CHAPTER 6

# SYNTHESIS OF COMBINATIONAL CIRCUITS

## 6.1 Introduction

Boolean functions can be expressed in the forms of sum-of-products and product-of-sums. These expressions can also be minimized using algebraic manipulations or Karnaugh maps. The goal of minimization is to reduce cost, which is defined by the number of gates and the number of gate inputs.

### Single-Rail and Double-Rail Signals

When a variable of a function or input to a logic circuit is available in only one form, it is called a single-rail variable or input. An inverter is required to get the other form. When both true and complemented forms of a variable or an input are available, it is called a double-rail variable or input.

### Fan-In and Fan-Out Limits

Fan-in limit is a constraint on the number of inputs to a gate. A product or sum term of more than n literals cannot be realized by an n-input AND gate or OR gate. Under such circumstances, either a gate with greater fan-in limit is used, or it is implemented by multiple gates. It is also possible to re-formulate the Boolean expression by applying the distributive law to eliminate fan-in limits.

When the output of a gate is connected to the inputs of other gates, the limit to the number of gates that it can be connected to is called fan-out limit. One technique to deal with this problem is to divert the output signal into a number of buffers within the fan-out limit. The inputs to the gates that were driven by the gate output can now be driven by the outputs of the buffers. Re-formulation of the Boolean expression is also another way of solving the problem.

### Level of Digital Circuits

When inputs are applied to a digital circuit, they will propagate to the output through various paths. Each path is the concatenation of a number of gates. Assume all gates have the same delay. If n is the number of gates in the path with the longest delay in a circuit, the circuit is called an n-level circuit.

## 6.2 Two-Level Circuits

In realizing sum-of-products and product-of-sums expressions, it is always assume that all inputs are double-rail. No inverters are required to generate the complemented form of the inputs. There is also no fan-in limit on the gates that are used to implement the SOP and POS expressions. Therefore the implementation of a sum-of-products or a product-of-sums expression is a 2-level circuit. For instance,

$$F(A,B,C,D) = \Sigma\, m(4, 5, 7, 9, 11, 13, 15)$$

The simplest sum-of-products expression and simplest product-of-sums expression are

$$F(A,B,C,D) = A'BC' + BCD + AD = (A + B)\,(A' + D)\,(C' + D)$$

The sum-of-products and product-of-sums expressions are implemented in Figure 6.1 as a 2-level AND-OR circuit and a 2-level OR-AND circuit respectively.

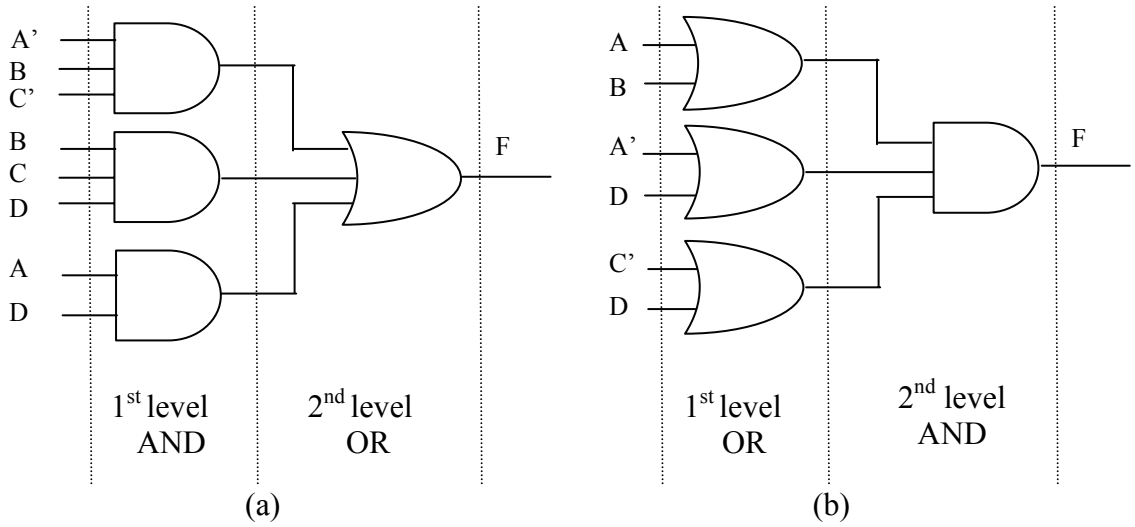
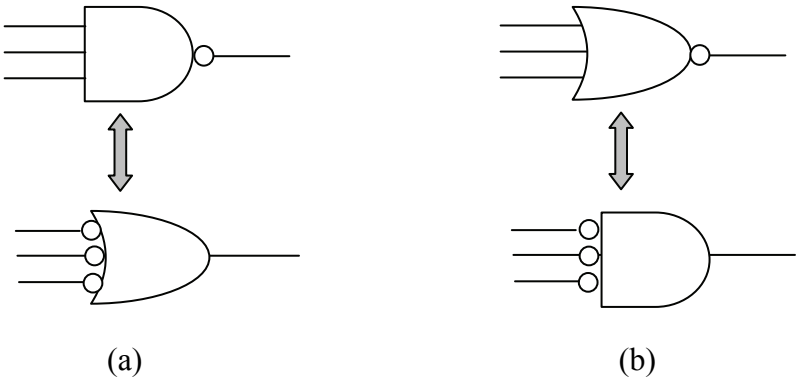



Figure 6.1 (a) 2-level AND-OR circuit. (b) 2-level OR-AND circuit.

