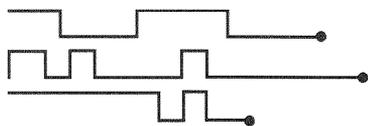


---

# Circuitos Lógicos Combinacionais



## ■ SUMÁRIO

- 4-1 Forma de Soma-de-Produtos
- 4-2 Simplificação de Circuitos Lógicos
- 4-3 Simplificação Algébrica
- 4-4 Projetando Circuitos Lógicos Combinacionais
- 4-5 Método do Mapa de Karnaugh
- 4-6 Circuitos *Exclusive-OR* e *Exclusive-NOR*
- 4-7 Circuitos Gerador e Verificador de Paridade
- 4-8 Circuitos para Habilitar/Desabilitar
- 4-9 Características Básicas de CIs Digitais
- 4-10 Pesquisa de Falhas em Sistemas Digitais
- 4-11 Falhas Internas dos CIs Digitais
- 4-12 Falhas Externas
- 4-13 Estudo de um Caso de Pesquisa de Falhas
- 4-14 Lógica Programável

## OBJETIVOS

Ao completar este capítulo, você deverá estar apto a:

- Converter uma expressão lógica na forma padrão do tipo soma-de-produtos.
- Realizar os passos necessários para obter uma expressão do tipo soma-de-produtos com o objetivo de projetar um circuito lógico correspondente na sua forma mais simples.
- Utilizar a álgebra booleana e o mapa de Karnaugh como ferramentas para simplificação e projeto de circuitos lógicos.
- Explicar o funcionamento dos circuitos *Exclusive-OR* e *Exclusive-NOR*.
- Projetar circuitos lógicos simples sem o auxílio da tabela-verdade.
- Implementar circuitos de habilitação.
- Citar as características básicas de CIs digitais.
- Compreender as diferenças de operação existentes entre circuitos TTL e CMOS.
- Utilizar regras básicas para pesquisa de falhas em sistemas digitais.
- Deduzir, a partir de resultados de medidas, as falhas de funcionamento em circuitos lógicos combinacionais.
- Descrever o princípio fundamental da lógica programável.

## INTRODUÇÃO

No Cap. 3, estudamos a operação de todas as portas lógicas básicas e utilizamos a álgebra booleana para descrever e analisar circuitos que foram feitos a partir da combinação de portas lógicas. Estes circuitos podem ser classificados como circuitos lógicos *combinacionais* porque, em qualquer instante de tempo, o nível lógico da saída do circuito depende da combinação dos níveis lógicos presentes nas entradas. Um circuito combinacional não possui *memória*, e portanto sua saída depende apenas dos valores atuais das entradas.

Neste capítulo continuaremos nosso estudo de circuitos combinacionais, começando por um aprofundamento na simplificação de circuitos lógicos. Dois métodos serão usados: o primeiro utilizará os teoremas da álgebra booleana, e o segundo utilizará uma técnica de *mapeamento*. Além disso, iremos estudar técnicas simples para projetar circuitos lógicos que satisfaçam um dado conjunto de requisitos. Um estudo completo sobre o projeto de circuitos lógicos não é um dos nossos objetivos, mas os métodos que estudaremos proporcionarão uma boa introdução a este assunto.

A última parte deste capítulo trata da pesquisa de falhas em circuitos combinacionais. Esta primeira exposição sobre pesquisa de falhas deve ajudá-lo a desenvolver a capacidade de análise necessária para ser bem-sucedido nesta atividade. De modo a tornar este material o mais prático

possível, primeiro introduziremos algumas características básicas de circuitos integrados de portas lógicas das famílias TTL e CMOS, juntamente com uma descrição dos tipos de falhas mais freqüentemente encontrados em circuitos digitais.

## 4-1 FORMA DE SOMA-DE-PRODUTOS

Os métodos de simplificação e projeto de circuitos lógicos que estudaremos exigem que a expressão esteja na forma de **soma-de-produtos**. Alguns exemplos de expressões deste tipo podem ser vistos a seguir:

1.  $ABC + \overline{ABC}$
2.  $AB + \overline{ABC} + \overline{CD} + D$
3.  $\overline{AB} + \overline{CD} + EF + GK + H\overline{L}$

Cada uma destas expressões do tipo soma-de-produtos consiste em dois ou mais termos AND (produtos) que por sua vez são conectados a uma porta OR. Cada termo AND consiste em uma ou mais variáveis que aparecem *individualmente* na sua forma complementada ou não. Por exemplo, na expressão  $ABC + \overline{ABC}$ , o primeiro produto AND contém as variáveis  $A$ ,  $B$  e  $C$  na sua forma não-complementada (não-invertida). O segundo produto contém  $A$  e  $C$  na sua forma complementada (invertida). Observe que em uma expressão do tipo soma-de-produtos, um sinal de inversão não pode cobrir mais do que uma variável em um termo (por exemplo, não poderíamos ter  $\overline{ABC}$  ou  $\overline{RST}$ ).

### Produto-de-Somas

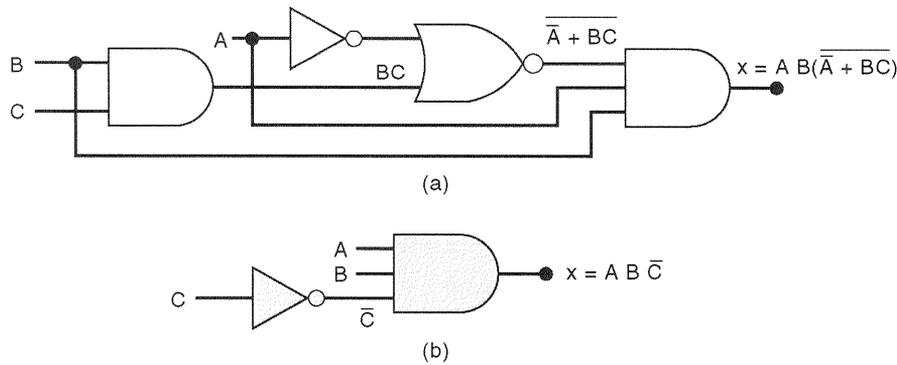
Uma outra forma geral para expressões lógicas às vezes é utilizada no projeto de circuitos lógicos. Ela é chamada de forma de **produto-de-somas**, e consiste em dois ou mais termos OR (somas) que por sua vez são conectados às entradas de uma porta AND. Cada termo OR contém uma ou mais variáveis na sua forma complementada ou não. A seguir, podemos ver algumas expressões do tipo produto-de-somas:

1.  $(A + \overline{B} + C)(A + C)$
2.  $(A + \overline{B})(\overline{C} + D)F$
3.  $(A + C)(B + \overline{D})(\overline{B} + C)(A + \overline{D} + \overline{E})$

Os métodos de simplificação e projeto que serão usados são baseados em expressões do tipo soma-de-produtos, e portanto não utilizaremos muito a forma produto-de-somas. Ela, entretanto, aparecerá em alguns circuitos que têm uma estrutura particular.

### Questões de Revisão

1. Quais das expressões a seguir estão na forma de soma-de-produtos?
  - (a)  $AB + CD + E$
  - (b)  $AB(C + D)$



**Fig. 4-1** Geralmente é possível simplificar um circuito lógico, como o que aparece em (a), e produzir uma implementação mais eficiente, mostrada em (b).

(c)  $(A + B)(C + D + F)$

(d)  $\overline{MN} + PQ$

2. Repita a Questão 1 para produto-de-somas.

## 4-2 SIMPLIFICAÇÃO DE CIRCUITOS LÓGICOS

Uma vez obtida a expressão de um circuito lógico, podemos ser capazes de reduzi-la a uma forma mais simples, que contenha um menor número de termos ou variáveis em um ou mais termos da expressão. Esta nova expressão pode ser usada para implementar um circuito que é equivalente ao circuito original, mas que contém um menor número de portas e conexões.

Para exemplificar, o circuito da Fig. 4-1(a) pode ser simplificado para produzir o circuito da Fig. 4-1(b). Uma vez que os circuitos implementam a mesma lógica, é óbvio que um circuito mais simples é mais desejável porque contém um menor número de portas e portanto será menor e mais barato do que o circuito original. Além disso, a confiabilidade será melhorada porque existe um menor número de ligações, diminuindo assim uma das causas potenciais de falhas no circuito.

Nas seções subseqüentes, estudaremos dois métodos utilizados para simplificar circuitos lógicos. Um dos métodos faz uso dos teoremas da álgebra booleana e, como veremos, é bastante dependente da inspiração e da experiência. O outro método (o mapa de Karnaugh) tem uma abordagem mais sistemática, com instruções passo a passo. Alguns professores podem querer omitir este método porque ele é bastante mecânico e provavelmente não contribui para uma melhor compreensão da álgebra booleana. Isto pode ser feito sem alterar a continuidade ou clareza do restante do texto.

## 4-3 SIMPLIFICAÇÃO ALGÉBRICA

Podemos usar os teoremas da álgebra booleana, que estudamos no Cap. 3, para nos ajudar a simplificar expressões para um circuito lógico. Infelizmente, nem sempre é óbvio qual teorema deve ser aplicado de modo a produzir o re-

sultado mais simples. Além disso, não existe um modo fácil de constatar se a expressão obtida está em sua forma mais simples ou se poderia ser ainda mais simplificada. Portanto, a simplificação algébrica freqüentemente se torna um processo de tentativa e erro. Com a experiência, no entanto, pode-se ficar perito e obter resultados razoavelmente bons.

Os exemplos que se seguem ilustram muitas maneiras pelas quais os teoremas booleanos podem ser aplicados na tentativa de simplificar uma expressão. Você deve notar que estes exemplos contêm dois passos essenciais:

1. A expressão original é colocada sob a forma de soma-de-produtos pela aplicação repetitiva dos teoremas de DeMorgan e pela multiplicação de termos.
2. Uma vez que a expressão original esteja nesta forma, os termos produto são verificados quanto a fatores comuns, realizando-se a fatoração sempre que possível. Com sorte, a fatoração resulta na eliminação de um ou mais termos.

### EXEMPLO 4-1

Simplifique o circuito lógico mostrado na Fig. 4-2(a).

#### Solução

O primeiro passo é determinar a expressão para a saída usando o método apresentado na Seção 3-6. O resultado é

$$z = ABC + A\bar{B} \cdot (\overline{AC})$$

Já com a expressão determinada, usualmente é uma boa idéia quebrar todos os grandes sinais de inversão usando os teoremas de DeMorgan e então multiplicar todos os termos.

$$\begin{aligned} z &= ABC + A\bar{B}(\overline{A + C}) && \text{[teorema (17)]} \\ &= ABC + A\bar{B}(A + C) && \text{[cancela inversões duplas]} \\ &= ABC + A\bar{B}A + A\bar{B}C && \text{[multiplica]} \\ &= ABC + A\bar{B} + A\bar{B}C && \text{[} A \cdot A = A \text{]} \end{aligned}$$

Com a expressão agora sob a forma de soma-de-produtos, devemos procurar por variáveis comuns dentre os vários termos com a intenção de fatorar. O primeiro e terceiro termos têm  $AC$  em comum, que pode ser fatorado:

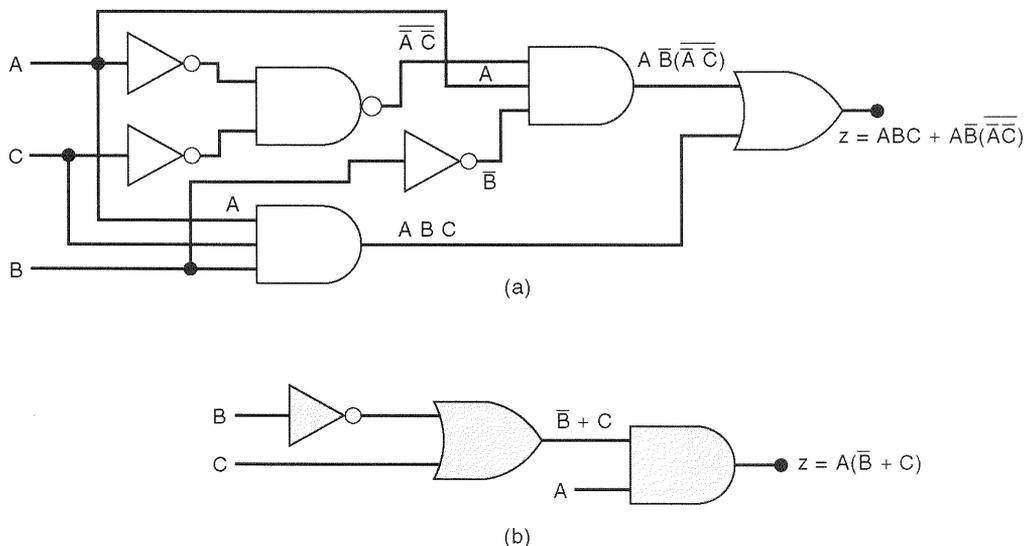


Fig. 4-2 Exemplo 4-1.

$$z = AC(B + \bar{B}) + A\bar{B}$$

Já que  $B + \bar{B} = 1$ , então

$$z = AC(1) + A\bar{B} = AC + A\bar{B}$$

Podemos agora fatorar  $A$ , o que resulta em

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

Este resultado não pode mais ser simplificado. A implementação do circuito é mostrada na Fig. 4-2(b). É óbvio que o circuito em (b) é bem mais simples do que o circuito original em (a).

último termo têm  $AC$  em comum. Como saber se devemos fatorar  $AB$  dos primeiros dois termos ou  $AC$  dos dois termos extremos? Na verdade, podemos fazer ambos usando o termo  $ABC$  duas vezes. Em outras palavras, podemos reescrever a expressão como

$$z = ABC + AB\bar{C} + A\bar{B}C + ABC$$

onde somamos um termo extra  $ABC$ . Isto é válido e não altera o valor da expressão, tendo em vista que  $ABC + ABC = ABC$  [teorema (7)]. Agora podemos fatorar  $AB$  dos dois primeiros termos e  $AC$  dos dois últimos termos:

$$z = AB(C + \bar{C}) + AC(\bar{B} + B) = AB \cdot 1 + AC \cdot 1 = AB + AC = A(B + C)$$

Este é, naturalmente, o mesmo resultado obtido com o método 1. Esse artifício de usar o mesmo termo sempre pode ser usado. De fato, o mesmo termo pode ser usado mais de duas vezes se for necessário.

**EXEMPLO 4-2**

Simplifique a expressão  $z = ABC + AB\bar{C} + A\bar{B}C$ .

**Solução**

Vamos ver dois modos diferentes de chegar ao mesmo resultado.

*Método 1:* Os primeiros dois termos na expressão têm o produto  $AB$  em comum. Logo,

$$z = AB(C + \bar{C}) + A\bar{B}C = AB(1) + A\bar{B}C = AB + A\bar{B}C$$

Podemos fatorar a variável  $A$  de ambos os termos:

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

Aplicando o teorema (15),

$$z = A(B + C)$$

*Método 2:* A expressão original é  $z = ABC + AB\bar{C} + A\bar{B}C$ . Os primeiros dois termos têm  $AB$  em comum. O primeiro e o

**EXEMPLO 4-3**

Simplifique  $z = \bar{A}C(\overline{ABD}) + \bar{A}B\bar{C}\bar{D} + A\bar{B}C$ .

**Solução**

Inicialmente, use o teorema de DeMorgan no primeiro termo:

$$z = \bar{A}C(A + \bar{B} + \bar{D}) + \bar{A}B\bar{C}\bar{D} + A\bar{B}C \quad (\text{passo 1})$$

Multiplicando-se obtemos

$$z = \bar{A}CA + \bar{A}C\bar{B} + \bar{A}C\bar{D} + \bar{A}B\bar{C}\bar{D} + A\bar{B}C \quad (2)$$

Visto que  $\bar{A} \cdot A = 0$ , o primeiro termo é eliminado:

$$z = \bar{A}\bar{B}C + \bar{A}C\bar{D} + \bar{A}B\bar{C}\bar{D} + A\bar{B}C \quad (3)$$

Esta é a forma de soma-de-produtos desejada. Agora devemos procurar por fatores comuns dentre os vários termos produto. A idéia é investigar o maior fator comum entre quaisquer dois ou mais termos produto. Por exemplo, o primeiro e o último termo têm o fator comum  $\overline{B}C$ , e o segundo e o terceiro termo compartilham o fator comum  $\overline{A}\overline{D}$ . Podemos fatorá-los como se segue:

$$z = \overline{B}C(\overline{A} + A) + \overline{A}\overline{D}(C + B\overline{C}) \quad (4)$$

Agora, sabendo que  $\overline{A} + A = 1$ , e  $C + B\overline{C} = C + B$  [teorema (15)], temos

$$z = \overline{B}C + \overline{A}\overline{D}(B + C) \quad (5)$$

Este mesmo resultado teria sido alcançado com outras escolhas para fatoração. Por exemplo, poderíamos ter fatorado  $C$  do primeiro, segundo e quarto termos produto, no passo 3, para obter

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

A expressão entre parênteses pode ser fatorada ainda mais:

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

Visto que  $\overline{A} + A = 1$ , ela se torna

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

Multiplicando obtém-se

$$z = \overline{B}C + \overline{A}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}\overline{D}$$

Agora podemos fatorar  $\overline{A}\overline{D}$  do segundo e terceiro termos para obter

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

Usando o teorema (15), a expressão entre parênteses se torna  $B + C$ . Assim, finalmente temos

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

Este é o mesmo resultado que obtivemos antes, mas exigiu muito mais passos. Isto ilustra por que devemos procurar pelos maiores fatores comuns: geralmente levará até a expressão final em menos passos.

#### EXEMPLO 4-4

Simplifique a expressão  $x = (\overline{A} + B)(A + B + D)\overline{D}$ .

#### Solução

A expressão pode ser colocada sob a forma de soma-de-produtos multiplicando-se todos os termos. O resultado é

$$x = \overline{A}\overline{A}\overline{D} + \overline{A}B\overline{D} + \overline{A}D\overline{D} + BA\overline{D} + BB\overline{D} + BD\overline{D}$$

O primeiro termo pode ser eliminado, já que  $\overline{A}\overline{A} = 0$ . Do mesmo modo, o terceiro e o sexto termo podem ser eliminados, visto que  $D\overline{D} = 0$ . O quinto termo pode ser simplificado para  $B\overline{D}$ , já que  $BB = B$ . Isto resulta em

$$x = \overline{A}B\overline{D} + AB\overline{D} + B\overline{D}$$

Podemos fatorar  $B\overline{D}$  de cada termo para obter

$$x = B\overline{D}(\overline{A} + A + 1)$$

É claro que o termo entre parênteses é sempre 1, portanto, finalmente temos

$$x = B\overline{D}$$

#### EXEMPLO 4-5

Simplifique o circuito da Fig. 4-3(a).

#### Solução

A expressão para a saída  $z$  é

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

Multiplicando para conseguir a forma de soma-de-produtos, obtemos

$$z = \overline{A}A + \overline{A}\overline{B} + BA + B\overline{B}$$

Podemos eliminar  $\overline{A}A = 0$  e  $B\overline{B} = 0$  para terminar com

$$z = \overline{A}\overline{B} + AB$$

Esta expressão está implementada na Fig. 4-3(b), e se for comparada com o circuito original vemos que ambos os circuitos contêm o mesmo número de portas e conexões. Neste caso o processo de simplificação produziu um circuito equivalente, mas não um circuito mais simples.

#### EXEMPLO 4-6

Simplifique  $x = \overline{A}\overline{B}C + \overline{A}BD + \overline{C}\overline{D}$ .

#### Solução

Você pode tentar, mas não será capaz de simplificar ainda mais esta expressão.

#### Questões de Revisão

1. Indique quais das seguintes expressões *não* estão sob a forma de soma-de-produtos:

(a)  $RS\overline{T} + \overline{R}S\overline{T} + \overline{T}$

(b)  $A\overline{D}\overline{C} + \overline{A}DC$

(c)  $MN\overline{P} + (M + \overline{N})P$

(d)  $AB + \overline{A}B\overline{C} + \overline{A}\overline{B}\overline{C}D$

2. Simplifique o circuito na Fig. 4-1(a) para chegar ao circuito da Fig. 4-1(b).

3. Troque cada porta AND na Fig. 4-1(a) por uma porta NAND. Determine a nova expressão de  $x$  e simplifique-a.

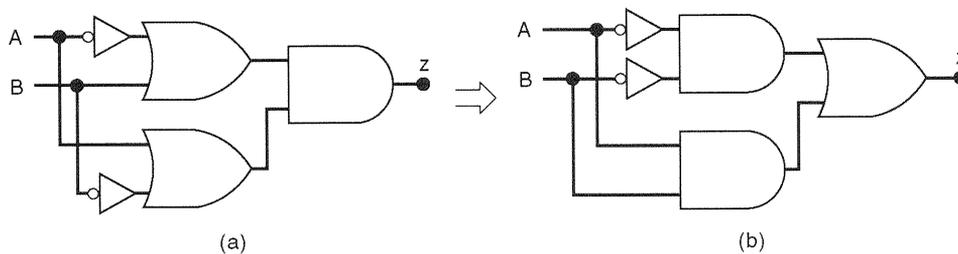


Fig. 4-3 Exemplo 4-5.

### 4-4 PROJETANDO CIRCUITOS LÓGICOS COMBINACIONAIS

Quando o nível de saída desejado de um circuito lógico é dado para todas as condições de entrada possíveis, os resultados podem ser convenientemente apresentados em uma tabela-verdade. A expressão booleana para o circuito pode ser derivada da tabela-verdade. Por exemplo, considere a Fig. 4-4(a), onde uma tabela-verdade é mostrada para um circuito que tem duas entradas,  $A$  e  $B$ , e uma saída  $x$ . A tabela mostra que a saída  $x$  está no nível 1 *somente* para o caso em que  $A = 0$  e  $B = 1$ . Agora, resta determinar que circuito lógico produz esta operação. Deveria estar claro que uma solução possível é apresentada na Fig. 4-4(b). Nela, uma porta AND é usada com entradas  $\bar{A}$  e  $B$ , de modo que  $x =$

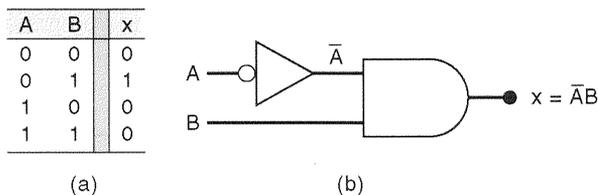


Fig. 4-4 Circuito que produz nível 1 na saída somente para a condição  $A = 0, B = 1$ .

$\bar{A} \cdot B$ . Obviamente,  $x$  será 1 *somente se* ambas as entradas da porta AND forem 1, isto é,  $\bar{A} = 1$  (o que significa  $A = 0$ ) e  $B = 1$ . Para todos os outros valores de  $A$  e  $B$ , a saída  $x$  deve ser 0.

Uma abordagem similar pode ser usada para outras condições de entrada. Por exemplo, se  $x$  tivesse que estar em alto somente para a condição  $A = 1, B = 0$ , o circuito resultante deveria ser uma porta AND com entradas  $A$  e  $\bar{B}$ . Em outras palavras, para qualquer uma das quatro possíveis condições de entrada podemos gerar uma saída alta  $x$  utilizando uma porta AND, com entradas apropriadas, para gerar o produto AND requerido. Os quatro casos distintos são mostrados na Fig. 4-5. Cada porta AND gera uma saída que é 1 *somente* para uma certa condição de entrada, e a saída é 0 para todas as outras condições. Deve-se notar que as entradas da AND são invertidas ou não, dependendo dos valores que as variáveis têm para a condição dada. Se a variável é 0 para a condição dada, ela é invertida antes de entrar na porta AND.

Vamos agora considerar o caso mostrado na Fig. 4-6(a), onde temos uma tabela-verdade indicando que a saída  $x$  deve ser 1 para dois casos distintos:  $A = 0, B = 1$  e  $A = 1, B = 0$ . Como isto pode ser implementado? Sabemos que o termo AND  $\bar{A} \cdot B$  gera 1 somente para a condição  $A = 0, B = 1$ , e o termo AND  $A \cdot \bar{B}$  gera 1 para a condição  $A = 1, B = 0$ . Como  $x$  deve ser ALTO para uma *ou* outra condição, deve ficar claro que estes termos devem ser unidos com OR para produzirem a saída desejada  $x$ . Esta implementação é

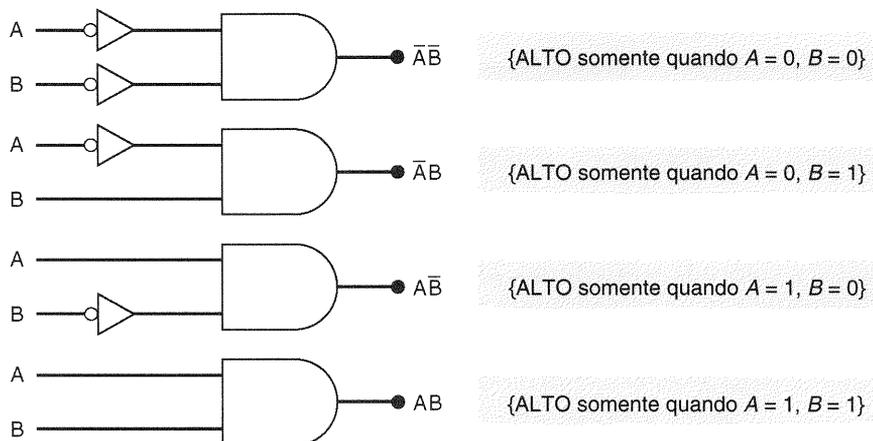
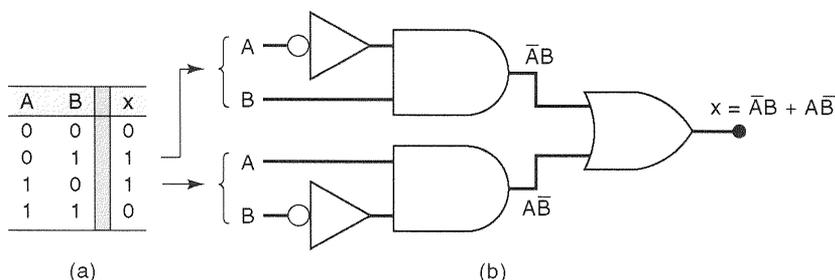


Fig. 4-5 Uma porta AND, com entradas apropriadas, pode ser usada para produzir uma saída em 1 para um conjunto específico de níveis de entrada.



**Fig. 4-6** Cada conjunto de condições de entrada que produz uma saída em ALTO é implementado por uma porta AND em separado. As saídas das portas AND são unidas com OR para produzir a saída final.

mostrada na Fig. 4-6(b), onde a expressão resultante para a saída é  $x = \bar{A}B + A\bar{B}$ .

