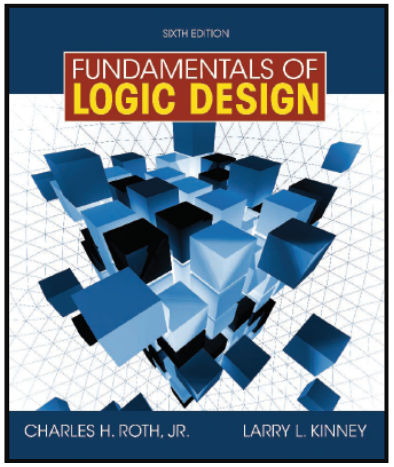
SLIDES FOR

CHAPTER 7

MULTI-LEVEL GATE CIRCUITS
NAND AND NOR GATES



This chapter in the book includes:

Click the mouse to move to the next page.
Use the ESC key to exit this chapter.

Multi-Level Gate Circuits

In this unit, we will use the following terminology:

1. *AND-OR circuit* means a two-level circuit composed of a level of AND gates followed by an OR gate at the output.
2. *OR-AND circuit* means a two-level circuit composed of a level of OR gates followed by an AND gate at the output.
3. *OR-AND-OR circuit* means a three-level circuit composed of a level of OR gates followed by a level of AND gates followed by an OR gate at the output.
4. Circuit of AND and OR gates implies no particular ordering of the gates; the output gate may be either AND or OR.

Section 7.1 (p. 191)

Tree Diagrams

Each node on a tree diagram represents a gate, and the number of gate inputs is written beside each node.

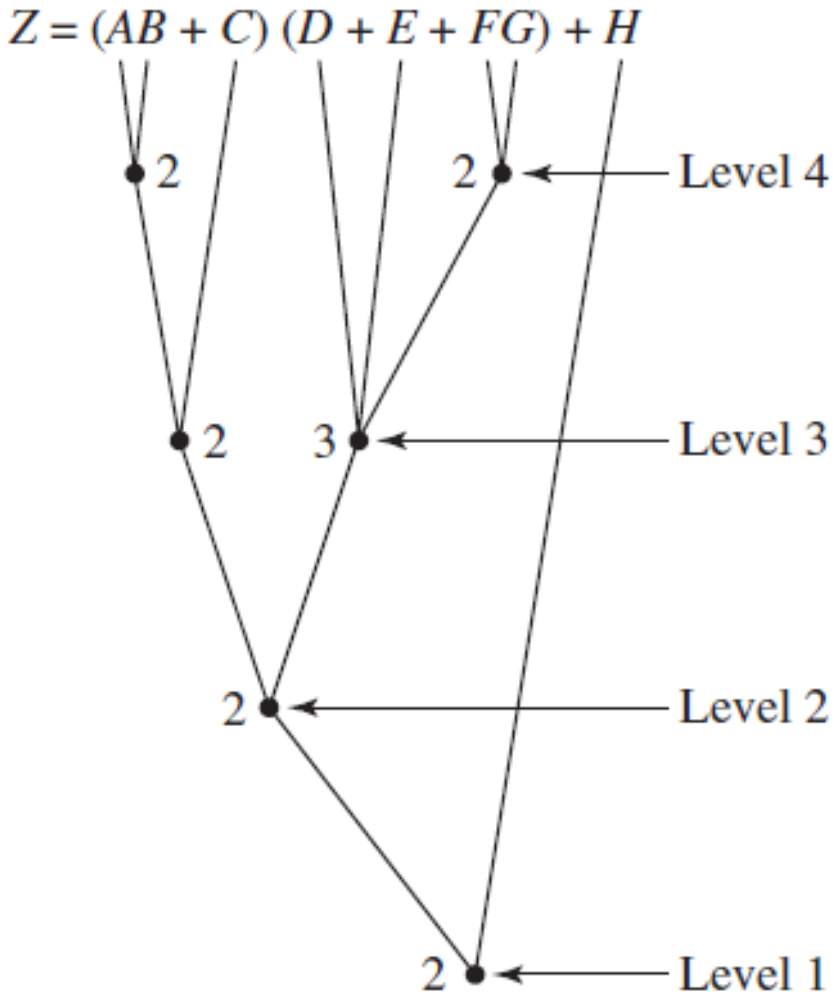


Figure 7-1: **Four-Level Realization of *Z***

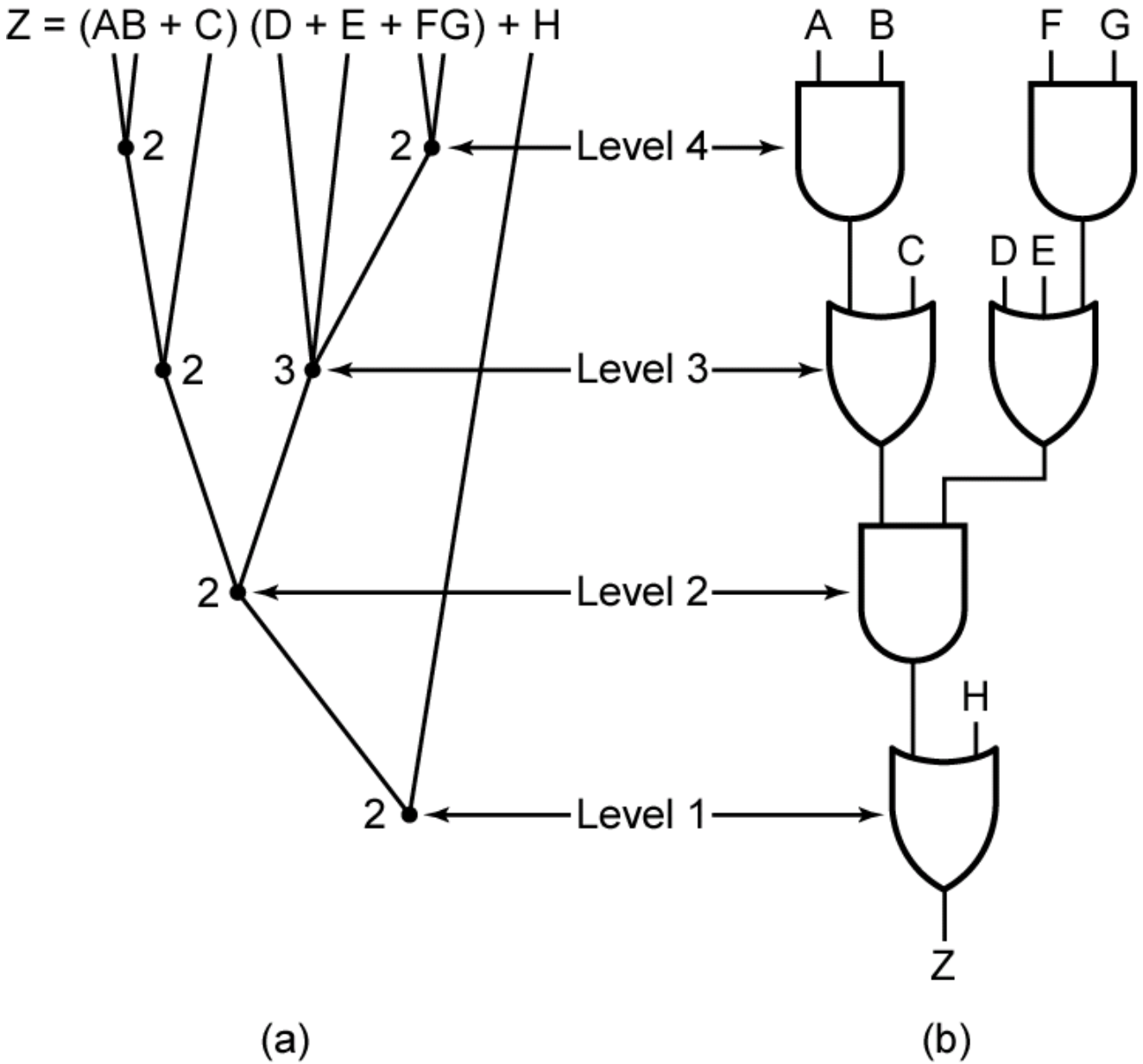


Figure 7-1: **Four-Level Realization of Z**

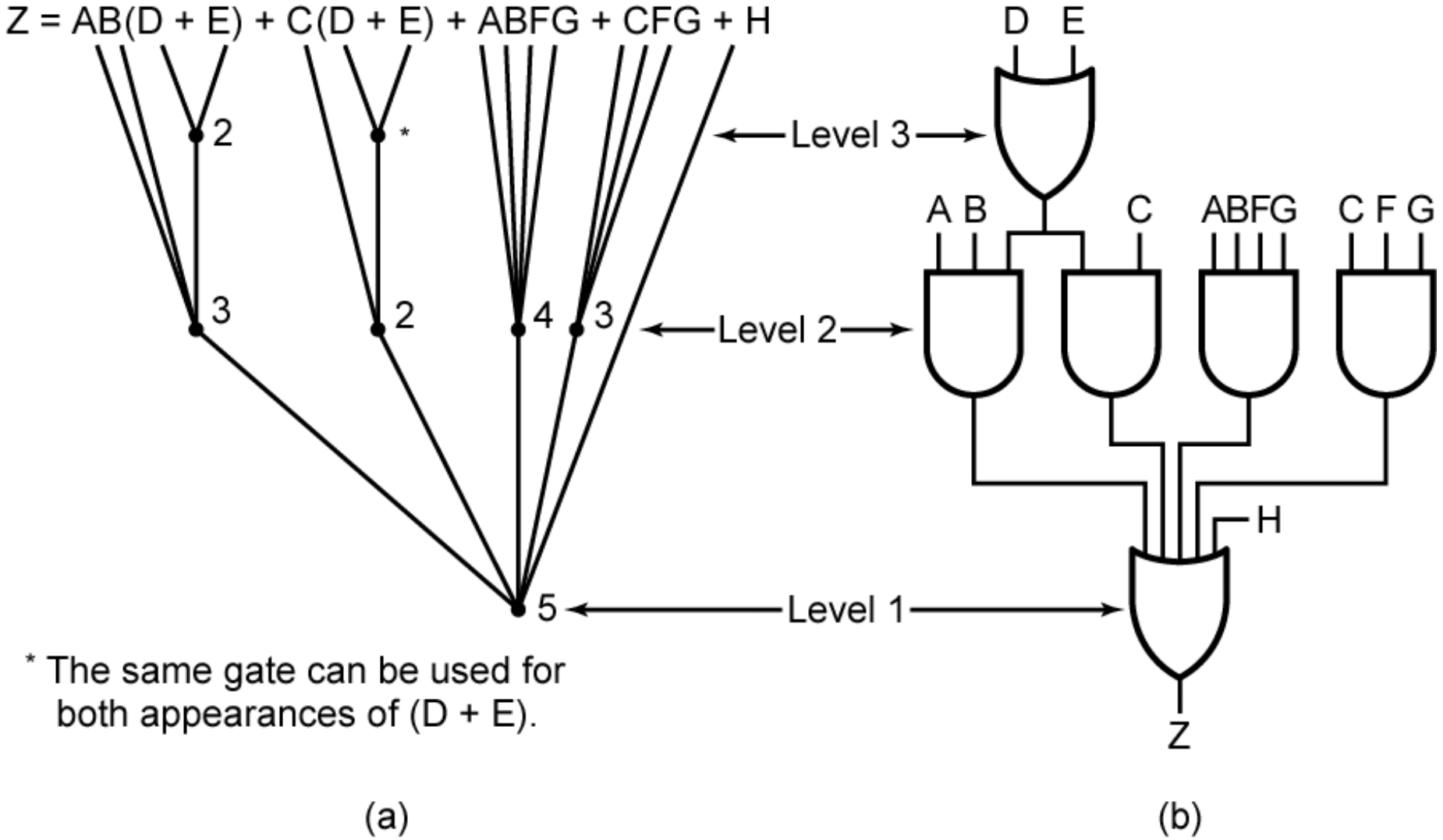


Figure 7-2: **Three-Level Realization of Z**

Example

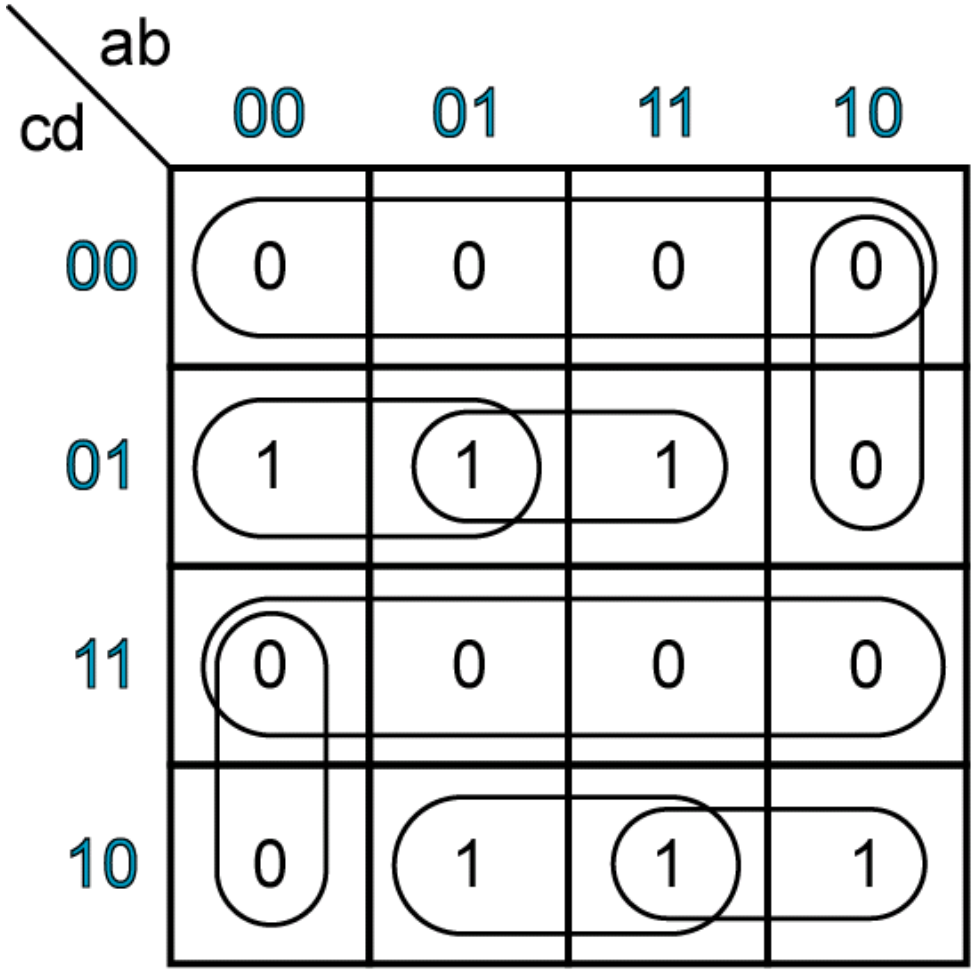
Find a circuit of AND and OR gates to realize

$f(a, b, c, d) = \Sigma\, m(1, 5, 6, 10, 13, 14)$

Consider solutions with two levels of gates and three levels of gates. Try to minimize the number of gates and the total number of gate inputs. Assume that all variables and their complements are available as inputs.

