SIMPLIFICAÇÃO DE CIRCUITOS DIGITAIS

Thiago Felski Pereira
Adaptado do livro Sistemas Digitais (Tocci)

FORMA DE SOMA-DE-PRODUTOS

- A expressão *soma-de-produtos* (SOP) aparecerá como dois ou mais termos AND combinados com operações OR.

$$1.\ ABC + \overline{A}B\overline{C}$$

$$2.\ AB + \overline{A}B\overline{C} + \overline{C}\,\overline{D} + D$$

$$3.\ \overline{A}B + C\overline{D} + EF + GK + H\overline{L}$$

Forma de Soma-de-Produtos

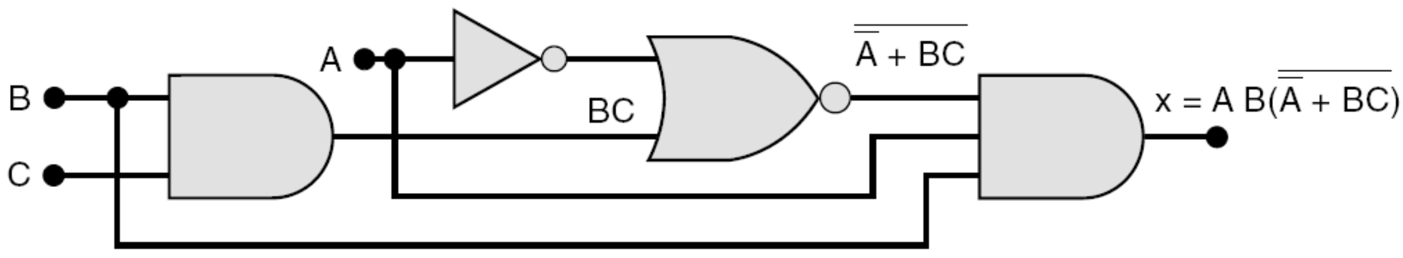
- A expressão *produto-de-somas* (POS) consiste de dois ou mais termos OR (soma) combinados com operações AND.

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

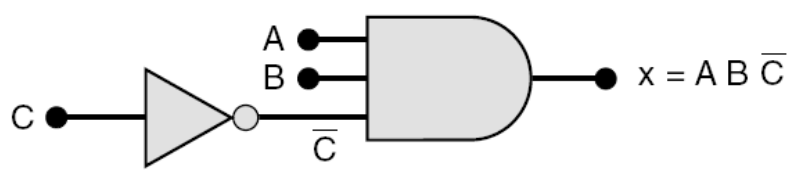
SIMPLIFICAÇÃO DE CIRCUITOS LÓGICOS

- Os circuitos mostrados fornecem a mesma saída.

  O circuito (b) é claramente menos complexo.



(a)



Circuitos lógicos podem ser simplificados utilizando-se álgebra booleana ou mapas de Karnaugh.

SIMPLIFICAÇÃO ALGÉBRICA

Algumas dicas:

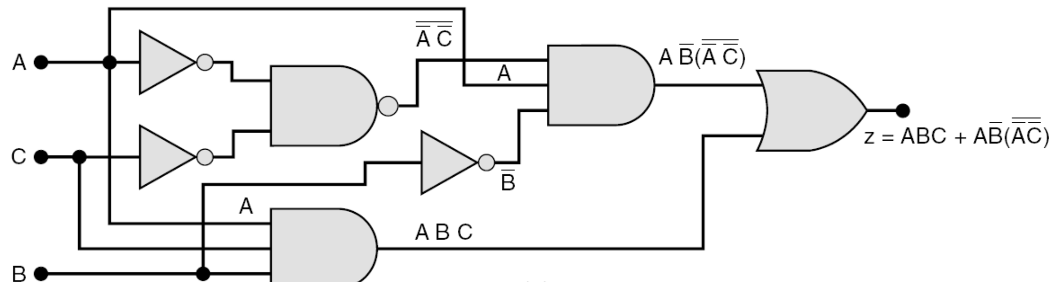
- Coloque a expressão na forma SOP através da aplicação de teoremas de DeMorgan e multiplicação de termos.

  Verifique a forma SOP de fatores comuns, utilizando a fatoração, sempre que possível.

- A fatoração resulta na eliminação de um ou mais termos.

Simplificação algébrica

Exemplo 1: Simplique o circuito lógico mostrado a seguir.



(a)

O primeiro passo é determinar a expressão para a saída:

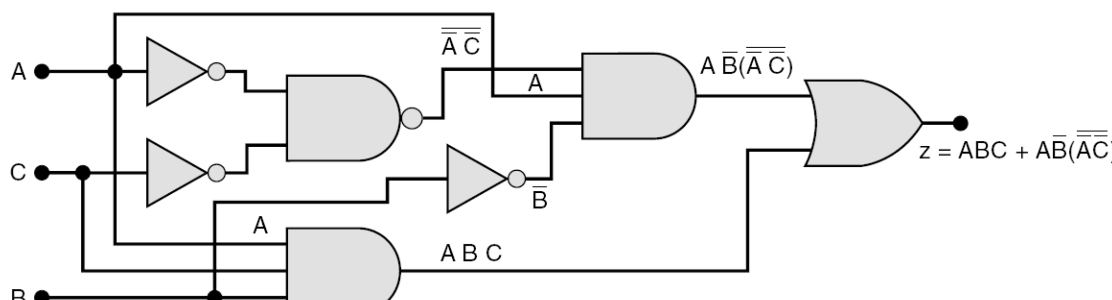
$$z = A\overline{B}C + AB \cdot \overline{(\overline{A}\ \overline{C})}$$

Uma vez que a expressão é determinada, deve-se quebrar as barras de inversão pelo teorema de DeMorgan e multiplicar todos os termos.