The 2-level circuits in Figure 6.1 can be implemented using other types of gates. To perform the conversion, the gate equivalencies in Figures 4.13 (a) and (b) are repeated in Figure 6.2 for convenience.

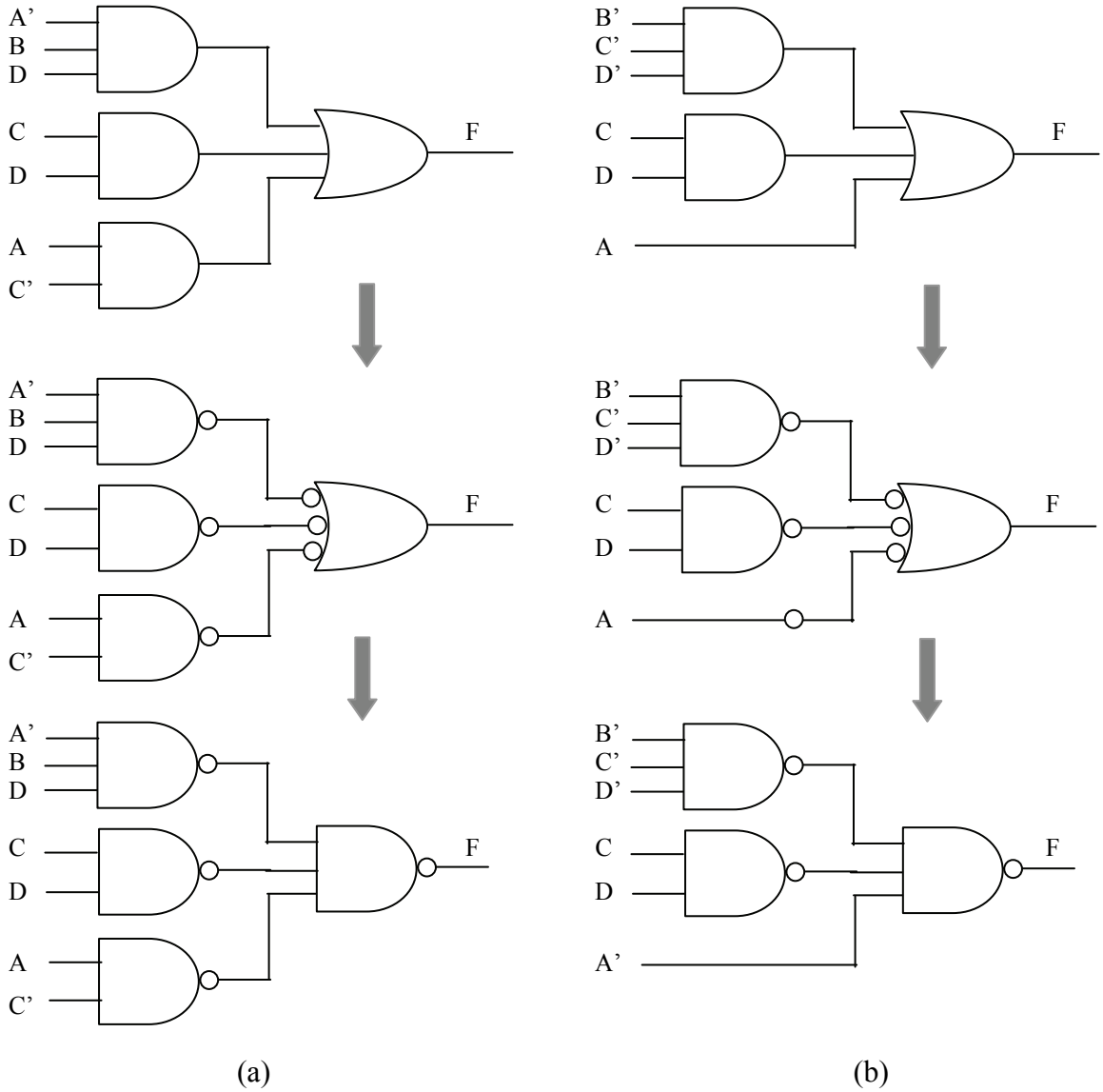
### AND-OR and NAND-NAND Circuits

Since NAND is a functionally complete set, a digital circuit can be implemented just by NAND gates. The conversion is illustrated in Figure 6.3. A 2-level AND-OR circuit in Figure 6.3(a) is used as an example. Bubbles are used in the conversion process for inversions that may or may not necessarily require inverters. A pair of bubbles is



(a) (b)

Figure 6.2 (a) Equivalence between OR with inverted inputs and NAND.
(b) Equivalence between AND with inverted inputs and NOR.