First, simplify f by using a Karnaugh map.

Section 7.1 (p. 191)



f = a'c'd + bc'd + bcd' + acd'

Figure 7-3

This leads directly to a two-level AND-OR gate circuit.

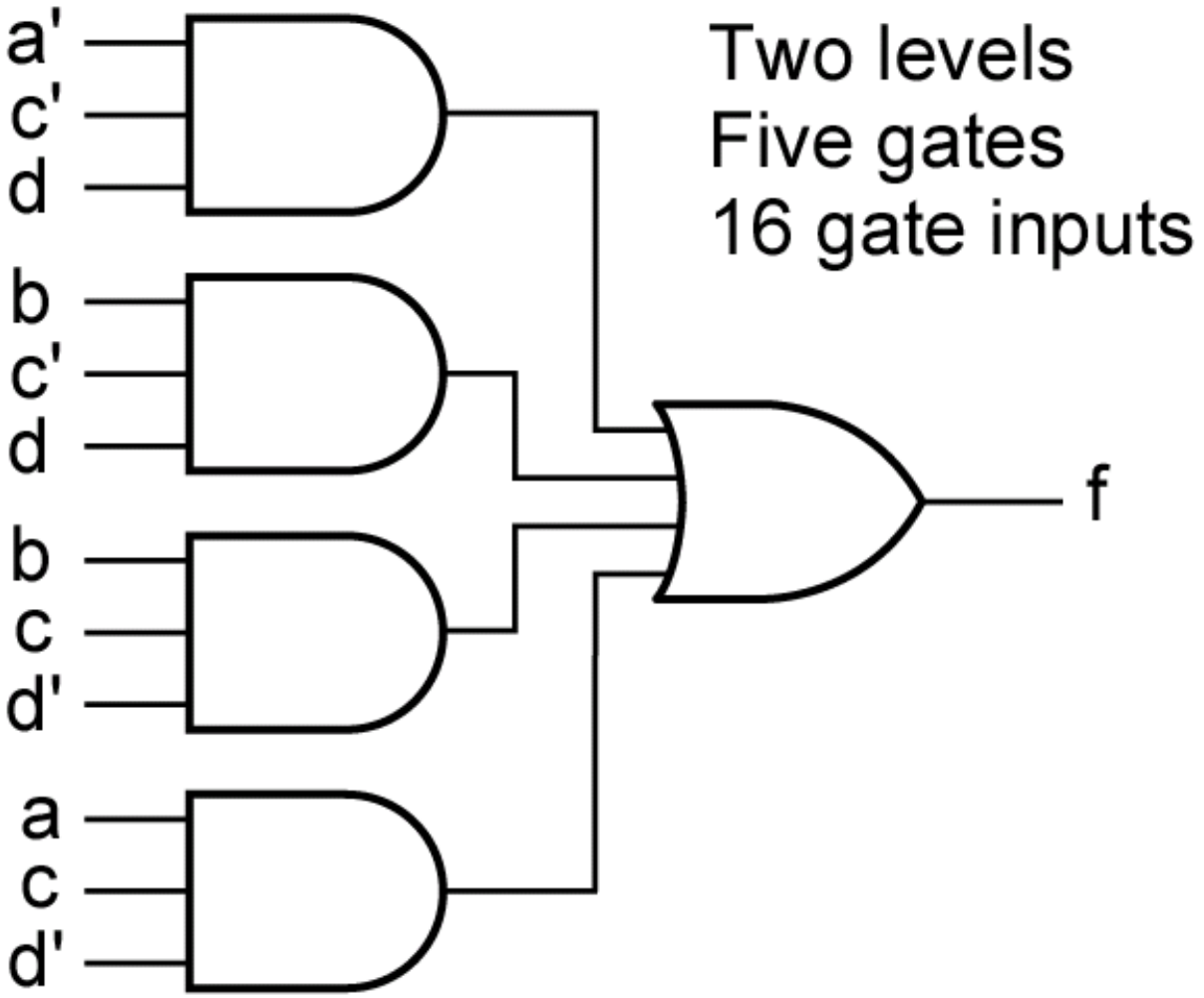


Figure 7-4

Factoring yields

$f = c'd(a' + b) + cd'(a + b)$

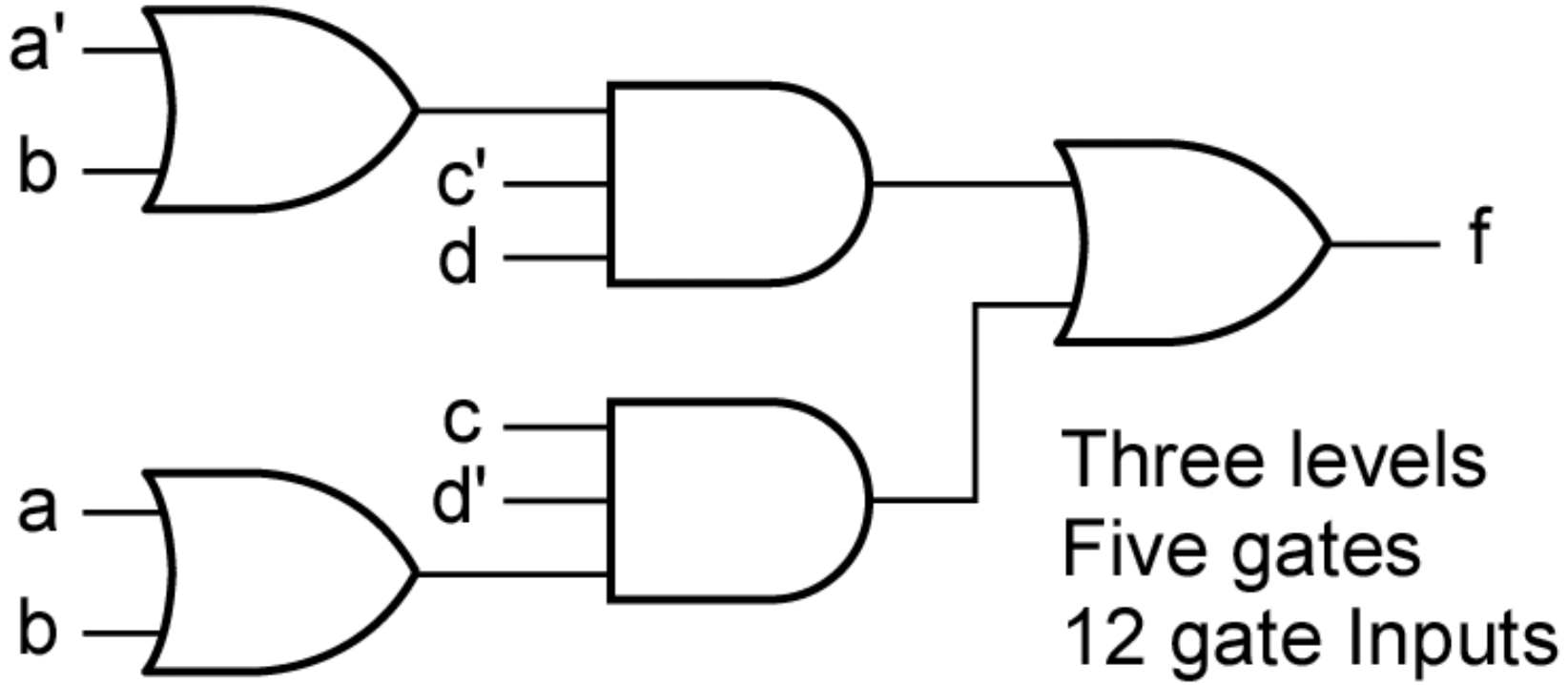


Figure 7-5

Both of these solutions have an OR gate at the output. A solution with an AND gate at the output might have fewer gates or gate inputs. A two-level OR-AND circuit corresponds to a product-of-sums expression for the function. This can be obtained from the 0′s on the Karnaugh map as follows:

$$f' = c'd + ab'c' + cd + a'b'c \qquad (7\text{-}3)$$

$$f = (c + d)(a' + b + c)(c' + d')(a + b + c') \qquad (7\text{-}4)$$

Equation (7-4) leads directly to a two-level OR-AND circuit.

Section 7.1 (p. 193)

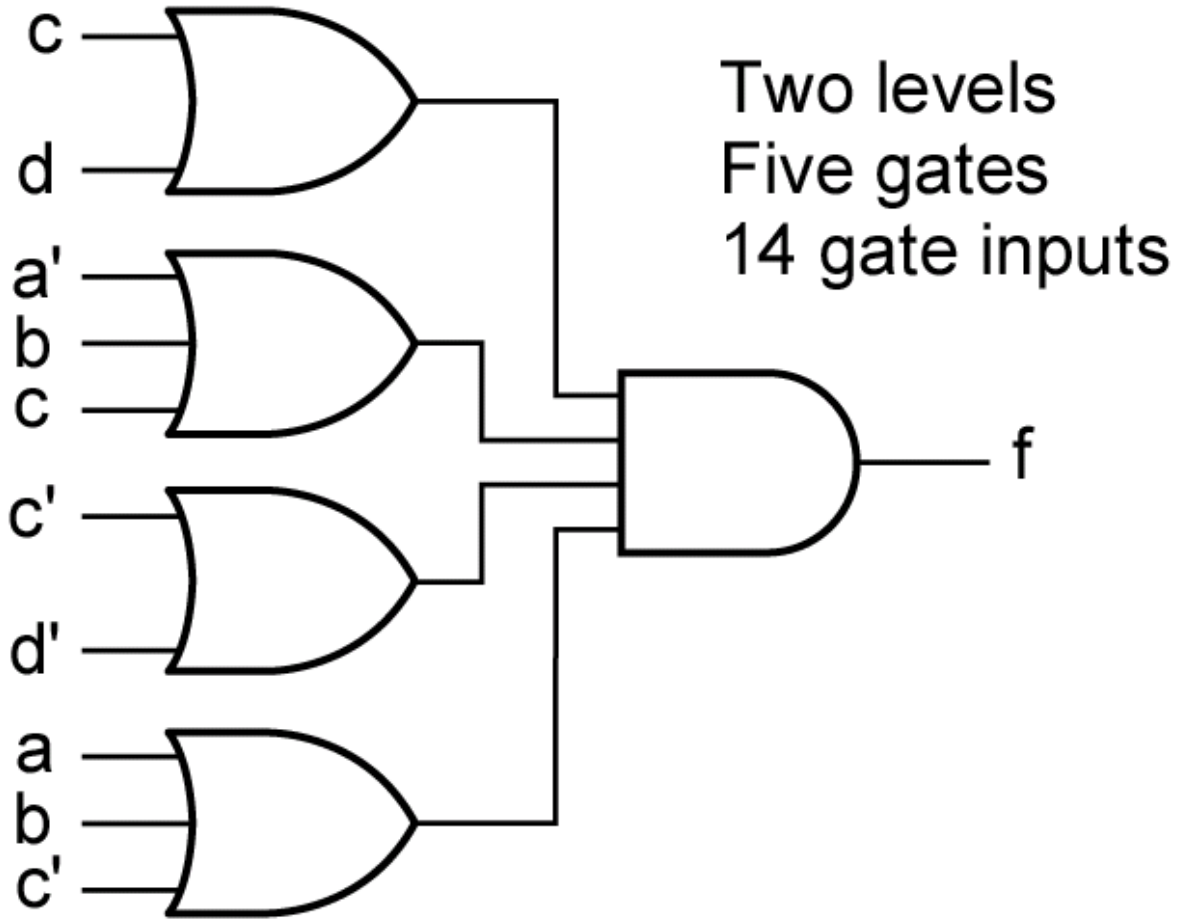


Figure 7-6

To get a three-level circuit with an AND gate output, we partially multiply out Equation (7-4) using $(X + Y)(X + Z) = X + YZ$:

$$f = [c + d(a' + b)][c' + d'(a + b)] \qquad (7\text{-}5)$$

Equation (7-5) would require four levels of gates to realize; however, if we multiply out $d'(a + b)$ and $d(a' + b)$, we get

$$f = (c + a'd + bd)(c' + ad' + bd') \qquad (7\text{-}6)$$

which leads directly to a three-level AND-OR-AND circuit.