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

Simplificação algébrica

Exemplo 2: Simplique o circuito lógico mostrado a seguir.



(a)

Fatoração - os primeiros e terceiros termos acima têm em comum **AC**, que pode ser fatorado como:

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

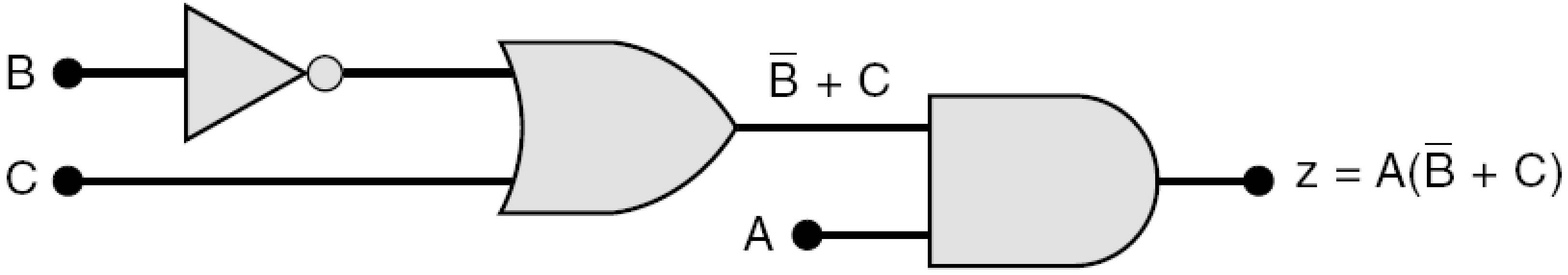
Desde que ***B*** + ***B*** = 1, assim…

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

Fatorar ***A***, que resulta em…

Simplificação algébrica

Circuitos lógicos simplificados.



(b)

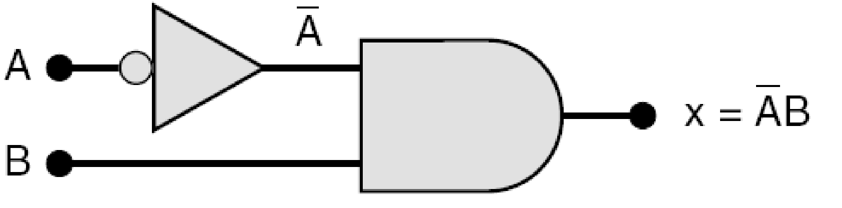
$$z = A(C + \overline{B})$$

PROJETANDO CIRCUITOS LÓGICOS COMBINACIONAIS

- Para a resolução de qualquer problema de lógica de projeto:
  - Interprete o problema e defina sua tabela-verdade.
  - Escreva o termo AND (produto) para cada caso de saída = 1.
  - Combine os termos na forma SOP.
  - Simplifique a expressão da saída, se possível.
  - Implemente o circuito para a expressão final, simplificada.

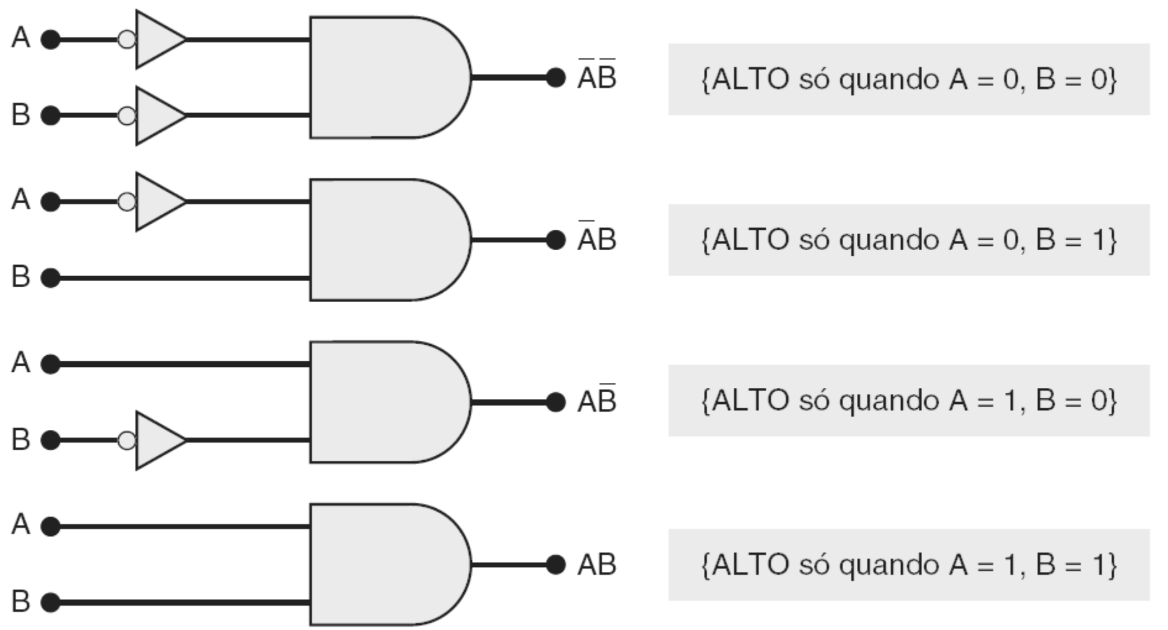
Circuito que produz uma saída 1 apenas para a condição $A = 0$ $B = 1$.

