Resumos ISD - Descodificadores, Codifcadores, Multiplexers. Este resumo contem a matéria do livro o Resumo dos ppt que está na sebenta e exercícios e as soluções dos impares (estava assim no livro)

Esta é a parte I, a II tem mais exercícios

Hugo Leal

6-5 DECODIFICADORES

Um **decodificador** é um circuito digital que detecta a presença de uma combinação específica de bits (código) em suas entradas indicando a presença desse código através de um nível de saída especificado. Em sua forma geral, um decodificador tem n linhas de entrada para manipular n bits e de uma a 2^n linhas de saída para indicar a presença de uma ou mais combinações de n bits. Nessa seção, diversos decodificadores são apresentados. Os princípios básicos podem ser estendidos para outros tipos de decodificadores.

Ao final do estudo desta seção você deverá ser capaz de:

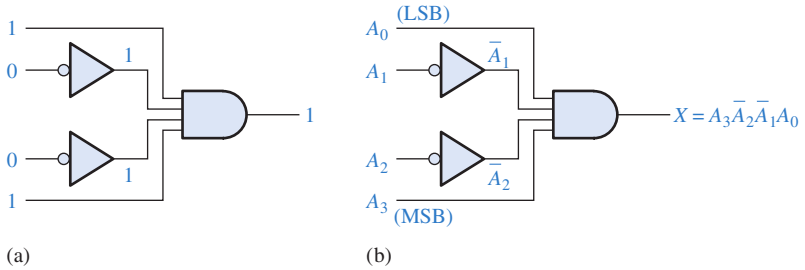
■ Definir *decodificador* ■ Projetar um circuito lógico para decodificar qualquer combinação de bits ■ Descrever o decodificador de binário para BCD (74HC154) ■ Expandir decodificadores para operar códigos com um grande número de bits ■ Descrever o decodificador de BCD para 7 segmentos (74LS47) ■ Discutir a supressão de zeros em displays de 7 segmentos ■ Usar decodificadores em aplicações específicas

NOTA: COMPUTAÇÃO

Uma *instrução* "diz" ao computador qual operação realizar. As instruções são códigos de máquina (1s e 0s) e, para o computador executar uma instrução, a instrução tem que ser decodificada. A decodificação de instrução é um dos passos da instrução *pipelining*, que são os seguintes: A instrução é lida na memória (busca da instrução), a instrução é decodificada, o(s) operando(s) é(são) lido(s) na memória (busca do operando), a instrução é executada e o resultado é escrito de volta na memória. Basicamente, o pipelining permite que o processamento da próxima instrução seja iniciado antes que o da atual seja completado.

Decodificador Binário Básico

Suponha que precisamos determinar quando um binário 1001 ocorre nas entradas de um circuito digital. Uma porta AND pode ser usada como o elemento de decodificação básico porque ela produz um nível ALTO na saída apenas quando todas as suas entradas estão em nível ALTO. Portanto, temos que ter certeza que todas as entradas da porta AND são nível ALTO quando ocorrer o número binário 1001; isso pode ser feito invertendo os dois bits do meio (os 0s), conforme mostra a Figura 6–26.



▲ **FIGURA 6–26**

Lógica de decodificação para o código binário 1001 com uma saída ativa em nível ALTO.

A equação lógica para o decodificador visto na Figura 6–26(a) é desenvolvida como ilustra a Figura 6–26(b). Devemos verificar que a saída é 0 exceto quando $A_0 = 1, A_1 = 0, A_2 = 0$ e $A_3 = 1$ forem aplicados nas entradas. A_0 é o LSB e A_3 é o MSB. *Na representação de um número binário ou outro código ponderado nesse livro, o LSB é o bit mais à direita numa representação horizontal e o bit mais alto numa representação vertical, a menos que seja especificado algo em contrário.*

Se uma porta NAND for usada no lugar da porta AND no circuito da Figura 6–26, uma saída de nível BAIXO indicará a presença do código binário próprio, que neste caso é 1001.

EXEMPLO 6–8

Determine a lógica necessária para decodificar o número binário 1011 produzindo um nível ALTO na saída.

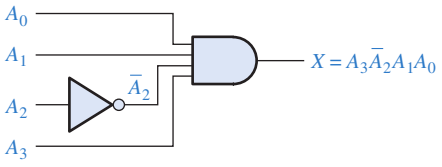
Solução A função de decodificação pode ser obtida complementando apenas as variáveis que aparecem como 0 no número binário desejado, como a seguir:

$$X = A_3\overline{A}_2A_1A_0 \qquad (1011)$$

Essa função pode ser implementada conectando as variáveis verdadeiras (não-complementadas) A_0, A_1 e A_3 diretamente nas entradas de uma porta AND e invertendo a variável A_2 antes de aplicá-la na entrada da porta AND. A lógica de decodificação é mostrada na Figura 6–27.

▶ FIGURA 6–27

Lógica de decodificação para gerar uma saída de nível ALTO quando 1011 estiver nas entradas.



Problema relacionado Desenvolva a lógica necessária para detectar o código binário 10010 e produzir uma saída ativa em nível BAIXO.

Decodificador de 4 Bits

Para decodificar todas as combinações possíveis de quatro bits, são necessárias dezesseis portas de decodificação ($2^4 = 16$). Esse tipo de decodificador é normalmente denominado *decodificador de 4 linhas para 16 linhas* porque existem quatro entradas e dezesseis saídas ou *decodificador 1 de 16* porque para um dado código nas entradas, uma das dezesseis saídas é ativada. A Tabela 6–4 mostra uma lista de dezesseis códigos binários e suas correspondentes funções de decodificação.

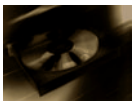
Se uma saída ativa em nível BAIXO for necessária para cada número decodificado, o decodificador completo pode ser implementado com portas NAND e inversores. Para decodificar cada um dos dezesseis códigos binários, são necessárias dezesseis portas NAND (portas AND podem ser usadas para produzir saídas ativas em nível ALTO).

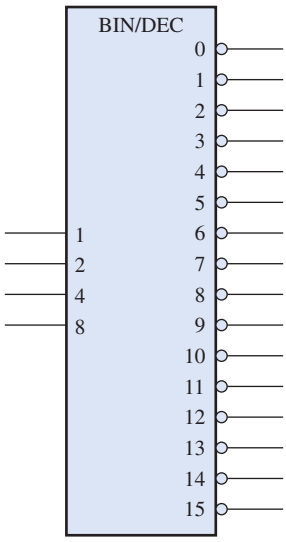
▼ TABELA 6–4

Funções de decodificação e tabela-verdade para um decodificador de 4 linhas para 16 linhas (1 de 16) com saídas ativas em nível BAIXO

DÍGITO DECIMAL	ENTRADAS BINÁRIAS				FUNÇÃO DE DECODIFICAÇÃO	SAÍDAS														
	A_3	A_2	A_1	A_0		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	$\overline{A}_3\overline{A}_2\overline{A}_1\overline{A}_0$	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	$\overline{A}_3\overline{A}_2\overline{A}_1A_0$	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	1	0	$\overline{A}_3\overline{A}_2A_1\overline{A}_0$	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
3	0	0	1	1	$\overline{A}_3\overline{A}_2A_1A_0$	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
4	0	1	0	0	$\overline{A}_3A_2\overline{A}_1\overline{A}_0$	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
5	0	1	0	1	$\overline{A}_3A_2\overline{A}_1A_0$	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
6	0	1	1	0	$\overline{A}_3A_2A_1\overline{A}_0$	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
7	0	1	1	1	$\overline{A}_3A_2A_1A_0$	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
8	1	0	0	0	$A_3\overline{A}_2\overline{A}_1\overline{A}_0$	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
9	1	0	0	1	$A_3\overline{A}_2\overline{A}_1A_0$	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
10	1	0	1	0	$A_3\overline{A}_2A_1\overline{A}_0$	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
11	1	0	1	1	$A_3\overline{A}_2A_1A_0$	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
12	1	1	0	0	$A_3A_2\overline{A}_1\overline{A}_0$	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
13	1	1	0	1	$A_3A_2\overline{A}_1A_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
14	1	1	1	0	$A_3A_2A_1\overline{A}_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
15	1	1	1	1	$A_3A_2A_1A_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Um símbolo lógico para um decodificador de 4 linhas para 16 linhas (1 de 16) com saídas ativas em nível BAIXO é mostrado na Figura 6–28. A denominação BIN/DEC indica que uma entrada binária ativa a correspondente saída decimal. As denominações de entrada 8, 4, 2 e 1 representam os pesos binários dos bits de entrada ($2^3 2^2 2^1 2^0$).

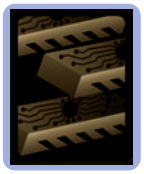




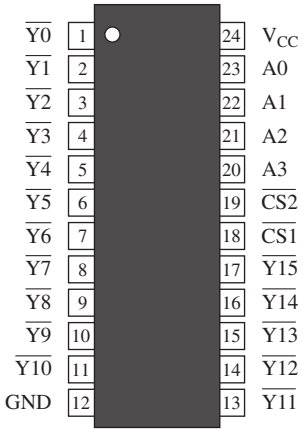
▶ FIGURA 6–28

Funções de decodificação e tabela-verdade para um decodificador de 4 linhas para 16 linhas (1 de 16) com saídas ativas em nível BAIXO.

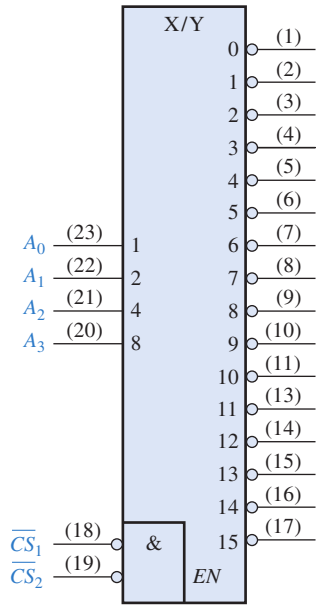
UM DECODIFICADOR 1 DE 16 (74HC154)



O CI 74HC154 é um bom exemplo de um decodificador. O símbolo lógico é mostrado na Figura 6–29. Existe uma função de habilitação (*EN*) fornecida nesse dispositivo, a qual é implementada com uma porta NOR usada com uma AND negativa. Um nível BAIXO em cada entrada de seleção de chip, $\overline{CS}_1$ e $\overline{CS}_2$, é necessário para tornar nível ALTO a saída da porta de habilitação (*EN*). A saída da porta de habilitação é conectada na entrada de cada porta NAND no decodificador, assim ela tem que ser nível ALTO para as portas NAND serem habilitadas. Se a porta de habilitação não for ativada por um nível BAIXO nas duas



(a) Diagrama de pinos



(b) Símbolo lógico

▶ FIGURA 6–29

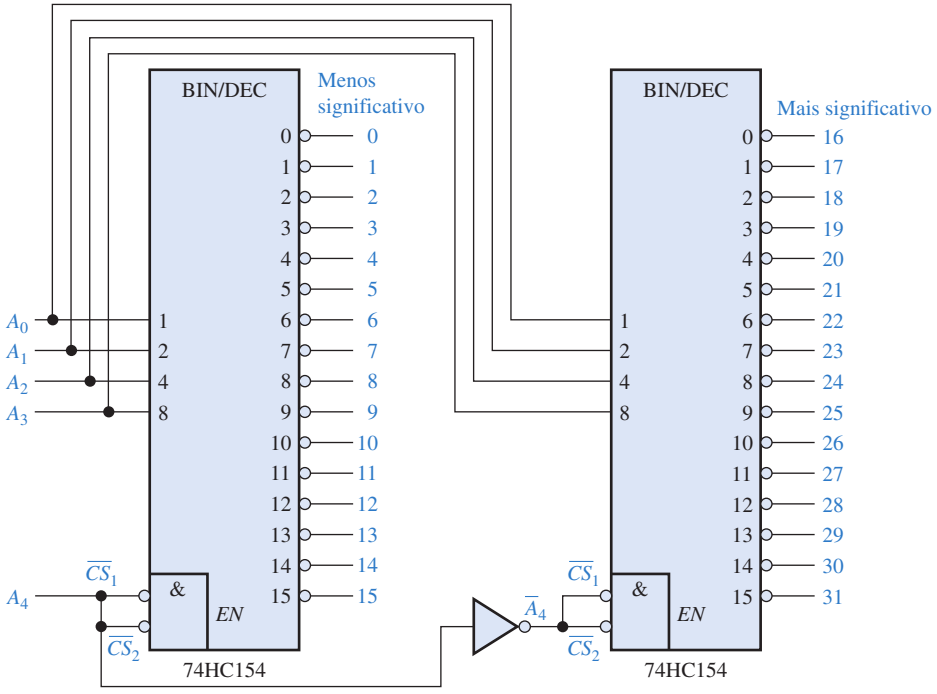
Diagrama de pino e símbolo lógico para o decodificador 1 de 16 (74HC154).

entradas, então todas as dezesseis saídas do decodificador (Y) estarão em nível ALTO independente dos estados das quatro variáveis de entrada (A_0, A_1, A_2 e A_3). Esse dispositivo pode ser comercializado em outras famílias CMOS ou TTL. Verifique o site da Texas Instruments (www.ti.com) ou o CD-ROM da Texas Instruments que acompanha esse livro.

EXEMPLO 6–9

Certa aplicação necessita que um número de 5 bits seja decodificado. Use CIs decodificadores para implementar a lógica. O número binário é representado pelo formato $A_4A_3A_2A_1A_0$.

Solução Como o CI 74HC154 pode operar apenas quatro bits, temos que usar dois decodificadores para decodificar 5 bits. O quinto bit (A_4) é conectado às entradas de seleção de chip $\overline{CS}_1$ e $\overline{CS}_2$ de um decodificador, e $\overline{A}_4$ é conectado às entradas $\overline{CS}_1$ e $\overline{CS}_2$ do outro decodificador, como mostra a Figura 6–30. Quando o número decimal for 15 ou menor, $A_4 = 0$, o decodificador menos significativo é habilitado e o decodificador mais significativo é desabilitado. Quando o número decimal for maior que 15, $A_4 = 1$ sendo $\overline{A}_4 = 0$, o decodificador mais significativo é habilitado e o decodificador menos significativo é desabilitado.



▶ **FIGURA 6–30**
Um decodificador de 5 bits usando CIs 74HC154.

Problema relacionado Determine a saída do circuito da Figura 6–30 que é ativada para a entrada binária 10110.

Uma Aplicação

Decodificadores são usados em muitos tipos de aplicações. Um exemplo é usado em computadores para seleção de entrada/saída conforme ilustrado no diagrama geral da Figura 6–31.

