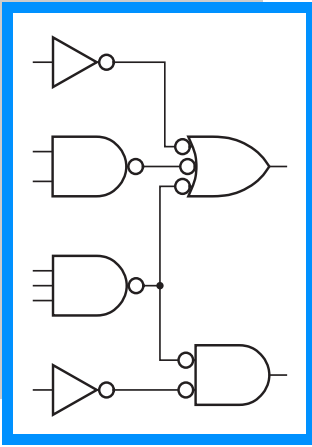


# 4



## COMBINATIONAL LOGIC DESIGN PRINCIPLES

### EXERCISE SOLUTIONS

4.2

T2:

X	$X + 1$	=	1
0	1	=	1
1	1	=	1

T3:

X	$X + X$	=	X
0	0	=	0
1	1	=	1

4.3

T3'

X	$X \cdot X$	=	X
0	0	=	0
1	1	=	1

4.4

T6

X	Y	$X + Y$	=	$Y + X$
0	0	0	=	0
0	1	1	=	1
1	0	1	=	1
1	1	1	=	1

## 4-2 DIGITAL CIRCUITS

4.5 The original expression assumes precedence of  $\cdot$  over  $+$ , that is, the expression is  $X + (Y \cdot Z)$ . The parenthesization must be retained for the correct result,  $X' \cdot (Y' + Z')$ , or the precedence must be swapped.

4.6 The answers for parts (a), (b), (c) are as follows.

$$\begin{aligned} & W \cdot X \cdot Y \cdot Z \cdot (W \cdot X \cdot Y \cdot Z' + W \cdot X' \cdot Y \cdot Z + W' \cdot X \cdot Y \cdot Z + W \cdot X \cdot Y' \cdot Z) \\ &= W \cdot X \cdot Y \cdot Z \cdot W \cdot X \cdot Y \cdot Z' + W \cdot X \cdot Y \cdot Z \cdot W \cdot X' \cdot Y \cdot Z + W \cdot X \cdot Y \cdot Z \cdot W' \cdot X \cdot Y \cdot Z + W \cdot X \cdot Y \cdot Z \cdot W \cdot X \cdot Y' \cdot Z \text{ (T8)} \\ &= 0 + 0 + 0 + 0 \text{ (T6', T5', T2')} \\ &= 0 \text{ (A4')} \end{aligned}$$

4.7

(a)

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

(b)

W	X	Y	Z	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

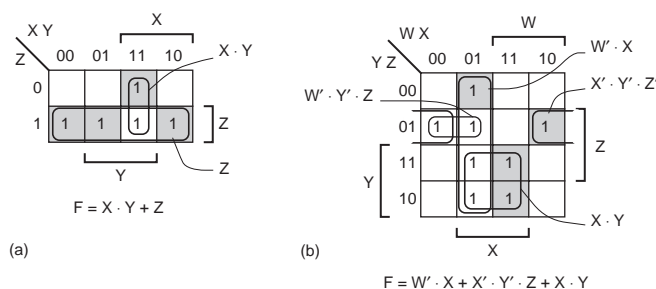
4.9

(a)  $F = X' \cdot Y + Y' \cdot X = (X + Y) \cdot (X' + Y')$

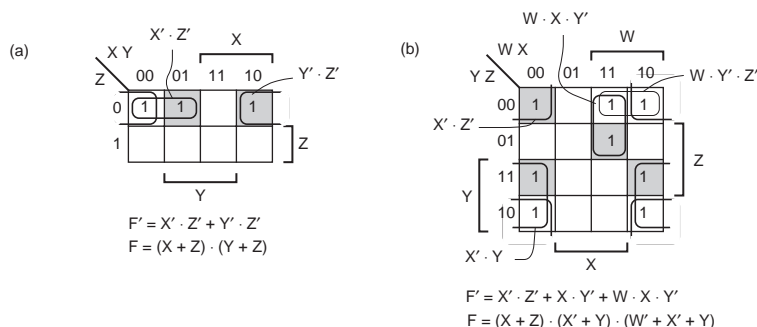
(b)  $F = A \cdot B = (A + B) \cdot (A + B') \cdot (A' + B)$

4.12 (1) Including inverters makes the problem too difficult. (2) In modern PLD-based designs, inverters do cost nothing and really can be ignored.