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



PROJETANDO CIRCUITOS LÓGICOS COMBINACIONAIS

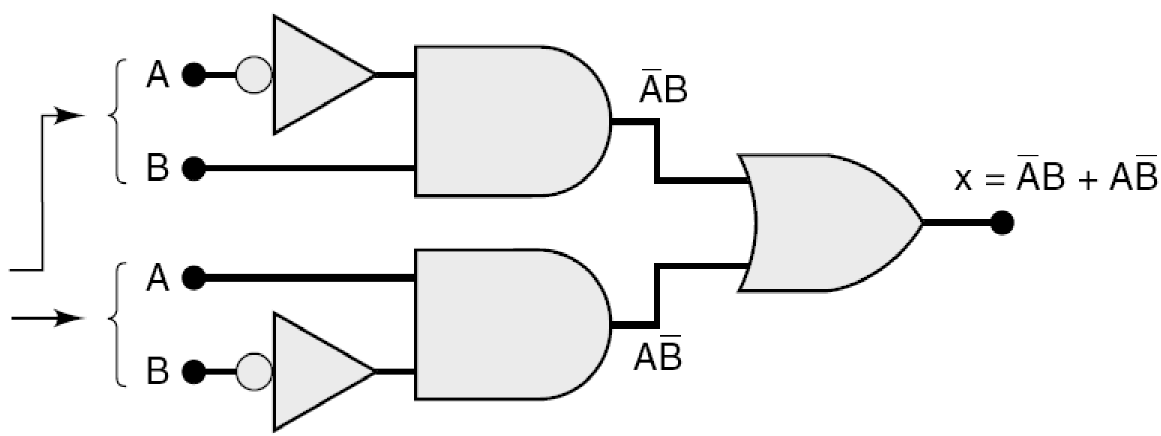
Uma porta AND, com entradas apropriadas, pode ser usada para produzir uma saída em nível 1 para um conjunto específico de níveis de entrada.



PROJETANDO CIRCUITOS LÓGICOS COMBINACIONAIS

Cada conjunto de condições de entrada, que gera uma saída em nível ALTO, é implementado por portas AND separadas.
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



PROJETANDO CIRCUITOS LÓGICOS COMBINACIONAIS

Tabela-verdade para um circuito de três entradas, *A*, *B* e *C*.

A	*B*	*C*	*x*	
0	0	0	0	
0	0	1	0	
0	1	0	1	$\longrightarrow \overline{A}B\overline{C}$
0	1	1	1	$\longrightarrow \overline{A}BC$
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	1	$\longrightarrow ABC$

Termos **AND** para cada caso em que a saída é 1.

PROJETANDO CIRCUITOS LÓGICOS COMBINACIONAIS

Projetar um circuito lógico com três entradas, *A*, *B* e *C*. As saídas devem ser ALTA somente quando a maioria das entradas for ALTA.

Tabela-verdade

A	*B*	*C*	*x*	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\longrightarrow \overline{A}BC$
1	0	0	0	
1	0	1	1	$\longrightarrow A\overline{B}C$
1	1	0	1	$\longrightarrow AB\overline{C}$
1	1	1	1	$\longrightarrow ABC$

Termos **AND** para cada caso em que a saída é 1.

Expressão SOP para a saída:

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

PROJETANDO CIRCUITOS LÓGICOS COMBINACIONAIS

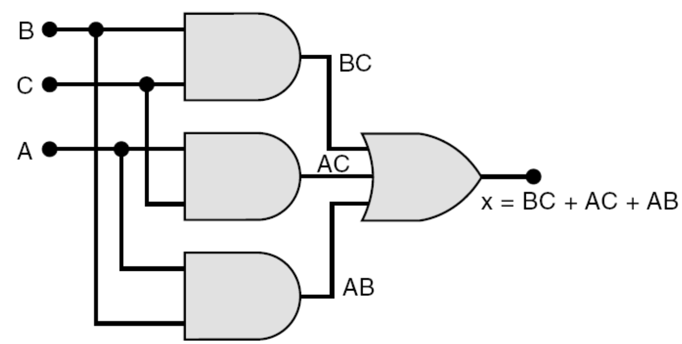
Exemplo 3: Projetar um circuito lógico com três entradas, *A*, *B* e *C*. As saídas devem ser ALTA somente quando a maioria das entradas for ALTA.

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

Expressão saída simplificada:

$$x = BC + AC + AB$$

Implementando o circuito após fatoração:



Uma vez que a expressão está na forma SOP, o circuito é um grupo de portas AND trabalhando em uma única porta OR.

