Rajmani Kumar,
Lecturer, Dept. of BCA
S.U.College, Hilsa (Nalanda)
Patliputra University, Patna

BCA-2nd Year **Paper-III**

Logic Gates

The graphic symbols and truth tables of the gates of the eight different operations are shown:

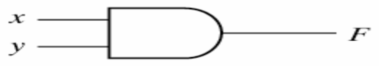
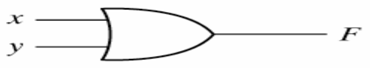
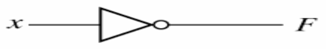
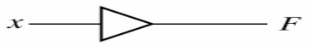
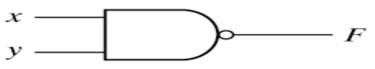
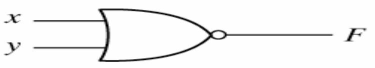
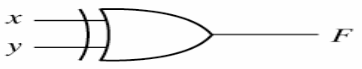
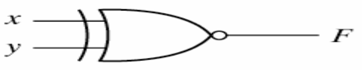
Name	Graphic symbol	Algebraic function	Truth table
AND		$F = xy$	x y F 0 0 0 0 1 0 1 0 0 1 1 1
OR		$F = x + y$	x y F 0 0 0 0 1 1 1 0 1 1 1 1
Inverter		$F = x'$	x F 0 1 1 0
Buffer		$F = x$	x F 0 0 1 1
NAND		$F = (xy)'$	x y F 0 0 1 0 1 1 1 0 1 1 1 0
NOR		$F = (x + y)'$	x y F 0 0 1 0 1 0 1 0 0 1 1 0
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	x y F 0 0 0 0 1 1 1 0 1 1 1 0
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	x y F 0 0 1 0 1 0 1 0 0 1 1 1

Fig. 2-5 Digital logic gates

Extension to multiple inputs

- In Fig.2-5, except for the inverter and buffer, can be extended to have more than two inputs.
- The AND and OR operations possess two properties: commutative and associative.

x + y = y + x (commutative)

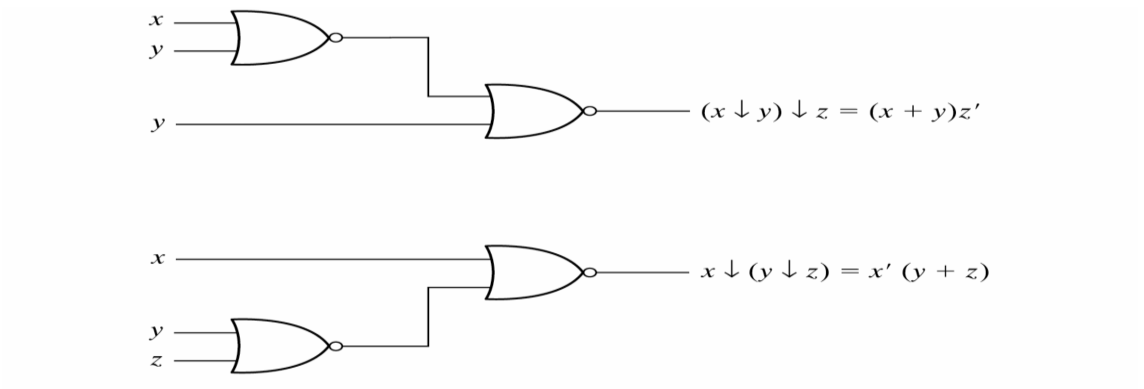
(x + y) + z = x + (y + z) (associative)

Non-associativity of the NOR operator

The NAND and NOR functions are commutative, not associative .

$(x{\downarrow}y){\downarrow}z = [(x + y)' + z]' = (x + y)\, z' = (x + y)z'$

$x{\downarrow}(y{\downarrow}z) = [x + (y + z)']' = x'(y + z) = x'(y + z)$



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

Fig. 2-6 Demonstrating the nonassociativity of the NOR operator; $(x \downarrow y) \downarrow z \neq x \downarrow (y \downarrow z)$

Cascade of NAND gates

In writing cascaded NOR and NAND operations, one must use the correct parentheses to signify the proper sequence of the gates.

