MOS can be classified into three sub-families:

- 1) PMOS (P-channel)
- 2) NMOS (N-channel)
- 3) CMOS (Complementary MOS, most common)

The following simplified symbols are used to represent MOSFET transistors in most CMOS. The gate of a MOS transistor controls the flow of the current between the drain and the source. The MOS transistor can be viewed as a simple ON/OFF switch.

Advantages of MOS Digital ICs

- 1) They are simple and inexpensive to fabricate.
- 2) Can be used for Higher integration and consume little power.

Disadvantages of MOS Digital ICs

- 1) There is possibility for Static-electricity damage.
- 2) They are slower than TTL.

CMOS Diagram

ECL: EMITTER-COUPLED LOGIC

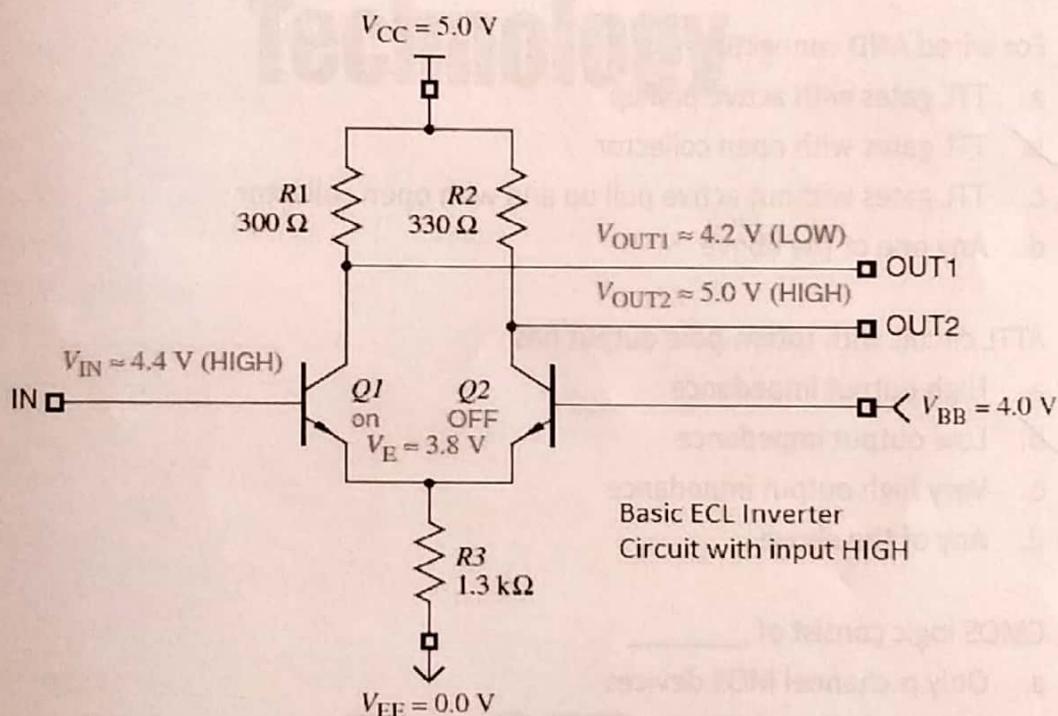
The key to reduce propagation delay in a bipolar logic family is to prevent a gate's transistors from saturating. It is possible to prevent saturation by using a radically different circuit structure, called current-mode logic (CML) or emitter-coupled logic (ECL).

Unlike the other logic families in this chapter, ECL does not produce a large voltage swing between the LOW and HIGH levels but it has a small voltage swing, less than a volt, and it internally switches current between two possible paths, depending on the output state.

Basic ECL Circuit

The basic idea of current-mode logic is illustrated by the inverter/buffer circuit in the figure. This circuit has both an inverting output (OUT1) and a non-inverting output (OUT2).

Two transistors are connected as a differential amplifier with a common emitter resistor.



Question

- 1) As compared to TTL, ECL has
 - a. Lower power dissipation
 - b. Lower propagation delay
 - c. Higher propagation delay
 - d. Higher noise margin
- 2) As compared to TTL CMOS logic has
 - a. Higher speed of operation
 - b. Higher power dissipation
 - c. Smaller physical size
 - d. All the above

- 3) Which logic family has the highest power dissipation per gate
- ECL
 - TTL
 - CMOS
 - PMOS
- 4) Which is the most commonly used logic family
- ECL
 - TTL
 - CMOS
 - PMOS
- 5) For wired AND connection use
- TTL gates with active pull up
 - TTL gates with open collector
 - TTL gates without active pull up and with open collector
 - Any one of the above
- 6) ATTL circuit with totem-pole output has
- High output impedance
 - Low output impedance
 - Very high output impedance
 - Any of the above
- 7) CMOS logic consist of _____
- Only p-channel MOS devices
 - Only N-channel MOS devices
 - MOS devices & capacitors
 - P-channel & n-channel devices

Answer

- 1) b 5) b
- 2) b 6) b
- 3) b 7) d
- 4) d