Portas Lógicas

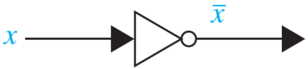
Portas lógicas: Introdução

- 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

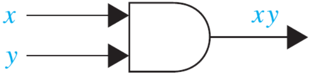
- Vamos nos concentrar em três tipos de portas lógicas.

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

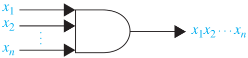


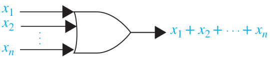
  - A **porta OR**, que implementa a operação de soma Booleana:

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



- É possível considerar portas com várias entradas:





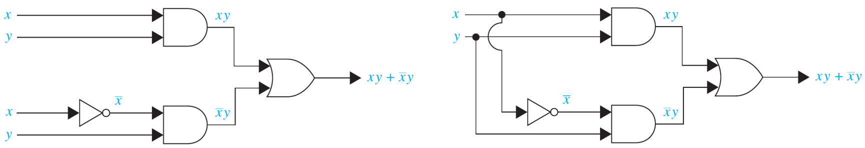
Circuitos combinatórios

- Circuitos combinatórios podem ser construídos usando uma combinação de inversores, portas OR e portas AND.

- Quando são formadas combinações de circuitos, alguma portas podem compartilhar entradas.

  Exitem maneiras diferentes de representar isto graficamente.

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

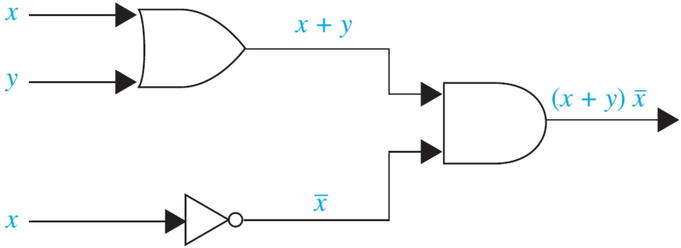


Cirbuitos combinatórios

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

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

Solução. O circuito desejado está desenhado a seguir.



Exemplos de circuitos: Votação majoritária

- $\boxed{\text{Exemplo 25}}$ 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

- $\boxed{\text{Exemplo 25}}$ (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:

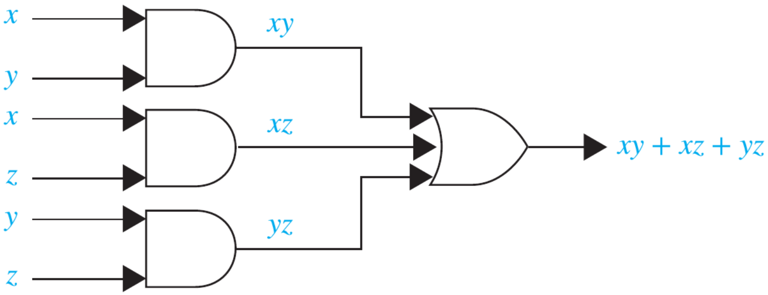
$$\overline{x}\,y\,z + x\,\overline{y}\,z + x\,y\,\overline{z} + x\,y\,z$$

que podemos mostrar ser equivalente a

$$xy + xz + yz\ .$$

- 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

- $\boxed{\text{Exemplo 26}}$ 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

- $\boxed{\text{Exemplo 26}}$ (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:

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

$$c = xy$$

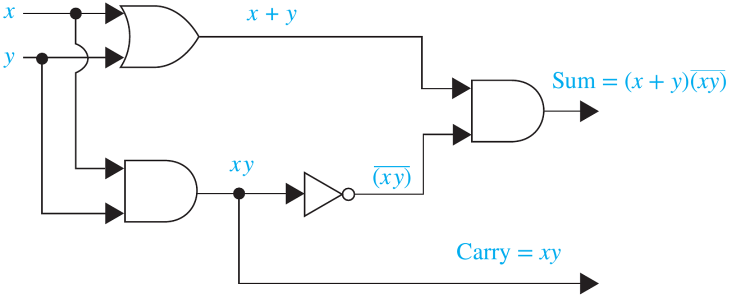
Por conveniência, vamos transformar a expressão de s em uma equivalente antes de implementar o circuito:

