EXAMPLE 5.16 YET ANOTHER IMPLEMENTATION USING NORS Let us derive an implementation based on two-input *NOR* gates for the function $\overline{AB + C + D}$. Assume that both the true and complement version of each of the inputs is available:

$$\overline{AB + C + D} = \overline{\overline{\overline{A}\,\overline{B}} + \overline{\overline{\overline{(C + D)}}}} \tag{5.35}$$

$$= \overline{\overline{\overline{A} + \overline{B}} + \overline{\overline{\overline{C + D}} + \overline{C + D}}} \tag{5.36}$$

$$= \overline{((\overline{\overline{A} + \overline{B}}) + ((\overline{\overline{\overline{C + D}} + \overline{C + D}}))}. \tag{5.37}$$

Implementing each of the expressions within parentheses using two-input *NOR* gates, we get the circuit shown in Figure 5.24.

Notice that the algebraic simplification process was quite cumbersome. We can actually perform the same transformation directly on a gate-level circuit with greater ease. Figure 5.25 shows how the original circuit for $\overline{AB + C + D}$ from one of the implementations in Figure 5.18 can be transformed into a two-input *NOR* implementation. The transformations exploit the fact that two inverters (or circles) in series cancel each other.

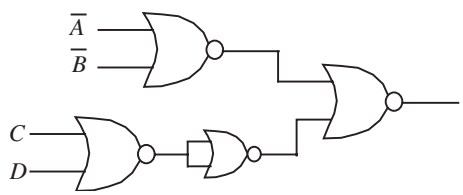

FIGURE 5.24 *NOR* implementation of $\overline{AB + C + D}$.

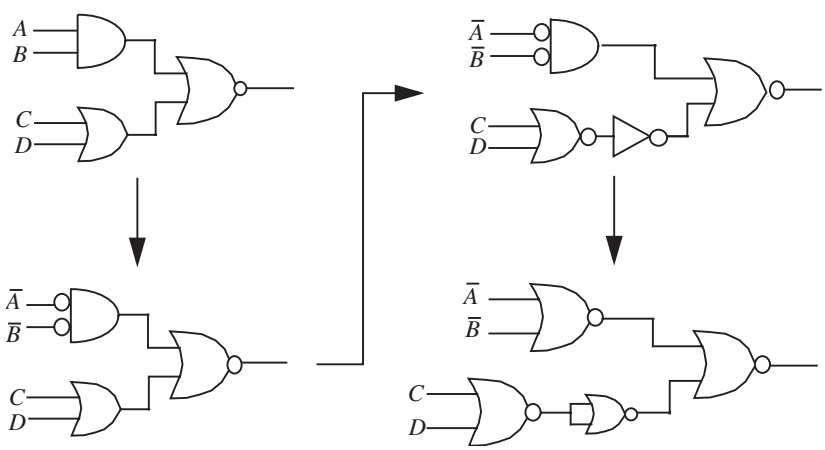

FIGURE 5.25 *NOR* transformations for $\overline{AB + C + D}$.