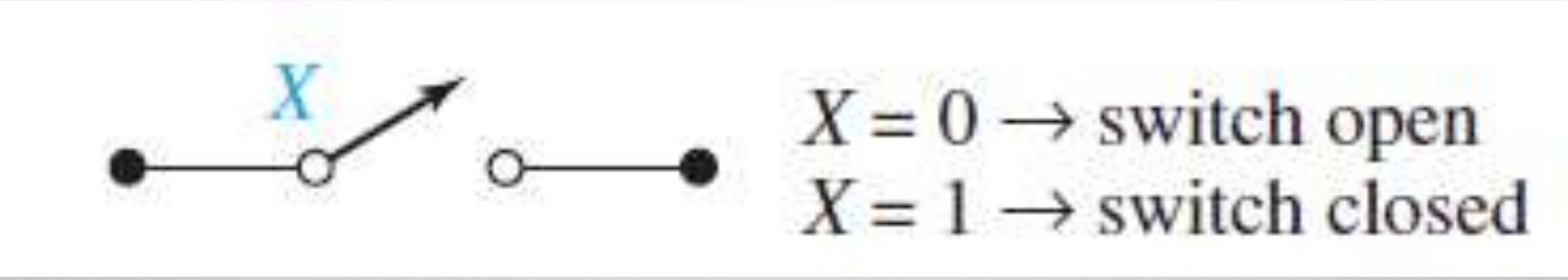
UNIT 2

BOOLEAN ALGEBRA

LETURE 4

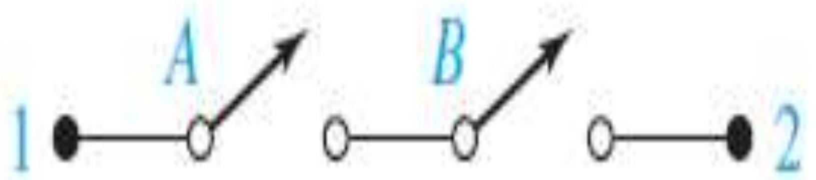
switching algebra to describe circuits

- We will apply switching algebra to describe circuits containing switches. We will label each switch with a variable. If switch X is open, then we will define the value of X to be 0; if switch X is closed, then we will define the value of X to be 1.



switching algebra to describe circuits

- Now consider a circuit composed of two switches in a series.
- We will define the transmission between the terminals as $T = 0$ if there is an open circuit between the terminals and $T = 1$ if there is a closed circuit between the terminals.



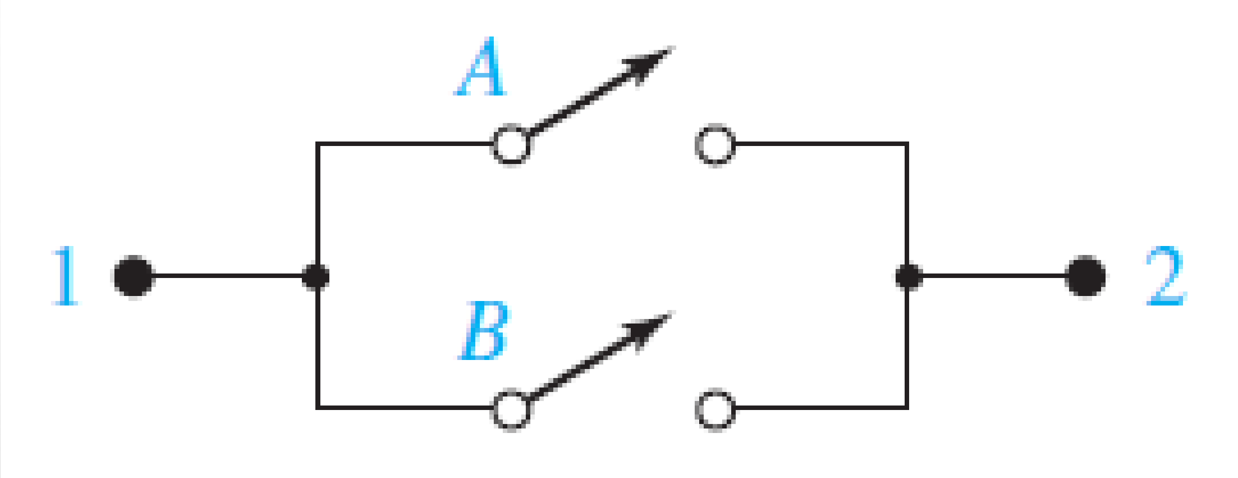
$T = 0 \rightarrow$ open circuit between terminals 1 and 2
$T = 1 \rightarrow$ closed circuit between terminals 1 and 2