Neste exemplo, um termo AND é gerado para cada caso na tabela onde a saída deve ser 1. As saídas das portas AND são então unidas com OR para produzir a saída  $x$ , que será 1 quando um dos termos AND for 1. Este mesmo procedimento pode ser estendido para exemplos com mais de duas entradas. Considere a tabela-verdade para um circuito de três entradas (Tabela 4-1). Nela existem três casos onde a saída  $x$  deve ser 1. O termo AND para cada caso está indicado. Novamente, observe que para cada caso onde a variável é 0 ela aparece complementada no termo AND. A expressão de soma-de-produtos para  $x$  é obtida unindo com OR os três termos AND.

$$x = \bar{A}B\bar{C} + \bar{A}BC + ABC$$

TABELA 4-1

A	B	C	x
0	0	0	0
0	0	1	0
0	1	0	1 $\rightarrow \bar{A}B\bar{C}$
0	1	1	1 $\rightarrow \bar{A}BC$
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1 $\rightarrow ABC$

### Procedimento Completo de Projeto

Uma vez que a expressão da saída tenha sido determinada da tabela-verdade sob a forma de soma-de-produtos, ela pode ser facilmente implementada usando portas AND, OR e INVERSORES. Usualmente, entretanto, a expressão pode ser simplificada, resultando num circuito mais eficiente. O exemplo seguinte ilustra o procedimento completo de projeto.

#### EXEMPLO 4-7

Projete um circuito lógico que tem três entradas,  $A$ ,  $B$  e  $C$ , e cuja saída vai para ALTO somente quando a maioria das entradas está em ALTO.

### Solução

**Passo 1.** Monte a tabela-verdade.

Com base no enunciado do problema, a saída  $x$  deve ser 1 sempre que duas ou mais entradas forem 1. Para todos os outros casos, a saída deve ser 0 (Tabela 4-2).

TABELA 4-2

A	B	C	x
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1 $\rightarrow \bar{A}BC$
1	0	0	0
1	0	1	1 $\rightarrow A\bar{B}C$
1	1	0	1 $\rightarrow AB\bar{C}$
1	1	1	1 $\rightarrow ABC$

**Passo 2.** Escreva o termo AND para cada caso onde a saída é 1.

Existem quatro destes casos. Os termos AND estão mostrados próximos à tabela (Tabela 4-2). Note mais uma vez que cada termo AND contém cada variável de entrada, invertida ou não.

**Passo 3.** Escreva a expressão da soma-de-produtos para a saída.

$$x = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

**Passo 4.** Simplifique a expressão de saída.

Esta expressão pode ser simplificada de muitos modos. Talvez o modo mais rápido seja reparar que o último termo  $ABC$  tem duas variáveis em comum com cada um dos outros termos. Logo, podemos usar o termo  $ABC$  para fatorar com cada um dos outros. A expressão é reescrita com o

termo  $ABC$  aparecendo três vezes (lembre-se do Exemplo 4-2 que isto é permitido em álgebra booleana):

$$x = \bar{A}BC + ABC + A\bar{B}C + ABC + AB\bar{C} + ABC$$

Fatorando os pares de termos apropriados, temos

$$x = BC(\bar{A} + A) + AC(\bar{B} + B) + AB(\bar{C} + C)$$

Visto que cada termo entre parênteses é igual a 1, temos

$$x = BC + AC + AB$$

**Passo 5.** Implemente o circuito para a expressão final.

Esta expressão está implementada na Fig. 4-7. Como a expressão está sob a forma de soma-de-produtos, o circuito consiste em um grupo de portas AND ligadas em uma única porta OR.

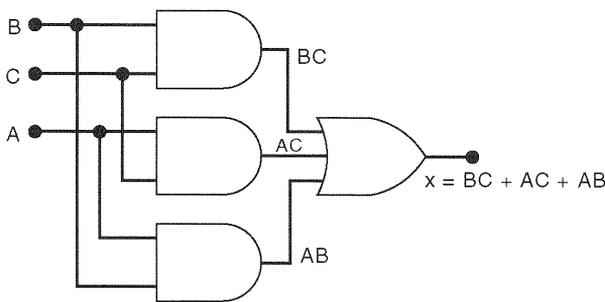


Fig. 4-7 Exemplo 4-7.

**EXEMPLO 4-8**

Veja a Fig. 4-8(a), onde um conversor analógico-digital está monitorando a tensão de uma bateria de 12 V de uma espaçonave em órbita. A saída do conversor é um número binário de quatro bits,  $ABCD$ , que corresponde à tensão da

bateria em degraus de 1 V, sendo  $A$  o MSB. As saídas binárias do conversor são ligadas em um circuito digital que deve produzir uma saída em ALTO sempre que o valor binário for maior do que  $0110_2 = 6_{10}$ , ou seja, quando a tensão da bateria for maior do que 6 V. Projete este circuito lógico.

**Solução**

A tabela-verdade é mostrada na Fig. 4-8(b). Para cada linha da tabela-verdade indicamos o equivalente decimal do número representado pela combinação  $ABCD$ .

A saída  $z$  é igual a 1 para todos os casos onde o número binário é maior do que 0110. Para todos os outros casos,  $z$  é igual a 0. Esta tabela-verdade fornece a seguinte expressão de soma-de-produtos:

$$z = \bar{A}BCD + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}BCD + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + \bar{A}BCD$$

Simplificar esta expressão é uma tarefa tremenda, mas com um pouco de cuidado ela pode ser feita. O processo passo a passo envolve fatorar e eliminar termos da forma  $A + \bar{A}$ :

$$\begin{aligned} z &= \bar{A}BCD + \bar{A}\bar{B}\bar{C}(\bar{D} + D) + \bar{A}\bar{B}C(\bar{D} + D) + \bar{A}B\bar{C}(\bar{D} + D) + \bar{A}BCD \\ &= \bar{A}BCD + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC \\ &= \bar{A}BCD + \bar{A}\bar{B}(\bar{C} + C) + \bar{A}B(\bar{C} + C) \\ &= \bar{A}BCD + \bar{A}\bar{B} + \bar{A}B \\ &= \bar{A}BCD + A(\bar{B} + B) \\ &= \bar{A}BCD + A \end{aligned}$$

Isto pode ainda ser reduzido aplicando-se o teorema (15), que é:  $x + \bar{x}y = x + y$ . Neste caso  $x = A$  e  $y = BCD$ . Logo,

$$z = \bar{A}BCD + A = BCD + A$$

Esta expressão final está implementada na Fig. 4-8(c).

Como esse exemplo demonstra, o método da simplificação algébrica pode ser maçante quando a expressão original contém um grande número de termos. Esta é uma limitação que não é partilhada pelo método do mapa de Karnaugh, como veremos posteriormente.

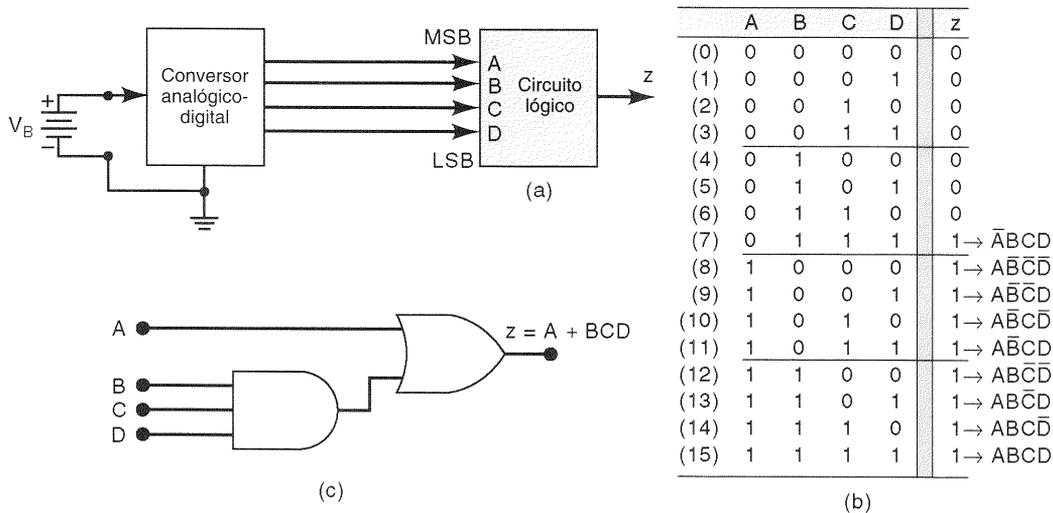


Fig. 4-8 Exemplo 4-8.

## Implementando o Projeto Final

Nos exemplos de projetos apresentados, o circuito final foi implementado usando-se portas AND e OR. De fato, a forma de soma-de-produtos sempre produz um circuito que contém uma ou mais portas AND acionando uma porta OR. Uma das razões para a utilização da forma de soma-de-produtos é que ela pode ser implementada usando-se apenas portas NAND com pouco, ou nenhum, aumento de complexidade em relação à implementação AND/OR. Tendo em vista que as portas NAND são as portas lógicas mais disponíveis na família lógica TTL, esta é uma característica importante.

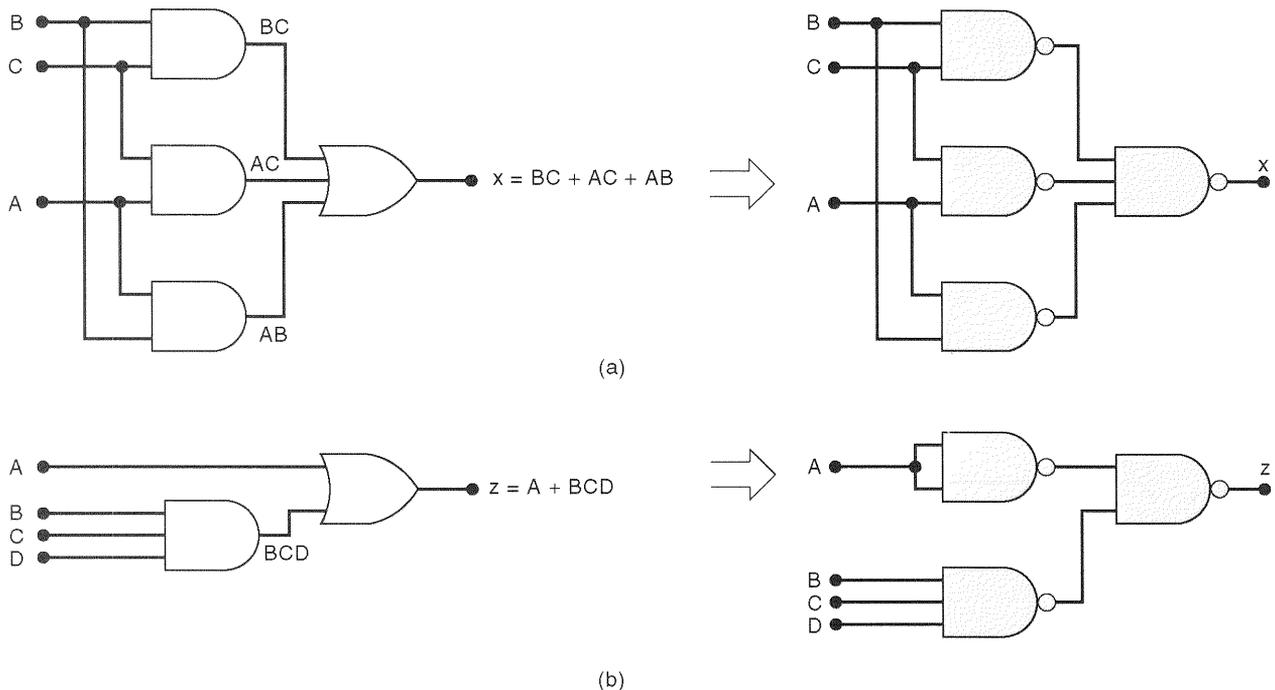
Para ilustrar, a Fig. 4-9 mostra as implementações equivalentes com portas NAND para os circuitos das Figs. 4-7 e 4-8(c). Você pode fazer essas conversões como revisão do procedimento apresentado no Cap. 3.

Comparando a implementação NAND com o circuito original na Fig. 4-9(a), observa-se que eles são idênticos na estrutura, isto é, *cada* porta do circuito original foi substituída por uma única porta NAND. Esta característica é verdadeira somente se o circuito original está sob a forma de soma-de-produtos. A única exceção é quando a forma de soma-de-produtos contém um termo de uma variável tal como  $z = A + BCD$  na Fig. 4-9(b). Neste caso a implementação NAND requer uma porta NAND extra usada como INVERSOR na entrada A.

Podemos resumir o processo de conversão de um circuito sob a forma de *soma-de-produtos* com portas AND/OR para portas NAND como se segue:

1. Substitua cada porta AND, porta OR e INVERSOR por uma *única* porta NAND.
2. Utilize uma porta NAND para inverter qualquer variável simples que aciona a porta OR final.

Verifique este processo para os circuitos na Fig. 4-9.



**Fig. 4-9** (a) Conversão do circuito da Fig. 4-7 para NANDs; (b) conversão do circuito da Fig. 4-8(c) para NANDs.

## EXEMPLO 4-9

Veja a Fig. 4-10(a). Numa máquina copiadora simples, um sinal de parada,  $S$ , deve ser gerado para interromper a operação da máquina e energizar uma luz indicadora, sempre que uma das seguintes condições existir: (1) a bandeja de alimentação de papel estiver vazia; ou (2) as duas chaves na trajetória do papel estiverem ativadas, indicando um congestionamento no caminho do papel. A presença de papel na bandeja de alimentação é indicada por um sinal lógico  $P$  em ALTO. Cada chave produz um sinal lógico ( $Q$  e  $R$ ) que vai para ALTO sempre que o papel passa sobre a chave para ativá-la. Projete o circuito lógico para produzir um nível ALTO no sinal de saída  $S$  para as condições estabelecidas, e implemente-o usando o chip 74LS00 (Fig. 3-31).

## Solução

Utilizaremos o processo de cinco passos usado no Exemplo 4-7.

A tabela-verdade está na Tabela 4-3. A saída  $S$  assume o valor lógico 1 sempre que  $P = 0$ , já que isto indica que não há papel na bandeja de alimentação.  $S$  também é 1 para os dois casos em que  $Q$  e  $R$  são ambos 1, indicando um congestionamento de papel. Conforme a tabela mostra, existem cinco diferentes condições de entrada que produzem uma saída em ALTO. (Passo 1)

Os termos AND para cada um dos casos estão indicados. (Passo 2)

A expressão da soma-de-produtos se torna

$$S = \overline{P}\overline{Q}\overline{R} + \overline{P}\overline{Q}R + \overline{P}Q\overline{R} + \overline{P}QR + PQR \quad (\text{Passo 3})$$

Podemos começar a simplificação fatorando  $\bar{P}\bar{Q}$  dos termos 1 e 2 e fatorando  $\bar{P}Q$  dos termos 3 e 4:

$$S = \bar{P}\bar{Q}(\bar{R} + R) + \bar{P}Q(\bar{R} + R) + PQR \quad (\text{Passo 4})$$

Agora podemos eliminar os termos  $\bar{R} + R$ , já que são iguais a 1:

$$S = \bar{P}\bar{Q} + \bar{P}Q + PQR$$

Fatorar  $\bar{P}$  dos termos 1 e 2 permite a eliminação de  $Q$  destes termos:

$$S = \bar{P} + PQR$$

Aplicando o teorema (15) ( $x + \bar{x}y = x + y$ ) obtemos

$$S = \bar{P} + QR$$

TABELA 4-3

P	Q	R	S
0	0	0	1 $\bar{P}\bar{Q}\bar{R}$
0	0	1	1 $\bar{P}\bar{Q}R$
0	1	0	1 $\bar{P}Q\bar{R}$
0	1	1	1 $\bar{P}QR$
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1 $PQR$

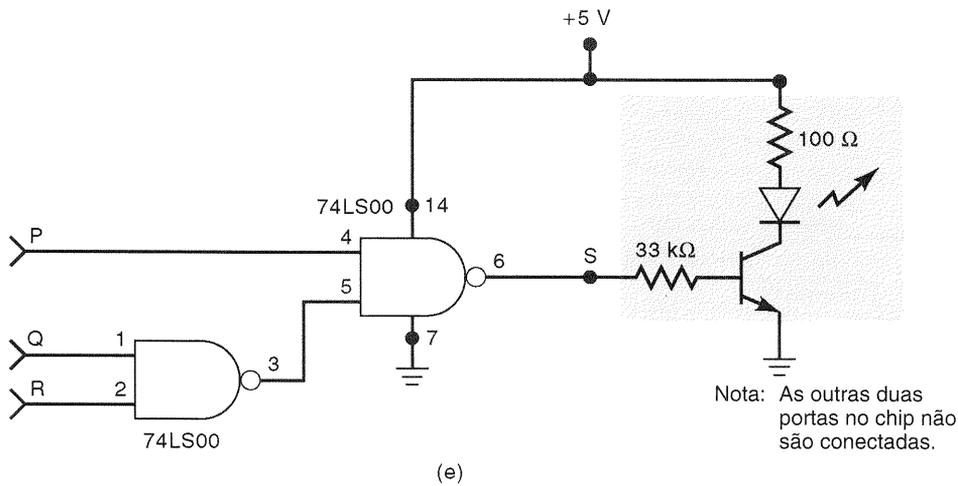
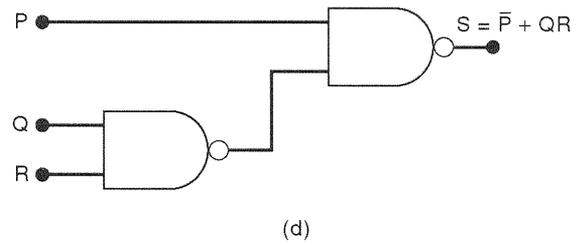
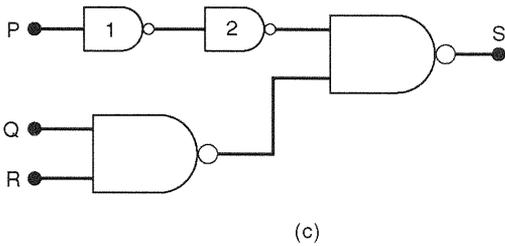
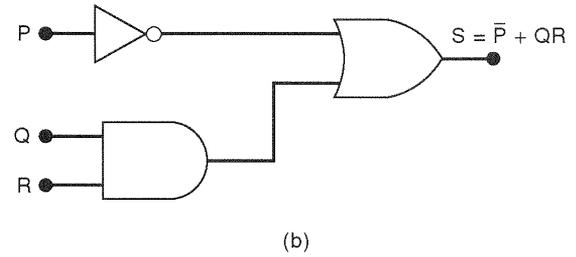
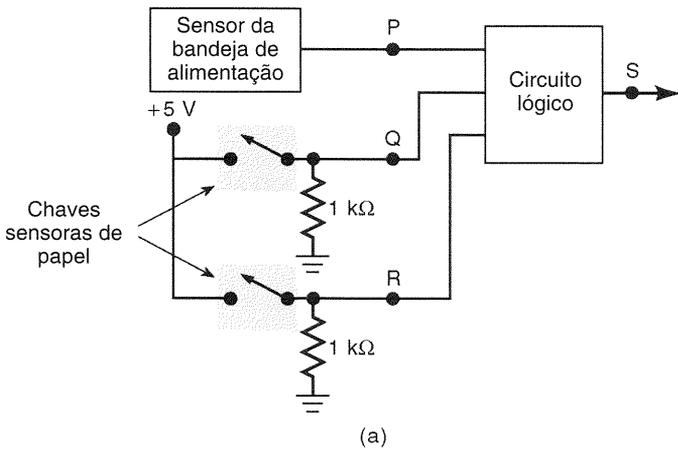


Fig. 4-10 Procedimento completo de projeto (Exemplo 4-9) implementado usando um chip NAND 74LS00.

A implementação AND/OR para este circuito está ilustrada na Fig. 4-10(b). (Passo 5)

Tendo em vista que o circuito deve ser implementado com o chip 74LS00, que tem quatro portas NAND de duas entradas, o circuito da Fig. 4-10(b) deve ser convertido para utilizar apenas portas NAND. Substitui-se cada porta OR e AND por uma porta NAND e troca-se o INVERSOR pela porta NAND INVERSORa [identificada com 1 na Fig. 4-10(c)]. Além disso, como a entrada superior da porta OR é uma variável simples ( $P$ ), uma porta NAND INVERSORa (identificada com 2) deve ser colocada nessa entrada. Obviamente, os dois INVERSORes podem ser eliminados para obter o circuito com NANDs da Fig. 4-10(d).

A Fig. 4-10(e) é a versão final do circuito mostrando a pinagem do CI, incluindo os pinos de alimentação (+5 V e TERRA) e o transistor de acionamento de saída com o LED indicador para o sinal  $S$ .

### Questões de Revisão

1. Escreva a expressão de soma-de-produtos para um circuito com quatro entradas e uma saída que deve estar em ALTO somente quando a entrada  $A$  está em BAIXO, ao mesmo tempo em que exatamente duas outras entradas estão em BAIXO.
2. Implemente a expressão da Questão 1 usando apenas portas NAND de quatro entradas. Quantas são necessárias?

## 4-5 MÉTODO DO MAPA DE KARNAUGH

O **mapa de Karnaugh** é um método gráfico usado para simplificar uma equação lógica ou para converter uma tabela-verdade no seu circuito lógico correspondente, de um modo simples e ordenado. Embora um mapa de Karnaugh (daqui para a frente abreviado como **mapa K**) possa ser usado em problemas que envolvem qualquer número de variáveis de entrada, sua utilidade prática está limitada a seis variáveis. A apresentação que se segue está restrita a problemas com até quatro entradas, pois mesmo os problemas com cinco ou seis entradas são demasiadamente complicados, sendo mais bem resolvidos por um programa de computador.

### Formato do Mapa de Karnaugh

O mapa K, como uma tabela-verdade, é um meio de mostrar a relação entre as entradas lógicas e a saída desejada. A Fig. 4-11 apresenta três exemplos de mapas K, para duas, para três e para quatro variáveis, em conjunto com as tabelas-verdade correspondentes. Estes exemplos ilustram os seguintes pontos importantes:

1. A tabela-verdade fornece o valor da saída  $X$  para cada combinação de valores da entrada. O mapa K fornece a mesma informação num formato diferente. Cada linha na tabela-verdade corresponde a um quadrado no mapa K. Por exemplo, na Fig. 4-11(a), a condição  $A = 0, B = 0$ , na tabela-verdade, corresponde ao quadrado  $\overline{A}\overline{B}$  no mapa K. Como a tabela-verdade mostra  $X = 1$  para este caso, 1

é colocado no quadrado  $\overline{A}\overline{B}$  no mapa K. Do mesmo modo, a condição  $A = 1, B = 1$  na tabela-verdade corresponde ao quadrado  $AB$  no mapa K. Como  $X = 1$  para este caso, 1 é colocado no quadrado  $AB$ . Todos os outros quadrados são preenchidos com 0s. Esta mesma idéia é usada nos mapas de três e quatro variáveis mostrados na figura.

2. Os quadrados do mapa K são identificados de modo que quadrados adjacentes horizontalmente diferem apenas em uma variável. Por exemplo, o quadrado do canto superior esquerdo no mapa de quatro variáveis é  $\overline{A}\overline{B}\overline{C}\overline{D}$ , enquanto o quadrado imediatamente à sua direita é  $\overline{A}\overline{B}C\overline{D}$  (apenas a variável  $D$  é diferente). Do mesmo modo, quadrados adjacentes verticalmente diferem apenas em uma variável. Por exemplo, o quadrado do canto superior esquerdo no mapa de quatro variáveis é  $\overline{A}\overline{B}\overline{C}\overline{D}$ , enquanto o quadrado diretamente abaixo dele é  $\overline{A}\overline{B}C\overline{D}$  (apenas a variável  $B$  é diferente).

Note que cada quadrado na linha superior é considerado adjacente ao quadrado correspondente na linha inferior. Por exemplo, o quadrado  $\overline{A}\overline{B}C\overline{D}$  na linha superior é adjacente ao quadrado  $\overline{A}\overline{B}C\overline{D}$  na linha inferior, pois diferem apenas na variável  $A$ . Você pode imaginar que a parte superior do mapa foi dobrada para tocar a parte inferior. Analogamente, os quadrados da coluna mais à esquerda são adjacentes aos quadrados correspondentes da coluna mais à direita.

3. Para que os quadrados adjacentes, tanto na horizontal quanto na vertical, difiram em apenas uma variável, a identificação de cima para baixo deve ser feita na ordem mostrada:  $\overline{A}\overline{B}, \overline{A}B, AB, A\overline{B}$ . O mesmo se aplica à identificação da esquerda para a direita.
4. Uma vez que um mapa K foi preenchido com 0s e 1s, a expressão da soma-de-produtos para a saída  $X$  pode ser obtida juntando-se com OR os quadrados que contêm 1. No mapa de três variáveis da Fig. 4-11(b), os quadrados  $\overline{A}\overline{B}C, \overline{A}B\overline{C}, \overline{A}B\overline{C}$  e  $AB\overline{C}$  contêm 1, portanto  $X = \overline{A}\overline{B}C + \overline{A}B\overline{C} + \overline{A}B\overline{C} + AB\overline{C}$ .

### Agrupamento de Termos no Mapa

A expressão para a saída  $X$  pode ser simplificada combinando-se adequadamente os quadrados no mapa K que contêm 1. O processo de combinar estes 1s é chamado de **agrupamento**.

