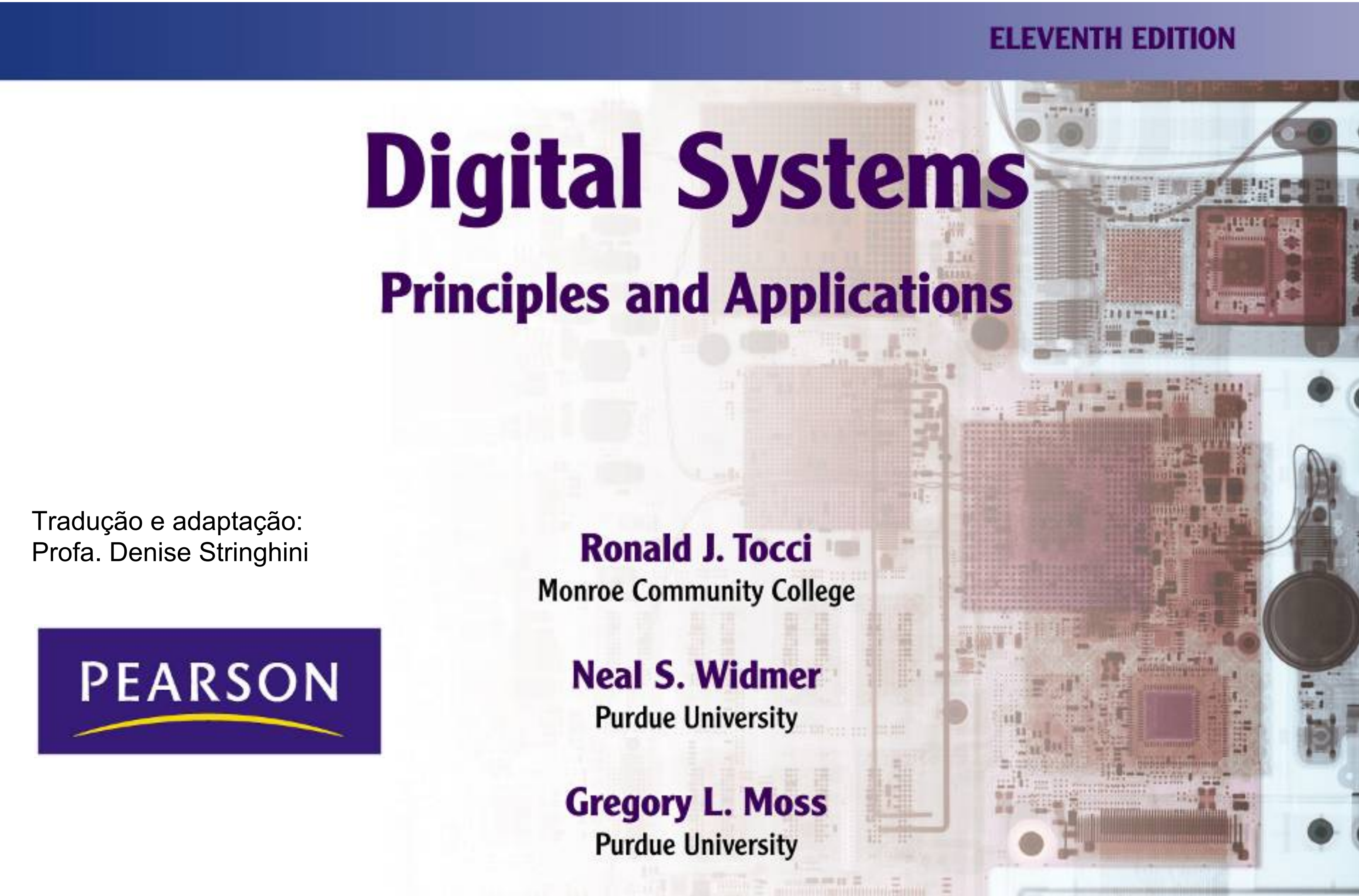
Capítulo 3 – Descrição de Circuitos Lógicos (parte 2)