Now we have a closed circuit between terminals 1 and 2 ($T = 1$) iff (if and only if) switch *A* is closed *and* switch *B* is closed. Stating this algebraically,

$$T = A \cdot B$$

switching algebra to describe circuits

- Next consider a circuit composed of two switches in parallel.



In this case, we have a closed circuit between terminals 1 and 2 iff switch A is closed or switch B is closed. Using the same convention for defining variables as above, an equation which describes the behavior of this circuit is

T = A + B

2.6 SIMPLIFICATION THEOREMS

$XY + XY' = X$	(2-12)	$(X + Y)(X + Y') = X$	(2-12D)
$X + XY = X$	(2-13)	$X(X + Y) = X$	(2-13D)
$(X + Y')Y = XY$	(2-14)	$XY' + Y = X + Y$	(2-14D)

- In each case, one expression can be replaced by a simpler one. Because each expression corresponds to a circuit of logic gates, simplifying an expression leads to simplifying the corresponding logic circuit.

2.6 SIMPLIFICATION THEOREMS

- Each of the preceding theorems can be proved by using a truth table, or they can be proved algebraically starting with the basic theorems.

Proof of (2-13): $X + XY = X \cdot 1 + XY = X(1 + Y) = X \cdot 1 = X$

Proof of (2-13D): $X(X + Y) = XX + XY = X + XY = X$

(by (2-6D) and (2-13))

Proof of (2-14D): $Y + XY' = (Y + X)(Y + Y') = (Y + X)1 = Y + X$

(by (2-11 D) and (2-8))

2.6 SIMPLIFICATION THEOREMS

- The following example illustrates simplification of a logic gate circuit using one of the theorems. In Figure 2-4, the output of circuit (a) is

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

$$F = AA' + AB = 0 + AB = AB$$

- By Theorem (2-14), the expression for F simplifies to AB. Therefore, circuit (a) can be replaced with the equivalent circuit (b).

