

DCC638 - Introdução à Lógica Computacional
2020/01

Álgebra Booleana e Circuitos Lógicos

Área de Teoria DCC/UFMG

Introdução

Álgebra Booleana: Introdução

- Circuitos operam com entradas e saídas **binárias**, i.e. com valores 0 e 1.
- Isto facilita a construção de circuitos
 - Qualquer dispositivo que diferencie dois estados (ligado/deligado, alta voltagem/baixa voltagem, etc.)
- Normalmente nos referimos a cada um dos estados binários como **bits** (do inglês “*binary digits*”), ou **valores Booleanos**.
- Em 1938 Claude Shannon demonstrou que as regras básicas da lógica, introduzidas por George Boole em 1854 no seu livro “*The Laws of Thought*”, podem ser usadas para projetar circuitos.
- Estas regras formam a base da **lógica Booleana**, que é o que vamos estudar aqui.

Álgebra Booleana: Introdução

- Nesta parte do curso vamos fazer a conexão da lógica proposicional e álgebra Booleana com os circuitos de computação.

- Para isto, vamos estudar os fundamentos da álgebra Booleana, incluindo:
 - Como representar números em base binária.
 - Como representar funções Booleanas.
 - Como projetar circuitos que implementem funções.
 - Como minimizar circuitos para obter implementações eficientes.

Representação de Números em Base Binária

Representação de números em base decimal

- O nosso sistema numérico é um sistema baseado em potências de 10.
 - Números representados como somas ponderadas de potências de 10, usando dígitos decimais 0, 1, 2, ..., 9 como pesos.

- Exemplo 1 O número decimal 237 pode ser decomposto como:

$$237_{10} = 2 \cdot 10^2 + 3 \cdot 10^1 + 7 \cdot 10^0$$

- Não há nada de especial na escolha de potências de 10 para decompor números.
- Podemos usar qualquer inteiro positivo como uma base numérica.
- Notação: n_b indica que o número n está representado na base b .

Representação de números em base binária

- Podemos representar números naturais em potências de 2, por exemplo.
- Somas ponderadas de potências de 2, usando dígitos binários 0, 1 como pesos.
- Exemplo 2

$$\begin{aligned}11101101_2 &= 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + \\ &\quad 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\ &= 1 \cdot 128 + 1 \cdot 64 + 1 \cdot 32 + 0 \cdot 16 + \\ &\quad 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 \\ &= 237_{10}\end{aligned}$$



Convertendo de base binária para decimal

- Formalmente, um número binário de k bits ($b_i \in \{0, 1\}$, para $k - 1 \geq i \geq 0$)

$$n_2 = b_{k-1} b_{k-2} \dots b_2 b_1 b_0$$

pode ser convertido para seu equivalente decimal m_{10} pela fórmula

$$m_{10} = \sum_{i=0}^{k-1} b_i \cdot 2^i .$$

Convertendo de base binária para decimal

- Um método rápido para converter um número binário para seu equivalente decimal é dado no exemplo a seguir.
- Exemplo 3 Encontre o decimal equivalente a 110101_2 .

Convertendo de base binária para decimal

- Um método rápido para converter um número binário para seu equivalente decimal é dado no exemplo a seguir.
- **Exemplo 3** Encontre o decimal equivalente a 110101_2 .

Solução. O equivalente ao binário pedido é o decimal 53, como mostra a tabela abaixo.

Posição i	5	4	3	2	1	0	
Dígito b_i	1	1	0	1	0	1	
Equivalente decimal: 2^i	32	16	8	4	2	1	
Contribuição $b_i \cdot 2^i$	32	16	0	4	0	1	53



Convertendo de base decimal para binária

- No caso geral, processo de conversão de um decimal m_{10} para seu equivalente binário n_2 se dá da seguinte forma:
 1. Realize a divisão inteira do número decimal m_{10} por 2 repetidas vezes, guardando o resto r_i de cada passo i .
 2. Pare quando o resultado da divisão for 0.
 3. Produza como resultado n_2 a concatenação dos restos r_i na ordem contrária em que foram encontrados.

Convertendo de base decimal para binária

- Exemplo 4 Converta 77 de decimal para seu equivalente binário.

Convertendo de base decimal para binária

- Exemplo 4 Converta 77 de decimal para seu equivalente binário.

Solução. Podemos seguir o seguinte processo:

$$\begin{array}{ccccccc} 77 & \xrightarrow[\text{resto}=1]{\text{div}2} & 38 & \xrightarrow[\text{resto}=0]{\text{div}2} & 19 & \xrightarrow[\text{resto}=1]{\text{div}2} & 9 & \xrightarrow[\text{resto}=1]{\text{div}2} \\ 4 & \xrightarrow[\text{resto}=0]{\text{div}2} & 2 & \xrightarrow[\text{resto}=0]{\text{div}2} & 1 & \xrightarrow[\text{resto}=1]{\text{div}2} & 0 & \end{array}$$

A representação binária é obtida invertendo-se a ordem dos restos produzidos:

$$77 = 1001101_2.$$



Convertendo de base decimal para binária

- Exemplo 5 Converta 237 de decimal para seu equivalente binário.

Solução. Podemos seguir o seguinte processo:

$$\begin{array}{cccccccc} 237 & \xrightarrow[\text{resto}=1]{\text{div}2} & 118 & \xrightarrow[\text{resto}=0]{\text{div}2} & 59 & \xrightarrow[\text{resto}=1]{\text{div}2} & 29 & \xrightarrow[\text{resto}=1]{\text{div}2} \\ 14 & \xrightarrow[\text{resto}=0]{\text{div}2} & 7 & \xrightarrow[\text{resto}=1]{\text{div}2} & 3 & \xrightarrow[\text{resto}=1]{\text{div}2} & 1 & \xrightarrow[\text{resto}=1]{\text{div}2} & 0 \end{array}$$

A representação binária é obtida invertendo-se a ordem dos restos produzidos:

$$237 = 11101101_2.$$



Somando dois números em base binária

- Podemos realizar operações usuais em números binários.
- Em particular, a adição binária é feita de forma análoga à decimal:
 - Nos decimais, ao somar o dígito 7 com o dígito 5, por exemplo, o resultado é o dígito 2, e sobra 1 como excedente para a próxima coluna de dígitos.
Isso ocorre sempre que o resultado ultrapassa o maior dígito decimal, que é 9.
 - Com os binários, isso ocorre quando o maior dígito binário, 1, é ultrapassado.
- Exemplo 6 Compute o valor da soma $1101_2 + 100_2$.

Funções Booleanas

Funções Booleanas: Introdução

- A **álgebra Booleana** fornece as operações e as regras para trabalharmos com o conjunto $\{0, 1\}$.

- Em particular, circuitos eletrônicos podem ser estudados usando este conjunto binário e as regras da álgebra Booleana.

Funções Booleanas: Operações Booleanas

- As três operações em álgebra Booleana que mais vamos usar são:

- O **complemento**, denotada por uma barra $\bar{}$, definido como $\bar{x} = 1 - x$:

$$\bar{0} = 1, \quad \text{e} \quad \bar{1} = 0.$$

- O **produto Booleano**, denotado por \cdot ou *AND*, e definido como $x \cdot y = \min(x, y)$:

$$0 \cdot 0 = 0, \quad 0 \cdot 1 = 0, \quad 1 \cdot 0 = 0, \quad 1 \cdot 1 = 1.$$

- A **soma Booleana**, denotada por $+$ ou *OR*, e definida como $x + y = \max(x, y)$:

$$0 + 0 = 0, \quad 0 + 1 = 1, \quad 1 + 0 = 1, \quad 1 + 1 = 1.$$

- Quando não há risco de ambiguidade, podemos omitir o símbolo \cdot de produto Booleano, escrevendo ab no lugar de $a \cdot b$.

Funções Booleanas: Operações Booleanas

- A ordem de precedência de operadores Booleanos é:
 - 1 primeiro avaliam-se complementos;
 - 2 em seguida avaliam-se produtos;
 - 3 por fim avaliam-se somas.
- Exemplo 7 Ache o valor de $1 \cdot 0 + \overline{(0 + 1)}$.

Funções Booleanas: Operações Booleanas

- A ordem de precedência de operadores Booleanos é:

- 1 primeiro avaliam-se complementos;
- 2 em seguida avaliam-se produtos;
- 3 por fim avaliam-se somas.

- Exemplo 7 Ache o valor de $1 \cdot 0 + \overline{(0 + 1)}$.

Solução.

$$\begin{aligned}1 \cdot 0 + \overline{(0 + 1)} &= 1 \cdot 0 + \bar{1} \\ &= 1 \cdot 0 + 0 \\ &= 0 + 0 \\ &= 0\end{aligned}$$



Funções Booleanas: Correspondência com lógica proposicional

- Existe uma correspondência entre a álgebra Booleana e a lógica proposicional no seguinte sentido:
 - O complemento $\bar{\quad}$ corresponde ao operador de negação \neg .
 - A soma Booleana $+$ corresponde ao operador de disjunção \vee .
 - O produto Booleano \cdot corresponde ao operador de conjunção \wedge .
 - O valor 0 corresponde ao valor de verdade F (falso).
 - O valor 1 corresponde ao valor de verdade T (verdadeiro).
- Por este motivo, igualdades em álgebra Booleana podem ser traduzidas imediatamente em equivalências proposicionais, e vice-versa.