### Agrupando Dois Termos (Pares)

Na Fig. 4-12(a) está o mapa K para uma determinada tabela-verdade de três variáveis. Este mapa contém um par de 1s que são adjacentes na vertical: o primeiro representa  $\overline{A}\overline{B}C$  e o segundo representa  $AB\overline{C}$ . Repare que nestes dois termos apenas a variável  $A$  aparece tanto na forma normal quanto na complementar ( $B$  e  $\overline{C}$  permanecem inalteradas). Estes dois termos podem ser agrupados (combinados) para dar um resultado que elimina a variável  $A$ , visto que ela aparece em ambas as formas, normal e complementar. Isto é facilmente provado como se segue:

$$\begin{aligned} X &= \overline{A}\overline{B}C + AB\overline{C} \\ &= \overline{B}\overline{C}(\overline{A} + A) \\ &= \overline{B}\overline{C}(1) = \overline{B}\overline{C} \end{aligned}$$

Este mesmo princípio permanece válido para qualquer par de 1s adjacentes na vertical ou na horizontal. A Fig. 4-12(b) mostra um exemplo de dois 1s horizontalmente adja-

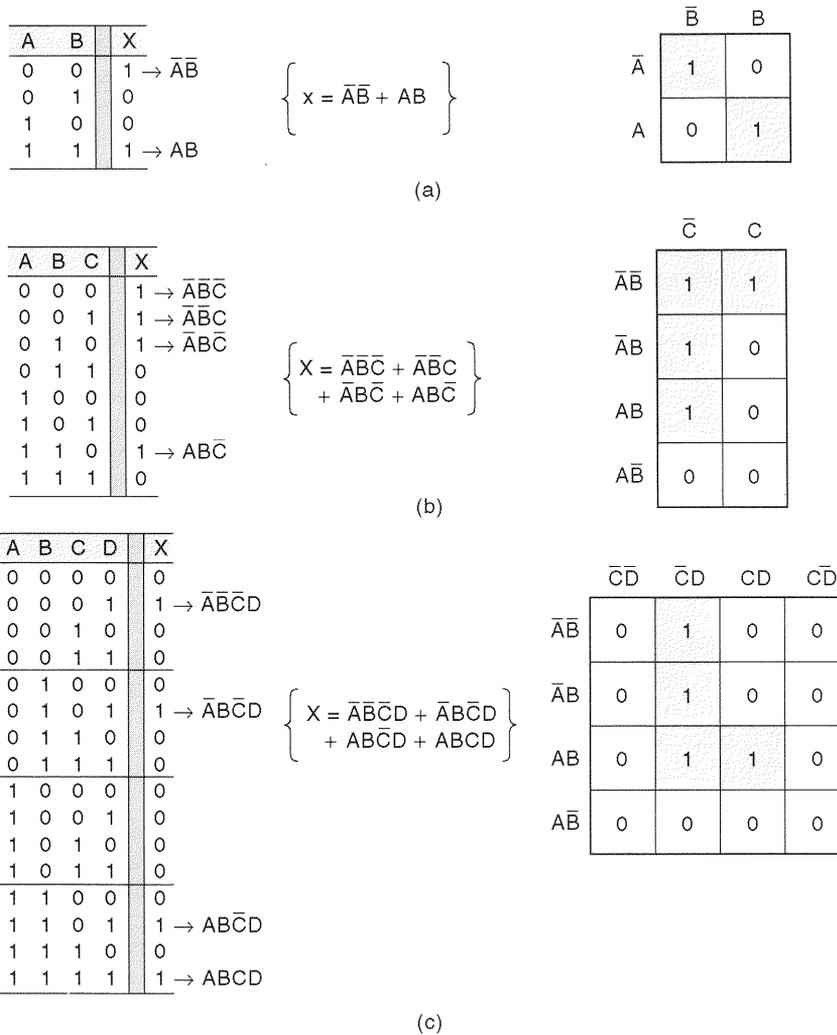


Fig. 4-11 Mapas de Karnaugh e tabelas-verdade para (a) duas, (b) três e (c) quatro variáveis.

centes. Estes dois podem ser agrupados e a variável *C* eliminada, já que ela aparece nas formas não-complementada e complementada para resultar em  $X = \bar{A}B$ .

Um outro exemplo está ilustrado na Fig. 4-12(c). Num mapa K a linha superior e a linha inferior são consideradas adjacentes. Assim, os dois 1s neste mapa podem ser agrupados para produzir como resultado  $\bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} = \bar{B}\bar{C}$ .

A Fig. 4-12(d) mostra um mapa K que tem dois pares de 1s que podem ser agrupados. Os dois 1s na linha superior são horizontalmente adjacentes. Os dois 1s na linha inferior também são adjacentes, já que, em um mapa K, a coluna de quadrados mais à esquerda é considerada adjacente com a coluna mais à direita. Quando o par de 1s superior é agrupado, a variável *D* é eliminada (pois ela aparece tanto como *D* quanto como  $\bar{D}$ ) para gerar o termo  $\bar{A}\bar{B}C$ . Agrupar o par inferior elimina a variável *C* para gerar o termo  $A\bar{B}\bar{D}$ . Estes dois termos são unidos por um OR, obtendo-se o resultado final para *X*.

Resumindo:

**Agrupar um par de 1s adjacentes num mapa K elimina a variável que aparece nas formas complementada e não-complementada.**

### Agrupando Quatro Termos (Quartetos)

Um mapa K pode conter um grupo de quatro 1s adjacentes entre si. Este grupo é denominado *quarteto*. A Fig. 4-13 mostra vários exemplos de quartetos. Na parte (a) os quatro 1s são verticalmente adjacentes, e na parte (b) eles são adjacentes na horizontal. O mapa K na Fig. 4-13(c) contém quatro 1s formando um quadrado, e eles são considerados adjacentes entre si. Os quatro 1s na Fig. 4-13(d) também são adjacentes, assim como os da Fig. 4-13(e) porque, conforme apresentado anteriormente, as linhas superior e inferior são consideradas adjacentes entre si, do mesmo modo que as colunas mais à esquerda e mais à direita.

Quando um quarteto é agrupado, o termo resultante contém apenas as variáveis que não mudam de forma para todos os quadrados do quarteto. Por exemplo, na Fig. 4-13(a), os quatro quadrados que contêm 1 são  $\bar{A}\bar{B}C$ ,  $\bar{A}BC$ ,  $ABC$  e  $A\bar{B}C$ . Um exame destes termos revela que apenas a variável *C* permanece inalterada (tanto *A* como *B* aparecem nas formas não-complementada e complementada). Assim, a expressão resultante para *X* é simplesmente  $X = C$ . Isto pode ser provado como se segue:

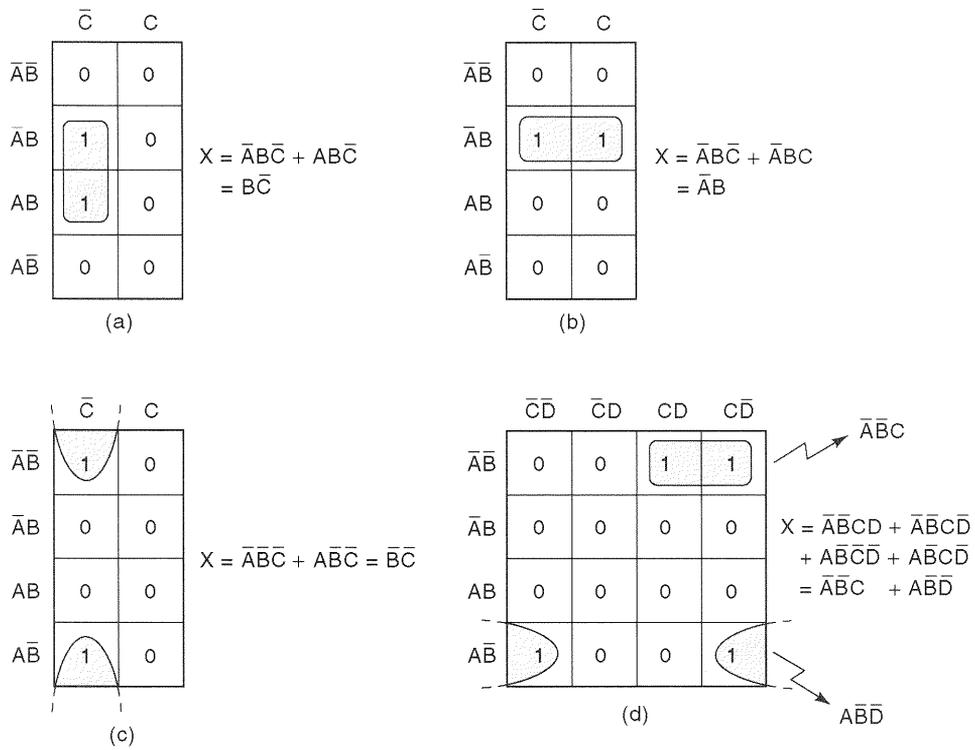


Fig. 4-12 Exemplos de agrupamentos de pares de 1s adjacentes.

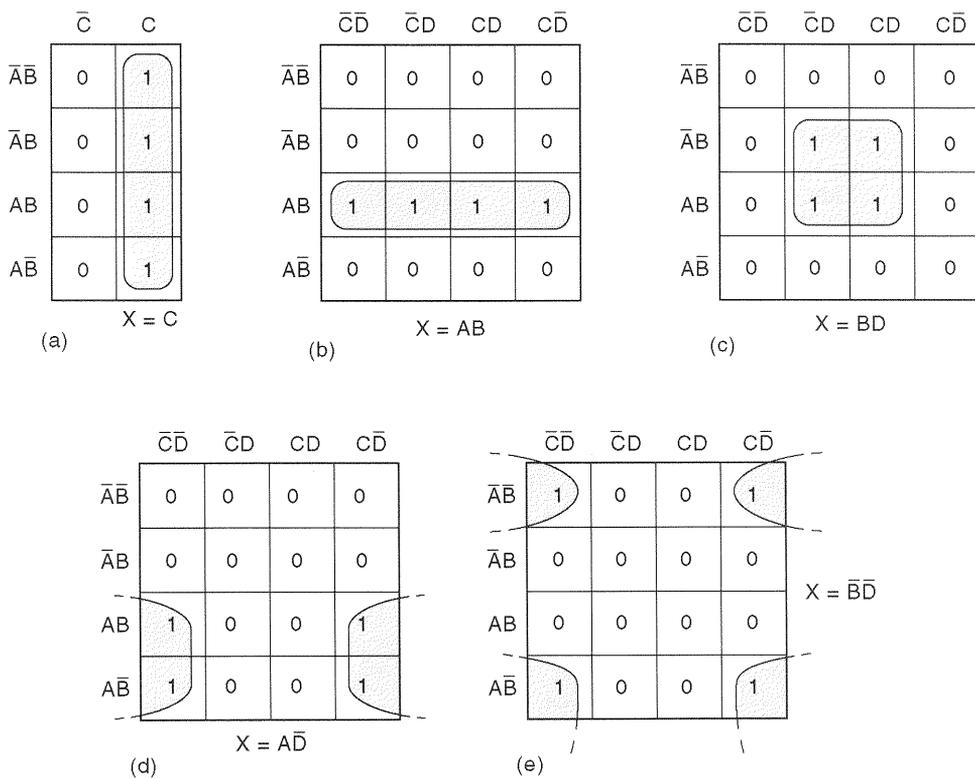


Fig. 4-13 Exemplos de agrupamentos de quatro 1s (quartetos).

$$\begin{aligned}
 X &= \overline{A}\overline{B}C + \overline{A}BC + ABC + A\overline{B}C \\
 &= \overline{A}C(\overline{B} + B) + AC(B + \overline{B}) \\
 &= \overline{A}C + AC \\
 &= C(\overline{A} + A) = C
 \end{aligned}$$

Como outro exemplo, considere a Fig. 4-13(d), onde os quatro quadrados que contêm 1s são:  $ABC\overline{D}$ ,  $A\overline{B}C\overline{D}$ ,  $ABC\overline{D}$  e  $A\overline{B}C\overline{D}$ . Um exame destes termos indica que somente as variáveis  $A$  e  $\overline{D}$  permanecem inalteradas, portanto a expressão simplificada para  $X$  é

$$X = A\overline{D}$$

Isto pode ser provado da mesma maneira que foi feito anteriormente. O leitor deve analisar cada um dos casos na Fig. 4-13 para verificar as expressões indicadas para  $X$ .

Resumindo:

**Agrupar um quarteto de 1s elimina as duas variáveis que aparecem nas formas complementada e não-complementada.**

### Agrupando Oito Termos (Octetos)

Um grupo de oito 1s que são adjacentes entre si é chamado de *octeto*. Muitos exemplos de octetos são mostrados na Fig. 4-14. Quando um octeto é agrupado num mapa de quatro variáveis, três das quatro variáveis são eliminadas, porque apenas uma variável permanece inalterada. Por exemplo, um exame dos oito quadrados agrupados na Fig. 4-14(a) mostra que somente a variável  $B$  está na mesma forma para

todos os oito quadrados; as outras variáveis aparecem nas formas complementada e não-complementada. Portanto, para este mapa,  $X = B$ . O leitor pode verificar os resultados para os outros exemplos na Fig. 4-14.

Resumindo:

**Agrupar um octeto de 1s elimina as três variáveis que aparecem nas formas complementada e não-complementada.**

### Processo Completo de Simplificação

Vimos que o agrupamento de pares, quartetos e octetos num mapa K pode ser usado para obtermos uma expressão simplificada. Podemos resumir a regra para grupos de *qualquer* tamanho:

**Quando uma variável aparece nas formas complementada e não-complementada dentro de um grupo, esta variável é eliminada da expressão. Variáveis que não mudam para todos os quadrados do grupo devem aparecer na expressão final.**

Deve ficar claro que um grupo maior de 1s elimina mais variáveis. Para ser exato, um grupo de dois elimina uma variável, um grupo de quatro elimina duas e um grupo de oito elimina três. Este princípio, agora, será utilizado para obter uma expressão lógica simplificada a partir de um mapa K que contenha qualquer combinação de 1s e 0s.

O procedimento será primeiramente resumido e então aplicado em vários exemplos. Os passos a seguir são reali-

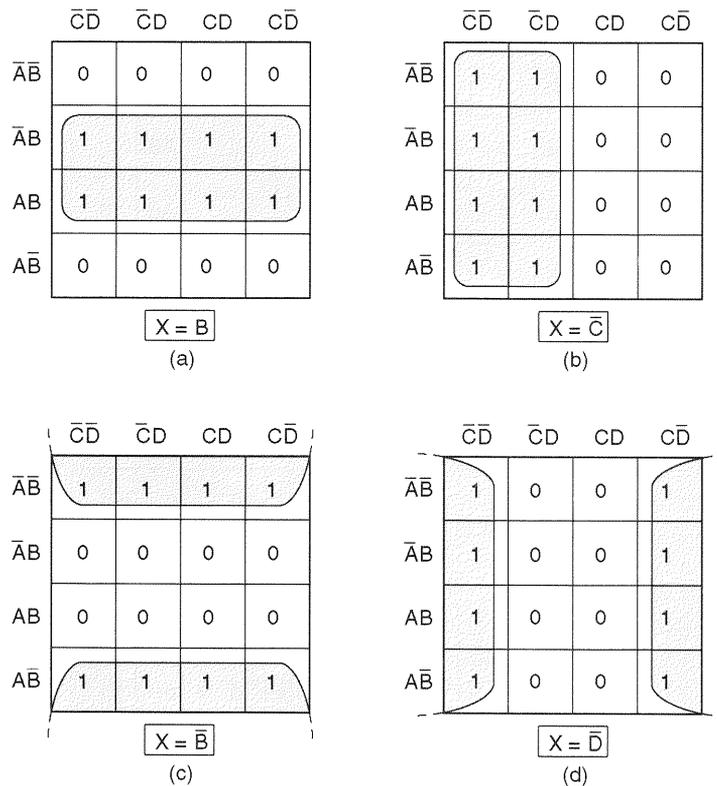


Fig. 4-14 Exemplos de agrupamentos de oito 1s (octetos).

zados para a utilização do método do mapa K para simplificação de uma expressão booleana:

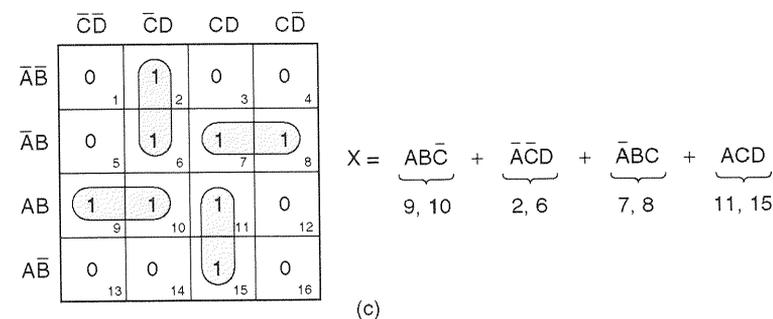
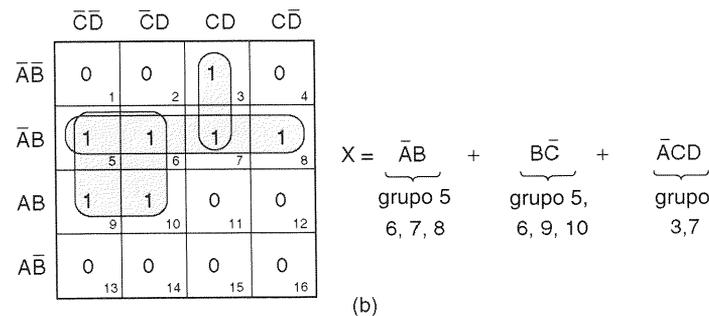
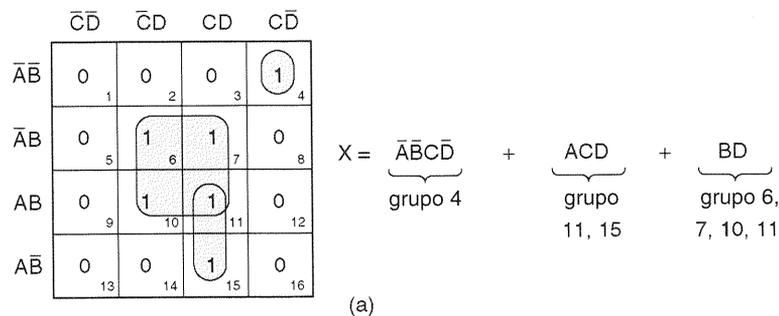
- Passo 1** Construa o mapa K e coloque 1s nos quadrados que correspondem aos 1s na tabela-verdade. Coloque 0s nos outros quadrados.
- Passo 2** Examine o mapa para detectar 1s adjacentes e agrupe aqueles 1s que *não* são adjacentes a quaisquer outros 1s. Estes são denominados 1s *isolados*.
- Passo 3** Em seguida, procure por aqueles 1s que são adjacentes a somente um outro 1. Agrupe *todo* par que contém tal 1.
- Passo 4** Agrupe qualquer octeto, mesmo que ele contenha alguns 1s que já tenham sido combinados.
- Passo 5** Agrupe qualquer quarteto que contém um ou mais 1s que ainda não tenham sido combinados, *certificando-se de usar o número mínimo de agrupamentos*.
- Passo 6** Agrupe quaisquer pares necessários para incluir quaisquer 1s que ainda não tenham sido combinados, *certificando-se de usar o número mínimo de agrupamentos*.
- Passo 7** Forme a soma OR de todos os termos gerados por cada agrupamento.

Estes passos são seguidos e mencionados nos exemplos seguintes. Em cada caso, a expressão lógica resultante está na sua forma de soma-de-produtos mais simples.

**EXEMPLO 4-10**

A Fig. 4-15(a) mostra o mapa K para um problema de quatro variáveis. Vamos supor que o mapa foi obtido a partir da tabela-verdade do problema (passo 1). Os quadrados estão numerados por conveniência para identificação de cada grupo.

- Passo 2** O quadrado 4 é o único quadrado que contém um 1 que não é adjacente a qualquer outro 1. Ele é separado e mencionado como grupo 4.
- Passo 3** O quadrado 15 é adjacente *apenas* ao quadrado 11. Este par é agrupado e mencionado como grupo 11, 15.
- Passo 4** Não existem octetos.
- Passo 5** Os quadrados 6, 7, 10 e 11 formam um quarteto. Este quarteto é agrupado (grupo 6, 7, 10, 11).



**Fig. 4-15** Exemplos 4-10 até 4-12.

- 11). Repare que o quadrado 11 é usado novamente, embora já seja parte do grupo 11, 15. Todos os 1s já estão agrupados.
- Passo 6** Cada grupo gera um termo na expressão para  $X$ . O grupo 4 é simplesmente  $\bar{A}\bar{B}C\bar{D}$ . O grupo 11, 15 é  $ACD$  (a variável  $B$  foi eliminada). O grupo 6, 7, 10, 11 é  $BD$  ( $A$  e  $C$  foram eliminadas).

**EXEMPLO 4-11**

Considere o mapa K na Fig. 4-15(b). Mais uma vez presumimos que o passo 1 já foi realizado.

- Passo 2** Não existem 1s isolados.
- Passo 3** O 1 no quadrado 3 é adjacente apenas ao 1 do quadrado 7. Agrupando-se este par (grupo 3, 7), produz-se o termo  $\bar{A}CD$ .
- Passo 4** Não existem octetos.
- Passo 5** Existem dois quartetos. Os quadrados 5, 6, 7 e 8 formam um quarteto. Reunindo-se este quarteto produz-se o termo  $\bar{A}B$ . O segundo quarteto é formado pelos quadrados 5, 6, 9 e 10. Este quarteto é agrupado porque contém dois quadrados que não tinham sido combinados anteriormente. Este grupo produz  $B\bar{C}$ .
- Passo 6** Todos os 1s já estão agrupados.
- Passo 7** Os termos gerados pelos três grupos são unidos por um OR para obtermos a expressão para  $X$ .

**EXEMPLO 4-12**

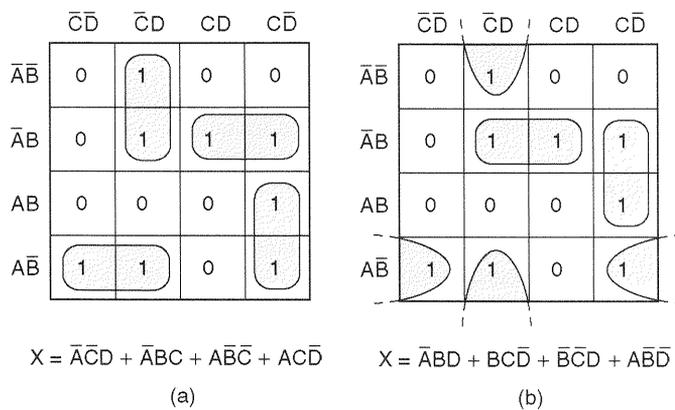
Considere o mapa K na Fig. 4-15(c).

- Passo 2** Não existem 1s isolados.
- Passo 3** O 1 no quadrado 2 é adjacente apenas ao 1 no quadrado 6. Este par é agrupado para produzir  $\bar{A}\bar{C}D$ . Analogamente, o quadrado 9 é adjacente apenas ao quadrado 10. Combinando-se este par produz-se  $AB\bar{C}$ . Do mesmo modo, o grupo 7, 8 e o grupo 11, 15 produzem os termos  $\bar{A}BC$  e  $ACD$ , respectivamente.
- Passo 4** Não existem octetos.
- Passo 5** Existe um quarteto formado pelos quadrados 6, 7, 10 e 11. Este quadrado, no entanto, não é combinado, porque todos os 1s no quarteto já foram incluídos em outros grupos.
- Passo 6** Todos os 1s já foram agrupados.
- Passo 7** A expressão para  $X$  está mostrada na figura.

**EXEMPLO 4-13**

Considere o mapa K na Fig. 4-16(a).

- Passo 2** Não existem 1s isolados.



**Fig. 4-16** O mesmo mapa K com duas soluções igualmente boas.

- Passo 3** Não existe nenhum 1 que seja adjacente apenas a um outro 1.
- Passo 4** Não existem octetos.
- Passo 5** Não existem quartetos.
- Passos 6 e 7** Existem muitos pares possíveis. O processo de agrupar deve usar o mínimo número de grupos para envolver todos os 1s. Para este mapa existem duas possibilidades, que requerem apenas quatro pares envolvidos. A Fig. 4-16(a) mostra uma solução e sua expressão resultante. A Fig. 4-16(b) mostra a outra. Note que ambas as expressões têm a mesma complexidade, e portanto nenhuma é melhor do que a outra.

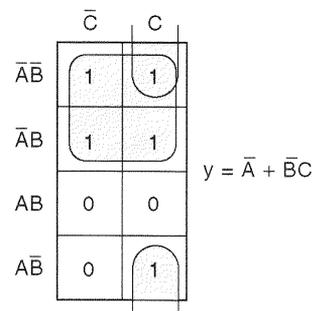
**EXEMPLO 4-14**

Utilize o mapa K para simplificar a expressão  $y = \bar{A}\bar{B}\bar{C} + \bar{B}C + \bar{A}B$ .

**Solução**

Neste problema não é apresentada uma tabela-verdade para o preenchimento do mapa K. Em vez disso, devemos preencher o mapa K tomando cada um dos termos produto na expressão e colocando 1s nos quadrados correspondentes.

O primeiro termo,  $\bar{A}\bar{B}\bar{C}$ , indica que um 1 deve ser colocado no quadrado  $\bar{A}\bar{B}\bar{C}$  do mapa (veja a Fig. 4-17). O segundo termo,  $\bar{B}C$ , indica que um 1 deve ser colocado em cada quadra-



**Fig. 4-17** Exemplo 4-14.

do que contém  $\bar{B}C$  no seu rótulo. Na Fig. 4-17, isto acontece nos quadrados  $A\bar{B}C$  e  $\bar{A}\bar{B}C$ . Do mesmo modo, o termo  $\bar{A}B$  indica que 1 deve ser colocado nos quadrados  $\bar{A}BC$  e  $\bar{A}B\bar{C}$ . Todos os outros quadrados devem ser preenchidos com 0s.

Agora o mapa K pode ser usado para simplificação. O resultado é  $y = \bar{A} + \bar{B}C$ , como apresentado na figura.

### Condições “Don’t Care”

Alguns circuitos lógicos podem ser projetados, de modo que existam certas condições de entrada para as quais não existam níveis de saída especificados, usualmente porque estas condições de entrada nunca ocorrerão. Em outras palavras, existem certas combinações de níveis de entrada em que “não importa” (do inglês “*don’t care*”) se a saída está em ALTO ou BAIXO. Isto está ilustrado na tabela-verdade da Fig. 4-18(a).

Aqui a saída  $z$  não está especificada nem como 0 nem como 1 para as seguintes condições:  $A, B, C = 1, 0, 0$  e  $A, B, C = 0, 1, 1$ . Em vez disso, um  $x$  é mostrado para estas condições. O  $x$  representa a **condição don’t care**. Uma condição *don’t care* pode surgir por várias razões; a mais comum é a existência de algumas situações nas quais certas combinações de entrada não podem nunca ocorrer, e portanto não existe saída especificada para estas condições.

Um projetista de circuitos está livre para fazer a saída ser 0 ou 1 para qualquer condição *don’t care*, de modo a produzir a expressão de saída mais simples. Por exemplo, o mapa K para esta tabela-verdade é mostrado na Fig. 4-18(b) com um  $x$  colocado nos quadrados  $A\bar{B}\bar{C}$  e  $\bar{A}B\bar{C}$ . Neste caso, o projetista deve ser inteligente para substituir o  $x$  no quadrado  $A\bar{B}\bar{C}$  por 1 e o  $x$  no quadrado  $\bar{A}B\bar{C}$  por 0, já que isto produz um quarteto que pode ser agrupado para resultar em  $z = A$ , conforme mostra a Fig. 4-18(c).

Toda vez que condições *don’t care* ocorrem, devemos decidir qual  $x$  deve mudar para 0 e qual deve mudar para 1, de modo a produzir o melhor grupo no mapa K (isto é, a expressão mais simples). Esta decisão nem sempre é fácil. Muitos problemas no fim do capítulo proporcionarão prática em lidar com casos de *don’t care*.