F = [(ABC)'(DE)']'= ABC + DE obtain from DeMorgan's theorem.

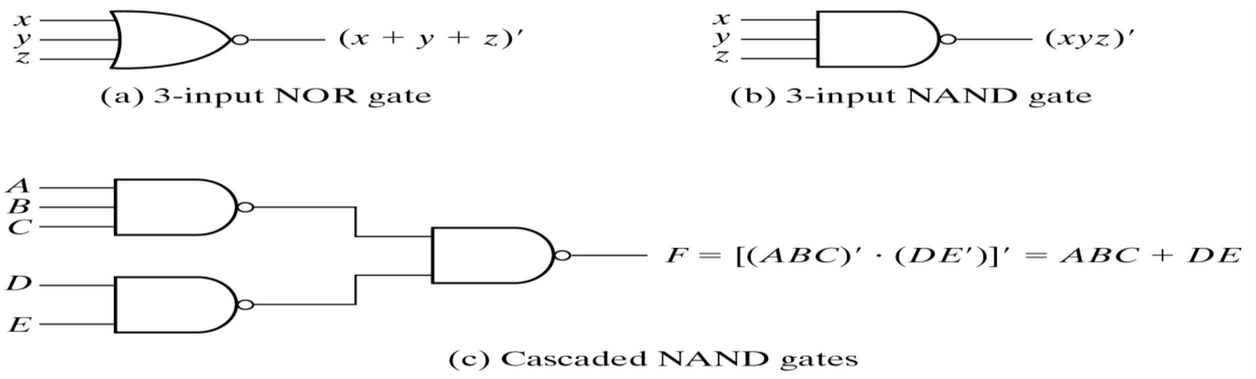
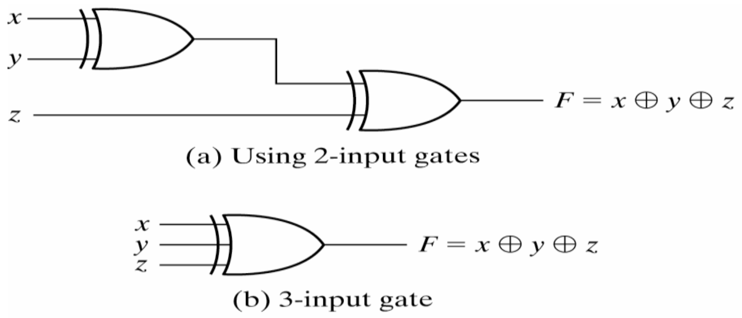


Fig. 2-7 Multiple-input and cascated NOR and NAND gates

XOR gate property

The XOR and equivalence gates are both commutative and associative and can be extended to more than two inputs.

- The XOR is an odd function.
- The three inputs XOR is normally implemented by cascading 2-input gates.



x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(c) Truth table

Fig. 2-8 3-input exclusive-OR gate

Positive and negative logic

We assign to the relative amplitudes of the two signal levels as the high-level and low-level.

1. High-level (H): represent logic-1 as a positive logic system.
2. Low-level (L): represent logic-0 as a negative logic system.

It is up to the user to decide on a positive or negative logic polarity between some certain potential.

Example

The electronic shown in Fig.2-10(b), truth table listed in (a).

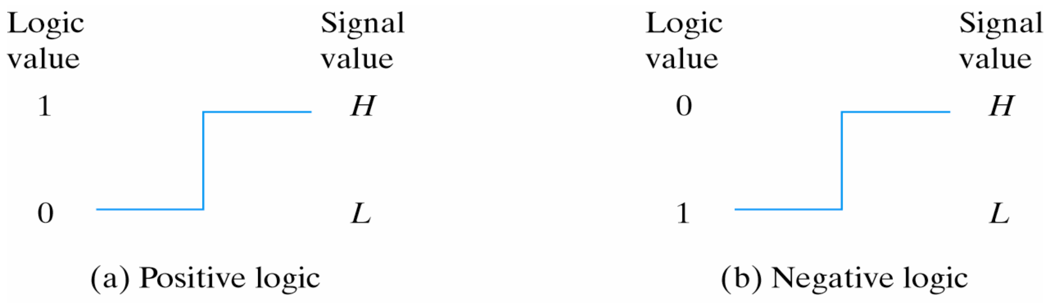
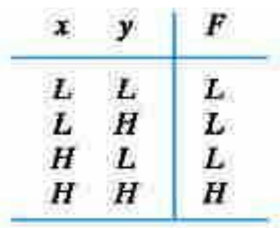
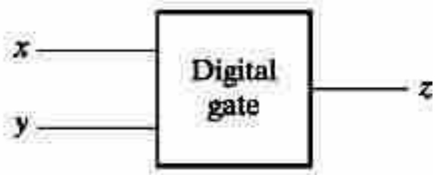