Os computadores têm que se comunicar com uma variedade de dispositivos externos denominados *periféricos* enviando e/ou recebendo dados através do que é conhecido como portas de entrada/saída (I/O). Esses dispositivos externos incluem impressoras, modems, scanners, acionadores de disco externos, teclados, monitores de vídeo e outros computadores. Conforme indicado na

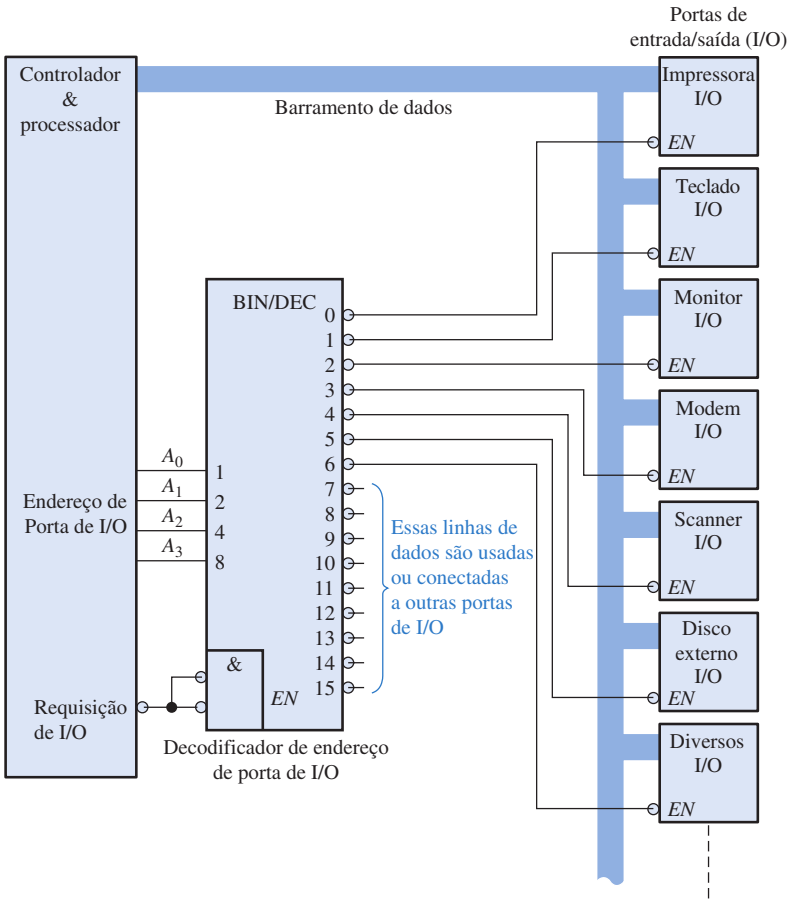


FIGURA 6–31

Um sistema simplificado de porta de I/O de computador com um decodificador de endereço de porta de I/O com apenas quatro linhas de endereço sendo mostradas.

Figura 6–31, um decodificador é usado para selecionar a porta de I/O conforme determinado pelo computador de forma que os dados podem ser enviados ou recebidos de um dispositivo externo específico.

Cada porta de I/O tem um número, denominado endereço, o qual o identifica unicamente. Quando o computador "quer" se comunicar com um dispositivo em particular, ele emite o código de endereço apropriado para a porta de I/O na qual o dispositivo em particular está conectado. Esse endereço binário de porta de I/O é decodificado e a saída apropriada do decodificador é ativada para habilitar a porta de I/O.

Conforme mostra a Figura 6–31, os dados binários são transferidos internamente ao computador através do barramento de dados, o qual é constituído de um conjunto de linhas paralelas. Por exemplo, um barramento de 8 bits consiste de oito linhas em paralelo que transportam um byte de dados de cada vez. O barramento de dados alcança todos as portas de I/O, porém todos os dados que entram ou saem passam através da porta de I/O que está habilitada pelo decodificador de endereço de porta de I/O.

Decodificador de BCD para Decimal

O decodificador de BCD para decimal converte cada código BCD (código 8421) em uma das dez indicações decimais possíveis. Ele é freqüentemente referido como um *decodificador de 4 linhas para 10 linhas* ou um *decodificador 1 de 10*.

O método de implementação é o mesmo que para o decodificador 1 de 16 discutido anteriormente, exceto que são necessárias apenas dez portas de decodificação porque o código BCD representa apenas os dígitos decimais de 0 a 9. A Tabela 6–5 apresenta uma lista de dez códigos BCD e suas correspondentes funções de decodificação. Cada uma dessas funções de decodificação é implementada com portas NAND para prover saídas ativas em nível BAIXO. Se for necessário uma saída ativa em nível ALTO, são usadas portas AND para decodificação. A lógica de decodificação é idêntica às dez primeiras portas de decodificação do decodificador 1 de 16 (veja a Tabela 6–4).

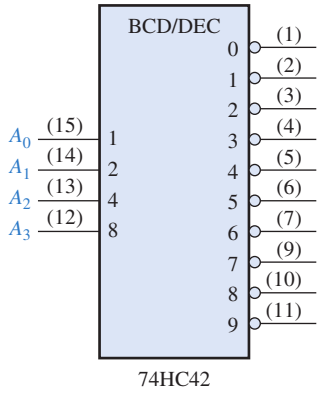
◀ TABELA 6–5

Funções de decodificação BCD

DÍGITO DECIMAL	CÓDIGO BCD A_3	A_2	A_1	A_0	FUNÇÃO DE DECODIFICAÇÃO
0	0	0	0	0	$\overline{A}_3\overline{A}_2\overline{A}_1\overline{A}_0$
1	0	0	0	1	$\overline{A}_3\overline{A}_2\overline{A}_1A_0$
2	0	0	1	0	$\overline{A}_3\overline{A}_2A_1\overline{A}_0$
3	0	0	1	1	$\overline{A}_3\overline{A}_2A_1A_0$
4	0	1	0	0	$\overline{A}_3A_2\overline{A}_1\overline{A}_0$
5	0	1	0	1	$\overline{A}_3A_2\overline{A}_1A_0$
6	0	1	1	0	$\overline{A}_3A_2A_1\overline{A}_0$
7	0	1	1	1	$\overline{A}_3A_2A_1A_0$
8	1	0	0	0	$A_3\overline{A}_2\overline{A}_1\overline{A}_0$
9	1	0	0	1	$A_3\overline{A}_2\overline{A}_1A_0$

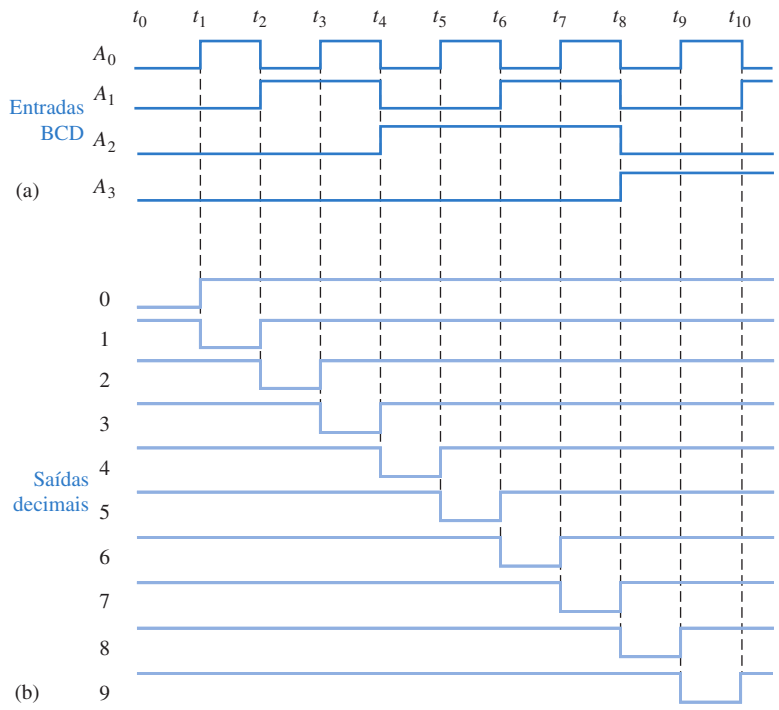
EXEMPLO 6–10

O CI 74HC42 é um decodificador de BCD para decimal. O símbolo lógico é mostrado na Figura 6–32. Se as formas de onda de entrada vistas na Figura 6–33(a) são aplicadas nas entradas do CI 74HC42, mostre as formas de onda de saída.



▲ FIGURA 6–32

Decodificador de BCD para decimal (74HC42).



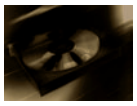
▲ FIGURA 6–33

Solução As formas de onda de saída são mostradas na Figura 6–33(b). Como podemos ver, as entradas são uma seqüência BCD para os dígitos de 0 a 9. As formas de onda de saída no diagrama de temporização indicam essa seqüência BCD nas saídas de valores decimais.

Problema relacionado Construa um diagrama de temporização mostrando as formas de onda de entrada e saída para o caso em que a seqüência de números decimais através das entradas BCD é a seguinte: 0, 2, 4, 6, 8, 1, 3, 5 e 9.

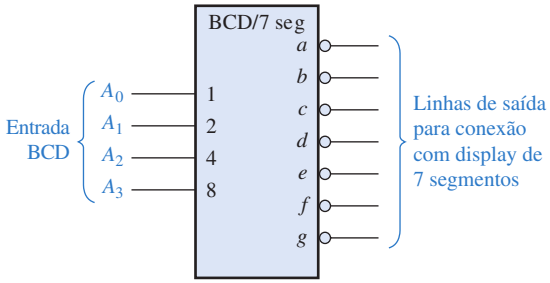
Decodificador de BCD para 7 Segmentos

O decodificador de BCD para 7 segmentos aceita o código BCD em suas entradas e fornece saídas para acionar displays de 7 segmentos para produzir uma leitura decimal. O diagrama lógico para um decodificador de 7 segmentos básico é mostrado na Figura 6–34.

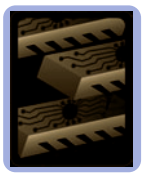


▶ FIGURA 6–34

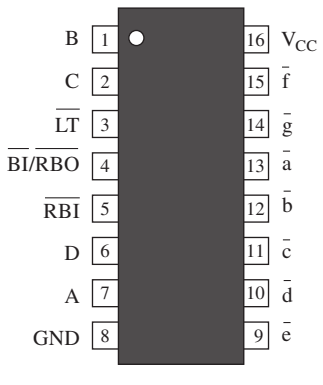
Símbolo lógico para um decodificador/driver de BCD para 7 segmentos com saídas ativas em nível BAIXO. Abra o arquivo F06-34 para verificar a operação.



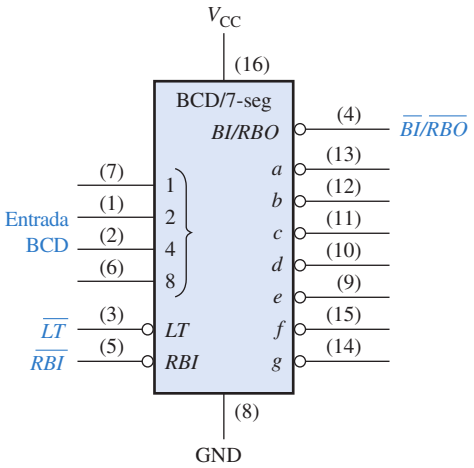
UM DECODIFICADOR/DRIVER DE BCD PARA 7 SEGMENTOS (74LS47)



O CI 74LS47 é um exemplo de um CI que decodifica uma entrada BCD e aciona um display de 7 segmentos. Além dessa capacidade de decodificação e acionamento de segmento, o CI 64LS47 tem algumas características adicionais conforme indicado pelas funções $\overline{LT}$, $\overline{RBI}$, $\overline{BI}/\overline{RBO}$ no símbolo lógico visto na Figura 6–35. Conforme indicado pelos pequenos círculos no símbolo lógico, todas as saídas (de *a* a *g*) são ativas em nível baixo como são as funções $\overline{LT}$ (teste de lâmpada), *RBI* (entrada de apagamento) e $\overline{BI}/\overline{RBO}$ (entrada de apagamento/saída de apagamento). As saídas podem acionar diretamente um display de 7 segmentos do tipo anodo comum. Lembre-se que os displays de 7 segmentos foram discutidos no Capítulo 4. Além de decodificar uma entrada BCD e produzir as saídas apropriadas de 7 segmentos, o CI 74LS47 tem capacidade de teste de lâmpada e supressão de zero. Esse dispositivo pode ser comercializado em outras famílias TTL ou CMOS. Verifique o site da Texas Instruments (www.ti.com) ou o CD-ROM da Texas Instruments que acompanha este livro.



(a) Diagrama de pinos



(b) Símbolo lógico

▶ FIGURA 6–35

Diagrama de pinos e símbolo lógico para o CI 74LS47 (decodificador/driver de BCD para 7 segmentos).

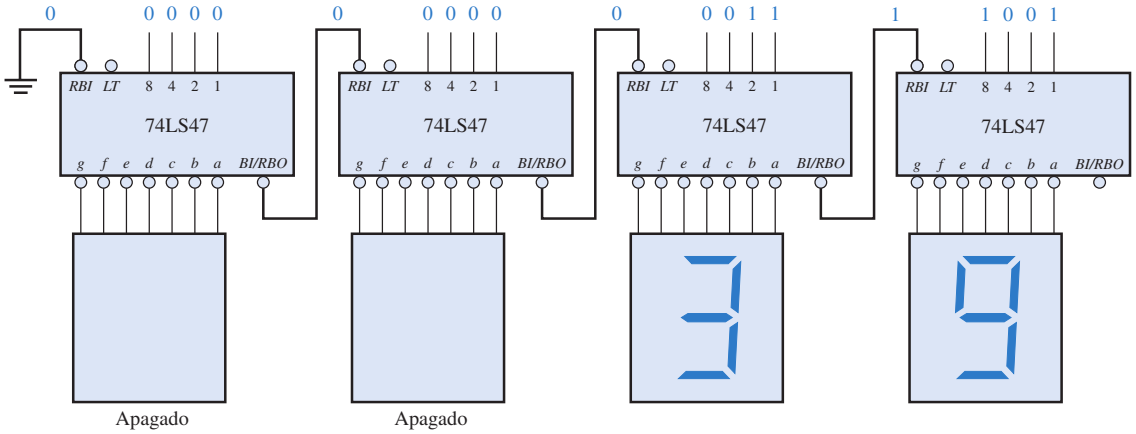
Teste de Lâmpada Quando um nível BAIXO é aplicado na entrada $\overline{LT}$ e $\overline{BI}/\overline{RBO}$ for nível ALTO, todos os 7 segmentos do display são ligados. O teste de lâmpada é usado para verificar se algum segmento está queimado.

Supressão de Zero A **supressão de zero** é uma característica usada por displays de múltiplos dígitos para apagar os zeros não necessários. Por exemplo, num display de 6 dígitos o número 6,4 pode ser mostrado como 006,400 se os zeros não forem apagados. O apagamento dos zeros no início do número é denominado de *supressão de zeros mais significativos* e o apagamento de zeros no final do número é denominado de *supressão de zeros menos significativos*. Tenha em mente que apenas os zeros não necessários são apagados. Com a supressão de zeros o número 030,080 será mostrado como 30,08 (os zeros essenciais são mantidos).

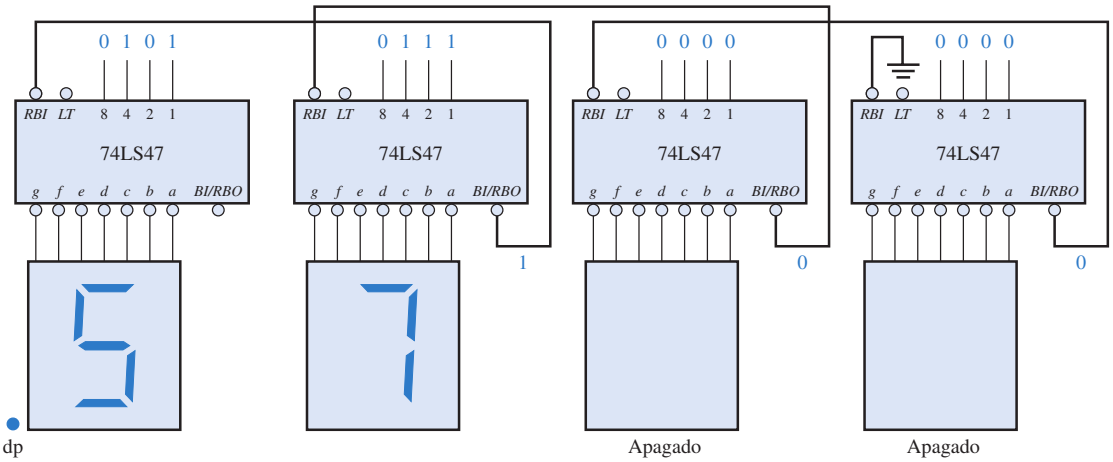
A supressão de zeros resulta em zeros mais significativos e menos significativos num número não mostrado no display.

