Logic Expressions:

$X = \overline{A}$ … Inverter

$X = AB$ … AND

$X = A + B$ ... OR

$X = \overline{AB}$ ... NAND

$X = \overline{A + B}$ ... NOR

$Q = (A \oplus B) = \overline{A}.B + A.\overline{B}$ … X-OR

$Q = \overline{(A \oplus B)} = (A.B) + (\overline{A}.\overline{B})$ … X-NOR

Boolean Algebra and Logic Simplification

Laws and Rules of Boolean algebra:

1. Commutative law

The *commutative law of addition* for two variables is written as

$$A + B = B + A \qquad \text{Equation 4–1}$$

This law states that the order in which the variables are ORed makes no difference. Remember, in Boolean algebra as applied to logic circuits, addition and the OR operation are the same. Figure 4–3 illustrates the commutative law as applied to the OR gate and shows that it doesn't matter to which input each variable is applied. (The symbol $\equiv$ means "equivalent to.")

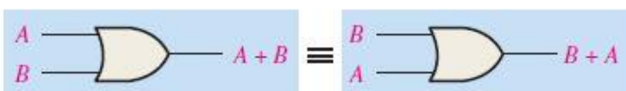


FIGURE 4–3 Application of commutative law of addition.

The *commutative law of multiplication* for two variables is

$$AB = BA \qquad \text{Equation 4–2}$$

This law states that the order in which the variables are ANDed makes no difference. Figure 4–4 illustrates this law as applied to the AND gate. Remember, in Boolean algebra as applied to logic circuits, multiplication and the AND function are the same.

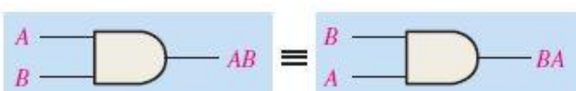


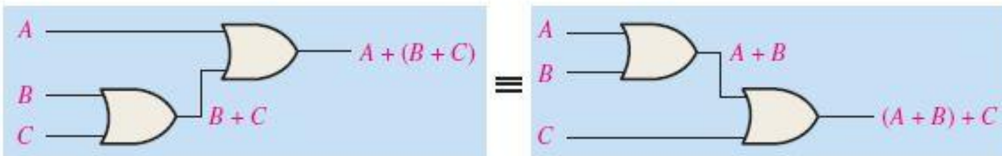
FIGURE 4–4 Application of commutative law of multiplication.

2. Associative law

The *associative law of addition* is written as follows for three variables:

$$A + (B + C) = (A + B) + C \qquad \text{Equation 4–3}$$

This law states that when ORing more than two variables, the result is the same regardless of the grouping of the variables. Figure 4–5 illustrates this law as applied to 2-input OR gates.

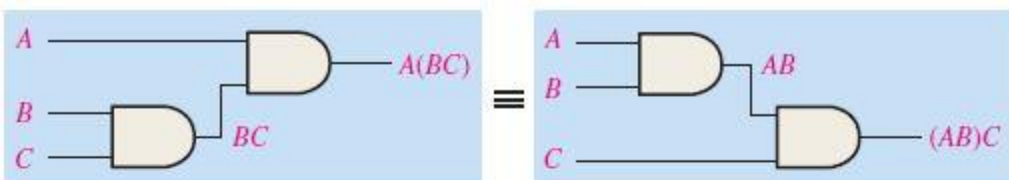


Application of associative law of addition

The associative law of multiplication is written as follows for three variables:

$$A (BC) = (AB) C$$

This law states that it makes no difference in what order the variables are grouped when ANDing more than two variables. Figure 4–6 illustrates this law as applied to 2-input AND gates.



Application of associative law of multiplication.

3. Distributive law

The distributive law is written for three variables as follows:

$$A (B + C) = AB + AC \qquad \text{Equation 4–5}$$

This law states that ORing two or more variables and then ANDing the result with a single variable is equivalent to ANDing the single variable with each of the two or more variables and then ORing the products. The distributive law also expresses the process of factoring in which the common variable A is factored out of the product terms, for example,

$AB + AC = A(B + C)$. Figure 4–7 illustrates the distributive law in terms of gate implementation.

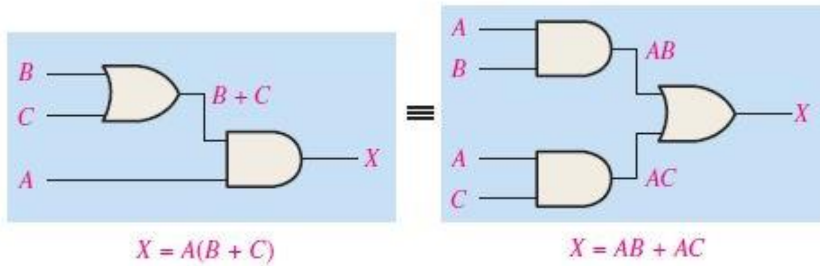


FIGURE 4–7 Application of distributive law. Open file F04-07 to verify.