### Resumo

O processo do mapa K tem muitas vantagens sobre o método algébrico. O mapa K é um processo mais ordenado, com passos bem-definidos quando comparado com o pro-

cesso de tentativa e erro, algumas vezes usado na simplificação algébrica. Usualmente, o mapa K necessita de menos etapas, sobretudo para expressões que contêm muitos termos, e ele sempre produz uma expressão mínima.

Contudo, alguns professores preferem o método algébrico porque ele requer um amplo conhecimento da álgebra booleana e não é apenas um procedimento mecânico. Cada método tem suas vantagens, e, embora a maioria dos projetistas de lógica seja adepta dos dois, ser competente em um método é o suficiente para produzir resultados aceitáveis.

Existem outras técnicas mais complexas que os projetistas usam para minimizar circuitos lógicos. Estas técnicas são especialmente apropriadas para circuitos com um grande número de entradas, nos quais tanto o método algébrico quanto o do mapa K são impraticáveis. A maioria dessas técnicas pode ser transformada em um programa de computador que realiza a minimização sobre a tabela-verdade ou sobre a expressão completa.

### Questões de Revisão

1. Use o mapa K para simplificar a expressão do Exemplo 4-7.
2. Use o mapa K para simplificar a expressão do Exemplo 4-8. Isto deve enfatizar a vantagem de utilização do mapa K para expressões que contêm muitos termos.
3. Simplifique a expressão do Exemplo 4-9 usando um mapa K.
4. O que é uma condição *don’t care*?

## 4-6 CIRCUITOS EXCLUSIVE-OR E EXCLUSIVE-NOR

Dois circuitos lógicos especiais que freqüentemente aparecem em sistemas digitais são os circuitos *exclusive-OR* e o *exclusive-NOR*.

### Exclusive-OR

Considere o circuito lógico da Fig. 4-19(a). A expressão de saída deste circuito é

$$x = \bar{A}B + A\bar{B}$$

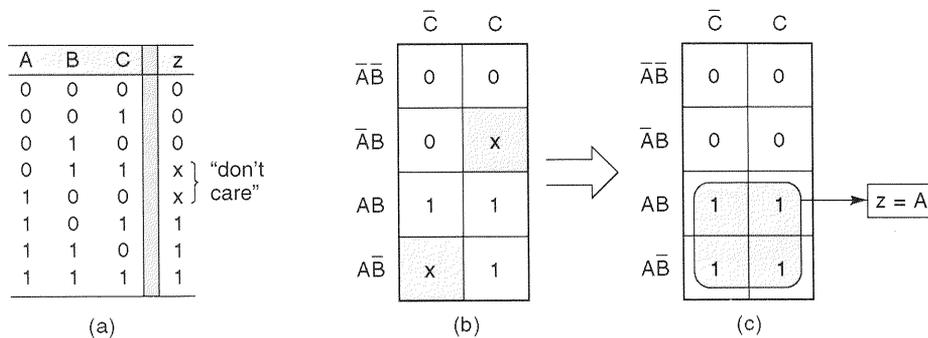
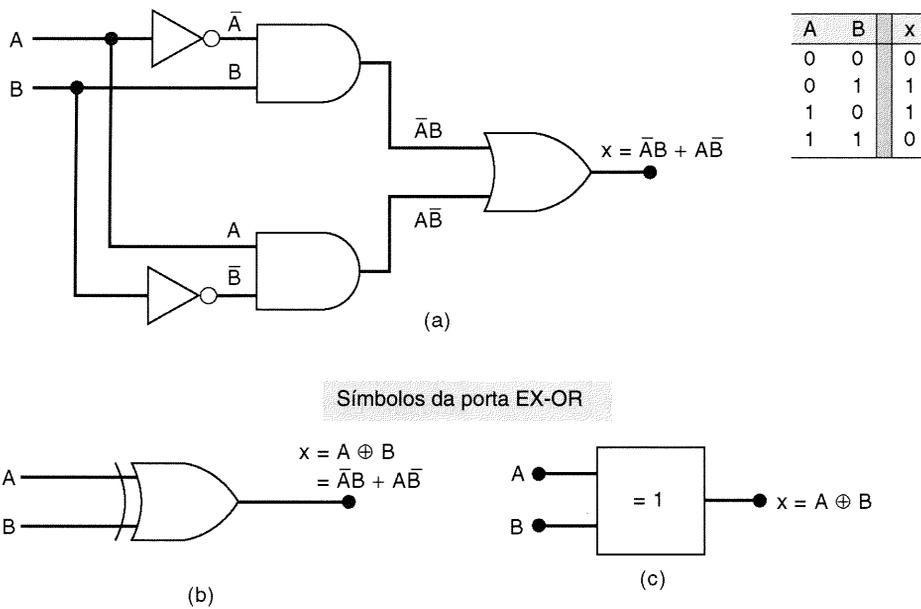


Fig. 4-18 Condições *don’t care* podem ser substituídas por 0 ou 1 para produzir o grupo que resulta na expressão mais simples.



**Fig. 4-19** (a) Tabela-verdade e circuito *exclusive-OR*; (b) símbolo tradicional da porta EX-OR; (c) símbolo IEEE/ANSI para a porta EX-OR.

A tabela-verdade apresentada mostra que  $x = 1$  para dois casos:  $A = 0, B = 1$  (o termo  $\bar{A}B$ ) e  $A = 1, B = 0$  (o termo  $A\bar{B}$ ). Em outras palavras:

**Este circuito produz uma saída em ALTO sempre que as duas entradas estão em níveis opostos.**

Este é o circuito **exclusive-OR**, que daqui para a frente será abreviado como **EX-OR**.

Essa combinação especial de portas lógicas ocorre frequentemente e é muito útil em certas aplicações. Na verdade, o circuito EX-OR tem um símbolo próprio, que é mostrado na Fig. 4-19(b). Supõe-se que este símbolo contém todas as portas lógicas de um circuito EX-OR e portanto tem a mesma expressão lógica e a mesma tabela-verdade. Esse circuito EX-OR é normalmente mencionado como uma *porta EX-OR*, que é considerada um outro tipo de porta lógica. O símbolo IEEE/ANSI para uma porta EX-OR é mostrado na Fig. 4-19(c). A notação de dependência (= 1) dentro do bloco indica que a saída está ativa-ALTO *somente* quando uma única entrada está em ALTO.

Uma porta EX-OR tem apenas *duas entradas*. Não existem portas EX-OR de três ou quatro entradas. As duas entradas são combinadas de modo que  $x = \bar{A}B + A\bar{B}$ . Um modo abreviado que algumas vezes é usado para indicar uma expressão de saída EX-OR é

$$x = A \oplus B$$

onde o símbolo  $\oplus$  representa a operação da porta EX-OR.

As características de uma porta EX-OR podem ser resumidas como se segue:

1. Tem apenas duas entradas e sua saída é

$$x = \bar{A}B + A\bar{B} = A \oplus B$$

2. Sua saída está em ALTO somente quando as duas entradas estão em níveis *diferentes*.

Diversos CIs que contêm portas EX-OR estão disponíveis. Os chips relacionados a seguir são EX-OR *quádruplos*, que contêm quatro portas EX-OR.

- **74LS86** EX-OR quádruplo (família TTL)
- **74C86** EX-OR quádruplo (família CMOS)
- **74HC86** EX-OR quádruplo (família HCMOS — *High-speed* CMOS — CMOS de alta velocidade)

### Exclusive-NOR

O circuito **exclusive-NOR** (abreviado como **EX-NOR**) opera ao contrário do circuito EX-OR. A Fig. 4-20(a) mostra um circuito EX-NOR e sua respectiva tabela-verdade. A expressão de saída é

$$x = AB + \bar{A}\bar{B}$$

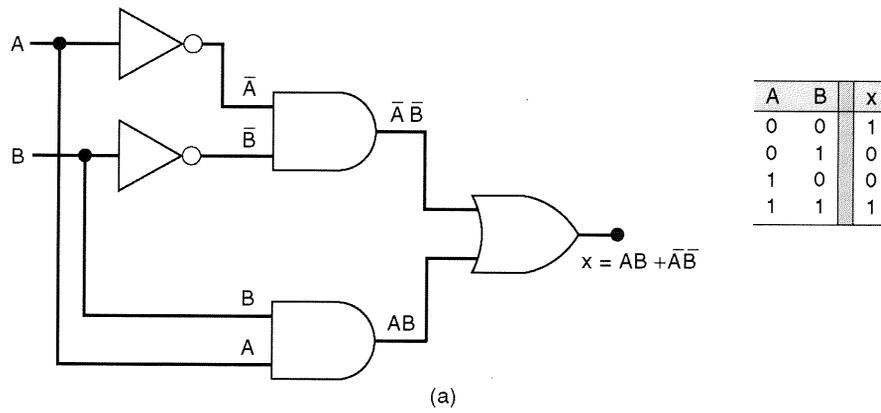
que indica juntamente com a tabela-verdade que  $x$  é 1 para dois casos:  $A = B = 1$  (o termo  $AB$ ) e  $A = B = 0$  (o termo  $\bar{A}\bar{B}$ ). Em outras palavras:

**Este circuito produz uma saída em ALTO sempre que as duas entradas estão no mesmo nível.**

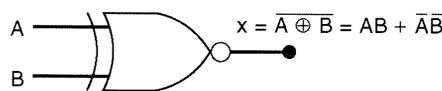
Deve estar claro que a saída de um circuito EX-NOR é exatamente o inverso da saída de um circuito EX-OR. O símbolo tradicional de uma porta EX-NOR é obtido simplesmente adicionando-se um pequeno círculo à saída do símbolo do EX-OR [Fig. 4-20(b)]. O símbolo IEEE/ANSI adiciona um pequeno triângulo à saída do símbolo EX-OR. Ambos os símbolos indicam uma saída que vai para o estado ativo em BAIXO quando *somente* uma entrada está em ALTO.

A porta EX-NOR também tem apenas *duas* entradas, e as combina de modo que sua saída é

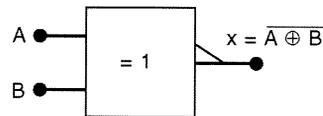
$$x = AB + \bar{A}\bar{B}$$



Símbolos da porta EX-NOR



(b)



(c)

Fig. 4-20 (a) Circuito *exclusive*-NOR; (b) símbolo tradicional da porta EX-NOR; (c) símbolo IEEE/ANSI.

Um modo abreviado de indicar uma expressão de saída de um EX-NOR é

$$x = \overline{A \oplus B}$$

que é simplesmente o inverso da operação EX-OR. A porta EX-NOR é resumida como se segue:

1. Tem apenas duas entradas e sua saída é

$$x = AB + \overline{A}\overline{B} = \overline{A \oplus B}$$

2. Sua saída está em ALTO somente quando as duas entradas estão no *mesmo* nível.

Diversos CIs que contêm portas EX-NOR estão disponíveis. Os chips relacionados a seguir são EX-NOR quádruplos, que contêm quatro portas EX-NOR.

- **74LS266** EX-NOR quádruplo (família TTL)
- **74C266** EX-NOR quádruplo (família CMOS)
- **74HC266** EX-NOR quádruplo (família HCMOS)

Cada um desses chips EX-NOR, entretanto, tem um circuito especial de saída que limita seu uso a certos tipos de aplicações. Muito frequentemente, um projetista obtém a função EX-NOR simplesmente conectando a saída de um EX-OR a um INVERSOR.

**EXEMPLO 4-15**

Determine a forma de onda da saída para as formas de onda de entrada na Fig. 4-21.

**Solução**

A forma de onda da saída é obtida sabendo que a saída EX-OR vai para ALTO somente quando suas entradas têm níveis diferentes. A forma de onda resultante revela vários pontos interessantes:

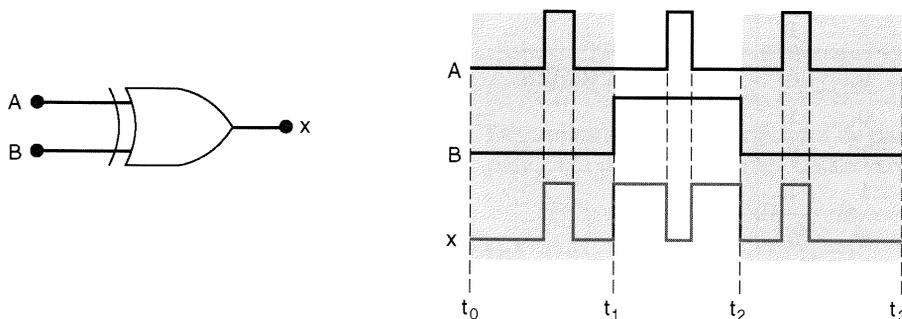


Fig. 4-21 Exemplo 4-15.

1. A forma de onda de  $x$  segue a forma de onda da entrada  $A$  durante os intervalos de tempo em que  $B = 0$ . Isto ocorre durante os intervalos de tempo  $t_0$  a  $t_1$  e  $t_2$  a  $t_3$ .
2. A forma de onda de  $x$  é o inverso da forma de onda da entrada  $A$  durante os intervalos de tempo em que  $B = 1$ . Isto ocorre durante o intervalo  $t_1$  a  $t_2$ .
3. Essas observações mostram que uma porta EX-OR pode ser usada como um *INVERSOR controlado*, isto é, uma de suas entradas pode ser utilizada para controlar se o sinal presente na outra entrada deve ou não ser invertido. Esta propriedade é muito útil em certas aplicações.

**EXEMPLO 4-16**

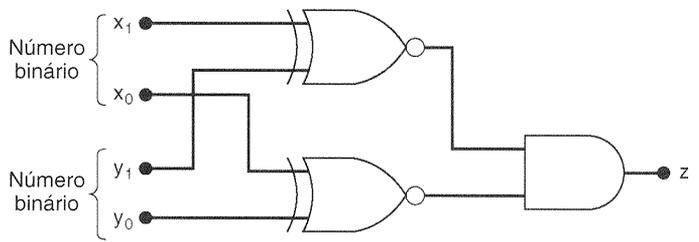
$x_1x_0$  representa um número binário de dois bits que pode ter qualquer valor (00, 01, 10 ou 11); por exemplo, quando  $x_1 = 1$  e  $x_0 = 0$ , o número binário é 10, e assim por diante. Analogamente,  $y_1y_0$  representa um outro número binário de dois bits. Projete um circuito lógico, usando as entradas  $x_1, x_0, y_1$  e  $y_0$ , cuja saída vai para ALTO somente quando os dois números binários  $x_1x_0$  e  $y_1y_0$  são iguais.

**Solução**

O primeiro passo é construir a tabela-verdade para as 16 condições de entrada (Tabela 4-4). A saída  $z$  deve estar em ALTO sempre que os valores de  $x_1x_0$  e  $y_1y_0$  coincidirem, isto é, sempre que  $x_1 = y_1$  e  $x_0 = y_0$ . A tabela mostra que existem quatro casos. Poderíamos continuar com o procedimento formal e obtermos a expressão de soma-de-produtos de  $z$ , tentamos simplificá-la e então implementarmos o resultado. Entretanto, a natureza desse problema torna-o ideal para implementação com portas EX-NOR, e um pouco de reflexão produz uma solução simples com um mínimo esforço.

TABELA 4-4

$x_1$	$x_0$	$y_1$	$y_0$	$z$ (Saída)
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1



**Fig. 4-22** Circuito para detectar a igualdade de dois números binários de dois bits.

No diagrama lógico da Fig. 4-22,  $x_1$  e  $y_1$  estão ligados a uma das portas EX-NOR, e  $x_0$  e  $y_0$  estão ligados à outra porta EX-NOR. A saída de cada EX-NOR está em ALTO somente quando as suas entradas são iguais. Assim, para  $x_0 = y_0$  e  $x_1 = y_1$ , ambas as saídas das portas EX-NOR estão em ALTO. Esta é a condição procurada porque significa que os dois números de dois bits são iguais. A saída da porta AND está em ALTO somente neste caso, realizando assim a saída desejada.

**EXEMPLO 4-17**

Quando se simplifica a expressão para a saída de um circuito lógico combinacional, pode-se encontrar operações EX-OR ou EX-NOR durante a fatoração. Isto freqüentemente conduz ao uso de portas EX-OR ou EX-NOR na implementação do circuito final. Para ilustrar, simplifique o circuito da Fig. 4-23(a).

**Solução**

A expressão não-simplificada para o circuito é

$$z = ABCD + A\bar{B}\bar{C}D + \bar{A}\bar{D}$$

Podemos fatorar  $AD$  dos dois primeiros termos:

$$z = AD(BC + \bar{B}\bar{C}) + \bar{A}\bar{D}$$

À primeira vista, pode-se pensar que a expressão entre parênteses pode ser substituída por 1. Isto somente seria possível se tivéssemos  $BC + \bar{B}\bar{C}$ . Você deveria reconhecer a expressão entre parênteses como uma combinação EX-NOR de  $B$  e  $C$ . Este fato pode ser usado para implementar novamente o circuito, conforme mostrado na Fig. 4-23(b). Este circuito é bem mais simples do que o original, pois ele usa portas com menos entradas, e dois INVERSORES foram eliminados.

**Questões de Revisão**

1. Utilize a álgebra booleana para provar que a expressão de saída para EX-NOR é exatamente o inverso da expressão de saída para EX-OR.
2. Qual é a saída de uma porta EX-NOR quando um sinal lógico e seu inverso estão conectados às suas entradas?
3. Um projetista necessita de um INVERSOR e tudo de que ele dispõe é uma porta EX-OR de um chip 74HC86. Ele precisa de outro chip?

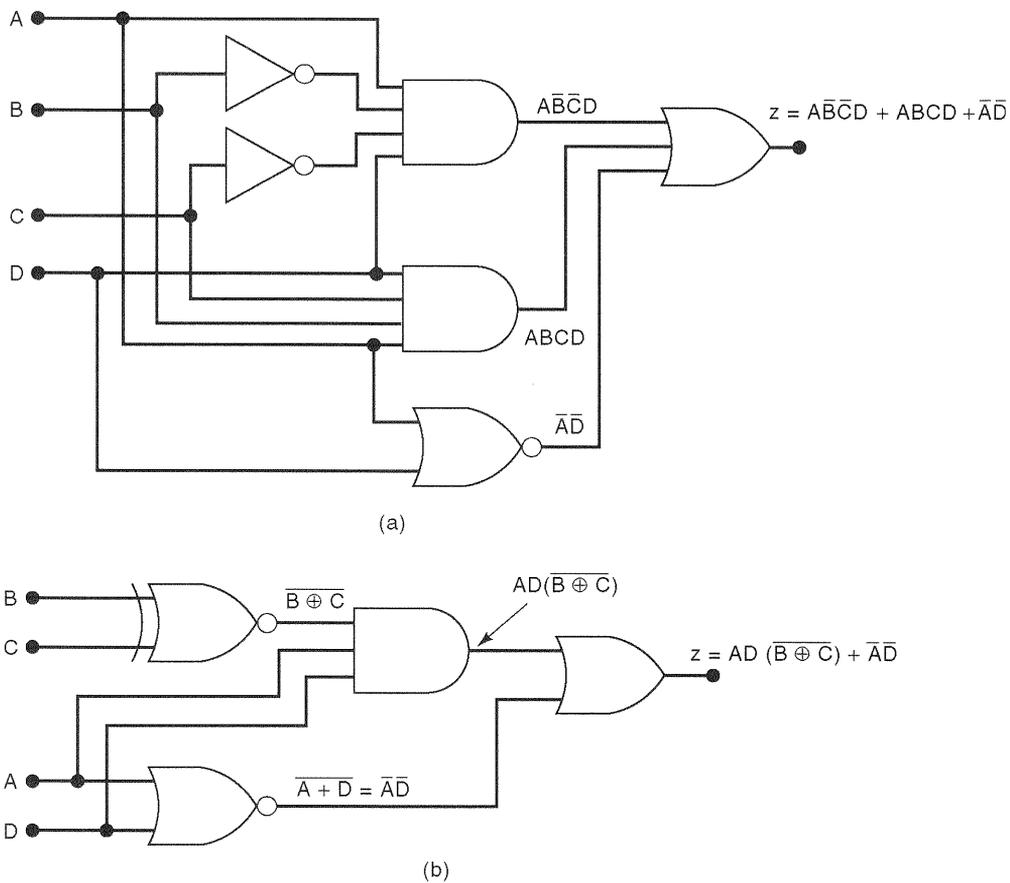


Fig. 4-23 O Exemplo 4-17 mostra como uma porta EX-NOR pode ser usada para simplificar a implementação de circuitos.

## 4-7 CIRCUITOS GERADOR E VERIFICADOR DE PARIDADE

No Cap. 2 vimos que um transmissor pode anexar um bit de paridade a um conjunto de bits antes de transmiti-lo para o receptor. Também foi visto como o receptor detecta qualquer erro simples em apenas um bit que possa ter ocorrido durante a transmissão. A Fig. 4-24 mostra um exemplo de circuito lógico que é usado para **geração de paridade** e **verificação de paridade**. Este exemplo em particular usa um grupo de quatro bits como sendo os dados a serem transmitidos e utiliza um bit de paridade par. Ele pode ser facilmente adaptado para utilização de paridade ímpar e para qualquer número de bits.

Na Fig. 4-24(a), os dados a serem transmitidos são aplicados ao circuito gerador de paridade que produz o bit de paridade par,  $P$ , como sua saída. Este bit de paridade é transmitido para o receptor junto com os bits do dado original, formando um total de cinco bits. Na Fig. 4-24(b), estes cinco bits (dado + paridade) chegam no circuito verificador de paridade do receptor, que produz uma saída de erro,  $E$ , que indica se ocorreu um erro simples em um bit.

Não deve causar surpresa a utilização de portas EX-OR nesses circuitos, quando consideramos que uma porta EX-OR opera de modo a produzir uma saída em 1 se um número ímpar de suas entradas está em 1, e uma saída em 0 se um número par de suas entradas está em 1.

### EXEMPLO 4-18

Determine a saída do gerador de paridade para cada um dos seguintes dados de entrada,  $D_3D_2D_1D_0$ : (a) 0111; (b) 1001; (c) 0000; (d) 0100. Vide Fig. 4-24(a).

### Solução

Para cada caso, aplique os dados de entrada no gerador de paridade e acompanhe a saída de cada porta até a saída  $P$ . Os resultados são: (a) 1; (b) 0; (c) 0; e (d) 1. Note que  $P$  é 1 somente quando o dado original contém um número ímpar de 1s. Assim, o número total de 1s enviados ao receptor (dado + paridade) é par.

### EXEMPLO 4-19

Determine a saída do verificador de paridade [vide Fig. 4-24(b)] para cada um dos seguintes dados enviados pelo transmissor:

	$P$	$D_3$	$D_2$	$D_1$	$D_0$
(a)	0	1	0	1	0
(b)	1	1	1	1	0
(c)	1	1	1	1	1
(d)	1	0	0	0	0

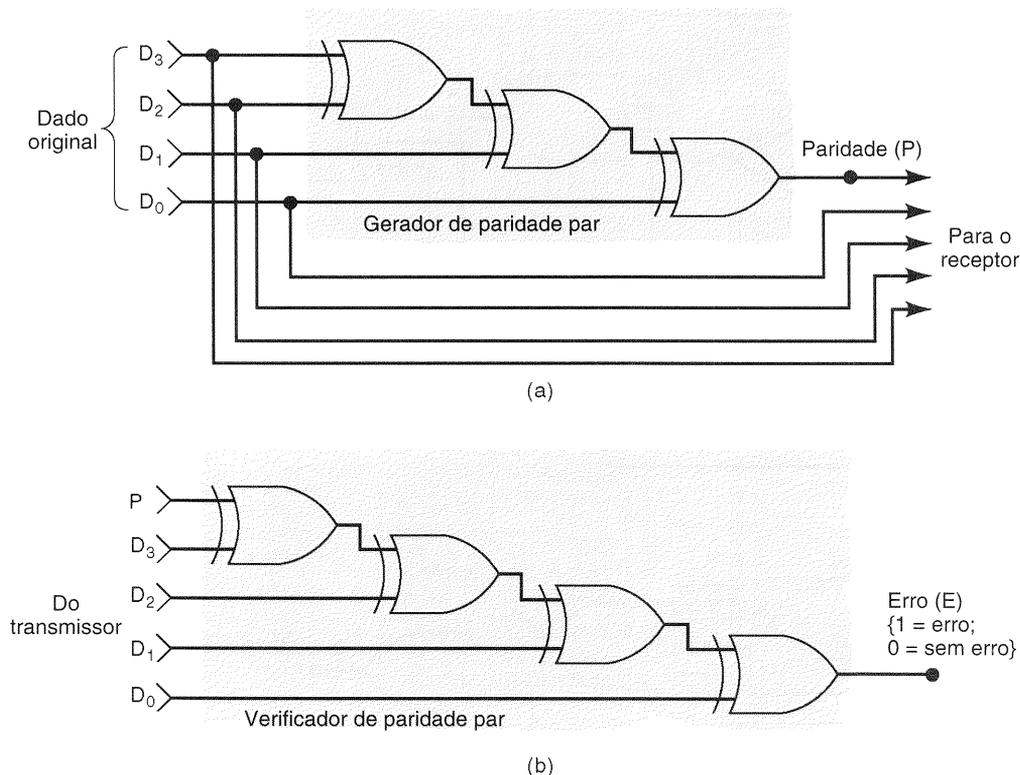


Fig. 4-24 Portas EX-OR utilizadas para implementar o gerador de paridade e o verificador de paridade para um sistema de paridade par.

### Solução

Para cada caso, aplique esses níveis às entradas do verificador de paridade e siga os sinais até a saída  $E$ . Os resultados são: **(a)** 0; **(b)** 0; **(c)** 1; **(d)** 1. Note que 1 é produzido em  $E$  somente quando um número ímpar de 1s aparece nas entradas do verificador de paridade. Isto indica que ocorreu um erro, pois paridade par está sendo usada.

#### EXEMPLO 4-20

O circuito verificador de paridade tem como “saber” qual bit está errado?

### Solução

Não. O verificador de paridade não sabe que estado cada bit de entrada deveria ter; ele sabe apenas que um número par de 1s deve estar presente. Independentemente de qual bit esteja errado, um erro simples em um bit muda o número total de 1s, de par para ímpar (ou removendo ou incluindo um bit 1), e acarreta que  $E$  vá para ALTO.

## 4-8 CIRCUITOS PARA HABILITAR/DESABILITAR

Cada uma das portas lógicas básicas pode ser usada para controlar a passagem de um sinal lógico da entrada para a

saída. Isto é detalhado na Fig. 4-25, onde um sinal lógico,  $A$ , é aplicado a uma entrada de cada uma das portas lógicas básicas. A outra entrada de cada porta é a entrada de controle,  $B$ . O nível lógico desta entrada de controle determina se o sinal de entrada está **habilitado** a alcançar a saída ou impedido (**desabilitado**) de alcançá-la. Esta ação de controle é o motivo pelo qual esses circuitos são chamados de *portas*.