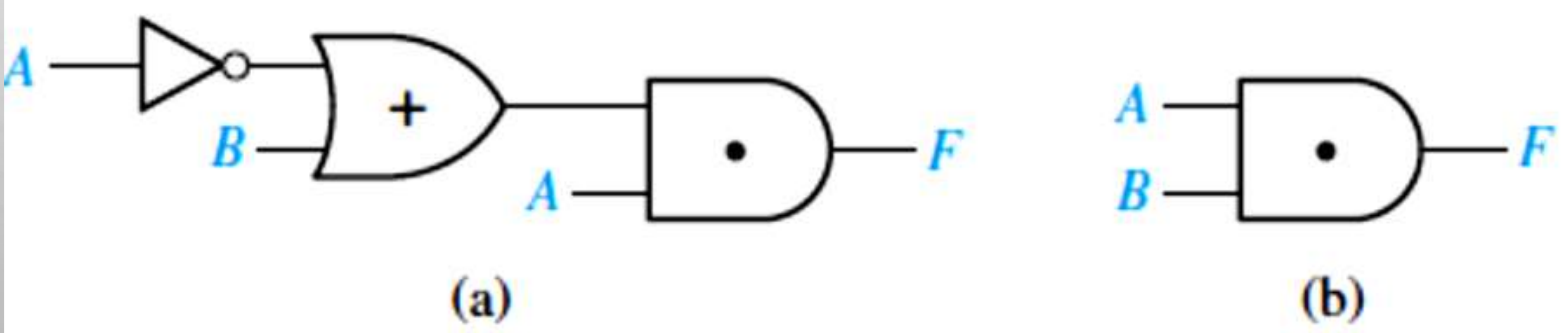


FIGURE 2-4: Equivalent Gate Circuits

EXAMPLE 1

SIMPLIFY $Z = A'BC + A'$

This expression has the same form as (2-13) if we let $X = A'$ and $Y = BC$.

Therefore, the expression simplifies to $Z = X + XY = X(1+Y) = X = A'$.

EXAMPLE 2

Simplify $$Z = [\underbrace{A + B'C}_{X} + \underbrace{D + EF}_{Y}]\,[\underbrace{A + B'C}_{X} + \underbrace{(D + EF)'}_{Y'}]$$

Substituting: $Z = [\ X + Y\ ]\,[\ X + Y'\ ]$

Then, by (2-12D), the expression reduces to

$$Z = X = A + B'C$$

EXAMPLE 3

Simplify $Z = \underbrace{(AB + C)}_{Y'}\ \underbrace{(B'D + C'E')}_{X} + \underbrace{(AB + C)'}_{Y}$

Substituting: $Z = Y'X + Y$

By, (2-14D): $Z = X + Y = B'D + C'E' + (AB + C)'$

2.8 Demorgan's Laws

- The inverse or complement of any boolean expression can easily be found by successively applying the following theorems, which are frequently referred to as demorgan's laws:

$$(X + Y)' = X'Y' \tag{2-21}$$

$$(XY)' = X' + Y' \tag{2-22}$$

- We will verify these laws using a truth table:

X Y	X′ Y′	X + Y	(X + Y)′	X′ Y′	XY	(XY)′	X′ + Y′
0 0	1 1	0	1	1	0	1	1
0 1	1 0	1	0	0	0	1	1
1 0	0 1	1	0	0	0	1	1
1 1	0 0	1	0	0	1	0	0

2.8 Demorgan's Laws

- Demorgan's laws are easily generalized to *n* variables:

$$(X_1 + X_2 + X_3 + \ldots + X_n)' = X_1' X_2' X_3' \ldots X_n' \tag{2-23}$$

$$(X_1 X_2 X_3 \ldots X_n)' = X_1' + X_2' + X_3' + \ldots + X_n' \tag{2-24}$$

For example, for $n = 3$,

$$(X_1 + X_2 + X_3)' = (X_1 + X_2)'X_3' = X_1'X_2'X_3'$$

2.8 Demorgan's Laws

Referring to the OR operation as the logical sum and the AND operation as logical product, DeMorgan's laws can be stated as

The complement of the product is the sum of the complements.

The complement of the sum is the product of the complements.

To form the complement of an expression containing both OR and AND operations, DeMorgan's laws are applied alternately.

Example 1

- To find the complement of $(A' + B)C'$, first apply (2-22) and then (2-21).

$$[(A' + B)C']' = (A' + B)' + (C')' = AB' + C$$

Example 2

$$\begin{aligned}
[(AB' + C)D' + E]' &= [(AB' + C)D']'E' && \text{(by (2-21))} \\
&= [(AB' + C)' + D]E' && \text{(by (2-22))} \\
&= [(AB')'C' + D]E' && \text{(by (2-21))} \\
&= [(A' + B)C' + D]E' && \text{(by (2-22))} \qquad (2\text{-}25)
\end{aligned}$$

Note that in the final expressions, the complement operation is applied only to single variables.

