A binary number system is one of the four types of number system system In computer applications, where binary numbers are represented by 2 digits and that is 0 and 1. The binary numbers are expressed in the base-2 numerical systems.

The octal number system is the base-8 number system and uses the digits 0 to 7.

Fenofex

Non-positional

e

eimal Hexa-Decimal

are also called Alphanumeric

Number

▷ Total number of digit in a number system is called Base. also called Radix (R).

▷ Radix point are called ~~diff~~ with different names in different number system.

e.g. Decimal point, Binary point.. etc.

used in all number syst

al Part.

▷ Decimal has a integer value and

The decimal number system is the base-10 number system and uses digits 0 to 9.

Hexa-decimal is a number system with the base-16. It can be used to represent large numbers with fewer digits and these are 0 to 9 and six alphabetic characters A, B, C, D, E, F.

Fenofex

A_{-1}	A_{-2}	A_{-3}	A_{-4}
2	9	7	6

~~are a power of po~~

Radix po integer

$5 \times 10^2 + 1 \times 10^1 + 2 \times 10^0$

$= 500 + 10 + 2 = 512$

$1 \times 10^{-1} + 2 \times 10^{-2}$

$= 0.1 + 0.02$

$= 0.12$

$\therefore 512.12$

Number System

- Positional
  - Basically 4 type
    - Binary
    - Octal
    - Decimal
    - Hexa-Decimal
- Non-positional

▷ Hexa-Decimal numbers are also called Alphanumeric Number

▷ Total number of digit in a number system is called Base. also called Radix (R).

▷ Radix point are called ~~diff~~ with different names in different number system.

e.g. Decimal point, Binary point.. etc.

▷ Decimal number has a integer value and a Fractional value.

A_3	A_2	A_1	A_0	.	A_{-1}	A_{-2}	A_{-3}	A_{-4}
9	2	0	8	.	2	9	7	6

~~Positional value~~ ~~10 power of pos~~

512.12

$5 \times 10^2 + 1 \times 10^1 + 2 \times 10^0$

$= 500 + 10 + 2 = 512$

$1 \times 10^{-1} + 2 \times 10^{-2}$

$= 0.1 + 0.02$

$= 0.12$

$\therefore$ 512.12

Radix point is a symbol which is used in all number system and it separates integer part from the fractional part.

▷ $122_{(3)} = 1 \times 3^2 + 2 \times 3^1 + 2 \times 3^0 = 9 + 6 + 2 = 17_{(10)}$

▷ $17_{(10)}$ → 3 | 17
3 | 5 − 2 → A_0
3 | 1 − 2 → A_1
0 − 1 → A_2

$= (122)_3$

10 (Base)	2 (Base)	4 (Base)
0	0	0
1	1	1
2	10	2
3	11	3
4	100	10
5	101	11
6	110	12
7	111	13
8	1000	20
9	1001	21
10	1010	22
11	1011	23
12	1100	30
13	1101	31
14	1110	32
15	1111	33
16	10000	100
17	10001	101
18	10010	102
19	10011	103
20	10100	110
21	10101	111

$(.375)_{10} \rightarrow (?)_2 \rightarrow (?)_8$

		.375
		×2
(A-1)	0	.750
		×2
A-2	1	.500
		×2
A-3	1	.500

$\therefore (011)_2$

		.375
		×8
A-1	3	.000

$\therefore (3)_8$

Base	Decimal (Integer)

Decimal (Fractio	
× Base	
.	

Binary → Octal

$\underbrace{010}_{2}\,\underbrace{101}_{5}\,\underbrace{011}_{3}\,.\,\underbrace{110}_{6}\,\underbrace{111}_{7}\,\underbrace{001}_{1}$

Binary → Hexa

$\underbrace{1011}_{11(B)}\,\underbrace{0011}_{3}\,.\,\underbrace{1101}_{13(D)}$

R's complement (Decimal) → 10's

$N_{(R)}$ or $(52520)_{10}$ Here, N = number

R = Radix or Decimal point

n = number of digit

1's complement, $= R^n - N = 10^5 - 52520 = 47480$

0.25 (Here n = 0)

$R^n - N = 10^0 - 0.25 = 0.75$

Hints: First number (right) 1 থেকে বিয়োগ,

পরের সংখ্যা 9 থেকে বাদ,

First number '0' থাকলে '0' হবে।

5.57

$10^1 - 5.57 = 4.43$

R's complement (Binary) 2's

1000000
101100
010100

101100

$2^6 - 101100 = 010100$

Right এ 0 থাকলে 0 হবে,

First '1' থাকলে 1 হবে

পরের সংখ্যা উল্টাবে, $0 \to 1$, $1 \to 0$

(r-1)'s complement (Decimal এর)