A supressão de zero no CI 74LS47 é realizada usando as funções $\overline{RBI}$ e $\overline{BI}/\overline{RBO}$. $\overline{RBI}$ é a entrada de apagamento e $\overline{RBO}$ é a saída de apagamento no 74LS47; essas são usadas para supressão de zeros. $\overline{BI}$ é a entrada de apagamento que compartilha o mesmo pino com $\overline{RBO}$; em outras palavras, o pino $\overline{BI}/\overline{RBO}$ pode ser usado como uma entrada ou uma saída. Quando usado como $\overline{BI}$ (entrada de apagamento), todas as saídas de segmentos são nível ALTO (desativadas) quando $\overline{BI}$ for nível BAIXO, o que anula todas as outras entradas. A função $\overline{BI}$ não faz parte da capacidade de supressão de zeros do dispositivo.

Todas as saídas de segmentos estarão desativadas (nível ALTO) se um código zero (0000) for colocado nas entradas BCD e se sua entrada $\overline{RBI}$ estiver em nível BAIXO. Isso faz com que o display apague e produza um nível BAIXO em $\overline{RBO}$.



(a) Ilustração da supressão de zeros mais significativos.



(b) Ilustração da supressão de zeros menos significativos.

▲ **FIGURA 6–36**

Exemplo de supressão de zeros usando um decodificador/driver de BCD para 7 segmentos (74LS47).

O diagrama lógico na Figura 6–36(a) ilustra a supressão de zeros mais significativos para um número inteiro. A posição do dígito mais significativo (mais à esquerda) estará sempre apagada se um código zero estiver nas entradas BCD porque a entrada $\overline{RBI}$ do decodificador mais significativo é colocada em nível BAIXO pela conexão em GND. A saída $\overline{RBO}$ de cada decodificador é conectada à entrada $\overline{RBI}$ do próximo decodificador de menor ordem de forma que todos os zeros à esquerda do primeiro dígito diferente de zero sejam apagados. Por exemplo, na parte (a) da figura os dois dígitos mais significativos são zeros e, portanto, estão apagados. Os dois dígitos restantes, 3 e 9, são mostrados.

O diagrama lógico visto na Figura 6–36(b) ilustra a supressão de zeros menos significativos para um número fracionário. O dígito de menor ordem (mais à direita) é sempre apagado se o código do zero estiver nas entradas BCD porque a entrada $\overline{RBI}$ está conectada em GND. A saída $\overline{RBO}$ de cada decodificador está conectada na entrada $\overline{RBI}$ do próximo decodificador de ordem maior de forma que todos os zeros à direita do primeiro dígito diferente de zero são apagados. Na parte (b) da figura, os dois dígitos de menor ordem são zeros e, portanto, são apagados. Os dois dígitos restantes, 5 e 7, são mostrados. Para combinar a supressão de zeros mais e menos significativos em um display e ter a capacidade de indicação de ponto (vírgula) decimal, é necessária uma lógica adicional.

SEÇÃO 6–5 REVISÃO

1. Um decodificador de 3 linhas para 8 linhas pode ser usado para a decodificação de octal para decimal. Quando um binário 101 for colocado nas entradas, qual linha de saída é ativada?
2. Quantos CIs 74HC154 (decodificador 1 de 16) são necessários para decodificar um número binário de 6 bits?
3. Você escolheria um decodificador/driver com saídas ativas em nível ALTO ou BAIXO para acionar um display de LEDs do tipo catodo comum?

6-6 CODIFICADORES

Um **codificador** é um circuito lógico que realiza essencialmente a função "inversa" do decodificador. Um codificador aceita um nível ativo em uma de suas entradas representando um dígito, tal como um dígito decimal ou octal, e o converte em uma saída codificada, tal como binário ou BCD. Codificadores também podem ser implementados para codificar vários símbolos e caracteres alfabéticos. O processo de conversão de símbolos familiares ou números para um formato codificado é denominado de *codificação*.

Ao final do estudo desta seção você deverá ser capaz de:

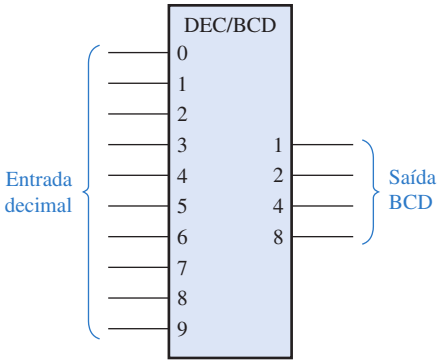
■ Determinar a lógica para um codificador decimal ■ Explicar a finalidade da característica de prioridade em codificadores ■ Descrever um codificador de prioridade de decimal para BCD (74HC154) ■ Descrever um codificador de prioridade de octal para binário (74LS148) ■ Expandir um codificador ■ Usar um codificador numa aplicação específica

Codificador de Decimal para BCD

Este tipo de codificador tem dez entradas – uma para cada dígito decimal – e quatro saídas correspondentes ao código BCD, conforme mostra a Figura 6–37. Esse é um codificador básico de 10 linhas para 4 linhas.

O código BCD (8421) é mostrado na Tabela 6–6. A partir dessa tabela podemos determinar a relação entre cada bit BCD e os dígitos decimais em ordem para analisar a lógica. Por exemplo, o bit mais significativo do código BCD, A_3, é sempre nível 1 para o dígito decimal 8 ou 9. Portanto, pode-se escrever uma expressão OR para o bit A_3 em termos dos dígitos decimais como

$$A_3 = 8 + 9$$



◄ FIGURA 6–37

Símbolo lógico para um codificador de decimal para BCD.

◄ TABELA 6–6

DÍGITO DECIMAL	CÓDIGO BCD A_3	A_2	A_1	A_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

O bit A_2 é sempre nível 1 para o dígito decimal 4, 5, 6 ou 7 e pode ser expresso como uma função OR conforme a seguir:

$$A_2 = 4 + 5 + 6 + 7$$

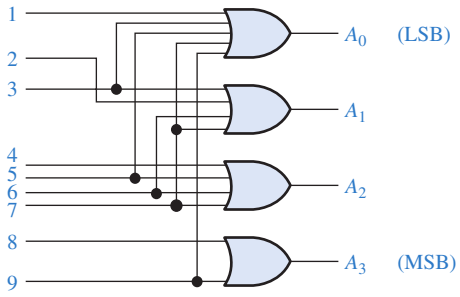
O bit A_1 é sempre nível 1 para o dígito decimal 2, 3, 6 ou 7 e pode ser expresso como

$$A_1 = 2 + 3 + 6 + 7$$

Finalmente, A_0 é sempre nível 1 para o dígito decimal 1, 3, 5, 7 ou 9. A expressão para A_0 é

$$A_0 = 1 + 3 + 5 + 7$$

Agora vamos implementar o circuito lógico necessário para a codificação de cada dígito decimal para o código BCD usando as expressões lógicas desenvolvidas. Para formar cada saída BCD basta simplesmente realizar uma operação OR entre as linhas de entrada dos dígitos decimais apropriados. A lógica do codificador básico resultante dessas expressões é mostrada na Figura 6–38.



◄ FIGURA 6–38

Diagrama lógico básico de um codificador de decimal para BCD. Uma entrada de dígito 0 não é necessária porque as saídas BCD são todas nível BAIXO quando não existirem entradas em nível ALTO.

NOTA: COMPUTAÇÃO

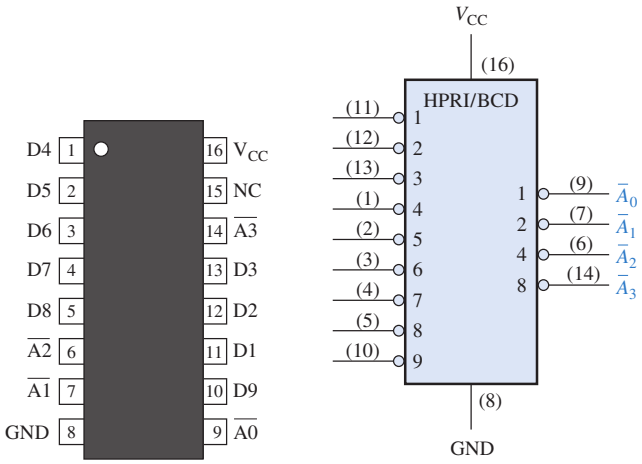
Um *assemblador* pode ser idealizado como um software codificador porque ele interpreta as instruções mnemônicas com as quais um programa é escrito e executa a codificação aplicável para converter cada mnemônico para uma instrução em código de máquina (uma série de 1s e 0s) a qual o computador pode entender. Como exemplos de mnemônicos temos ADD, MOV (mover dados), MUL (multiplicar), XOR, JMP (jump) e OUT (saída para uma porta de I/O).

A operação básica do circuito visto na Figura 6–38 é a seguinte: quando um nível ALTO aparece em uma das linhas de entrada de dígito decimal, os níveis apropriados aparecem nas quatro linhas de saída BCD. Por exemplo, se a linha de entrada 9 for nível ALTO (considerando que todas as outras linhas de entrada estejam em nível BAIXO), essa condição produzirá um nível ALTO nas saídas A_0 e A_3 e um nível BAIXO nas saídas A_1 e A_2, que é o código BCD (1001) para o decimal 9.

Codificador de Prioridade de Decimal para BCD Esse tipo de codificador realiza a mesma função de codificação básica discutida anteriormente. Um **codificador de prioridade** oferece também uma flexibilidade adicional na qual ele pode ser usado em aplicações que requerem detecção de prioridade. A função de prioridade significa que o codificador produzirá uma saída BCD correspondente à entrada do *dígito decimal mais significativo* que estiver ativado ignorando qualquer outra entrada ativa menos significativa. Por exemplo, se as entradas 6 e 3 estiverem ativas, a saída BCD será 0110 (que representa o decimal 6).

UM CODIFICADOR DE DECIMAL PARA BCD (74HC147)

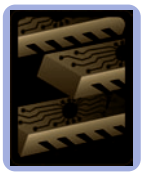
O CI 74HC147 é um codificador de prioridade com entradas ativas em nível BAIXO (0) para os dígitos decimais de 1 a 9 e saídas BCD ativas em nível BAIXO conforme indicado no símbolo lógico mostrado na Figura 6–39. Uma saída BCD zero é representada quando nenhuma das entradas estiver ativa. Os números dos pinos do dispositivo estão entre parênteses. Esse dispositivo pode ser comercializado em outras famílias CMOS ou TTL. Verifique o site da Texas Instruments (www.ti.com) ou o CD-ROM da Texas Instruments que acompanha esse livro.



► **FIGURA 6–39**

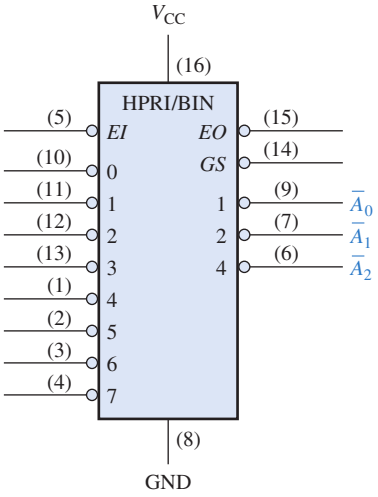
Diagrama de pinos e símbolo lógico para o codificador de prioridade de decimal para BCD 74HC147 (HPRI significa *highest value input has priority* – a entrada de valor mais alto tem prioridade).

UM CODIFICADOR DE 8 LINHAS PARA 3 LINHAS (74LS148)



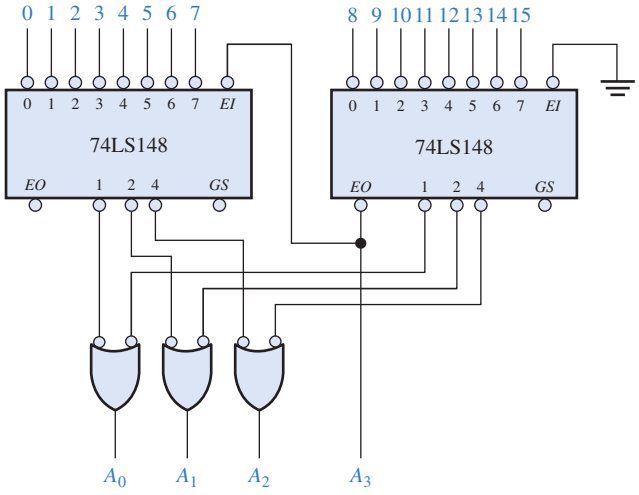
O CI 74LS148 é um codificador de prioridade que tem oito entradas ativas em nível baixo e três saídas binárias ativas em nível ALTO, conforme mostra a Figura 6–40. Esse dispositivo pode ser usado para converter entradas octal (lembre-se que os dígitos octais são de 0 a 7) para um código binário de 3 bits. Para habilitar o dispositivo, a entrada *EI* (entrada de habilitação) tem que ser nível BAIXO. Ele também tem a saída *EO* (saída de habilitação) e a saída *GS* para fins de expansão. A saída *EO* é nível BAIXO quando a entrada *EI* for nível BAIXO e nenhuma das entradas (de 0 a 7) estiver ativa. A saída *GS* é nível BAIXO quando a entrada *EI* for nível BAIXO e qualquer uma das entradas estiver ativa. Esse dispositivo pode

ser comercializado em outras famílias TTL ou CMOS. Verifique o site da Texas Instruments (www.ti.com) ou o CD-ROM da Texas Instruments que acompanha esse livro.



◄ **FIGURA 6–40**
Símbolo lógico para o codificador de 8 para 3 linhas 74LS148.

O CI 74LS148 pode ser expandido para um codificador de 16 linhas para 4 linhas conectando a saída *EO* do codificador mais significativo na entrada *EI* do codificador menos significativo e fazendo uma operação OR negativa entre as correspondentes saídas binárias como mostra a Figura 6–41. A saída *EO* é usada como o quarto bit (MSB). Essa configuração particular produz saídas ativas em nível ALTO para o número binário de 4 bits.



◄ **FIGURA 6–41**
Um codificador de 16 linhas para 4 linhas usando CIs 74LS148 e lógica externa.

EXEMPLO 6–11

Se aparecerem níveis BAIXOs nos pinos 1, 4 e 13 do CI 74HC147 mostrado na Figura 6–39, indique o estado das quatro saídas. Todas as outras entradas estão em nível ALTO.