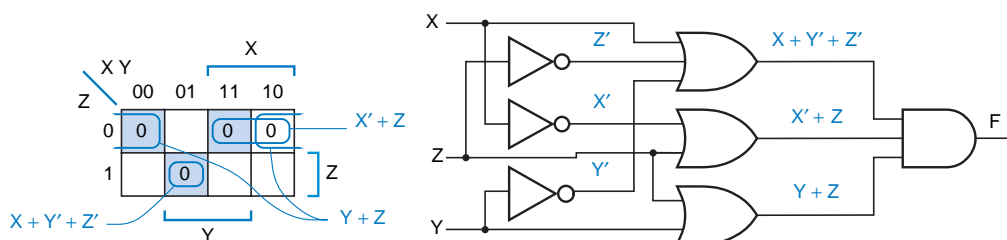
4.13



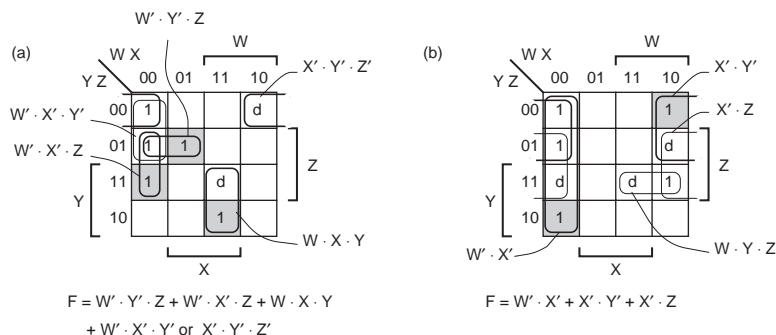
4.14



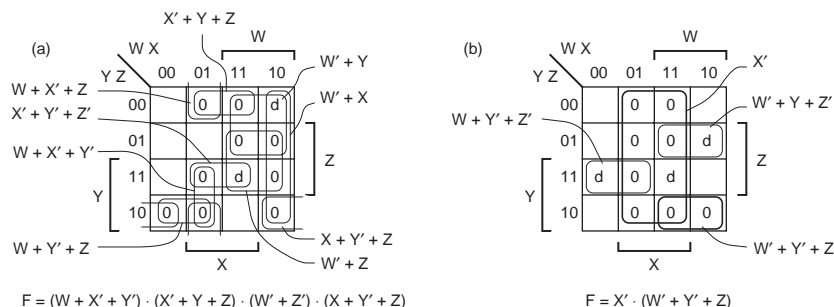
4.15 (a) Cost is less—one less gate input.



4.19



4.20



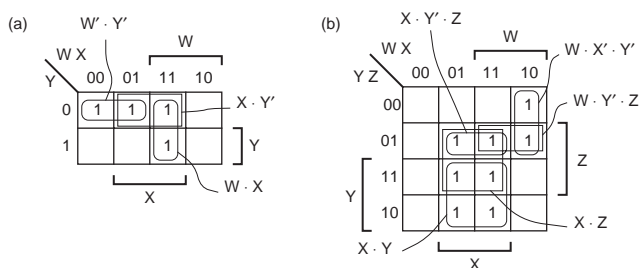
4.21 For the minimal sum-of-products expression to equal the minimal product-of-sums expression, the corresponding maps must have the opposite don't-cares covered, so that the expressions yield the same value for the don't-care input combinations.

(a) Both minimal sum-of-products expressions cover cell 15; they are equal. The minimal product-of-sums expression also covers cell 15, so the expressions are not equal. The s-of-p and p-of-s expressions require the same number of gates, but the p-of-s requires one fewer input.

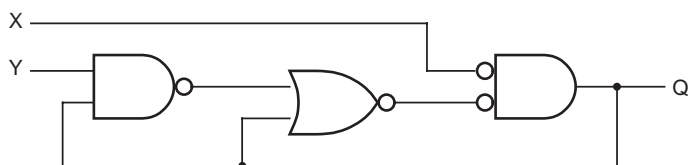
(b) Both minimal sum-of-products expressions cover cell 3 and 9 and not 15; they are equal. The minimal product-of-sums expression covers cell 15, and not 3 or 9, so the expressions are equal. The p-of-s expression requires fewer gates and inputs.