(a) (b)

Figure 6.3 Conversion between 2-level AND-OR and 2-level NAND-NAND circuits.

inserted between each AND gate output in the first level and the input to the OR gate in the second level, as shown in the middle circuit diagram of Figure 6.3(a). Attaching a bubble to the output of an AND gate in the first level turns it into a NAND gate. The second level becomes an OR gate with inverted inputs, which is equivalent to a NAND gate, as shown in Figure 6.2(a). With the replacement of the OR symbol with inverted inputs by a NAND symbol, the 2-level AND-OR circuit becomes a 2-level NAND-NAND circuit in the bottom diagram of Figure 6.3(a). Thus each and every gate in a 2-level AND-OR circuit can be replaced with a NAND gate. A 2-level AND-OR circuit is equivalent to a 2-level NAND-NAND circuit.

The conversion of another 2-level AND-OR circuit to a 2-level NAND-NAND circuit is illustrated in Figure 6.3(b). It is noted that the third input to the second level OR gate is not from the output of an AND gate. It is simply a literal, which is A. The absence of an AND gate before the third input of the OR gate leaves a lone bubble. To take care of this bubble or inversion, a NAND gate may be used in place of the bubble. Since the inputs are double-rail, the bubble can be removed by complementing the literal from A to A', as shown in the bottom circuit diagram of Figure 6.3(b). A 2-level AND-OR circuit is said to be non-standard if some inputs to the second level OR gate are literals instead of product terms. Each and every gate in such a circuit can still be replaced with a NAND gate to make it a 2-level NAND-NAND circuit. However, each and every literal to the OR gate must be complemented.

## OR-AND and NOR-NOR Circuits

Digital circuits can be implemented with only NOR gates because NOR is functionally complete. The conversion from 2-level OR-AND circuit to a 2-level NOR-NOR circuit is illustrated in Figure 6.4. Similar to the conversion from 2-level AND-OR to 2-level NAND-NAND, a pair of bubbles is inserted between a gate output in the first level and an input to the second level. One of the bubble is combined with the OR gate in the first level to make it a NOR. The second bubble inverts the input to the AND gate in the second level. When all the inputs to the AND gate are inverted, as shown in Figure 6.2(b), it is equivalent to a NOR gate. Thus a 2-level OR-AND circuit is equivalent to a 2-level NOR-NOR circuit. Figure 6.4(b) is the conversion of a non-standard 2-level OR-AND circuit. One of the inputs to the second level AND gate is just a literal. This literal has to be complemented in the conversion.

## AND-NOR Circuit

Two-level AND-NOR circuits are a standard configuration in some integrated circuits and programmable logic devices. A 2-level AND-NOR circuit has an array of AND gates in the first level. The outputs of the AND gates are connected to a NOR gate in the second level. Because of the inversion after OR, it is also called an AOI (AND-OR-INVERSION) circuit. In implementing a function F as a 2-level AND-NOR circuit, imagine that the bubble at the NOR gate output could be detached from the OR. As shown in Figure 6.5, the signal before the bubble is a 2-level AND-OR circuit for F'. Thus the design is to find the sum-of-products for F'. After F' is implemented as a 2-level

AND-OR circuit, the bubble is then attached to the OR gate to make it a 2-level AND-NOR circuit for F.