Section 7.1 (p. 194)

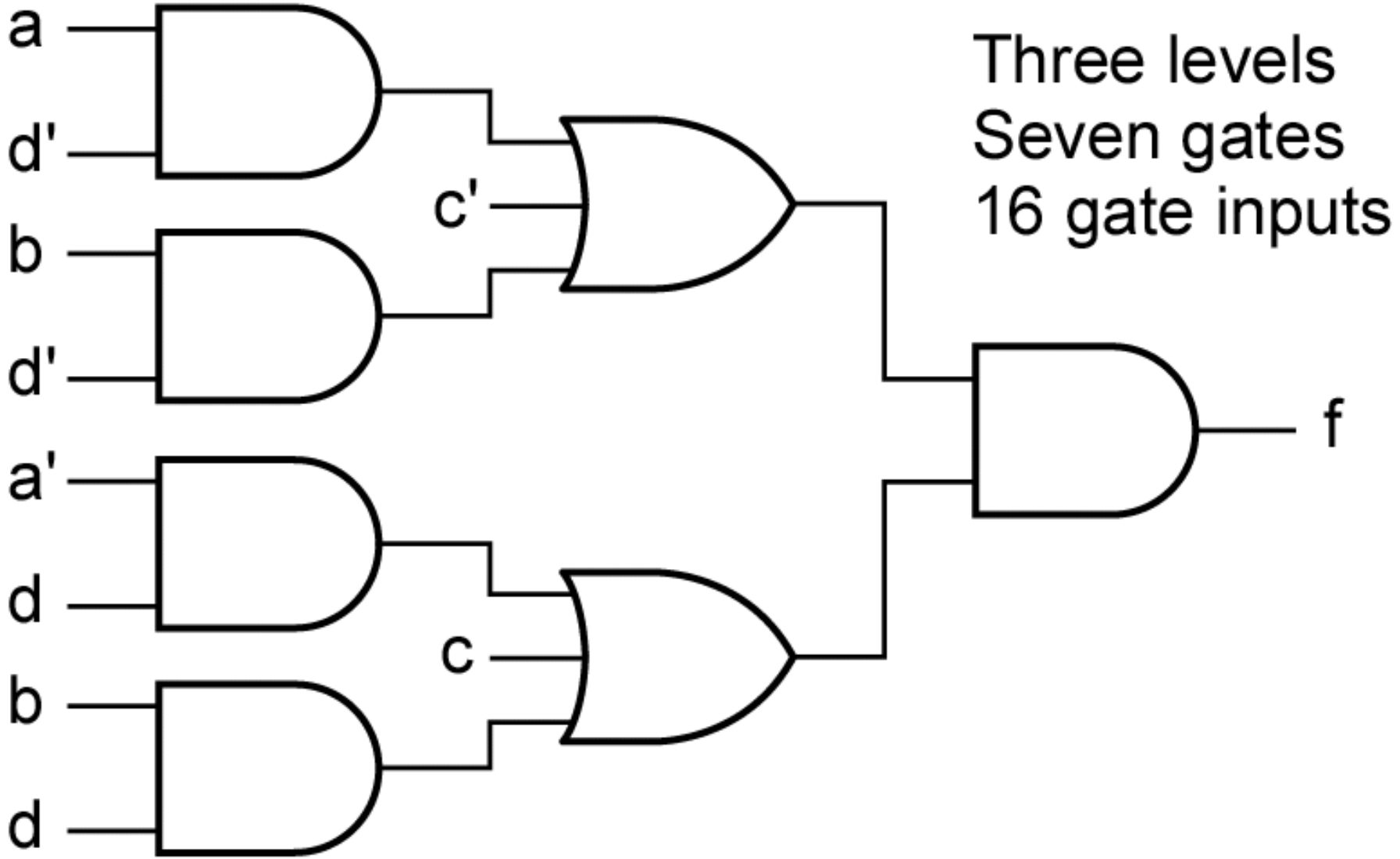


Figure 7-7

For this particular example, the best two-level solution had an AND gate at the output, and the best three-level solution had an OR gate at the output. In general, to be sure of obtaining a minimum solution, one must find *both* the circuit with the AND-gate output and the one with the OR-gate output.

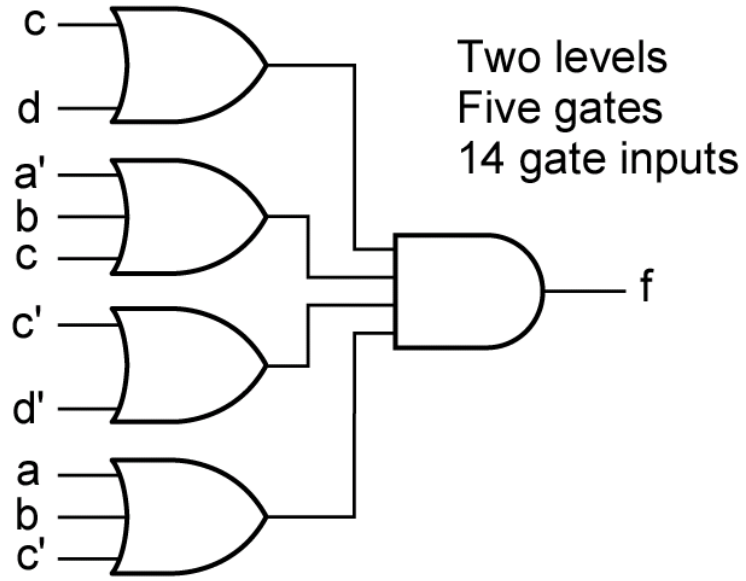


Figure 7-6

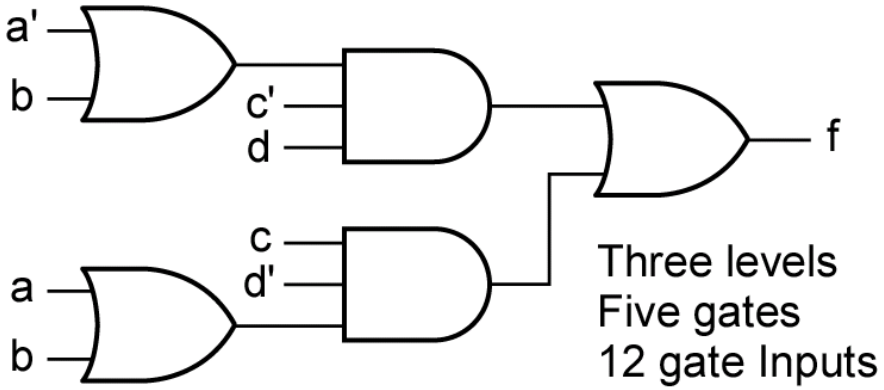


Figure 7-5

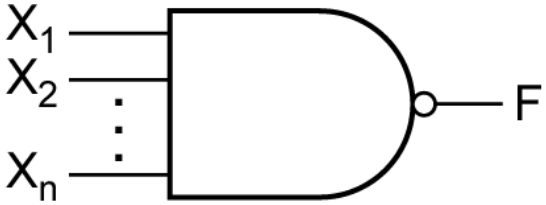
NAND gates

Figure 7-8(a) shows a three-input NAND gate. The small circle (or "bubble") at the gate output indicates inversion, so the NAND gate is equivalent to an AND gate followed by an inverter, as shown in Figure 7-8(b). The gate output is

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

(a) 3-input NAND gate

(b) NAND gate equivalent



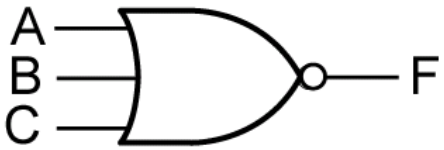
(c) n-input NAND gate

Figure 7-8: **NAND Gates**

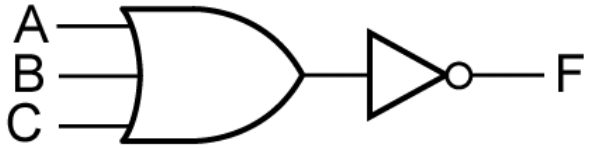
NOR gates

Figure 7-9(a) shows a three-input NOR gate. The small circle at the gate output indicates inversion, so the NOR gate is equivalent to an OR gate followed by an inverter. The gate output is

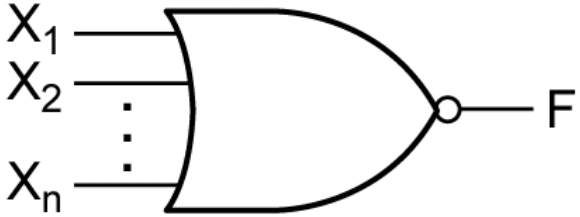
$$F = (A + B + C)' = A'B'C'$$



(a) 3-input NOR gate



(b) NOR gate equivalent

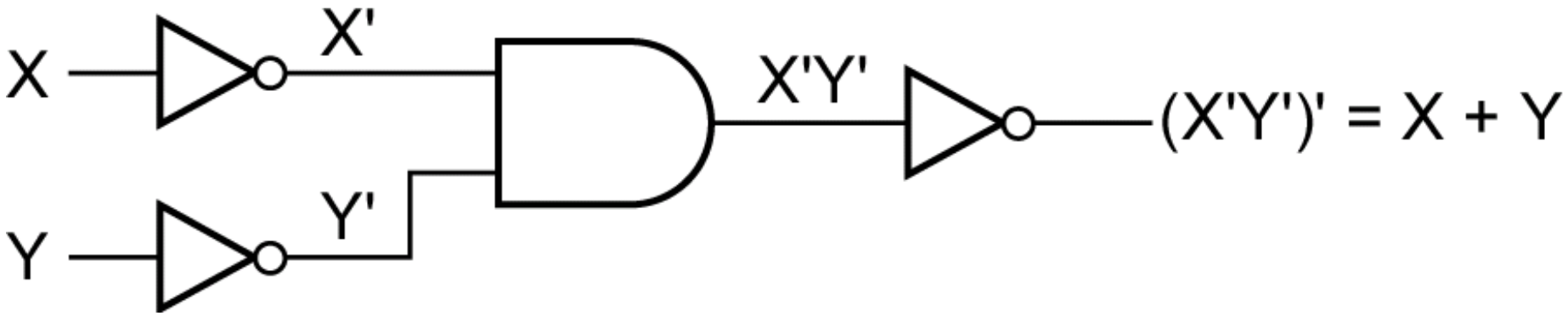


(c) n-input NOR gate

Figure 7-9: **NOR Gates**

Functionally Complete Set of Gates

AND and NOT are a functionally complete set of gates because OR can also be realized using AND and NOT:



Section 7.2 (p. 196)

NAND Gates

Similarly, any function can be realized using only NAND gates:

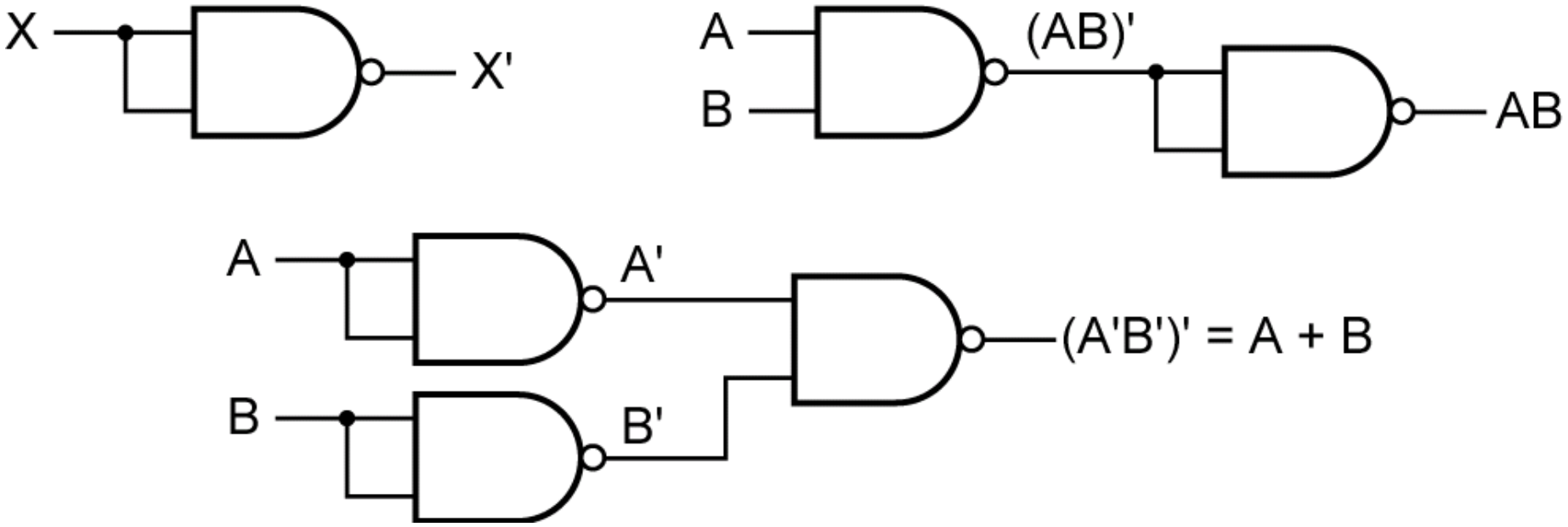


Figure 7-10: **NAND Gate Realization of NOT, AND, and OR**

Design of Two-Level NAND-Gate Circuits

A two-level circuit composed of AND and OR gates is easily converted to a circuit composed of NAND gates or NORE gates using $F = (F')'$ and then applying DeMorgan's laws:

$$(X_1 + X_2 + \ldots + X_n)' = X_1' X_2' \ldots X_n' \qquad (7\text{-}11)$$

$$(X_1 X_2 \ldots X_n)' = X_1' + X_2' + \ldots + X_n' \qquad (7\text{-}12)$$

Section 7.3 (p. 197)