Fig. 2-9 signal assignment and logic polarity



x	y	F
L	L	L
L	H	L
H	L	L
H	H	H

(a) Truth table with *H* and *L*



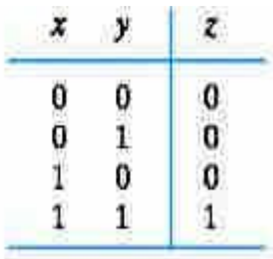
(b) Gate block diagram

Fig. 2-10 Demonstration of positive and negative logic

It specifies the physical behavior of the gate when H is 3 volts and L is 0 volts.

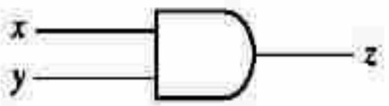
Positive logic

- The truth table of Fig.2-10(c) assumes positive logicassignment , with H=1 and L=0.
- It is the same as the one for the AND operation



x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

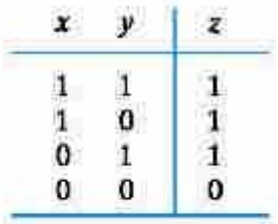
(c) Truth table for positive logic



(d) Positive logic AND gate

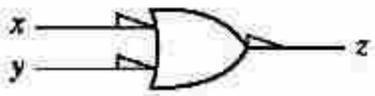
Negative logic

- The table represents the OR operation even though the entries are reversed.
- The conversion from positive logic to negative logic, and vice versa, is essentially an operation that changes 1's to 0's and 0's to 1's(dual) in both the inputs and the outputs of a gate.



x	y	z
1	1	1
1	0	1
0	1	1
0	0	0

(e) Truth table for negative logic



(f) Negative logic OR gate

Fig. 2-10 Demonstration of positive and negative logic

Integrated circuits

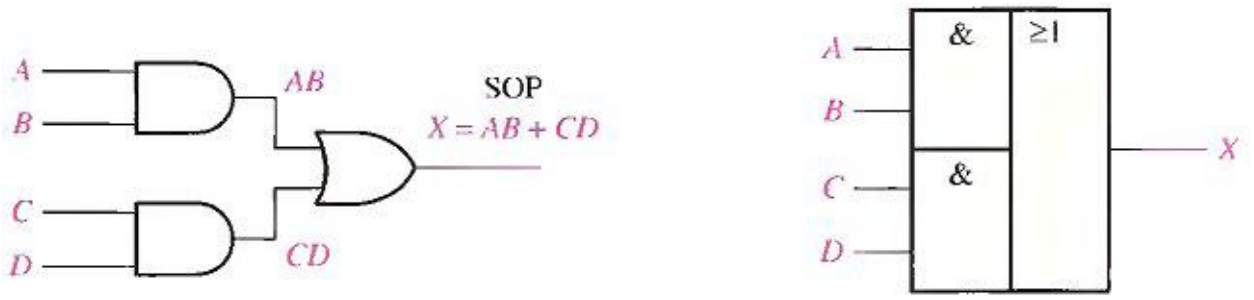
- An integrated circuit(IC) is a silicon semiconductor crystal, called chip, containing the electronic components for constructing digital gates.

COMBINATIONAL LOGIC ANALISIS

1- AND-OR Logic

Fig.(6-1)(a) shows an AND-OR circuit consisting of two 2-input AND gates and one 2-input OR gate; Fig.(6-1)(b) is the ANSI standard rectangular outline symbol. The Boolean expressions for the AND gate outputs and the resulting SOP expression for the output X are shown in the diagram. In general, all AND-OR circuit can have any number of AND gates each with any number of inputs.