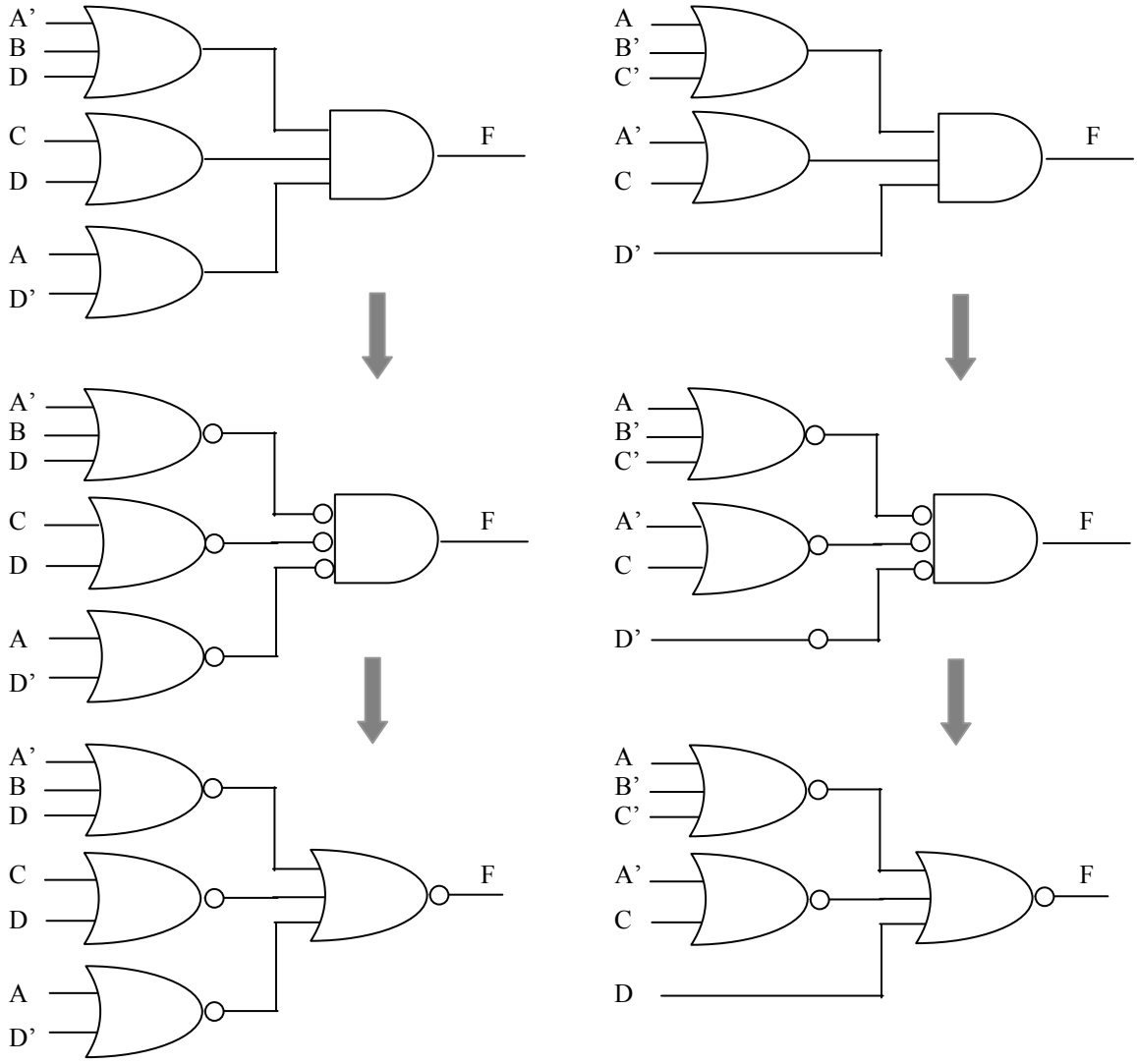



Figure 6.4 Conversion between 2-level OR-AND and 2-level NOR-NOR circuits.

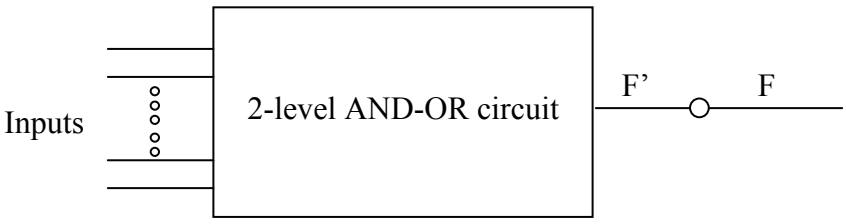



Figure 6.5 Synthesis of 2-level AND-NOR circuit.

❖ Example 6.1

$$F(A,B,C,D) = \Sigma\, m(2, 3, 4, 5, 7, 10, 11, 15)$$

The function F is to be implemented as a 2-level AND-NOR circuit. As previously explained, a sum-of-products expression for F' should be obtained in the synthesis. The minterm list representation for F' is

$$F'(A,B,C,D) = \Sigma\, m(0, 1, 6, 8, 9, 12, 13, 14)$$

The simplest sum-of-products for F' is

$$F'(A,B,C,D) = BCD' + AC' + B'C'$$

The simplest sum-of-products for F' can also be derived from the product-of-sums expression for F, which is

$$F(A,B,C,D) = \pi\, M(0, 1, 6, 8, 9, 12, 13, 14)$$

The simplest product-of-sums expression for F is

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

This expression for F, when complemented, will become a sum-of-products for F' by applying DeMorgan's theorem.

Assume that an AND-NOR circuit with four 3-input AND gates in the first level is available for implementation. In two of the AND gates, only two inputs are needed, the third inputs can be properly assigned a value of 1 or connected to either one of the other two inputs. The fourth AND gate is not used. Its inputs should be properly connected to generate an output of 0. The signal at a gate output may not be valid if unused inputs are left unintended (or floating).