Design of Two-Level NAND-Gate Circuits

The following example illustrates conversion of a minimum sum-of-products form to several other two-level forms:

$F = A + BC' + B'CD = [(A + BC' + B'CD)']'$ AND-OR

$= [A' \cdot (BC')' \cdot (B'CD)']'$ NAND-NAND

$= [A' \cdot (B' + C) \cdot (B + C' + D')]'$ OR-NAND

$= A + (B' + C)' + (B + C' + D')'$ NOR-OR

Section 7.3 (p. 197)

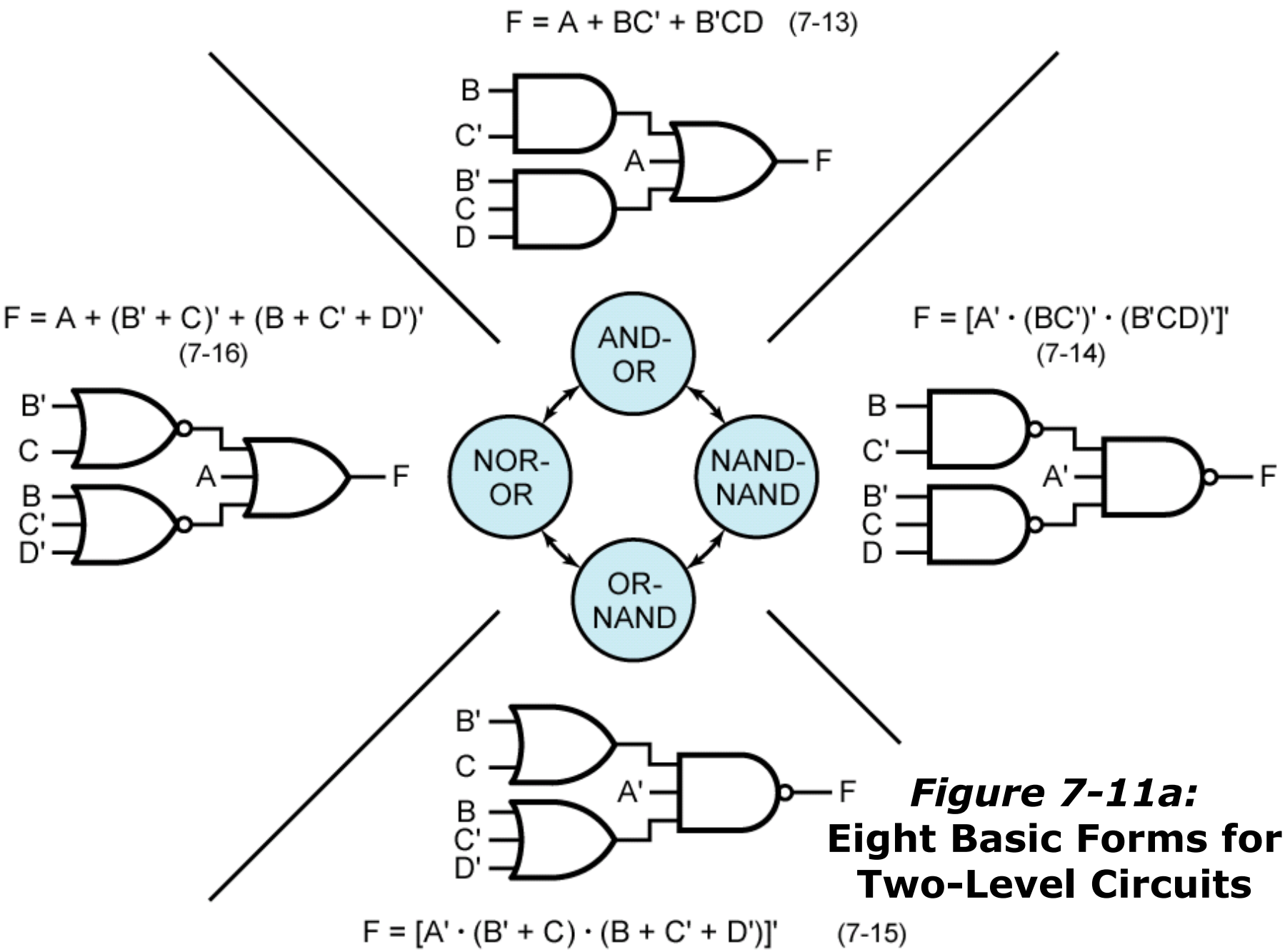


Figure 7-11a:
Eight Basic Forms for Two-Level Circuits

Design of Two-Level NOR-Gate Circuits

If we want a two-level circuit containing only NOR gates, we should start with the minimum product-of-sums form for *F* instead of the minimum sum-of-products. After obtaining the minimum product-of-sums from a Karnaugh map, *F* can be written in the following two-level forms:

$F = (A + B + C)(A + B' + C')(A + C' + D)$ OR-AND

$= \{[(A + B + C)(A + B' + C')(A + C' + D)]' \}'$

$= [(A + B + C)' + (A + B' + C')' + (A + C' + D)']'$ NOR-NOR

$= (A'B'C' + A'BC + A'CD')'$ AND-NOR

$= (A'B'C')' \cdot (A'BC)' \cdot (A'CD')'$ NAND-AND

Section 7.3 (p. 197)

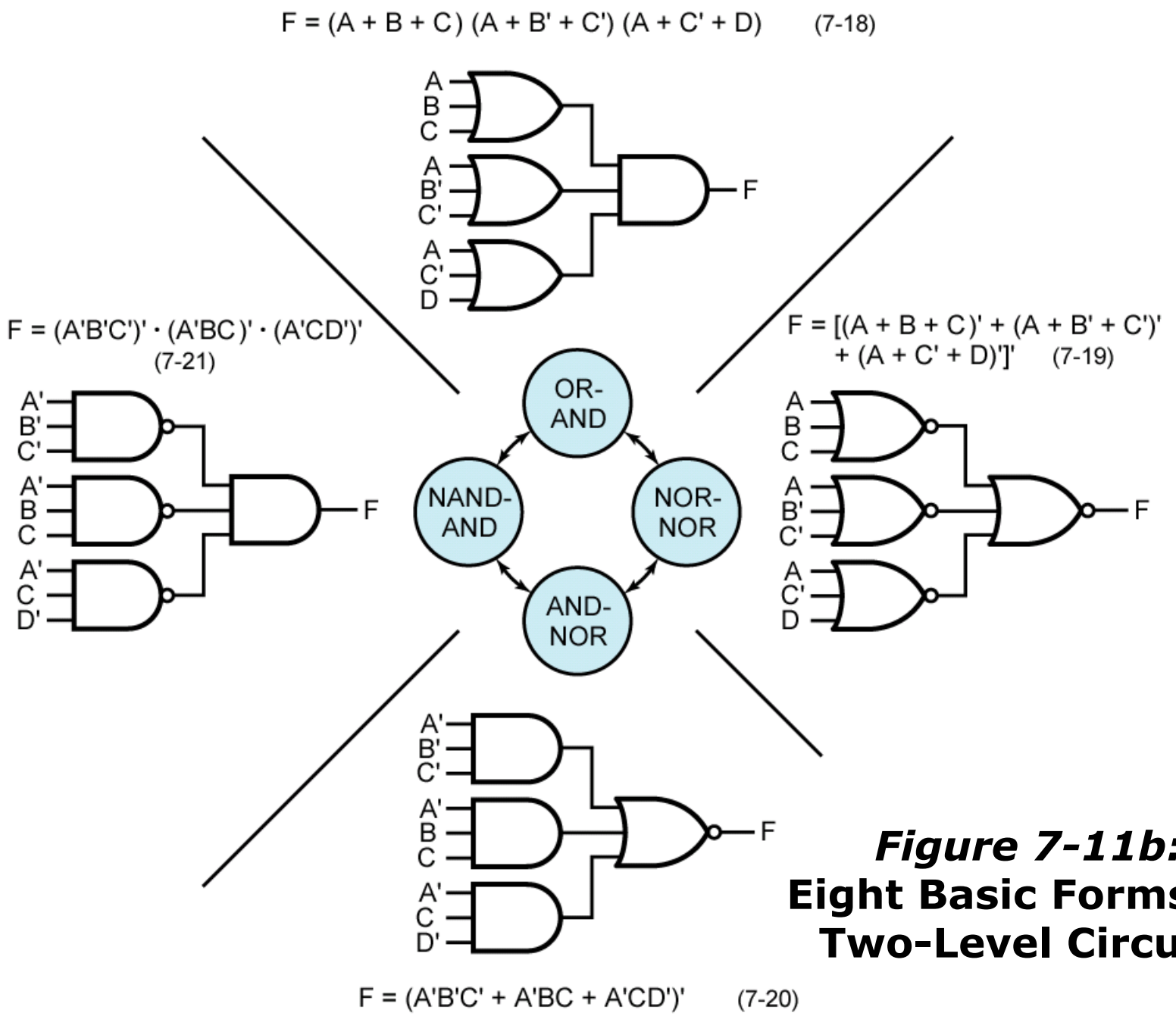
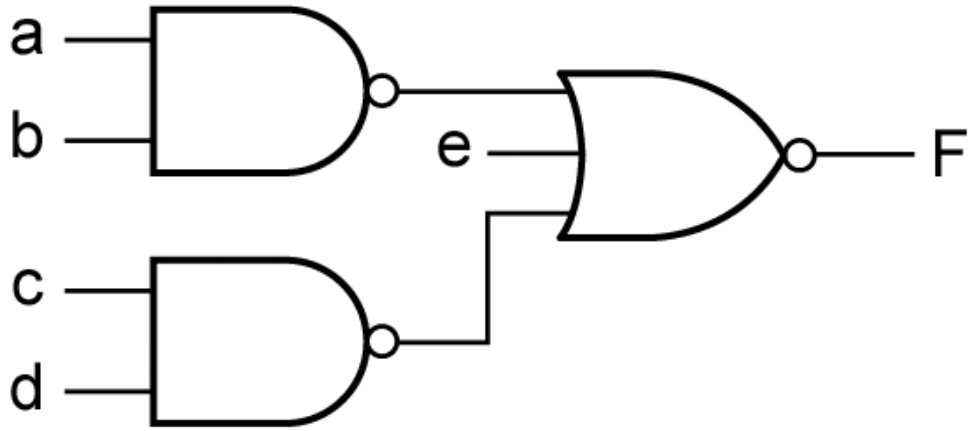


Figure 7-11b:
Eight Basic Forms for Two-Level Circuits

The other eight possible two-level forms are degenerate in the sense that they cannot realize all switching functions. Consider, for example, the following NAND-NOR circuit:



$$F = [(ab)' + (cd)' + e]' = abcde'$$

From this example, it is clear that the NAND-NOR form can realize only a product of literals and not a sum of products.

Section 7.3 (p. 199)

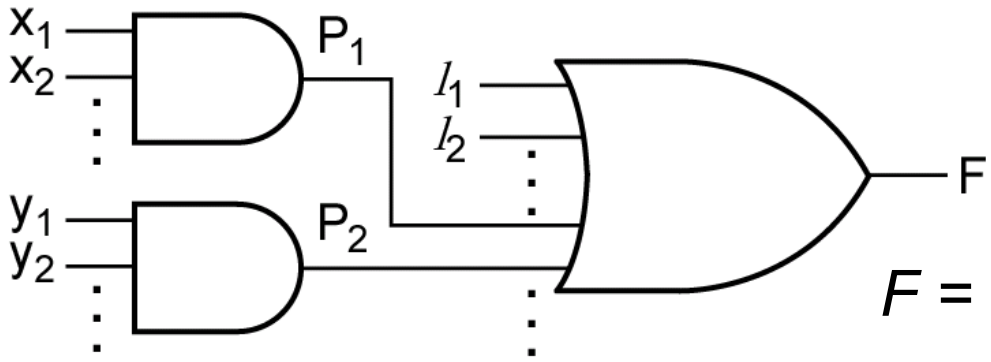
Design of Minimum Two-Level NAND-NAND Circuits

Procedure for designing a minimum two-level NAND-NAND circuit:

1. Find a minimum *sum-of-products* expression for F.
2. Draw the corresponding two-level AND-OR circuit.
3. Replace all gates with NAND gates leaving the gate interconnection unchanged. If the output gate has any single literals as inputs, complement these literals.