MÉTODO DO MAPA DE KARNAUGH

- Também chamado de mapa K, é um método gráfico para simplificar equações lógicas ou converter tabelas-verdade no circuito lógico correspondente.
- Teoricamente, pode ser usado para qualquer número de variáveis de entradas, porém sua utilidade prática é limitada a cinco ou seis variáveis.

Os valores da tabela-verdade são colocados no mapa K.

O mostrado aqui é de duas variáveis.

A	B	X	
0	0	1	$\rightarrow \overline{A}\,\overline{B}$
0	1	0	
1	0	0	
1	1	1	$\rightarrow AB$

$$\left\{ x = \overline{A}\,\overline{B} + AB \right\}$$

	$\overline{B}$	B
$\overline{A}$	1	0
A	0	1

MÉTODO DO MAPA DE KARNAUGH

Mapa K de quatro variáveis.

A	B	C	D	X	
0	0	0	0	0	
0	0	0	1	1	$\rightarrow \bar{A}\bar{B}\bar{C}D$
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	0	
0	1	0	1	1	$\rightarrow \bar{A}B\bar{C}D$
0	1	1	0	0	
0	1	1	1	0	
1	0	0	0	0	
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	0	
1	1	0	0	0	
1	1	0	1	1	$\rightarrow AB\bar{C}D$
1	1	1	0	0	
1	1	1	1	1	$\rightarrow ABCD$

$$\left\{ \begin{array}{l} X = \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D \\ \quad + AB\bar{C}D + ABCD \end{array} \right\}$$

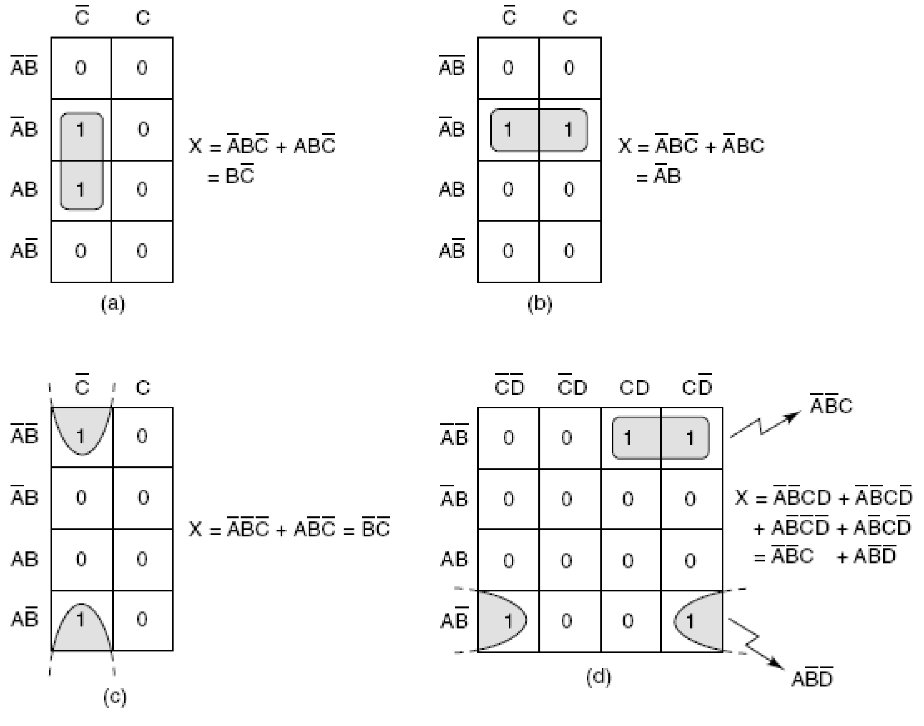
	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	0	0
$\bar{A}B$	0	1	0	0
AB	0	1	1	0
$A\bar{B}$	0	0	0	0

Células adjacentes diferem em apenas uma variável, tanto na horizontal quanto na vertical.

Uma expressão SOP pode ser obtida combinando todos os quadrados que contêm 1.

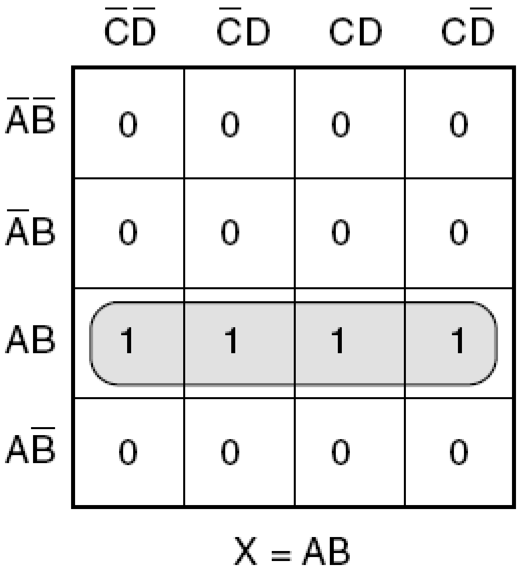
MÉTODO DO MAPA DE KARNAUGH

Agrupando-se 1s em adjacentes de dois, quatro ou oito quadros têm-se uma maior simplificação.