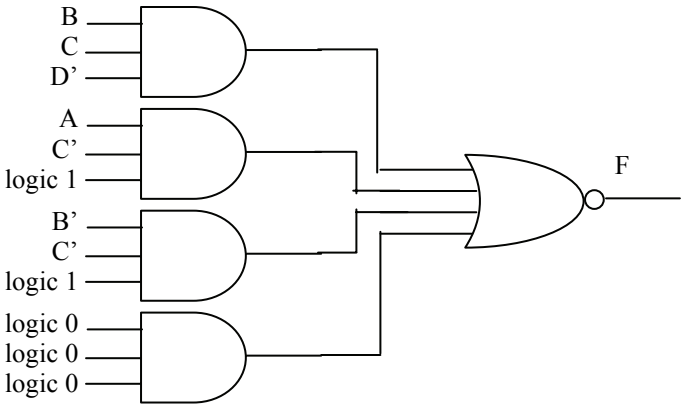



Figure 6.6 Realization of the function in Example 6.1 as an AND-NOR circuit.

## 6.3 Multi-Level Circuits

When fan-in limit becomes a problem in the implementation of a Boolean function, re-formulation of the function is required and it is no longer possible to implement the function as a 2-level circuit. Sometimes, a multi-level circuit may also be more economical than a 2-level circuit. On the other hand, increasing the gate level of a circuit will increase the propagation delay that may slow down the speed of operation. When speed is critical in the operation of a circuit, multi-level circuits are not the right option.

❖ Example 6.2

$$F(A,B,C,D) = BD + CD + A'BC + ABC'$$

Given above is the simplest sum-of-products expression of a 4-variable function F(A,B,C,D). If the gates used for implementation have a fan-in limit of two, the sum-of-products expression can be changed to the following expression by factoring.

$$F(A,B,C,D) = D(B + C) + B(A'C + AC')$$

The function F is the same function used in Example 3.14. The minimization of a circuit using only 2-input AND gates and 2-input OR gates is equivalent to minimizing the function to a minimum number of literals. It is implemented as a 4-level circuit as shown in Figure 6.7.

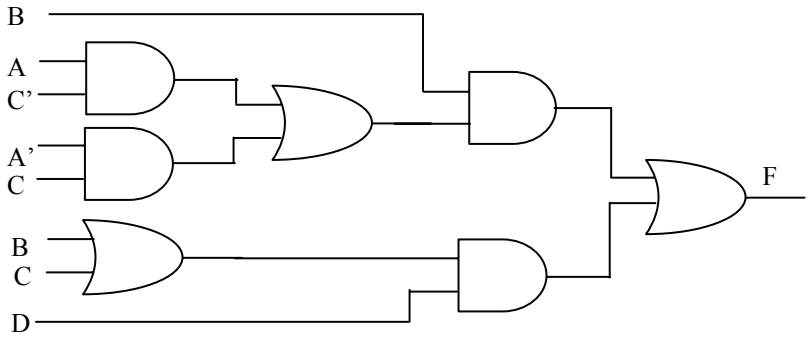



Figure 6.7 4-level circuit for Example 6.2.

❖ Example 6.3

In this example, a 5-variable function is used to show that literal minimization of a function may start from either a sum-of-products or product-of-sums expression. Although one form can always be converted to the other form using Boolean algebra, it may be better to get both forms and find out which one is easier to start with.

Given that $F(A,B,C,D,E) = \Sigma\, m(0, 1, 2, 3, 11, 16 - 23, 27, 31)$

The simplest sum-of products and product-of-sums expressions can be obtained using K-maps and are as follows:

$$AB' + ADE + B'C' + C'DE$$

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

From the above expressions, it is obvious that the product-of-sums expression is better because it has fewer literals. To implement the function using 2-input gates, the expression can be changed to

$$(A + C') (B' + DE)$$

The circuit has three levels and is shown in Figure 6.8.

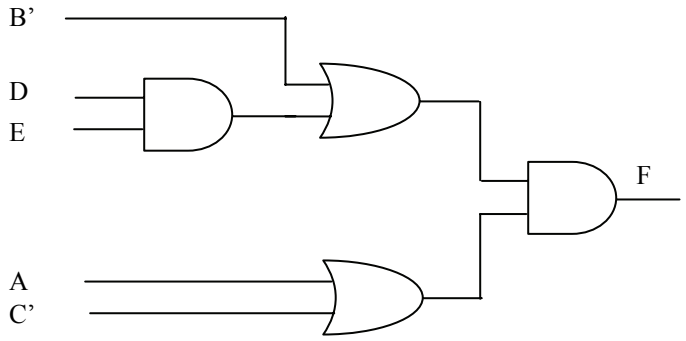



Figure 6.8 3-level circuit for F in Example 6.3.