#### 4-4 DIGITAL CIRCUITS

4.22 Consensus terms that must be added to cover the hazards are “circled” with rectangles.



4.28



Analyzing this circuit with the standard method for feedback sequential circuits (Section 5.5), we get the following excitation equation:

$$\begin{aligned} Q^* &= X' \cdot ((Y \cdot Q)' + Q) \\ &= X' \cdot (Y' + Q' + Q) \\ &= X' \cdot 1 = X' \end{aligned}$$

Thus,  $Q^*$  is a function of  $X$  alone, and is independent of the circuit's previous “state.”

4.29

$$\begin{aligned} X \cdot 1 &= X (T1') \\ X \cdot (Y + Y') &= X (T5) \\ X \cdot Y + X \cdot Y' &= X (T8) \end{aligned}$$

4.31

$$\begin{aligned} (X + Y') \cdot Y &= X \cdot Y + Y' \cdot Y (T8) \\ &= X \cdot Y + Y \cdot Y' (T6') \\ &= X \cdot Y + 0 (T5') \\ &= X \cdot Y (T1) \end{aligned}$$

4.35

$$\begin{aligned} X_1 \cdot X_2 \cdot \dots \cdot X_n \cdot X_n &= X_1 \cdot X_2 \cdot \dots \cdot (X_n \cdot X_n) (T6', T7' \text{ as required}) \\ &= X_1 \cdot X_2 \cdot \dots \cdot X_n (T3') \\ X_1 + X_2 + \dots + X_n + X_n &= X_1 + X_2 + \dots + (X_n + X_n) (T6, T7 \text{ as required}) \\ &= (X_1 + X_2 + \dots + X_n) (T3) \end{aligned}$$

4.37 Figure 3-4(d) is more appropriate, since electrically a TTL NOR gate is just the wired-AND of inverters.

4.39

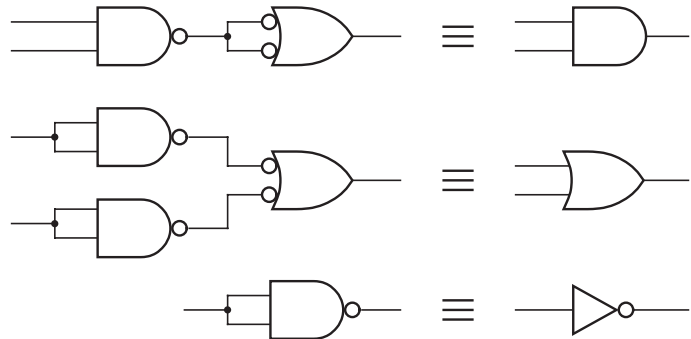
(a) True. If  $A \cdot B = 0$  then either  $A = 0$  or  $B = 0$ . If  $A + B = 1$  then either  $A = 1$  or  $B = 1$ . Therefore,  $A, B = 0, 1$  or  $1, 0$ , and  $A = B'$ .

4.41

$$\begin{aligned}
 F(X_1, \dots, X_i, X_{i+1}, \dots, X_n) &= X_1', \dots, X_i' \cdot F(0, \dots, 0, X_{i+1}, \dots, X_n) \\
 &+ X_1', \dots, X_i' \cdot F(0, \dots, 1, X_{i+1}, \dots, X_n) \\
 &\dots \\
 &+ \underbrace{X_1', \dots, X_i'}_{2^i \text{ minterms}} \cdot \underbrace{F(1, \dots, 1, X_{i+1}, \dots, X_n)}_{2^i \text{ combs}}
 \end{aligned}$$

A dual theorem may be written based on maxterms.

- 4.46 Yes, 2-input NAND gates form a complete set of logic gates. We prove the result in the figure on the right by showing that these gates can be used to make 2-input AND gates, 2-input OR gates, and inverters, which form a complete set.



- 4.52 Take the dual, “multiply out,” and take the dual again. The result is the same as “adding out.”

- 4.58 (a) 16 ns. (c) 18 ns. (d) 10 ns.