Rules of Boolean algebra

The following table lists 12 basic rules that are useful in manipulating and simplifying Boolean expressions. Rules 1 through 9 will be viewed in terms of their application to logic gates. Rules 10 through 12 will be derived in terms of the simpler rules and the laws previously discussed.

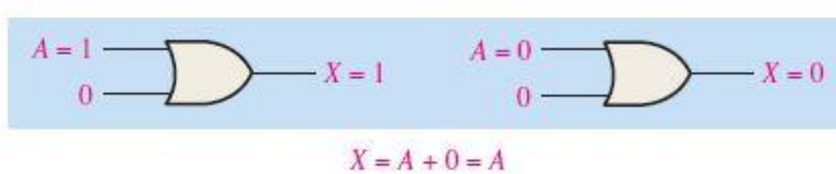
TABLE 4–1

Basic rules of Boolean algebra.

1. $A + 0 = A$	**7.** $A \cdot A = A$
2. $A + 1 = 1$	**8.** $A \cdot \overline{A} = 0$
3. $A \cdot 0 = 0$	**9.** $\overline{\overline{A}} = A$
4. $A \cdot 1 = A$	**10.** $A + AB = A$
5. $A + A = A$	**11.** $A + \overline{A}B = A + B$
6. $A + \overline{A} = 1$	**12.** $(A + B)(A + C) = A + BC$

A, *B*, or *C* can represent a single variable or a combination of variables.

Rule 1: $A + 0 = A$ A variable ORed with 0 is always equal to the variable. If the input variable A is 1, the output variable X is 1, which is equal to A. If A is 0, the output is 0, which is also equal to A. This rule is illustrated in Figure 4–8, where the lower input is fixed at 0.



Rule 2: $A + 1 = 1$ A variable ORed with 1 is always equal to 1. A 1 on an input to an OR gate produces a 1 on the output, regardless of the value of the variable on the other input. This rule is illustrated in Figure 4–9, where the lower input is fixed at 1.

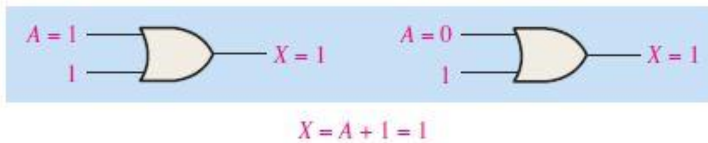
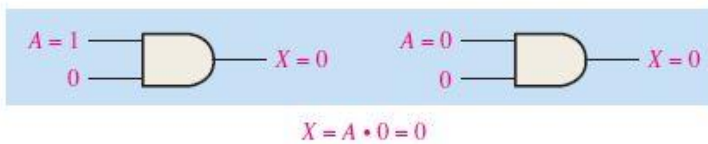
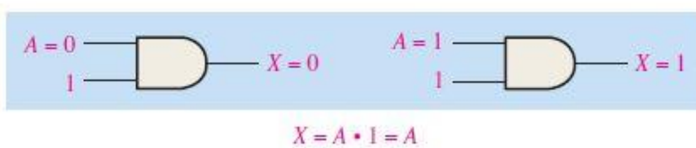


FIGURE 4–9

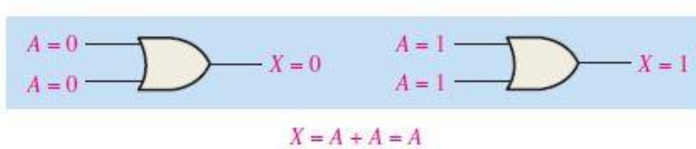
Rule 3: $A \cdot 0 = 0$ A variable ANDed with 0 is always equal to 0. Any time one input to an AND gate is 0, the output is 0, regardless of the value of the variable on the other input. This rule is illustrated in Figure 4–10, where the lower input is fixed at 0.



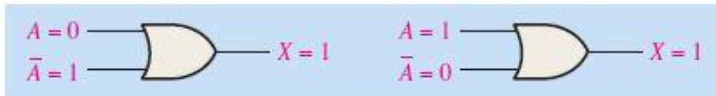
Rule 4: $A \cdot 1 = A$ A variable ANDed with 1 is always equal to the variable. If A is 0, the output of the AND gate is 0. If A is 1, the output of the AND gate is 1 because both inputs are now 1s. This rule is shown in Figure 4–11, where the lower input is fixed at 1.



Rule 5: $A + A = A$ A variable ORed with itself is always equal to the variable. If A is 0, then $0 + 0 = 0$; and if A is 1, then $1 + 1 = 1$. This is shown in Figure 4–12, where both inputs are the same variable.

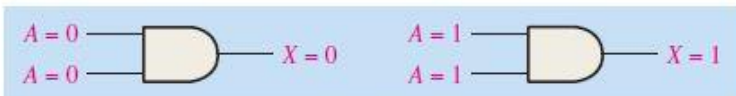


Rule 6: $A + \overline{A} = 1$ A variable ORed with its complement is always equal to 1. If A is 0, then $0 + \overline{0} = 0 + 1 = 1$. If A is 1, then $1 + \overline{1} = 1 + 0 = 1$. See Figure 4–13, where one input is the complement of the other.