## 6.4 All-NAND and All-NOR Circuits

In section 6.2, it is shown that a 2-level circuit may be realized using only NAND gates or NOR gates. A multi-level circuit implemented with AND gates and OR gates can also be transformed into a circuit with only NAND gates or NOR gates, which are also called an all-NAND and an all-NOR circuit respectively. Transformation does not have to be performed gate by gate. The equivalency between AND-OR and NAND-NAND, as well as OR-AND and NOR-NOR, can be applied to gate conversions in a multi-level circuit. To convert a circuit to one with only NAND gates, look for all the 2-level AND-OR sub-circuits in a multi-level circuit. Then replace each sub-circuit with a 2-level NAND-NAND circuit. Gates not included in the 2-level AND-OR sub-circuits (isolated gates) can be transformed one by one using the equivalencies in Figure 6.2. Transformation of a multi-level circuit to a circuit with only NOR gates can also be carried out in a similar manner. A multi-level circuit is partitioned into a number of 2-level OR-AND sub-circuit and some (if there is any) isolated gates. Each 2-level OR-AND sub-circuit is replaced with a 2-level NOR-NOR circuit. Isolated gates are converted one by one using the equivalencies in Figure 6.2.

❖ Example 6.4

$$F(A,B,C,D) = \Sigma\, m(3, 5, 6, 7, 12, 13, 14, 15)$$

The simplest sum-of-products and simplest product-of-sums expressions for F are respectively

$$AB + BC + BD + A'CD$$

and $(B + D)\,(B + C)\,(A' + B)\,(A + C + D)$

If the function is implemented using only 2-input NAND gates, F is first implemented using 2-input AND gates and 2-input OR gates. The circuit is shown in Figure 6.9(a) using the right-hand-side of the following equation.

$$AB + BC + BD + A'CD = B(A + C) + D(B + A'C)$$

There are two 2-level AND-OR sub-circuits and an OR gate. Each sub-circuit as well as the OR gate is shown by a different gray level. The conversion to an all-NAND circuit is given in Figure 6.9(b).

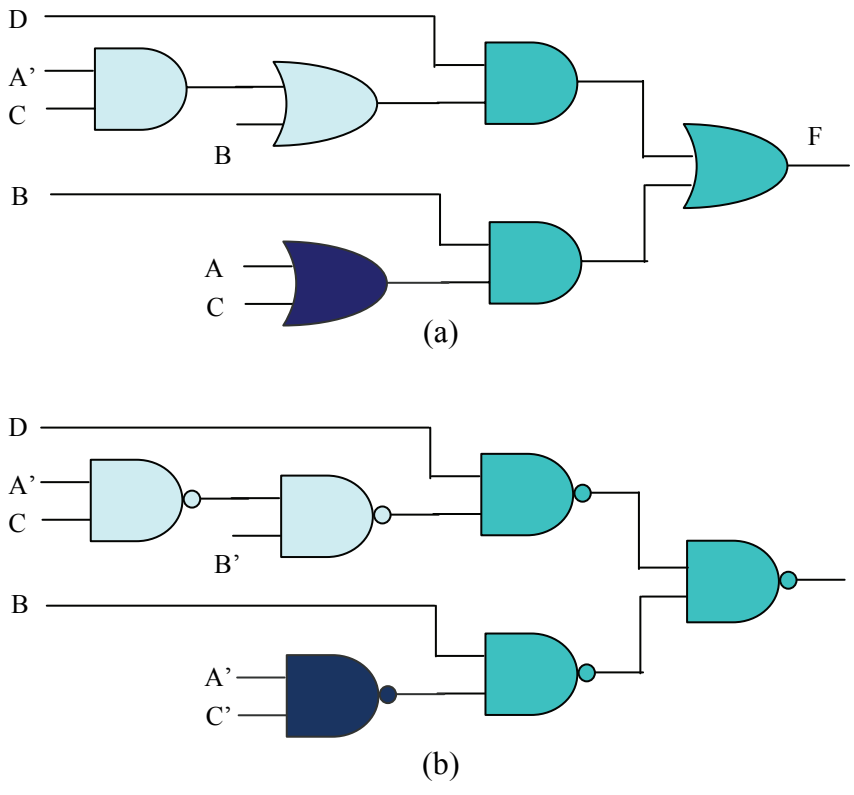



Figure 6.9 Conversion of a multi-level circuit to an all-NAND circuit.

The implementation of Figure 6.9(a) as an all-NOR circuit is shown in Figure 6.10. There are two 2-level OR-AND sub-circuits and two isolated gates, which are shown by four different gray levels. The isolated AND gate is replaced by a NOR gate with inverted inputs. The OR gate at the output is replaced with two NOR gates.