Funções Booleanas: Correspondência com lógica proposicional

- **Exemplo 8** Converta a igualdade Booleana $1 \cdot 0 + \overline{(0 + 1)} = 0$ na equivalência proposicional correspondente.

Funções Booleanas: Correspondência com lógica proposicional

- **Exemplo 8** Converta a igualdade Booleana $1 \cdot 0 + \overline{(0 + 1)} = 0$ na equivalência proposicional correspondente.

Solução. A fórmula Booleana $1 \cdot 0 + \overline{(0 + 1)} = 0$ corresponde à equivalência proposicional

$$T \wedge F \vee \neg(F \vee T) \equiv F .$$

- **Exemplo 9** Converta a equivalência proposicional $(T \wedge T) \vee \neg F \equiv T$ na igualdade Booleana correspondente.

Funções Booleanas: Correspondência com lógica proposicional

- Exemplo 8 Converta a igualdade Booleana $1 \cdot 0 + \overline{(0 + 1)} = 0$ na equivalência proposicional correspondente.

Solução. A fórmula Booleana $1 \cdot 0 + \overline{(0 + 1)} = 0$ corresponde à equivalência proposicional

$$T \wedge F \vee \neg(F \vee T) \equiv F .$$

- Exemplo 9 Converta a equivalência proposicional $(T \wedge T) \vee \neg F \equiv T$ na igualdade Booleana correspondente.

Solução. A equivalência proposicional $(T \wedge T) \vee \neg F \equiv T$ corresponde à fórmula Booleana

$$(1 \cdot 1) + \bar{0} = 1 .$$

Expressões Booleanas e Funções Booleanas

- Seja $B = \{0, 1\}$ o **conjunto de valores Booleanos**.

Então

$$B^n = \{(x_1, x_2, \dots, x_n) \mid x_i \in B \text{ para } 1 \leq i \leq n\}$$

é o **conjunto de todas as n -tuplas formadas por 0s e 1s**.

- Uma variável x que assume um valor em B é chamada de **variável Booleana**.
- Uma função de B^n para B é chamada de **função Booleana de grau n** .
- Exemplo 10 A função Booleana $F(x, y) = x\bar{y}$ de tipo $B^2 \rightarrow B$ é representada na tabela abaixo.

x	y	$F(x, y) = x\bar{y}$
0	0	0
0	1	0
1	0	1
1	1	0



Expressões Booleanas e Funções Booleanas

- Funções Booleanas podem ser representadas por expressões formadas por variáveis Booleanas e operações Booleanas.
- As **expressões Booleanas** sobre as variáveis x_1, x_2, \dots, x_n são definidas recursivamente como:
 1. $0, 1, x_1, x_2, \dots, x_n$ são expressões Booleanas.
 2. Se E_1 e E_2 são expressões Booleanas, então também são expressões Booleanas:
 - a) $\neg E_1$,
 - b) $(E_1 + E_2)$,
 - c) $(E_1 \cdot E_2)$.
- Cada expressão Booleana representa uma função Booleana.

(Mais para frente vamos ver também a afirmação conversas: cada função Booleana pode ser representada por uma expressão Booleana.)
- Para computar o valor da função, basta substituir cada variável pelo seu valor (0 ou 1).

Expressões Booleanas e Funções Booleanas

- Exemplo 11 Determine os valores da função Booleana $F(x, y, z) = xy + \bar{z}$.

Solução. A tabela abaixo especifica o comportamento da função desejada.

x	y	z	xy	\bar{z}	$F(x, y, z) = xy + \bar{z}$
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	1	0	1



Expressões Booleanas e Funções Booleanas

- Duas funções Booleanas F e G de n variáveis são **iguais** se, e somente se:

$$F(b_1, b_2, \dots, b_n) = G(b_1, b_2, \dots, b_n)$$

para todos os valores $b_1, b_2, \dots, b_n \in B$.

- Duas expressões Booleanas que representam a mesma função são chamadas de **equivalentes**.

Por exemplo, as três expressões abaixo são todas equivalentes:

① xy

② $xy + 0$

③ $xy \cdot 1$

Expressões Booleanas e Funções Booleanas

- Sejam F e G funções Booleanas de grau n . Podemos definir:

- O **complemento** de F é a função \bar{F} , definida como

$$\bar{F}(b_1, b_2, \dots, b_n) = \overline{F(b_1, b_2, \dots, b_n)}$$

- A **soma Booleana** de F e G é a função $F + G$, definida como

$$(F + G)(b_1, b_2, \dots, b_n) = F(b_1, b_2, \dots, b_n) + G(b_1, b_2, \dots, b_n)$$

- O **produto Booleano** de F e G é a função $F \cdot G$, definida como

$$(F \cdot G)(b_1, b_2, \dots, b_n) = F(b_1, b_2, \dots, b_n) \cdot G(b_1, b_2, \dots, b_n)$$

Expressões Booleanas e Funções Booleanas

- Note que uma função Booleana de grau 2 possui

- como domínio: um conjunto de quatro elementos:

$$\{ (0,0), (0,1), (1,0), (1,1) \} ,$$

- como contra-domínio: um conjunto de dois elementos:

$$\{0,1\} .$$

- Portanto, o número de funções Booleanas distintas de grau 2 é $2^4 = 16$.

Expressões Booleanas e Funções Booleanas

- Note que uma função Booleana de grau 2 possui

- como domínio: um conjunto de quatro elementos:

$$\{ (0,0), (0,1), (1,0), (1,1) \} ,$$

- como contra-domínio: um conjunto de dois elementos:

$$\{0,1\} .$$

- Portanto, o número de funções Booleanas distintas de grau 2 é $2^4 = 16$.
 - 2 saídas possíveis pra cada entrada

Expressões Booleanas e Funções Booleanas

- Exemplo 12 Enumere todas as 16 possíveis funções Booleanas de grau 2.

Solução. Todas as possíveis funções Booleanas F_0, F_1, \dots, F_{15} de grau 2 estão representadas na tabela abaixo.

x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
0	0	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0

x	y	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1



Expressões Booleanas e Funções Booleanas

- Note que no caso geral, uma função Booleana de grau n possui
 - como domínio: um conjunto de 2^n elementos,
 - como contra-domínio: um conjunto de dois elementos: $\{0, 1\}$.

Portanto, o número de funções Booleanas distintas de grau n é 2^{2^n} .

- Exemplo 13 Número de funções Booleanas de grau n :

Grau = n	Número de funções = 2^{2^n}
1	4
2	16
3	256
4	65 536
5	4 294 967 296
6	18 446 744 073 709 551 616



Identidades Booleanas

- Existem muitas **identidades** na álgebra Booleana.
Essas identidades são úteis na simplificação do projeto de circuitos.
- Identidades Booleanas podem ser verificadas usando tabelas de valores.
- Exemplo 14 Verifique a identidade Booleana de que $x(y + z) = xy + xz$.

Identidades Booleanas

- Existem muitas **identidades** na álgebra Booleana.
Essas identidades são úteis na simplificação do projeto de circuitos.
- Identidades Booleanas podem ser verificadas usando tabelas de valores.
- Exemplo 14** Verifique a identidade Booleana de que $x(y + z) = xy + xz$.

Solução. Podemos verificar a identidade usando a tabela abaixo.

x	y	z	$y + z$	$x(y + z)$	xy	xz	$xy + xz$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Note que esta identidade Booleana corresponde à equivalência lógica

$$x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z),$$

que é uma das leis de distributividade. •

Identidades Booleanas

- Algumas identidades Booleanas importantes:

Nome	Identidade
Lei da dupla complementação	$\overline{\overline{x}} = x$
Leis de idempotência	$x + x = x$ $x \cdot x = x$
Leis de identidade	$x + 0 = x$ $x \cdot 1 = x$
Leis de dominância	$x + 1 = 1$ $x \cdot 0 = 0$
Leis de comutatividade	$x + y = y + x$ $xy = yx$
Leis de associatividade	$x + (y + z) = (x + y) + z$ $x(yz) = (xy)z$
Leis de distributividade	$x + yz = (x + y)(x + z)$ $x(y + z) = xy + xz$
Leis de De Morgan	$\overline{xy} = \overline{x} + \overline{y}$ $\overline{x + y} = \overline{x} \overline{y}$
Leis de absorção	$x + xy = x$ $x(x + y) = x$
Propriedade da unidade	$x + \overline{x} = 1$
Propriedade do zero	$x \cdot \overline{x} = 0$

Identities Booleanas

- É possível verificar identidades Booleanas usando identidades já verificadas.
- **Exemplo 15** Verifique a lei de absorção $x(x + y) = x$ usando identidades Booleanas dadas na tabela anterior.