2.8 Demorgan's Laws

The inverse of $F = A'B + AB'$ is

$$F' = (A'B + AB')' = (A'B)'(AB')' = (A + B')(A' + B)$$
$$= AA' + AB + B'A' + BB' = A'B' + AB$$

We will verify that this result is correct by constructing a truth table for *F* and *F'*:

$A\ B$	$A'B$	AB'	$F = A'B + AB'$	$A'B'$	AB	$F' = A'B' + AB$
0 0	0	0	0	1	0	1
0 1	1	0	1	0	0	0
1 0	0	1	1	0	0	0
1 1	0	0	0	0	1	1

2.8 Demorgan's Laws

In the table, note that for every combination of values of A and B for which $F = 0$, $F' = 1$; and whenever $F = 1$, $F' = 0$.

Given a Boolean expression, the **dual** is formed by replacing AND with OR,OR with AND, 0 with 1, and 1 with 0. Variables and complements are left unchanged.

The dual of AND is OR and the dual of OR is AND:

$$(XYZ\ldots)^D = X + Y + Z + \ldots \qquad (X + Y + Z + \ldots)^D = XYZ\ldots \qquad (2\text{-}26)$$

2.8 Demorgan's Laws

The dual of an expression may be found by complementing the entire expression and then complementing each individual variable. For example, to find the dual of $AB' + C$,

$$(XYZ\ldots)^D = X + Y + Z + \ldots \qquad (X + Y + Z + \ldots)^D = XYZ\ldots \qquad (2\text{-}26)$$

Laws and Theorems of Boolean Algebra

Operations with 0 and 1:

1. $X + 0 = X$
2. $X + 1 = 1$

1D. $X \cdot 1 = X$
2D. $X \cdot 0 = 0$

Idempotent laws:

3. $X + X = X$

3D. $X \cdot X = X$

Involution law:

4. $(X')' = X$

Laws of complementarity:

5. $X + X' = 1$

5D. $X \cdot X' = 0$

Commutative laws:

6. $X + Y = Y + X$

6D. $XY = YX$

Associative laws:

7. $(X + Y) + Z = X + (Y + Z) = X + Y + Z$

7D. $(XY)Z = X(YZ) = XYZ$

Laws and Theorems of Boolean Algebra

Distributive laws:

8. $X(Y + Z) = XY + XZ$ 8D. $X + YZ = (X + Y)(X + Z)$

Simplification theorems:

9. $XY + XY' = X$ 9D. $(X + Y)(X + Y') = X$
10. $X + XY = X$ 10D. $X(X + Y) = X$
11. $(X + Y')Y = XY$ 11D. $XY' + Y = X + Y$

DeMorgan's laws:

12. $(X + Y + Z + \ldots)' = X'Y'Z' \ldots$ 12D. $(XYZ \ldots)' = X' + Y' + Z' + \ldots$

Duality:

13. $(X + Y + Z + \ldots)^D = XYZ \ldots$ 13D. $(XYZ \ldots)^D = X + Y + Z + \ldots$

Theorem for multiplying out and factoring:

14. $(X + Y)(X' + Z) = XZ + X'Y$ 14D. $XY + X'Z = (X + Z)(X' + Y)$

Consensus theorem:

15. $XY + YZ + X'Z = XY + X'Z$ 15D. $(X + Y)(Y + Z)(X' + Z) = (X + Y)(X' + Z)$

Problems

2.1 Prove the following theorems algebraically:
(a) $X(X' + Y) = XY$ (b) $X + XY = X$
(c) $XY + XY' = X$ (d) $(A + B)(A + B') = A$

2.2 Illustrate the following theorems using circuits of switches:
(a) $X + XY = X$ (b) $X + YZ = (X + Y)(X + Z)$
In each case, explain why the circuits are equivalent.