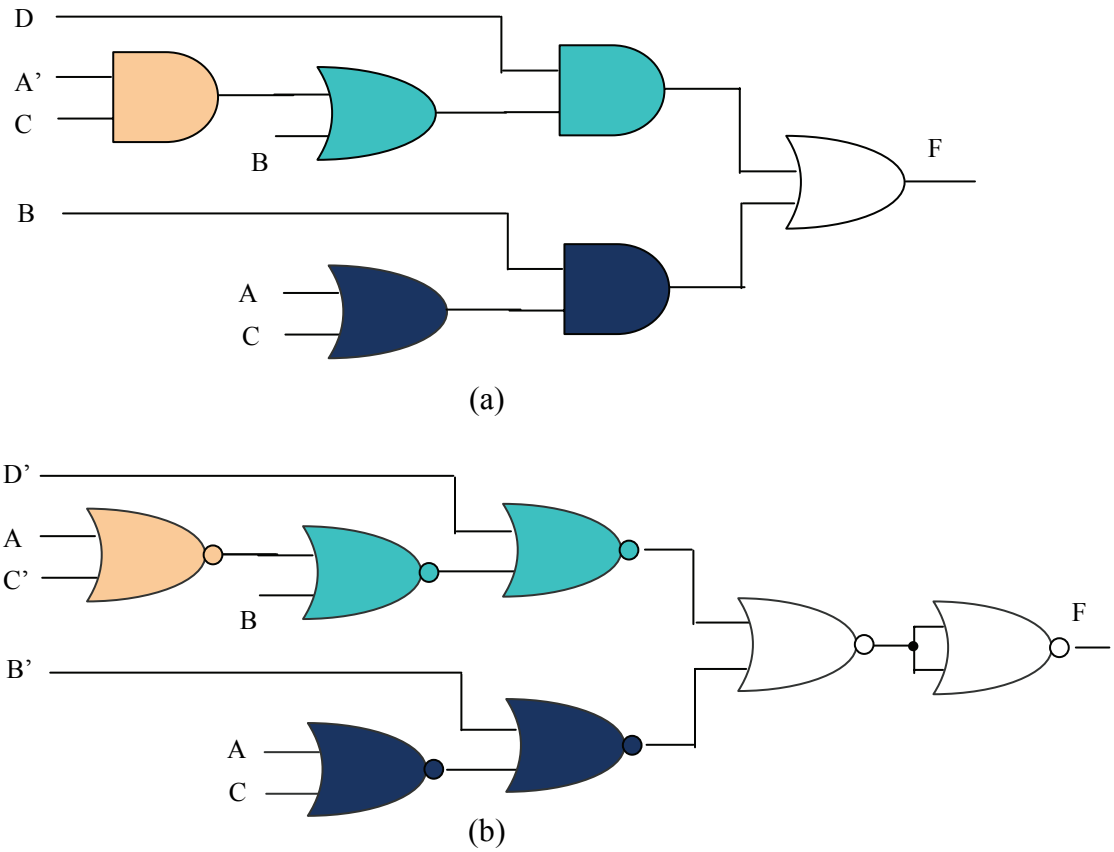



Figure 6.10 Conversion of a multi-level circuit to an all-NOR circuit.

An all-NOR circuit can also be implemented from the product-of-sums expression.

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

The realization of the right-hand-side of the above equation is shown in Figure 6.11(a). There are two 2-level OR-AND sub-circuits that are replaced with two 2-level NOR-NOR circuits. The AND gate is changed to a NOR gate with inverted inputs. The all-NOR circuit is shown in Figure 6.11(b). By comparing Figures 6.10(b) and 6.11(b), it is seen that the all-NOR circuit in figure 6.11(b) is more economical. Thus it is suggested that a product-of-sums expression be used for the implementation of an all-NOR circuit. Similarly, a sum-of-products expression is suggested for the implementation of an all-NAND circuit.

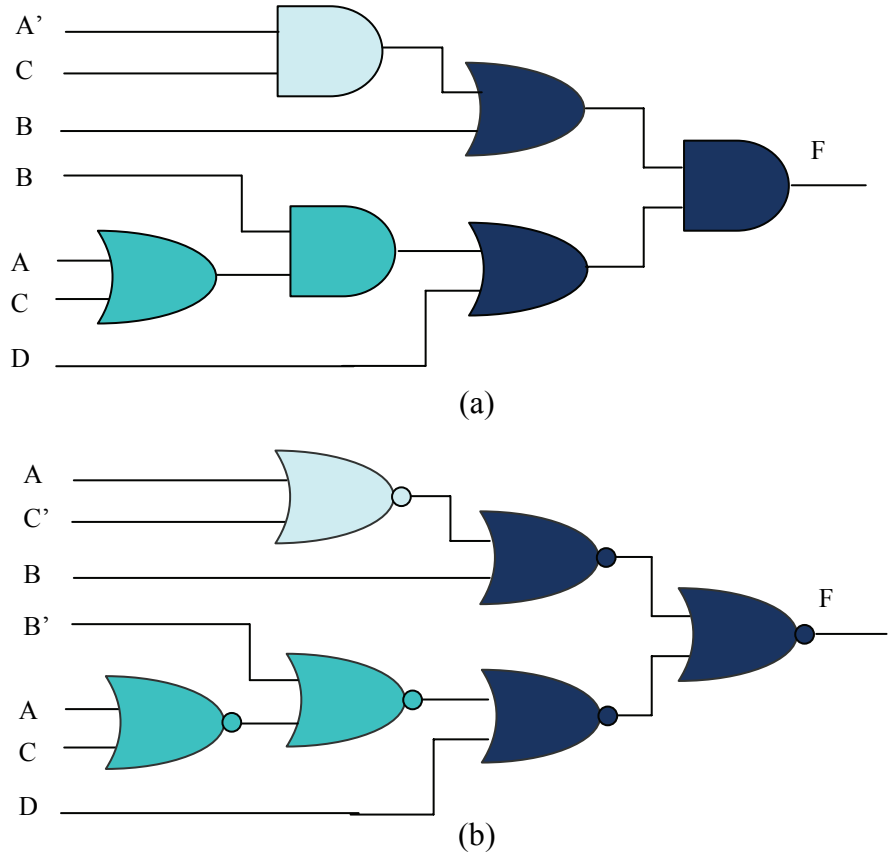



Figure 6.11 A different all-NOR implementation from Figure 6.10.

