Properties of a Boolean Algebra

1. Operations are *commutative*.

$$A \bullet B = B \bullet A$$
$$A + B = B + A$$

2. Operations are *associative*.

$$(A \bullet B) \bullet C = A \bullet (B \bullet C)$$
$$(A + B) + C = A + (B + C)$$

3. Each operation is *distributive* over the other.

$$A \bullet (B + C) = (A \bullet B) + (A \bullet C)$$
$$A + (B \bullet C) = (A + B) \bullet (A + C)$$

4. There exists an *identity* element for each operation.

$+$ Identity $= 0$ $\quad A + 0 = A$

$\bullet$ Identity $= 1$ $\quad A \bullet 1 = A$

5. There exists a *complement* for each element.

Complement of $A = \overline{A}$

Complement of $1 = 0$

Complement of $0 = 1$

6. There exists an *inverse* for each operation.

$$A \bullet \overline{A} = 0$$
$$A + \overline{A} = 1$$

7. Each element is *idempotent*.

$$A \bullet A = A$$
$$A + A = A$$

8. The *absorption* property holds for each element.

$$A \bullet (A + B) = A$$
$$A + (A \bullet B) = A$$

Boolean Algebra

A AND B	A * B	
A OR B	A + B	
NOT A	$\overline{A}$	
1's and 0's	True and False	Hi and Lo

Boolean Algebra

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

Boolean Algebra

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

DeMorgan's Theorem

$\overline{A \text{ AND } B} = \overline{A} \text{ OR } \overline{B}$

$\overline{A \text{ OR } B} = \overline{A} \text{ AND } \overline{B}$

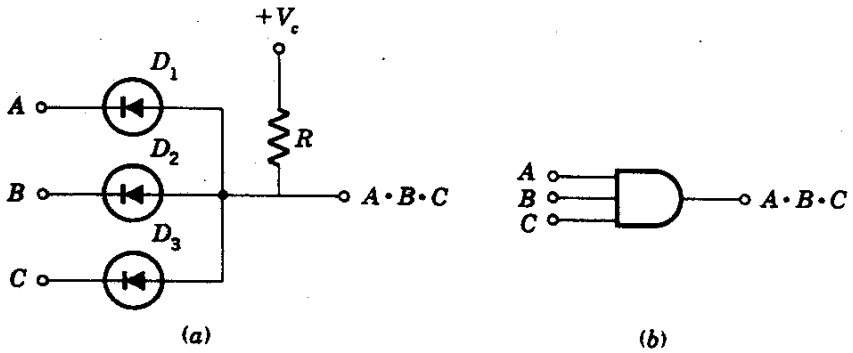


Figure 9-2 (*a*) Diode AND gate and (*b*) circuit symbol.

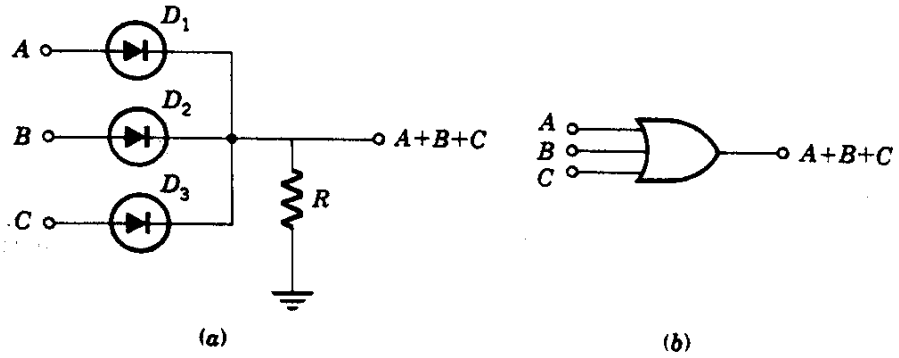


Figure 9-3 (*a*) Diode OR gate and (*b*) circuit symbol.