$r^n - r^{-m} - N$

n = number of digit of integer part
m = number of digit of fractional part
N = Given number
r = ~~Base~~ Radix or base of a number system

$(52520)_{10}$

$10^5 - 10^0 - 52520 = 47479$

$(0.3267)_{10}$

$10^0 - 10^{-4} - 0.3267 = 0.6732$

(r-1)'s complement এ decimal এর সব Digit 9 (যত বাড়)

Binary (r-1)'s mean 1's complement

$(101100)_2$

$r^n - r^{-m} - N = 2^6 - 2^0 - 101100$

$= 1000000 - 1 - 101100$

$= 10011$

(সবগুলো change হয়ে বিপরীত হয়ে যাবে) অর্থাৎ 1 এবং 0 Digit বিপরীত হবে.

r's

(m−n) Substraction (Decimal)

① First determine the r's complement of n. For that the number of digit of n must be equal to m, if $n < m$.

72532 − 3250

⇒ 72530 − 03250

⇒ 72530 + 96750

= $\boxed{1}$ 69280

End carry.

② Then observe the summation of n and m. If there present a end carry, do nothing. If not present, Again find r's complement of these result and put a negative sign.

3250 − 72532 = 3250 + 27468

= 30718

No end carry (End carry is 0).

∴ 10^5 − 30718 = −69282 Ans.

POS

(m-n) Substraction (Binary)

Rules same,

Ex-1) 1010100 − 1000100

= 1010100 + 111100

= 10010000

End carry

Ex-2) 1000100 − 1010100

= 1000100 + 101100

= 1110000

No carry, then,

2's complement

= −10000

(m-n) substraction using (r-1)'s complement,

Primal: m and n both are positive integer.

Rule: (i) determine the (r-1)'s complement of n. (ii) ~~subtr~~ determine sum of m and (r-1)'s of n.

then observe, If there ~~ex~~ exist End carry (1), simple add 1 in the sum.

otherwise, again determine (r-1)'s complement of (m+n) and put minus sign before the ~~sum~~ result.

# m = 72532
n = 03250
(r-1)'s complement of n is 96749

Now, 72532
+ 96749
$\boxed{1}$69281

End carry +1

69282

# m = 03250
n = 72532
(r-1)'s complement of n, 27467

Now, 03250
27467
30717 carry ze.

again (r-1)'s complement of 30717 is 69282 so the ans is -69282

(m-n) subtraction using (r-1)'s complement

Binary / Rule is same.

m = 1010100
n = 0100100

(r-1)'s complement of n, is 1011011

Now,

$$
\begin{array}{r}
1010100 \\
+\ 1011011 \\
\hline
\boxed{1}\,0101111 \\
\text{End carry} \quad +1 \\
\hline
1100000
\end{array}
$$

m = 1000100
n = 1010100

(r-1)'s complement of n is, 0101011

Now,

$$
\begin{array}{r}
1000100 \\
+\ 0101011 \\
\hline
1101111
\end{array}
$$

No carry, then,

-10000

Comparison:

In the respect of complement, (r-1)'s complement is useful to use.

In the respect of substraction, r's complement is useful. Because in (r-1)'s complement, sometime the result is both positive and negative zero.

Binary logic: It is necessary for processing and manipulate Binary information.

Chapter — 02

Boolean Algebra or symbolic logic: Boolean algebra may be defined with a set of elements, a set of operators, and a number of unproved axioms or postulates.

* It defines rules for manipulating symbolic binary logic expressions.
  * a symbolic binary logic expression consists of binary variables and the operators AND, OR, and NOT (e.g. $A+B.C$)
* The posible values for any Boolean expression ~~consists of~~ can be tabulated in a truth table.
* Boolean algebra can defined circuit for expression by combining gates.

Basic operations are, OR, AND and NOT.

Algebras: mathematical system consisting of
- set of elements
- set of operations
- Axioms or postulates

It is important because it defines rules of 'calculations'.

Example: set of elements = $N = \{1, 2, 3, \ldots\}$

operation: $+, -, *, /, \%$

axioms: Associativity, Distributivity, closure, etc.