The truth table for a 4-input AND-OR logic circuit is shown in Table 6-1. The intermediate AND gate outputs (AB and CD columns) are also shown in the table.



(a) Logic diagram (b) ANSI standard rectangular outline symbol.

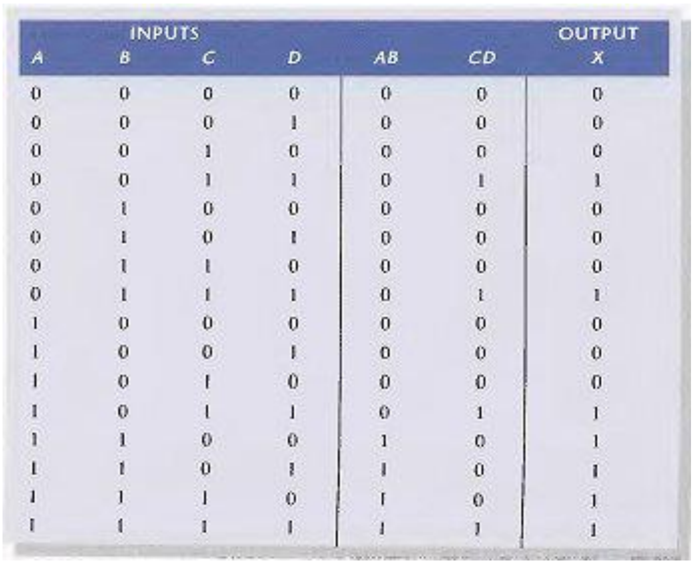
Fig.(6-1)

For a 4-input AND-OR logic circuit, the output X is HIGH (1) if both input A and input B are HIGH (1) or both input C and input D are HIGH (1).

2-AND-OR-Invert Logic

When the output of an AND-OR circuit is complemented (inverted), it results inan AND-OR-Invert circuit. Recall that AND-OR logic directly implements SOP expressions. POS expressions can be implemented with AND-OR-Invert logic. This is illustrated as follows, starting with a POS expression and developing the corresponding AND-OR-Invert expression.

Table 6-1



INPUTS						OUTPUT
A	B	C	D	AB	CD	X
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	1	1
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	1	1
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	0	1	1
1	1	0	0	1	0	1
1	1	0	1	1	0	1
1	1	1	0	1	0	1
1	1	1	1	1	1	1

$$X = (\overline{A} + \overline{B})(\overline{C} + \overline{D}) = (\overline{AB})(\overline{CD}) = \overline{\overline{(\overline{AB})(\overline{CD})}} = \overline{\overline{\overline{AB}} + \overline{\overline{CD}}} = \overline{AB + CD}$$

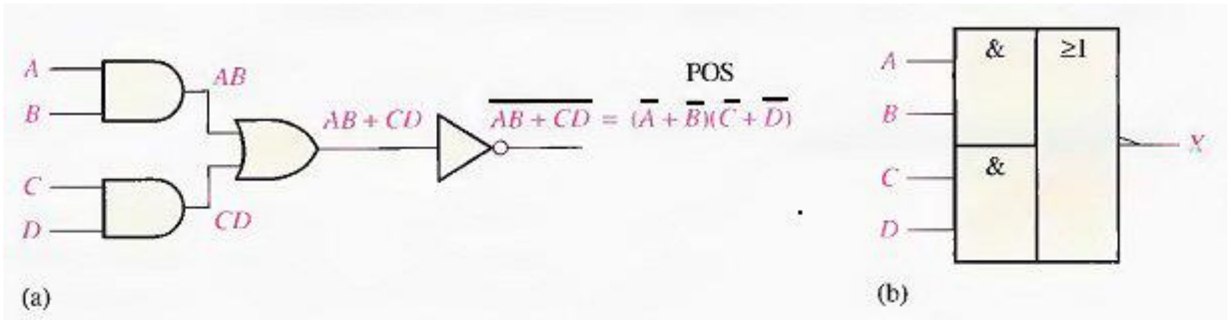

A
B
C
D
AB
CD
AB + CD
$\overline{AB + CD} = (\overline{A} + \overline{B})(\overline{C} + \overline{D})$
POS
(a)
&
≥1
&
X
(b)

Fig.(6-2)

For a 4-input AND-OR-Invert logic circuit, the output X is LOW (0) if both input A and input B are HIGH (1) or both input C and input D are HIGH (1).