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

Exemplos de circuitos: Somadores

- $\boxed{\text{Exemplo 26}}$ (Continuação)

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



Exemplos de circuitos: Somadores

- $\boxed{\text{Exemplo 26}}$ (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

- $\boxed{\text{Exemplo 26}}$ (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

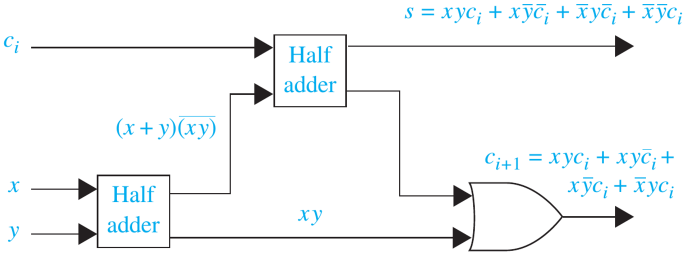
Podemos derivar as expressões:

$$s = \overline{x}\,\overline{y}\,c_i + \overline{x}\,y\,\overline{c_i} + x\,\overline{y}\,\overline{c_i} + x\,y\,c_i$$

$$c_{i+1} = \overline{x}\,y\,c_i + x\,\overline{y}\,c_i + x\,y\,\overline{c_i} + x\,y\,c_i$$

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!)

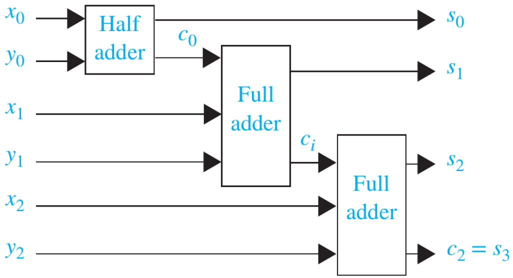
A prática de construir circuitos maiores usando componentes já desenhados é muito comum no projeto de circuitos reais.

Exemplos de circuitos: Somadores

- $\boxed{\text{Exemplo 26}}$ (Continuação)

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

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



Minimização de Circuitos

Minimização de Circuitos: Introdução

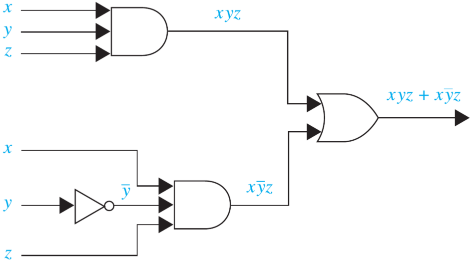
- A eficiência de um circuito combinatório depende do número e da disposição de suas portas.
- O processo de projetar um circuito combinatório começa com a tabela especificando a saída para cada combinação de valores de entrada.
- Podemos sempre usar a forma normal disjuntiva (DNF) pra implementar um 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

- $\boxed{\text{Exemplo 27}}$ 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:
  - $x\,y$
  - $\overline{x}\,y$
  - $\overline{y}\,x$
  - $\overline{x}\,\overline{y}$
- 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 $x\,y$ é adjacente às células que representam $\overline{x}\,y$ e $x\,\overline{y}$.

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

Minimização de circuitos: Mapas de Karnaugh

- $\boxed{\text{Exemplo 28}}$ Encontre mapas de Karnaugh para as expressões Booleanas:

  a) $x\,y + \overline{x}\,y$

  b) $x\,\overline{y} + \overline{x}\,y$

  c) $x\,\overline{y} + \overline{x}\,y + \overline{x}\,\overline{y}$

Minimização de circuitos: Mapas de Karnaugh

- $\boxed{\text{Exemplo 28}}$ Encontre mapas de Karnaugh para as expressões Booleanas:

  a) $x\,y + \overline{x}\,y$ b) $x\,\overline{y} + \overline{x}\,y$ c) $x\,\overline{y} + \overline{x}\,y + \overline{x}\,\overline{y}$