N.B: Operatory with two inputs are called ~~ope~~ binary operation, does not mean that restricted to binary numbers.

Common Axioms on Postulates:

1. Closure: A set S is closed with respect to a binary operation if, for every pair of elements of S, the binary operation specifies a rule for obtaining an element of S. For example, the set of natural numbers is closed w.r.t. binary operation (+) but not closed w.r.t. binary operation (−)

2. Associative law: A binary operation $*$ on a set S is said to be associative if

$$(x * y) * z = x * (y * z) \text{ for all } x, y, z \in S.$$

~~3. Commuta~~

3. <u>Commutative law:</u> A binary operation $*$ on a set S is said to be commutative if $x * y = y * x$ for all $x, y \in S$.

4. <u>Identity element:</u> set S is said to have an identity element with respect to binary operation $*$ on S if there exist an element e member of S such that

$$e * x = x * e = x \text{ for every } x \text{ member of } S.$$

5. <u>Inverse:</u> A set S having the identity element e with respect to a binary operation $*$ is said to have an inverse whenever, for every x member of S. There exist an element y ~~such that~~, member of S such that $x * y = e$

6. <u>Distributive law:</u> If '$*$' and '$\cdot$' are two binary operations on a set S, $*$ is said to be distributive over '$\cdot$' whenever

$$x * (y \cdot z) = (x * y) \cdot (x * z)$$

N.B. Identity element depend on operation.

The operator and postulates have the following meaning.

→ The binary operator + defines addition.

→ The additive identity is 0.

→ The additive inverse defines substraction.

→ The binary operator . defines multiplication

→ The multiplicative identity is 1.

→ The multiplicative inverse of $a = 1/a$ defines division.

That is $a \cdot 1/a = 1$.

→ The only distributive law is applicable is that of . over +: $a \cdot (b+c) = (a \cdot b) + (a \cdot c)$

Axiomatic Definition of boolean algebra

Boolean algebra is an algebraic structure defined on a set of elements B together with two binary operators '+' and '.' provided the following postulates are satisfied.

1. closure w.r.t '+' and closure w.r.t. '.'
2. Identity element w.r.t '+' is 0, and w.r.t '.' is 1
   $x+0 = x$, $x \cdot 1 = 1 \cdot x = x$
3. Communicative w.r.t '.' $x \cdot y = y \cdot x$ same as, $x+y = y+x$
4. Distributive over '+' $x \cdot (y+z) = (x \cdot y) + (y \cdot z)$
   Same as '.', $x + (y \cdot z) = (x+y) \cdot (x+z)$

5. For every element $x \in B$, there exist an element $x' \in B$ (called the complement of B).

Such that, $x + x' = 1$ $x \cdot x' = 0$

6. There exist at least two elements $x, y \in B$ such that $x \neq y$

Two valued Boolean Algebra

A two valued boolean algebra is defined on a set of two elements, $B = \{0, 1\}$. with rules for the two binary operators '+' and '·' are shown in the table.

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

x	x'
0	1
1	0

The rules are exactly same as the operations AND, OR and NOT respectively. We must now show that the ~~Hin~~ Huntington postulates are valid for the set $B = \{0, 1\}$ and the two binary operators defined above.

1. Closure is obvious from the tables since the result of each operation is either 1 or 0 and $1, 0 \in B$.

2. from the tables (a) $0+0=0$ $0+1=1+0=1$
(b) $1 \cdot 1 = 1$ $1 \cdot 0 = 0 \cdot 1 = 0$

we establishes the two identity elements 0 for '+' and 1 for '·' as defined by postulate 2.

(2)

3. The communicative laws are obvious from the summary of the binary operators tables.

4. (a) For each combination, we derive $x.(y+z)$ and show that the value is the same as $(x.y)+(x.z)$, from the table below.

x	y	z	$y+z$	$x.(y+z)$	$x.y$	$x.z$	$(x.y)+(x.z)$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

(b) is similar to above table.

(5) From the complement table we easily shown that.

(a) $x+x' = 1$, $0+0' = 0+1 = 1$, $1+1' = 1+0 = 1$

(b) $x.x' = 0$ since, $0.0' = 0.1 = 1$ $1.1' = 1.0 = 0$

(6) Postulate 6 is satisfied because the two-valued Boolean algebra has two distinct elements 1 and 0 with $1 \neq 0$.