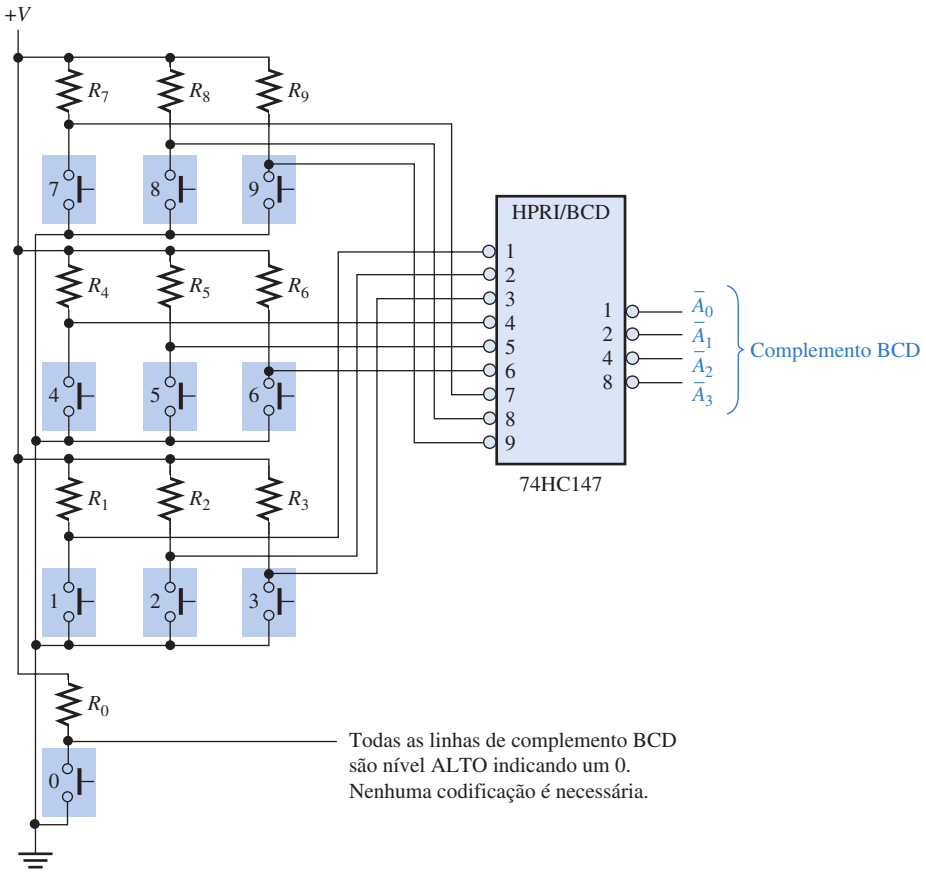
Solução O pino 4 é a entrada de dígito decimal mais significativo que tem um nível BAIXO e representa o decimal 7. Portanto, os níveis de saída indicam o código BCD para o decimal 7 onde $\overline{A}_0$ é o LSB e $\overline{A}_3$ é o MSB. A saída $\overline{A}_0$ é nível BAIXO, $\overline{A}_1$ é nível BAIXO, $\overline{A}_2$ é nível BAIXO e $\overline{A}_3$ é nível ALTO.

Problema relacionado Quais são as saídas do CI 74HC147 se todas as entradas estiverem em nível BAIXO? E se todas as entradas estiverem em nível ALTO?

Uma Aplicação

Um exemplo de aplicação clássica é um codificador de teclado. Os dez dígitos decimais no teclado de um computador, por exemplo, tem que ser codificado para ser processado pelo circuito lógico. Quando uma das teclas é pressionada, o dígito decimal é codificado para o código BCD correspondente. A Figura 6–42 mostra a configuração de um codificador de teclado simples usando um CI codificador de prioridade 74HC147. As teclas são representadas por dez chaves push-button, cada uma com um **resistor de pull-up** (elevador) para +V. O resistor de pull-up garante que a linha será nível ALTO quando a tecla não estiver pressionada. Quando a tecla for pressionada, a linha é conectada em GND, sendo que um nível BAIXO é aplicado na entrada correspondente do codificador. A tecla zero não é conectada porque a saída BCD representa zero quando nenhuma das teclas é pressionada.

A saída BCD complementada do codificador vai para um dispositivo de armazenamento, sendo que cada código BCD sucessivo é armazenado até que o número completo tenha sido digitado. Os métodos para armazenamento de números BCD e dados binários são abordados em capítulos posteriores.



▶ **FIGURA 6–42**

Um codificador de teclado simplificado.

SEÇÃO 6–6 REVISÃO

1. Suponha que sejam aplicados níveis ALTOs nas entradas 2 e 9 do circuito visto na Figura 6–38.

(a) Quais são os estados das linhas de saída?

(b) Elas representam um código BCD válido?

(c) Qual é a restrição na lógica de codificação mostrada na Figura 6–38?

2. (a) Quais são os estados das saídas $\overline{A}_3\overline{A}_2\overline{A}_1\overline{A}_0$ quando são aplicados níveis BAIXOs nos pinos 1 e 5 do CI 74HC147 visto na Figura 6–39?

(b) O que essa saída representa?

6-8 MULTIPLEXADORES (SELETORES DE DADOS)

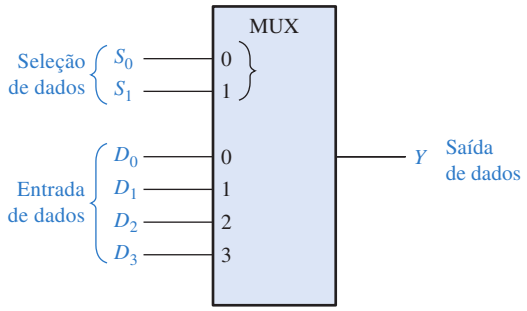
Um **multiplexador (MUX)** é um dispositivo que permite que informações digitais de diversas fontes sejam encaminhadas para uma única linha para serem transmitidas nessa linha para um destino comum. Um multiplexador básico tem várias linhas de entrada de dados e uma única linha de saída. Ele também possui entradas de seleção de dados, as quais permitem que os dados digitais de quaisquer entradas sejam comutados para a linha de saída. Os multiplexadores também são conhecidos como seletores de dados.

Ao final do estudo desta seção você deverá ser capaz de:

■ Explicar a operação básica de um multiplexador ■ Descrever os CIs multiplexadores 74LS151 e 74HC157 ■ Expandir um multiplexador para poder operar com mais entradas de dados ■ Usar um multiplexador como um gerador de expressões lógicas

Em um multiplexador, os dados passam das diversas linhas para uma linha.

A Figura 6–46 mostra o símbolo lógico para um multiplexador (MUX) de 4 bits. Observe que existem duas linhas de seleção de dados porque com dois bits de seleção, qualquer uma das quatro linhas de entrada de dados pode ser selecionada.



▶ **FIGURA 6–46**

Símbolo lógico para um seletor/multiplexador de dados 1 de 4.

Na Figura 6–46, um código de 2 bits nas entradas de seleção de dados (S) permitem que o dado na entrada selecionada passe para a saída de dados. Se um binário 0 ($S_1 = 0$ e $S_0 = 0$) for aplicado nas linhas de seleção de dados, o dado na entrada D_0 aparece na linha de saída de dados. Se um binário 1 ($S_1 = 0$ e $S_0 = 1$) for aplicado nas linhas de seleção de dados, o dado na entrada D_1 aparece na linha de saída de dados. Se um binário 2 ($S_1 = 1$ e $S_0 = 0$) for aplicado, o dado na entrada D_2 aparece na linha de saída de dados. Se um binário 3 ($S_1 = 1$ e $S_0 = 1$) for aplicado, o dado na entrada D_3 é comutado para a linha de saída de dados. A Tabela 6–8 mostra um resumo dessa operação.

▶ **TABELA 6–8**

Seleção de dados para um multiplexador 1 de 4

ENTRADAS DE SELEÇÃO DE DADOS S_1	S_0	ENTRADA SELECIONADA
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

NOTA: COMPUTAÇÃO

Um barramento é uma via ao longo da qual sinais elétricos são enviados de um ponto para outro no computador. Numa rede de computadores, um *barramento compartilhado* é aquele que está conectado a todos os microprocessadores do sistema com a finalidade de troca de dados. Um barramento compartilhado pode conter dispositivos de memória e de entrada/saída (in/out) os quais podem ser acessados por todos os microprocessadores do sistema. O acesso ao barramento compartilhado é controlado pelo árbitro de barramento (um multiplexador de classificações) o qual permite que apenas um microprocessador de cada vez use o barramento compartilhado do sistema.

Agora vamos pensar no circuito lógico necessário para realizar essa operação de multiplexação. A saída de dados é igual ao estado da entrada selecionada. Portanto, podemos deduzir uma expressão para a saída em termos da entrada de dados e das entradas de seleção.

A saída de dados é igual a D_0 apenas se $S_1 = 0$ e $S_0 = 0$: $D_0\overline{S}_1\overline{S}_0$.

A saída de dados é igual a D_1 apenas se $S_1 = 0$ e $S_0 = 1$: $D_1\overline{S}_1S_0$.

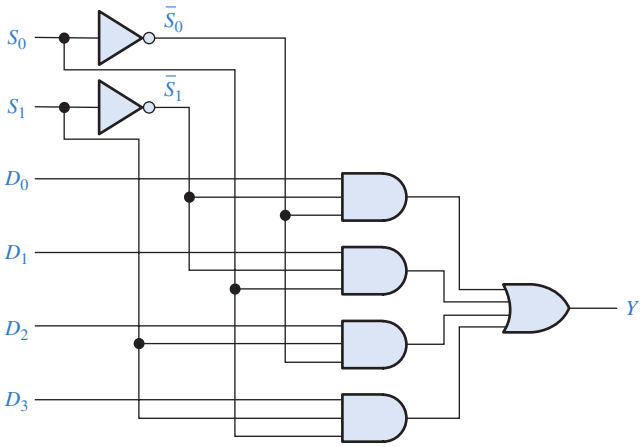
A saída de dados é igual a D_2 apenas se $S_1 = 1$ e $S_0 = 0$: $D_2S_1\overline{S}_0$.

A saída de dados é igual a D_3 apenas se $S_1 = 1$ e $S_0 = 1$: $D_3S_1S_0$.

Quando esses termos são relacionados por uma operação OR, a expressão total para a saída de dados é

$$Y = D_0\overline{S}_1\overline{S}_0 + D_1\overline{S}_1S_0 + D_2S_1\overline{S}_0 + D_3S_1S_0$$

A implementação dessa equação requer quatro portas AND de 3 entradas, uma porta OR de 4 entradas e dois inversores para gerar o complemento de S_1 e S_0, conforme mostra a Figura 6–47. Como os dados podem ser selecionados a partir de qualquer uma das linhas de entrada, esse circuito também é denominado de **seletor de dados**.

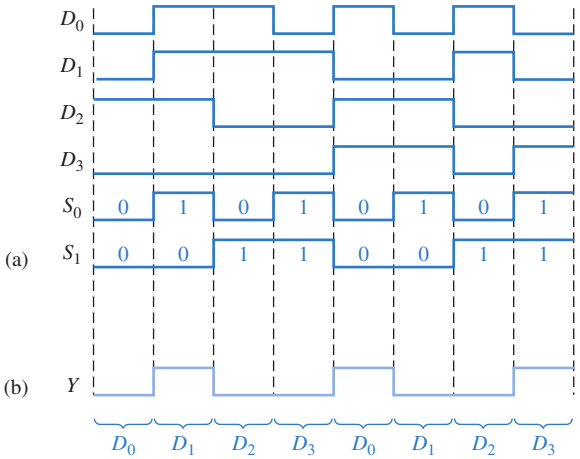


◀ **FIGURA 6–47**

Diagrama lógico para um multiplexador de 4 entradas. Abra o arquivo F06-47 para verificar a operação.

EXEMPLO 6–14

As formas de onda da entrada de dados e das entradas de seleção de dados vistas na Figura 6–48(a) são aplicadas no multiplexador mostrado na Figura 6–47. Determine a forma de onda de saída em relação às entradas.



▶ **FIGURA 6–48**

Solução Os estados binários das entradas de seleção de dados durante cada intervalo determina qual dado de entrada é selecionado. Observe que as entradas de seleção de dados passam pela seqüência binária repetitiva: 00, 01, 10, 11, 00, 01, 10, 11 e assim por diante. A forma de onda de saída resultante é mostrada na Figura 6–48(b).

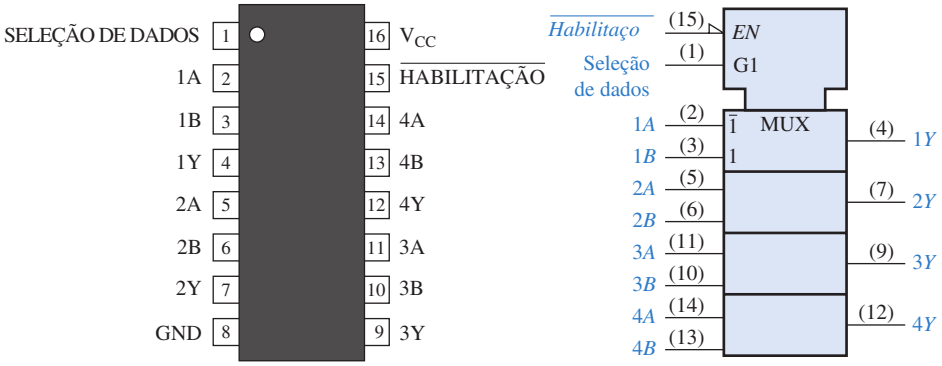
Problema relacionado Construa um diagrama de temporização mostrando todas as entradas e a saída se as formas de onda de S_0 e S_1, vistas na Figura 6–48(b), forem trocadas entre si.

UM SELETOR/MULTIPLEXADOR DE DADOS QUÁDRUPLO DE 2 ENTRADAS (74HC157)

O CI 74HC157, bem como a sua versão LS, consiste em quatro multiplexadores de 2 entradas separados. Cada um dos quatro multiplexadores compartilha a linha de seleção de dados e a entrada de habilitação (*EN*). Como existem apenas duas entradas a serem selecionadas em cada multiplexador, uma única entrada de seleção de dados é suficiente.

Um nível BAIXO na entrada de habilitação permite que o dado da entrada selecionada passe para a saída. Um nível ALTO na entrada evita a passagem do dado para a saída; ou seja, com a entrada nesse estado os multiplexadores estão desabilitados. Esse dispositivo pode ser comercializado em outras famílias CMOS e TTL. Verifique o site da Texas Instruments (www.ti.com) ou o CD-ROM da Texas Instruments que acompanha esse livro.

O Símbolo Lógico ANSI/IEEE O diagrama de pinos para o CI 74HC157 é mostrado na Figura 6–49(a). O símbolo lógico ANSI/IEEE para o CI 74HC157 é mostrado na Figura 6–49(b). Observe que os quatro multiplexadores são indicados por um contorno retangular e que as entradas comuns aos quatro multiplexadores são indicadas como entradas de um bloco com entalhes na parte superior, o qual é denominado de *bloco de controle comum*. Todas as identificações dentro do bloco MUX superior se aplicam aos outros blocos abaixo dele.



(a) Diagrama de pinos (b) Símbolo lógico

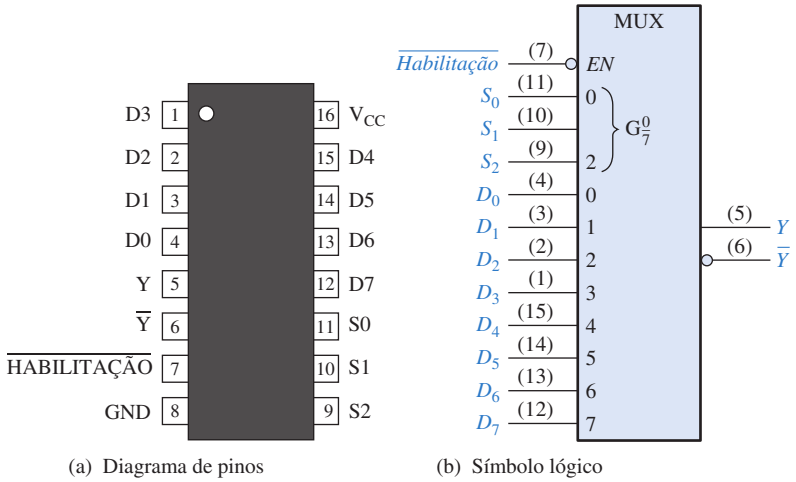
► FIGURA 6–49

Diagrama de pinos e símbolo lógico para o CI 74HC157 (quatro seletores/multiplexadores de dados de 2 entradas).