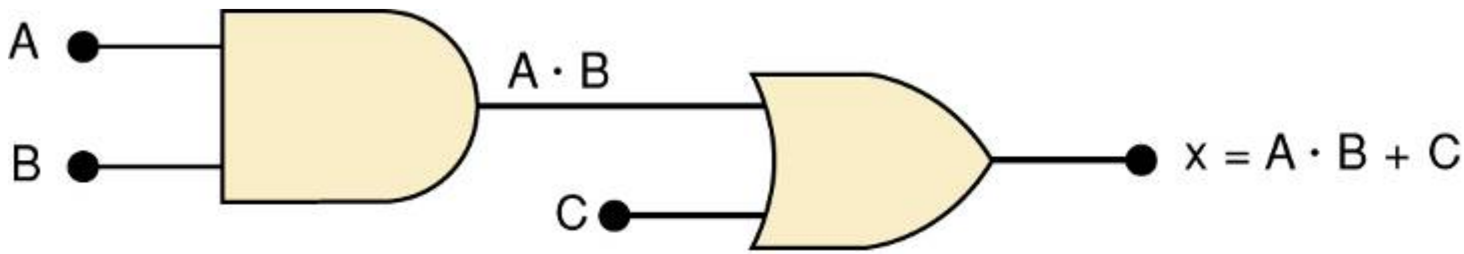


Tradução e adaptação:
Profa. Denise Stringhini

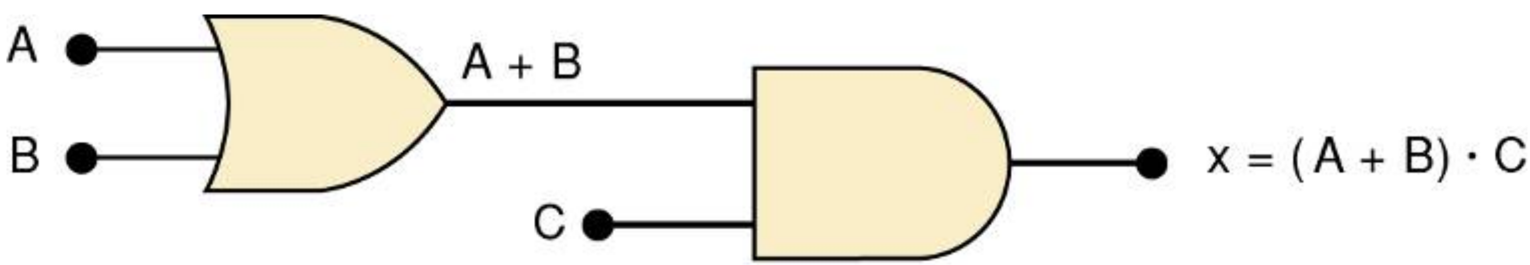
PEARSON

3-6 Descrição algébrica de circuitos lógicos

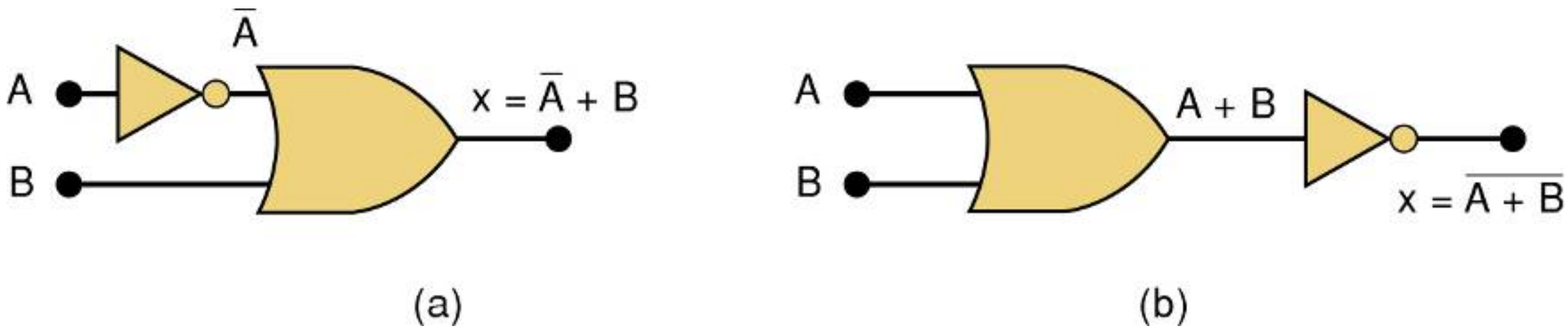
Se uma expressão contém portas AND e OR, AND será realizada antes.