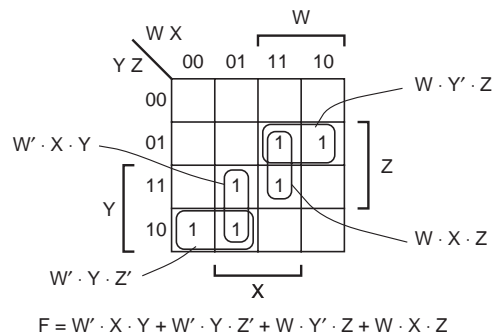
- 4.61 To make it easier to follow, we’ll take the dual, multiply out, and then take the dual again. Also, we’ll simplify using theorems T3’ and T6’, otherwise we’ll get a nonminimal result for sure. For Figure 4-27:

$$\begin{aligned}
 F &= X \cdot Z + Y' \cdot Z + X' \cdot Y \cdot Z' \\
 F^D &= (X + Z) \cdot (Y' + Z) \cdot (X' + Y + Z') \\
 &= X \cdot Y' \cdot Z' + X \cdot Z \cdot Y + Z \cdot Y' \cdot X' + Z \cdot Y' \cdot X' + Z \cdot Z \cdot X' + Z \cdot Z \cdot Y \text{ (T8, T5', T2')} \\
 &= X \cdot Y' \cdot Z' + X \cdot Y \cdot Z + X' \cdot Y' \cdot Z' + X' \cdot Z + Y \cdot Z \text{ (T3', T6')} \\
 F &= (X + Y' + Z') \cdot (X + Y + Z) \cdot (X' + Y' + Z) \cdot (X' + Z) \cdot (Y + Z) \text{ (not minimal)}
 \end{aligned}$$

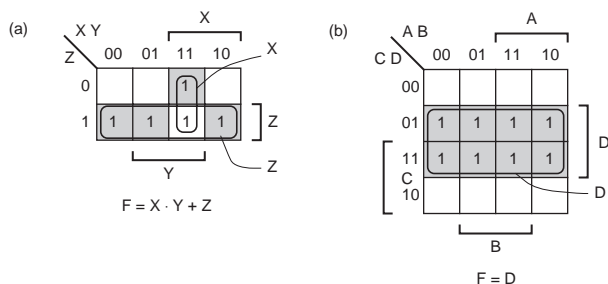
For Figure 4-29:

$$\begin{aligned}
 F &= X \cdot Z' + Y' \\
 F^D &= (X + Z') \cdot Y' \\
 &= X \cdot Y' + Z' \cdot Y' \text{ (T8)} \\
 &= X \cdot Y' + Y' \cdot Z' \text{ (T6')} \\
 F &= (X + Y') \cdot (Y' + Z') \text{ (minimal)}
 \end{aligned}$$

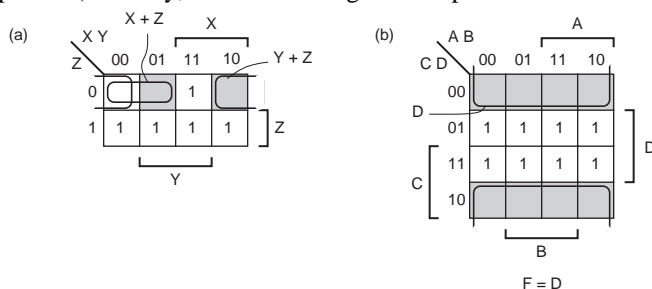
4.63



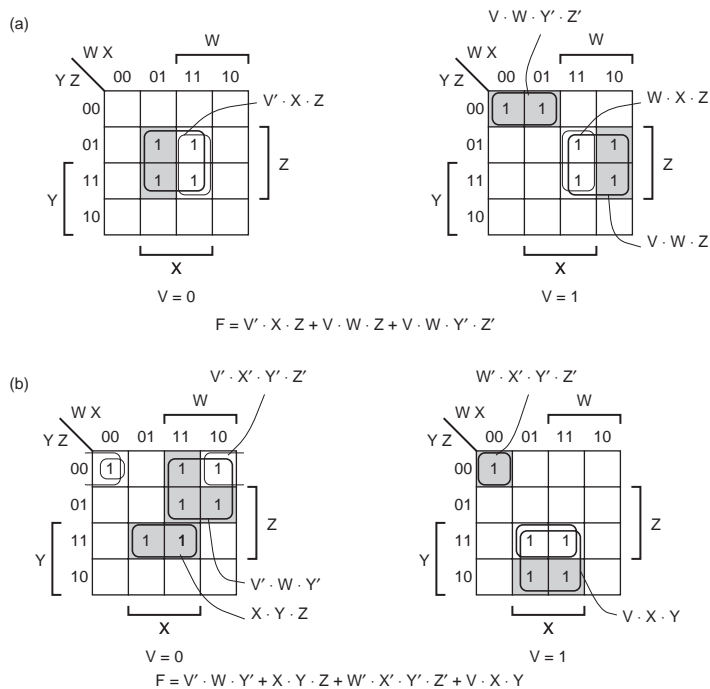
4.69 For part (d), note that it is easiest to work with the product-of-sums directly; rather than multiplying out, one simply enters the 0s on the map.



