Circuitos Lógicos Combinacionais

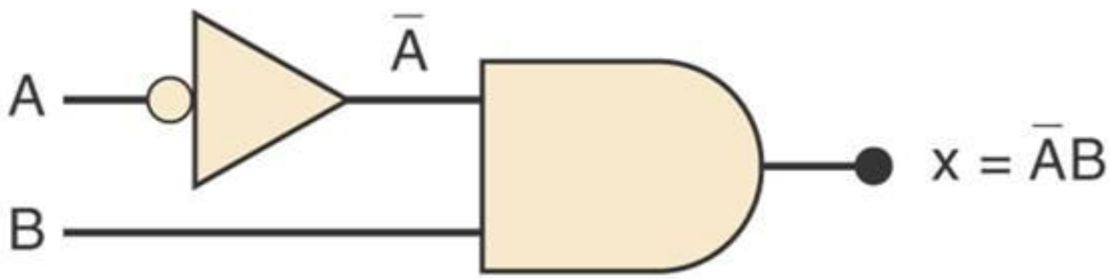
Sistemas de Informação
CPCX – UFMS
Slides: Prof. Renato F. dos Santos
Adaptação: Prof. Fernando Maia da Mota

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 então ser obtida**

A	B	x
0	0	0
0	1	1
1	0	0
1	1	0

(a)



(b)

4.4 Projetando circuitos lógicos combinacionais (Continuação)

- **Uma abordagem semelhante pode ser usada para outras condições de entrada**
- **Se *x* fosse nível alto apenas para condição *A* = 1 e *B* = 0, o circuito resultante seria uma porta AND com entradas *A* e $\overline{B}$**
- **Para qualquer uma das quatro condições possíveis de entrada podemos usar uma saída x em nível alto usando uma porta AND com entradas apropriadas (Figura 4.5)**

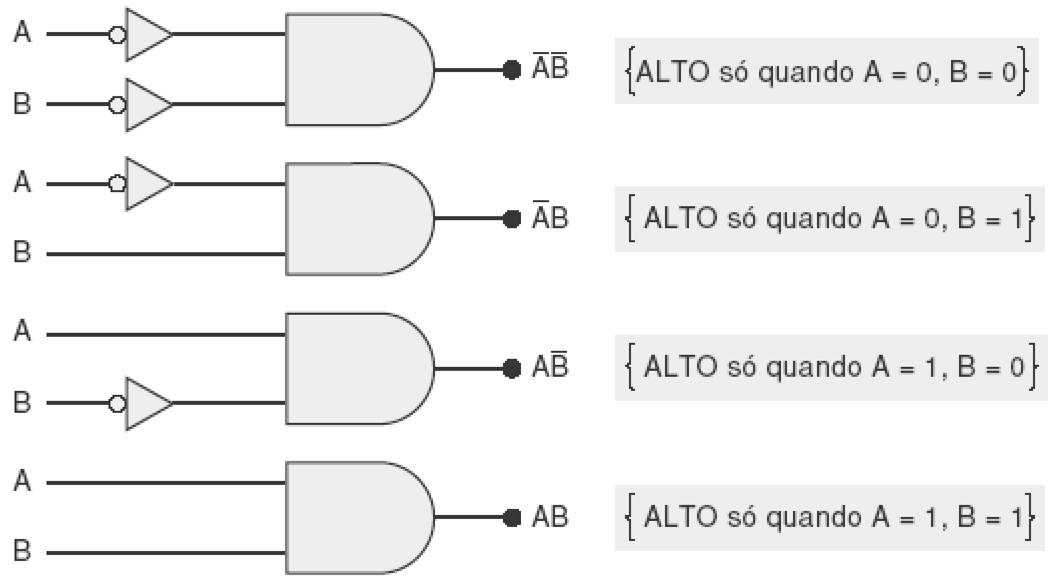


FIGURA 4.5
Uma porta AND, com entradas apropriadas, pode ser usada para gerar uma saída em nível 1 para um conjunto específico de níveis de entrada.