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, $x\,\overline{y}$ e $\overline{x}\,\overline{y}$ são adjacentes e podem ser combinados em $\overline{y}$ porque

$$x\,\overline{y} + \overline{x}\,\overline{y} \quad = \quad (x + \overline{x})\,\overline{y} \quad = \quad 1\,\overline{y} \quad = \quad \overline{y}$$

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, $x\,\overline{y}$ e $\overline{x}\,\overline{y}$ são adjacentes e podem ser combinados em $\overline{y}$ porque

     $$x\,\overline{y} + \overline{x}\,\overline{y} \quad = \quad (x + \overline{x})\,\overline{y} \quad = \quad 1\,\overline{y} \quad = \quad \overline{y}$$

  2. Se todas as células contém 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, $x\,\overline{y}$ e $\overline{x}\,\overline{y}$ são adjacentes e podem ser combinados em $\overline{y}$ porque

       $$x\,\overline{y} + \overline{x}\,\overline{y} \quad = \quad (x + \overline{x})\,\overline{y} \quad = \quad 1\,\overline{y} \quad = \quad \overline{y}$$
  2. Se todas as células contém 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

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

  a) $x\,y + \overline{x}\,y$ b) $x\,\overline{y} + \overline{x}\,y$ c) $x\,\overline{y} + \overline{x}\,y + \overline{x}\,\overline{y}$

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

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

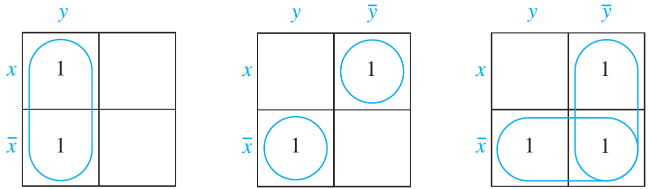
	y	$\bar{y}$
x		1
$\bar{x}$	1	

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

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

  a) $x\,y + \overline{x}\,y$ b) $x\,\overline{y} + \overline{x}\,y$ c) $x\,\overline{y} + \overline{x}\,y + \overline{x}\,\overline{y}$

  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) $x\,\overline{y} + \overline{x}\,y$ c) $\overline{x} + \overline{y}$

- O processo de simplificação de uma expressão em um mapa de Karnaugh de três variáveis é semelhante àquele usado para duas variáveis.

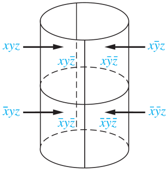
- Um mapa de Karnaugh de três variáveis x, y, z é um retângulo dividido em oito células, uma para cada mintermo possível:

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

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



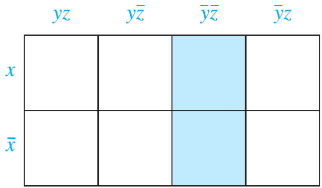
Minimização de circuitos: Mapas de Karnaugh

- Blocos de duas células adjacentes (tamanho 1×2 ou 2×1) representam mintermos que podem ser combinados em um produto de dois literais.

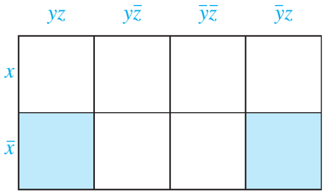
- Blocos de tamanho 2×2 e 4×1 representam mintermos que podem ser combinados em um único literal.

- O bloco de todas as oito células representa um produto sem literais, ou seja, a função Booleana 1.

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



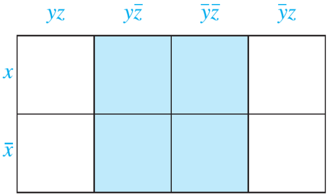
$$x\,\overline{y}\,\overline{z} + \overline{x}\,\overline{y}\,\overline{z} \quad = \quad \overline{y}\,\overline{z}$$



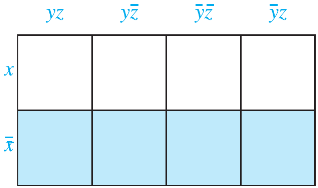
$$\overline{x}\,y\,z + \overline{x}\,\overline{y}\,z \quad = \quad \overline{x}\,z$$