Observe as identificações 1 e $\overline{1}$ nos blocos MUX e a identificação G1 no bloco de controle comum. Essas identificações são um exemplo do sistema de **notação de dependência** especificado no padrão 91-1984 da ANSI/IEEE. Nesse caso, G1 indica uma relação AND entre a entrada de seleção de dados e as entradas de dados com indicações 1 ou $\overline{1}$. (O $\overline{1}$ significa que a relação AND se aplica ao complemento da entrada G1). Em outras palavras, quando a entrada de seleção de dados for nível ALTO, as entradas *B* dos multiplexadores são selecionadas; e quando a entrada de seleção de dados for nível BAIXO, as entradas *A* são selecionadas. Um "G" sempre é usado para indicar uma dependência AND. Outros aspectos da notação de dependência são apresentados em momentos apropriados ao longo desse livro.

UM SELETOR/MULTIPLEXADOR DE DADOS DE 8 ENTRADAS (74LS151)

O CI 74LS151 tem oito entradas de dados (D_0–D_7) e, portanto, três entradas de seleção de dados, ou endereço, (S_0–S_2). Três bits são necessários para selecionar qualquer uma das oito entradas de dados ($2^3 = 8$). Um nível BAIXO na entrada de $\overline{habilitação}$ permite que a entrada de dados selecionada passe para a saída. Observe que a saída de dados e o seu complemento estão disponíveis. A Figura 6–50(a) mostra o diagrama de pinos e a parte (b) mostra o símbolo lógico. Nesse caso não existe um bloco de controle comum porque existe apenas um multiplexador a ser controlado, e não quatro como no CI 74HC157. A indicação $G\frac{0}{7}$ dentro do símbolo lógico representa a relação AND entre as entradas de seleção de dados e cada uma das entradas de dados (de 0 a 7). Esse dispositivo pode ser comercializado em outras famílias TTL e CMOS. Verifique o site da Texas Instruments (www.ti.com) ou o CD-ROM da Texas Instruments que acompanha este livro.



◄ **FIGURA 6–50**

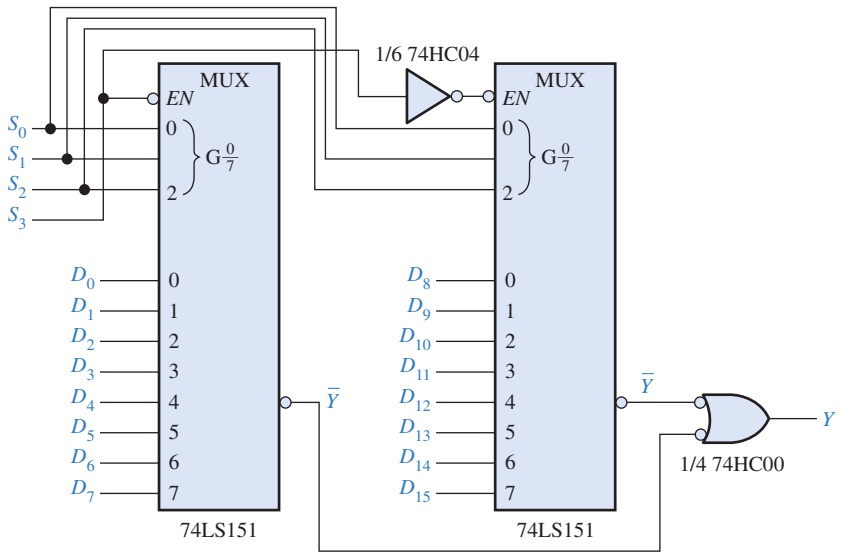
Diagrama de pinos e símbolo lógico para o CI 74LS151 (seletor/multiplexador de dados de 8 entradas).

EXEMPLO 6–15

Use CIs 74LS151 e qualquer outra lógica necessária para multiplexar 16 linhas de dados em uma única linha de dados de saída.

Solução

A Figura 6–51 mostra uma implementação desse sistema. São necessários quatro bits para selecionar uma das 16 entradas de dados ($2^4 = 16$). Nessa aplicação a entrada de $\overline{habilitação}$ é usada como o bit de seleção de dados mais significativo. Quando o MSB no código de seleção de dados for nível BAIXO, o CI 74LS151 à esquerda será habilitado, sendo que um dos dados de entrada (D_0 a D_7) será selecionado pelos outros três bits de seleção de dados. Quando o MSB da seleção de dados for nível ALTO, o CI 74LS151 à direita será habilitado, sendo que uma das entradas de dados (D_8 a D_{15}) será selecionada. O dado da entrada selecionada passa então pela porta OR negativa saindo pela única linha de saída.



► **FIGURA 6–51**

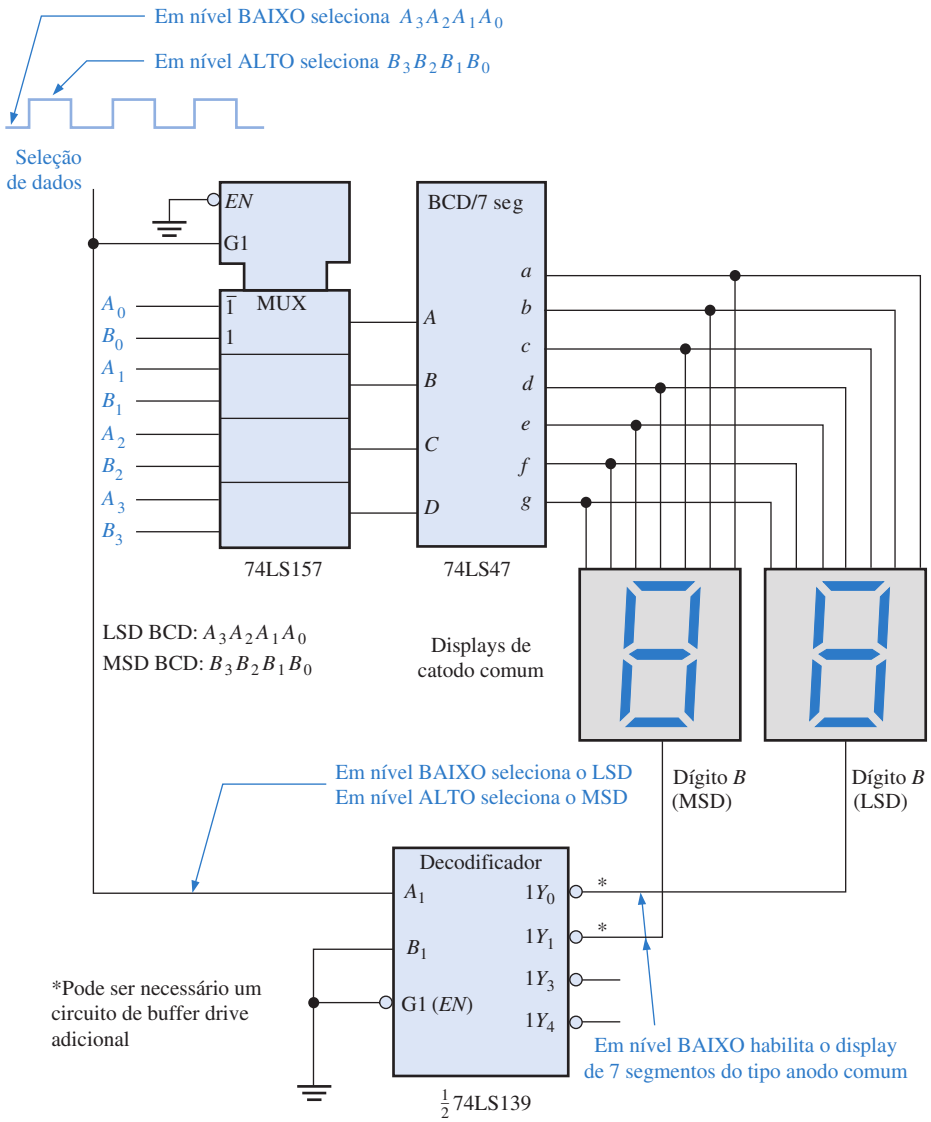
Um multiplexador de 16 entradas.

Problema relacionado

Determine os códigos nas entradas de seleção necessários para selecionar cada uma das seguintes entradas de dados: D_0, D_4, D_8 e D_{13}.

Aplicações

Multiplexador para Display de 7 Segmentos A Figura 6–52 mostra um método simplificado de multiplexação de números BCD para displays de 7 segmentos. Nesse exemplo são mostrados números de 2 dígitos em displays de 7 segmentos usando um único decodificador de BCD para 7 segmentos. Esse método básico de multiplexação de displays pode ser estendido para sistemas com qualquer número de dígitos.



▶ **FIGURA 6–52**
Lógica simplificada de multiplexação de displays de 7 segmentos.

A operação básica é descrita logo a seguir. Os dois dígitos BCD ($A_3A_2A_1A_0$ e $B_3B_2B_1B_0$) são aplicados nas entradas do multiplexador. Uma onda quadrada é aplicada na linha de seleção de dados, sendo que quando essa linha for nível BAIXO, os bits A ($A_3A_2A_1A_0$) passam para as entradas do decodificador de BCD para 7 segmentos (74LS47). Um nível BAIXO na entrada de seleção de dados é aplicado também na entrada A_1 do decodificador de 2 linhas para 4 linhas (74LS139), ativando então a saída 0 deste CI e habilitando o display do dígito *A* conectando efetivamente o seu terminal comum em GND. Desta forma, o dígito *A* estará *ligado* (*on*) e o dígito *B* *desligado* (*off*).

Quando a linha de seleção de dados for para nível ALTO, os bits B ($B_3B_2B_1B_0$) passam para as entradas do decodificador de BCD para 7 segmentos. Além disso, a saída 1 do CI decodificador 74LS139 é ativada, habilitando então o display do dígito B. Agora o dígito B estará *ligado* (*on*) e o dígito A *desligado* (*off*). O ciclo se repete na freqüência da onda quadrada na entrada de seleção de dados. Essa freqüência tem que ser alta o suficiente (cerca de 30 Hz) para evitar cintilações (*flickers*) conforme os displays dos dígitos são multiplexados.

Gerador de Funções Lógicas Uma aplicação útil para um seletor/multiplexador de dados é na geração de funções lógicas combinacionais na forma de soma-de-produtos. Quando dessa forma, o dispositivo pode substituir portas discretas, podendo freqüentemente diminuir bastante o número de CIs, e pode tornar muito fáceis as alterações de projetos.

Para ilustrar, podemos usar um seletor/multiplexador de dados de 8 entradas (74LS151) para implementar qualquer função lógica especificada de 3 variáveis sendo as variáveis conectadas nas entradas de seleção de dados e cada entrada de dado submetida ao nível lógico necessário de acordo com a tabela-verdade para a função em questão. Por exemplo, se a função for nível 1 quando a combinação de variáveis for $\overline{A}_2A_1\overline{A}_0$, a entrada 2 (selecionada por 010) deverá estar conectada ao nível ALTO. Esse nível ALTO passará para a saída quando essa combinação particular de variáveis ocorrer nas linhas de seleção de dados. Um exemplo ajudará a esclarecer essa aplicação.

EXEMPLO 6–16

Implemente a função lógica especificada na Tabela 6–9 usando um CI 74LS151 (seletor/multiplexador de dados de 8 entradas). Compare esse método com uma implementação que usa portas lógicas discretas.

▼ **TABELA 6–9**

ENTRADAS			SAÍDA
A_2	A_1	A_0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Solução Observe a partir da tabela-verdade que Y é nível 1 para as seguintes combinações das variáveis de entrada: 001, 011, 101 e 110. Para todas as outras combinações, Y é nível 0. Para essa função ser implementada com o seletor de dados definido, a entrada de dados selecionada para cada uma das combinações apresentadas tem que ser conectada ao nível ALTO (5 V). Todas as outras entradas de dados têm que ser conectadas ao nível BAIXO (GND), conforme mostra a Figura 6–53.

A implementação dessa função com portas lógicas necessitaria de quatro portas AND de 3 entradas, uma porta OR de 4 entradas e três inversores, a menos que essa expressão possa ser simplificada.

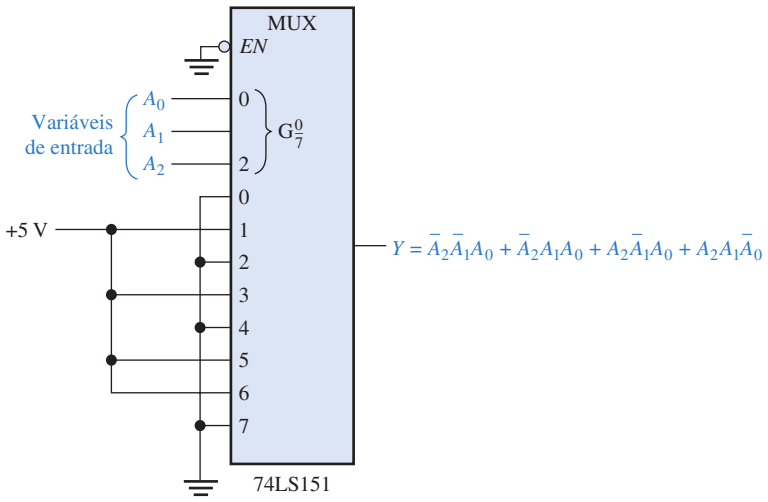


FIGURA 6–53

Seletor/multiplexador de dados conectado como um gerador de funções lógicas de 3 variáveis.

Problema relacionado Use o CI 74LS151 para implentar a seguinte expressão:

$$Y = \overline{A}_2\overline{A}_1\overline{A}_0 + A_2\overline{A}_1\overline{A}_0 + \overline{A}_2A_1\overline{A}_0$$

O Exemplo 6–16 ilustra como um seletor de dados de 8 entradas pode ser usado como um gerador de função lógica para três variáveis. Na realidade, esse dispositivo também pode ser usado como um gerador de função lógica de 4 variáveis fazendo uso de um dos bits (A_0) em conjunto com as entradas de dados.

Uma tabela-verdade de 4 variáveis tem 16 combinações das variáveis de entrada. Quando um seletor de dados de 8 bits é usado, cada entrada é selecionada duas vezes: a primeira vez quando A_0 é nível 0 e a segunda vez quando A_0 é nível 1. Com isso em mente, as regras a seguir podem ser aplicadas (Y é a saída e A_0 é o bit menos significativo):

1. Se $Y = 0$ nas duas vezes em que uma dada entrada for selecionada por uma certa combinação de variáveis de entrada, $A_3A_2A_1$, conecte essa entrada de dados em GND (0).
2. Se $Y = 1$ nas duas vezes em que uma dada entrada for selecionada por uma certa combinação de variáveis de entrada, $A_3A_2A_1$, conecte essa entrada de dados em $+V$ (1).
3. Se Y for diferente nas duas vezes em que uma dada entrada de dados for selecionada por uma certa combinação de variáveis de entrada, $A_3A_2A_1$, e se $Y = A_0$, conecte essa entrada de dados em A_0.
4. Se Y for diferente nas duas vezes em que uma dada entrada de dados for selecionada por uma certa combinação de variáveis de entrada, $A_3A_2A_1$, e se $Y = A_0$, conecte essa entrada de dados em $\overline{A}_0$.