Examine a Fig. 4-25 e observe que quando portas não-inversoras (AND, OR) estão habilitadas, a saída segue o sinal  $A$ . Ao contrário, quando portas inversoras (NAND, NOR) são habilitadas, a saída é exatamente o complemento do sinal  $A$ .

Repare também que portas AND e NOR produzem uma saída constante em BAIXO quando estão desabilitadas. Ao contrário, portas NAND e OR produzem uma saída constante em ALTO na condição desabilitada.

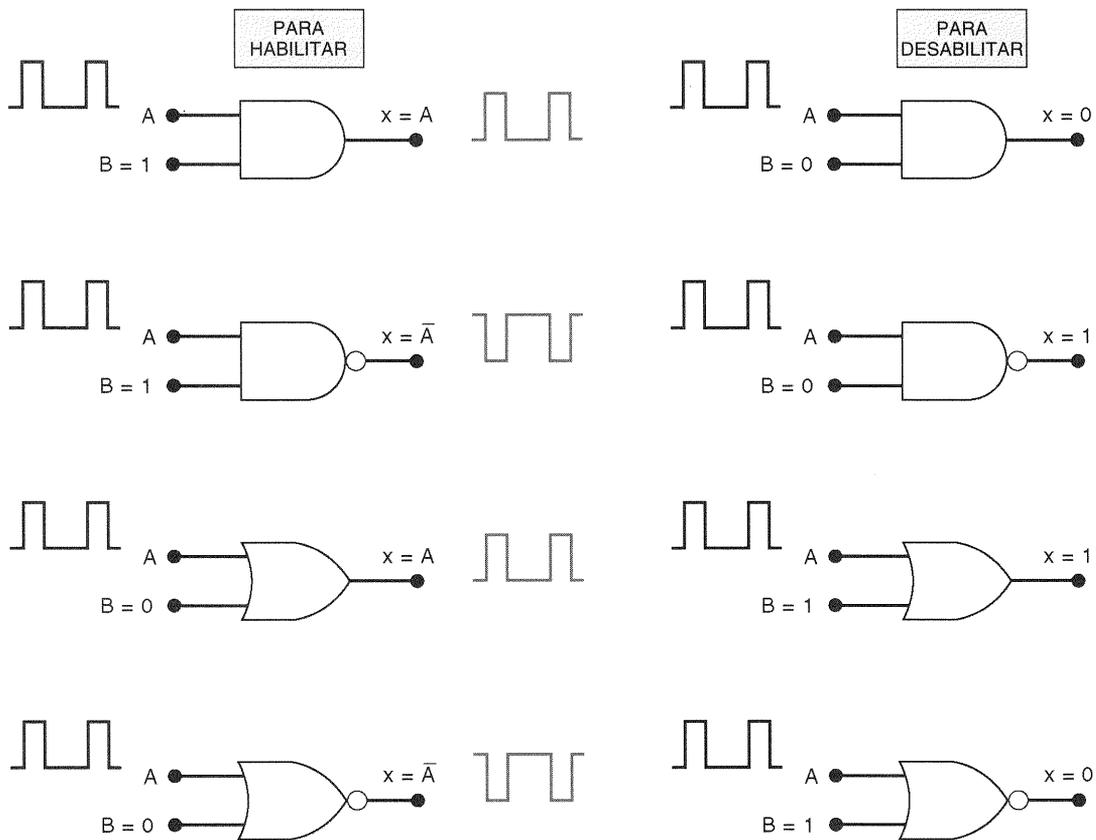
Existem muitas situações no projeto de circuitos digitais em que a passagem de um sinal lógico deve ser habilitada ou não, dependendo de condições presentes em uma ou mais entradas de controle. Muitas dessas situações são mostradas nos exemplos a seguir.

#### EXEMPLO 4-21

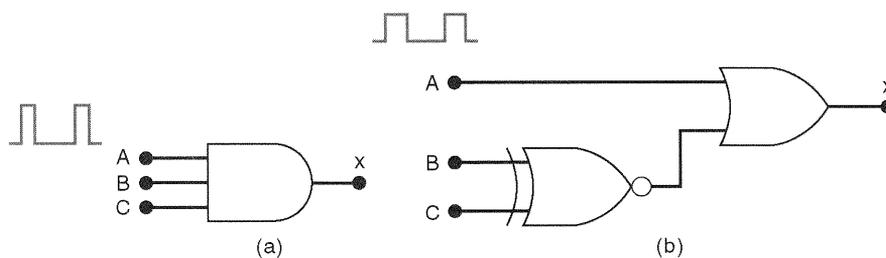
Projete um circuito lógico que permita um sinal passar para a saída somente quando as entradas de controle  $B$  e  $C$  estiverem ambas em ALTO, senão a saída deve ficar em nível BAIXO.

### Solução

Uma porta AND pode ser usada porque o sinal deve ser passado sem inversão e ela produz um nível BAIXO na



**Fig. 4-25** As quatro portas básicas podem habilitar ou desabilitar a passagem de um sinal de entrada, A, dependendo do nível lógico da entrada de controle B.



**Fig. 4-26** Exemplos 4-21 e 4-22.

saída quando está desabilitada. Como a condição de habilitação deve ocorrer quando  $B = C = 1$ , uma porta AND de três entradas é usada, conforme ilustra a Fig. 4-26(a).

das de controle está em ALTO, caso contrário a saída fica em ALTO.

**Solução**

O resultado é apresentado na Fig. 4-26(b). Uma porta OR é usada porque desejamos que a saída esteja em ALTO quando a porta estiver desabilitada e não desejamos inverter o sinal. As entradas de controle B e C são combinadas numa porta EX-NOR. Quando B e C são diferentes, a porta EX-NOR envia um nível BAIXO para habilitar a porta OR. Quando B e C são iguais, o EX-NOR envia um nível ALTO para desabilitar a porta OR.

**EXEMPLO 4-22**

Projete um circuito lógico que permite um sinal passar para a saída somente quando uma, mas não ambas, das entra-

**EXEMPLO 4-23**

Projete um circuito lógico com um sinal de entrada  $A$ , uma entrada de controle  $B$  e saídas  $X$  e  $Y$  que opera do seguinte modo:

1. Quando  $B = 1$ , a saída  $X$  segue a entrada  $A$ , e a saída  $Y$  é 0.
2. Quando  $B = 0$ , a saída  $X$  é 0, e a saída  $Y$  segue a entrada  $A$ .

**Solução**

As duas saídas são 0 quando desabilitadas e seguem o sinal de entrada quando habilitadas. Assim, uma porta AND deve ser usada para cada saída. Como  $X$  deve ser habilitado quando  $B = 1$ , sua porta AND deve ser controlada por  $B$ , como mostra a Fig. 4-27. Tendo em vista que  $Y$  deve ser habilitado quando  $B = 0$ , sua porta AND deve ser controlada por  $\bar{B}$ .

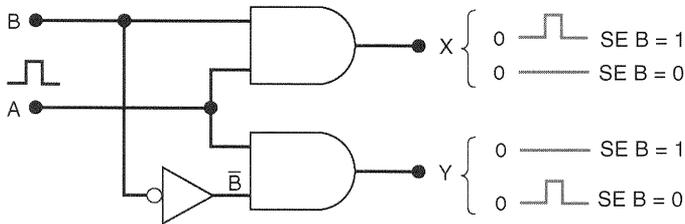


Fig. 4-27 Exemplo 4-23.

Este circuito é denominado *circuito direcionador de pulsos* porque direciona o pulso de entrada para uma ou outra saída, dependendo de  $B$ .

**Questões de Revisão**

1. Projete um circuito lógico com três entradas  $A$ ,  $B$  e  $C$  e uma saída que vai para BAIXO somente quando  $A$  está ALTO enquanto  $B$  e  $C$  são diferentes.
2. Quais portas lógicas produzem uma saída em 1 quando estão desabilitadas?
3. Quais portas lógicas passam o inverso do sinal de entrada quando estão habilitadas?

**4-9 CARACTERÍSTICAS BÁSICAS DE CIs DIGITAIS**

CIs digitais são uma coleção de resistores, diodos e transistores fabricados em uma única peça de material semicondutor (geralmente silício), chamado de *substrato*, que é comumente conhecido pela denominação de *chip*. Este é encapsulado em uma embalagem protetora de plástico ou de cerâmica, a partir da qual saem pinos para tornar possível a ligação do CI com outros dispositivos. Um dos tipos de encapsulamento mais comum é o *dual-in-line package (DIP)*, mostrado na Fig. 4-28(a). Este encapsulamento tem esse nome porque contém duas linhas de pinos em paralelo. Os pinos são numerados no sentido anti-horário, a partir da marca de identificação, quando visto de cima do encapsulamento [veja Fig. 4-28(b)]. O DIP mostrado é de

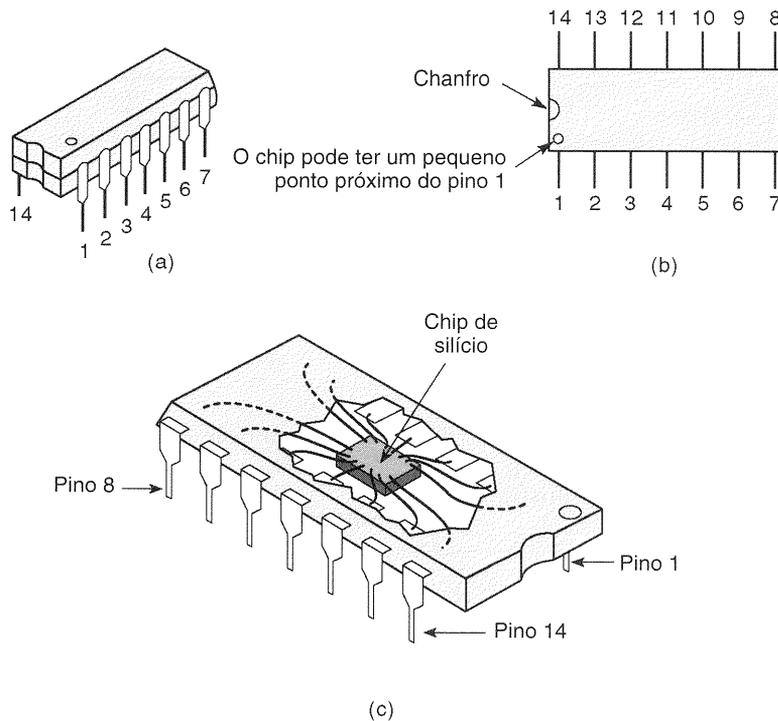


Fig. 4-28 (a) Encapsulamento *dual-in-line* (DIP); (b) vista superior; (c) o chip de silício é muito menor que o encapsulamento.

14 pinos e mede, aproximadamente, 19 mm por 6 mm. DIPs de 16, 20, 24, 28, 40 e 64 pinos também são usados.

A Fig. 4-28(c) mostra que o chip é, na verdade, muito menor que seu DIP. Ele pode ser tão pequeno quanto um quadrado de 1,2 mm. O chip de silício é conectado aos pinos do DIP através de fios bastante finos (0,025 mm de diâmetro).

O DIP é provavelmente o encapsulamento para CIs mais comum de ser encontrado em equipamentos digitais, embora outros tipos estejam se tornando cada vez mais populares. Veremos alguns desses outros tipos de encapsulamento no Cap. 8.

CIs digitais são muitas vezes classificados de acordo com a complexidade de seus circuitos, que é medida pelo número de portas lógicas equivalentes no seu substrato. Existem atualmente seis níveis de complexidade, definidos na Tabela 4-5.

Todos os CIs especificados no Cap. 3, e também neste capítulo, são chips **SSI**, que contêm um pequeno número de portas. Nos sistemas digitais modernos, dispositivos com grau médio de integração (**MSI**) e com alto grau de integração (**LSI**, **VLSI**, **ULSI** e **GSI**) realizam a maior parte das funções que antes eram implementadas por várias placas de circuito impresso cheias de chips SSI. Entretanto, eles ainda são usados como interface, ou também como lógica adicional de chips mais complexos. Geralmente, pequenas combinações de portas discretas são usadas para conectar CIs maiores entre si ou a dispositivos externos. Portanto, é necessário saber como analisar, projetar, testar e consertar circuitos combinacionais simples.

TABELA 4-5

Complexidade	Número de Portas
Small-scale integration ( <b>SSI</b> )	Menor do que 12
Medium-scale integration ( <b>MSI</b> )	12 a 99
Large-scale integration ( <b>LSI</b> )	100 a 9.999
Very large-scale integration ( <b>VLSI</b> )	10.000 a 99.999
Ultra large-scale integration ( <b>ULSI</b> )	100.000 a 999.999
Giga-scale integration ( <b>GSI</b> )	1.000.000 ou mais

## CIs Digitais Bipolares e Unipolares

CIs digitais também podem ser classificados de acordo com o tipo de componente eletrônico usado nos seus circuitos. CIs *bipolares* são aqueles que são feitos utilizando o transistor de junção bipolar (NPN e PNP) como seu elemento principal. CIs *unipolares* são aqueles que usam transistores por efeito-de-campo (MOSFETs canal P e canal N) como seu elemento principal.

A família **TTL** (**Transistor-Transistor Logic**) é a principal família de CIs digitais bipolares nos últimos 25 anos. A Fig. 4-29(a) mostra o circuito de um INVERSOR TTL da série 74 padrão. Esta série foi a primeira dos CIs TTL. Ela não é mais utilizada em novos projetos, mas ainda é o padrão com o qual todas as outras famílias lógicas são comparadas. Observe que o circuito do INVERSOR TTL contém vários transistores bipolares, logo este tipo de transistor é o elemento principal do circuito. A família TTL é a líder nas categorias SSI e MSI há muito tempo; entretanto, essa liderança vem sendo ameaçada pela família **CMOS** (**Complementary Metal-Oxide-Semiconductor**). Esta pertence à categoria de CIs digitais unipolares porque usa MOSFETs canal P e canal N como elemento principal do circuito. A Fig. 4-29(b) mostra o circuito de um INVERSOR CMOS padrão.

Os CIs TTL e CMOS dominam o segmento de dispositivos SSI e MSI, e portanto nos concentraremos nestas duas famílias ao longo do texto. O Cap. 8 fornecerá um estudo detalhado dos circuitos e das características de CIs TTL e CMOS. Por enquanto, precisaremos abordar apenas algumas características básicas para que possamos falar sobre pesquisa de falhas em circuitos combinacionais simples.

## Família TTL

A família TTL é, na verdade, constituída de várias subfamílias ou séries. A Tabela 4-6 relaciona o nome de cada uma das séries com o prefixo usado para identificar o CI como per-

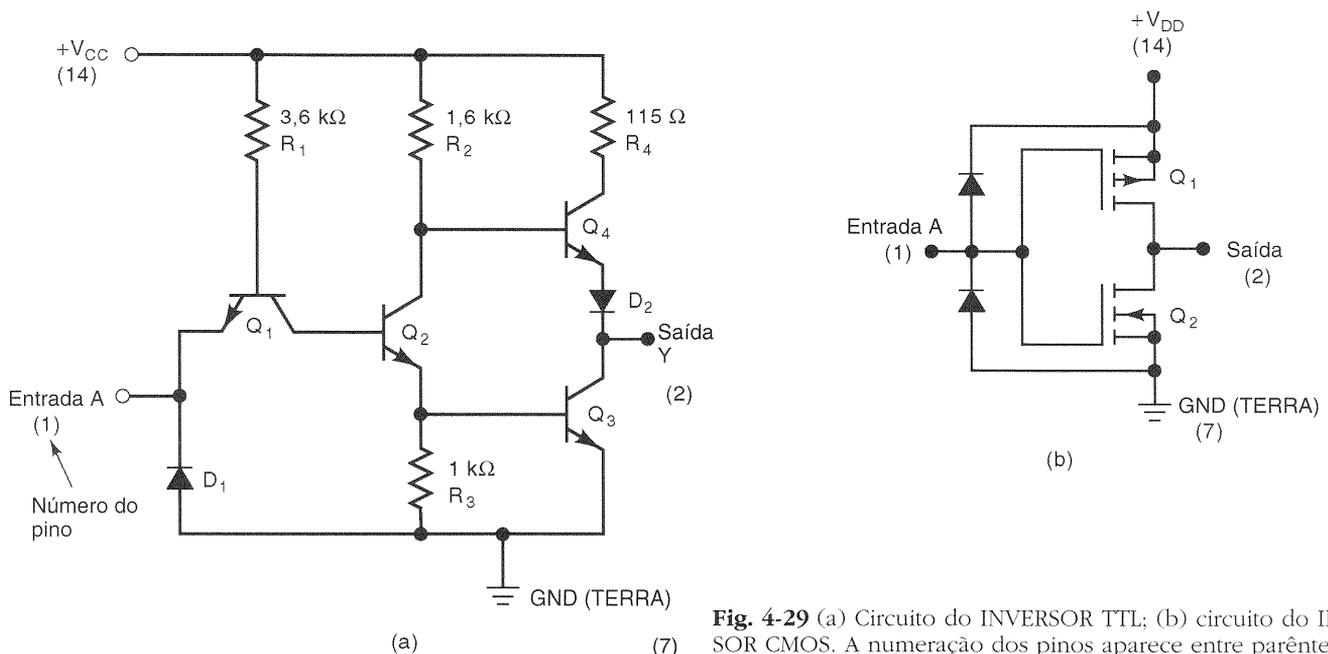


Fig. 4-29 (a) Circuito do INVERSOR TTL; (b) circuito do INVERSOR CMOS. A numeração dos pinos aparece entre parênteses.

tencente a esta série. Por exemplo, CIs que fazem parte da série TTL padrão têm um número de identificação que inicia com 74. O 7402, o 7438 e o 74123 são todos pertencentes a essa série. Do mesmo modo, CIs que pertencem à série TTL Schottky de baixa potência (*low-power Schottky*) têm o seu número de identificação começando por 74LS. O 74LS02, o 74LS38 e o 74LS123 são exemplos de dispositivos da série 74LS.

TABELA 4-6 Várias séries dentro de uma família TTL

Subfamílias TTL	Prefixo	Exemplo de CI
TTL padrão	74	7404 INVERSOR sêxtuplo
TTL Schottky	74S	74S04 INVERSOR sêxtuplo
TTL Schottky de baixa potência	74LS	74LS04 INVERSOR sêxtuplo
TTL Schottky avançada	74AS	74AS04 INVERSOR sêxtuplo
TTL Schottky avançada de baixa potência	74ALS	74ALS04 INVERSOR sêxtuplo

As diferenças entre as várias subfamílias TTL estão nas suas características elétricas, como: dissipação de potência, tempos de propagação e velocidade de comutação. Elas não diferem na disposição dos pinos ou na operação lógica realizada pelos circuitos internos. Por exemplo, o 7402, o 74S02, o 74LS02 e o 74ALS02 são todos compostos de quatro portas NOR de duas entradas. Vamos comparar as características elétricas das diferentes séries TTL no Cap. 8.

## Família CMOS

Várias subfamílias CMOS disponíveis estão relacionadas na Tabela 4-7. A série 4000 é a mais antiga das séries CMOS. Ela possui muitas das funções lógicas da família TTL, mas não foi projetada para ser *compatível pino a pino* com os dispositivos TTL. Por exemplo, o chip quádruplo NOR 4001 contém quatro portas NOR de duas entradas, como o chip TTL 7402, mas as entradas e saídas das portas do chip CMOS não têm a mesma pinagem que os sinais correspondentes no chip TTL.

TABELA 4-7 Várias séries dentro de uma família CMOS

Subfamílias CMOS	Prefixo	Exemplo de CI
CMOS de porta metálica	40	4001 portas NOR quádruplas
CMOS de porta metálica; pinagem compatível com TTL	74C	74C02 portas NOR quádruplas
CMOS de porta de silício; alta velocidade; pinagem compatível com TTL	74HC	74HC02 portas NOR quádruplas
CMOS de porta de silício; alta velocidade; pinagem compatível com TTL; eletricamente compatível com TTL	74HCT	74HCT02 portas NOR quádruplas
CMOS avançada; pinagem incompatível com TTL; eletricamente incompatível com TTL	74AC	74AC02 portas NOR quádruplas
CMOS avançada; pinagem incompatível com TTL; eletricamente compatível com TTL	74ACT	74ACT02 portas NOR quádruplas

As séries 74C, 74HC, 74HCT, 74AC e 74ACT são as mais novas da família CMOS. As três primeiras são compatíveis pino a pino com os dispositivos TTL de mesma numeração. Por exemplo, o 74C02, 74HC02 e 74HCT02 possuem a mesma pinagem que o 7402, 74LS02, e assim por diante. As séries 74HC e 74HCT operam a uma velocidade mais alta do que os dispositivos da 74C. A série 74HCT é projetada para ser *eletricamente compatível* com dispositivos TTL; isto significa que um circuito integrado 74HCT pode ser diretamente conectado a dispositivos TTL sem que seja necessário nenhum circuito de interface. As séries 74AC e 74ACT são CIs de altíssimo desempenho. Nenhum deles é compatível pino a pino com TTL. Dispositivos 74ACT são eletricamente compatíveis com TTL. Exploraremos as várias subfamílias TTL e CMOS com bastante detalhes no Cap. 8.

## Alimentação e Terra

Para utilizar CIs digitais, é necessário que se façam as conexões apropriadas aos pinos do CI. As conexões mais importantes são as de *alimentação e terra*. Estas conexões são necessárias para que o chip opere de modo correto. Observando a Fig. 4-29, podemos ver que tanto os circuitos TTL quanto os CMOS têm a fonte de alimentação ligada a um pino e a terra conectada em outro.  $V_{cc}$  é o nome dado ao pino no qual conectamos a alimentação em circuitos TTL. Nos circuitos CMOS, este pino é chamado de  $V_{DD}$ . Uma vez que muitos circuitos integrados CMOS são projetados para serem compatíveis com circuitos TTL,  $V_{cc}$  também é usado para designar o pino da fonte de alimentação.

Caso a ligação com a fonte de alimentação ou com terra não seja feita, as portas lógicas no chip não vão responder de modo correto às entradas lógicas, e ele não fornecerá os níveis lógicos de saída esperados.

## Faixas de Tensão para os Níveis Lógicos

Para dispositivos TTL,  $V_{cc}$  deve ser de +5 V. Para dispositivos CMOS,  $V_{DD}$  pode estar situado na faixa que vai de +3 a +18 V, embora +5 V seja a tensão mais usada, principalmente quando dispositivos CMOS são usados em um mesmo circuito em conjunto com dispositivos TTL.

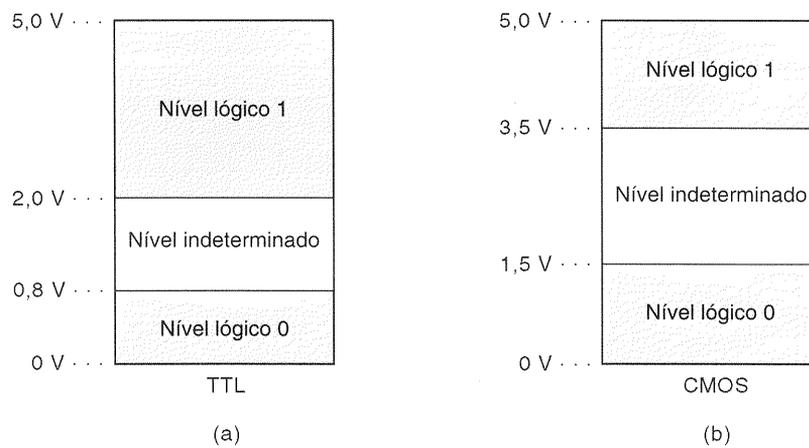


Fig. 4-30 Níveis lógicos de entrada para CIs digitais TTL e CMOS.

Para os CIs TTL padrões, as tensões de entrada aceitáveis para os níveis lógicos 0 e 1 estão definidas na Fig. 4-30(a). Um nível lógico 0 é qualquer tensão na faixa entre 0 e 0,8 V, e para o nível 1 é qualquer tensão na faixa entre 2 e 5 V. Tensões que não estão localizadas em nenhuma dessas faixas são consideradas **indeterminadas** e não devem ser usadas como entrada em nenhum dispositivo TTL. Os fabricantes de CIs não podem garantir como o circuito responderá a esses níveis que estão na faixa de indeterminação (entre 0,8 V e 2,0 V).

As faixas de tensões de entrada para circuitos integrados CMOS, que operam com  $V_{dd} = +5$  V, podem ser vistas na Fig. 4-30(b). Tensões entre 0 e 1,5 V são definidas como nível 0, e tensões entre 3,5 e 5 V são definidas como nível 1. A faixa de indeterminação inclui tensões na faixa entre 1,5 e 3,5 V.

### Entradas Não-Conectadas

O que acontece quando as entradas de um CI não estão conectadas a nenhum sinal lógico? Uma entrada desconectada é geralmente chamada de entrada em **flutuação**. A resposta para a pergunta anterior será uma para a família TTL e uma outra diferente para a CMOS.

Uma entrada TTL em flutuação funciona exatamente como se ela estivesse em nível 1. Em outras palavras, o CI vai responder como se tivéssemos aplicado um nível lógico ALTO a esta entrada. Essa característica é bastante usada quando se testa um circuito TTL. Um técnico preguiçoso poderia deixar determinadas entradas desconectadas em vez de conectá-las a um nível ALTO. Embora seja teoricamente correto, não é uma boa prática no projeto de circuitos, uma vez que uma entrada TTL em aberto é extremamente sensível a ruídos, o que pode vir a afetar o bom funcionamento do circuito.

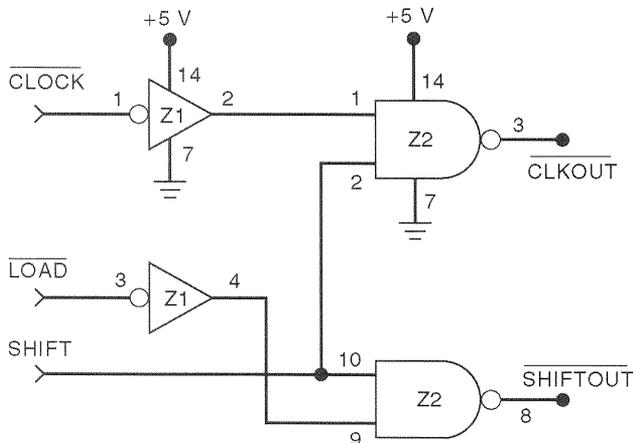
Um voltímetro ou um osciloscópio que esteja medindo uma entrada em aberto fornecerá como leitura um valor de tensão entre 1,4 e 1,8 V. Apesar de o valor lido estar situado na faixa de indeterminação para a família TTL, ele produzirá a mesma resposta que um nível 1 produziria. Lembre-se dessa característica das entradas em flutuação quando estiver pesquisando falhas em circuitos TTL.