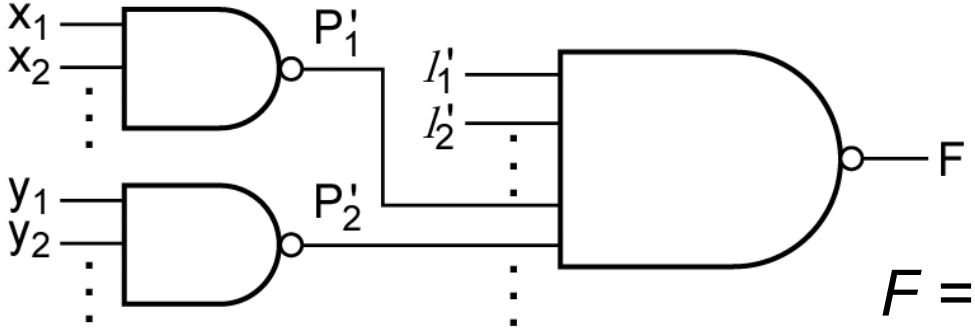
Figure 7-12: **AND-OR to NAND-NAND Transformation**

Design of Minimum Two-Level NAND-NAND Circuit



$$F = l_1 + l_2 + \cdots + P_1 + P_2 + \cdots$$

(a) Before transformation



$$F = (l_1' l_2' \cdots P_1' P_2' \cdots)'$$

(b) After transformation

Figure 7-12: **AND-OR to NAND-NAND Transformation**

Design of Multi-Level NAND- and NOR-Gate Circuits

The following procedure may be used to design multi-level NAND-gate circuits:

1. Simplify the switching function to be realized.
2. Design a multi-level circuit of AND and OR gates. The output gate must be OR. AND gate outputs cannot be used as AND-gate inputs; OR-gate outputs cannot be used as OR-gate inputs.
3. Number the levels starting with the output gate as level 1. Replace all gates with NAND gates, leaving all interconnections between gates unchanged, leave the inputs to levels 2,4,6,… unchanged. Invert any literals which appear as inputs to levels 1,3,5,…

Section 7.4 (p. 200)

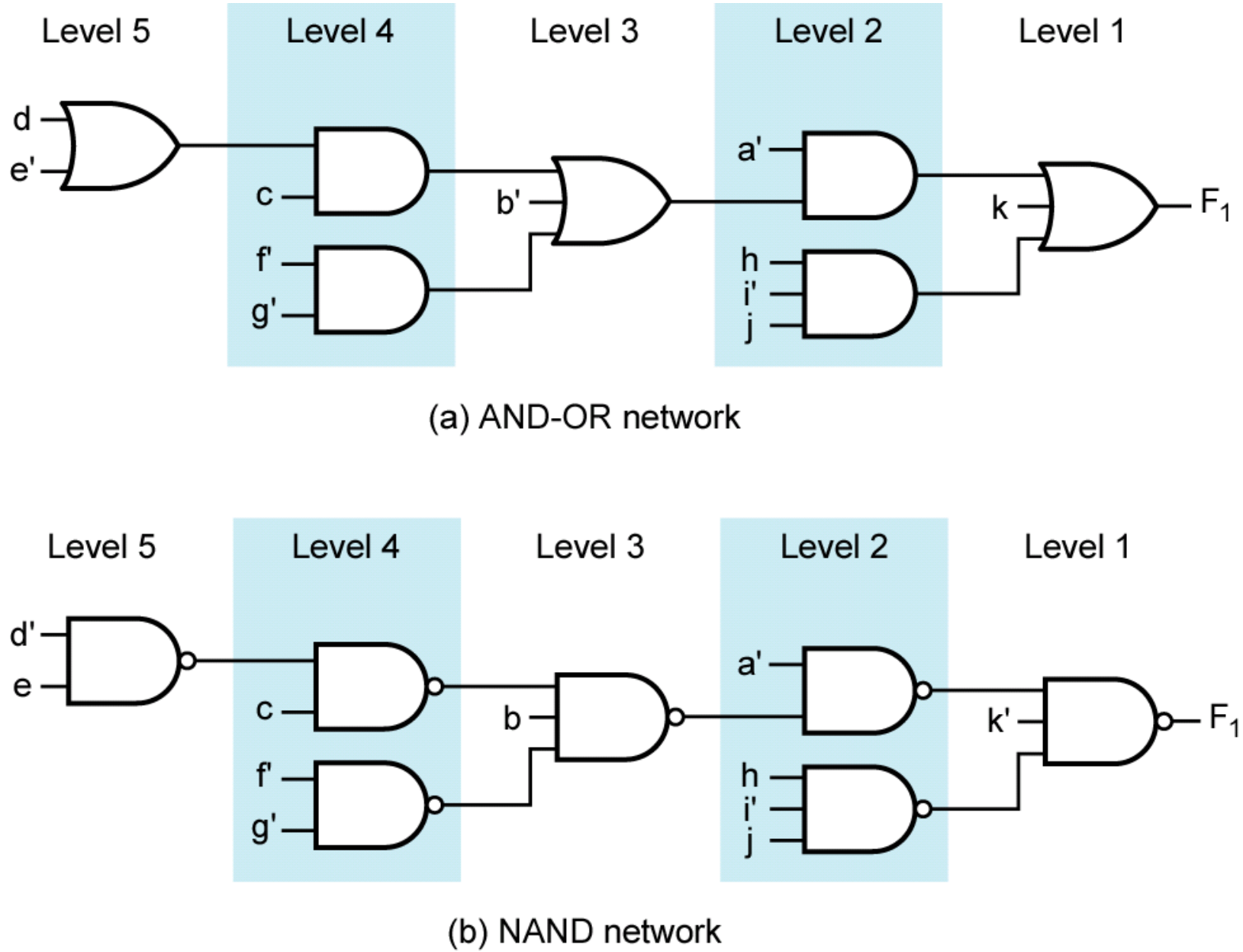
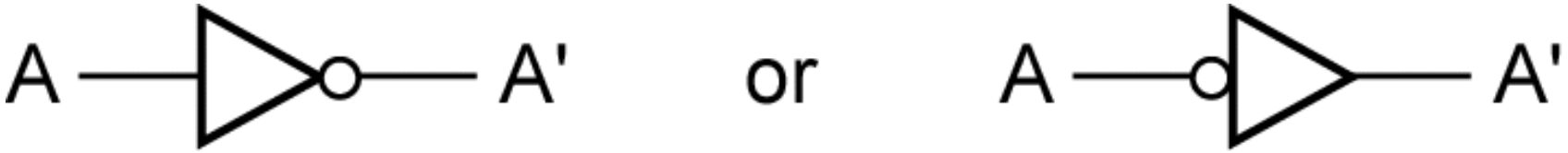


Figure 7-13: **Multi-Level Circuit Conversion to NAND Gates**

Alternative Gate Symbols

Logic designers who design complex digital systems often find it convenient to use more than one representation for a given type of gate. For example, an inverter can be represented by



Section 7.5 (p. 201)

Equivalent gate symbols based on DeMorgan's Laws

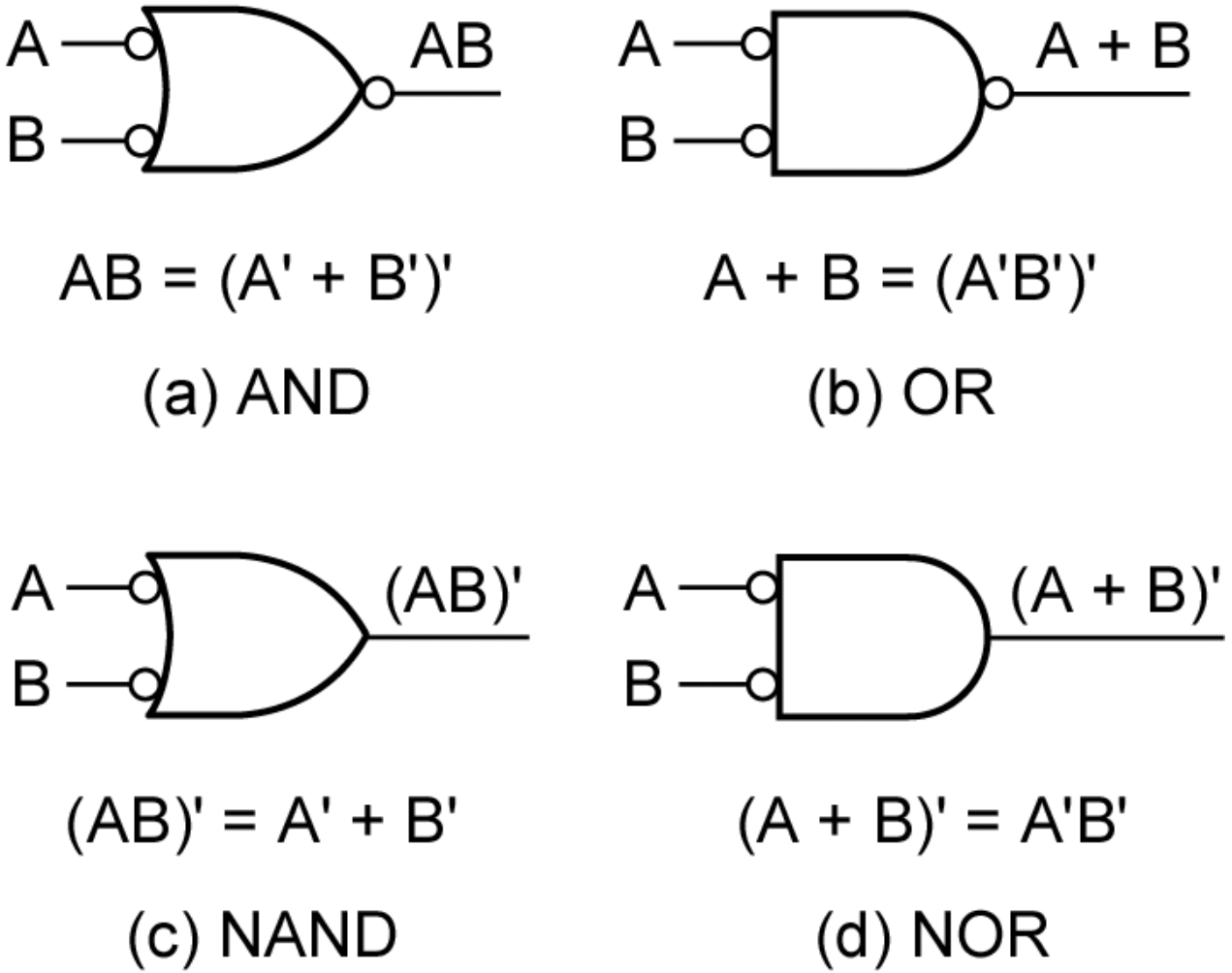
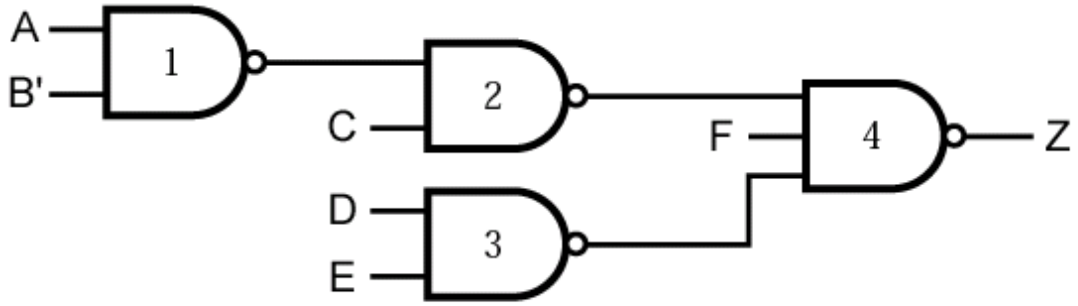
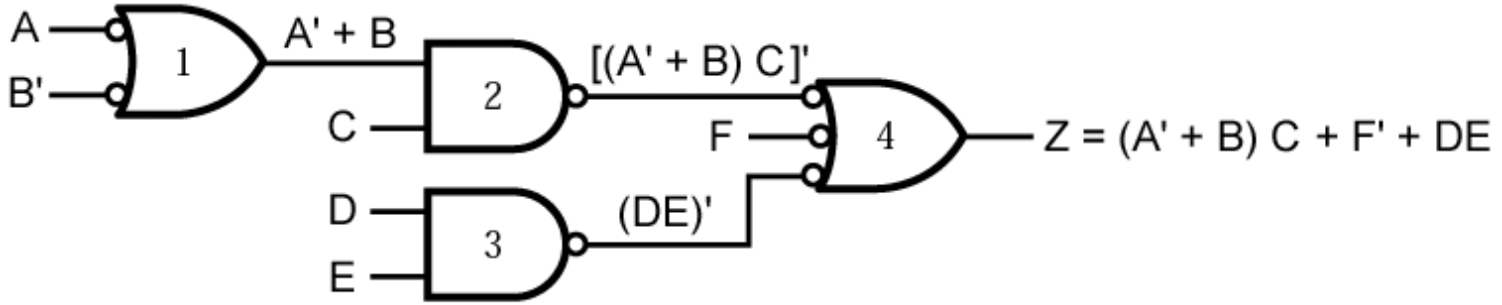


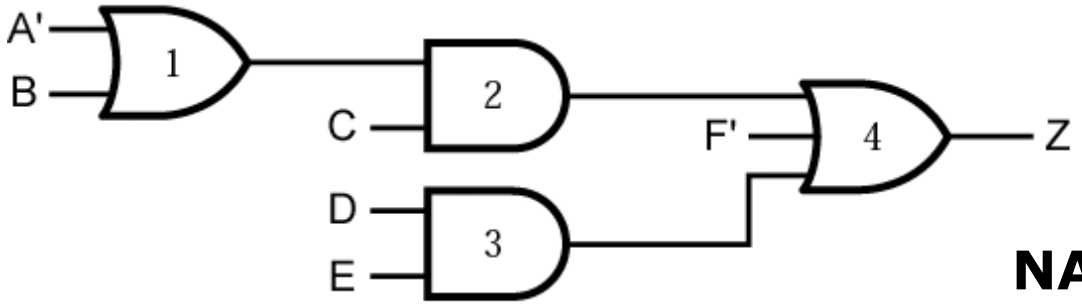
Figure 7-14: **Alternative Gate Symbols**