Basic Theorem of boolean Algebra:

Duality:

6 Theorem's of boolean algebra and 4 of it's postulates:

Postulate-2 (a) $x+0=x$ (b) $x.1=x$

P-5 (a) $x+x'=x$ (b) $x.x'=0$

Theorem-1 (a) $x+x=x$ (b) $x.x=x$

T-2 (a) $x+1=x$ (b) $x.0=0$

T-3, involution $(x')'=x$

P-3 communicative, $x+y=y+x$, $xy=yx$

T-4 associative $x+(y+z)=(x+y)+z$

P-4 distributive $x(y+z)=xy+yz$, $x+yz=(x+y)(x+z)$

T-5 DeMorgan (a) $(x+y)'=x'y'$ (b) $(xy)'=x'+y'$

T-6 absorption (a) $x+xy=x$, $x(1+y)=x$ | (b) $x(x+y)=x$, $x+xy=x(x+y)$

Theorem-1 (a)

$$x+x=(x+x).1$$
$$=(x+x).(x+x')$$
$$=x+xx'$$
$$=x+0$$
$$=x$$

1(b)

$$x.x=xx+0$$
$$=xx+xx'$$
$$=x(x+x')$$
$$=x.1$$
$$=x$$

Theorem-2(a)

$$x+1=1.(x+1)$$
$$=(x+x')(x+1)$$
$$=x+x'.1$$
$$=x+x'$$
$$=1$$

2(b) $x.0=0$ by duality

Theorem-6(a)

$$x+xy=x.1+xy$$
$$=x(1+y)$$
$$=x.1$$
$$=x$$

6(b) $x(x+y)=x$ by duality

প্রথমে NOT → AND → OR এর কাজ করতে হবে.

Theorem-3: $(x')' = x$. From postulate 5, we have $x+x'=1$ and $x.x'=0$, which defines the complement of x. The complement of x' is x and is also $(x')'$. Therefore, since the complement is unique, we have that $(x')' = x$.

Boolean Functions:

(1) Truth table for $F_3 = x'y'z + x'yz + xy'$

also, $F_4 = xy' + x'z$

x	y	z	$x'y'$	$x'z$	$x'y'z$	$x'yz$	F_3	F_4
0	0	0	0	0	0	0	0	0
0	0	1	0	1	1	0	1	1
0	1	0	0	0	0	0	0	0
0	1	1	0	1	0	1	1	1
1	0	0	0	0	0	0	1	1
1	0	1	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0

Complement of a function:

$$F_1 = (x'yz' + x'y'z)'$$
$$= (x'yz')'(x'y'z)'$$
$$= (x+y'+z)(x+y+z')$$

$$F_2 = (x(y'z' + yz))'$$
$$= (xy'z' + xyz)'$$
$$= (x'+y+z)(x'+y+z')$$

$$= (x(y \odot z))'$$
$$= x' + (y \odot z)'$$
$$= x' + x \oplus z$$

* Exclusive OR এর Input অযুগ্ম হলে বিজোড় সংখ্যক '1' এর জন্য Output এ 1 আসবে।

Algebra:

Boolean Algebra: Boolean algebra is a branch of mathmatics that deals with operations on logical values with binary ~~values~~ variables. The boolean variables are represented as binary numbers to represent truth: 1 = true and 0 = false.

Truth table: A truth table is a tabular representation of all the combinations of values for inputs and their corresponding outputs. Truth tables are usually used for logical problems as in boolean algebra and electronics circuit.

Simplify the following Boolean functions to a minimum number of literals:

1. $x + x'y = (x+x')(x+y) = 1 \cdot (x+y) = x+y$

2. $x(x'+y) = x'x + xy = 0 + xy = xy$

4. $xy + x'z + yz = xy + x'z + yz(x+x') = xy + x'z + xyz + x'yz$

$= xy(1+z) + x'z(1+y) = xy + x'z$

Canonical AND standard forms:

			Minterms		Maxterms	
x	y	z	Term	Designation	Term	Designation
0	0	0	$x'y'z'$	m_0	$x+y+z$	M_0
0	0	1	$x'y'z$	m_1	$x+y+z'$	M_1
0	1	0	$x'yz'$	m_2	$x+y'+z$	M_2
0	1	1	$x'yz$	m_3	$x+y'+z'$	M_3
1	0	0	$xy'z'$	m_4	$x'+y+z$	M_4
1	0	1	$xy'z$	m_5	$x'+y+z'$	M_5
1	1	0	xyz'	m_6	$x'+y'+z$	M_6
1	1	1	xyz	m_7	$x'+y'+z'$	M_7

Minterm: In boolean algebra, Boolean function can be expressed as canonical Disjunctive Normal form known as minterm.