Deixar uma entrada CMOS em flutuação pode ter resultados desastrosos. O CI pode superaquecer e possivelmente se danificar. Por essa razão, todas as entradas de circuitos integrados CMOS devem ser conectadas a um nível lógico definido (ALTO ou BAIXO) ou à saída de um outro CI. A tensão medida em uma entrada CMOS em flutuação varia em função do ruído presente, e portanto não age como um nível 0 ou 1. Isto faz com que o nível de tensão de saída oscile em função do ruído existente na entrada.

### Diagramas de Circuitos Lógicos

Um diagrama de um circuito lógico mostra *todas* as conexões, a numeração dos pinos, os números dos CIs, os valores dos componentes, os nomes dos sinais e as tensões de alimentação do circuito. A Fig. 4-31 mostra um diagrama típico para um circuito lógico simples. Examine-o com cuidado e observe os seguintes pontos importantes:

1. O circuito usa portas lógicas de dois CIs diferentes. Os dois INVERSORES fazem parte do chip 74HC04, ao qual foi dada a designação Z1. O 74HC04 possui seis INVERSORES, dois dos quais são usados neste circuito, e foram associados ao chip através da designação Z1. Do mesmo modo, as portas NAND fazem parte do chip 74HC00, que contém quatro portas. Todas as portas do chip recebem a designação Z2. Referindo-se a cada porta por Z1, Z2, Z3 etc., é possível determinar a que chip pertence cada porta. Isto é essencialmente importante em circuitos mais complexos que contenham muitos CIs com muitas portas por CI.
2. O número de cada pino de entrada e de saída está indicado no diagrama. A numeração dos pinos, juntamente com a designação dos CIs, torna bastante simples a tarefa de referenciar qualquer ponto do circuito. Por exemplo, Z1, pino 2 se refere à saída do INVERSOR situado na parte superior do diagrama. Do mesmo modo, podemos dizer que Z1, pino 4 está conectado a Z2, pino 9.
3. As conexões da fonte de alimentação e do terra de cada CI são mostradas no diagrama. Por exemplo, Z1, pino 14 está conectado a +5 V e Z1, pino 7 está conectado a terra. Essas duas conexões fornecem alimentação e terra para *todos* os seis INVERSORES que fazem parte de Z1.



CI	Tipo
Z1	74HC04 INVERSOR sêxtuplo
Z2	74HC00 NAND quádruplo

Fig. 4-31 Diagrama típico de um circuito lógico.

Os fabricantes de equipamentos eletrônicos geralmente fornecem diagramas esquemáticos detalhados como o da Fig. 4-31. Esses diagramas são de grande importância quando precisamos pesquisar a causa de um problema que está ocorrendo no circuito. Escolhemos identificar os CIs como Z1, Z2, Z3 e assim por diante. Outras designações bastante usadas são CI1, CI2, CI3 etc. e U1, U2, U3 etc.

### Questões de Revisão

1. Qual é o tipo mais comum de encapsulamento de CIs digitais?
2. Enumere as seis classificações existentes para a complexidade de CIs digitais.
3. *Verdadeiro* ou *falso*: Um 74S74 contém a mesma lógica e a mesma disposição de pinos que o 74LS74.
4. *Verdadeiro* ou *falso*: Um 74HC74 contém a mesma lógica e a mesma disposição de pinos que o 74AS74.
5. Quais subfamílias CMOS não são compatíveis pino a pino com a família TTL?
6. Qual é a faixa aceitável de tensão de entrada para um nível lógico 0 na família TTL? Qual é a faixa para o nível 1?
7. Repita a Questão 6 para um circuito CMOS operando com  $V_{DD} = 5$  V.
8. Como um circuito TTL responde a uma entrada em flutuação?
9. Como um circuito CMOS responde a uma entrada em flutuação?
10. Quais são as subfamílias CMOS que podem ser conectadas a TTL sem que seja necessária a utilização de circuitos de interface?

## 4-10 PESQUISA DE FALHAS EM SISTEMAS DIGITAIS

Existem três passos básicos a serem seguidos quando estamos depurando, isto é, pesquisando problemas ou falhas em um circuito:

1. **Detecção da falha.** Observe a operação do circuito (ou sistema) e compare-a com a operação correta esperada.

2. **Isolamento da falha.** Realize testes e faça medições que o ajudem a isolar a falha.

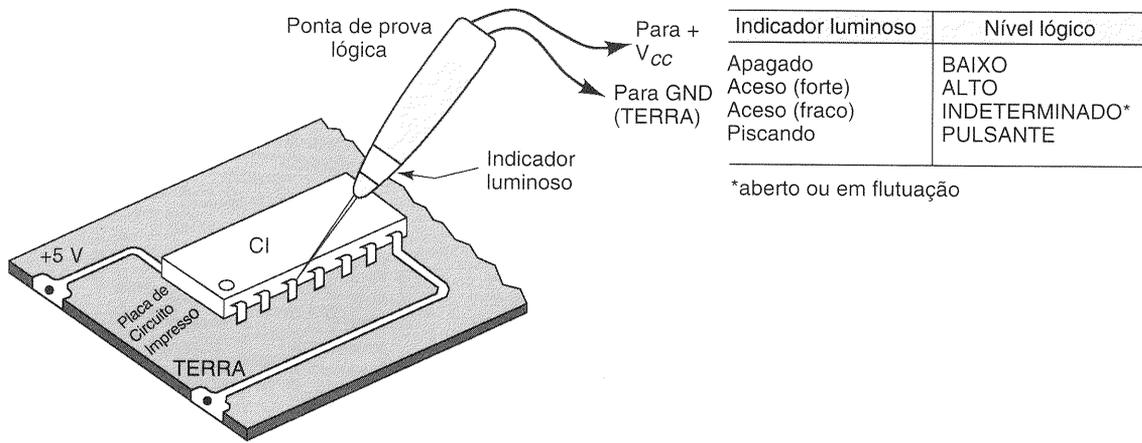
3. **Correção da falha.** Troque o componente defeituoso, conserte a ligação defeituosa, remova o curto-circuito e assim por diante.

Apesar de esses passos parecerem relativamente simples, o procedimento real a ser seguido depende muito do tipo e da complexidade do circuito, das ferramentas e da documentação disponíveis.

Boas técnicas para depuração de circuitos só podem ser aprendidas em um ambiente de laboratório, através da experimentação e da prática na depuração de circuitos e sistemas reais. Não existe melhor maneira de se tornar hábil em depuração de circuitos do que praticar o máximo possível. Não importa quantos bons livros você leia, nenhum deles será capaz de lhe transmitir esse tipo de experiência. Eles podem, entretanto, ajudá-lo a desenvolver sua capacidade de análise, que é fundamental para uma depuração eficiente. Inicialmente, vamos descrever os tipos de falhas mais comuns em sistemas, que são compostos basicamente de CIs digitais, e diremos a você como reconhecê-los. Estudaremos casos típicos para ilustrar os processos de análise envolvidos na manutenção de circuitos (ou sistemas). Além disso, no final deste capítulo, apresentaremos problemas que lhe fornecerão a oportunidade de utilizar estes processos de análise, para chegar a conclusões sobre circuitos digitais defeituosos.

Para as discussões e exercícios sobre depuração que faremos neste livro, presumimos que o estudante tem acesso aos instrumentos básicos para depuração de circuitos, como por exemplo: *ponta de prova lógica*, *osciloscópio*, *gerador de pulsos*, *rastreador de corrente* etc. Queremos deixar bem claro que a ferramenta mais importante e eficiente na depuração é o seu cérebro, e esta é a ferramenta que esperamos desenvolver ao apresentar princípios básicos, técnicas de depuração, exemplos e problemas neste capítulo e nos seguintes.

Nas próximas três seções sobre depuração, utilizaremos apenas o seu cérebro e uma **ponta de prova lógica** como a que aparece na Fig. 4-32. As outras ferramentas serão usadas nos capítulos seguintes. A ponta de prova possui uma ponta de metal que deve tocar o ponto específico do circuito que queremos testar. A figura mostra o teste do pino



**Fig. 4-32** Uma ponta de prova lógica é utilizada para monitorar os níveis lógicos no pino do CI ou em qualquer outro ponto acessível do circuito.

3 do CI. Também podemos utilizar a ponta de prova para testar uma trilha em uma placa de circuito impresso, um fio desencapado, um pino de um conector, um terminal de um componente discreto como um transistor, ou qualquer outro ponto que seja condutor no circuito. O nível lógico presente na ponta de prova será indicado pelo estado do indicador luminoso ou LED existente na ponta de prova. Os quatro estados possíveis são mostrados na tabela da Fig. 4-32. Observe que o nível lógico *indeterminado* produz uma intensidade fraca no indicador luminoso. Isto também acontece quando a ponta de prova testa um ponto do circuito que está em aberto ou flutuando, isto é, não está conectado a nenhuma fonte de tensão.

## 4-11 FALHAS INTERNAS DOS CIs DIGITAIS

As falhas internas mais comuns dos CIs digitais são:

1. Mau funcionamento de circuitos internos do CI.
2. Entradas ou saídas em curto com a terra ou  $V_{cc}$ .
3. Circuito aberto nas entradas ou saídas.
4. Curto-circuito em dois pinos do CI (desde que não seja terra ou  $V_{cc}$ ).

Vamos descrever cada um desses tipos de falha.

### Mau Funcionamento de Circuitos Internos do CI

Este problema é causado, na maioria das vezes, quando um dos componentes internos do circuito está danificado ou opera fora de suas especificações. Quando isto acontece, as saídas do CI não respondem de modo correto às suas entradas. Não existe nenhum modo de prever qual será o comportamento da saída, uma vez que este depende do componente interno que está apresentando problema. Exemplos desse tipo de falha seriam: um curto entre a base e o emissor do transistor  $Q_1$  ou um valor extremamente alto de resistência para  $R_2$  no INVERSOR TTL da Fig. 4-29(a). Esse tipo de falha interna não é tão comum quanto as outras três.

### Entradas Internamente em Curto com a Terra ou com a Fonte de Alimentação

Este tipo de falha interna faz com que a entrada do CI fique permanentemente em estado BAIXO ou ALTO. A Fig. 4-33(a) mostra o pino de entrada 2 da porta NAND em curto com a terra, internamente ao CI. Isto faz com que o pino 2 esteja sempre no estado BAIXO. Se um sinal lógico  $B$  for conectado ao pino 2, haverá um curto-circuito entre  $B$  e a terra. Portanto, esse tipo de falha vai afetar a saída do dispositivo que estiver gerando o sinal  $B$ .

Do mesmo modo, um pino de entrada de um CI poderia estar internamente em curto com +5 V, como na Fig. 4-33(b). Isto faz com que este pino esteja sempre no nível ALTO. Se esta entrada for conectada a um sinal lógico  $A$ , haverá um curto entre  $A$  e +5 V.

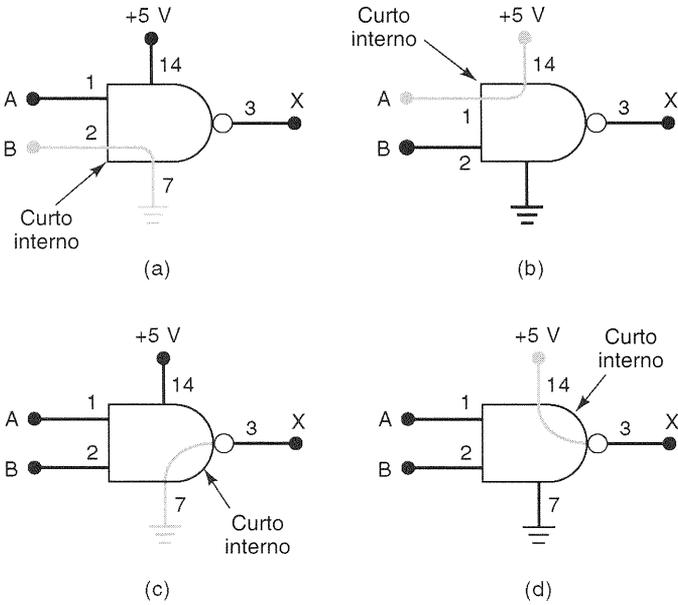
### Saídas Internamente em Curto com a Terra ou com a Fonte de Alimentação

Este tipo de falha faz com que o pino de saída fique permanentemente no estado BAIXO ou ALTO. A Fig. 4-33(c) mostra o pino 3 da porta NAND internamente em curto com a terra. Esta saída estará sempre em BAIXO e não responderá as condições aplicadas aos pinos de entrada 1 e 2. Em outras palavras, as entradas  $A$  e  $B$  não terão efeito sobre a saída  $X$ .

Um pino de saída de um CI também pode estar internamente em curto com +5 V, como podemos ver na Fig. 4-33(d). Isto faz com que o pino de saída 3 esteja em ALTO independentemente do estado dos sinais nos pinos de entrada. Observe que este tipo de falha não afeta os sinais lógicos presentes nas entradas.

### EXEMPLO 4-24

Observe o circuito da Fig. 4-34. Um estudante pode utilizar uma ponta de prova para determinar as condições existentes em vários pinos do CI. Os resultados estão mostrados na tabela da figura. Examine estes resultados e determine



**Fig. 4-33** (a) Entrada do CI internamente em curto com a terra; (b) entrada do CI internamente em curto com a fonte de alimentação. Esses dois tipos de falha forçam o sinal de entrada que está neste pino a ficar no mesmo estado. (c) Saída de um CI internamente em curto com a terra; (d) saída de um CI internamente em curto com a fonte de alimentação. Esses dois tipos de falha não afetam os sinais que estão nas entradas do CI.

se o circuito está operando corretamente. Caso não esteja, indique algumas das possíveis falhas.

**Solução**

O pino de saída 4 do INVERSOR deveria estar pulsando, uma vez que a sua entrada está. Entretanto, os resultados mostram que o pino 4 está sempre em BAIXO. Este pino está conectado ao pino 1 de Z2 e portanto faz com que a saída da porta NAND seja ALTA. A partir desta discussão, podemos enumerar três possíveis falhas que causariam esse comportamento.

A primeira seria devido à existência de um componente interno danificado, que impediria que o INVERSOR funcionasse corretamente. A segunda seria devido a um curto-circuito interno entre o pino 4 e o terra de Z1. Isto faria com que o pino 4 permanecesse sempre em BAIXO, e finalmente a terceira seria o resultado de um curto-circuito interno entre o pino 1 e o terra de Z2. Isto impediria que o estado do pino de saída do INVERSOR se modificasse.

Além dessas possíveis falhas, pode haver curtos-circuitos externos entre a terra e qualquer ponto na ligação entre o pino 4 de Z1 e o pino 1 de Z2. Falaremos sobre como determinar a falha presente no exemplo seguinte.

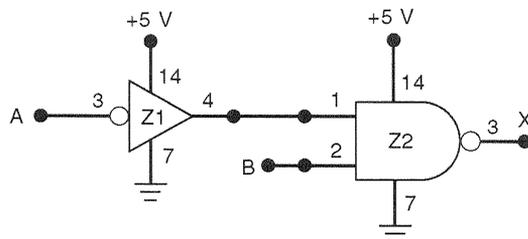
**Circuito Aberto nas Entradas ou Saídas**

Às vezes é possível que o fio extremamente fino que liga o pino do CI ao seu circuito interno se quebre, produzindo um circuito aberto. A Fig. 4-35 mostra esta situação para uma entrada (pino 13) e para uma saída (pino 6). Se o sinal for aplicado ao pino 13, ele não alcançará a entrada da porta NAND 1 e, portanto, não terá efeito sobre a saída da NAND 1. O circuito aberto deixa a entrada em flutuação. Como foi dito anteriormente, dispositivos TTL responderão a uma entrada em flutuação como se ela tivesse um nível lógico 1. Já os dispositivos CMOS vão responder de modo imprevisível e podem até ser danificados por superaquecimento.

Um circuito aberto na saída do NAND 4 impede que este sinal chegue ao pino 6, e portanto não há uma tensão estável presente neste pino. Se ele for conectado a uma entrada de um outro CI, ele vai produzir uma condição de flutuação nesta entrada.

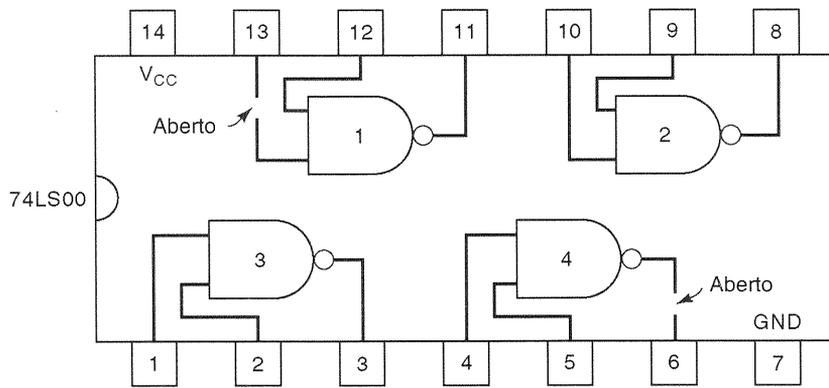
**EXEMPLO 4-25**

O que uma ponta de prova lógica indicaria no pino 13 e no pino 6 do CI da Fig. 4-35?



Pino	Condição
Z1-3	Pulsante
Z1-4	BAIXO
Z2-1	BAIXO
Z2-2	ALTO
Z2-3	ALTO

**Fig. 4-34** Exemplo 4-24.



**Fig. 4-35** Um CI com uma entrada desconectada internamente não responderá aos sinais aplicados naquele pino de entrada. Uma saída internamente desconectada vai produzir uma tensão imprevisível neste pino de saída.

**Solução**

No pino 13, a ponta de prova indicaria o nível lógico do sinal externo que está conectado ao pino 13 (e que não aparece na figura). No pino 6, a ponta de prova indicaria uma luz fraca referente a um nível lógico indeterminado. Isto ocorre porque o nível de saída da porta NAND não chega até o pino 6.

A partir de afirmações feitas anteriormente sobre entradas TTL em aberto, você poderia esperar que a tensão no pino 1 de Z2 estivesse entre 1,4 e 1,8 V, e isto deveria ter sido indicado como um nível indeterminado pela ponta de prova. Isto seria verdadeiro se o circuito aberto fosse *externo* ao chip NAND. Não existe circuito aberto entre o pino 4 de Z1 e o pino 1 de Z2, e portanto a tensão existente no pino 4 de Z1 chega até o pino 1 de Z2, logo o circuito está aberto no *interior* do chip.

**EXEMPLO 4-26**

Observe o circuito da Fig. 4-36 e as anotações feitas das indicações da ponta de prova. Quais são as possíveis falhas que poderiam causar essas indicações? Suponha que os CIs são TTL.

**Solução**

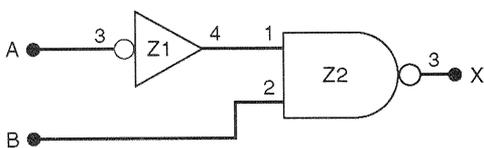
Examinando as anotações feitas, podemos verificar que o INVERSOR está funcionando corretamente, mas a saída da porta NAND está incompatível com suas entradas. A saída da porta NAND deveria estar em ALTO, uma vez que o pino de entrada 1 está em BAIXO. Este nível BAIXO deveria impedir que a porta NAND respondesse aos pulsos no pino 2. É provável que este nível BAIXO não esteja chegando ao circuito interno da porta NAND, em virtude de a entrada estar internamente em aberto. Como este CI é TTL, esse circuito aberto iria produzir o mesmo efeito de um nível ALTO no pino 1. Se o CI fosse CMOS, este circuito aberto interno no pino 1 poderia produzir um nível indeterminado na saída e possivelmente causar a destruição do CI por superaquecimento.

**Curto-Circuito entre Dois Pinos**

Um curto interno entre dois pinos de um CI obriga que os sinais lógicos nestes pinos sejam sempre idênticos. Sempre que dois sinais supostamente diferentes apresentarem as mesmas variações em seus níveis lógicos, é provável que estes sinais estejam em curto.

Considere o circuito da Fig. 4-37, onde os pinos 5 e 6 de uma porta NOR estão em curto por dentro do CI. Este curto faz com que os pinos de saídas dos INVERSOres estejam conectados, obrigando que os sinais dos pinos Z1-2 e Z1-4 sejam idênticos, mesmo quando seus respectivos sinais de entrada estejam tentando produzir saídas diferentes. Para ilustrar, considere as formas de onda das entradas mostradas no diagrama. Apesar de estas formas de onda serem diferentes, as formas de onda nas saídas Z1-2 e Z1-4 são as mesmas.

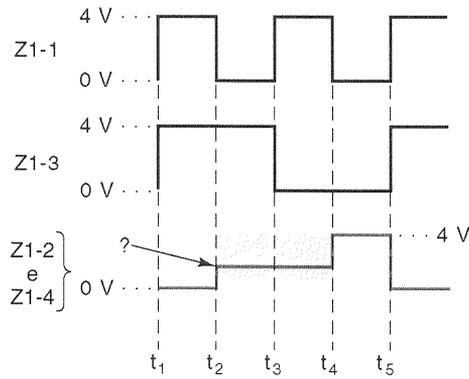
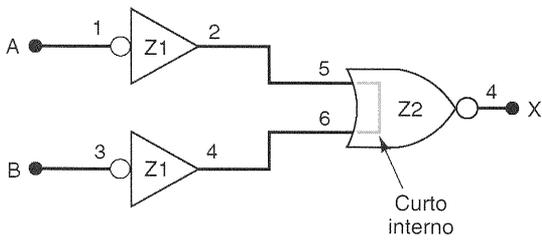
Durante o intervalo de tempo entre  $t_1$  e  $t_2$ , os dois INVERSOres têm suas entradas em ALTO e portanto estão tentando produzir saídas em BAIXO. Logo, o fato de as saídas estarem em curto não afeta o resultado. Entretanto, durante os intervalos de  $t_2$  até  $t_3$  e de  $t_3$  até  $t_4$ , um dos INVERSOres



Pino	Condição
Z1-3	ALTO
Z1-4	BAIXO
Z2-1	BAIXO
Z2-2	Pulsante
Z2-3	Pulsante

Nota: As ligações dos CIs a  $V_{cc}$  e a terra foram omitidas.

**Fig. 4-36** Exemplo 4-26.



**Fig. 4-37** Quando dois pinos estão internamente em curto, os sinais que estão conectados a esses pinos são forçados a ser idênticos, possuindo geralmente três níveis distintos.

está tentando fazer com que sua saída vá para ALTO, enquanto o outro está tentando fazer com que sua saída vá para BAIXO. Nessa situação, a tensão que aparece de fato nas saídas em curto depende dos circuitos internos dos CIs. Para dispositivos TTL, esta tensão estará, geralmente, próxima ao extremo superior da faixa que determina o nível 0 (isto é, próxima a 0,8 V), embora também seja possível que ela possa estar na faixa de indeterminação. Para o caso de dispositivos CMOS, a tensão estará geralmente na faixa para a qual o nível lógico é indeterminado.

Sempre que você vir uma forma de onda como a do sinal Z1-2 e Z1-4 na Fig. 4-37, isto é, como três níveis diferentes, você deve suspeitar de que os dois sinais de saída podem estar em curto.

**Questões de Revisão**

1. Enumere as diferentes falhas internas dos CIs digitais.
2. Que tipo de falha interna pode produzir sinais como aqueles que contêm três níveis diferentes de tensão?
3. O que a ponta de prova indicaria em Z1-2 e Z1-4 da Fig. 4-37 se  $A = 0$  e  $B = 1$ ?

**Linhas de Sinal Abertas**

Esta categoria inclui qualquer falha que produza uma descontinuidade no caminho elétrico, de tal modo que o nível de tensão (ou sinal) seja impedido de ir de um ponto a outro. Algumas das causas de linhas de sinal abertas são:

1. Fio partido.
2. Conexão com solda fria ou conexão com *wire-wrap*\* frouxa.
3. Cortes ou fissuras nas trilhas do circuito impresso (alguns são tão pequenos que são difíceis de ver sem o auxílio de uma lente de aumento).
4. Pino do CI dobrado ou quebrado.
5. Mau contato no soquete do CI.

Esse tipo de falha no circuito pode ser descoberto através de uma inspeção visual cuidadosa e de uma posterior verificação da continuidade (isto é, da baixa resistência em um caminho elétrico), desconectando a fonte de alimentação e colocando o ohmímetro entre os dois pontos em questão.

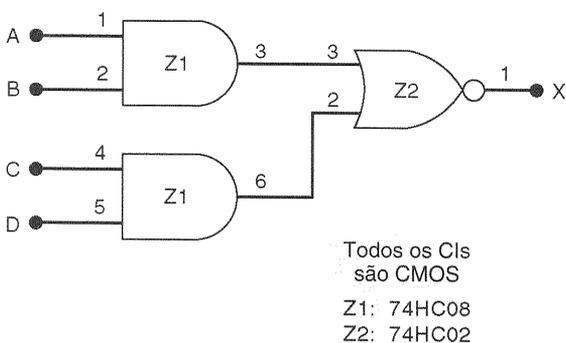
**EXEMPLO 4-27**

Considere o circuito CMOS da Fig. 4-38 e as indicações da ponta de prova. Qual é a falha mais provável deste circuito?

\*Técnica de montagem de circuitos digitais utilizada principalmente na construção de protótipos. Os CIs são montados em soquetes especiais de pinos longos sobre os quais os fios são enrolados (daí o nome *wire-wrap*). (N. T.)

**4-12 FALHAS EXTERNAS**

Vimos como reconhecer os efeitos dos vários tipos de falhas internas dos CIs digitais. Muitas coisas erradas podem acontecer fora do CI, fazendo com que este não funcione corretamente. Descrevemos as mais comuns nesta seção.



Pino	Condição
Z1-1	Pulsante
Z1-2	ALTO
Z1-3	Pulsante
Z1-4	BAIXO
Z1-5	Pulsante
Z1-6	BAIXO
Z2-3	Pulsante
Z2-2	Indeterminado
Z2-1	Indeterminado

**Fig. 4-38** Exemplo 4-27.

## Solução

O nível indeterminado na saída da porta NOR é provavelmente devido a um nível indeterminado na entrada no pino 2. Como Z1-6 está em BAIXO, este mesmo nível deveria estar presente em Z2-2. Não é difícil perceber que o nível BAIXO de Z1-6 não está chegando a Z2-2 e portanto deve haver um circuito aberto no caminho do sinal entre estes dois pontos. A localização desse circuito aberto pode ser determinada começando a rastrear com uma ponta de prova este nível BAIXO, a partir de Z1-6 e indo em direção a Z2-2. Quando a ponta de prova indicar um nível indeterminado, teremos a localização do circuito aberto.