(a) NAND gate network

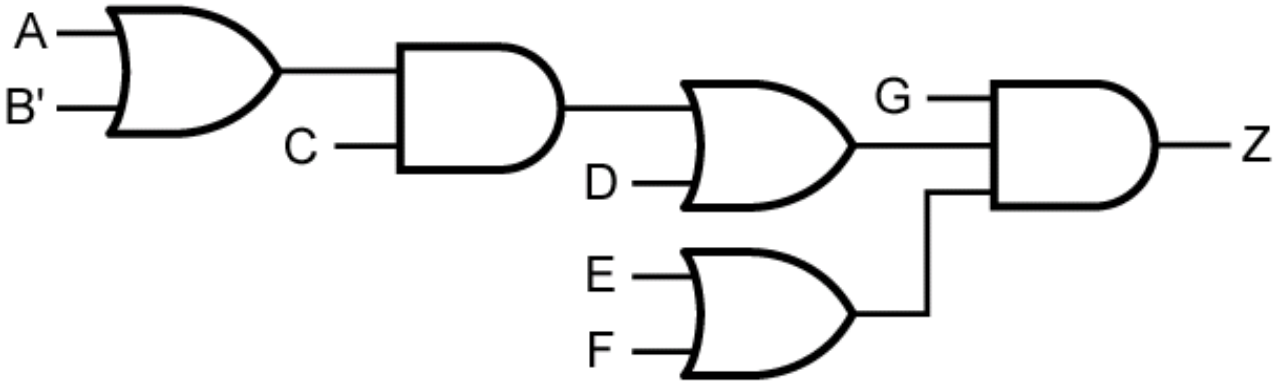


(b) Alternate form for NAND gate network



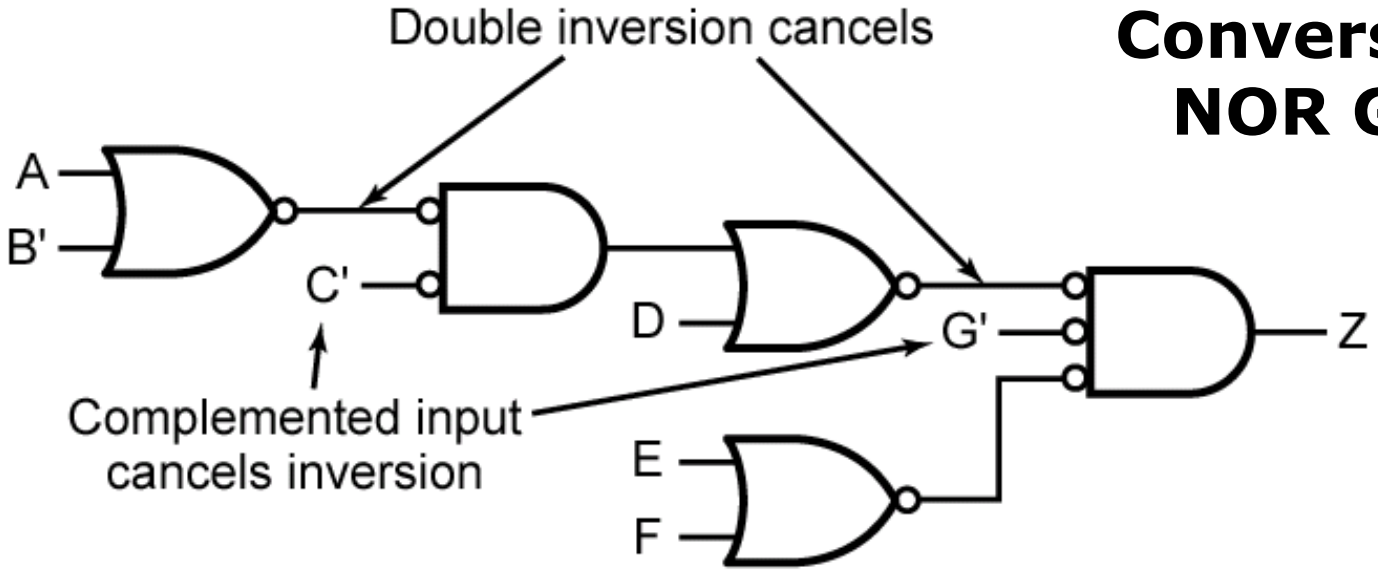
(c) Equivalent AND-OR network

Figure 7-15:
NAND Gate Circuit Conversion



(a) Circuit with OR and AND gates

Figure 7-16:
Conversion to NOR Gates



(b) Equivalent circuit with NOR gates

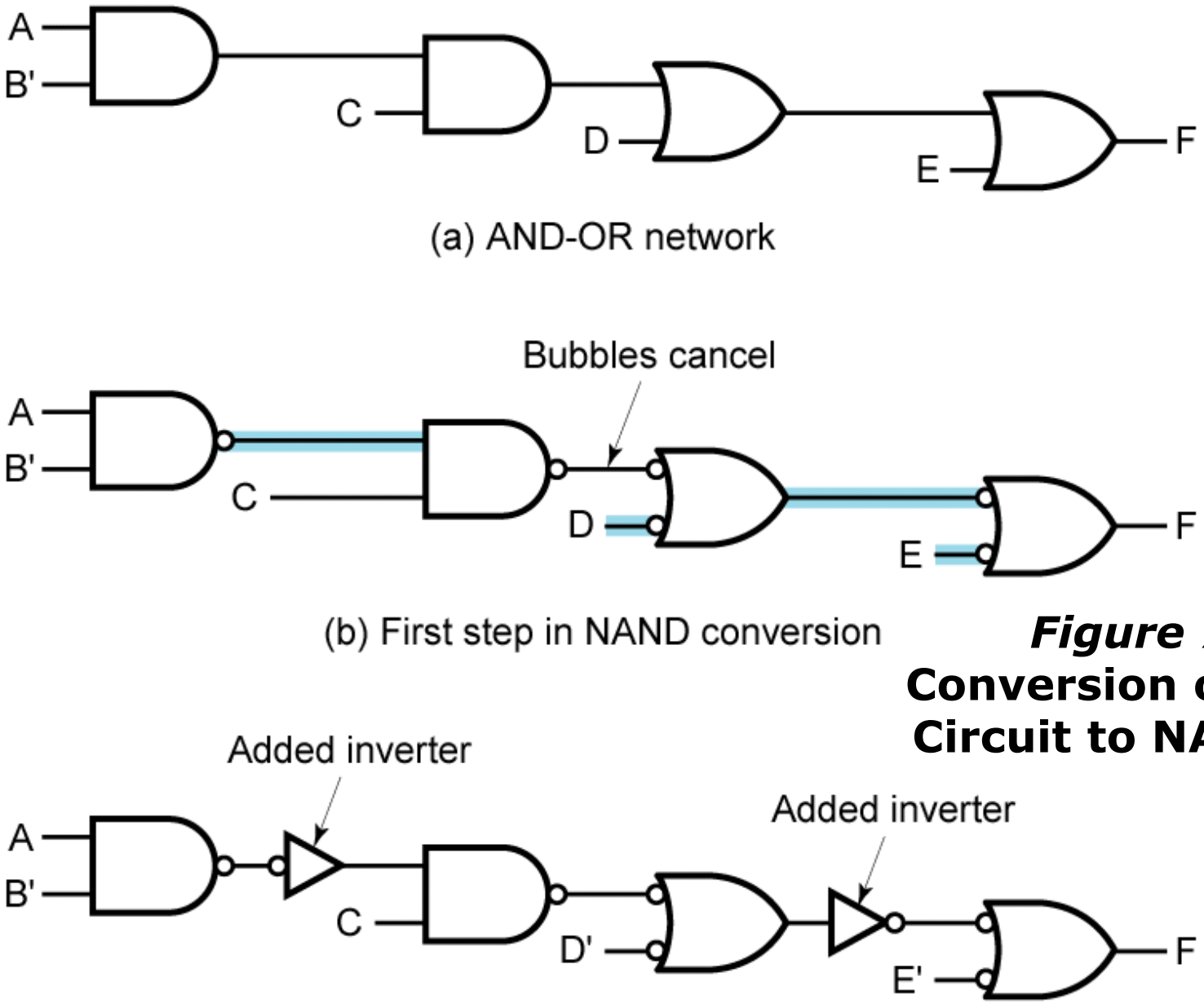


Figure 7-17:
Conversion of AND-OR Circuit to NAND Gates

Design of Two-Level, Multiple-Output Circuits

Solution of digital design problems often requires the realization of several functions of the same variables. Although each function could be realized separately, the use of some gates in common between two or more functions sometimes leads to a more economical realization.

Example:

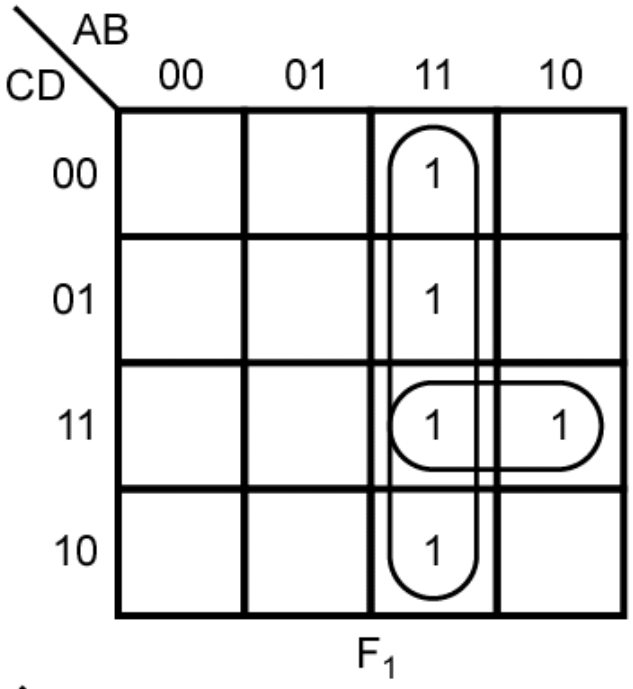
Design a circuit with four inputs and three outputs which realizes the functions

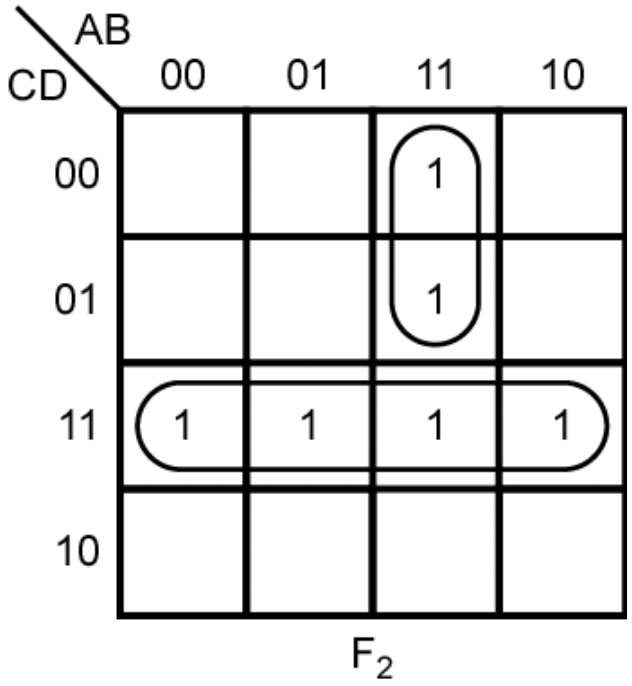
$$F_1(A, B, C, D) = \Sigma\, m(11, 12, 13, 14, 15)$$

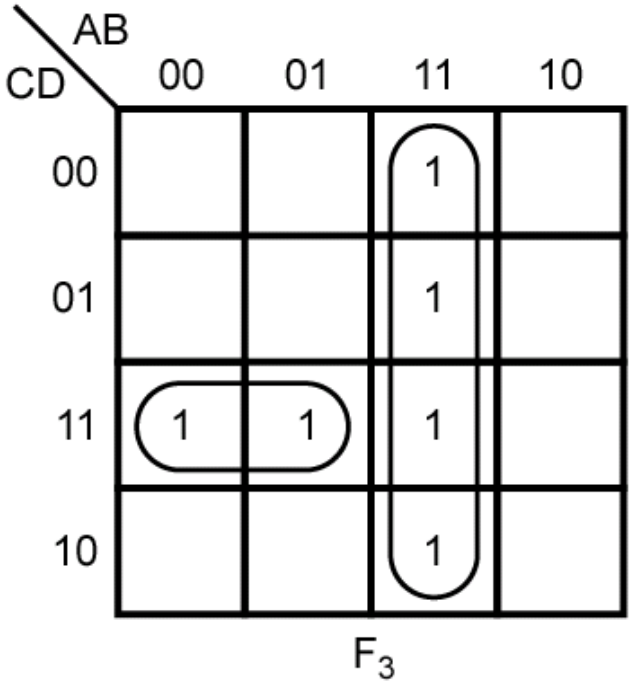
$$F_2(A, B, C, D) = \Sigma\, m(3, 7, 11, 12, 13, 15)$$

$$F_3(A, B, C, D) = \Sigma\, m(3, 7, 12, 13, 14, 15) \qquad (7\text{-}22)$$

Section 7.6 (p. 204)