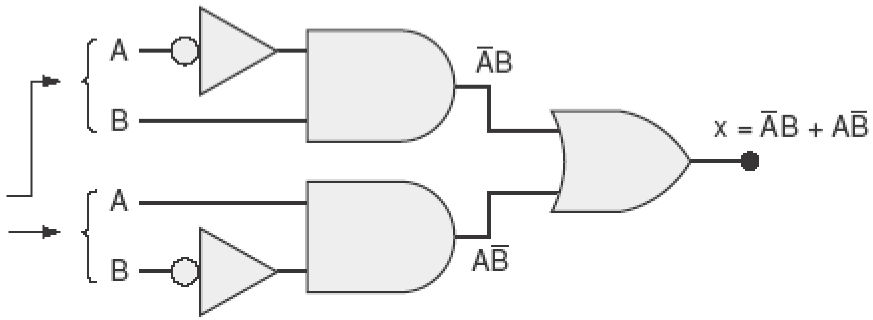
4.4 Projetando circuitos lógicos combinacionais (Continuação)

- **Na Figura 4.6(a), temos uma tabela-verdade que indica que a saída *x* será 1 para dois casos distintos:**
  - $A = 0, B = 1$ **e** $A = 1, B = 0$
- **Como isso pode ser implementado?**
- ***x* pode ser nível alto para uma ou outra condição**

FIGURA 4.6
Cada conjunto de condições de entrada, que gera uma saída em nível ALTO, é implementado por portas AND independentes. As saídas das portas AND são as entradas de uma OR que produz a saída final.

A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

(a)



(b)

4.4 Projetando circuitos lógicos combinacionais (Continuação)

- **Nesse exemplo, o termo AND é gerado para cada caso da tabela em que a saída *x* é nível 1**
- **As saídas das portas AND são entradas de uma OR, que produz a saída final *x*, que será nível 1 quando um outro termo da AND for nível 1**
- **Esse mesmo procedimento pode ser estendido para exemplos com mais de duas entradas**

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

TABELA 4.1

A	*B*	*C*	*X*	
0	0	0	0	
0	0	1	0	
0	1	0	1	$\longrightarrow \bar{A}B\bar{C}$
0	1	1	1	$\longrightarrow \bar{A}BC$
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	1	$\longrightarrow ABC$

4.4 Projetando circuitos lógicos combinacionais (Continuação)

- **Nessa tabela há três casos em que a saída *x* é nível 1**
- **O termo AND requerido para cada um dos casos é mostrado**
- **Para cada caso em que a variável é 0, a mesma aparece invertida no termo AND**
- **A expressão na forma de soma-de-produtos para a saída *x* é obtida fazendo a operação OR dos três termos AND**

Procedimento completo de projeto

– **Qualquer problema lógico pode ser resolvido, usando o seguinte procedimento passo a passo:**

1. **Interprete o problema e construa uma tabela-verdade para descrever o seu funcionamento**
2. **Escreva o termo AND (produto) para cada caso em que a saída seja 1**
3. **Escreva a expressão da soma-de-produtos para a saída**
4. **Simplifique a expressão de saída, se possível**
5. **Implemente o circuito para a expressão final, simplificada**

Exemplo 4.7

– **Projete um circuito lógico com três entradas, A, B e C, cuja saída será nível ALTO apenas quando a maioria das entradas for nível ALTO.**

Solução

Passo 1. Construa a tabela-verdade.

Com base no enunciado do problema, a saída x deve ser nível 1 sempre que duas ou mais entradas forem nível 1; para todos os outros casos, a saída deve ser nível 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	$\rightarrow \overline{A}BC$
1	0	0	0	
1	0	1	1	$\rightarrow A\overline{B}C$
1	1	0	1	$\rightarrow AB\overline{C}$
1	1	1	1	$\rightarrow ABC$

Solução (Continuação)

Passo 2. Escreva o termo AND (produto) para cada caso em que a saída seja 1.

Existem quatro desses casos. Os termos AND são mostrados junto à tabela-verdade (Tabela 4.2). Observe, novamente, que cada termo AND contém cada uma das variáveis de entrada em sal forma invertida ou não-invertida.

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

$$X = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

Solução (Continuação)

Passo 4. Simplifique a expressão de saída.

Essa expressão pode ser simplificada de várias formas. Talvez a forma mais rápida seja perceber que o último termo ABC tem duas variáveis em comum com cada um dos outros termos. Assim, podemos usar o termos ABC para fatorar cada um dos termos. A expressão é reescrita coma o termo ABC aparecendo três vezes (lembre-se do Exemplo 4.2, que atestou que essa operação é permitida na álgebra booleana):

$$X = \overline{A}BC + ABC + A\overline{B}C + ABC + AB\overline{C} + ABC$$

Fatorando apropriadamente os pares de termos, obtemos

$$X = BC(\overline{A} + A) + AC(\overline{B} + B) + AB(\overline{C} + C)$$

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

$$X = BC + AC + AB$$

Solução (Continuação)

Passo 5. Implemente o circuito para a expressão final.

Essa expressão está implementada na Figura 4.7. Visto que a expressão está na forma de soma-de-produtos, o circuito consiste em um grupo de portas AND ligadas em uma única porta OR.