TRANSISTOR SWITCH

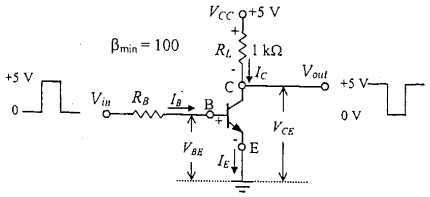


Figure 3-32: The BJT Inverter

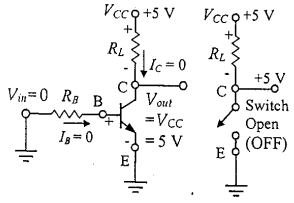


Figure 3-32(a): Transistor OFF

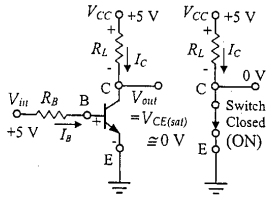
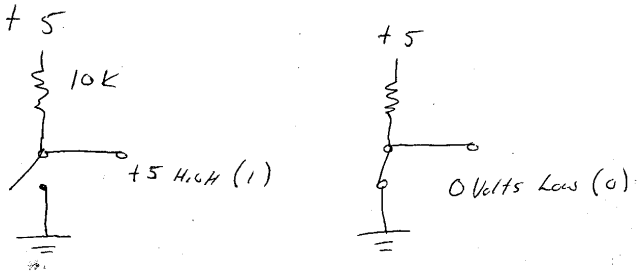


Figure 3-32(b): Transistor ON

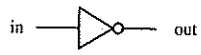


12.2 Logic Gates

Logic gates are the building blocks of digital electronics. The fundamental logic gates include the INVERT (NOT), AND, NAND, OR, NOR, exclusive OR (XOR), and exclusive NOR (XNOR) gates. Each of these gates performs a different logical operation. Figure 12.10 provides a description of what each logic gate does and gives a switch and transistor analogy for each gate.

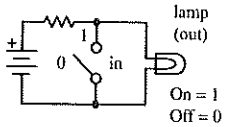
FIGURE 12.10

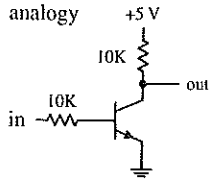
INVERT (NOT)



in	out
0	1
1	0

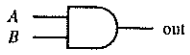
0 = LOW voltage level
1 = HIGH voltage level



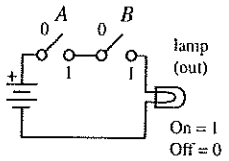


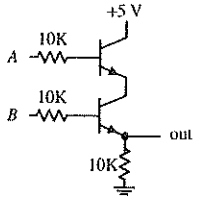
A NOT gate or *inverter* outputs a logic level that's the opposite (complement) of the input logic level.

AND



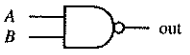
A	B	out
0	0	0
0	1	0
1	0	0
1	1	1



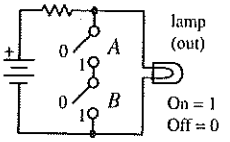


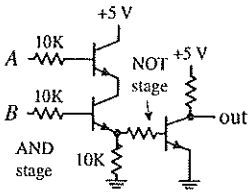
The ouput of an AND gate is HIGH only when both inputs are HIGH.

NAND



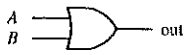
A	B	out
0	0	1
0	1	1
1	0	1
1	1	0



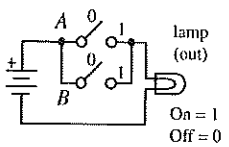


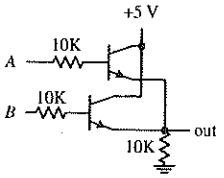
Combines the NOT function with an AND gate; output only goes LOW when both inputs are HIGH.