EXEMPLO 6–17

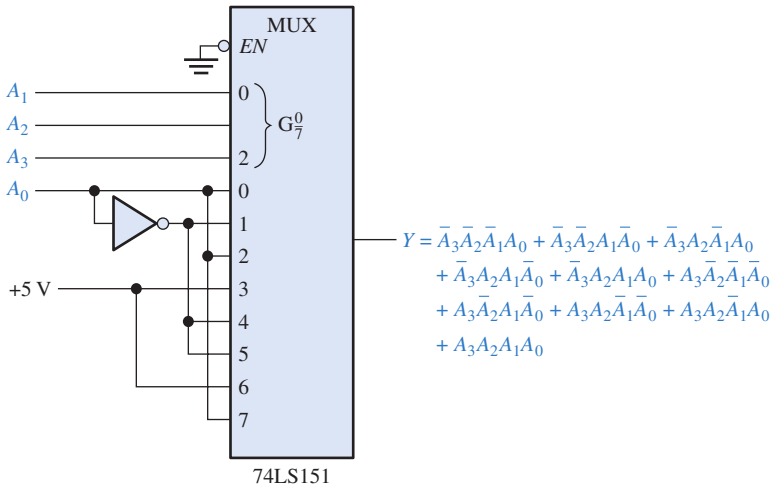
Implemente a função lógica dada pela Tabela 6–10 usando um CI 74LS151 (seletor/multiplexador de dados de 8 entradas). Compare esse método com uma implementação que usa portas lógicas discretas.

Solução As entradas de seleção de dados são $A_3A_2A_1$. Na primeira linha da tabela, $A_3A_2A_1 = 000$ e $Y = A_0$. Na segunda linha, onde $A_3A_2A_1$ é novamente 000, $Y = A_0$. Portanto, A_0 é conectada na entrada 0. Na terceira linha da tabela, $A_3A_2A_1 = 001$ e $Y = \overline{A}_0$. Além disso, na quarta linha, quando $A_3A_2A_1$ também é 001, $Y = \overline{A}_0$. Portanto, $\overline{A}_0$ é invertida e conectada na entra-

▶ **TABELA 6–10**

DÍGITO DECIMAL	ENTRADAS A_3	A_2	A_1	A_0	SAÍDA Y
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	1

da 1. Essa análise prossegue até que cada entrada seja conectada adequadamente de acordo com as regras especificadas. A implementação é mostrada na Figura 6–54.



▶ **FIGURA 6–54**
Seletor/multiplexador de dados conectado como um gerador de função lógica de 4 variáveis.

Caso fosse implementado com portas lógicas, a função necessitaria de dez portas AND de 4 entradas, uma porta OR de 10 entradas e quatro inversores, embora uma possível simplificação reduzisse essas especificações.

Problema relacionado Na Tabela 6–10, se $Y = 0$ quando as entradas estão todas em zero e alternadamente for 1 e 0 para as linhas restantes da tabela, use um CI 74LS151 para implementar a função lógica resultante.

SEÇÃO 6–8 REVISÃO

1. Na Figura 6–47, $D_0 = 0$, $D_1 = 0$, $D_2 = 1$, $D_3 = 0$, $S_0 = 1$ e $S_1 = 0$. Qual é o nível lógico da saída?
2. Identifique cada dispositivo a seguir.
   (a) 74LS157 **(b)** 74LS151
3. Um CI 74LS151 tem alternadamente níveis BAIXO e ALTO em suas entradas de dados começando por D_0. As linhas de seleção de dados são recebem uma seqüência de contagem binária (000, 001, 010 e assim por diante) numa freqüência de 1 kHz. A entrada de habilitação é nível BAIXO. Descreva a forma de onda na saída de dados.
4. Descreva resumidamente a finalidade de cada um dos seguintes dispositivos vistos na Figura 6–52.
   (a) 74LS157 **(b)** 74LS47 **(c)** 74LS139

6-9 DEMULTIPLEXADORES

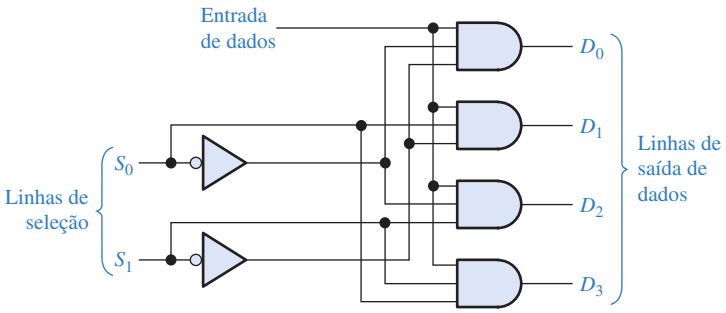
Um **demultiplexador (DEMUX)** basicamente inverte a função da multiplexação. Ele recebe informações digitais a partir de uma linha e as distribui para um determinado número de linhas de saída. Por essa razão, o demultiplexador também é conhecido como distribuidor de dados. Conforme estudaremos, os decodificadores também podem ser usados como demultiplexadores.

Ao final do estudo desta seção você deverá ser capaz de:

■ Explicar a operação básica de um demultiplexador ■ Descrever como o CI decodificador de 4 para 16 linhas 74HC154 pode ser usado como um demultiplexador ■ Desenvolver o diagrama de temporização para um demultiplexador com dados e entradas de seleção de dados especificadas

Em um demultiplexador, os dados são transferidos de uma linha para diversas linhas.

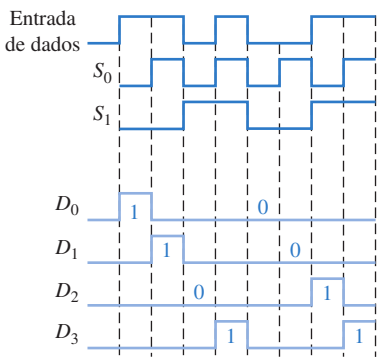
A Figura 6–55 mostra o circuito de um demultiplexador (DEMUX) de 1 linha para 4 linhas. A linha de entrada de dados está conectada em todas as portas AND. As duas linhas de seleção de dados habilitam uma porta de cada vez, e os dados que aparecem na linha de entrada de dados passam, através da porta selecionada, para a linha de saída de dados associada.



▶ **FIGURA 6–55**
Demultiplexador de 1 linha para 4 linhas.

EXEMPLO 6–18

A forma de onda de entrada de dados em série e as entradas de seleção de dados (S_0 e S_1) são mostradas na Figura 6–56. Determine as formas de onda da saída de dados D_0 a D_3 para o demultiplexador visto na Figura 6–55.

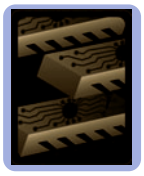


▶ **FIGURA 6–56**

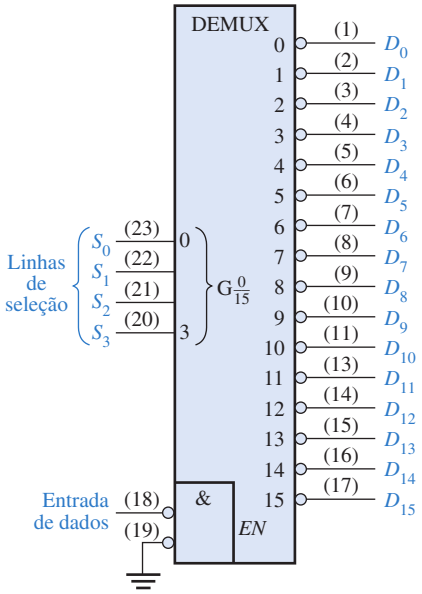
Solução Observe que as linhas de seleção seguem uma seqüência binária de forma que cada bit sucessivo de entrada é direcionado para D_0, D_1, D_2 e D_3 na seqüência, conforme mostra as formas de onda vistas na Figura 6–56.

Problema relacionado Desenvolva o diagrama de temporização para o demultiplexador se as formas de onda de S_0 e S_1 forem invertidas.

O CI DEMULTIPLEXADOR 74HC154



Já discutimos o CI decodificador 74HC154 sendo usado numa aplicação como um decodificador de 4 linhas para 16 linhas (Seção 6–5). Esse dispositivo e outros decodificadores também podem ser usados em aplicações de demultiplexação. O símbolo lógico para esse dispositivo quando usado como um demultiplexador é mostrado na Figura 6–57. Em aplicações como demultiplexador, as linhas de entrada são usadas como linhas de dados. Uma das entradas de seleção de chip é usada como linha de entrada de dados, enquanto a outra entrada de seleção de chip é mantida em nível BAIXO para habilitar a porta AND negativa interna na parte inferior do diagrama. Esse dispositivo pode ser comercializado em outras famílias CMOS ou TTL. Verifique o site da Texas Instruments (www.ti.com) ou o CD-ROM da Texas Instruments que acompanha esse livro.



◀ **FIGURA 6–57**

O CI decodificador 74HC154 usado como um demultiplexador.

Resumos dos ppts
--Coloquei um pouco a mais, nunca se sabe

6. Blocos Combinatórios

Até ao momento desenvolvemos e estudámos métodos básicos usados em sistemas digitais. Por outras palavras, foi uma introdução ao conteúdo que agora vamos trabalhar: **blocos combinatórios**. Logo na primeira unidade (ver §Ambiente digital) referimos que existiam dois tipos genéricos de circuitos: circuitos combinatórios e circuitos sequenciais. Nesta disciplina vamos estudar ambos, mas começaremos pelos circuitos combinatórios, e alguns exemplos. Mas primeiro há que saber alguns aspetos, primários aos circuitos.

blocos combinatórios

Quando nos apresentam um circuito convém que nós tenhamos alguma informação sobre ele, de forma a que o possamos usar em alguma situação. Para tal, existe sempre, ou deve existir, informação detalhada, por outras palavras, **documentação**, que sirva de suporte ao equipamento. Assim, em suma, é essencial que, em toda a vida de um projeto, haja documentação devidamente realizada. Em sistemas digitais há conteúdo dessa documentação, quase, mais-que-essencial. Entre diagramas de blocos, esquemas lógicos (logigramas), esquemas elétricos, HDL (ABEL, Verilog, VHDL), diagramas temporais ou especificações técnicas dos componentes. Podemos sempre encontrar exemplos de diagramas de blocos, esquemas lógicos, diagramas temporais, esquemas elétricos, VHDL, entre outros... tal como também teremos oportunidade de ver ao longo deste capítulo e do seguinte.

documentação

Lógica de Polaridade

Tal como referimos no primeiro capítulo, o conceito lógico de 0's e 1's é apenas uma pura convenção, pois podia ser apenas verdadeiro/falso, maria/joão, ... Do mesmo modo, a convenção que temos de 1 ser correspondente ao nível elétrico mais positivo e o 0 ao nível elétrico mais baixo também é pura convenção. Além disso, existem duas convenções distintas, dependendo da realização eletrónica do equipamento em questão. Assim, podemos encontrar duas lógicas: a **lógica positiva** e a **lógica negativa**.

lógica positiva
lógica negativa

A descrição tabular do comportamento do circuito recorrendo aos níveis *high* e *low* (geralmente usados pelo seu acrónimo H e L, significam lógica positiva e lógica negativa, respetivamente) é inversa uma da outra. Em contextos não algébricos, a descrição funcional deve confinar-se aos símbolos H e L.

Por razões do contexto físico, interessa associar a um sinal uma determinada ação no circuito e o nível elétrico que o desencadeia: este pode ser H ou L (grande parte dos nossos equipamentos utilizam preferencialmente a L); o nível assim definido designa-se por **nível ativo**. Os nomes dos sinais para além de sugestivos devem mencionar também o nível ativo, como podemos ver alguns exemplos na tabela que se segue:

nível ativo

active low	active high
READY- ERROR.L ADDR15(L) RESET* ENABLE~ ~GO /RECEIVE TRANSMIT_L	READY+ ERROR.H ADDR15(H) RESET ENABLE GO RECEIVE TRANSMIT

tabela 9
exemplos de nomes dos sinais

Descodificadores (decoders)

Um **descodificador** é um circuito combinatório que permite, para uma palavra em código à entrada, ativar uma e uma só saída da função de descodificação. O descodificador mais comum de todos é o descodificador para o código binário natural. A **estrutura genérica** de um descodificador é a seguinte:

descodificador

estrutura genérica

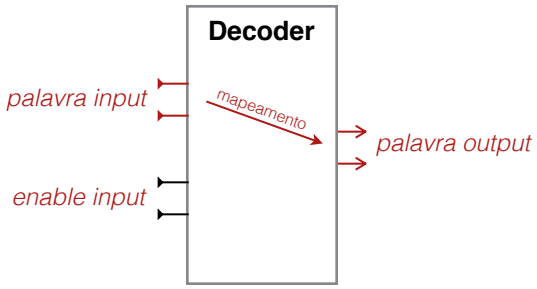
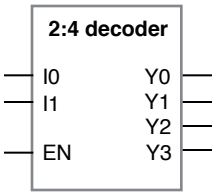


figura 17
estrutura genérica de um decodificador

Como podemos reparar temos um dado número de entradas para um dado número de saídas. Tipicamente esses números são n entradas (código) para 2^n saídas, como são o caso os descodificadores 2:4, 3:8, 4:16, etc...

Na página seguinte podemos ver um descodificador 2 para 4 (2:4) e a sua tabela de verdade. Devemo-nos lembrar que o valor "x" corresponde a um *don't care*, valor a ser desprezado.



inputs			outputs			
EN	I1	I0	Y3	Y2	Y1	Y0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

figura 18
descodificador 2:4 e respetiva tabela de verdade

Como nos foi designada, esta é a tabela de verdade de um descodificador 2:4 e à esquerda encontra-se uma estrutura genérica do mesmo bloco combinatório. Mas qual é o seu diagrama lógico? E o que é a entrada EN?

A entrada EN, de nome **enable**, tal como o nome indica, pode ser vista como um interruptor, para nós, indicando qual a lógica que o sistema em questão usa. Neste caso, quando o EN está a zero, quaisquer que sejam os I1 o I0, o valor de Y0, Y1, Y2, Y3 é sempre 0, logo o nosso sistema é *active high*.

enable

O **diagrama lógico** deste descodificador é o seguinte:

diagrama lógico

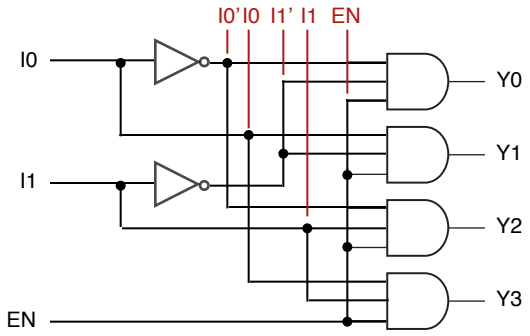


figura 19
diagrama lógico de um descodificador 2:4

É possível, de forma a construir um descodificador, por exemplo, 4:16, através de dois descodificadores 3:8, agrupando-os em **cascata**.

cascata

Podemos recorrer a um bloco de decodificarão para obter os termos mínimos necessários à implementação de uma função booleana genérica.

Codificadores (encoder)

Os **codificadores**, ao contrário dos descodificadores, têm como função principal, codificar uma dada palavra de código. Tendo tipicamente 2^n entradas, codifica essas em palavras, com n saídas.

codificadores

Há um grande problema com os codificadores - há que criar prioridade, de modo a evitar conflitos arbitrários no atendimento do código. Para tal existe um **codificador de prioridade**, o qual é um produto da redefinição da lógica interna através de variáveis intermédias H_n.

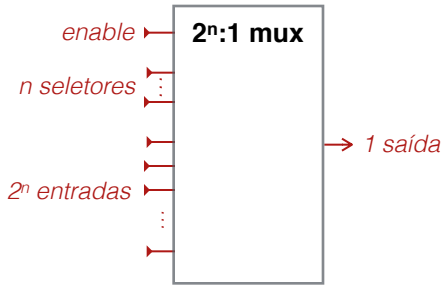
codificador de prioridade

Um codificador de prioridade conhecido é o 74x148, o qual tem lógica *active low*, *enable input*, *enable output* e *"got something"*.