<u>**3-Exclusive-OR logic**</u>

The exclusive-OR gate was introduced before. Although, because of its importance, this circuit is considered a type of logic gate with its own unique symbol it is actually a combination of two AND gates, one OR gate, and two inverters, as shown in Fig.(6-3)(a). The two standard logic symbols are shown in parts (b) and (c).

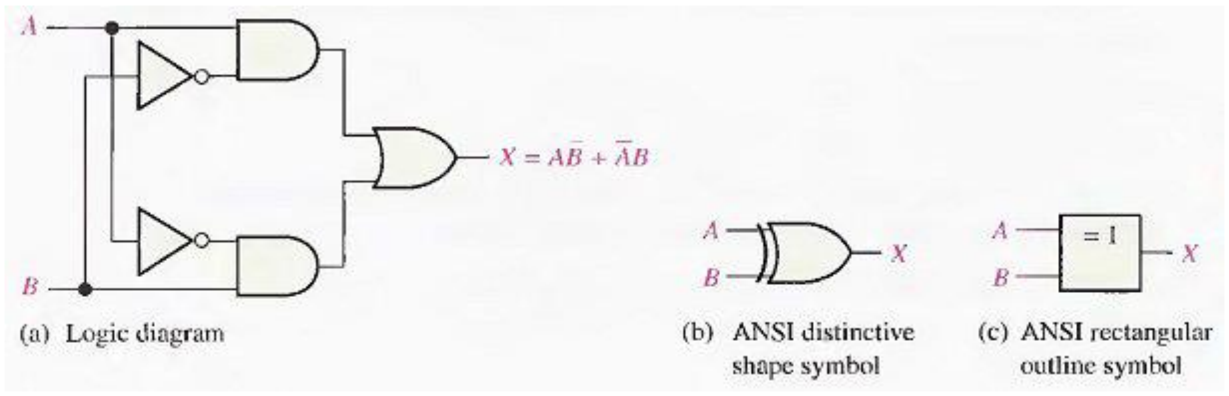

A
B
$X = A\overline{B} + \overline{A}B$
(a) Logic diagram
A
B
X
(b) ANSI distinctive shape symbol
A
B
= 1
X
(c) ANSI rectangular outline symbol

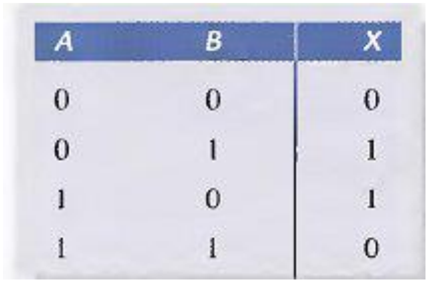
Fig.(6-3)

The output expression for the circuit in Fig.(6-3) is

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

Can be written as $X = A \oplus B$

Table 6-2 Truth table for an exclusive-OR.



A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

4- Exclusive-NOR Logic

As you know, the complement of the exclusive-OR function is the exclusive-NOR, whichis derived as follows:

$$X = \overline{A\overline{B} + \overline{A}B} = (\overline{A\overline{B}})\,(\overline{\overline{A}B}) = (\overline{A} + B)(A + \overline{B}) = \overline{A}\,\overline{B} + AB$$

Notice that the output X is HIGH only when the two inputs, A and B, are at the same level.

The exclusive-NOR can be implemented by simply inverting the output of an exclusive- OR, as shown in Fig(6-4)(a), or by directly implementing then expression AB + AB, as shown in part (b).