OR



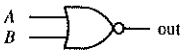
A	B	out
0	0	0
0	1	1
1	0	1
1	1	1



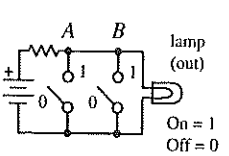


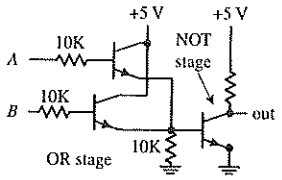
The output of an OR gate will go HIGH if one or both inputs goes HIGH. The output only goes LOW when both inputs are LOW.

NOR



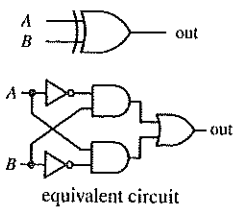
A	B	out
0	0	1
0	1	0
1	0	0
1	1	0





Combines the NOT function with an OR gate; output goes LOW if one or both inputs are LOW, ouput goes HIGH when both inputs are LOW.

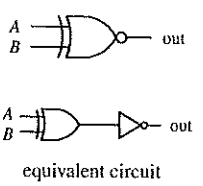
Exclusive OR (XOR)



A	B	out
0	0	0
0	1	1
1	0	1
1	1	0

The output of an XOR gate goes HIGH if the inputs are different from each other. XOR gates only come with two inputs.

Exclusive NOR (XNOR)