Solução.

$$x(x + y) \equiv$$

Identities Booleanas

- É possível verificar identidades Booleanas usando identidades já verificadas.
- **Exemplo 15** Verifique a lei de absorção $x(x + y) = x$ usando identidades Booleanas dadas na tabela anterior.

Solução.

$$x(x + y) \equiv$$

Identities Booleanas

- É possível verificar identidades Booleanas usando identidades já verificadas.
- Exemplo 15 Verifique a lei de absorção $x(x + y) = x$ usando identidades Booleanas dadas na tabela anterior.

Solução.

$$x(x + y) \equiv (x + 0)(x + y) \quad (\text{lei de identidade para a soma})$$

Identities Booleanas

- É possível verificar identidades Booleanas usando identidades já verificadas.
- **Exemplo 15** Verifique a lei de absorção $x(x + y) = x$ usando identidades Booleanas dadas na tabela anterior.

Solução.

$$\begin{aligned}x(x + y) &\equiv (x + 0)(x + y) && \text{(lei de identidade para a soma)} \\ &\equiv x + (0 \cdot y) && \text{(distributividade da soma sobre o produto)}\end{aligned}$$

Identities Booleanas

- É possível verificar identidades Booleanas usando identidades já verificadas.
- **Exemplo 15** Verifique a lei de absorção $x(x + y) = x$ usando identidades Booleanas dadas na tabela anterior.

Solução.

$$\begin{aligned}x(x + y) &\equiv (x + 0)(x + y) && \text{(lei de identidade para a soma)} \\ &\equiv x + (0 \cdot y) && \text{(distributividade da soma sobre o produto)} \\ &\equiv x + y \cdot 0 && \text{(comutatividade do produto)}\end{aligned}$$

Identities Booleanas

- É possível verificar identidades Booleanas usando identidades já verificadas.
- **Exemplo 15** Verifique a lei de absorção $x(x + y) = x$ usando identidades Booleanas dadas na tabela anterior.

Solução.

$$\begin{aligned}x(x + y) &\equiv (x + 0)(x + y) && \text{(lei de identidade para a soma)} \\ &\equiv x + (0 \cdot y) && \text{(distributividade da soma sobre o produto)} \\ &\equiv x + y \cdot 0 && \text{(comutatividade do produto)} \\ &\equiv x + 0 && \text{(lei de dominância para o produto)}\end{aligned}$$

Identities Booleanas

- É possível verificar identidades Booleanas usando identidades já verificadas.
- **Exemplo 15** Verifique a lei de absorção $x(x + y) = x$ usando identidades Booleanas dadas na tabela anterior.

Solução.

$$\begin{aligned}x(x + y) &\equiv (x + 0)(x + y) && \text{(lei de identidade para a soma)} \\ &\equiv x + (0 \cdot y) && \text{(distributividade da soma sobre o produto)} \\ &\equiv x + y \cdot 0 && \text{(comutatividade do produto)} \\ &\equiv x + 0 && \text{(lei de dominância para o produto)} \\ &\equiv x && \text{(lei de identidade para a soma)}\end{aligned}$$



Dualidade

- As identidades da tabela anterior vêm em pares (exceto pelas leis da dupla complementação, da unidade, e do zero).
- Para explicar a relação entre as duas identidades em cada par nós usamos o conceito **dualidade**.
- O **dual** de uma expressão Booleana é obtido pelo intercâmbio entre:
 - somas Booleanas e produtos Booleanos,
 - valores 0 e valores 1.
- Exemplo 16 Encontre os duais das expressões abaixo.
 - O dual de $x(y + 0)$ é: $x + (y \cdot 1)$.
 - O dual de $\bar{x} \cdot 1 + (\bar{y} + z)$ é: $(\bar{x} + 0)(\bar{y}z)$



Dualidade: O princípio da dualidade

- O **dual de uma função Booleana** F representada por uma expressão Booleana é a função F^d representada pelo dual desta expressão correspondente.

A função dual F^d não depende da expressão Booleana particular usada para representar F .

- Um resultado importante é o **princípio da dualidade**:

“Uma identidade entre funções representadas por expressões Booleanas permanece válida quando ambos os lados da igualdade são substituídos por seus duais.”

- O princípio da dualidade é particularmente útil para derivar novas identidades.
- Exemplo 17 Construa uma identidade a partir de $x(x + y) = x$ (uma das leis de absorção) usando o princípio da dualidade.

Dualidade: O princípio da dualidade

- O **dual de uma função Booleana** F representada por uma expressão Booleana é a função F^d representada pelo dual desta expressão correspondente.

A função dual F^d não depende da expressão Booleana particular usada para representar F .

- Um resultado importante é o **princípio da dualidade**:

“Uma identidade entre funções representadas por expressões Booleanas permanece válida quando ambos os lados da igualdade são substituídos por seus duais.”

- O princípio da dualidade é particularmente útil para derivar novas identidades.

- Exemplo 17 Construa uma identidade a partir de $x(x + y) = x$ (uma das leis de absorção) usando o princípio da dualidade.

Solução. Tomando duais em ambos lados da identidade, chegamos à nova identidade $x + xy = x$ (que é a outra lei de absorção). ●

Definição abstrata de álgebra Booleana

- Note que os resultados que estabelecemos podem ser traduzidos:
 - em resultados sobre lógica proposicional (onde complemento, $+$, \cdot , 0 , 1 , correspondem a, respectivamente, a negação, \vee , \wedge , F , T);
 - em resultados sobre teoria de conjuntos (onde complemento, $+$, \cdot , 0 , 1 , correspondem a, respectivamente, a complemento de conjuntos, \cup , \cap , \emptyset , conjunto universo U).

(Verifique você mesmo(a) que as identidades Booleanas correspondem às identidades de conjuntos!)
- Formalizamos estes princípios como uma **álgebra Booleana abstrata**.
- Ao mostrar que uma estrutura (e.g., lógica proposicional, teoria de conjuntos, etc.) é uma álgebra Booleana, então os resultados estabelecidos sobre álgebras Booleanas em geral podem ser aplicados a esta estrutura particular!

Definição abstrata de álgebra Booleana

- Uma **álgebra Booleana** é um conjunto B com duas operações binárias \wedge e \vee , elementos 0 e 1 , e uma operação unária $\bar{}$ tais que as seguintes propriedades sejam válidas para todo x, y , e z em B :

$$\text{Leis de identidade: } \begin{cases} x \vee 0 = x \\ x \wedge 1 = x \end{cases}$$

$$\text{Leis de complemento: } \begin{cases} x \vee \bar{x} = 1 \\ x \wedge \bar{x} = 0 \end{cases}$$

$$\text{Leis de associatividade: } \begin{cases} (x \vee y) \vee z = x \vee (y \vee z) \\ (x \wedge y) \wedge z = x \wedge (y \wedge z) \end{cases}$$

$$\text{Leis de comutatividade: } \begin{cases} x \vee y = y \vee x \\ x \wedge y = y \wedge x \end{cases}$$

$$\text{Leis de distributividade: } \begin{cases} x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z) \\ x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z) \end{cases}$$

Representação de Funções Booleanas

Representação de Funções Booleanas: Introdução

- Aqui vamos atacar dois problemas importantes da álgebra Booleana:
 1. Como encontrar uma expressão Booleana que represente uma função pretendida?
 - Veremos como qualquer função Booleana pode ser representada por expressões apenas com soma, produto, complemento, variáveis e valores Booleanos.
 2. Existe um conjunto menor de operadores que pode ser usado para representar todas as funções Booleanas?
 - Veremos que todas as funções Booleanas podem ser representadas usando apenas um operador!
- Ambos os problemas têm enorme importância prática no design de circuitos.

Forma normal disjuntiva

- Uma maneira de representar qualquer função Booleana
- Facilmente computável a partir da tabela de valores da função
- **Exemplo 18** Encontre uma expressão Booleana para a função F dada na tabela abaixo.

x	y	z	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Forma normal disjuntiva

- Exemplo 18 (Continuação)

Solução. Primeiramente, note que para representar a função F precisamos de uma expressão Booleana que:

- assuma o valor 1 quando $x = 1$, $y = 0$, e $z = 1$, e
- assuma o valor 0 em caso contrário.

Esta expressão pode ser formada tomando o produto Booleano de x , \bar{y} e z , pois

$$x\bar{y}z = \begin{cases} 1, & \text{se } x = 1, y = 0 \text{ e } z = 1 \\ 0, & \text{em caso contrário} \end{cases}$$

Assim, chegamos à conclusão de que uma expressão para F é

$$F = x\bar{y}z .$$



Forma normal disjuntiva

- Exemplo 19 Encontre uma expressão Booleana para as a função G dada na tabela abaixo.

x	y	z	G
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Forma normal disjuntiva

- Exemplo 19 Encontre uma expressão Booleana para a função G dada na tabela abaixo.

x	y	z	G
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Solução. Primeiramente, note que para representar a função G precisamos de uma expressão Booleana que:

- assuma o valor 1 quando: $x = 0, y = 1$ e $z = 0$, ou $x = 1, y = 1$ e $z = 0$.
- assuma o valor 0 em caso contrário.