2.3 Simplify each of the following expressions by applying *one* of the theorems. State the theorem used (see page 55).
(a) $X'Y'Z + (X'Y'Z)'$ (b) $(AB' + CD)(B'E + CD)$
(c) $ACF + AC'F$ (d) $A(C + D'B) + A'$
(e) $(A'B + C + D)(A'B + D)$ (f) $(A + BC) + (DE + F)(A + BC)'$

UNIT 3

BOOLEAN ALGEBRA (CONTINUED)

OBJECTIVES

- When you complete this unit, you should know from memory and be able to use any of the laws and theorems of boolean algebra listed at the end of unit 2. Specifically, you should be able to
- **1.** Apply these laws and theorems to the manipulation of algebraic expressions including:

  A. Simplifying an expression.

  B. Finding the complement of an expression.

  C. Multiplying out and factoring an expression.
- **2.** Prove any of the theorems using a truth table or give an algebraic proof if appropriate.
- **3.** Define the exclusive-or and equivalence operations. State, prove, and use the basic theorems that concern these operations.
- **4.** Use the consensus theorem to delete terms from and add terms to a switching expression.
- **5.** Given an equation, prove algebraically that it is valid or show that it is not valid.

3.1 MULTIPLYING OUT AND FACTORING EXPRESSIONS

- Given an expression in product-of-sums form, the corresponding sum-of-products expression can be obtained by multiplying out, using the two distributive laws:

$$X(Y + Z) = XY + XZ \tag{3-1}$$

$$(X + Y)(X + Z) = X + YZ \tag{3-2}$$

In addition, the following theorem is very useful for factoring and multiplying out:

$$(X + Y)(X' + Z) = XZ + X'Y \tag{3-3}$$

3.1 MULTIPLYING OUT AND FACTORING EXPRESSIONS

- **Note that** the variable that is paired with *X* on one side of the equation is paired with *X* on the other side, and vice versa.
- Proof:

If $X = 0$, (3-3) reduces to $Y(1 + Z) = 0 + 1 \cdot Y$ or $Y = Y$.
If $X = 1$, (3-3) reduces to $(1 + Y)Z = Z + 0 \cdot Y$ or $Z = Z$.

Because the equation is valid for both $X = 0$ and $X = 1$, it is always valid.
The following example illustrates the use of Theorem (3-3) for factoring:

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

3.1 MULTIPLYING OUT AND FACTORING EXPRESSIONS

- Note that the theorem can be applied when we have two terms, one which contains a variable and another which contains its complement.
- Theorem (3-3) is very useful for multiplying out expressions. In the following example, we can apply (3-3) because one factor contains the variable Q, and the other factor contains Q'.

$$\boxed{(Q + AB')(C'D + Q') = QC'D + Q'AB'}$$

3.1 MULTIPLYING OUT AND FACTORING EXPRESSIONS

- If we simply multiplied out by using the distributive law, we would get four terms instead of two:

$$(Q + AB')(C'D + Q') = QC'D + QQ' + AB'C'D + AB'Q'$$

- Because the term *AB'C'D* is difficult to eliminate, it is much better to use (3-3) instead of the distributive law.

3.1 MULTIPLYING OUT AND FACTORING EXPRESSIONS

- **In general**, when we multiply out an expression, we should use (3-3) along with (3-1) and (3-2).
- **To avoid generating unnecessary terms when multiplying out**, (3-2) and (3-3) should generally be applied before (3-1), and terms should be grouped to expedite their application.

EXAMPLE

$$(A + B + C')(A + B + D)(A + B + E)(A + D' + E)(A' + C)$$

$$= (A + B + C'D)(A + B + E)[AC + A'(D' + E)]$$

$$= (A + B + C'DE)(AC + A'D' + A'E)$$

$$= AC + \cancel{ABC} + A'BD' + A'BE + A'C'DE \qquad (3\text{-}4)$$

- **The same theorems that are useful for multiplying out expressions are useful for factoring. By repeatedly applying (3-1), (3-2), and (3-3), any expression can be converted to a product-of-sums form.**

EXAMPLE OF FACTORING

$$AC + A'BD' + A'BE + A'C'DE$$

$$= \underbrace{AC}_{XZ} + \underbrace{A'}_{X'}\underbrace{(BD' + BE + C'DE)}_{Y}$$

$$= (A + BD' + BE + C'DE)(A' + C)$$

$$= [\underbrace{A + C'DE}_{X} + \underbrace{B}_{Y}(\underbrace{D' + E}_{Z})](A' + C)$$

$$= (A + B + C'DE)(A + \cancel{C'DE} + D' + E)(A' + C)$$

$$= (A + B + C')(A + B + D)(A + B + E)(A + D' + E)(A' + C) \quad (3\text{-}5)$$

- This is the same expression we started with in (3-4).