Minimização de circuitos: Mapas de Karnaugh

- Exemplo 30 (Continuação)



$$x\,y\,\overline{z} + x\,\overline{y}\,\overline{z} + \overline{x}\,y\,\overline{z} + \overline{x}\,\overline{y}\,\overline{z} \quad = \quad \overline{z}$$



$$\overline{x}\,y\,z + \overline{x}\,y\,\overline{z} + \overline{x}\,\overline{y}\,\overline{z} + \overline{x}\,\overline{y}\,z \quad = \quad \overline{x}$$

Minimização de circuitos: Mapas de Karnaugh

- $\boxed{\text{Exemplo 30}}$ (Continuação)

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

$$\begin{aligned} & x\,y\,z + x\,y\,\bar{z} + x\,\bar{y}\,\bar{z} + x\,\bar{y}\,z + \\ & + \bar{x}\,y\,z + \bar{x}\,y\,\bar{z} + \bar{x}\,\bar{y}\,\bar{z} + \bar{x}\,\bar{y}\,z \quad = \quad 1 \end{aligned}$$

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.

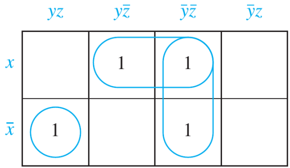
  Tal bloco representa um **implicante primo essencial**.
- Ao cobrir todos os 1s no mapa com blocos correspondentes aos implicantes primos podemos expressar a soma dos produtos como um soma dos implicantes primos.

  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.

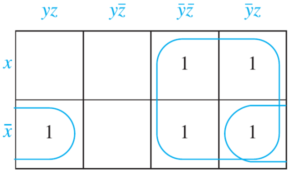
a) $x\,y\,\overline{z} + x\,\overline{y}\,\overline{z} + \overline{x}\,y\,z + \overline{x}\,\overline{y}\,\overline{z}$



Expressão minimizada:

$$x\,\overline{z} + \overline{y}\,\overline{z} + \overline{x}\,y\,z$$

b) $x\,\overline{y}\,z + x\,\overline{y}\,\overline{z} + \overline{x}\,y\,z + \overline{x}\,\overline{y}\,z + \overline{x}\,\overline{y}\,\overline{z}$



Expressão minimizada:

$$\overline{y} + \overline{x}\,z$$

- $\boxed{\text{Exemplo 31}}$ (Continuação)

c) $x\,y\,z + x\,y\,\overline{z} + x\,\overline{y}\,z + x\,\overline{y}\,\overline{z} + \overline{x}\,y\,z + \overline{x}\,\overline{y}\,z + \overline{x}\,\overline{y}\,\overline{z}$

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

Expressão minimizada:

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

d) $x\,y\,\overline{z} + x\,\overline{y}\,\overline{z} + \overline{x}\,\overline{y}\,z + \overline{x}\,\overline{y}\,\overline{z}$

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

Expressão minimizada:

$$x\,\overline{z} + \overline{x}\,\overline{y}$$

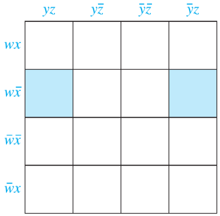
(Note que não precisamos incluir $\overline{y}\,\overline{z}$ porque as células já estão cobertas por outros mintermos.)

Minimização de circuitos: Mapas de Karnaugh

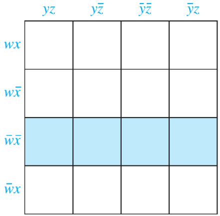
- Mapas de Karnaugh com 4 variáveis funcionam de forma análoga aos mapas com menos variáveis.
- A forma geral de um mapa de Karnaugh de 4 variáveis é a seguinte:

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

- $\boxed{\text{Exemplo 32}}$ Exemplos de blocos de células adjacentes em um mapa de Karnaugh de 4 variáveis e suas expressões equivalentes minimizadas:

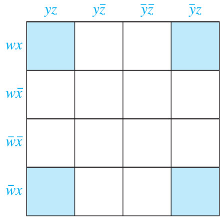


$$w\,\overline{x}\,y\,z + w\,\overline{x}\,\overline{y}\,z \quad = \quad w\,\overline{x}\,z$$