## PROBLEMS

1. Given $f(A,B,C,D) = \Sigma\, m\,(2, 3, 4, 5, 9, 12, 13)$, realize f as

   (a) a 2-level AND-OR circuit.
   (b) a 2-level NAND-NAND circuit.
   (c) a 2-level OR-AND circuit.
   (d) a 2-level NOR-NOR circuit.
   (e) a 2-level AND-NOR circuit.

2. Given $f(A,B,C,D,E) = \Sigma\, m\,(2, 3, 5 – 11, 13, 21, 23 – 27, 29, 31)$, realize f as

   (a) a 2-level AND-OR circuit.

(b) a 2-level NAND-NAND circuit.
(c) a 2-level OR-AND circuit.
(d) a 2-level NOR-NOR circuit.
(e) a 2-level AND-NOR circuit.

3. A, B, C, and D are four chairs in a room. Each chair is either occupied (1) or empty (0). F(A,B,C,D) = 1 if the number of chairs being occupied is more than 2: otherwise F = 0.

(a) Construct a truth table for F(A,B,C,D).
(b) Express F in minterm list form.
(c) Express F in maxterm list form.
(d) Find the simplest sum-of-products for F.
(e) Find the simplest product-of-sums for F.
(f) Find the simplest sum-of-products for F'.
(g) Find the simplest product-of-sums for F'.
(h) Design a 2-level NAND-NAND circuit for F.
(i) Design a 2-level NOR-NOR circuit for F.
(j) Design a 2-level NAND-NAND circuit for F'.
(k) Design a 2-level NOR-NOR circuit for F'.
(l) Design a 2-level AND-NOR circuit for F.
(m) Design a 2-level AND-NOR circuit for F'.

4. Given $f(A,B,C,D) = \Sigma\, m\,(2, 3, 5, 7, 10, 13, 14, 15)$, realize f as

(a) a 2-level AND-NOR circuit.
(b) a 2-level NAND-AND circuit.
(c) a 2-level OR-NAND circuit.
(d) a 2-level NOR-OR circuit.

5. Given $f(A,B,C,D,E) = \Sigma\, m\,(0 – 7, 12, 13, 15, 16 – 23, 26, 30, 31)$, realize f as

(a) a 2-level AND-OR circuit.
(b) a 2-level NAND-NAND circuit.
(c) a 2-level OR-AND circuit.
(d) a 2-level NOR-NOR circuit.
(e) a 2-level AND-NOR circuit.
(f) a 2-level NAND-AND circuit.
(g) a 2-level OR-NAND circuit.
(h) a 2-level NOR-OR circuit.

6. Realize each of the following functions using a minimum number of 2-input NAND gates. All inputs are double-rail.

(a) $f(A,B,C,D) = \Sigma\, m\,(3,4,5,7,8,9,13)$
(b) $f(A,B,C,D) = \Sigma\, m\,(2,3,4,5,7,9,13) + d(6,8,14,15)$

7. Realize each of the following functions using a minimum number of 2-input NOR gates. All inputs are double-rail.

   (a) $f(A,B,C,D) = \Sigma\, m\,(3,4,5,7,8,9,10,13)$
   (b) $f(A,B,C,D) = \pi\, M\,(0,1,10,11,12) \bullet D(6,8,14,15)$

8. Minimize each of the following functions using a minimum number of 2-input NAND gates and exclusive-OR gates. All inputs are single-rail.

   (a) $f(A,B,C,D) = \Sigma\, m\,(3,4,5,7,8,9,13)$
   (b) $f(A,B,C,D) = \Sigma\, m\,(0,1,2,3,5,11,12,13,14,15)$

9. A 2-bit magnitude comparison circuit has four inputs $a_1$, $a_0$, $b_1$, $b_0$. The circuit compares the magnitudes of two 2-bit numbers $A = (a_1a_0)_2$ and $B = (b_1b_0)_2$ and generates three outputs EQ (1 if A = B, 0 if $A \neq B$), GT (1 if A greater than B, 0 if A not greater than B), and LT (1 if A less than B, 0 if A not less than B). Design the circuit using a minimum number of gates. Use exclusive-NOR gates if necessary.

10. Repeat Problem 9 for two 3-bit numbers $A = (a_2a_1a_0)_2$ and $B = (b_2b_1b_0)_2$.