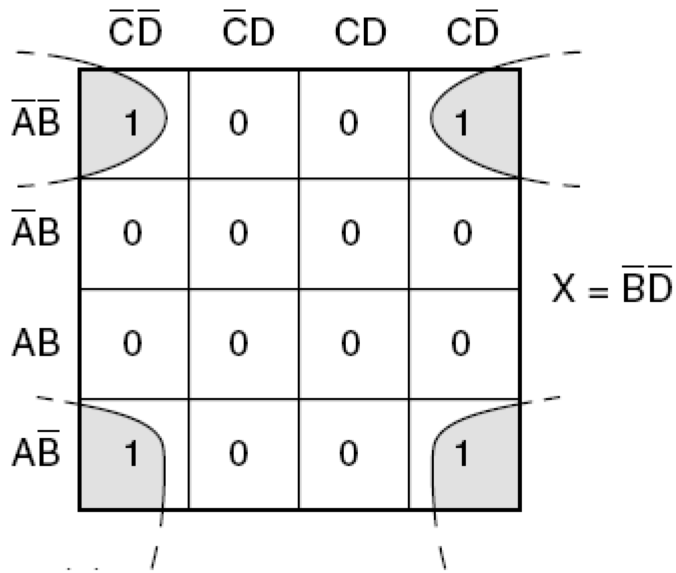


MÉTODO DO MAPA DE KARNAUGH

Grupos de dois quadros (Pares)



Grupo de quatro (Quarteto)



Grupo de oito (Octeto)

MÉTODO DO MAPA DE KARNAUGH

- Quando os maiores grupos possíveis forem usados, somente os termos comuns são colocados na expressão final.

Agrupamentos também podem ser realizados entre superior, inferior e laterais.

MÉTODO DO MAPA DE KARNAUGH

- Passos para uso do mapa K para simplificação de uma expressão booleana:

  Construção do mapa K, com os 1s como indicado na tabela-verdade.

  Agrupamento dos 1s que não são adjacentes a quaisquer outros 1s (1s *isolados*).

  Agrupamento dos 1s que estão em pares.

  Agrupamento dos 1s em octetos, mesmo que já tenham sido agrupados.

  Agrupamento dos quartetos com um ou mais 1s e que ainda não estejam em grupos.

  Agrupamento de quaisquer pares necessários para incluir 1s ainda não agrupado.

  Formação da soma OR dos termos gerados por cada grupo.

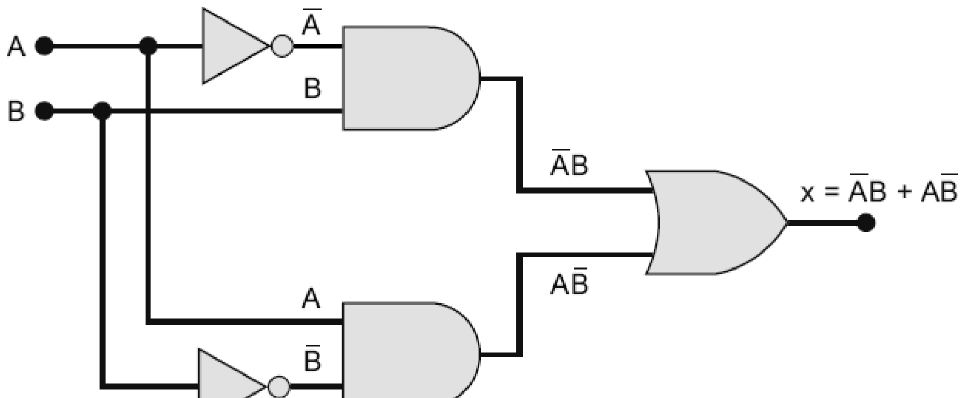
Quando uma variável aparece na forma complementada e não complementada dentro de um grupo é eliminada da expressão.
Variáveis que são as mesmas para todos os quadrados do grupo devem aparecer na expressão final.

CIRCUITOS EXCLUSIVE-OR E EXCLUSIVE-NOR

- O exclusive-OR (XOR) produz uma saída em nível ALTO sempre que as duas entradas estejam em níveis opostos.

CIRCUITOS EXCLUSIVE-OR E EXCLUSIVE-NOR

Circuito exclusive-OR e tabela-verdade.



(a)

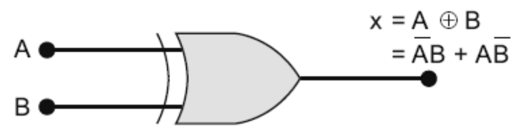
A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

Expressão da saída : $\boldsymbol{x = \overline{A}B + \overline{A}B}$

Esse circuito produz uma saída ALTA sempre que as duas entradas estiverem em níveis diferentes.

CIRCUITOS EXCLUSIVE-OR E EXCLUSIVE-NOR

Símbolo tradicional para a porta XOR:



(b)

Uma porta XOR tem apenas *duas* entradas combinadas, de forma que $x = \overline{A}B + A\overline{B}$.

A forma abreviada para indicar a expressão de saída XOR é: $x = A \oplus B$.

…em que o símbolo $\oplus$ representa a operação da porta XOR.

A saída é ALTA somente quando as duas entradas estão em níveis diferentes.