Forma normal disjuntiva

- Exemplo 19 (Continuação)

Esta expressão pode ser formada tomando a soma Booleana de dois produtos Booleanos, pois

$$\bar{x}y\bar{z} = \begin{cases} 1, & \text{se } x = z = 1 \text{ e } y = 0 \\ 0, & \text{em caso contrário} \end{cases}$$

e

$$xy\bar{z} = \begin{cases} 1, & \text{se } x = z = 1 \text{ e } y = 0 \\ 0, & \text{em caso contrário} \end{cases}$$

Assim, chegamos à conclusão de que uma expressão para G é

$$G = \bar{x}y\bar{z} + xy\bar{z}$$



Forma normal disjuntiva

- Cada combinação de valores das variáveis para as quais a função tem o valor 1 leva a um produto Booleano das variáveis ou seus complementos.
 - Como generalizar?
- Um **literal** é uma variável Booleana ou seu complemento.
- Um **mintermo** das variáveis Booleanas x_1, x_2, \dots, x_n é um produto Booleano y_1, y_2, \dots, y_n em que para cada i temos $y_i = x_i$ ou $y_i = \bar{x}_i$.
(Um mintermo é um produto de n literais, com um literal para cada variável.)
- Exemplo 20 Encontre um mintermo que seja igual a 1 se $x_1 = x_3 = 0$ e $x_2 = x_4 = x_5 = 1$.

Forma normal disjuntiva

- Cada combinação de valores das variáveis para as quais a função tem o valor 1 leva a um produto Booleano das variáveis ou seus complementos.
 - Como generalizar?
- Um **literal** é uma variável Booleana ou seu complemento.
- Um **mintermo** das variáveis Booleanas x_1, x_2, \dots, x_n é um produto Booleano y_1, y_2, \dots, y_n em que para cada i temos $y_i = x_i$ ou $y_i = \bar{x}_i$.
(Um mintermo é um produto de n literais, com um literal para cada variável.)
- **Exemplo 20** Encontre um mintermo que seja igual a 1 se $x_1 = x_3 = 0$ e $x_2 = x_4 = x_5 = 1$.

Solução. O mintermo é $\bar{x}_1 x_2 \bar{x}_3 x_4 x_5$.



Forma normal disjuntiva

- Dada uma tabela de valores para uma função Booleana F de grau n :
 1. Tome o mintermo de cada combinação de variáveis que fazem F igual a 1.
 2. Defina F como a soma Booleana destes mintermos.

Esta soma dos mintermos é a **forma normal disjuntiva** ou **expansão em soma de produtos** da função Booleana.

- Exemplo 21 Encontre a forma normal disjuntiva da função Booleana \oplus (XOR) definida na tabela abaixo.

x_1	x_2	$x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0

Forma normal disjuntiva

- Dada uma tabela de valores para uma função Booleana F de grau n :
 1. Tome o mintermo de cada combinação de variáveis que fazem F igual a 1.
 2. Defina F como a soma Booleana destes mintermos.

Esta soma dos mintermos é a **forma normal disjuntiva** ou **expansão em soma de produtos** da função Booleana.

- Exemplo 21 Encontre a forma normal disjuntiva da função Booleana \oplus (*XOR*) definida na tabela abaixo.

x_1	x_2	$x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0

Solução. A função $x_1 \oplus x_2$ assume valor 1 quando:

- $x_1 = 0$ e $x_2 = 1$ (mintermo $\bar{x}_1 x_2$); ou
- $x_1 = 1$ e $x_2 = 0$ (mintermo $x_1 \bar{x}_2$).

Logo $x_1 \oplus x_2 = \bar{x}_1 x_2 + x_1 \bar{x}_2$.

•

Forma normal disjuntiva

- Exemplo 22 Encontre a forma normal disjuntiva da função
 $F(x, y, z) = (x + y)\bar{z}$.

Forma normal disjuntiva

- Exemplo 22 Encontre a forma normal disjuntiva da função $F(x, y, z) = (x + y)\bar{z}$.

Solução. Vamos resolver esta questão de duas maneiras.

A primeira maneira é construir a tabela do comportamento de F e derivar dela a forma normal conjuntiva.

x	y	z	$x + y$	\bar{z}	$(x + y)\bar{z}$
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	1	0	0
1	0	0	1	1	1
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	0	0

Forma normal disjuntiva

- Exemplo 22 Encontre a forma normal disjuntiva da função $F(x, y, z) = (x + y)\bar{z}$.

Solução. Vamos resolver esta questão de duas maneiras.

A primeira maneira é construir a tabela do comportamento de F e derivar dela a forma normal conjuntiva.

x	y	z	$x + y$	\bar{z}	$(x + y)\bar{z}$
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	1	0	0
1	0	0	1	1	1
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	0	0

A função F assume valor 1 quando:

- $x = z = 0$ e $y = 1$
(mintermo $\bar{x}y\bar{z}$); ou
- $x = 1$ e $y = z = 0$
(mintermo $x\bar{y}\bar{z}$); ou
- $x = y = 1$ e $z = 0$
(mintermo $xy\bar{z}$).

Logo $F(x, y, z) = \bar{x}y\bar{z} + x\bar{y}\bar{z} + xy\bar{z}$.

Forma normal disjuntiva

- Exemplo 22 (Continuação)

A segunda maneira é usar identidades Booleanas para expandir o produto que representa a função F , e depois simplificá-lo.

$$\begin{aligned} F(x, y, z) &= (x + y) \bar{z} \\ &= x \bar{z} + y \bar{z} && \text{(Lei da distributividade)} \\ &= x 1 \bar{z} + 1 y \bar{z} && \text{(Lei da identidade)} \\ &= x (y + \bar{y}) \bar{z} + (x + \bar{x}) y \bar{z} && \text{(Lei da unidade)} \\ &= x y \bar{z} + x \bar{y} \bar{z} + x y \bar{z} + \bar{x} y \bar{z} && \text{(Lei da distributividade)} \\ &= x y \bar{z} + x \bar{y} \bar{z} + \bar{x} y \bar{z} && \text{(Lei de idempotência)} \end{aligned}$$



Completude de Operadores

- Toda função Booleana pode ser expressa em forma normal disjuntiva
 - (Como você demonstraria isso?)

- Na forma normal disjuntiva utilizamos os operadores Booleanos de:

- complemento $\bar{}$,
- soma $+$, e
- produto \cdot .

- Logo o conjunto $\{\bar{}, +, \cdot\}$ é **funcionalmente completo**.

- Isto leva à pergunta:

“É possível encontrar um conjunto menor de operadores que seja funcionalmente completo?”

Completude de Operadores

- Podemos eliminar um dos operadores se ele poder ser expresso por outros.
- Em particular, podemos eliminar a soma Booleana, notando que:

$$x + y =$$

Completude de Operadores

- Podemos eliminar um dos operadores se ele poder ser expresso por outros.
- Em particular, podemos eliminar a soma Booleana, notando que:

$$x + y = \overline{\overline{x + y}} \quad (\text{Lei da dupla complementação})$$

=

Completude de Operadores

- Podemos eliminar um dos operadores se ele poder ser expresso por outros.
- Em particular, podemos eliminar a soma Booleana, notando que:

$$\begin{aligned}x + y &= \overline{\overline{x + y}} && \text{(Lei da dupla complementação)} \\ &= \overline{\overline{x} \overline{y}} && \text{(De Morgan)}\end{aligned}$$

Isso significa que $\{\overline{}, \cdot\}$ também é funcionalmente completo.

Completude de Operadores

- Podemos eliminar um dos operadores se ele poder ser expresso por outros.
- Em particular, podemos eliminar a soma Booleana, notando que:

$$\begin{aligned}x + y &= \overline{\overline{x + y}} && \text{(Lei da dupla complementação)} \\ &= \overline{\overline{x} \overline{y}} && \text{(De Morgan)}\end{aligned}$$

Isso significa que $\{\overline{}, \cdot\}$ também é funcionalmente completo.

- De forma similar, podemos eliminar o produto Booleana, notando que:

$$xy =$$

Completude de Operadores

- Podemos eliminar um dos operadores se ele poder ser expresso por outros.
- Em particular, podemos eliminar a soma Booleana, notando que:

$$\begin{aligned}x + y &= \overline{\overline{x + y}} && \text{(Lei da dupla complementação)} \\ &= \overline{\overline{x} \overline{y}} && \text{(De Morgan)}\end{aligned}$$

Isso significa que $\{\overline{}, \cdot\}$ também é funcionalmente completo.