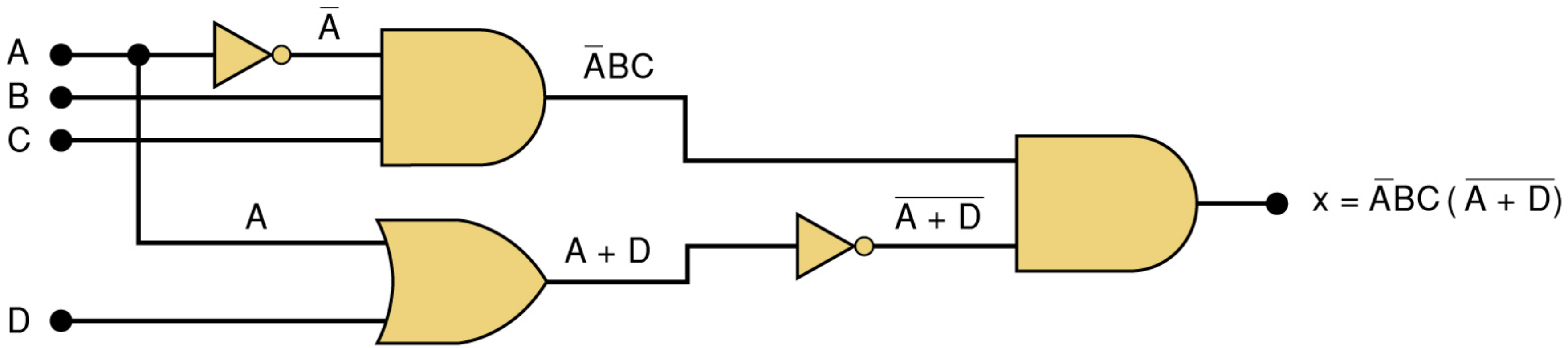
A menos que haja um parêntese na expressão.