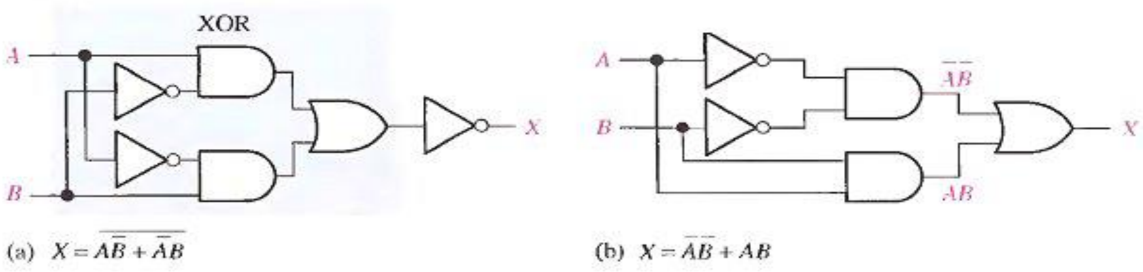

XOR
A
B
X
(a) $X = \overline{A\overline{B} + \overline{A}B}$
A
B
$\overline{A}\,\overline{B}$
AB
X
(b) $X = \overline{A}\,\overline{B} + AB$

Fig.(6-4)

Example

Develop a logic circuit with four input variables that will only produce a 1 output when exactly three input variables are 1s. Fig.(6-5) shows the circuit.

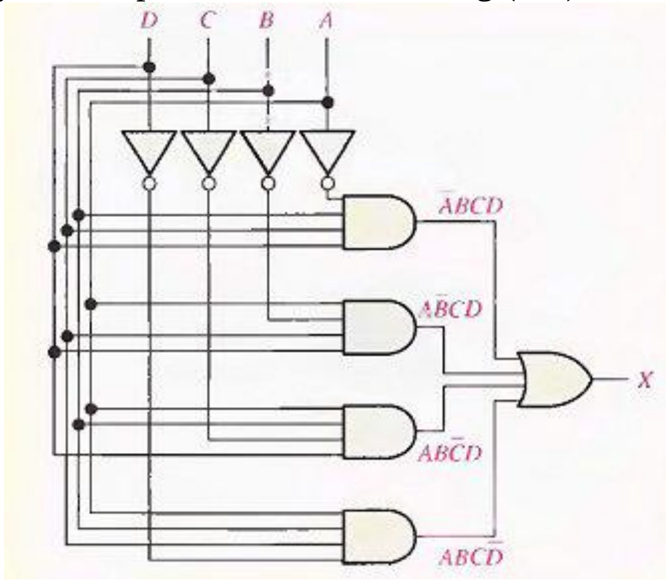


Fig.(6-5)

Example

Reduce the combinational logic circuit in Fig.(6-6) to a minimum form.

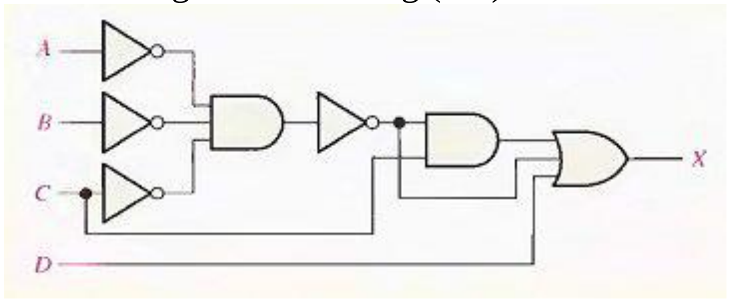


Fig.(6-6)

Solution

The expression for the output of the circuit is

$$X = (\bar{A}\,\bar{B}\,\bar{C})\,C + \overline{\bar{A}\bar{B}\bar{C}} + D$$

Applying DeMorgan's theorem and Boolean algebra,

$$\begin{aligned} X &= (\bar{\bar{A}} + \bar{\bar{B}} + \bar{\bar{C}})C + \bar{\bar{A}} + \bar{\bar{B}} + \bar{\bar{C}} + D \\ &= AC + BC + CC + A + B + C + D \\ &= AC + BC + C + A + B + \cancel{C} + D \\ &= C(A + B + 1) + A + B + D \\ X &= A + B + C + D \end{aligned}$$

The simplified circuit is a 4-input OR gate as shown in Fig.(6-7).

