

4. Um circuito XOR tem a expressão  $x = \overline{A}\overline{B} + \overline{A}B$ . Sua saída  $x$  será nível ALTO apenas quando as entradas  $A$  e  $B$  estiverem em níveis opostos.
5. Um circuito XNOR tem a expressão  $x = \overline{A}\overline{B} + AB$ . Sua saída  $x$  será nível ALTO apenas quando as entradas  $A$  e  $B$  estiverem no mesmo nível lógico.
6. Todas as portas básicas (AND, OR, NAND e NOR) podem ser usadas para habilitar ou desabilitar a passagem de um sinal lógico da entrada para a saída.
7. As principais famílias de CIs digitais são as famílias TTL e CMOS. Os CIs digitais estão disponíveis em uma ampla gama de complexidade (portas por CI), desde as funções lógicas básicas até as de alta complexidade.
8. Para realizar uma manutenção básica é necessário, pelo menos, saber como o circuito funciona, ter conhecimento sobre os tipos de falhas possíveis, um diagrama completo de conexão do circuito lógico e uma ponta de prova lógica.
9. Um dispositivo de lógica programável (PLD) é um CI que contém um grande número de portas lógicas, cujas interconexões podem ser programadas pelo usuário para gerar as relações lógicas desejadas entre entradas e saídas.
10. Para programar um PLD, é necessário um sistema de desenvolvimento, que consiste em um computador, um software de desenvolvimento para PLD e um programador, que realiza a programação do PLD.
11. O sistema da Altera permite técnicas de projeto hierárquico adequadas, usando qualquer forma de descrição de hardware.
12. Os tipos dos objetos de dados precisam ser especificados para que o compilador HDL saiba o intervalo de números a serem representados.
13. Tabelas-verdade podem ser fornecidas diretamente no arquivo fonte, usando os recursos de HDL.
14. As estruturas de controle lógico como IF, ELSE e CASE podem ser usadas para descrever o funcionamento de um circuito lógico, o que torna o código e a análise de defeitos muito mais simples.

## TERMOS IMPORTANTES

agrupamento de quadros	<i>exclusive</i> -OR (XOR)	ponta de prova lógica
atribuição de sinal selecionada	geração de paridade	PROCESS
bibliotecas	habilitar/desabilitar	produto-de-somas
BIT_VECTOR	IF/THEN	programador
CASE	índice	projeto hierárquico
complementar metal-óxido-semicondutor (CMOS)	integer	sequencial
concatenação	ISP	soma-de-produtos
concorrente	JEDEC	soquete ZIF
condição de irrelevância	JTAG	SSI, MSI, LSI, VLSI, ULSI, GSI
contenção	lista de sensibilidade	STD_LOGIC
dual-in-line package (DIP)	literais	STD_LOGIC_VECTOR
ELSE	lógica transistor-transistor (TTL)	tensão indeterminada
ELSIF	macrofunção	top-down
entrada flutuante	mapa de Karnaugh (mapa K)	verificação de paridade
estrutura de controle de decisão	matriz de bits	vetor de bits
<i>exclusive</i> -NOR (XNOR)	maxplus2	vetores de teste
	objetos	

## PROBLEMAS<sup>2</sup>

### SEÇÕES 4.2 E 4.3

**B 4.1\*** Simplifique as seguintes expressões usando a álgebra booleana.

(a)  $x = ABC + \overline{A}C$

(b)  $y = (Q + R)(\overline{Q} + \overline{R})$

(c)  $w = ABC + \overline{A}BC + \overline{A}$

(d)  $q = \overline{R}ST(R + S + \overline{T})$

(e)  $x = \overline{A}\overline{B}\overline{C} + \overline{A}BC + ABC + \overline{A}\overline{B}C + \overline{A}BC$

(f)  $z = (B + \overline{C})(\overline{B} + C) + \overline{A} + B + \overline{C}$

(g)  $y = (\overline{C} + \overline{D}) + \overline{A}C\overline{D} + \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}CD + AC\overline{D}$

(h)  $x = AB(\overline{C}\overline{D}) + \overline{A}BD + \overline{B}\overline{C}\overline{D}$

**B 4.2**

Simplifique o circuito mostrado na Figura 4.65 usando a álgebra booleana.