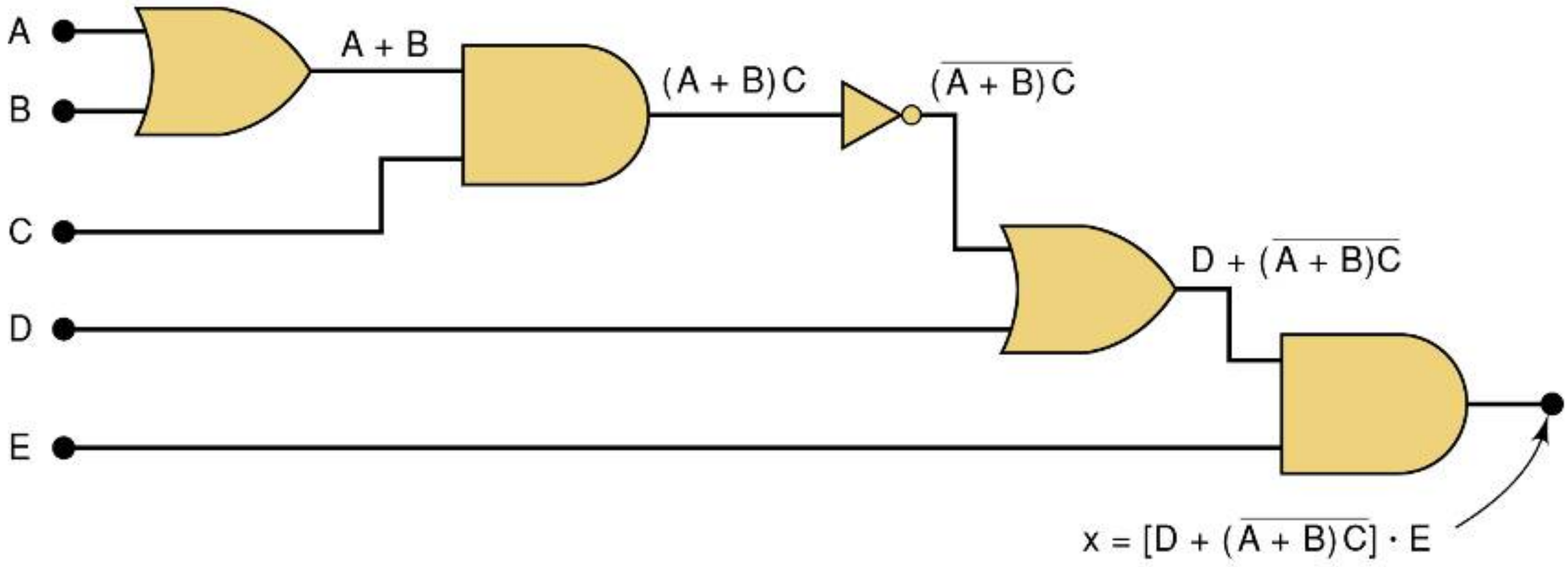
Sempre que um INVERSOR está presente, a saída é equivalente à entrada, com uma barra sobre ele.



Outros exemplos...



Outros exemplos...



3-7 Avaliação das saídas de um circuito lógico

Regras para avaliar uma expressão booleana:

- Executar todas as **inversões** de termos individuais.
- Realizar todas as operações dentro de **parênteses**.
- Executar operação **AND** antes de uma operação **OR** (a menos que parênteses indiquem o contrário).
- Se uma **expressão tem uma barra sobre** ela, realizar as operações dentro da expressão, e, em seguida, inverter o resultado.

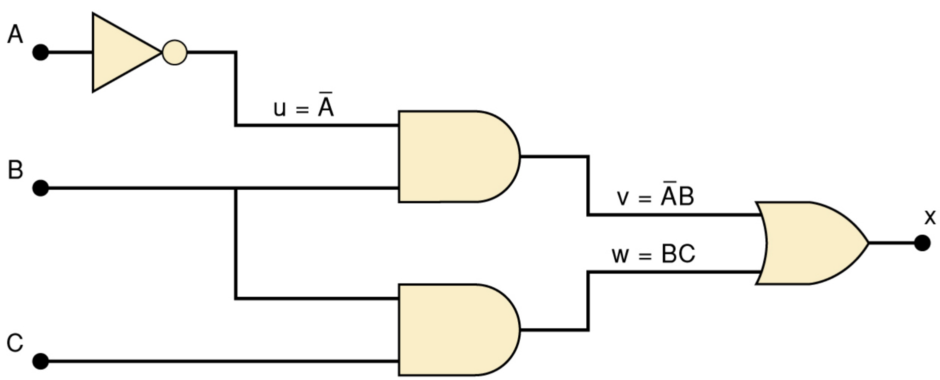
3-7 Avaliação das saídas de um circuito lógico

A melhor maneira de analisar um circuito composto de múltiplas portas lógicas é a utilização de uma **tabela verdade:**

- Ela permite que você analise uma porta ou combinação lógica de cada vez.
- Ela permite que você verifique facilmente o seu trabalho.
- Quando você tiver terminado, terá uma tabela de grande benefício na solução de problemas no circuito lógico.