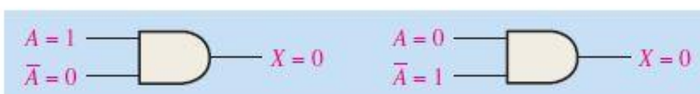
$X = A + \overline{A} = 1$

Rule 7: $A \cdot A = A$ A variable ANDed with itself is always equal to the variable. If $A = 0$, then $0 \cdot 0 = 0$; and if $A = 1$, then $1 \cdot 1 = 1$. Figure 4–14 illustrates this rule.



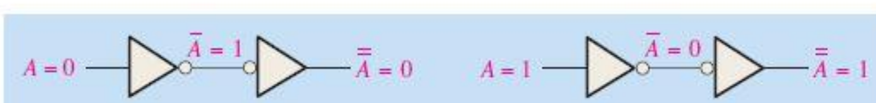
$X = A \cdot A = A$

Rule 8: $A \cdot \overline{A} = 0$ A variable ANDed with its complement is always equal to 0. Either A or $\overline{A}$ will always be 0; and when a 0 is applied to the input of an AND gate, the output will be 0 also. Figure 4–15 illustrates this rule.



$X = A \cdot \overline{A} = 0$

Rule 9: $\overline{\overline{A}} = A$ The double complement of a variable is always equal to the variable. If you start with the variable A and complement (invert) it once, you get $\overline{A}$. If you then take $\overline{A}$ and complement (invert) it, you get A, which is the original variable. This rule is shown in Figure 4–16 using inverters.



$\overline{\overline{A}} = A$

Rule 10: $A + AB = A$ This rule can be proved by applying the distributive law, rule 2, and rule 4 as follows:

$$\begin{aligned} A + AB = A \cdot 1 + AB = A(1 + B) &\quad \text{Factoring (distributive law)} \\ = A \cdot 1 &\quad \text{Rule 2: } (1 + B) = 1 \\ = A &\quad \text{Rule 4: } A \cdot 1 = A \end{aligned}$$

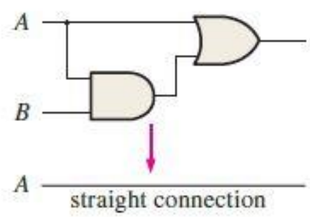
The proof is shown in Table 4–2, which shows the truth table and the resulting logic circuit simplification.

TABLE 4–2

Rule 10: $A + AB = A$. Open file T04-02 to verify.

A	B	AB	$A + AB$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

equal



Rule 11: $A + \overline{A}B = A + B$ This rule can be proved as follows:

$$\begin{aligned} A + \overline{A}B &= (A + AB) + \overline{A}B &\quad \text{Rule 10: } A = A + AB \\ &= (AA + AB) + \overline{A}B &\quad \text{Rule 7: } A = AA \\ &= AA + AB + A\overline{A} + \overline{A}B &\quad \text{Rule 8: adding } A\overline{A} = 0 \\ &= (A + \overline{A})(A + B) &\quad \text{Factoring} \\ &= 1 \cdot (A + B) &\quad \text{Rule 6: } A + \overline{A} = 1 \\ &= A + B &\quad \text{Rule 4: drop the 1} \end{aligned}$$

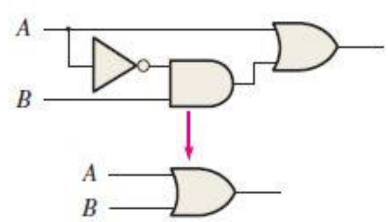
The proof is shown in Table 4–3, which shows the truth table and the resulting logic circuit simplification.

TABLE 4–3

Rule 11: $A + \overline{A}B = A + B$.

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

equal



Rule 12: $(A + B)(A + C) = A + BC$ This rule can be proved as follows:

$$
\begin{aligned}
(A + B)(A + C) &= AA + AC + AB + BC && \text{Distributive law} \\
&= A + AC + AB + BC && \text{Rule 7: } AA = A \\
&= A(1 + C) + AB + BC && \text{Factoring (distributive law)} \\
&= A \cdot 1 + AB + BC && \text{Rule 2: } 1 + C = 1 \\
&= A(1 + B) + BC && \text{Factoring (distributive law)} \\
&= A \cdot 1 + BC && \text{Rule 2: } 1 + B = 1 \\
&= A + BC && \text{Rule 4: } A \cdot 1 = A
\end{aligned}
$$

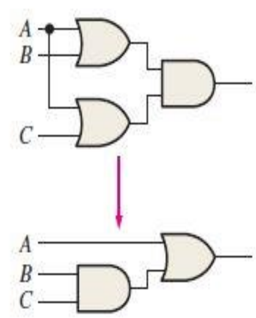
The proof is shown in Table 4–4, which shows the truth table and the resulting logic circuit simplification.

TABLE 4–4

Rule 12: $(A + B)(A + C) = A + BC$. Open file T04-04 to verify.

A	B	C	$A+B$	$A+C$	$(A+B)(A+C)$	BC	$A+BC$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

equal



H.W:

Prove that

(X+Y) (X+Z) =X+YZ