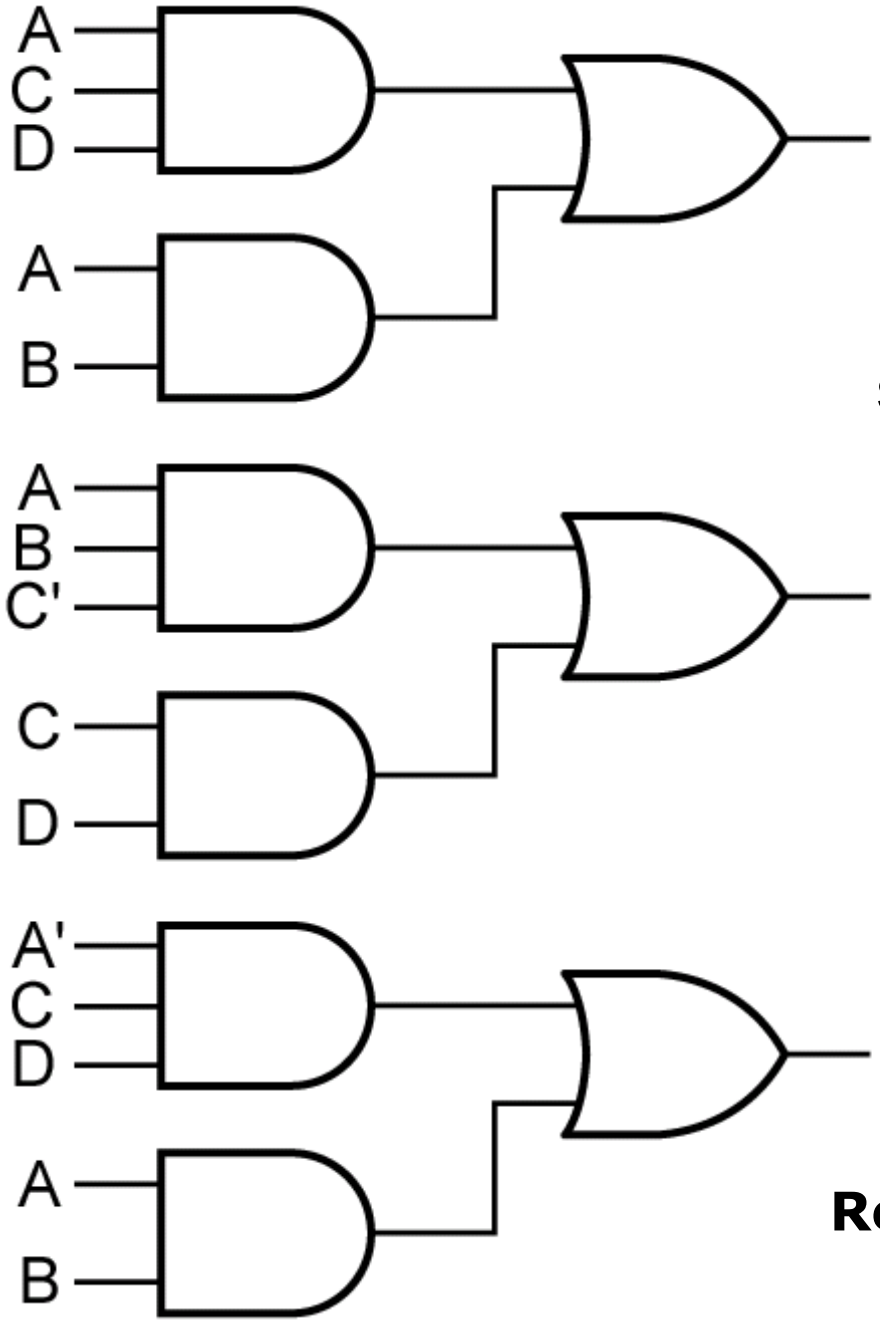


$F_1(A, B, C, D) = \Sigma\, m(11, 12, 13, 14, 15)$

$F_2(A, B, C, D) = \Sigma\, m(3, 7, 11, 12, 13, 15)$

$F_3(A, B, C, D) = \Sigma\, m(3, 7, 12, 13, 14, 15)$

Figure 7-18: **Karnaugh Maps for Equations (7-22)**



Realization of functions separately (9 Gates)

Figure 7-19:
Realization of Equations (7-22)

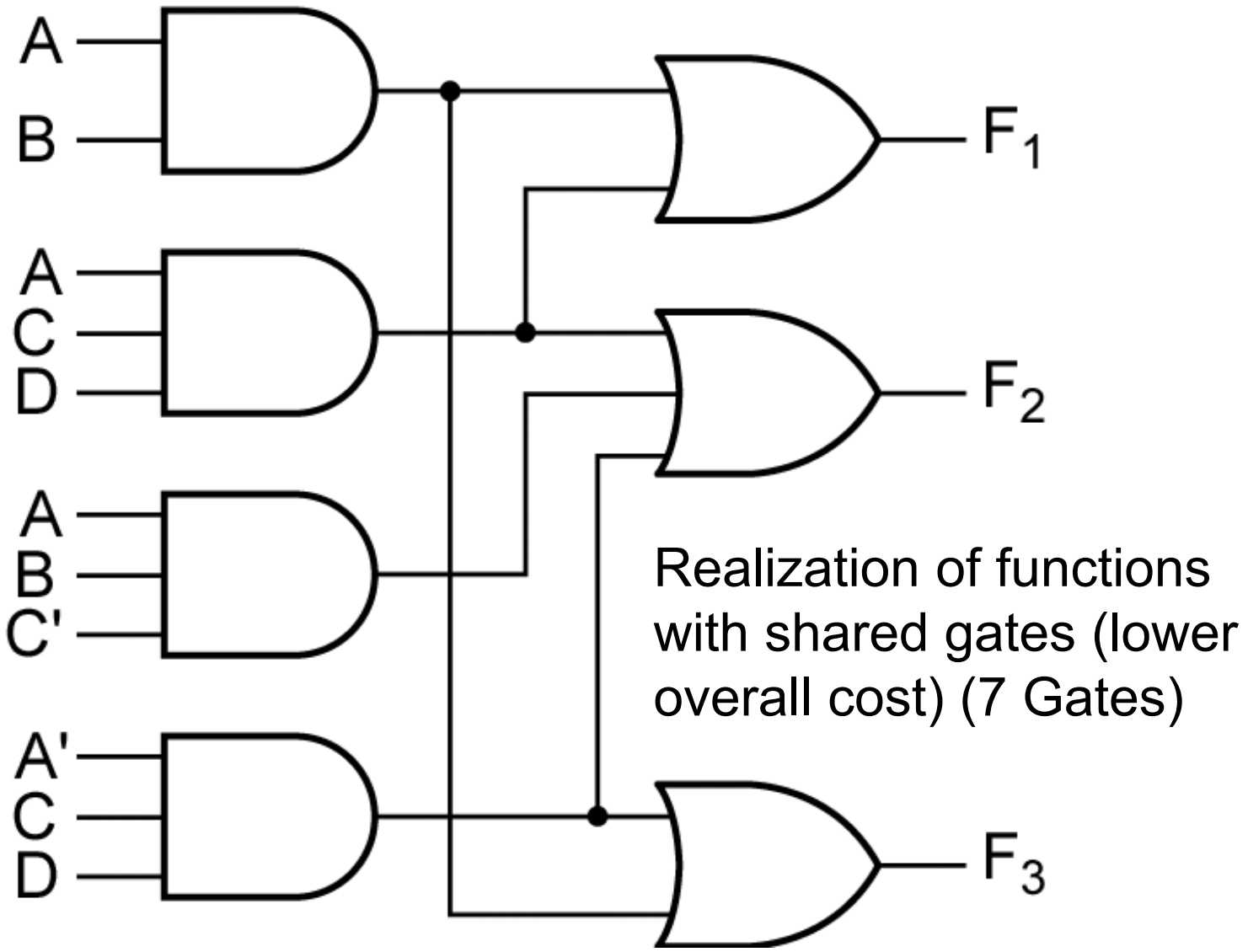


Figure 7-20: **Multiple-Output Realization of Equations (7-22)**

Another example of sharing gates among multiple outputs to reduce cost.

$f_1 = \Sigma\, m(2, 3, 5, 7, 8, 9, 10, 11, 13, 15)$

$f_2 = \Sigma\, m(2, 3, 5, 6, 7, 10, 11, 14, 15)$

$f_3 = \Sigma\, m(6, 7, 8, 9, 13, 14, 15)$

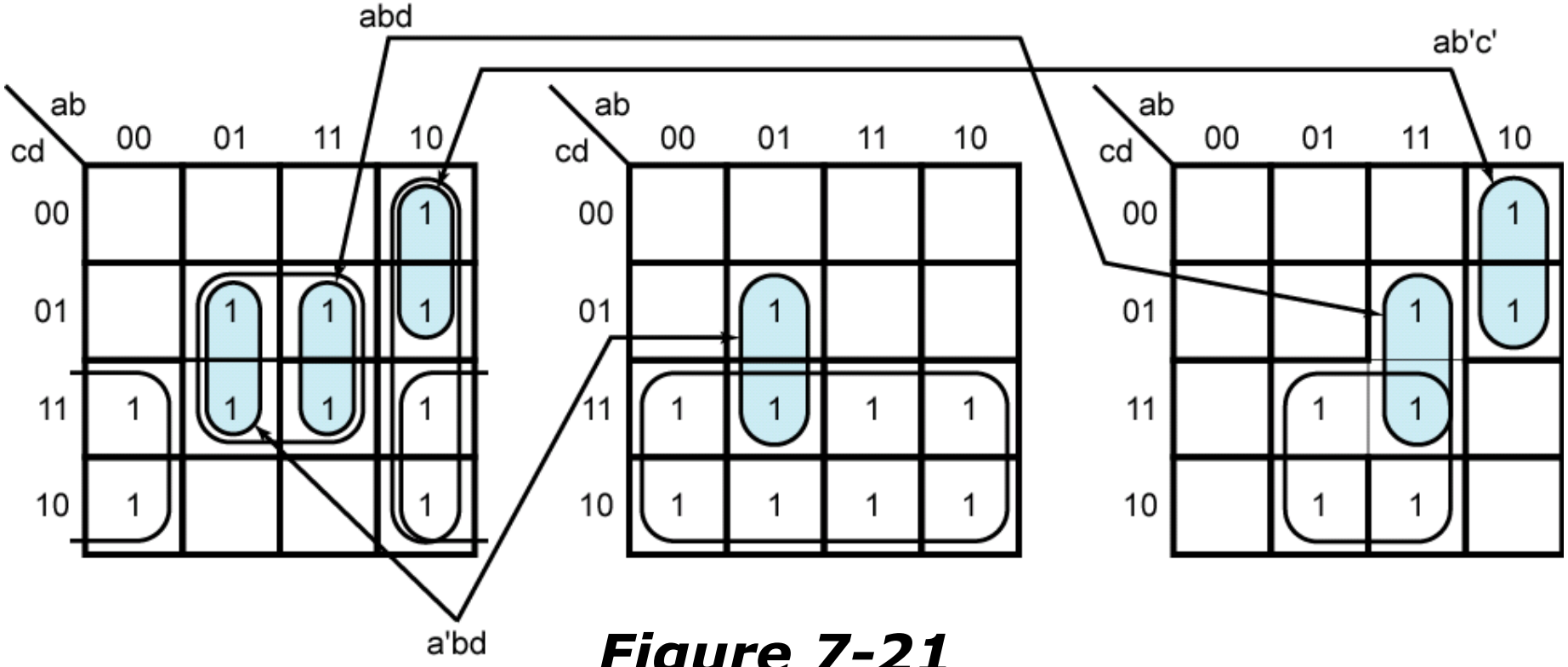
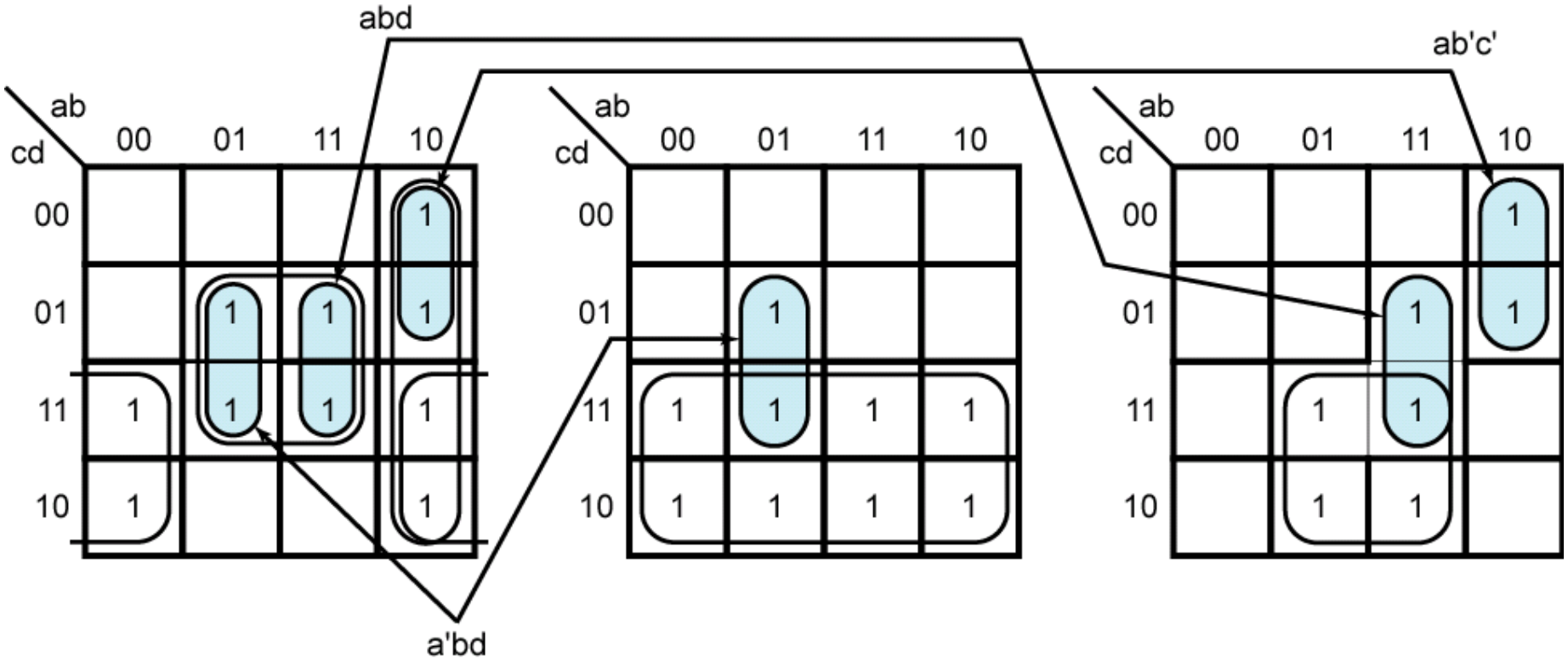