3.2 EXCLUSIVE-OR AND EQUIVALENCE OPERATIONS

- The *exclusive-or* operation ($\oplus$) is defined as follows:

$$0 \oplus 0 = 0 \qquad 0 \oplus 1 = 1$$
$$1 \oplus 0 = 1 \qquad 1 \oplus 1 = 0$$

- The truth table for $X \oplus Y$ is

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

3.2 EXCLUSIVE-OR AND EQUIVALENCE OPERATIONS

From this table, we can see that $X \oplus Y = 1$ iff $X = 1$ or $Y = 1$, but *not* both. The ordinary OR operation, which we have previously defined, is sometimes called inclusive OR because $X + Y = 1$ iff $X = 1$ or $Y = 1$, or both.

Exclusive OR can be expressed in terms of AND and OR. Because $X \oplus Y = 1$ iff X is 0 and Y is 1 or X is 1 and Y is 0, we can write

$$X \oplus Y = X'Y + XY' \tag{3-6}$$

3.2 EXCLUSIVE-OR AND EQUIVALENCE OPERATIONS

The first term in (3-6) is 1 if $X = 0$ and $Y = 1$; the second term is 1 if $X = 1$ and $Y = 0$. Alternatively, we can derive Equation (3-6) by observing that $X \oplus Y = 1$ iff $X = 1$ or $Y = 1$ *and* X and Y are not both 1. Thus,

$$X \oplus Y = (X + Y)(XY)' = (X + Y)(X' + Y') = X'Y + XY' \qquad (3\text{-}7)$$

In (3-7), note that $(X\,Y)' = 1$ if X and Y are not both 1.

We will use the following symbol for an exclusive-OR gate:

3.2 EXCLUSIVE-OR AND EQUIVALENCE OPERATIONS

The following theorems apply to exclusive OR:

$$X \oplus 0 = X \tag{3-8}$$

$$X \oplus 1 = X' \tag{3-9}$$

$$X \oplus X = 0 \tag{3-10}$$

$$X \oplus X' = 1 \tag{3-11}$$

$$X \oplus Y = Y \oplus X \text{ (commutative law)} \tag{3-12}$$

$$(X \oplus Y) \oplus Z = X \oplus (Y \oplus Z) = X \oplus Y \oplus Z \text{ (associative law)} \tag{3-13}$$

$$X(Y \oplus Z) = XY \oplus XZ \text{ (distributive law)} \tag{3-14}$$

$$(X \oplus Y)' = X \oplus Y' = X' \oplus Y = XY + X'Y' \tag{3-15}$$

3.2 EXCLUSIVE-OR AND EQUIVALENCE OPERATIONS

- Any of these theorems can be proved by using a truth table or by replacing $X \oplus Y$ with one of the equivalent expressions from Equation (3-7).
- Proof of the distributive law follows:

$$\begin{aligned} XY \oplus XZ &= XY(XZ)' + (XY)'XZ = XY(X' + Z') + (X' + Y')XZ \\ &= XYZ' + XY'Z \\ &= X(YZ' + Y'Z) = X(Y \oplus Z) \end{aligned}$$

3.2 EXCLUSIVE-OR AND EQUIVALENCE OPERATIONS

The *equivalence* operation (≡) is defined by

$$(0 \equiv 0) = 1 \qquad (0 \equiv 1) = 0 \tag{3-16}$$
$$(1 \equiv 0) = 0 \qquad (1 \equiv 1) = 1$$