3-7 Avaliação das saídas de um circuito lógico

O primeiro passo depois de enumerar todas as combinações de entrada é criar uma coluna na tabela verdade para cada sinal intermediário (nó).

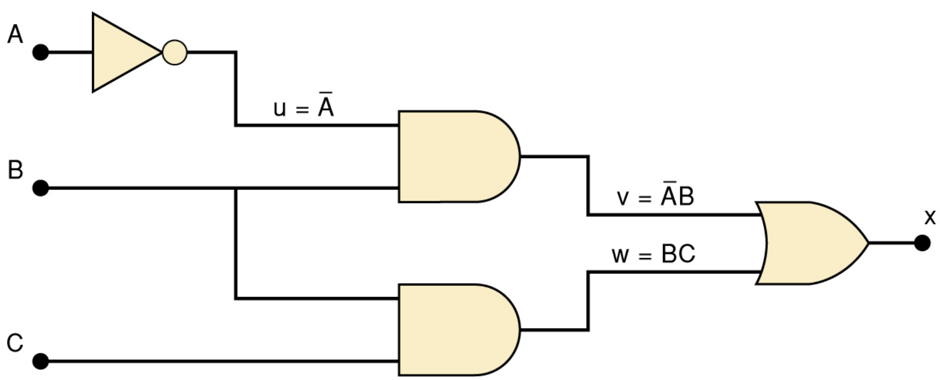


A	B	C	u= $\overline{A}$	v= $\overline{A}B$	w= BC	x= v+w
0	0	0	1			
0	0	1	1			
0	1	0	1			
0	1	1	1			
1	0	0	0			
1	0	1	0			
1	1	0	0			
1	1	1	0			

Nó *u* foi preenchido com o complemento de *A*

3-7 Avaliação das saídas de um circuito lógico

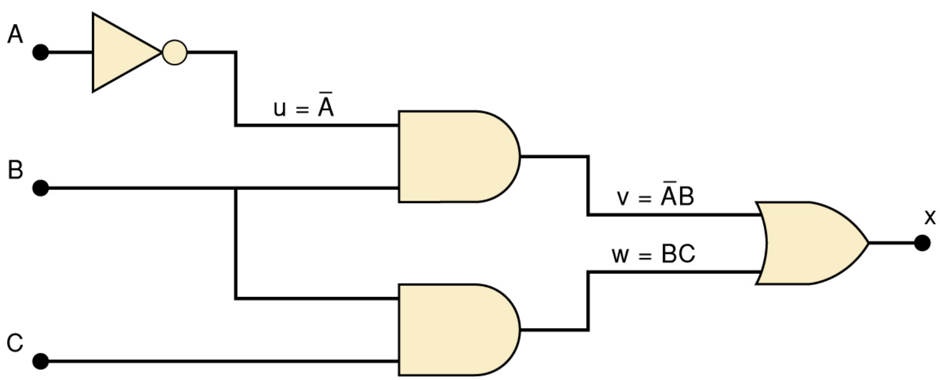
- O próximo passo é preencher os valores para coluna ***v***.



A	B	C	u= $\overline{A}$	v= $\overline{A}B$	w= BC	x= v+w
0	0	0	1	0		
0	0	1	1	0		
0	1	0	1	1		
0	1	1	1	1		
1	0	0	0	0		
1	0	1	0	0		
1	1	0	0	0		
1	1	1	0	0		

v =*A'B* — Nó *v será* ALTO quando *A'* (nó *u*) estiver ALTO **E** *B* estiver ALTO

- O terceiro passo é de prever os valores **w** no nó que é o produto lógico de BC.