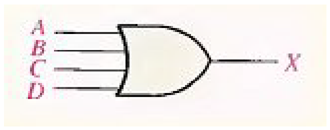


Fig.(6-7)

THE UNIVERSAL PROPERTY OF NAND AND NOR GATES

1.The NAND Gate as a Universal Logic Element

The NAND gate is a universal gate because it can be used to produce the NOT, the AND, the OR, and the NOR functions. An inverter can be made from a NAND gate by connecting all of the inputs together and creating, in effect, a single input, as shown in Fig.(6-8)(a) for a 2-input gate. An AND function can be generated by the use of NAND gates alone, as shown in Fig.(6-8)(b). An OR function can be produced with only NAND gates, as illustrated in part (c). Finally. a NOR function is produced as shown in part (d).

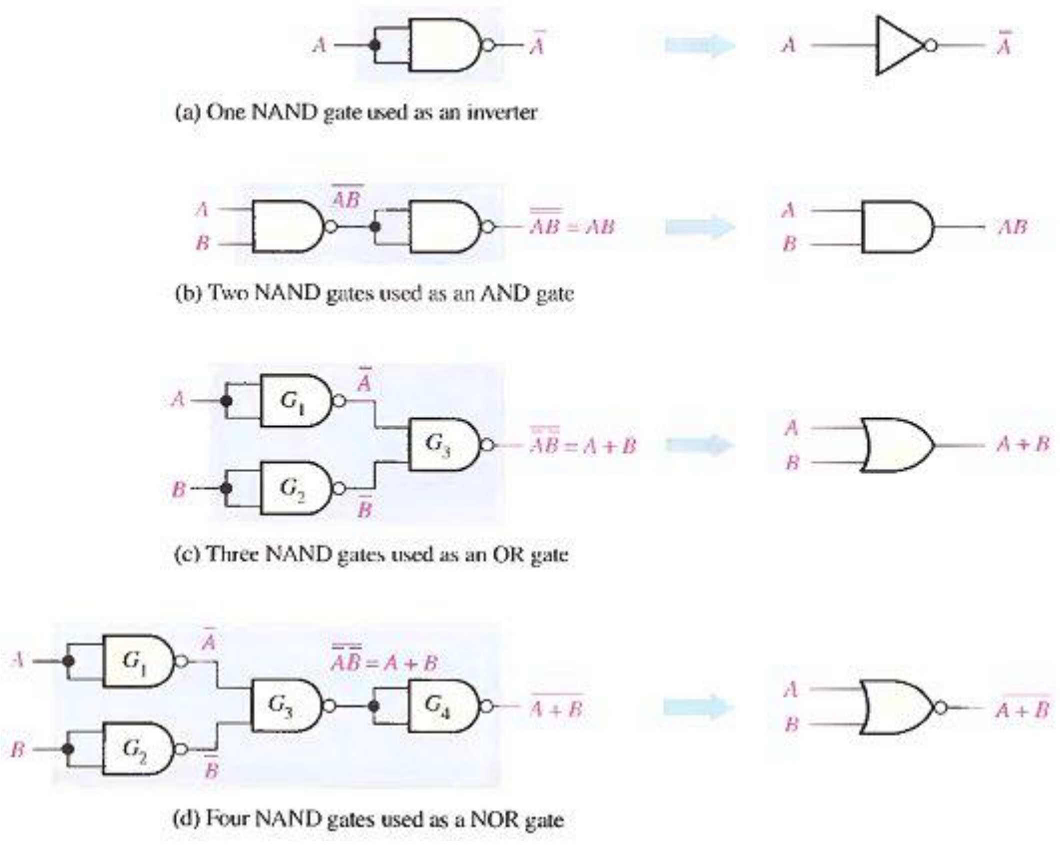

A
$\overline{A}$
(a) One NAND gate used as an inverter
$\overline{AB}$
$\overline{\overline{AB}} = AB$
AB
(b) Two NAND gates used as an AND gate
G_1
G_2
G_3
$\overline{B}$
$\overline{\overline{A}\,\overline{B}} = A + B$
$A + B$
(c) Three NAND gates used as an OR gate
G_4
$\overline{\overline{A}\,\overline{B}} = A + B$
$\overline{A + B}$
(d) Four NAND gates used as a NOR gate