Multiplexers e demultiplexers

multiplexers

Enquanto os codificadores transmitem uma mensagem traduzida, os **multiplexers** (abreviatura *mux*) são dispositivos que selecionam uma de várias entradas e a faz seguir numa só saída. Um multiplexer de 2^n entradas tem n linhas de seleção. Estes dispositivos são usados para aumentar a quantidade de dados a serem enviados numa rede num determinado intervalo de tempo e numa determinada largura de banda. Vejamos, assim, o seguinte multiplexer genérico, juntamente com uma tabela de verdade de um multiplexer 4:1.



inputs			output
EN	S1	S0	Y
0	x	x	0
1	0	0	I0
1	0	1	I1
1	1	0	I2
1	1	1	I3

figura 20
multiplexer genérico e tabela de verdade de um multiplexer 2:1

A função booleana que parte destes dispositivos é muito própria. Um multiplexer 2:1 tem a seguinte expressão:

$$Y = (I_0\overline{S}) + (I_1 S)$$

equação 42
equação de um mux 2:1

De uma forma mais geral, um *mux* tem como expressão:

$$Y = EN.\left(\sum_k m_k(S) I_k\right) \qquad k = 0,...,2^n - 1$$

equação 43
equação geral de um mux 2^n:1

onde $m_k(S)$ é o k-ésimo mintermo nas variáveis de seleção S_0, S_1, …, S_n.

demultiplexers

Por sua vez, os **demultiplexers** têm uma função inversa aos *mux's*. Estes, ao invés de partirem de 2^n entradas, partem de uma só, abrindo para 2^n saídas.

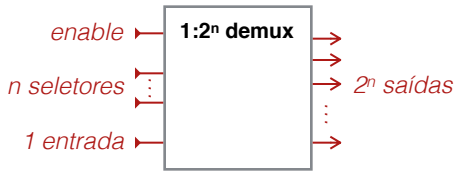


figura 21
demultiplexer genérico

A equação geral de um *demux* - abreviatura de demultiplexer - é a seguinte, considerando O_k como as nossas saídas e D_{in} como entrada:

$$O_k = EN.D_{in}.m_k(S) \qquad k = 0,...,2^n - 1$$

equação 44
equação geral de um demux 1:2^n

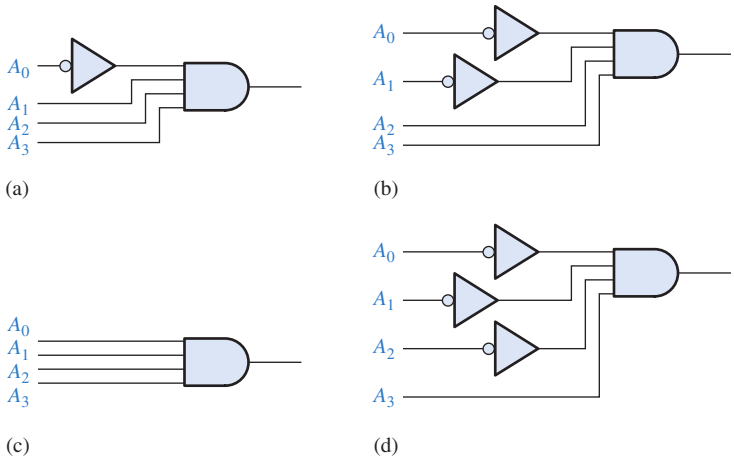
hierarquizar

É possível **hierarquizar** multiplexers. A chamada hierarquia de multiplexagem existe muitas vezes por causa da falta de maiores multiplexers. Assim, conseguimos,

Exercícios do Livro (Depois faço um ficheiro com mais)

SEÇÃO 6–5 Decodificadores

14. Quando um nível ALTO estiver presente em cada uma das portas de decodificação vistas na Figura 6–81, qual é o código binário que aparece nas entradas? A_3 é o MSB.



▶ FIGURA 6–81

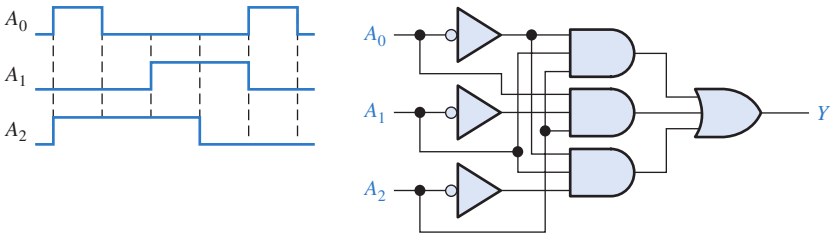
15. Mostre a lógica de decodificação para cada um dos seguintes códigos se uma saída ativa em nível ALTO for necessária:

(a) 1101 **(b)** 1000 **(c)** 11011 **(d)** 11100
(e) 101010 **(f)** 111110 **(g)** 000101 **(h)** 1110110

16. Resolva o Problema 15 sendo que uma saída ativa em nível BAIXO é necessária:

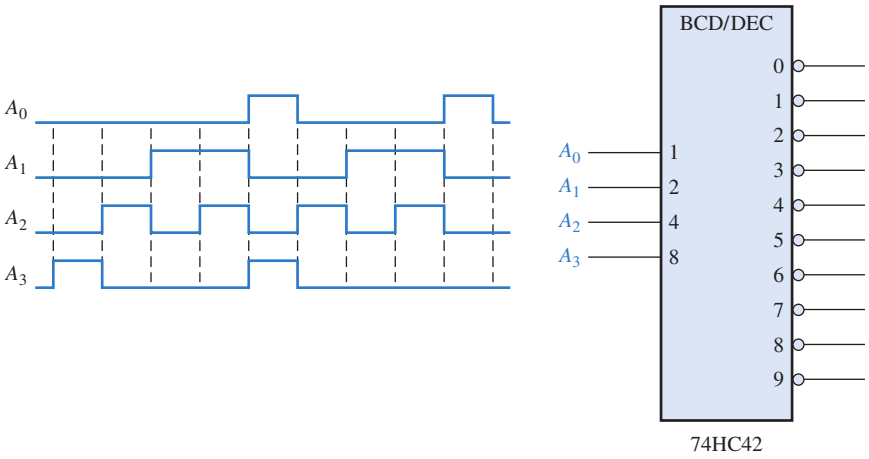
17. Você deseja apenas detectar a presença dos códigos 1010, 1100, 0001 e 1011. Uma saída ativa em nível ALTO é necessária para indicar a presença desses códigos. Desenvolva uma lógica de decodificação mínima com uma única saída que indique quando qualquer um desses códigos estiver nas entradas. Para qualquer outro código a saída tem que ser nível BAIXO.

18. Se as formas de onda de entrada são aplicadas na lógica de decodificação vista na Figura 6–82, determine a forma de onda de saída relacionando com as entradas.



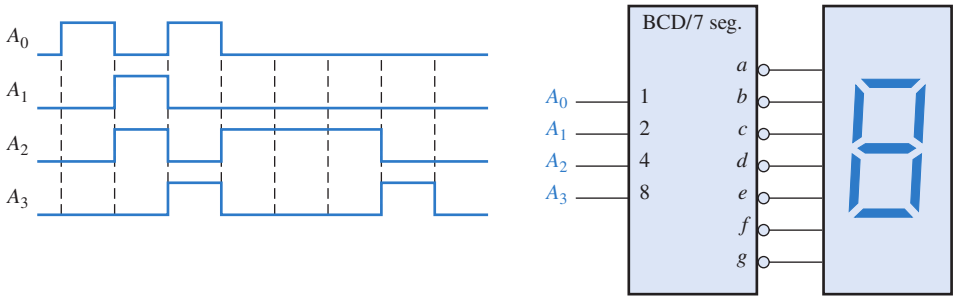
▶ FIGURA 6–82

19. Números BCD são aplicados seqüencialmente no decodificador de BCD para decimal conforme a Figura 6–83. Desenhe o diagrama de temporização mostrando cada saída devidamente relacionada com as outras e com as entradas.



▶ **FIGURA 6–83**

20. Um decodificador/driver de 7 segmentos aciona um display conforme mostra a Figura 6–84. Se as formas de onda são aplicadas conforme indicado, determine a seqüência de dígitos que aparecem no display.



▶ **FIGURA 6–84**

SEÇÃO 6–6 Codificadores

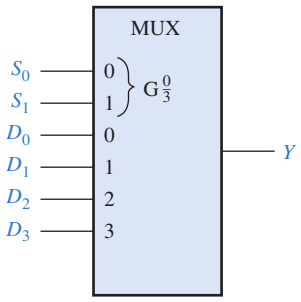
21. Para o codificador de decimal para BCD mostrado na Figura 6–38, considere que as entradas 9 e 3 sejam nível ALTO. Qual o código de saída? É um código BCD (8421) válido?

22. Um CI codificador 74HC147 tem nível BAIXO nos pinos 2, 5 e 12. Qual o código BCD que aparece nas saídas se todas as outras entradas estiverem em nível ALTO?

SEÇÃO 6–8 Multiplexadores (Seletores de Dados)

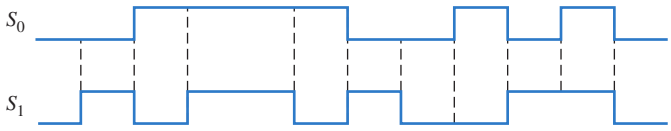
26. Para o multiplexador visto na Figura 6–85, determine a saída para os seguintes estados das entradas:

$D_0 = 0, D_1 = 1, D_2 = 1, D_3 = 0, S_0 = 1, S_1 = 0.$



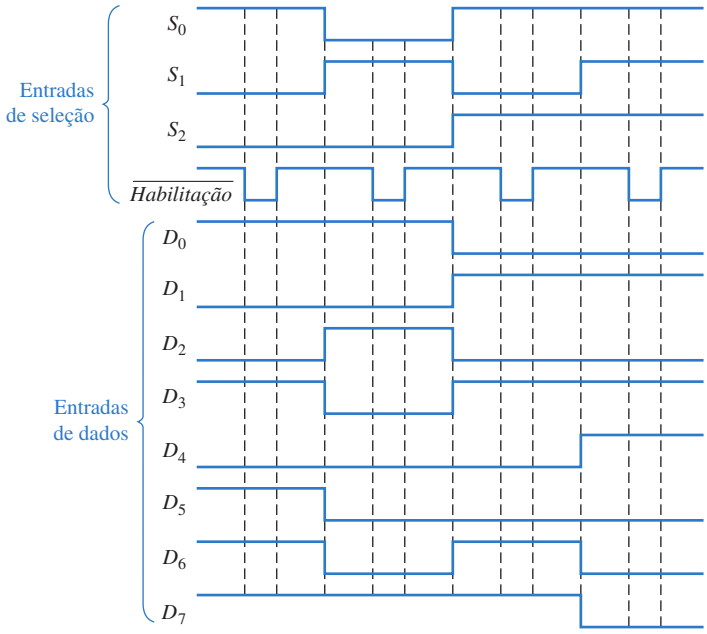
▶ FIGURA 6–85

27. Se as entradas de seleção de dados do multiplexador mostrado na Figura 6–85 são seqüenciadas como mostrado pelas formas de onda na Figura 6–86, determine a forma de onda de saída com as entradas de dados especificadas no Problema 26.



▶ FIGURA 6–86

28. As formas de onda vistas na Figura 6–87 são observadas nas entradas de um CI multiplexador de 8 entradas 74LS151. Desenhe a forma de onda da saída Y.



▶ FIGURA 6–87

SEÇÃO 6–9 Demultiplexadores

29. Desenvolva o diagrama de temporização total (entradas e saídas) para um 74HC154 usado numa aplicação de demultiplexação na qual as entradas são as seguintes: As entradas de seleção de dados recebem de forma repetitiva e em seqüência a saída de um contador que inicia com 0000 e a entrada de dados é um fluxo de dados em série que representa o número 2468 em BCD. O dígito menos significativo (8) é o primeiro na seqüência, sendo o primeiro bit o LSB, e ele deve aparecer nas posições dos 4 primeiros bits na saída.

Soluções, estão aqui exercícos que não eram então eu tirei, os números correspondem.

CAPÍTULO 6

1. **(a)** $A \oplus B = 0, \Sigma = 1, (A \oplus B)C_{ent.} = 0, AB = 1, C_{saída} = 1$
(b) $A \oplus B = 1, \Sigma = 0, (A \oplus B)C_{ent.} = 1, AB = 0, C_{saída} = 1$
(c) $A \oplus B = 1, \Sigma = 1, (A \oplus B)C_{ent.} = 0, AB = 0, C_{saída} = 0$

3. **(a)** $\Sigma = 1, C_{saída} = 0$;
(b) $\Sigma = 1, C_{saída} = 0$;
(c) $\Sigma = 0, C_{saída} = 1$;
(d) $\Sigma = 1, C_{saída} = 1$

5. 11100

7. $\Sigma_1 = 0110; \Sigma_2 = 1011; \Sigma_3 = 0110; \Sigma_4 = 0001; \Sigma_5 = 1000$

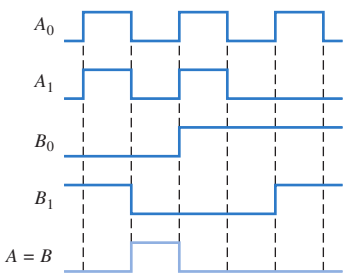
9. 225 ns

11. $A = B$ é nível ALTO quando $A_0 = B_0$ e $A_1 = B_1$; veja a Figura P–32.