**B 4.3\***

Troque cada porta no circuito do Problema 4.2 por uma porta NOR e simplifique o circuito, usando a álgebra booleana.

<sup>2</sup> As respostas às questões assinaladas com um asterisco encontram-se no final do livro.

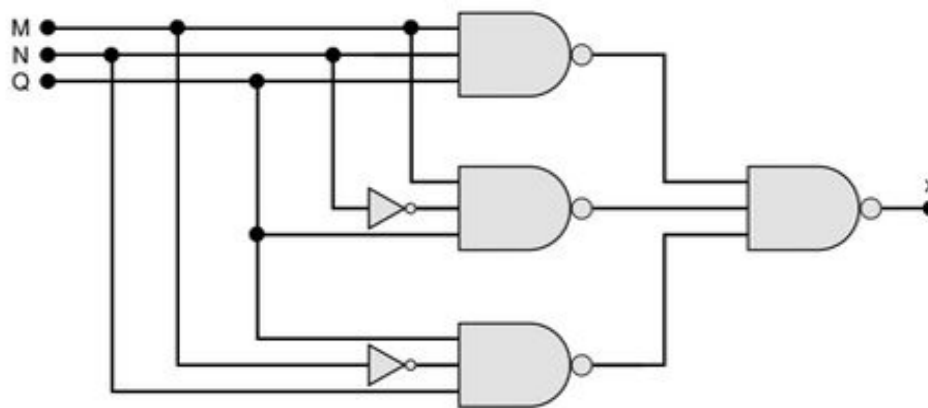


FIGURA 4.65 Problemas 4.2 e 4.3.

## SEÇÃO 4.4

**B, D 4.4\*** Projete o circuito lógico correspondente à tabela-verdade mostrada na Tabela 4.11.

**B, D 4.5** Projete um circuito lógico cuja saída seja nível ALTO *apenas* quando a maioria das entradas  $A$ ,  $B$  e  $C$  for nível BAIXO.

**D 4.6** Uma fábrica precisa de uma sirene para indicar o fim do expediente. A sirene deve ser ativada quando ocorrer uma das seguintes condições:

1. Já passou das cinco horas e todas as máquinas estão desligadas.
2. É sexta-feira, a produção do dia foi atendida e todas as máquinas estão paradas.

Projete um circuito lógico para controle da sirene. (*Sugestão*: use quatro variáveis lógicas de entrada para representar as diversas condições; por exemplo, a entrada  $A$  será nível ALTO apenas quando for 5 horas ou mais.)

**D 4.7\*** Um número de quatro bits é representado como  $A_3A_2A_1A_0$ , em que  $A_3$ ,  $A_2$ ,  $A_1$  e  $A_0$  são os bits individuais e  $A_0$  é o LSB. Projete um circuito lógico

**D 4.8**

**4.9\***

**4.10**

que gere um nível ALTO na saída sempre que o número binário for maior que 0010 e menor que 1000.

A Figura 4.66 mostra um diagrama para um circuito de alarme de automóvel usado para detectar determinada condição indesejada. As três chaves são usadas para indicar, respectivamente, o estado da porta do motorista, da ignição e dos faróis. Projete um circuito lógico com essas três chaves como entrada, de modo que o alarme seja ativado sempre que ocorrer uma das seguintes condições:

- Os faróis estão acesos e a ignição está desligada.
  - A porta está aberta e a ignição está ligada.
- Implemente o circuito do Problema 4.4 usando apenas portas NAND.
- Implemente o circuito do Problema 4.5 usando apenas portas NAND.

TABELA 4.11

$A$	$B$	$C$	$x$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1