- De forma similar, podemos eliminar o produto Booleana, notando que:

$$\begin{aligned}xy &= \overline{\overline{xy}} && \text{(Lei da dupla complementação)} \\ &= && \end{aligned}$$

Completude de Operadores

- Podemos eliminar um dos operadores se ele poder ser expresso por outros.
- Em particular, podemos eliminar a soma Booleana, notando que:

$$\begin{aligned}x + y &= \overline{\overline{x + y}} && \text{(Lei da dupla complementação)} \\ &= \overline{\overline{x} \overline{y}} && \text{(De Morgan)}\end{aligned}$$

Isso significa que $\{\overline{}, \cdot\}$ também é funcionalmente completo.

- De forma similar, podemos eliminar o produto Booleana, notando que:

$$\begin{aligned}xy &= \overline{\overline{xy}} && \text{(Lei da dupla complementação)} \\ &= \overline{\overline{x} + \overline{y}} && \text{(De Morgan)}\end{aligned}$$

Isso significa que $\{\overline{}, +\}$ também é funcionalmente completo.

Completude de Operadores

- O operador **NAND**, denotado por $|$, é definido como

$$0 | 0 = 1, \quad 0 | 1 = 1, \quad 1 | 0 = 1, \quad 1 | 1 = 0.$$

- O conjunto $\{|\}$ é funcionalmente completo pois $\{\bar{}, \cdot\}$ também o é e *NAND* pode representar complemento e produtos Booleanos.
- Primeiro, note que $\bar{x} = x | x$, como mostra a tabela abaixo.
- Segundo, note que $xy = (x | y) | (x | y)$, como mostra a tabela abaixo.

x	\bar{x}	$x x$
0	1	1
1	0	0

x	y	xy	$x y$	$(x y) (x y)$
0	0	0	1	0
0	1	0	1	0
1	0	0	1	0
1	1	1	0	1

Portas Lógicas

Portas lógicas: Introdução

- A álgebra Booleana pode ser usada para modelar circuitos eletrônicos:
 - Entradas e saídas de um circuito podem ser consideradas 0 ou 1.
 - Computadores e dispositivos eletrônicos em geral são compostos por vários destes circuitos.
- Os elementos básicos dos circuitos são chamadas **portas lógicas**.
 - Cada porta lógica implementa uma operação Booleana.
- Os circuitos que vamos estudar não têm capacidade de memória
 - Comportamento funcional: saídas dependem apenas das entradas dadas
 - Chamamos tais dispositivos de **circuitos combinatórios**.

Portas lógicas

- Vamos nos concentrar em três tipos de portas lógicas.

- A **porta NOT** (ou **inversor**) que implementa a operação de complementação:



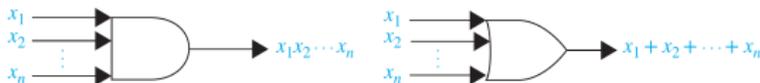
- A **porta OR**, que implementa a operação de soma Booleana:



- A **porta AND**, que implementa a operação de produto Booleano:



- É possível considerar portas com várias entradas:

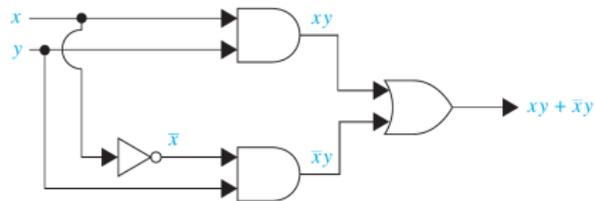
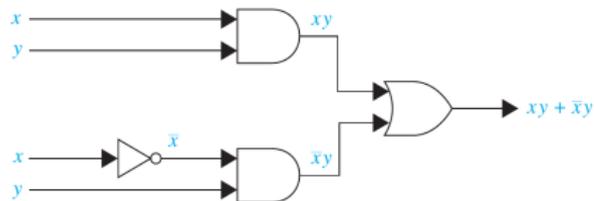


Circuitos combinatórios

- Circuitos combinatórios podem ser construídos usando uma combinação de inversores, portas OR e portas AND.
- Quando são formadas combinações de circuitos, algumas portas podem compartilhar entradas.

Existem maneiras diferentes de representar isto graficamente.

- **Exemplo 23** As figuras abaixo são representações distintas do mesmo circuito combinatório que produz a saída $xy + \bar{x}y$, em que algumas portas compartilham entradas.

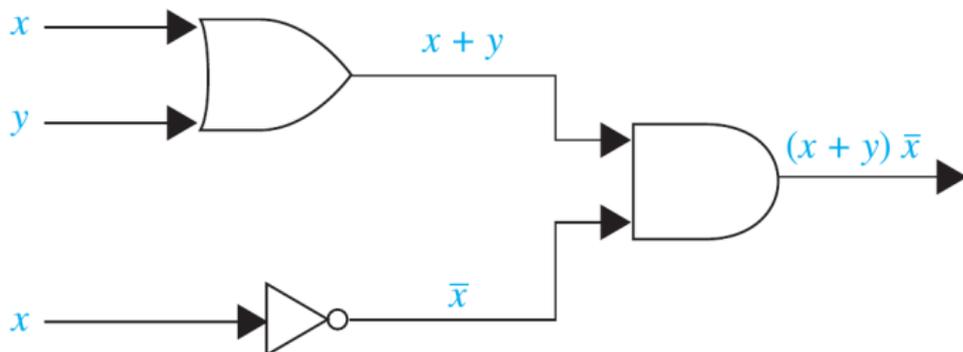


- Exemplo 24 Projete um circuito que compute a função Booleana $(x + y)\bar{x}$.

Circuitos combinatórios

- Exemplo 24 Projete um circuito que compute a função Booleana $(x + y)\bar{x}$.

Solução. O circuito desejado está desenhado a seguir.



Exemplos de circuitos: Votação majoritária

- **Exemplo 25** Um comitê de três indivíduos decide questões para uma organização da seguinte forma:
 1. Cada indivíduo vota sim ou não para cada proposta que surgir.
 2. Uma proposta é aprovada se receber pelo menos dois votos sim.

Projete um circuito que determine se uma proposta é ou não aprovada.

Solução. Começamos por modelar as entradas do sistema de votação. Sejam:

- $x = 1$ se o primeiro indivíduo votar sim, e $x = 0$ se este indivíduo votar não;
- $y = 1$ se o segundo indivíduo votar sim, e $y = 0$ se este indivíduo votar não;
- $z = 1$ se o terceiro indivíduo vota sim e $z = 0$ se este indivíduo votar não.

Então um circuito deve ser projetado que produz a saída 1 a partir das entradas x , y e z quando dois ou mais de x , y , e z são 1.

Exemplos de circuitos: Votação majoritária

- Exemplo 25 (Continuação)

Para obter uma expressão Booleana que expresse a função desejada, vamos especificar a tabela da função de votação majoritária.

x	y	z	Votação majoritária
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Podemos derivar a forma normal disjuntiva da expressão desejada:

$$\bar{x}y z + x\bar{y}z + xy\bar{z} + xyz$$

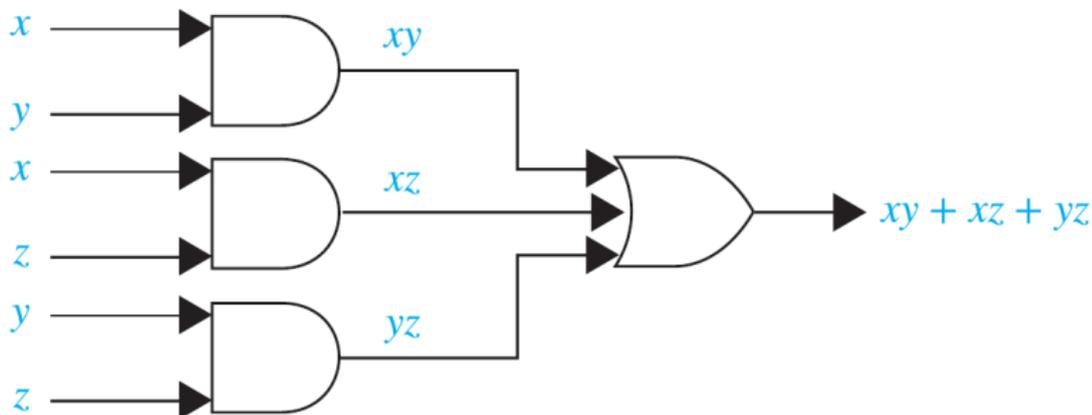
que podemos mostrar ser equivalente a

$$xy + xz + yz .$$

Exemplos de circuitos: Votação majoritária

- Exemplo 25 (Continuação)

A votação majoritária é modelada, portanto, pela expressão $xy + xz + yz$, que pode ser implementada no circuito a seguir.



Exemplos de circuitos: Somadores

- **Exemplo 26** Neste exemplo vamos mostrar como circuitos lógicos podem ser usados para realizar a adição de inteiros positivos em forma binária.

Vamos construir o circuito para fazer essa adição a partir de alguns circuitos componentes.

Primeiro, vamos construir um circuito **meio somador**, que adiciona dois bits sem considerar o “vai um” de uma adição anterior.

Tal circuito tem:

- Como entradas: dois bits x e y ; e
- Como saídas: dois bits s e c , onde s é o bit da soma e c é o bit do “vai um”.