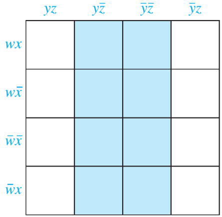


$$\overline{w}\,\overline{x}\,y\,z + \overline{w}\,\overline{x}\,y\,\overline{z} +$$
$$+ \overline{w}\,\overline{x}\,\overline{y}\,\overline{z} + \overline{w}\,\overline{x}\,\overline{y}\,z \quad = \quad \overline{w}\,\overline{x}$$

- $\boxed{\text{Exemplo 32}}$ (Continuação)



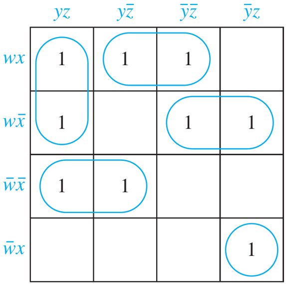
$$\begin{aligned} & w\,x\,y\,z + w\,x\,\overline{y}\,z + \\ & +\overline{w}\,x\,y\,z + \overline{w}\,x\,\overline{y}\,z \quad = \quad x\,z \end{aligned}$$



$$\begin{aligned} & w\,x\,y\,\overline{z} + w\,x\,\overline{y}\,\overline{z} + \\ & +w\,\overline{x}\,y\,\overline{z} + w\,\overline{x}\,\overline{y}\,\overline{z} + \\ & +\overline{w}\,\overline{x}\,y\,\overline{z} + \overline{w}\,\overline{x}\,\overline{y}\,\overline{z} + \\ & +\overline{w}\,x\,y\,\overline{z} + \overline{w}\,x\,\overline{y}\,\overline{z} \quad = \quad \overline{z} \end{aligned}$$

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

  a) $w\,x\,y\,z + w\,x\,y\,\overline{z} + w\,x\,\overline{y}\,\overline{z} + w\,\overline{x}\,y\,z + w\,\overline{x}\,\overline{y}\,z + w\,\overline{x}\,\overline{y}\,\overline{z} + \overline{w}\,x\,\overline{y}\,z + \overline{w}\,\overline{x}\,y\,z + \overline{w}\,\overline{x}\,y\,\overline{z}$



Expressão minimizada:

$w\,y\,z + w\,x\,\overline{z} + w\,\overline{x}\,\overline{y} + \overline{w}\,\overline{x}\,y + \overline{w}\,x\,\overline{y}\,z$

Minimização de circuitos: Mapas de Karnaugh

- $\boxed{\text{Exemplo 33}}$ (Continuação)

  b) $w\,x\,\overline{y}\,\overline{z} + w\,\overline{x}\,y\,z + w\,\overline{x}\,y\,\overline{z} + w\,\overline{x}\,\overline{y}\,\overline{z} + \overline{w}\,x\,\overline{y}\,\overline{z} + \overline{w}\,\overline{x}\,y\,\overline{z} + \overline{w}\,\overline{x}\,\overline{y}\,\overline{z}$

	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:

$$\overline{y}\,\overline{z} + w\,\overline{x}\,y + \overline{x}\,\overline{z}$$

Minimização de circuitos: Mapas de Karnaugh

- $\boxed{\text{Exemplo 33}}$ (Continuação)

  c) $w\,x\,y\,\overline{z} + w\,x\,\overline{y}\,\overline{z} + w\,\overline{x}\,y\,z + w\,\overline{x}\,y\,\overline{z} + w\,\overline{x}\,\overline{y}\,\overline{z} + \overline{w}\,x\,y\,z + \overline{w}\,x\,y\,\overline{z} + \overline{w}\,x\,\overline{y}\,\overline{z} + \overline{w}\,x\,\overline{y}\,z + \overline{w}\,\overline{x}\,y\,\overline{z} + \overline{w}\,\overline{x}\,\overline{y}\,\overline{z}$

	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:

$$\overline{z} + \overline{w}\,x + w\,\overline{x}\,y$$

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*.)