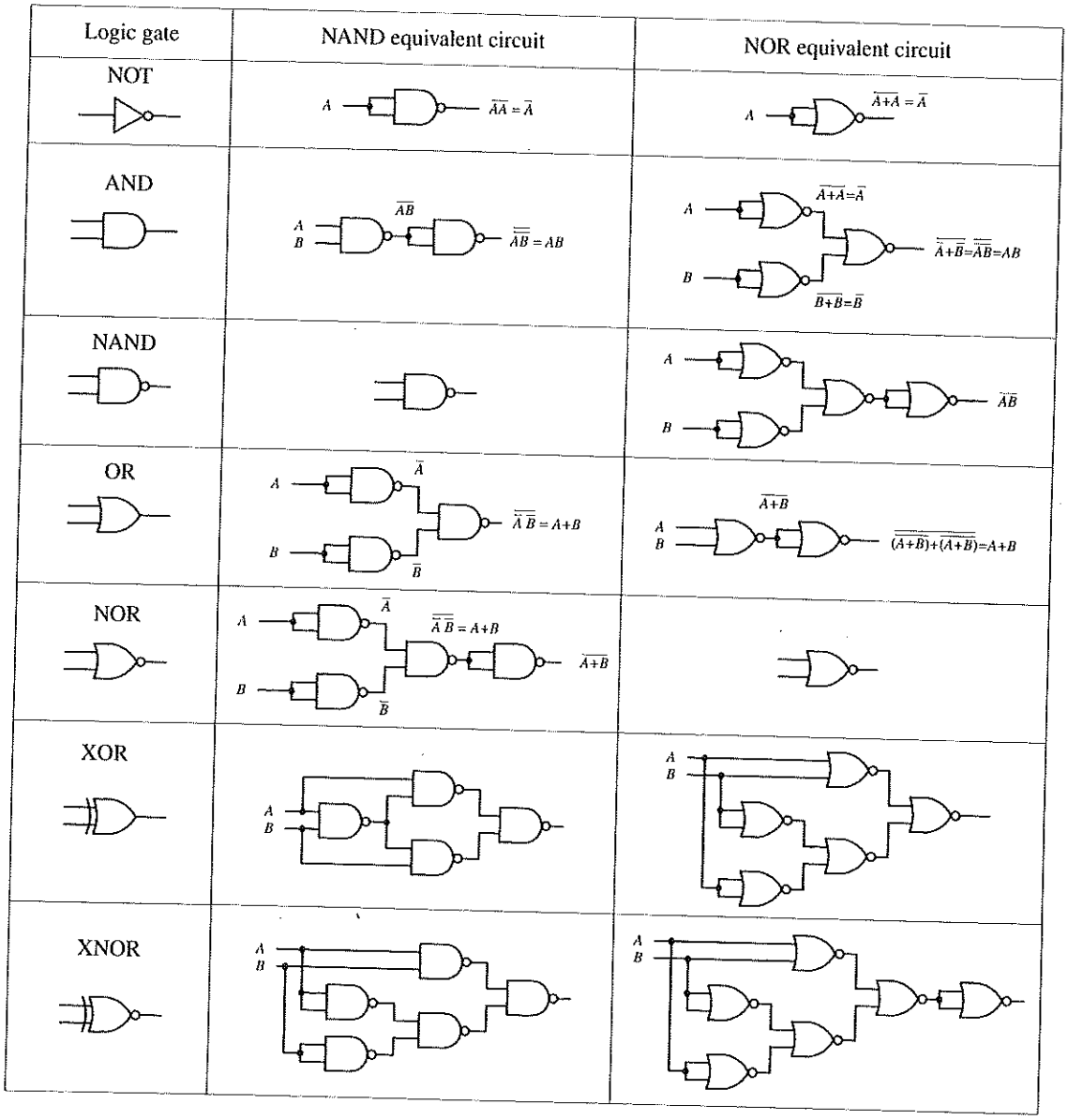
A	B	out
0	0	1
0	1	0
1	0	0
1	1	1

Combines the NOT function with an XOR gate; output goes HIGH if the inputs are the same.

Universal Capability of NAND and NOR Gates

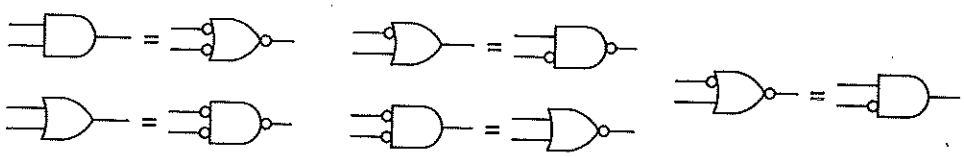
NAND and NOR gates are referred to as *universal gates* because each alone can be combined together with itself to form all other possible logic gates. The ability to create any logic gate from NAND or NOR gates is obviously a handy feature. For example, if you do not have an XOR IC handy, you can use a single multigate NAND gate (e.g., 74HC00) instead. The figure below shows how to wire NAND or NOR gates together to create equivalent circuits of the various logic gates.

FIGURE 12.24



Bubble Pushing

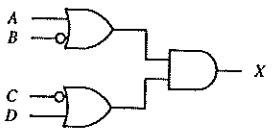
A shortcut method for forming equivalent logic circuits, based on De Morgan's theorem, is to use what's called *bubble pushing*.



AND-OR-INVERT Gates (AOIs)

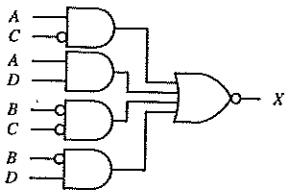
When a Boolean expression is reduced, the equation that is left over typically will be of one of the following two forms: *product-of-sums* (POS) or *sum-of-products* (SOP). A POS expression appears as two or more ORed variables ANDed together with two or more additional ORed variables. An SOP expression appears as two or more ANDed variables ORed together with additional ANDed variables. The figure below shows two circuits that provide the same logic function (they are equivalent), but the circuit to the left is designed to yield a POS expression, while the circuit to the right is designed to yield a SOP expression.