Circuitos assim são chamados de **circuitos de saída múltipla**, pois têm mais de um saída.

Exemplos de circuitos: Somadores

- Exemplo 26 (Continuação)

A tabela a seguir demonstra o comportamento desejado do meio somador.

Entradas		Saídas	
x	y	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Podemos derivar as expressões:

$$s = \bar{x}y + x\bar{y}$$

$$c = xy$$

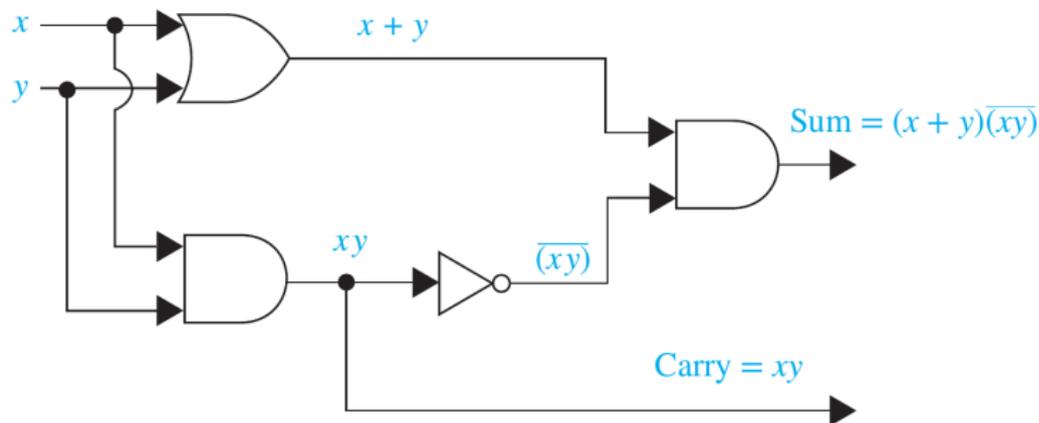
Por conveniência, vamos transformar a expressão de s em uma equivalente antes de implementar o circuito:

$$\begin{aligned} s &= \bar{x}y + x\bar{y} && \text{(Exp. original)} \\ &= (\bar{x}y + 0) + (0 + x\bar{y}) && \text{(Identidade)} \\ &= (\bar{x}y + y\bar{y}) + (x\bar{x} + x\bar{y}) && \text{(Prop. do zero)} \\ &= (x + y)(\bar{x} + \bar{y}) && \text{(Distrib.)} \\ &= (x + y)\overline{(xy)} && \text{(De Morgan)} \end{aligned}$$

Exemplos de circuitos: Somadores

- Exemplo 26 (Continuação)

Assim, um circuito que implementa o meio somador é o seguinte:



Exemplos de circuitos: Somadores

- Exemplo 26 (Continuação)

Agora, para completar o somador, vamos projetar um **somador completo**, que considera o “vai um” de uma soma de bits anterior.

Este circuito tem:

- Como entradas: dois bits x e y , além de um bit de “vai um” c_i
- Como saídas: dois bits s e c , onde s é o bit da soma e c_{i+1} é o bit do “vai um”.

Exemplos de circuitos: Somadores

- Exemplo 26 (Continuação)

A tabela a seguir demonstra o comportamento desejado somador completo.

Entradas			Saídas	
x	y	c_i	s	c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

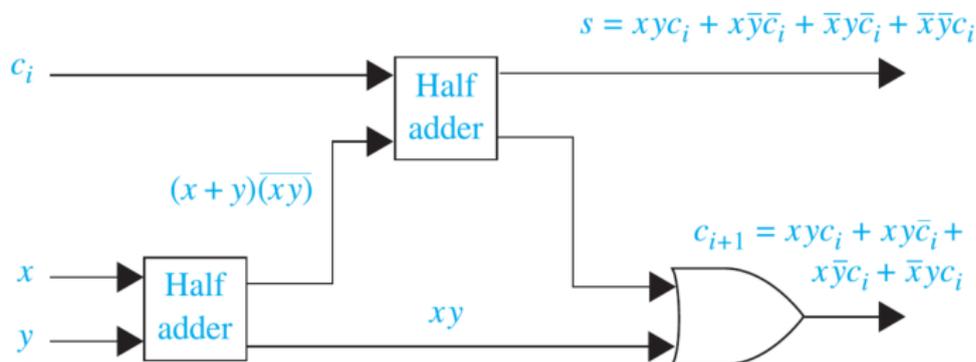
Podemos derivar as expressões:

$$s = \bar{x} \bar{y} c_i + \bar{x} y \bar{c}_i + x \bar{y} \bar{c}_i + x y c_i$$

$$c_{i+1} = \bar{x} y c_i + x \bar{y} c_i + x y \bar{c}_i + x y c_i$$

Exemplos de circuitos: Somadores

- O circuito do somador completo encontra-se abaixo.



Note que poderíamos ter construído o somador completo diretamente a partir das fórmulas derivadas para s e c_{i+1} , mas optamos por uma forma equivalente usando meio-somadores, que é um componente que já projetamos. (Desafio: você pode checar que a implementação é equivalente usando tabelas da verdade!)

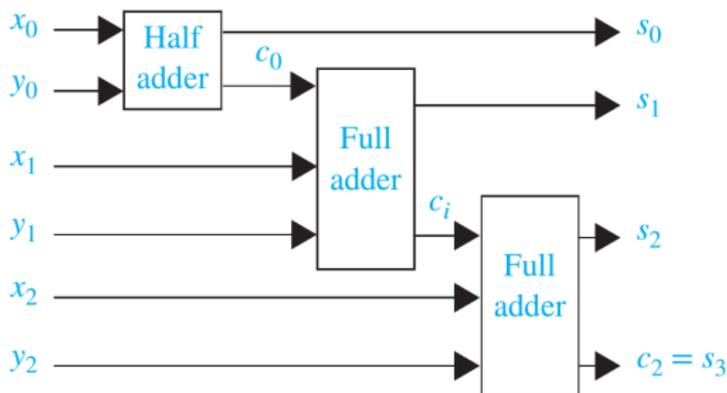
A prática de construir circuitos maiores usando componentes já desenhados é muito comum no projeto de circuitos reais.

Exemplos de circuitos: Somadores

- Exemplo 26 (Continuação)

Para finalizar o exemplo, vemos um circuito para somar dois inteiros de 3 bits cada, usando um meio somador e dois somadores completos.

$$\begin{array}{rcccccc} & (c_2) & (c_1) & (c_0) & & \text{("vai um")} \\ & & x_2 & x_1 & x_0 & \text{(1ª parcela)} \\ + & & y_2 & y_1 & y_0 & \text{(2ª parcela)} \\ \hline s_3 & s_2 & s_1 & s_0 & & \text{(resultado)} \end{array}$$



Minimização de Circuitos

Minimização de Circuitos: Introdução

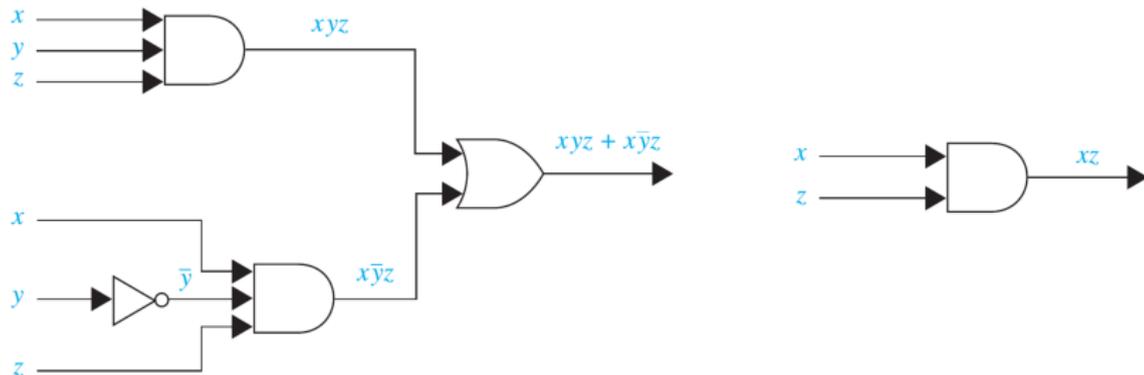
- A eficiência de um circuito combinatório depende do número e da disposição de suas portas.
- O processo de projetar um circuito combinatório começa com a tabela especificando a saída para cada combinação de valores de entrada.
- Podemos sempre usar a forma normal disjuntiva (DNF) pra implementar um circuito.
 - No entanto ela pode conter mais termos do que o necessário.
- Veremos como simplificar expressões Booleanas para minimizar circuitos que as implementam.