## Linhas de Sinal em Curto

Este tipo de falha causa o mesmo efeito que um curto interno entre pinos de um CI. Ele faz com que dois sinais sejam exatamente iguais. Uma linha de sinal pode ser colocada em curto com a terra ou com  $V_{cc}$ , em vez de uma outra linha de sinal. Neste caso, o sinal será forçado para um estado BAIXO ou ALTO. As principais causas para curtos inesperados entre dois pontos do circuito são:

- 1. Ligações malfeitas.** Um exemplo disto é retirar uma grande parte do isolamento das pontas de fios muito próximos.
- 2. Pontes de solda.** Elas nada mais são do que respingos de solda que colocam dois ou mais pontos em curto. Comumente ocorrem entre pontos que são muito próximos, como por exemplo pinos adjacentes de um chip.
- 3. Corrosão incompleta.** O cobre existente entre duas trilhas adjacentes em uma placa de circuito impresso não foi completamente removido durante a corrosão da placa.

Mais uma vez, uma inspeção visual cuidadosa geralmente identifica esse tipo de falha. Além disso, uma verificação com um ohmímetro pode indicar que dois pontos no circuito estão em curto.

## Falha na Fonte de Alimentação

Todos os sistemas digitais possuem uma ou mais fontes de alimentação para fornecer as tensões  $V_{cc}$  e  $V_{dd}$  de que os chips necessitam. Uma fonte de alimentação defeituosa ou em sobrecarga (suprindo mais corrente do que é capaz) fornecerá tensões de alimentação mal-reguladas aos CIs, fazendo com que eles não funcionem ou o façam de modo imprevisível.

Uma fonte de alimentação pode sair de regulação devido a uma falha em seus circuitos internos, ou porque os circuitos que ela está alimentando estão drenando mais corrente do que a fonte foi projetada para suprir. Isto ocorre se um chip ou um componente tem um defeito que faça com que ele drene muito mais corrente do que o normal.

É uma boa prática de depuração verificar se os níveis de tensão em cada fonte de alimentação do circuito estão dentro das faixas de operação especificadas. Além disso, também é uma boa idéia utilizar um osciloscópio para verificar se não há uma quantidade excessiva de *ripple* (ondulação) sobre o nível de tensão contínua e para observar se a ten-

são da fonte permanece regulada durante a operação do sistema.

Um dos sinais mais comuns de uma fonte defeituosa é que um ou mais CIs funcionam de modo incorreto ou simplesmente não funcionam. Alguns CIs são mais tolerantes a variações na fonte de alimentação e podem funcionar corretamente, enquanto outros não. Você sempre deve verificar os níveis de tensão nos pinos de alimentação e terra de cada CI que parece não funcionar corretamente.

## Carregamento da Saída

Quando um circuito digital tem sua saída conectada a muitas entradas, sua capacidade de fornecer corrente pode ser excedida fazendo com que a tensão de saída caia na faixa para a qual o nível lógico é indeterminado. Este efeito é chamado *carregamento* do sinal (o que ocorre na verdade é uma sobrecarga do sinal de saída) e é geralmente causado por um projeto malfeito ou por ligações incorretas.

### Questões de Revisão

1. Quais são os tipos de falhas externas mais comuns?
2. Enumere algumas das causas de circuitos abertos nos caminhos elétricos dos sinais.
3. Quais os sintomas causados por uma fonte de alimentação defeituosa?
4. De que modo o carregamento pode afetar o nível de tensão de um CI?

## 4-13 ESTUDO DE UM CASO DE PESQUISA DE FALHAS

O exemplo seguinte ilustra o processo de análise envolvido na depuração de circuitos digitais. Embora este exemplo seja de um circuito lógico combinacional bastante simples, o raciocínio e os procedimentos utilizados podem ser aplicados a circuitos digitais mais complexos, que encontraremos nos capítulos seguintes.

### EXEMPLO 4-28

Considere o circuito da Fig. 4-39. A saída  $Y$  deveria estar em ALTO para as seguintes condições:

1.  $A = 1$  e  $B = 0$ , independentemente do nível de  $C$ .
2.  $A = 0$ ,  $B = 1$  e  $C = 1$ .

Você mesmo pode verificar essas condições.

Quando o circuito é testado, o estudante observa que a saída  $Y$  vai para ALTO sempre que  $A$  ou  $C$  estão em ALTO, independentemente do nível de  $B$ . Ele faz as medições com uma ponta de prova para o caso em que  $A = B = 0$  e  $C = 1$  e anota as indicações, conforme pode ser visto na Fig. 4-39.

Examine os níveis observados e relacione as possíveis causas do mau funcionamento. Depois disso, desenvolva um procedimento com etapas para determinar qual é a falha de fato.

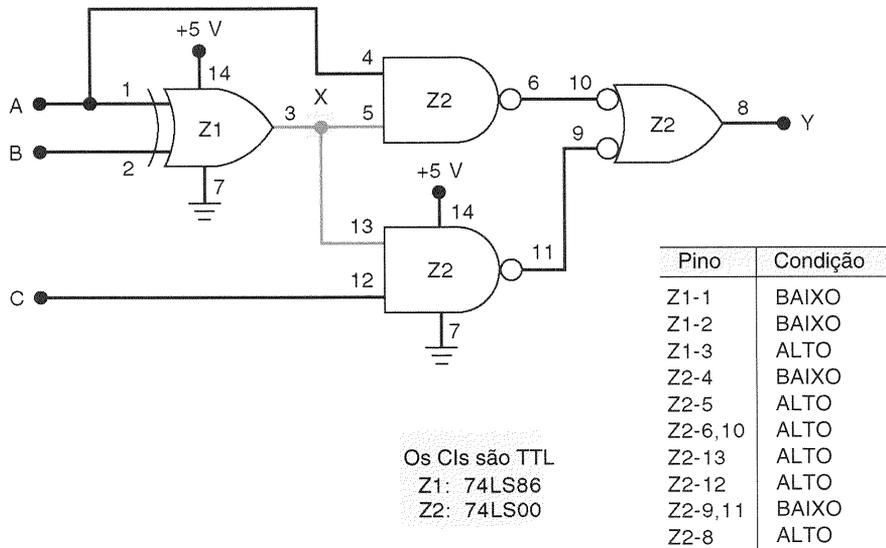


Fig. 4-39 Exemplo 4-28.

### Solução

Todas as saídas das portas NAND estão corretas em função dos níveis presentes em suas entradas. Entretanto, a porta EX-OR deveria ter produzido um nível BAIXO no pino de saída 3, uma vez que ambas as entradas estão em BAIXO. Aparentemente, Z1-3 está sempre em ALTO, mesmo quando suas entradas indicam que ela deveria estar em BAIXO. As possíveis causas para que isto aconteça estão listadas a seguir:

1. Um componente interno danificado em Z1 que impede que a saída vá para BAIXO.
2. Um curto externo entre  $V_{cc}$  e qualquer ponto ao longo do caminho de condução para o ponto X.
3. Um curto interno entre Z1-3 e  $V_{cc}$ .
4. Um curto interno entre Z2-5 e  $V_{cc}$ .
5. Um curto interno entre Z2-13 e  $V_{cc}$ .

Todas essas alternativas, com exceção da primeira, vão colocar em curto o ponto X (e qualquer pino conectado a ele) e  $V_{cc}$ .

O procedimento a seguir pode ser usado para isolar a falha. Este não é o único que pode ser usado, e, como afirmamos anteriormente, o procedimento a ser usado em cada caso depende da pessoa que está fazendo a depuração e dos equipamentos de teste disponíveis.

1. Verifique os níveis de tensão dos pinos de  $V_{cc}$  e terra de Z1. Embora seja improvável que a ausência de um desses níveis pudesse fazer com que Z1-3 ficasse em ALTO, é uma boa prática fazer essa verificação em qualquer CI que esteja produzindo uma saída incorreta.
2. Desligue a fonte de alimentação do circuito e use um ohmímetro para verificar se existe um curto (uma resistência menor que  $1 \Omega$ ) entre o ponto X e qualquer ponto conectado a  $V_{cc}$  (como Z1-14 ou Z2-14). Se não houver indicação de um curto, as últimas quatro possibilida-

des da nossa lista podem ser eliminadas. Isto significa que Z1 está danificado e deve ser trocado.

3. Se na etapa 2 existir um curto entre o ponto X e  $V_{cc}$ , faça uma inspeção visual na placa do circuito e procure pontes de solda, resíduos de cobre não-corroído, fios desencapados em contato e qualquer outra possível causa de curto com  $V_{cc}$ . Um lugar provável para encontrar uma ponte de solda é entre os pinos adjacentes 13 e 14 de Z2, pois o pino 14 é conectado ao  $V_{cc}$  e o pino 13 ao ponto X. Se um curto externo for encontrado, remova-o e faça uma verificação com um ohmímetro para determinar se o ponto X não está mais em curto com  $V_{cc}$ .
4. Se a etapa 3 não revelar nenhum curto externo, as possíveis causas que nos restam são curtos internos entre  $V_{cc}$  e Z1-3, Z2-13 ou Z2-5. Um destes está colocando o ponto X em curto com  $V_{cc}$ .

Para determinar qual desses CIs é o culpado, você deve desconectar cada um deles do ponto X, *um de cada vez*, e verificar se o curto com  $V_{cc}$  permanece após cada desconexão, pois quando o pino que está internamente em curto com  $V_{cc}$  for desconectado o ponto X não estará mais em curto com  $V_{cc}$ .

O processo de desconectar cada um dos pinos suspeitos do ponto X pode ser fácil ou difícil, dependendo de como o circuito está montado. Se os CIs estão em soquetes, tudo o que você tem a fazer é retirar o CI do soquete, dobrar o pino suspeito para fora e recolocar o CI no soquete. Se os CIs estão soldados na placa de circuito impresso, você terá que cortar a trilha na qual o pino está conectado, ou o próprio pino. Após o teste ter sido feito, você terá que fazer reparos na trilha ou no pino, conforme o caso.

Existe uma técnica de depuração que torna desnecessário dobrar pinos ou cortar trilhas quando estamos tentando isolar um curto. Ela envolve a utilização de um equipamento chamado *rastreador de corrente*. Este instrumento é capaz de rastrear o fluxo de corrente que atravessa um curto-circuito, através da mudança do fluxo magnético em torno

do condutor onde se verifica o curto. Vamos examinar isto em detalhe no Cap. 8.

O Exemplo 4-28, apesar de ser bastante simples, mostra os tipos de raciocínio que alguém que faz depuração de circuitos (engenheiro, técnico ou estudante) deve empregar de modo a isolar a falha do circuito. Você terá a oportunidade de desenvolver suas habilidades de depuração de circuitos resolvendo os Problemas 4-34 a 4-44.

## 4-14 LÓGICA PROGRAMÁVEL\*

Como foi visto anteriormente, a última etapa no projeto de um circuito lógico é reunir os CIs necessários e fazer as conexões apropriadas, de modo que as saídas deste circuito sejam as funções lógicas das entradas que desejamos. Uma vez feito o projeto, este circuito pode ser testado. Você provavelmente já fez isso muitas vezes no laboratório.

Mais adiante neste livro gastaremos bastante tempo aprendendo uma outra maneira de se implementar funções lógicas, que é bastante diferente desta que estamos aprendendo agora. Ela utiliza algo a que chamamos *lógica programável* e é especialmente útil na implementação de circuitos mais complexos que contêm dezenas ou centenas de portas lógicas. Discutiremos os detalhes da lógica programável mais tarde, mas o conceito básico será introduzido aqui com o auxílio da Fig. 4-40. O bloco retangular representa conceitualmente um exemplo de um **dispositivo de lógica programável (PLD — programmable logic device)**, que é um circuito integrado que contém um arranjo particular de portas lógicas. Existem muitos tipos de PLDs; todos eles contêm muito mais do que as poucas portas lógicas que vemos na Fig. 4-40.

Entretanto, usaremos este exemplo simples para mostrar a idéia básica de toda lógica programável.

Podemos reconhecer a lógica utilizada nesse PLD simplificado como uma estrutura do tipo soma-de-produtos, onde as saídas das portas AND são conectadas a uma porta OR. A saída  $X$  será uma função do tipo soma-de-produtos das entradas  $A$  e  $B$ . A função de saída implementada dependerá de quais saídas das portas AND são conectadas às entradas da porta OR. Na figura, todas as saídas das portas AND são conectadas às entradas da porta OR através dos elos de ligação 1, 2, 3 e 4. Estes elos podem ficar intactos, como mostrado na figura, ou podem ser seletivamente abertos, para desconectar a saída correspondente da entrada da porta OR. Por exemplo, se os elos 1, 2 e 3 fossem abertos, apenas a porta AND 4 estaria conectada, e a saída da porta OR seria  $X = AB$ . Se os elos 1 e 4 fossem abertos, a saída seria  $X = \bar{A}\bar{B} + \bar{A}B$ . O circuito interno é tal que um elo aberto produz um nível BAIXO na sua entrada da porta OR.

Desse modo, podemos implementar qualquer expressão do tipo soma-de-produtos de duas variáveis abrindo os elos apropriados. O chip PLD vem com todos os seus elos intactos, e todos eles estão dentro do chip. Como isto é feito? Bem, descobriremos mais tarde. Por enquanto, vamos dizer que o PLD tem entradas de programação (representadas pela grande seta no diagrama) que podemos usar para, de algum modo, abrir determinados elos de modo a implementar a função lógica em particular que queremos. Este procedimento é chamado de *programação* do PLD. As entradas de programação são usadas somente durante o processo de programação, para configurar as conexões internas do chip. Uma vez feito isto, o PLD foi *programado* para realizar a operação lógica sobre suas entradas  $A$  e  $B$  e está pronto para ser utilizado com esse objetivo.

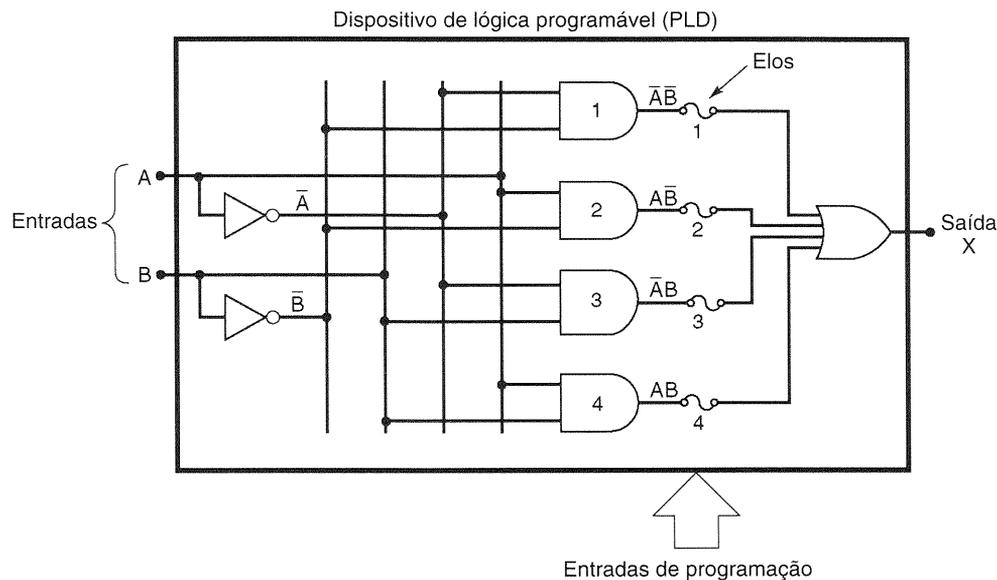


Fig. 4-40 Exemplo simplificado de um dispositivo de lógica programável.

\*Esta seção pode ser omitida sem que haja perda de continuidade.

A estrutura mostrada na Fig. 4-40 é apenas um dos muitos tipos de estruturas de PLD que estudaremos nos Caps. 11 e 12. É relativamente fácil ver como essa estrutura AND-OR pode ser expandida para acomodar um número maior de entradas. Por exemplo, para três entradas, o PLD teria três INVERSORES, oito portas AND (uma para cada produto AND possível das três variáveis), oito elos e uma porta OR de 8 entradas.

**Questões de Revisão**

1. Qual seria a função de saída do PLD se os elos 1 e 2 na Fig. 4-40 fossem abertos?
2. Qual seria a função se todos os elos permanecessem intactos?
3. Descreva os componentes de um PLD que está estruturado como visto na Fig. 4-40 e que possui *quatro* entradas.

**RESUMO**

1. As duas formas gerais para as expressões lógicas são soma-de-produtos e produto-de-somas.
2. Um dos métodos de projeto de circuitos lógicos combinacionais é constituído das seguintes etapas: (1) construir sua tabela-verdade; (2) converter a tabela-verdade em uma expressão do tipo soma-de-produtos; (3) simplificar esta expressão utilizando álgebra booleana ou mapa de Karnaugh; (4) implementação da expressão final obtida.
3. O mapa de Karnaugh é um método gráfico de representar a tabela-verdade do circuito e gerar a expressão simplificada para a saída do mesmo.
4. Um circuito EX-OR tem como expressão  $x = A\bar{B} + \bar{A}B$ . Sua saída será ALTA apenas quando as entradas *A* e *B* estiverem em níveis lógicos opostos.
5. Um circuito EX-NOR tem como expressão  $x = \bar{A}\bar{B} + AB$ . Sua saída será ALTA apenas quando as entradas *A* e *B* estiverem em um mesmo nível lógico.
6. Cada uma das portas básicas (AND, OR, NAND e NOR) pode ser usada para habilitar/desabilitar a passagem de um sinal de entrada para a sua saída.
7. As principais famílias de CIs são a TTL e a CMOS. Os CIs digitais disponíveis implementam uma grande variedade de funções lógicas, desde as mais simples (poucas portas por chip) até as mais complexas.

8. Para realizar a depuração básica de um circuito, você necessita, pelo menos, compreender o seu funcionamento, conhecer os tipos de falhas possíveis, um diagrama esquemático completo e uma ponta de prova lógica.
9. Um dispositivo lógico programável (PLD) é um CI que contém um grande número de portas lógicas cujas interconexões podem ser programadas pelo usuário para gerar a relação lógica desejada entre as entradas e as saídas.

**TERMOS IMPORTANTES**

- soma-de-produtos
- produto-de-somas
- mapa de Karnaugh (mapa K)
- agrupamento
- condição *don't care*
- exclusive-OR (EX-OR)
- exclusive-NOR (EX-NOR)
- geração e verificação de paridade
- habilitado/desabilitado
- Dual-In-Line Package (DIP)
- SSI, MSI, LSI, VLSI, ULSI, GSI
- Transistor-Transistor Logic (TTL)
- Complementary Metal Oxide Semiconductor (CMOS)
- indeterminado(a)(s)
- flutuação
- ponta de prova lógica
- dispositivo de lógica programável (PLD)

**PROBLEMAS**

**SEÇÕES 4-2 E 4-3**

- 4-1. Simplifique as expressões a seguir usando a álgebra booleana.
  - (a)  $x = ABC + \bar{A}C$
  - (b)  $y = (Q + R)(\bar{Q} + \bar{R})$
  - (c)  $w = ABC + A\bar{B}C + \bar{A}$
  - (d)  $q = \overline{RST}(\bar{R} + S + T)$
  - (e)  $x = \bar{A}\bar{B}\bar{C} + \bar{A}BC + ABC + A\bar{B}\bar{C} + A\bar{B}C$
  - (f)  $z = (B + \bar{C})(\bar{B} + C) + \bar{A} + B + \bar{C}$
  - (g)  $y = \overline{(C + D)} + \bar{A}C\bar{D} + A\bar{B}\bar{C} + \bar{A}\bar{B}CD + A\bar{C}\bar{D}$

- 4-2. Simplifique o circuito da Fig. 4-41 usando a álgebra booleana.

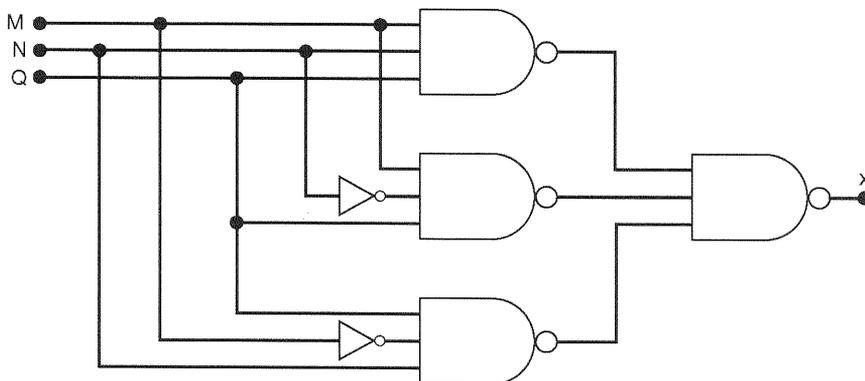


Fig. 4-41 Problemas 4-2 e 4-3.

4-3. Troque cada uma das portas do Problema 4-2 por uma porta NOR e simplifique o circuito usando a álgebra booleana.

SEÇÃO 4-4

D 4-4. Projete um circuito lógico que corresponde à tabela-verdade mostrada na Tabela 4-8.

TABELA 4-8

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

D 4-5. Projete um circuito lógico no qual a saída está em ALTO apenas quando a maioria das entradas A, B e C está em BAIXO.

D 4-6. Uma fábrica necessita de uma sirene para indicar o fim do expediente. Esta sirene deve ser tocada em uma das seguintes condições:  
 1. Já passa das cinco horas e todas as máquinas estão desligadas.  
 2. É sexta-feira, a produção do dia foi atingida e todas as máquinas estão desligadas.

Projete um circuito que controle a sirene. (Sugestão: Use quatro variáveis lógicas para representar as diversas condições. Por exemplo, a entrada A estará em ALTO somente quando já for mais de cinco horas.)

D 4-7. Um número binário de quatro bits é representado por  $A_3A_2A_1A_0$  onde  $A_3, A_2, A_1$  e  $A_0$  representam cada um dos bits, sendo  $A_0$  o LSB. Projete um circuito que produz uma saída em ALTO sempre que o número for maior do que 0010 e menor do que 1000.

D 4-8. A Fig. 4-42 mostra um diagrama de um circuito de alarme de automóvel usado para detectar algumas situações inde-

sejáveis. As três chaves são usadas para indicar, respectivamente, o estado da porta do motorista, da ignição e dos faróis. Projete um circuito que tenha como entrada essas três chaves e ative o alarme em uma das seguintes condições:

- Os faróis estão acesos e a ignição está desligada.
- A porta do motorista está aberta e a ignição está ligada.

4-9. Implemente o circuito do Problema 4-4 usando apenas portas NAND.

4-10. Implemente o circuito do Problema 4-5 usando apenas portas NAND.

4-11. Implemente a expressão  $z = \bar{D} + ABC + \bar{A}\bar{C}$ , utilizando portas AND, OR e INVERSORES. A partir do circuito obtido, construa um outro usando somente portas NAND.

SEÇÃO 4-5

4-12. Simplifique a expressão do Problema 4-1(e) usando o mapa K.

4-13. Simplifique a expressão do Problema 4-1(g) usando o mapa K.

4-14. Simplifique a expressão do Problema 4-7 usando o mapa K.

4-15. Determine a expressão mínima para cada mapa K da Fig. 4-43. Para o mapa do item (a), preste atenção especialmente no passo 5.

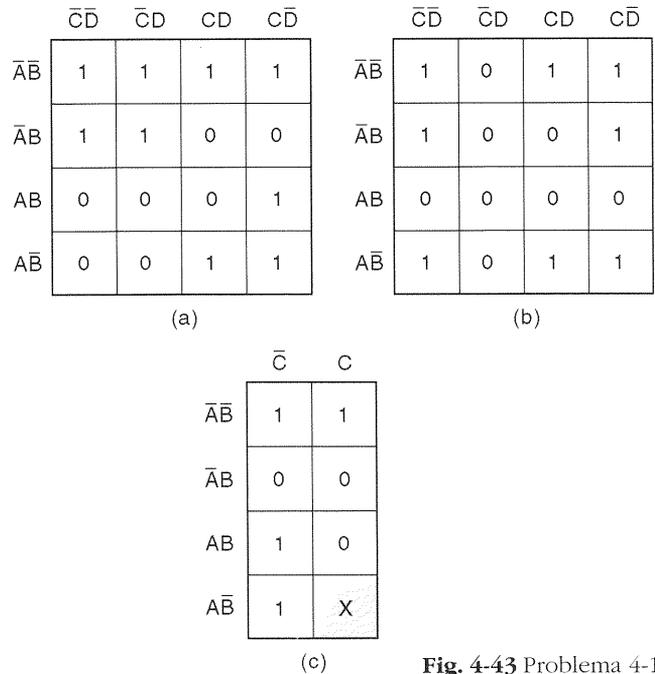


Fig. 4-43 Problema 4-15.

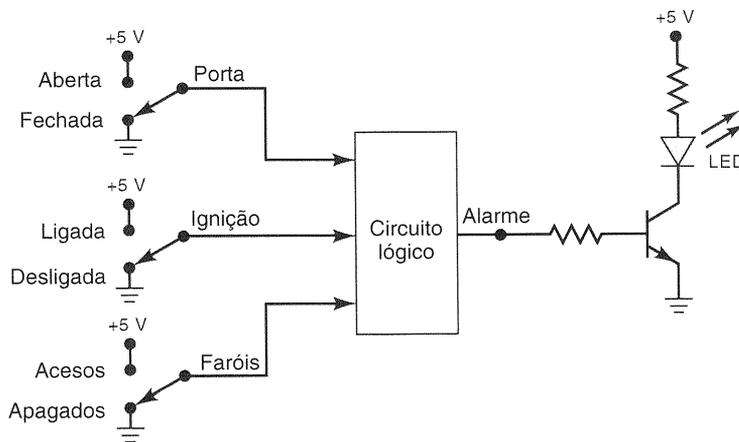


Fig. 4-42 Problema 4-8.

D

4-16. A Fig. 4-44 mostra um *contador BCD* que produz uma saída de quatro bits, que representa, em código BCD, o número de pulsos que foram aplicados à entrada do contador. Por exemplo, após quatro pulsos, a saída do contador é  $DCBA = 0100_2 = 4_{10}$ . O contador retorna a 0000 no décimo pulso e inicia a contagem novamente. Assim, as saídas  $DCBA$  nunca representarão um número maior que  $1001_2 = 9_{10}$ . Projete um circuito que produza um nível ALTO quando a contagem for igual a 2, 3 ou 9. Use o mapa de Karnaugh e aproveite as condições *don't care*.

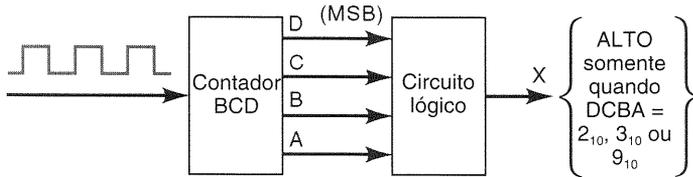


Fig. 4-44 Problema 4-16.