Logic circuit for POS expression



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

Logic circuit for SOP expression



$X = A\overline{C} + AD + \overline{B}\overline{C} + \overline{B}D$

Table made using SOP expression (it's easier than POS)

A	B	C	D	$A\overline{C}$	AD	$\overline{B}\overline{C}$	$\overline{B}D$	X
0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	1	1	1
0	0	1	0	0	0	0	0	0
0	0	1	1	0	0	0	1	1
0	1	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0
1	0	0	0	1	0	1	0	1
1	0	0	1	1	1	1	1	1
1	0	1	0	0	0	0	0	0
1	0	1	1	0	1	0	1	1
1	1	0	0	1	0	0	0	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	0	0	0
1	1	1	1	0	1	0	0	1

FIGURE 12.25

Bubble pushing involves the flowing tricks: First, change an AND gate to an OR gate or change an OR gate to an AND gate. Second, add inversion bubbles to the inputs and outputs where there were none, while removing the original bubbles. That's it. You can prove to yourself that this works by examining the corresponding truth tables for the original gate and the bubble-pushed gate, or you can work out the Boolean expressions using De Morgan's theorem. Figure 12.23 shows examples of bubble pushing.

LOGIC IDENTITIES

1) $A + B = B + A$
2) $AB = BA$
3) $A + (B + C) = (A + B) + C$
4) $A(BC) = (AB)C$
5) $A(B + C) = AB + AC$
6) $(A + B)(C + D) = AC + AD + BC + BD$
7) $\overline{1} = 0$
8) $\overline{0} = 1$
9) $A \cdot 0 = 0$
10) $A \cdot 1 = A$
11) $A + 0 = A$
12) $A + 1 = 1$
13) $A + A = A$
14) $AA = A$
15) $\overline{\overline{A}} = A$
16) $A + \overline{A} = 1$
17) $A\overline{A} = 0$
18) $\overline{A + B} = \overline{A}\,\overline{B}$
19) $\overline{AB} = \overline{A} + \overline{B}$
20) $A + \overline{A}B = A + B$
21) $\overline{A} + AB = \overline{A} + B$
22) $A \oplus B = \overline{A}B + A\overline{B} = (A + B)(\overline{AB})$
23) $\overline{A \oplus B} = AB + \overline{A}\,\overline{B}$