CIs com chips quádruplos de portas XOR:

74LS86 CI quádruplo XOR (família TTL)
74C86 CI quádruplo XOR (família CMOS)
74HC86 CI quádruplo XOR (CMOS de alta velocidade)

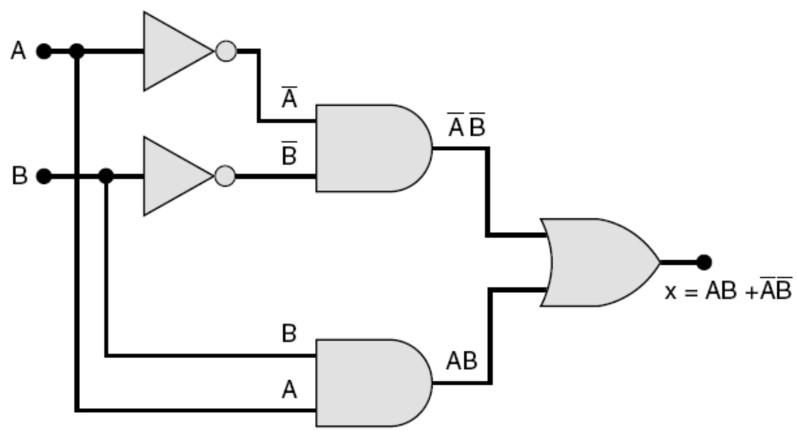
CIRCUITOS EXCLUSIVE-OR E EXCLUSIVE-NOR

- O exclusive-NOR (XOR) produz uma saída em nível ALTO sempre que as duas entradas estão no mesmo nível.

As saídas XOR e XNOR são opostas.

CIRCUITOS EXCLUSIVE-OR E EXCLUSIVE-NOR

Circuito exclusive-**NOR** e tabela-verdade.



(a)

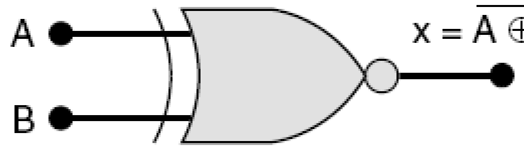
A	B	x
0	0	1
0	1	0
1	0	0
1	1	1

Expressão de saída : $x = AB + \overline{A}\,\overline{B}$

XNOR produz uma saída ALTA sempre que duas entradas estão no mesmo nível.

CIRCUITOS EXCLUSIVE-OR E EXCLUSIVE-NOR

Símbolo tradicional de porta XNOR



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

Uma porta XNOR tem apenas *duas* entradas combinadas. Assim, $x = AB + \overline{A}\,\overline{B}.$

A forma abreviada para indicar a expressão de saída XOR é: $x = \overline{A \oplus B}.$

XNOR representa o inverso da operação **XOR**.

A saída é ALTA apenas quando as entradas estão no mesmo nível.

CIs com chips quádruplos de portas XNOR

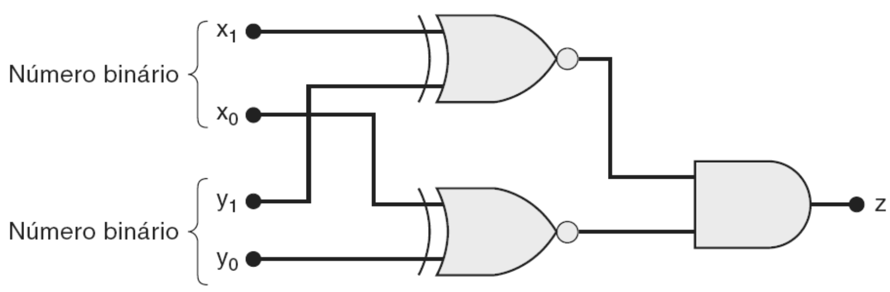
74LS266 CI quádruplo **XNOR** (família TTL)
74C266 CI quádruplo **XOR** (família CMOS)
74HC266 CI quádrulo **XOR** (CMOS de alta velocidade)

CIRCUITOS EXCLUSIVE-OR E EXCLUSIVE-NOR

Tabela-verdade e circuito de detecção de igualdade de números binários de dois bits.

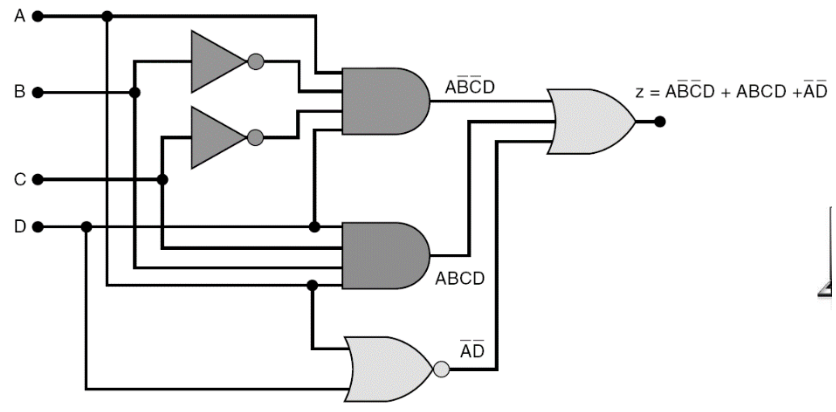
x_1	x_0	y_1	y_0	*z* (Saída)
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0