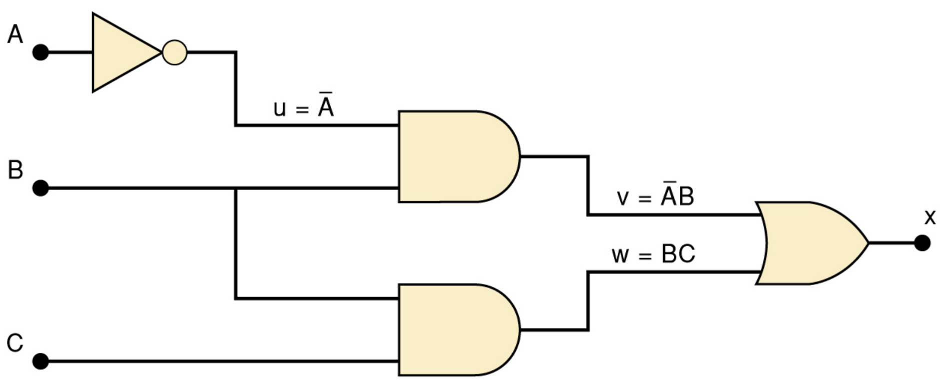


A	B	C	u= $\bar{A}$	v= $\bar{A}B$	w= BC	x= v+w
0	0	0	1	0	0	
0	0	1	1	0	0	
0	1	0	1	1	0	
0	1	1	1	1	1	
1	0	0	0	0	0	
1	0	1	0	0	0	
1	1	0	0	0	0	
1	1	1	0	0	1	

Esta coluna é ALTA sempre que *B* é ALTO **E** *C* é ALTO

3-7 Avaliação das saídas de um circuito lógico

- O passo final consiste em combinar logicamente colunas **v** e **w** para prever a saída **x**.



A	B	C	u= $\bar{A}$	v= $\bar{A}B$	w= BC	x= v+w
0	0	0	1	0	0	0
0	0	1	1	0	0	0
0	1	0	1	1	0	1
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	0	0	1	1

Visto que *x* = *v* + w, a saída *x é* ALTA quando *v* ***OU*** *w são* ALTOS

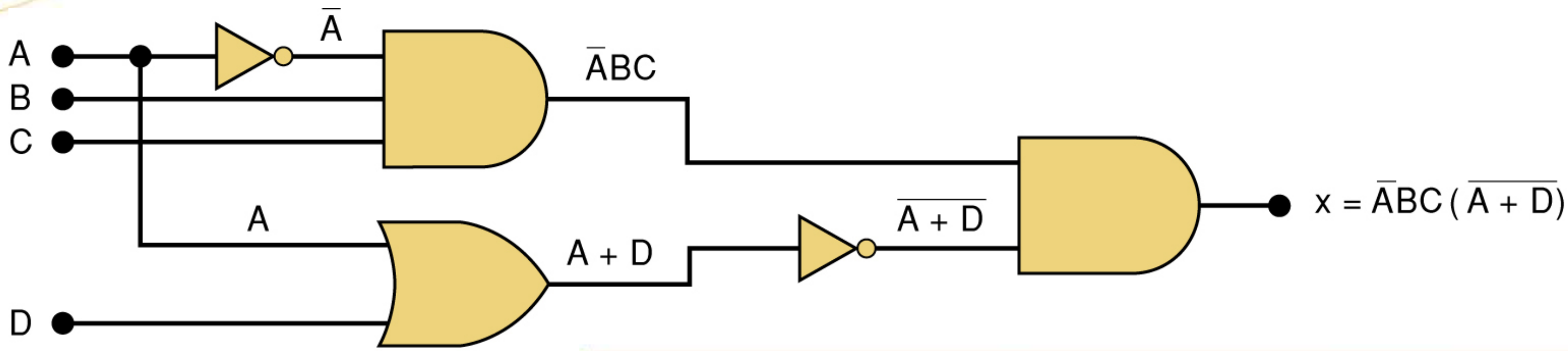
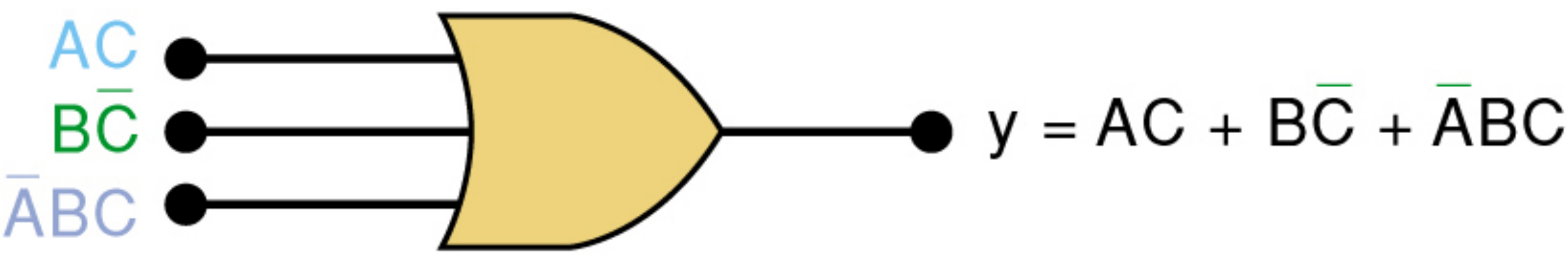