Maxterm: In boolean algebra, boolean function can be expressed as canonical conjunctive Normal form known as maxterm.

$f_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$

canonical form

$f' = x'y'z' + x'y'z' + x'yz + xy'z + xyz'$
$= m_0 + m_2 + m_3 + m_5 + m_6$

$\therefore f = (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z')(x'+y'+z)$

$= M_0 \cdot M_2 M_3 M_5 M_6$

<u>Ex-2.4)</u> $F = A + B'C$. Express sum of minterms.

The first A missing two variables

$A = A(B+B') = AB + AB'$

still missing one variable

$AB + AB' = AB(C+C') + AB'(C+C')$

$= ABC + ABC' + AB'C + AB'C'$

$B'C$ ~~still~~ missing one variable.

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

$= AB'C + A'B'C$

$\therefore F = ABC + ABC' + AB'C + AB'C' + A'B'C$

$\cancel{= m_0 + m}$

$\therefore F = m_7 + m_6 + m_5 + m_4 + m_1$

The following short notation is

$F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$

Product of maxterms: Each of the 2^{2^n} functions of n binary variables can ~~also~~ be also expressed as a product of maxterm.

Ex-2.5) Express the boolean function $F = xy + x'z$

Soln

$\cancel{F = xy(z+z')}$

$$F = xy + x'z = (xy + x')(xy + z)$$

$$= (x + x')(x' + y)(x + z)(y + z)$$

$$= (x' + y)(x + z)(y + z)$$

The function has three variables: x, y and z. Each OR term is missing one variable.

Therefore, $x' + y = x' + y + zz'$

$$= (x' + y + z)(x' + y + z')$$

$$x + z = x + z + y \cdot y'$$

$$= (x + y + z)(x + y' + z)$$

$$y + z = x \cdot x' + y + z$$

$$= (x + y + z)(x' + y + z)$$

$$\therefore F = (x' + y + z)(x' + y + z')(x + y + z)(x + y' + z)$$

$$\cancel{(x + y + z)(x' + y + z)}$$

$$= M_3 M_5 M_0 M_2 = M_0 M_2 M_4 M_5$$

A convenient way to express the function is

$$F(x, y, z) = \Pi(0, 2, 4, 5)$$

The product symbol, Π denotes the ANDing of maxterms, the numbers are the ~~may~~ maxterms of the function.

And the summation symbol, Σ stands for ORing ~~the~~ of terms; the numbers following it are the minterm of the function.

Conversion between ~~Cano~~ Canonical Forms:

$$F'(A,B,C) = \Sigma(0,2,3) = m_0 + m_2 + m_3$$

Now, if we take the complement of F' De Morgan's theorem, we obtain F in a different form:

$$F = (m_0 + m_1 + m_3)' = m_0' m_2' \cdot m_3' = M_0 M_2 M_3$$
$$= \Pi(0,2,3).$$

$\therefore$ $m_j' = M_j$

Standard Forms: A Boolean variable can be expressed in either true form or complemented form. In standard form boolean function will contain all the variables in either true form or complemented form while in canonical numbers of variables depends on the output of SOP or POS.

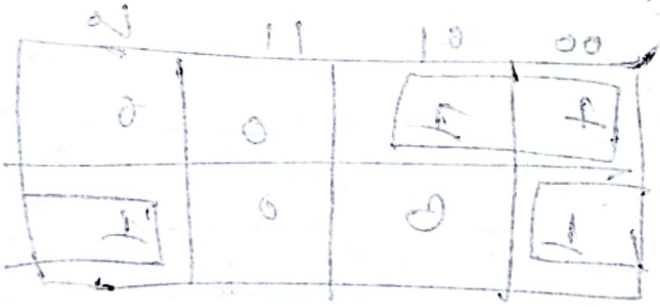
Binary operation এর operand নিয়ে কাজ করে।

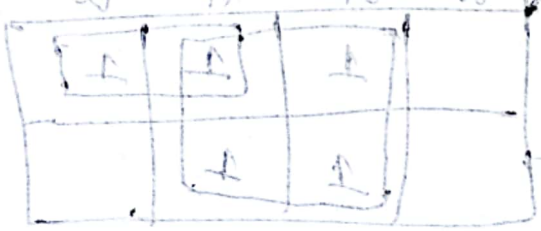
Sum of products: A canonical sum of products is a boolean expression that entirely consists of minterms. It containing AND terms, called product terms of one or more literals each. The sum denotes the ORing of these terms.

$$F_1 = y' + xy + x'yz'$$

Product of Sum: The product of sum form is a form in which products of different sum terms of inputs are taken. It containing OR terms, called sum terms. Each term may have any literals. The product denotes ~~ANDin~~ ANDing the terms.