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

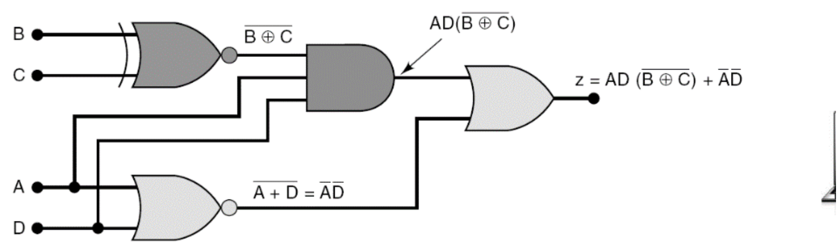


CIRCUITOS EXCLUSIVE-OR E EXCLUSIVE-NOR

Exemplo de como uma porta **XNOR** pode ser usada para simplificar a implementação de um circuito.



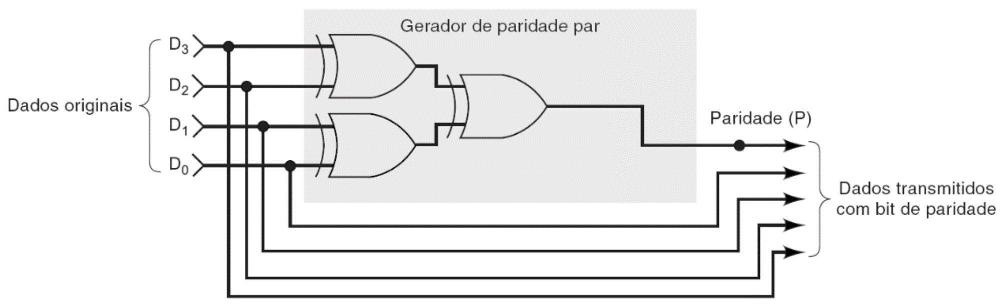
(a)

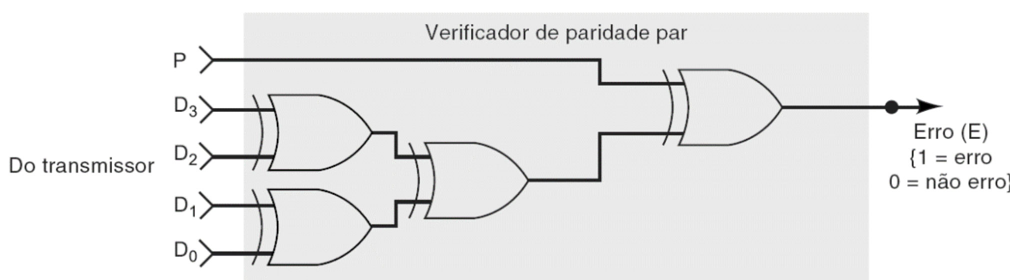


(b)

CIRCUITOS GERADOR E VERIFICADOR DE PARIDADE

Portas XOR e XNOR são úteis em circuitos para geração e verificação de paridade.





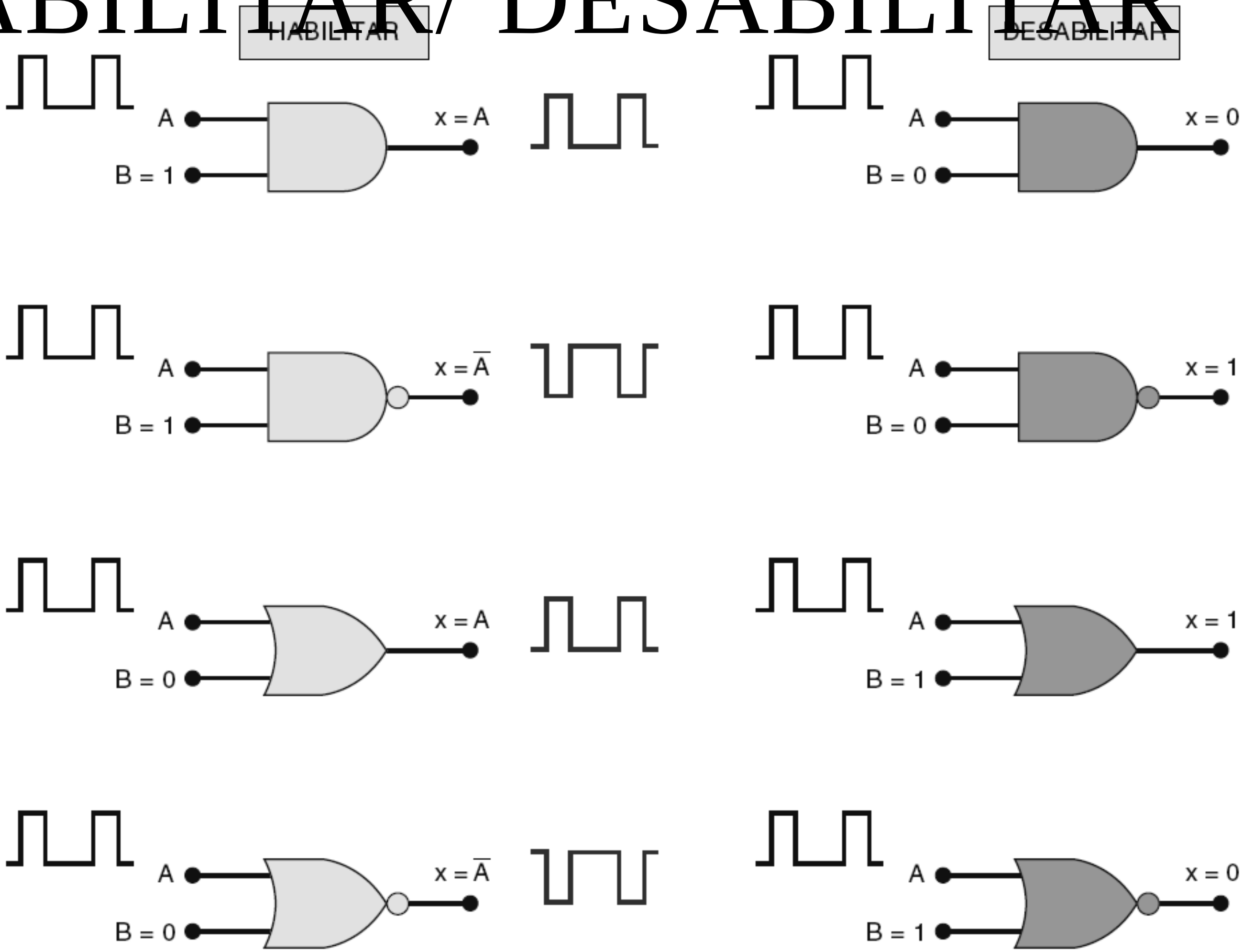
CIRCUITOS GERADOR E VERIFICADOR DE PARIDADE