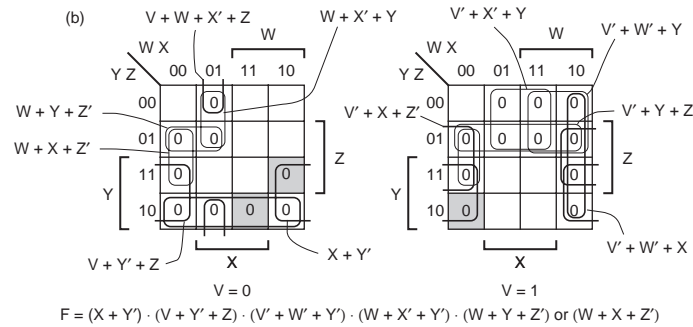
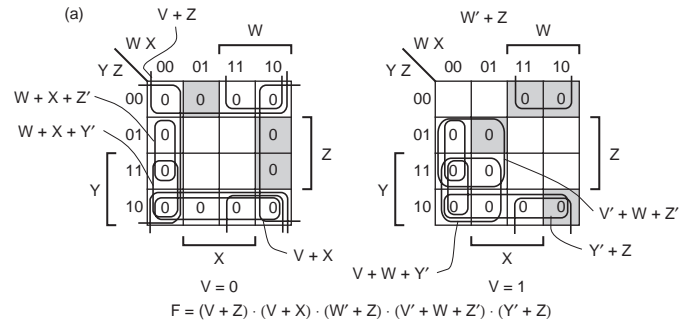
4.70 Note that in these maps are drawn for the “true” function and we’ve written sum terms for the prime implicants (the dual of prime implicants) directly, instead of using the complement method suggested in Section 4.3.6.



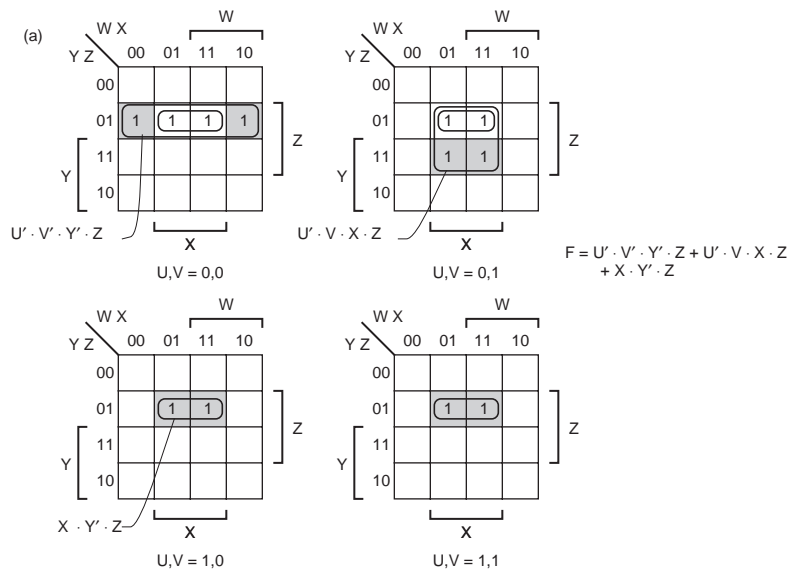
4.72



4.73 Note that in these maps are drawn for the “true” function and we’ve written sum terms for the prime implicants (the dual of prime implicants) directly, instead of using the complement method suggested in Section 4.3.6.



4.74



4.83 The name of the circuit comes from its output equation,  
 $F = 2B \text{ OR NOT } 2B.$