Minimização de Circuitos: Introdução

- Exemplo 27 Considere a seguinte simplificação de uma expressão Booleana:

$$\begin{aligned}xyz + x\bar{y}z &= (y + \bar{y})xz && \text{(distributividade)} \\ &= 1xz && \text{(unidade)} \\ &= xz && \text{(identidade)}\end{aligned}$$

Note que a implementação de um circuito para a mesma função pode ser muito mais econômica após a simplificação:



Minimização de circuitos: Mapas de Karnaugh

- A minimização de funções Booleanas:
 - reduz o número de portas em um chip, produzindo circuitos mais confiáveis e baratos,
 - possibilita a implementação de mais circuitos em um mesmo chip, contribuindo para a minimização do tamanho físico de componentes eletrônicos, e
 - reduz o tempo usado por um circuito para calcular sua saída.
- Estudaremos o procedimento conhecido como **mapas de Karnaugh** ou **mapas-K**)
 - Projetado na década de 1950 para ajudar a minimizar os circuitos à mão.
 - Úteis na minimização de circuitos com até seis variáveis (embora se tornem bastante complexos para cinco ou seis variáveis.)

Minimização de circuitos: Mapas de Karnaugh

- Considere as DNFs de funções Booleanas de duas variáveis x e y .

- Existem quatro mintermos possíveis:

- xy

- $\bar{x}y$

- $\bar{y}x$

- $\bar{x}\bar{y}$

- O mapa de Karnaugh correspondente tem quatro células, uma pra cada mintermo

- "1" é colocado na célula se o mintermo ocorrer da DNF

- Células são ditas **adjacentes** se os mintermos que elas representam diferirem exatamente um literal.

- Por exemplo, a célula que representa xy é adjacente às células que representam $\bar{x}y$ e $x\bar{y}$.

	y	\bar{y}
x	xy	$x\bar{y}$
\bar{x}	$\bar{x}y$	$\bar{x}\bar{y}$

Minimização de circuitos: Mapas de Karnaugh

- Exemplo 28 Encontre mapas de Karnaugh para as expressões Booleanas:

a) $x y + \bar{x} y$

b) $x \bar{y} + \bar{x} y$

c) $x \bar{y} + \bar{x} y + \bar{x} \bar{y}$

Minimização de circuitos: Mapas de Karnaugh

- Exemplo 28 Encontre mapas de Karnaugh para as expressões Booleanas:

a) $xy + \bar{x}y$

b) $x\bar{y} + \bar{x}y$

c) $x\bar{y} + \bar{x}y + \bar{x}\bar{y}$

Solução. Os mapas de Karnaugh são dados abaixo.

	y	\bar{y}
x	1	
\bar{x}	1	

	y	\bar{y}
x		1
\bar{x}	1	

	y	\bar{y}
x		1
\bar{x}	1	1



Minimização de circuitos: Mapas de Karnaugh

- O mapa de Karnaugh identifica mintermos a serem combinados:
 - ① Mintermos adjacentes com 1s em suas células podem ser combinados em um produto envolvendo apenas uma das variáveis.
 - A outra variável ocorre com e sem complemento
 - Por exemplo, $x\bar{y}$ e $\bar{x}\bar{y}$ são adjacentes e podem ser combinados em \bar{y} porque

$$x\bar{y} + \bar{x}\bar{y} = (x + \bar{x})\bar{y} = 1\bar{y} = \bar{y}$$

Minimização de circuitos: Mapas de Karnaugh

- O mapa de Karnaugh identifica mintermos a serem combinados:
 1. Mintermos adjacentes com 1s em suas células podem ser combinados em um produto envolvendo apenas uma das variáveis.
 - A outra variável ocorre com e sem complemento
 - Por exemplo, $x\bar{y}$ e $\bar{x}\bar{y}$ são adjacentes e podem ser combinados em \bar{y} porque

$$x\bar{y} + \bar{x}\bar{y} = (x + \bar{x})\bar{y} = 1\bar{y} = \bar{y}$$

2. Se todas as células contêm 1s, os quatro mintermos podem ser combinados, eliminando todas as variáveis e resultando na expressão Booleana 1.

Minimização de circuitos: Mapas de Karnaugh

- O mapa de Karnaugh identifica mintermos a serem combinados:
 1. Mintermos adjacentes com 1s em suas células podem ser combinados em um produto envolvendo apenas uma das variáveis.
 - A outra variável ocorre com e sem complemento
 - Por exemplo, $x\bar{y}$ e $\bar{x}\bar{y}$ são adjacentes e podem ser combinados em \bar{y} porque

$$x\bar{y} + \bar{x}\bar{y} = (x + \bar{x})\bar{y} = 1\bar{y} = \bar{y}$$

2. Se todas as células contêm 1s, os quatro mintermos podem ser combinados, eliminando todas as variáveis e resultando na expressão Booleana 1.
 3. Circulamos blocos de células no mapa de Karnaugh, representando os mintermos que podem ser combinados.
- O objetivo é identificar os maiores blocos possíveis

Minimização de circuitos: Mapas de Karnaugh

- Exemplo 29 Simplifique as somas de produtos das expressões Booleanas do exemplo anterior:

a) $x y + \bar{x} y$

b) $x \bar{y} + \bar{x} y$

c) $x \bar{y} + \bar{x} y + \bar{x} \bar{y}$

Solução. Agrupamos os mintermos com o procedimento descrito anteriormente, obtendo os seguintes mapas de Karnaugh:

	y	\bar{y}
x	1	
\bar{x}	1	

	y	\bar{y}
x		1
\bar{x}	1	

	y	\bar{y}
x		1
\bar{x}	1	1

Minimização de circuitos: Mapas de Karnaugh

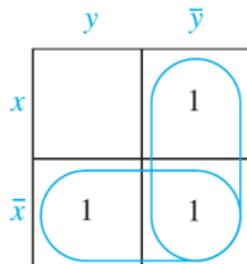
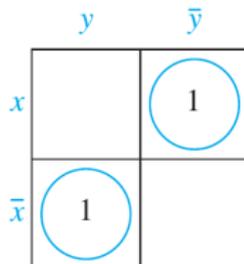
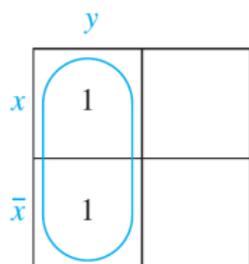
- Exemplo 29 Simplifique as somas de produtos das expressões Booleanas do exemplo anterior:

a) $xy + \bar{x}y$

b) $x\bar{y} + \bar{x}y$

c) $x\bar{y} + \bar{x}y + \bar{x}\bar{y}$

Solução. Agrupamos os mintermos com o procedimento descrito anteriormente, obtendo os seguintes mapas de Karnaugh:



A partir dos mapas acima, podemos encontrar as fórmulas minimizadas:

a) y

b) $x\bar{y} + \bar{x}y$

c) $\bar{x} + \bar{y}$



Minimização de circuitos: Mapas de Karnaugh

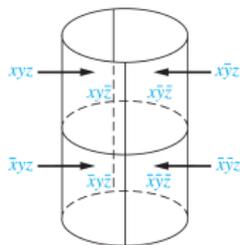
- O processo de simplificação de uma expressão em um mapa de Karnaugh de três variáveis é semelhante àquele usado para duas variáveis.
- Um mapa de Karnaugh de três variáveis x, y, z é um retângulo dividido em oito células, uma para cada mintermo possível:

- $x y z,$
- $x \bar{y} z,$
- $\bar{x} y z,$
- $\bar{x} \bar{y} z,$ e
- $x y \bar{z},$
- $x \bar{y} \bar{z},$
- $\bar{x} y \bar{z},$
- $\bar{x} \bar{y} \bar{z}.$

- Duas células são ditas **adjacentes** se os mintermos que eles representam diferem em exatamente um literal.

- Podemos formar um mapa de Karnaugh em três variáveis assim:

	yz	$y\bar{z}$	$\bar{y}z$	$\bar{y}\bar{z}$
x	xyz	$xy\bar{z}$	$x\bar{y}z$	$x\bar{y}\bar{z}$
\bar{x}	$\bar{x}yz$	$\bar{x}y\bar{z}$	$\bar{x}\bar{y}z$	$\bar{x}\bar{y}\bar{z}$



Minimização de circuitos: Mapas de Karnaugh

- Blocos de duas células adjacentes (tamanho 1×2 ou 2×1) representam mintermos que podem ser combinados em um produto de dois literais.
- Blocos de tamanho 2×2 e 4×1 representam mintermos que podem ser combinados em um único literal.
- O bloco de todas as oito células representa um produto sem literais, ou seja, a função Booleana 1.