The truth table for $X \equiv Y$ is

X	Y	$X \equiv Y$
0	0	1
0	1	0
1	0	0
1	1	1

3.2 EXCLUSIVE-OR AND EQUIVALENCE OPERATIONS

From the definition of equivalence, we see that $(X \equiv Y) = 1$ iff $X = Y$. Because $(X \equiv Y) = 1$ iff $X = Y = 1$ or $X = Y = 0$, we can write

$$(X \equiv Y) = XY + X'Y' \tag{3-17}$$

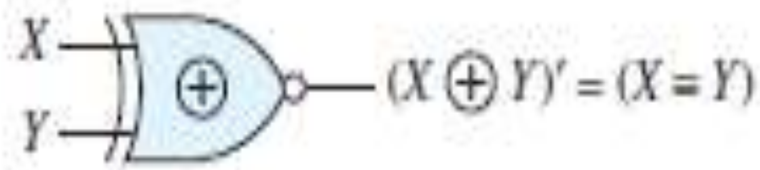
Equivalence is the complement of exclusive-OR:

$$\begin{aligned}(X \oplus Y)' &= (X'Y + XY')' = (X + Y')(X' + Y) \\ &= XY + X'Y' = (X \equiv Y) \end{aligned} \tag{3-18}$$

Just as for exclusive-OR, the equivalence operation is commutative and associative.
We will use the following symbol for an equivalence gate:

3.2 EXCLUSIVE-OR AND EQUIVALENCE OPERATIONS

Because equivalence is the complement of exclusive-OR, an alternate symbol for the equivalence gate is an exclusive-OR gate with a complemented output:



The equivalence gate is also called an exclusive-NOR gate.

3.2 EXCLUSIVE-OR AND EQUIVALENCE OPERATIONS

In order to simplify an expression which contains AND and OR as well as exclusive OR and equivalence, it is usually desirable to first apply (3-6) and (3-17) to eliminate the $\oplus$ and $\equiv$ operations. As an example, we will simplify

$$F = (A'B \equiv C) + (B \oplus AC')$$

By (3-6) and (3-17),

$$\begin{aligned} F &= [(A'B)C + (A'B)'C'] + [B'(AC') + B(AC')'] \\ &= A'BC + (A + B')C' + AB'C' + B(A' + C) \\ &= B(A'C + A' + C) + C'(A + B' + AB') = B(A' + C) + C'(A + B') \end{aligned}$$

When manipulating an expression that contains several exclusive-OR or equivalence operations, it is useful to note that

$$(XY' + X'Y)' = XY + X'Y' \tag{3-19}$$

3.2 EXCLUSIVE-OR AND EQUIVALENCE OPERATIONS

For example,

$$A' \oplus B \oplus C = [A'B' + (A')'B] \oplus C$$

$$= (A'B' + AB)C' + (A'B' + AB)'C \qquad \text{(by (3-6))}$$

$$= (A'B' + AB)C' + (A'B + AB')C \qquad \text{(by (3-19))}$$

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

3.3 THE CONSENSUS THEOREM

The consensus theorem is very useful in simplifying Boolean expressions. Given an expression of the form $XY + X'Z + YZ$, the term YZ is redundant and can be eliminated to form the equivalent expression $XY + X'Z$.

The term that was eliminated is referred to as the *consensus term*. Given a pair of terms for which a variable appears in one term and the complement of that variable in another, the consensus term is formed by multiplying the two original terms together, leaving out the selected variable and its complement. For example, the consensus of ab and $a'c$ is bc; the consensus of abd and $b'de'$ is $(ad)(de') = ade'$. The consensus of terms $ab'd$ and $a'bd'$ is 0.