Figure 7-21

Minimal Solution



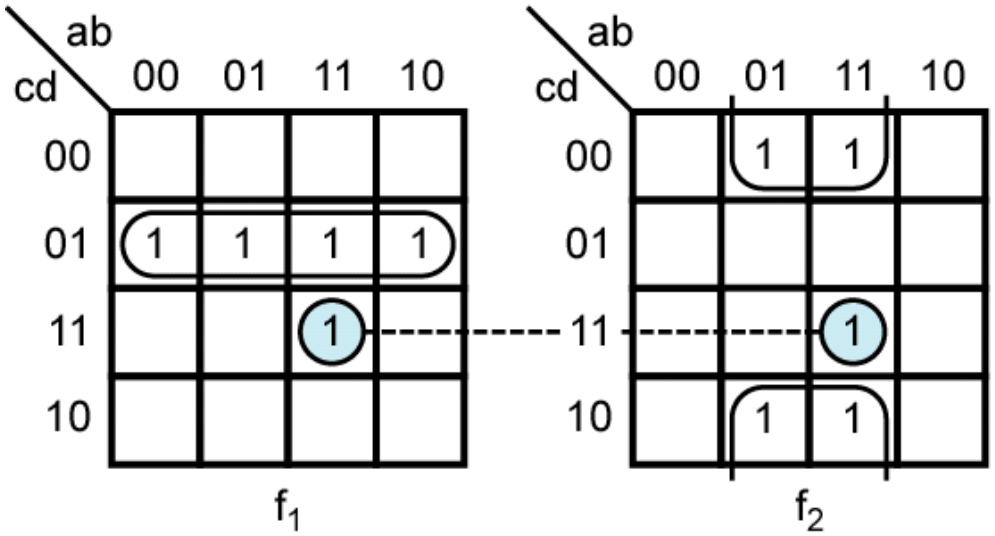
$f_1 = \underline{a'bd} + \underline{abd} + \underline{ab'c'} + b'c$

$f_2 = c + \underline{a'bd}$

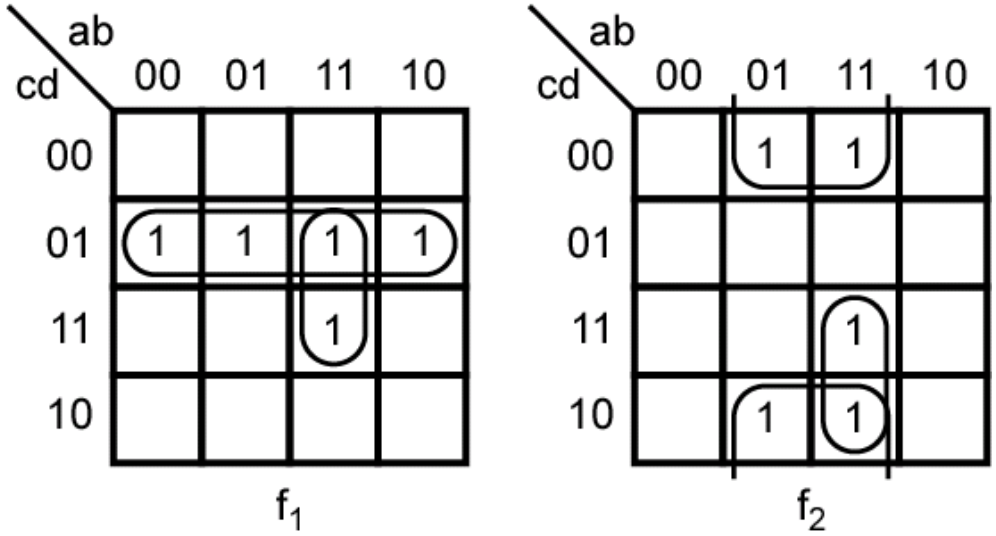
$f_3 = bc + \underline{ab'c'} + \underline{abd}$

eight gates
22 gate inputs

In this example, the best solution is obtained by not combining the circled 1 with adjacent 1's.



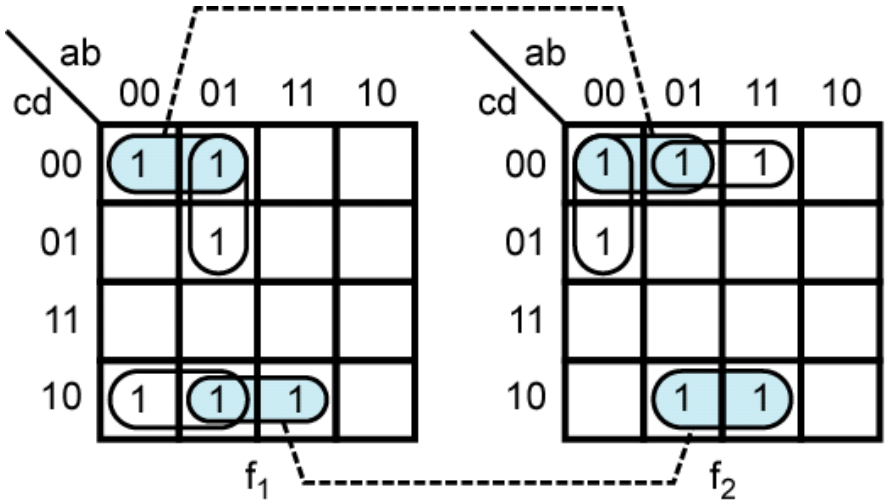
(a) Best solution



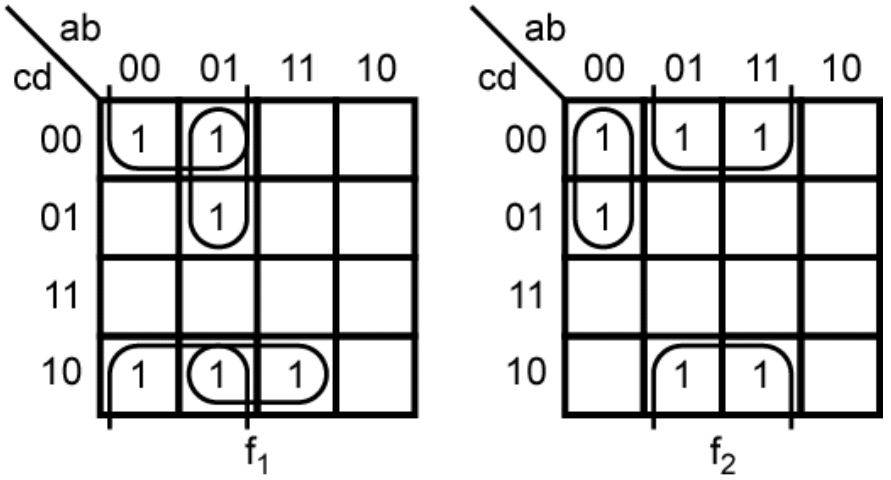
(b) Solution requires an extra gate

Figure 7-22

The solution with the maximum number of common terms is not necessarily the best solution, as illustrated by this example.

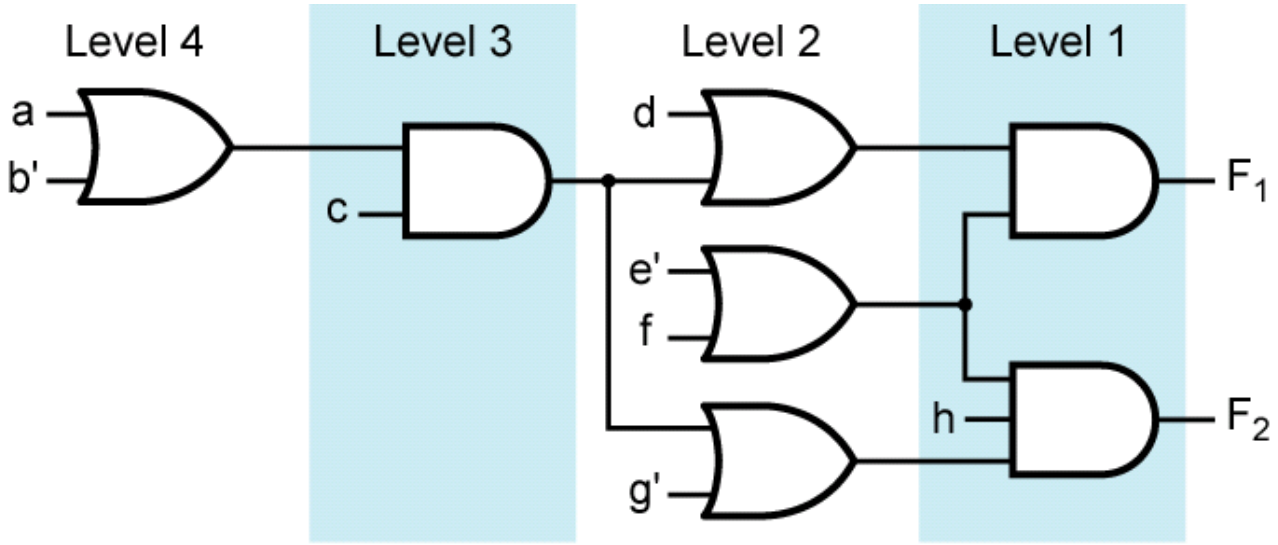


(a) Solution with maximum number of common terms requires 8 gates, 26 inputs

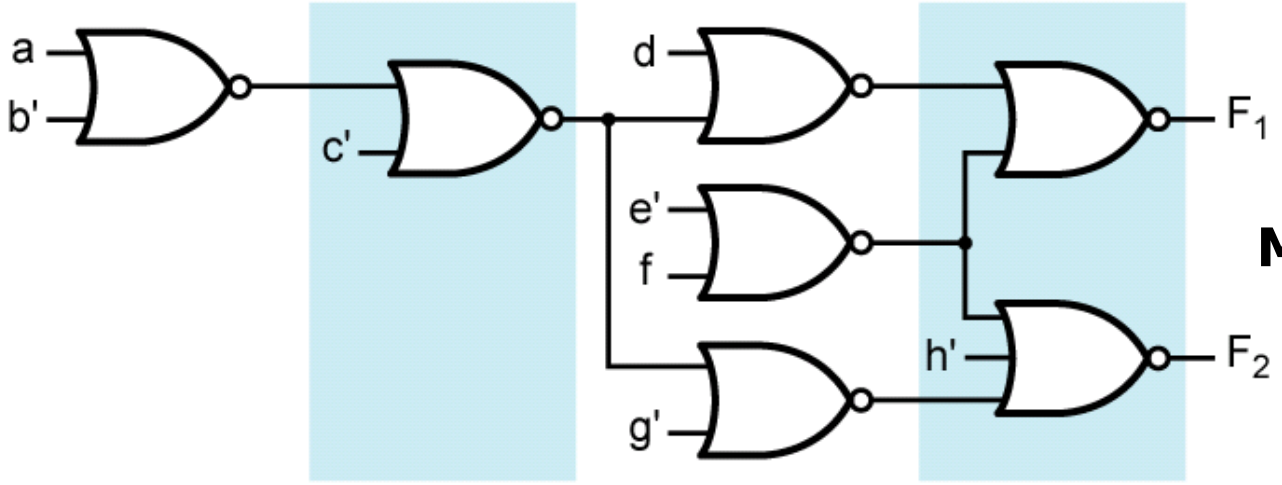


(b) Best solution requires 7 gates, 18 inputs and has no common terms

Figure 7-23



(a) Network of AND and OR gates



(b) NOR network

The procedure for design of single-output, multi-level NAND- and NOR-gate circuits also applies to multiple-output circuits

Figure 7-24:
Multi-Level Circuit Conversion to NOR Gates