Situações que exigem habilitar/ desabilitar os circuitos ocorrem com frequência em projeto de circuitos digitais.

Um circuito é *habilitado* quando se permite a passagem de um sinal de entrada para saída.

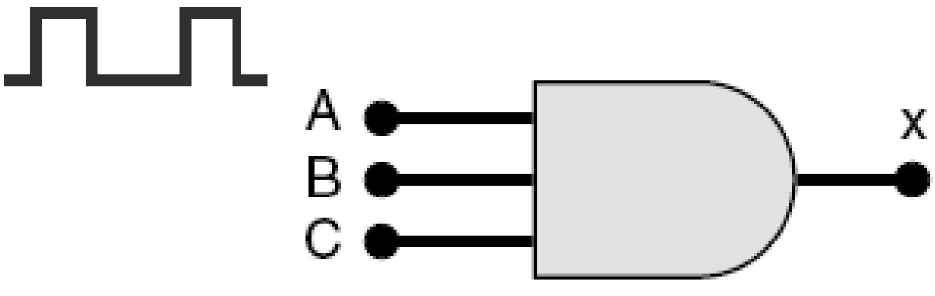
Um circuito é *desabilitado* quando se impede a passagem de um sinal de entrada para saída.

CIRCUITOS PARA HABILITAR/ DESABILITAR



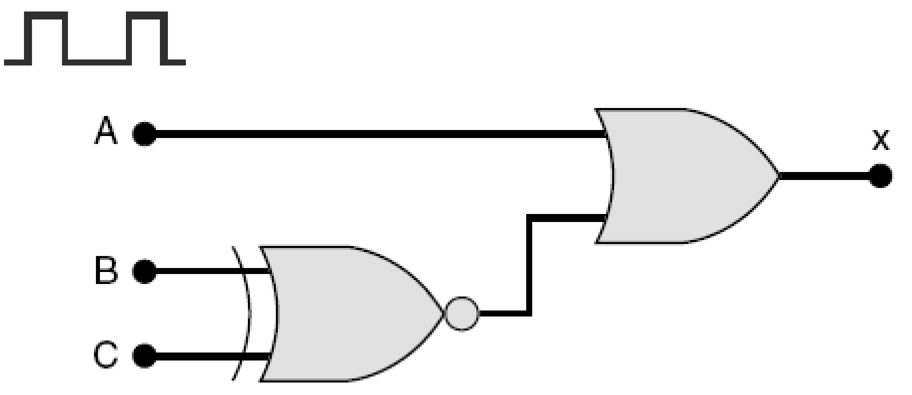
CIRCUITOS PARA HABILITAR/ DESABILITAR

Um circuito lógico que permite a passagem de um sinal para a saída somente quando entradas de controle *B* e *C* forem ambas nível ALTO. Caso contrário, a saída permanecerá em nível BAIXO.



CIRCUITOS PARA HABILITAR/ DESABILITAR

Um circuito lógico que permite a passagem de um sinal para a saída apenas quando uma entrada, mas não ambas, for nível ALTO. Caso contrário, a saída permanecerá ALTA.



CIRCUITOS PARA HABILITAR/ DESABILITAR

Um circuito lógico com sinal de entrada *A*, controle de entrada *B* e saídas *X* e *Y*, que atuam como:
Quando *B* = 1, a saída *X* vai seguir a entrada *A*, e a saída *Y* será 0.

Quando *B* = 0, a saída *X* vai ser 0, e a saída *Y* vai seguir a entrada *A*.