Tabela de estado lógico em cada nó do circuito mostrado.

A	B	C	D	$t = \overline{A}BC$	$u = A + D$	$v = \overline{A + D}$	$x = tv$
0	0	0	0	0	0	1	0
0	0	0	1	0	1	0	0
0	0	1	0	0	0	1	0
0	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	1
0	1	1	1	1	1	0	0
1	0	0	0	0	1	0	0
1	0	0	1	0	1	0	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	0	0
1	1	0	0	0	1	0	0
1	1	0	1	0	1	0	0
1	1	1	0	0	1	0	0
1	1	1	1	0	1	0	0

- É importante ser capaz de desenhar um circuito lógico a partir de uma expressão booleana.
  – A expressão $X = A \bullet B \bullet C$, poderia ser desenhada como uma porta AND de três entradas.
- Um circuito definido por $\boldsymbol{X} = \overline{\boldsymbol{A}} + \boldsymbol{B}$, iria usar uma porta OR de duas entradas com um inversor em uma das entradas.

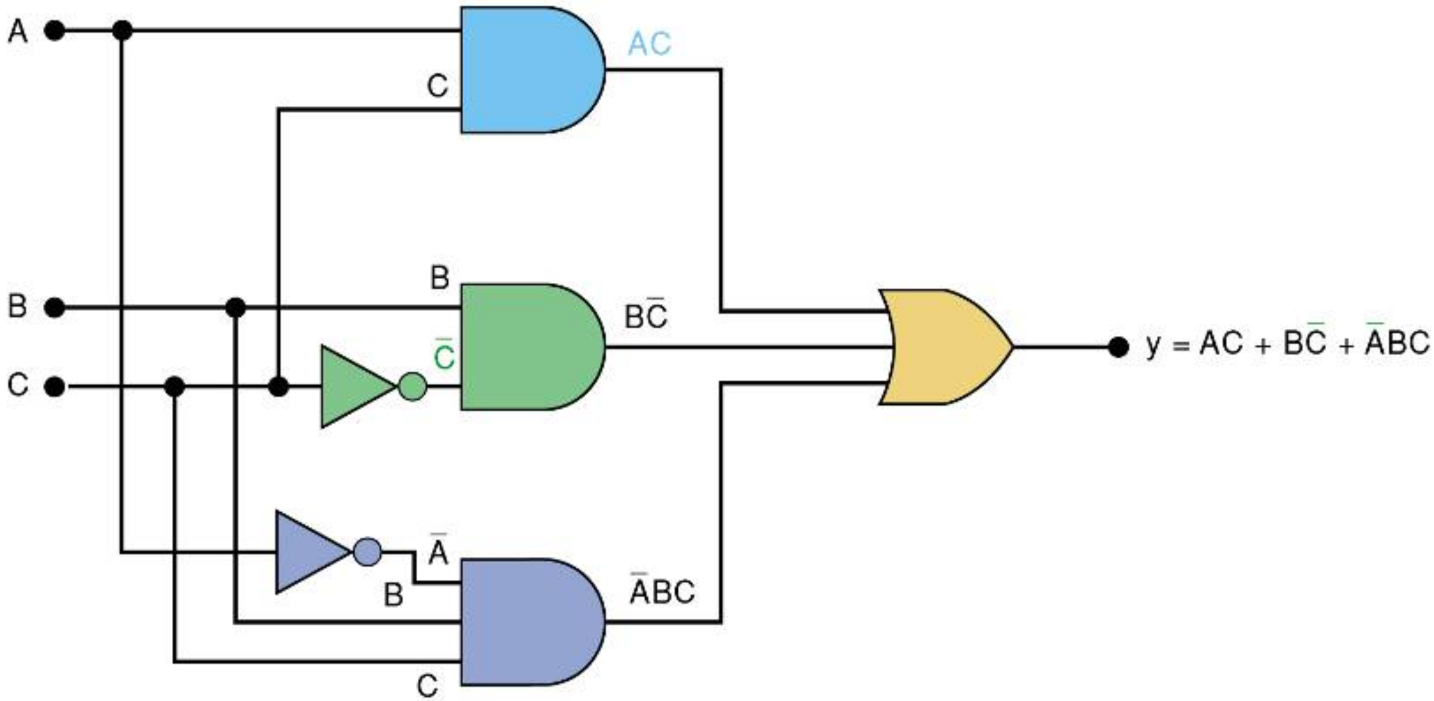
Um circuito com saída $\boldsymbol{y = AC + B\overline{C} + \overline{A}BC}$ contém três termos unidos por um **OR**.