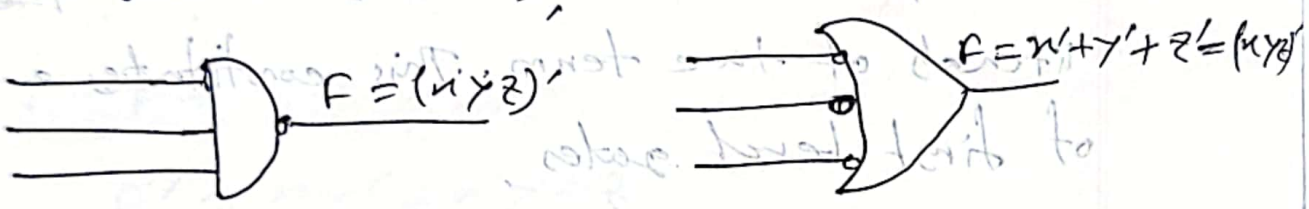
$$F_2 = x(y'+z)(x'+y+z'+w)$$





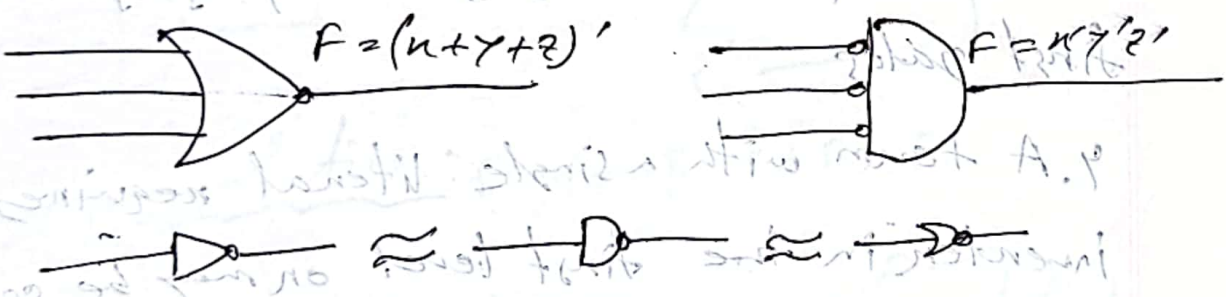
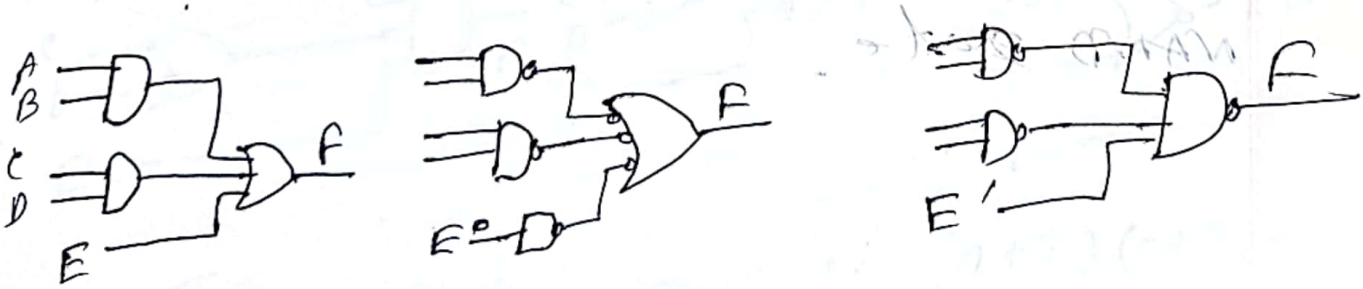
NAND AND NOR Implementation: Digital circuits are more frequently constructed with NAND or NOR gates than with AND and OR gates. NAND and NOR gates are easier to fabricate with electronic components and are the basic gates used in all IC digital logic families.

symbols of NAND gates:



Two graphic symbols for NAND gate.

symbols of NOR gates:



$F = AB + CD + E$