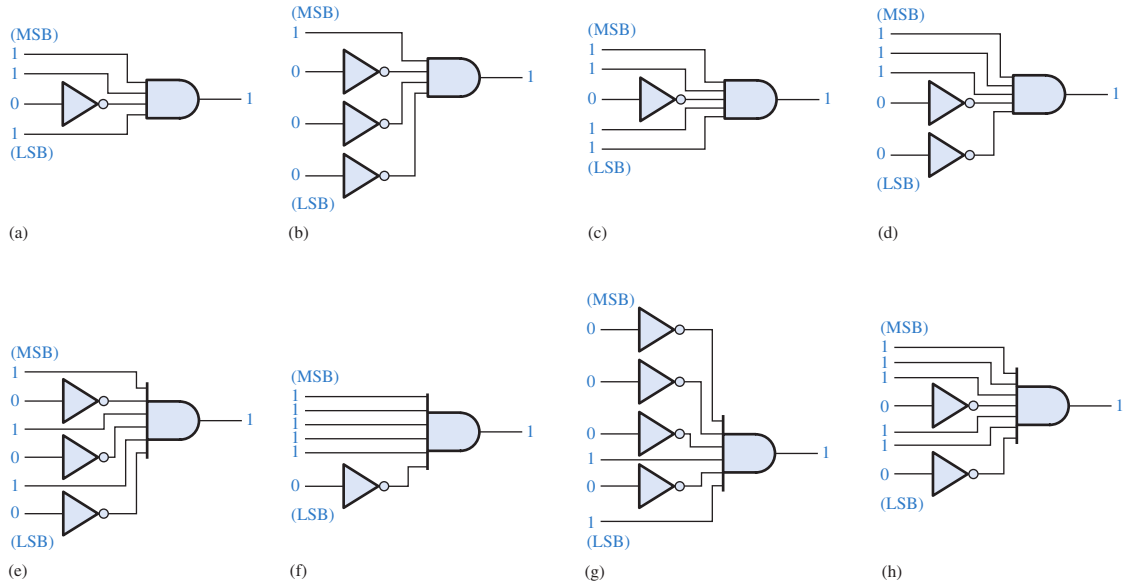
13. **(a)** $A > B = 1; A = B = 0; A < B = 0$
(b) $A < B = 1; A = B = 0; A > B = 0$
(c) $A = B = 1; A < B = 0; A > B = 0$

15. Veja a Figura P–33.

17. $X = A_3A_2\overline{A}_1\overline{A}_0 + \overline{A}_3\overline{A}_2\overline{A}_1A_0 + A_3\overline{A}_2A_1$

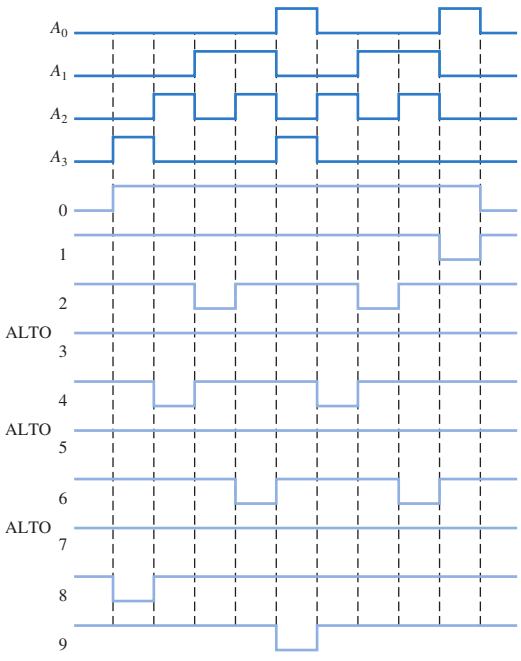


▲ FIGURA P–32



▲ **FIGURA P–33**

19. Veja a Figura P–34.



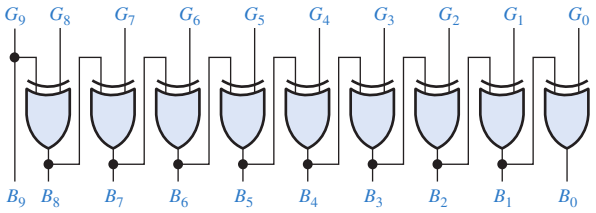
▲ **FIGURA P–34**

21. $A_3A_2A_1A_0 = 1011$, que é um código BCD inválido.

23. **(a)** $2 = 0010 = 0010_2$
(b) $8 = 1000 = 1000_2$
(c) $13 = 00010011 = 1101_2$
(d) $26 = 00100110 = 11010_2$
(e) $33 = 00110011 = 100001_2$

25. **(a)** 1010000000 Gray → 1100000000 binário
(b) 0011001100 Gray → 0010001000 binário
(c) 1111000111 Gray → 1010000101 binário
(d) 0000000001 Gray → 0000000001 binário

Veja a Figura P–35.



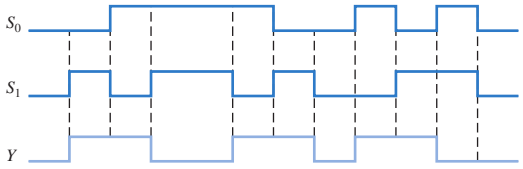
▲ **FIGURA P–35**

27. Veja a Figura P–36.

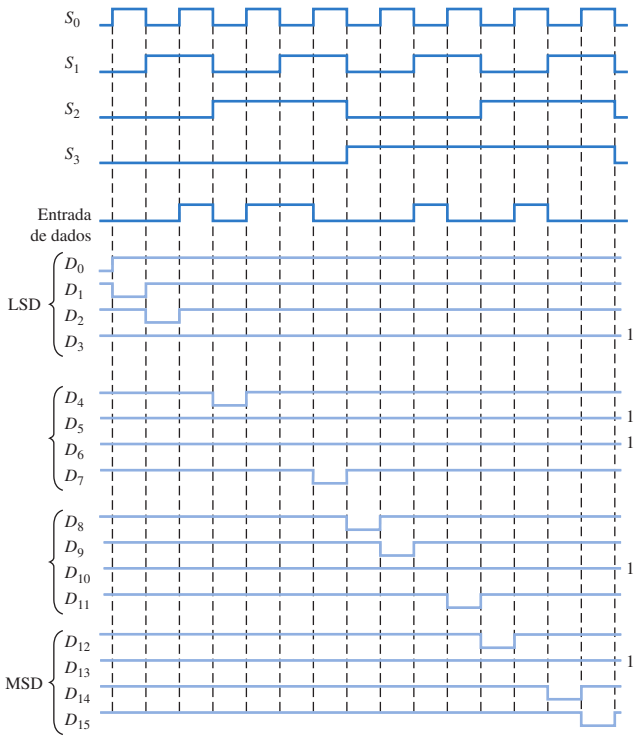
29. Veja a Figura P–37 na página 856.

31. Veja a Figura P–38.

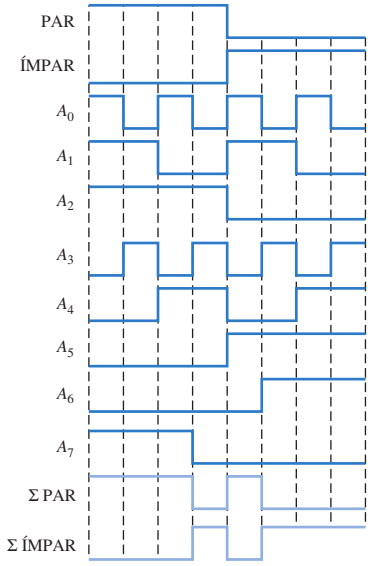
33. **(a)** OK
(b) segmento *g* "queimado"; saída G aberta
(c) a saída do segmento *b* presa em nível BAIXO.



▲ **FIGURA P–36**



▲ **FIGURA P–37**



▲ **FIGURA P–38**

35. (a) A entrada A_1 do somador superior está aberta: Todos os valores binários que correspondem aos números BCD 0, 1, 4, 5, 8 ou 9 terão um acréscimo de 2. Vemos isso primeiro com o valor BCD 0000 0000.

(b) A saída de carry do somador superior está aberta: Todos os valores que normalmente não envolvem um carry de saída terão um acréscimo de 32. Vemos isso primeiro com o valor BCD 0000 0000.

(c) A saída Σ_4 do somador superior está em curto-circuito com GND: Alguns valores binários acima de 15 terão uma diferença de 16. O primeiro valor BCD que indica isso é 0001 1000.

(d) A saída Σ_3 do somador inferior está em curto-circuito com GND: os outros 16 valores a partir do 16 terão uma diferença de 16. o primeiro valor BCD que indica isso é 0001 0110.

37. 1. Coloque um nível BAIXO no pino 7 (*Enable* – Habilitação).

2. Aplique um nível ALTO em D_0 e nível BAIXO de D_1 a D_7.

3. Coloque uma seqüência binária nas entradas de seleção e verifique as saídas e de acordo com a Tabela P–8.

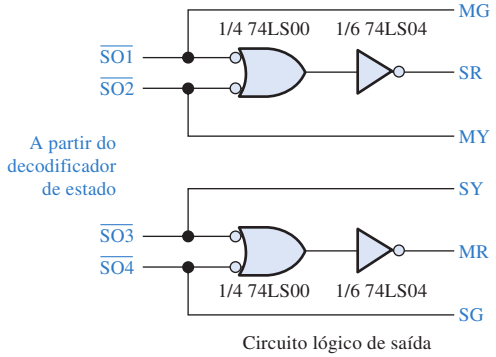
▼ **TABELA P–8**

S_2	S_1	S_0	Y	$\overline{Y}$
0	0	0	1	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

4. Repita a seqüência binária nas entradas de seleção para cada conjunto de dados de entrada listados na Tabela P–9. Um nível ALTO na saída Y deve ocorrer apenas para as combinações correspondentes das entradas de seleção mostradas.

39. Aplique um nível ALTO de cada vez na Entrada de dados, D_0 a D_7, com as entradas restante em nível BAIXO. Para cada nível ALTO aplicado a uma entrada de dados, coloque a seqüência das oito combinações binárias na entradas de seleção ($S_2S_1S_0$) e verifique se há nível ALTO na saída de dados correspondente e nível BAIXO nas outras saídas de dados.

41. Veja a Figura P–39.



▲ **FIGURA P–39**

TABELA P–9

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	Y	$\overline{Y}$	S_2	S_1	S_0
L	H	L	L	L	L	L	L	1	0	0	0	1
L	L	H	L	L	L	L	L	1	0	0	1	0
L	L	L	H	L	L	L	L	1	0	0	1	1
L	L	L	L	H	L	L	L	1	0	1	0	0
L	L	L	L	L	H	L	L	1	0	1	0	1
L	L	L	L	L	L	H	L	1	0	1	1	0
L	L	L	L	L	L	L	H	1	0	1	1	1

43. $\Sigma = \overline{A}\,\overline{B}C_{ent.} + \overline{A}B\overline{C}_{ent.} + A\overline{B}\,\overline{C}_{ent.} + ABC_{ent.}$

$C_{saída} = \overline{A}BC_{ent.} + A\overline{B}C_{ent.} + AB\overline{C}_{ent.} + ABC_{ent.}$

Veja a Figura P–40.

45. Veja o diagrama em bloco na Figura P–41 na página 858.

47. Veja a Figura P–42.

49. Veja a Figura P–43.

51. A saída de carry do somador LSB está aberta.

53. O pino 12 do 74148 superior está aberto.

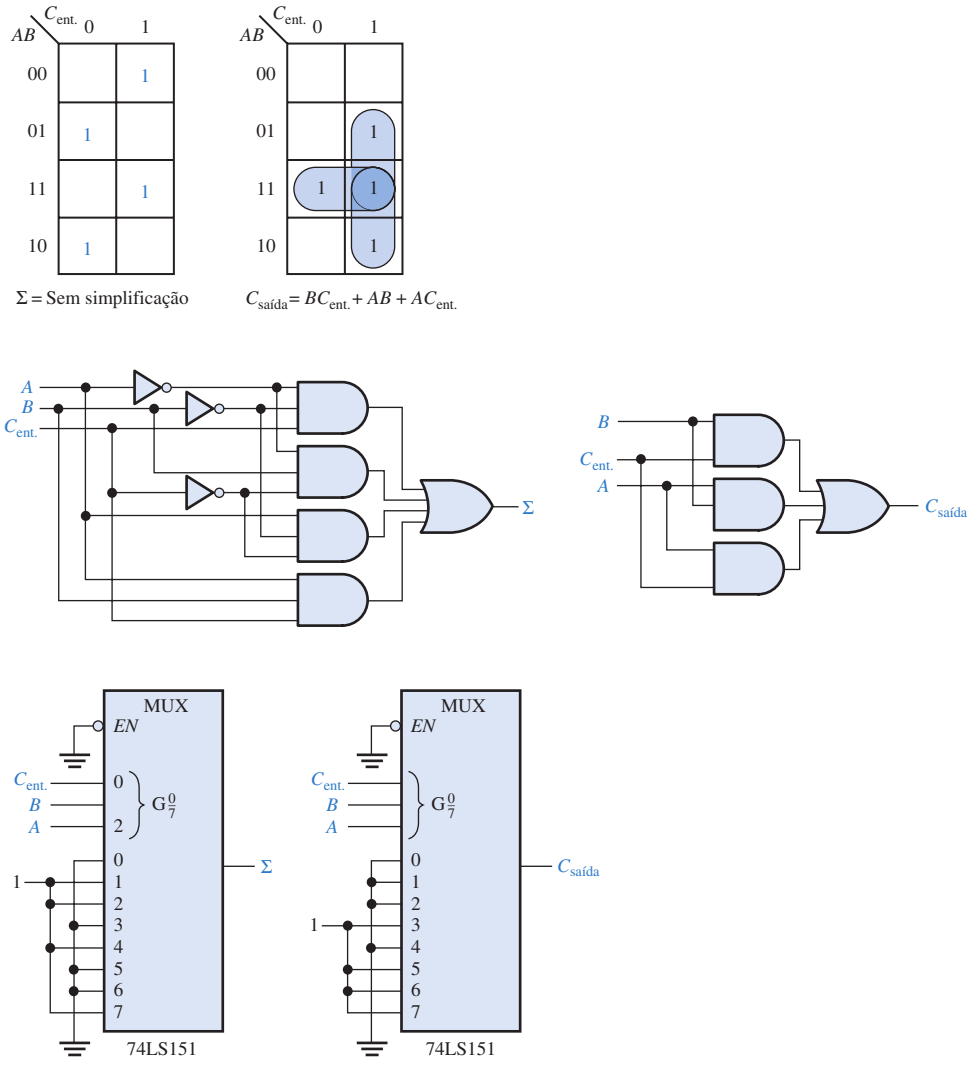
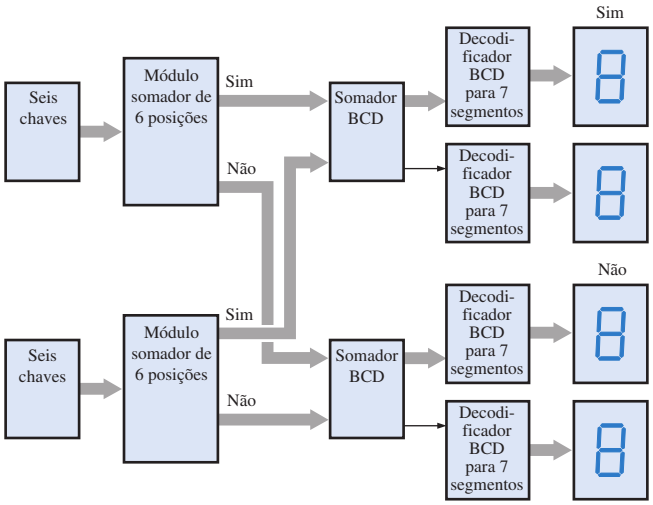
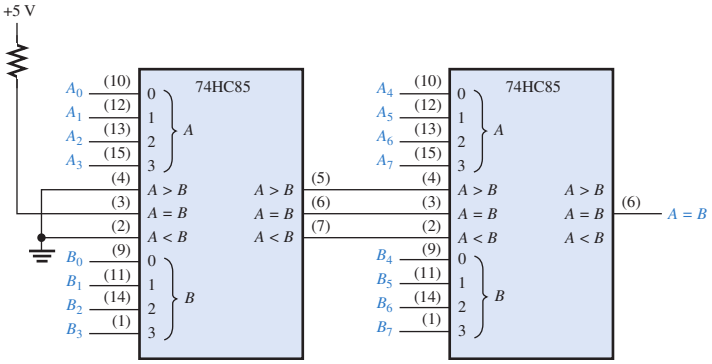


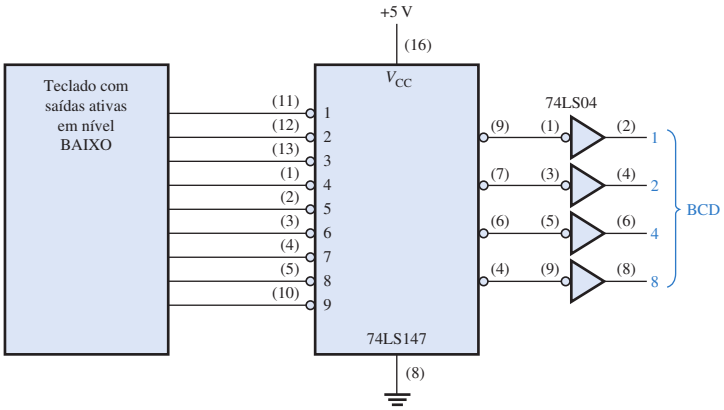
FIGURA P–40



▶ **FIGURA P–41**



▶ **FIGURA P–42**



▶ **FIGURA P–43**