3.3 THE CONSENSUS THEOREM

The consensus theorem can be stated as follows:

$$XY + X'Z + YZ = XY + X'Z \tag{3-20}$$

Proof:

$$\begin{aligned} XY + X'Z + YZ &= XY + X'Z + (X + X')YZ \\ &= (XY + XYZ) + (X'Z + X'YZ) \\ &= XY(1 + Z) + X'Z(1 + Y) = XY + X'Z \end{aligned}$$

3.3 THE CONSENSUS THEOREM

The consensus theorem can be used to eliminate redundant terms from Boolean expressions. For example, in the following expression, *bc* is the consensus of *ab* and *ac*, and *ab* is the consensus of *ac* and *bc*, so both consensus terms can be eliminated:

$$a'b' + ac + bc' + b'c + ab = a'b' + ac + bc'$$

The brackets indicate how the consensus terms are formed. The dual form of the consensus theorem is

$$(X + Y)(X' + Z)(Y + Z) = (X + Y)(X' + Z) \qquad (3\text{-}21)$$

3.4 ALGEBRAIC SIMPLIFICATION OF SWITCHING EXPRESSIONS

- In this section we review and summarize methods for simplifying switching expressions, using the laws and theorems of boolean algebra.
- In addition to multiplying out and factoring, three Basic ways of simplifying switching functions are combining terms, eliminating terms, and eliminating literals.

1. *Combining terms.* Use the theorem $XY + XY' = X$ to combine two terms. For example,

$$abc'd' + abcd' = abd' \qquad [X = abd', Y = c] \qquad (3\text{-}24)$$

3.4 ALGEBRAIC SIMPLIFICATION OF SWITCHING EXPRESSIONS

- When combining terms by this theorem, the two terms to be combined should contain exactly the same variables, and exactly one of the variables should appear complemented
- In one term and not in the other. Because $X \quad X \quad X$, a given term may be duplicated and combined with two or more other terms. **For example,**

$$ab'c + abc + a'bc = ab'c + abc + abc + a'bc = ac + bc$$

3.4 ALGEBRAIC SIMPLIFICATION OF SWITCHING EXPRESSIONS

- The theorem still can be used, of course, when X and Y are replaced with more complicated expressions. **For example,**

$$(a + bc)(d + e') + a'(b' + c')(d + e') = d + e'$$

$$[X = d + e', Y = a + bc, Y' = a'(b' + c')]$$

3.4 ALGEBRAIC SIMPLIFICATION OF SWITCHING EXPRESSIONS

2. *Eliminating terms.* Use the theorem $X + XY = X$ to eliminate redundant terms if possible; then try to apply the consensus theorem ($XY + X'Z + YZ = XY + X'Z$) to eliminate any consensus terms. For example,

$$a'b + a'bc = a'b \qquad [X = a'b]$$

$$a'bc' + bcd + a'bd = a'bc' + bcd \qquad [X = c, Y = bd, Z = a'b] \qquad (3\text{-}25)$$

3. *Eliminating literals.* Use the theorem $X + X'Y = X + Y$ to eliminate redundant literals. Simple factoring may be necessary before the theorem is applied.

3.4 ALGEBRAIC SIMPLIFICATION OF SWITCHING EXPRESSIONS

Example

$$
\begin{aligned}
A'B + A'B'C'D' + ABCD' &= A'(B + B'C'D') + ABCD' \\
&= A'(B + C'D') + ABCD' \\
&= B(A' + ACD') + A'C'D' \\
&= B(A' + CD') + A'C'D' \\
&= A'B + BCD' + A'C'D' \qquad (3\text{-}26)
\end{aligned}
$$

3.4 ALGEBRAIC SIMPLIFICATION OF SWITCHING EXPRESSIONS

4. *Adding redundant terms.* Redundant terms can be introduced in several ways such as adding xx', multiplying by $(x + x')$, adding yz to $xy + x'z$, or adding xy to x. When possible, the added terms should be chosen so that they will combine with or eliminate other terms.

Example

$WX + XY + X'Z' + WY'Z'$	(add WZ' by consensus theorem)
$= WX + XY + X'Z' + WY'Z' + WZ'$	(eliminate $WY'Z'$)
$= WX + XY + X'Z' + WZ'$	(eliminate WZ')
$= WX + XY + X'Z'$	(3-27)