D

4-17. A Fig. 4-45 mostra quatro chaves que são parte de um circuito de controle de uma máquina copiadora. As chaves estão localizadas ao longo do caminho que o papel passa pela máquina. Cada uma das chaves está normalmente aberta, e quando o papel passa pela chave, ela é fechada. É impossível que as chaves SW1 e SW4 estejam fechadas ao mesmo tempo. Projete um circuito que produza uma saída em ALTO quando duas ou mais chaves estiverem fechadas ao mesmo tempo. Use o mapa de Karnaugh e aproveite as condições *don't care*.

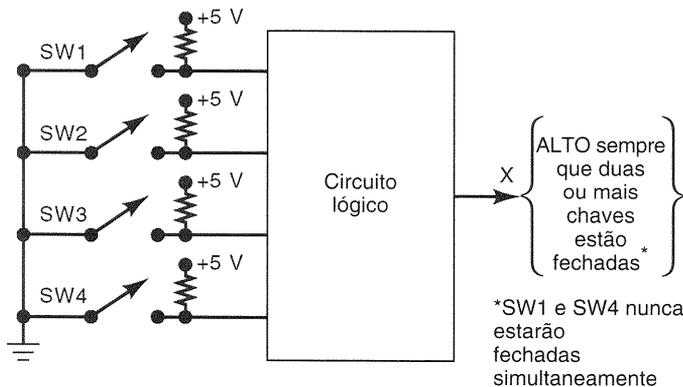


Fig. 4-45-Problema 4-17.

SEÇÃO 4-6

4-18. (a) Determine a forma de onda da saída do circuito da Fig. 4-46.

- (b) Repita o item (a) para o caso em que B está permanentemente em BAIXO.
- (c) Repita o item (a) para o caso em que B está permanentemente em ALTO.

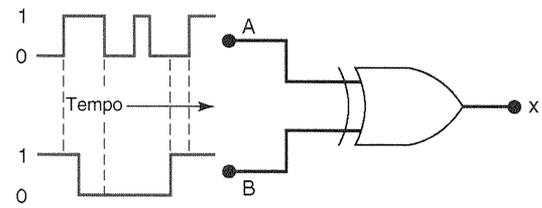


Fig. 4-46 Problema 4-18.

4-19. Determine as condições de entrada necessárias para fazer  $x = 1$  na Fig. 4-47.

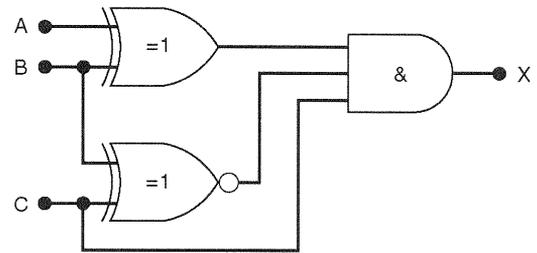


Fig. 4-47 Problema 4-19.

C, D

4-20. Projete um circuito com três entradas que produza uma saída em ALTO apenas quando todas as entradas estiverem em um mesmo nível. Use apenas portas EX-OR e uma outra porta de outro tipo.

C

4-21. A Fig. 4-48 representa um *detector de magnitude relativa*. Este circuito recebe dois números binários de três bits  $x_2x_1x_0$  e  $y_2y_1y_0$  e determina se os números são iguais, e, se não forem, indica qual é o maior. As três saídas estão definidas a seguir:

1.  $M = 1$  apenas se os dois números forem iguais.
2.  $N = 1$  apenas se  $x_2x_1x_0$  for maior que  $y_2y_1y_0$ .
3.  $P = 1$  apenas se  $y_2y_1y_0$  for maior que  $x_2x_1x_0$ .

Projete o circuito para esse detector. Ele possui seis entradas e três saídas e portanto é complexo demais para usar uma tabela-verdade. A título de sugestão, estude o Exemplo 4-16. Ele pode lhe dar uma dica de como resolver este problema.

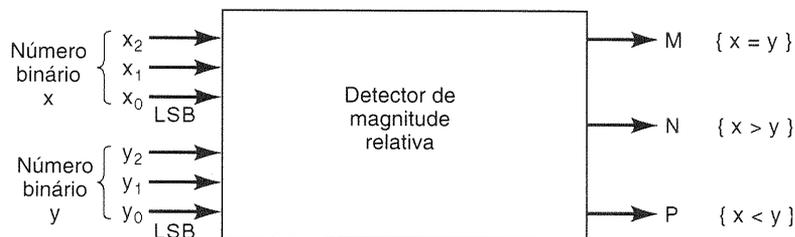


Fig. 4-48 Problema 4-21.

## MAIS PROBLEMAS DE PROJETO

C, D

4-22. A Fig. 4-49 apresenta um circuito multiplicador que recebe dois números binários de dois bits  $x_1x_0$  e  $y_1y_0$  e produz como saída um número binário  $z_3z_2z_1z_0$  que é igual ao produto aritmético dos números de entrada. Projete um circuito para o multiplicador. (Sugestão: O circuito possui quatro entradas e quatro saídas.)

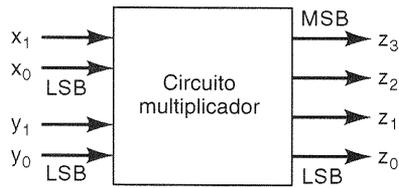


Fig. 4-49 Problema 4-22.

D

4-23. Um código BCD está sendo transmitido a um receptor remoto. Os bits são  $A_3A_2A_1A_0$ , onde  $A_3$  é o MSB. Dentre os circuitos existentes no receptor existe um denominado *detector de erros BCD*. Este circuito verifica se o código recebido é um código BCD válido (isto é,  $\leq 1001$ ). Projete este circuito de modo a produzir um nível ALTO em uma situação de erro.

D

4-24. Projete um circuito cuja saída esteja em ALTO sempre que  $A$  e  $B$  estiverem em ALTO e enquanto  $C$  e  $D$  estiverem ambos em nível ALTO, ou ambos em BAIXO. Tente fazer este exercício sem utilizar uma tabela-verdade. Verifique o resultado construindo uma tabela-verdade para o seu circuito, e observe se ele obedece ao enunciado do problema.

D

4-25. Quatro grandes tanques em uma indústria química contêm diferentes líquidos que estão sendo aquecidos. Sensores de nível de líquido são utilizados para detectar se o nível do tanque  $A$  ou do tanque  $B$  sobe acima de um nível predeterminado. Sensores de temperatura existentes nos tanques  $C$  e  $D$  detectam se a temperatura de um desses tanques cai abaixo de um determinado limite. Suponha que as saídas dos sensores de nível de líquido  $A$  e  $B$  estarão em BAIXO quando o nível for satisfatório e estarão em ALTO quando o nível for muito alto. Além disso, as saídas dos sensores de temperatura  $C$  e  $D$  estarão em BAIXO quando a temperatura for satisfatória e estarão em ALTO quando a temperatura for muito BAIXA. Projete um circuito que detecte quando o nível no tanque  $A$  ou no  $B$  estiver muito alto, ao mesmo tempo em que a temperatura em um dos tanques  $C$  ou  $D$  estiver muito baixa.

C, D

4-26. A Fig. 4-50 mostra o cruzamento de uma rodovia com uma via de acesso. Sensores detectores de veículos são colocados ao longo das pistas  $C$  e  $D$  da rodovia e das pistas  $A$  e  $B$  da via de acesso. A saída desse tipo de sensor está em BAIXO quando não existe nenhum carro presente e está em ALTO quando um veículo está presente. Um sinal de trânsito colocado no cruzamento deve funcionar de acordo com a seguinte lógica:

1. O sinal da direção leste-oeste (L-O) deve estar verde quando *ambas* as pistas  $C$  e  $D$  estiverem ocupadas.
2. O sinal da direção (L-O) deve estar verde quando *ou*  $C$  ou  $D$  estiverem ocupadas mas *ambas* as pistas  $A$  e  $B$  não estiverem.
3. O sinal da direção norte-sul (N-S) deve estar verde quando *ambas* as pistas  $A$  e  $B$  estiverem ocupadas mas *ambas* as pistas  $C$  e  $D$  não estiverem.

4. O sinal da direção (N-S) deve estar verde quando *ou*  $A$  ou  $B$  estiverem ocupadas e enquanto *ambas* as pistas  $C$  e  $D$  estiverem vazias.
5. O sinal da direção (L-O) deve estar verde quando *não* houver nenhum veículo presente.

Utilizando as saídas dos sensores  $A$ ,  $B$ ,  $C$  e  $D$  como entradas, projete um circuito que controle esse sinal de trânsito. Devem existir duas saídas, N-S e L-O, que devem ir para ALTO quando o sinal correspondente tiver que estar *verde*. Simplifique esse circuito ao máximo, e mostre todas as etapas de simplificação.

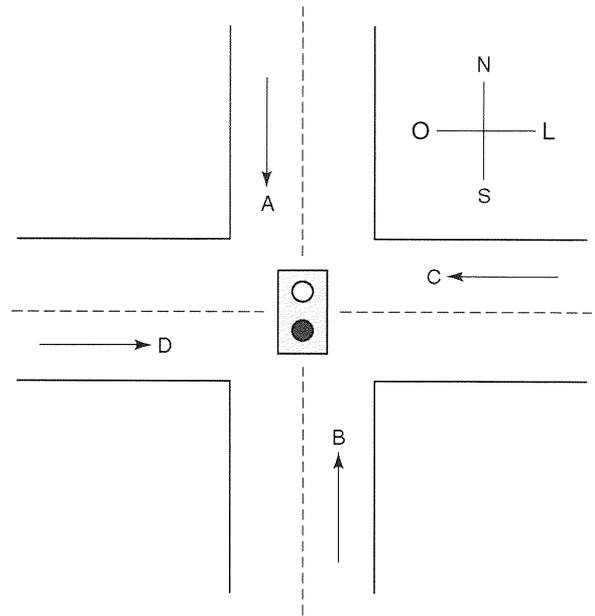


Fig. 4-50 Problema 4-26.

## SEÇÃO 4-7

D

4-27. Projete novamente o verificador e gerador de paridade da Fig. 4-24 para operar com paridade ímpar. (Sugestão: Qual é a relação entre o bit de paridade, quando usamos paridade ímpar, e aquele que é usado quando usamos paridade par para um mesmo conjunto de bits de dados?)

D

4-28. Projete novamente o verificador e gerador de paridade da Fig. 4-24 para operar com oito bits de dados.

## SEÇÃO 4-8

D

4-29. Projete um circuito que permitirá que o sinal de entrada  $A$  chegue até a saída somente quando a entrada de controle  $B$  estiver em BAIXO e enquanto a outra entrada de controle  $C$  estiver em ALTO. Caso isso não ocorra, a saída deve estar em BAIXO.

D

4-30. Projete um circuito que *desabilita* a passagem do sinal de entrada somente quando as entradas de controle  $B$ ,  $C$  e  $D$  estiverem em ALTO. Quando a passagem estiver desabilitada, a saída deve estar em ALTO.

D

4-31. Projete um circuito que controla a passagem de um sinal  $A$  de acordo com os seguintes requisitos:

1. A saída  $X$  deve ser igual a  $A$  quando as entradas de controle  $B$  e  $C$  estiverem em um mesmo nível.
2.  $X$  deve permanecer em ALTO quando  $B$  e  $C$  estiverem em níveis diferentes.

D

4-32. Projete um circuito que possui dois sinais de entrada  $A_1$  e  $A_0$ , uma entrada de controle  $S$ , e funciona conforme os requisitos mostrados na Fig. 4-51. Esse tipo de circuito é chamado de *multiplexador* (e será estudado no Cap. 9).

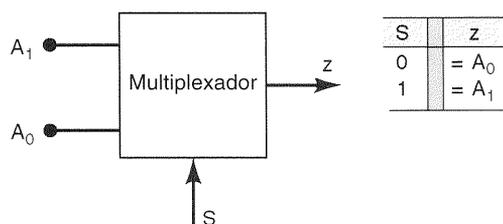


Fig. 4-51 Problema 4-32.

D

4-33. Use o mapa de Karnaugh para projetar um circuito que obedeça aos requisitos do Exemplo 4-16. Compare o circuito obtido com aquele que está na Fig. 4-22. Este exercício mostra que o mapa de Karnaugh não pode aproveitar-se das portas lógicas EX-OR e EX-NOR. O projetista deve ser capaz de determinar quando estas portas são aplicáveis.

## SEÇÕES 4-9 A 4-13

T\*

- 4-34. (a) Um técnico está testando um circuito lógico e verifica que a saída de um determinado INVERSOR está sempre em BAIXO, mesmo que sua entrada esteja pulsando. Enumere as possíveis razões deste mau funcionamento.
- (b) Repita o item (a) para o caso em que a saída do INVERSOR está sempre em um nível indeterminado.

T

- 4-35. Os sinais mostrados na Fig. 4-52 são aplicados às entradas do circuito da Fig. 4-31. Suponha que existe um circuito aberto interno em Z1-4.
- (a) O que a ponta de prova lógica indicaria em Z1-4?
  - (b) Qual seria a tensão contínua que você esperaria ler em um voltímetro colocado em Z1-4? (Lembre-se que todos os CIs são TTL.)
  - (c) Faça um esboço de como seriam os sinais  $\overline{CLOCKOUT}$  e  $\overline{SHIFTOUT}$ .
  - (d) Em vez de um circuito aberto, suponha que os pinos 9 e 10 de Z2 estejam em curto internamente. Desenhe como seriam os sinais em Z2-10,  $\overline{CLOCKOUT}$  e  $\overline{SHIFTOUT}$ .

T

4-36. Suponha que os CIs da Fig. 4-31 são CMOS. Descreva como a operação deste circuito seria afetada se houvesse um circuito aberto na ligação entre Z2-2 e Z2-10.

T

4-37. No Exemplo 4-27, relacionamos três possíveis falhas para a situação mostrada na Fig. 4-34. Que procedimento você utilizaria para determinar qual das possíveis falhas está realmente causando o problema?

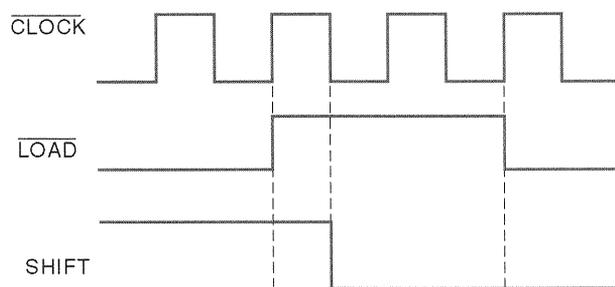


Fig. 4-52 Problema 4-35.

T

4-38. Veja o circuito da Fig. 4-36. Suponha que esses dispositivos são todos CMOS. Além disso, suponha que a indicação da ponta de prova lógica em Z2-3 é "indeterminado", em vez de pulsante. Relacione as possíveis falhas e escreva um procedimento que o ajude a determinar qual dessas falhas está provocando o problema.

T

4-39. Veja o circuito da Fig. 4-36. Recorde que a saída  $Y$  deve estar em ALTO para qualquer uma das seguintes condições:

1.  $A = 1, B = 0$ , independentemente de  $C$ .
2.  $A = 0, B = 1$  e  $C = 1$ .

Durante o teste do circuito, o técnico observa que  $Y$  vai para ALTO somente para a primeira condição, mas permanece em BAIXO para todas as outras. Considere a seguinte lista de possíveis falhas. Para cada uma delas, escreva "sim" ou "não" para indicar se essa falha pode ou não ser a causa do problema. Explique o seu raciocínio para cada item marcado com um "não".

- (a) Um curto interno entre terra e Z2-13.
- (b) Um circuito aberto na ligação para Z2-13.
- (c) Um curto interno entre  $V_{cc}$  e Z2-11.
- (d) Um circuito aberto na ligação de  $V_{cc}$  para Z2.
- (e) Um circuito aberto interno em Z2-9.
- (f) Um circuito aberto na ligação entre Z2-11 e Z2-9.
- (g) Uma ponte de solda entre os pinos 6 e 7 de Z2.

T

4-40. Desenvolva um procedimento para isolar a falha que está causando o mau funcionamento descrito no Problema 4-39.

T

4-41. Suponha que as portas lógicas da Fig. 4-39 são todas CMOS. Quando um técnico testa o circuito, ele percebe que o mesmo opera corretamente, exceto nas seguintes condições:

1.  $A = 1, B = 0, C = 0$ .
2.  $A = 0, B = 1, C = 1$ .

Para essas condições, a ponta de prova indica níveis indeterminados em Z2-6, Z2-11 e Z2-8. Você seria capaz de dizer qual a causa provável do mau funcionamento? Explique o seu raciocínio.

T

4-42. A Fig. 4-53 é um circuito lógico combinacional que ativa o alarme do carro sempre que os assentos do motorista e/ou do passageiro estão ocupados mas o cinto de segurança não está colocado quando o carro é ligado. Os sinais  $DRIV$  e  $PASS$  são ativos em ALTO e indicam a presença do motorista e do passageiro, respectivamente. Estes sinais são fornecidos por chaves ativadas por pressão colocadas nos assentos. O sinal  $IGN$  é ativo em ALTO quando a chave de ignição está ligada. O sinal  $BELTD$  é ativo em BAIXO e indica que o cinto de segurança do motorista *não* está colocado. O sinal  $BELTP$  é o sinal referente ao cinto do passageiro. O alarme será ativado (BAIXO) sempre que o carro for ligado e um dos bancos dianteiros estiver ocupado e seu cinto não estiver colocado.

\*Lembre-se de que T significa que o exercício é de depuração [do inglês troubleshooting].

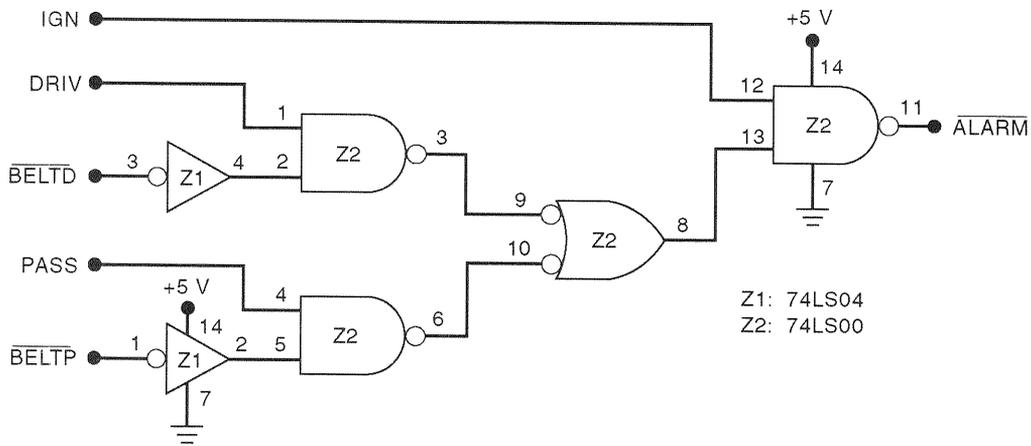


Fig. 4-53 Problemas 4-42, 4-43 e 4-44.

- (a) Verifique que o circuito funciona conforme a descrição dada.  
 (b) Descreva como esse sistema de alarme iria operar se Z1-2 estivesse internamente em curto com a terra.  
 (c) Descreva como esse circuito iria operar se existisse um circuito aberto na ligação entre Z2-6 e Z2-10.

T

4-43. Suponha que o sistema da Fig. 4-53 está funcionando de tal modo que o alarme é ativado assim que o motorista ou o passageiro estejam sentados e o carro seja ligado, sem levar em conta se os cintos estão colocados ou não. Quais são as possíveis falhas que podem estar ocorrendo? Como você faria para determinar a falha que está causando esse problema?

T

4-44. Suponha que o sistema da Fig. 4-53 está funcionando de tal modo que o alarme é ativado tão logo a ignição seja ligada, não importando o estado das outras entradas. Relacione as possíveis falhas e escreva um procedimento para isolar a falha que está causando o problema.

## SEÇÃO 4-14

C

- 4-45. (a) Modifique a estrutura do PLD da Fig. 4-40 de modo que ela possa receber três entradas.  
 (b) Usando esse PLD de três entradas, mostre como implementar o circuito do Exemplo 4-7. Observe que não é necessário simplificar a expressão lógica da saída para fazer isso.  
 (c) O circuito do Exemplo 4-7 foi implementado usando um CI NAND quádruplo [Fig. 4-9(a)]. Compare o número de conexões *externas* entre CIs dessa implementação com a outra que utiliza um PLD.

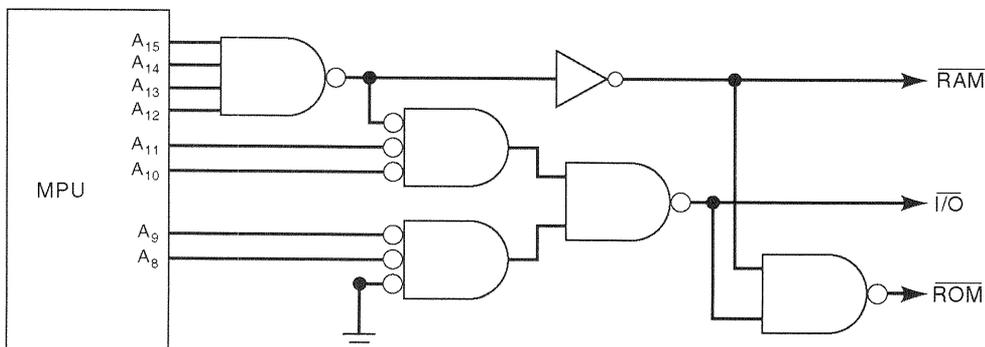


Fig. 4-54

## QUESTÃO DE FIXAÇÃO

4-46. Defina cada um dos seguintes termos.

- (a) Mapa de Karnaugh  
 (b) Forma de soma-de-produtos  
 (c) Gerador de paridade  
 (d) Octeto  
 (e) Circuito habilitador  
 (f) Estado *don't care*  
 (g) Entrada em flutuação  
 (h) Nível de tensão indeterminado

## APLICAÇÕES EM MICROCOMPUTADORES

C

4-47. Em um microcomputador, a unidade microprocessadora (MPU — *MicroProcessor Unit*) está sempre se comunicando com um dos seguintes dispositivos: (1) memória de acesso aleatório (RAM — *random access memory*), que armazena programas e dados que podem ser prontamente modificados; (2) memória apenas de leitura (ROM — *read only memory*), que armazena programas e dados que nunca são modificados; (3) dispositivos externos de entrada e saída (E/S), tais como: teclados, monitores de vídeo, impressoras e unidades de disco. Quando está executando um programa, a MPU gera o endereço que seleciona o tipo de dispositivo (RAM, ROM ou E/S) com o qual ela quer se comunicar. A Fig. 4-54 mostra um esquema típico em que a MPU gera oito bits de endereço, de  $A_5$  até  $A_8$ . Na verdade, a MPU gera um endereço de dezesseis bits; entretanto, os bits de ordem mais baixa de  $A_7$  até  $A_0$  não são utilizados nos processos de seleção do dispositivo. O endereço é fornecido como entrada de um circuito que, então, gera os seguintes sinais de seleção de dispositivos:  $\overline{RAM}$ ,  $\overline{ROM}$  e  $\overline{E/S}$ .

Analise o circuito e responda:

(a) A faixa de endereços de  $A_{15}$  até  $A_8$  que irá ativar o sinal  $\overline{RAM}$ .

(b) A faixa de endereços que irá ativar o sinal  $\overline{E/S}$ .

(c) A faixa de endereços que irá ativar o sinal  $\overline{ROM}$ .

Escreva os endereços em binário e em hexadecimal. Por exemplo, a resposta do item (a) é  $A_{15} - A_8$ :  $00000000_2$  até  $11101111_2 = 00_{16}$  até  $EF_{16}$ .

**C, D**

**4-48.** Em alguns microcomputadores, a MPU pode ser *desabilitada* por curtos períodos de tempo, enquanto um outro dispositivo controla a RAM, ROM e E/S. Durante esse intervalo, um sinal especial de controle,  $\overline{DMA}$ , é ativado pela MPU, e é usado para desabilitar (desativar) a lógica de seleção de dispositivo, de modo que  $\overline{RAM}$ ,  $\overline{ROM}$  e  $\overline{E/S}$  estão todos em seus estados inativos. Modifique o circuito da Fig. 4-54 para que  $\overline{RAM}$ ,  $\overline{ROM}$  e  $\overline{E/S}$  sejam desativados sempre que  $\overline{DMA}$  estiver ativo, independentemente do endereço.

## RESPOSTAS PARA AS QUESTÕES DE REVISÃO DAS SEÇÕES

### SEÇÃO 4-1

1. Somente (a)
2. Somente (c)

### SEÇÃO 4-3

1. A expressão (b) não está na forma de soma-de-produtos, porque o sinal de inversão está sobre  $C$  e  $D$  (como por exemplo no termo  $\overline{ACD}$ ). A expressão (c) não está na forma de soma-de-produtos, por causa do termo  $(M + \overline{N})P$

3.  $x = \overline{A} + \overline{B} + \overline{C}$

### SEÇÃO 4-4

1.  $x = \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}B\overline{C}\overline{D}$
2. Oito

### SEÇÃO 4-5

1.  $x = AB + AC + BC$
2.  $x = A + BCD$
4. Uma condição de entrada para qual não existe uma condição de saída especificada

### SEÇÃO 4-6

2. A saída está permanentemente em BAIXO
3. Não. A porta EX-OR disponível pode ser usada como um INVERSOR se conectarmos uma das entradas a um nível ALTO constante (veja o Exemplo 4-15)

### SEÇÃO 4-8

1.  $x = \overline{A(B \oplus C)}$
2. OR, NAND
3. NAND, NOR

### SEÇÃO 4-9

1. DIP
2. SSI, MSI, LSI, VLSI, ULSI e GSI
3. Verdadeiro
4. Verdadeiro
5. famílias 40, 74AC, 74ACT
6. 0 a 0,8 V; 2,0 a 5,0 V
7. 0 a 1,5 V; 3,5 a 5,0 V
8. Como se as entradas estivessem em ALTO
9. Imprevisível; pode superaquecer e ser destruído
10. 74HCT e 74ACT

### SEÇÃO 4-11

1. Entradas ou saídas em aberto; entradas ou saídas em curto com  $V_{cc}$ ; entradas ou saídas em curto com a terra; pinos em curto entre si; circuito internamente danificado
2. Pinos em curto entre si
3. Para TTL, nível BAIXO; para CMOS, indeterminado

### SEÇÃO 4-12

1. Circuito aberto em linhas de sinal; linhas de sinal em curto; fonte de alimentação com defeito; carregamento da saída
2. Fios partidos; soldas malfeitas; fissuras ou cortes na placa de circuito impresso; pinos de CI tortos ou amassados; soquetes defeituosos
3. CIs funcionando de modo incorreto ou simplesmente não funcionando
4. Nível lógico indeterminado

### SEÇÃO 4-14

1.  $x = B$
2.  $x = 1$
3. Quatro INVERSORES, 16 portas AND, 16 elos e uma OR de 16 entradas