Minimização de circuitos: Mapas de Karnaugh

- Exemplo 30 Exemplos de blocos de células adjacentes em um mapa de Karnaugh de 3 variáveis e suas expressões equivalentes minimizadas:

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x				
\bar{x}				

$$x\bar{y}\bar{z} + \bar{x}\bar{y}\bar{z} = \bar{y}\bar{z}$$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x				
\bar{x}				

$$\bar{x}yz + \bar{x}\bar{y}z = \bar{x}z$$

Minimização de circuitos: Mapas de Karnaugh

- Exemplo 30 (Continuação)

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x				
\bar{x}				

$$xy\bar{z} + x\bar{y}\bar{z} + \bar{x}y\bar{z} + \bar{x}\bar{y}\bar{z} = \bar{z}$$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x				
\bar{x}				

$$\bar{x}yz + \bar{x}y\bar{z} + \bar{x}\bar{y}\bar{z} + \bar{x}\bar{y}z = \bar{x}$$

Minimização de circuitos: Mapas de Karnaugh

- Exemplo 30 (Continuação)

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x				
\bar{x}				

$$xyz + xy\bar{z} + x\bar{y}\bar{z} + x\bar{y}z + \bar{x}yz + \bar{x}y\bar{z} + \bar{x}\bar{y}\bar{z} + \bar{x}\bar{y}z = 1$$



Minimização de circuitos: Mapas de Karnaugh

- O produto de literais que correspondem a um bloco só de 1s é chamado de **implicante** da função sendo minimizada.
- Um implicante é chamado de **implicante primo** se ele não estiver contido em um bloco maior de 1s representando o produto de menos literais.
- O objetivo é identificar os maiores blocos possíveis no mapa e cobrir todos os 1s com o menor número de blocos, usando os maiores blocos primeiro.
- Os maiores blocos possíveis sempre são escolhidos, mas devemos escolher um bloco sempre que ele for o único cobrindo um 1 no mapa de Karnaugh.

Tal bloco representa um **implicante primo essencial**.

- Ao cobrir todos os 1s no mapa com blocos correspondentes aos implicantes primos podemos expressar a soma dos produtos como um soma dos implicantes primos.

Note que pode haver mais de uma maneira de cobrir todos os 1s usando o menor número de blocos.

Minimização de circuitos: Mapas de Karnaugh

- Exemplo 31 Vamos usar mapas de Karnaugh para minizar as seguintes expressões Booleanas de 3 variáveis.

a) $x y \bar{z} + x \bar{y} \bar{z} + \bar{x} y z + \bar{x} \bar{y} \bar{z}$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x		1	1	
\bar{x}	1		1	

Expressão minimizada:

$$x \bar{z} + \bar{y} \bar{z} + \bar{x} y z$$

b) $x \bar{y} z + x \bar{y} \bar{z} + \bar{x} y z + \bar{x} \bar{y} z + \bar{x} \bar{y} \bar{z}$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x			1	1
\bar{x}	1		1	1

Expressão minimizada:

$$\bar{y} + \bar{x} z$$

Minimização de circuitos: Mapas de Karnaugh

- Exemplo 31 (Continuação)

c) $x y z + x y \bar{z} + x \bar{y} z + x \bar{y} \bar{z} + \bar{x} y z + \bar{x} \bar{y} z + \bar{x} \bar{y} \bar{z}$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x	1	1	1	1
\bar{x}	1		1	1

Expressão minimizada:

$$x + \bar{y} + z$$

d) $x y \bar{z} + x \bar{y} \bar{z} + \bar{x} \bar{y} z + \bar{x} \bar{y} \bar{z}$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x		1	1	
\bar{x}			1	1

Expressão minimizada:

$$x \bar{z} + \bar{x} \bar{y}$$

(Note que não precisamos incluir $\bar{y}\bar{z}$ porque as células já estão cobertas por outros mintermos.)



Minimização de circuitos: Mapas de Karnaugh

- Mapas de Karnaugh com 4 variáveis funcionam de forma análoga aos mapas com menos variáveis.
- A forma geral de um mapa de Karnaugh de 4 variáveis é a seguinte:

	yz	$y\bar{z}$	$\bar{y}z$	$\bar{y}\bar{z}$
wx	$wxyz$	$wxy\bar{z}$	$wx\bar{y}z$	$wx\bar{y}\bar{z}$
$w\bar{x}$	$w\bar{x}yz$	$w\bar{x}y\bar{z}$	$w\bar{x}\bar{y}z$	$w\bar{x}\bar{y}\bar{z}$
$\bar{w}x$	$\bar{w}xyz$	$\bar{w}xy\bar{z}$	$\bar{w}x\bar{y}z$	$\bar{w}x\bar{y}\bar{z}$
$\bar{w}\bar{x}$	$\bar{w}\bar{x}yz$	$\bar{w}\bar{x}y\bar{z}$	$\bar{w}\bar{x}\bar{y}z$	$\bar{w}\bar{x}\bar{y}\bar{z}$

Minimização de circuitos: Mapas de Karnaugh

- Exemplo 32 Exemplos de blocos de células adjacentes em um mapa de Karnaugh de 4 variáveis e suas expressões equivalentes minimizadas:

	yz	$y\bar{z}$	$\bar{y}z$	$\bar{y}\bar{z}$
wx				
$w\bar{x}$				
$\bar{w}\bar{x}$				
$\bar{w}x$				

$$w\bar{x}yz + w\bar{x}\bar{y}z = w\bar{x}z$$

	yz	$y\bar{z}$	$\bar{y}z$	$\bar{y}\bar{z}$
wx				
$w\bar{x}$				
$\bar{w}\bar{x}$				
$\bar{w}x$				

$$\begin{aligned} &\bar{w}\bar{x}yz + \bar{w}\bar{x}y\bar{z} + \\ &+ \bar{w}\bar{x}\bar{y}\bar{z} + \bar{w}\bar{x}\bar{y}z = \bar{w}\bar{x} \end{aligned}$$

Minimização de circuitos: Mapas de Karnaugh

- Exemplo 32 (Continuação)

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				
$w\bar{x}$				
$\bar{w}\bar{x}$				
$\bar{w}x$				

$$wx y z + wx \bar{y} z + \bar{w} x y z + \bar{w} x \bar{y} z = x z$$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				
$w\bar{x}$				
$\bar{w}\bar{x}$				
$\bar{w}x$				

$$wx y \bar{z} + wx \bar{y} \bar{z} + w\bar{x} y \bar{z} + w\bar{x} \bar{y} \bar{z} + \bar{w}\bar{x} y \bar{z} + \bar{w}\bar{x} \bar{y} \bar{z} + \bar{w}x y \bar{z} + \bar{w}x \bar{y} \bar{z} = \bar{z}$$



Minimização de circuitos: Mapas de Karnaugh

- Exemplo 33 Vamos usar mapas de Karnaugh para minizar as seguintes expressões Booleanas de 3 variáveis.

a) $wxyz + wxy\bar{z} + wx\bar{y}\bar{z} + w\bar{x}yz + w\bar{x}\bar{y}z + w\bar{x}\bar{y}\bar{z} + \bar{w}x\bar{y}z + \bar{w}\bar{x}yz + \bar{w}\bar{x}y\bar{z}$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1	1	1	
$w\bar{x}$	1		1	1
$\bar{w}\bar{x}$	1	1		
$\bar{w}x$				1

Expressão minimizada:

$$wyz + wx\bar{z} + w\bar{x}\bar{y} + \bar{w}\bar{x}y + \bar{w}x\bar{y}z$$

Minimização de circuitos: Mapas de Karnaugh

- Exemplo 33 (Continuação)

b) $w x \bar{y} \bar{z} + w \bar{x} y z + w \bar{x} y \bar{z} + w \bar{x} \bar{y} \bar{z} + \bar{w} x \bar{y} \bar{z} + \bar{w} \bar{x} y \bar{z} + \bar{w} \bar{x} \bar{y} \bar{z}$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx			1	
$w\bar{x}$	1	1	1	
$\bar{w}\bar{x}$		1	1	
$\bar{w}x$			1	

Expressão minimizada:

$$\bar{y}\bar{z} + w\bar{x}y + \bar{x}\bar{z}$$

Minimização de circuitos: Mapas de Karnaugh

- Exemplo 33 (Continuação)

c) $wxy\bar{z} + wx\bar{y}\bar{z} + w\bar{x}yz + w\bar{x}y\bar{z} + w\bar{x}\bar{y}\bar{z} + \bar{w}xyz + \bar{w}xy\bar{z} + \bar{w}x\bar{y}\bar{z} + \bar{w}\bar{x}\bar{y}z + \bar{w}\bar{x}y\bar{z} + \bar{w}\bar{x}\bar{y}\bar{z}$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx		1	1	
$w\bar{x}$	1	1	1	
$\bar{w}\bar{x}$		1	1	
$\bar{w}x$	1	1	1	1

Expressão minimizada:

$$\bar{z} + \bar{w}x + w\bar{x}y$$



Minimização de circuitos: Método de Quine-McCluskey

- Como já discutimos, mapas de Karnaugh são úteis na minimização de circuitos com até seis variáveis, mas eles se tornem bastante complexos para cinco ou seis variáveis.
- Para minimizar circuitos com mais de 6 variáveis existe o **método Quine-McCluskey**, que foi inventado nos anos 60.
- Este método automatiza o processo de minimizar circuitos combinatórios e pode ser implementado como um programa de computador.

(Neste curso, entretanto, não vamos estudar o método de Quine-McCluskey: este conteúdo é coberto em curso de *hardware*.)