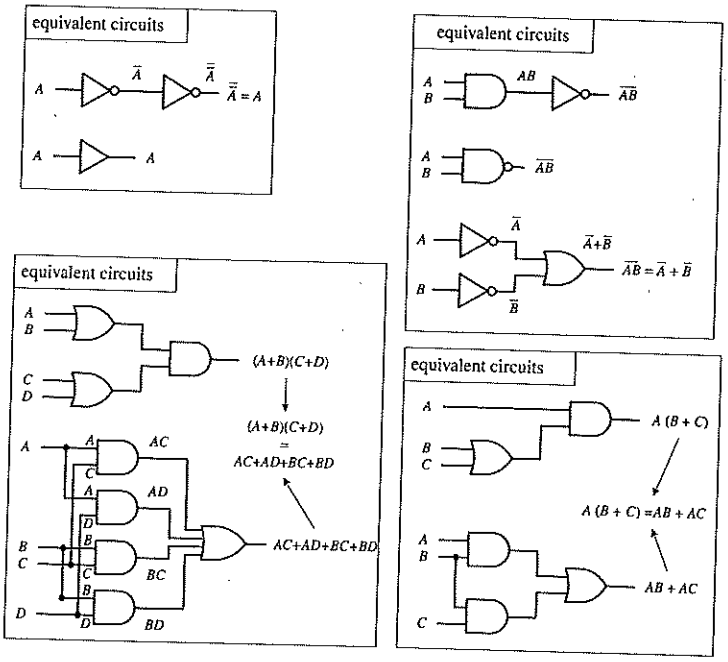
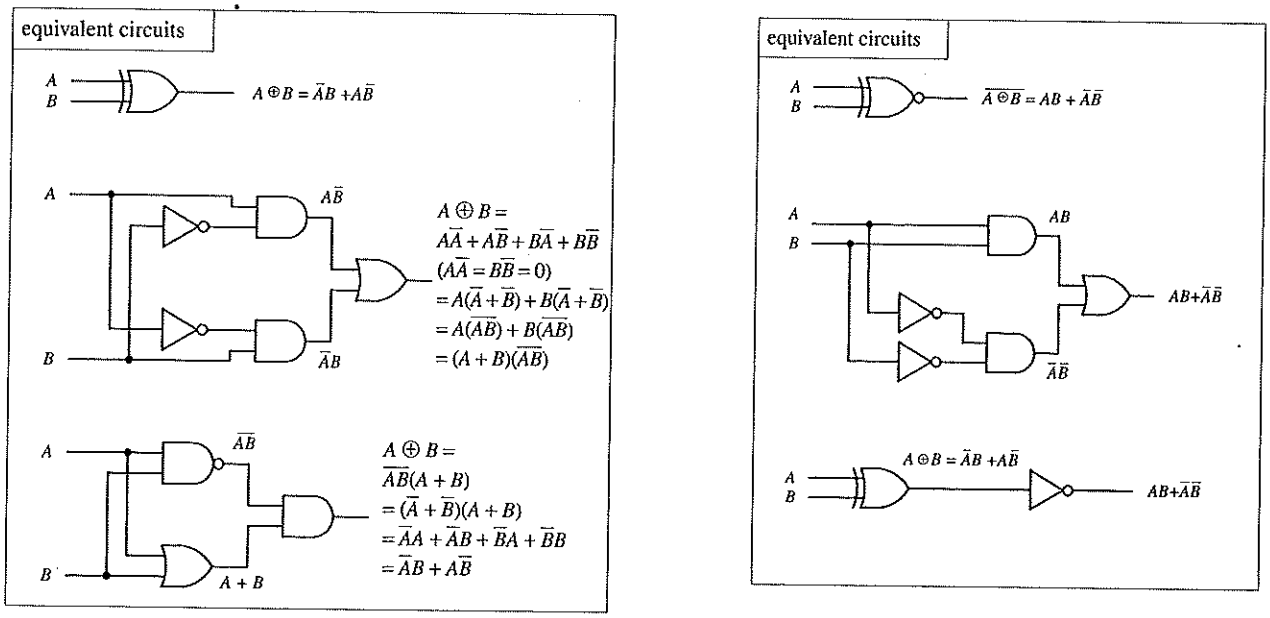
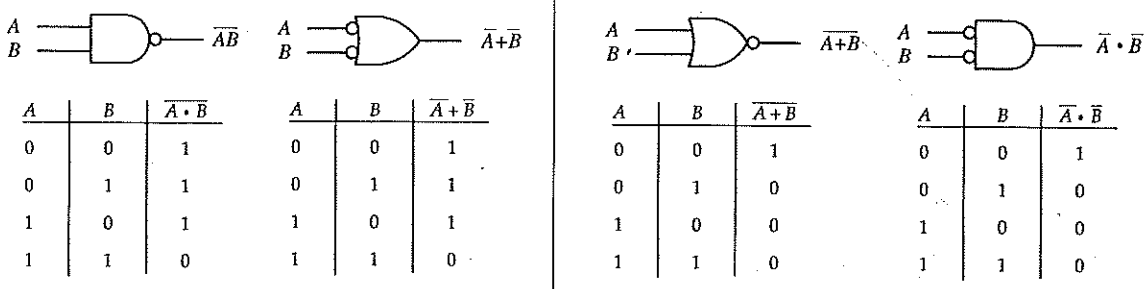


FIGURE 12.19

FIGURE 12.21





A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

A	B	$\overline{A} + \overline{B}$
0	0	1
0	1	1
1	0	1
1	1	0

A	B	$\overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

A	B	$\overline{A} \cdot \overline{B}$
0	0	1
0	1	0
1	0	0
1	1	0

FIGURE 12.22