Fig.(6-9)

2- The NOR Gate as a Universal Logic Element

Like the NAND gate, the NOR gate can be used to produce the NOT, AND. ORand NAND functions. A NOT circuit, or inverter, can be made from a NOR gate by connecting all of the inputs together to effectively create a single input, as shown in Fig.(6-10)(a) with a 2-input example. Also, an OR gate can be produced from NOR gates, as illustrated in Fig.(6-10)(b). An AND gate can be constructed by the use of NOR gates, as shown in Fig.(6-10)(c). In this case the NOR gates G 1 and G 2 are used as inverters, and the final output is derived by the use of DeMorgan's theorem as follows:

$$X=\overline{\overline{A}+\overline{B}}=AB$$

Fig.(6-10)(d) shows how NOR gates are used t0 form a NAND function.

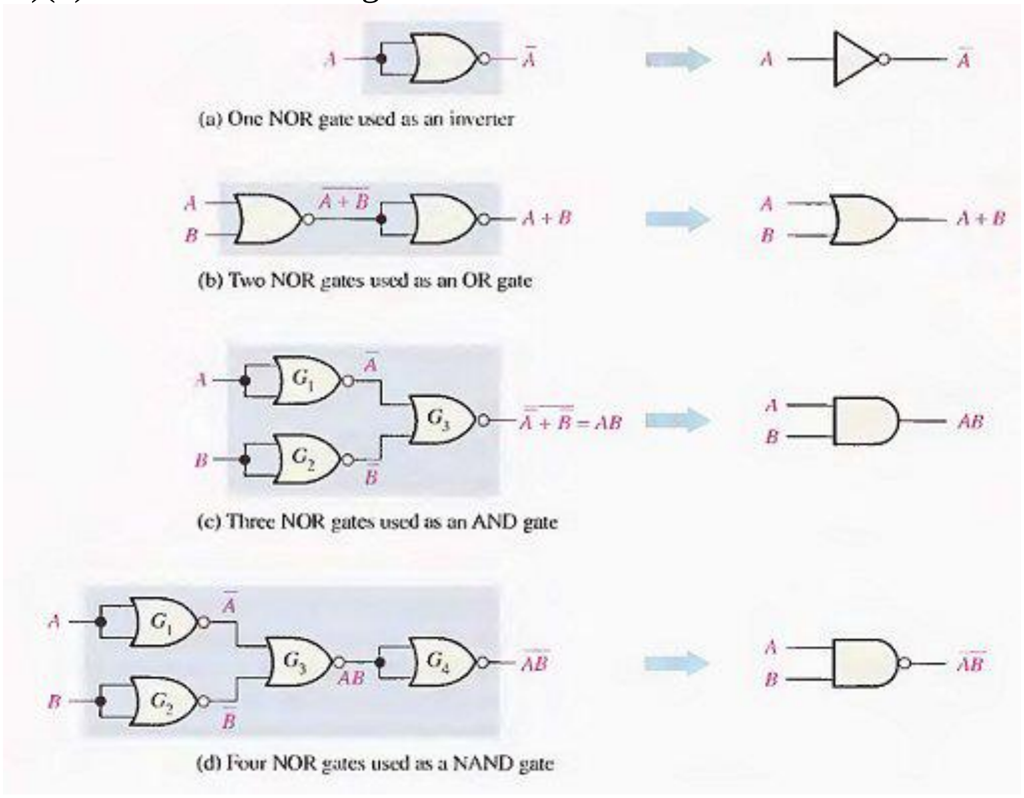

A
$\overline{A}$
(a) One NOR gate used as an inverter
A
B
$\overline{A+B}$
$A+B$
(b) Two NOR gates used as an OR gate
G_1
G_2
G_3
$\overline{A}$
$\overline{B}$
$\overline{\overline{A}+\overline{B}}=AB$
AB
(c) Three NOR gates used as an AND gate
G_4
AB
$\overline{AB}$
(d) Four NOR gates used as a NAND gate

Fig.(6-10)