…e exige uma porta **OR** com três entradas.

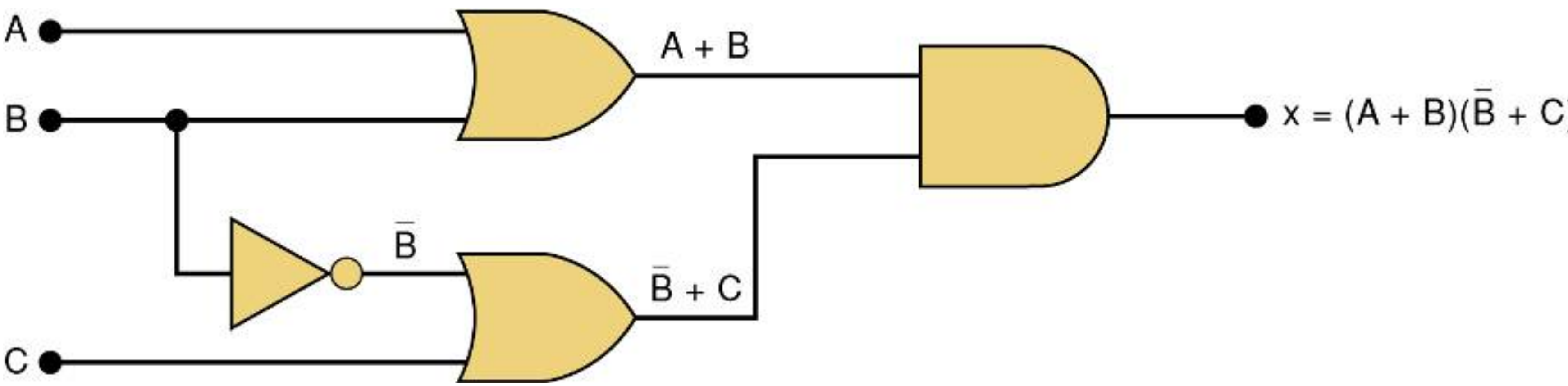
Cada entrada da porta OR é um termo do produto AND.

- Uma porta AND com entradas apropriadas pode ser usada para gerar cada um destes termos .



3-8 Circuitos a partir de Expressões Booleanas

Circuito que implementa $\boldsymbol{x = (A + B)(\overline{B} + C)}$



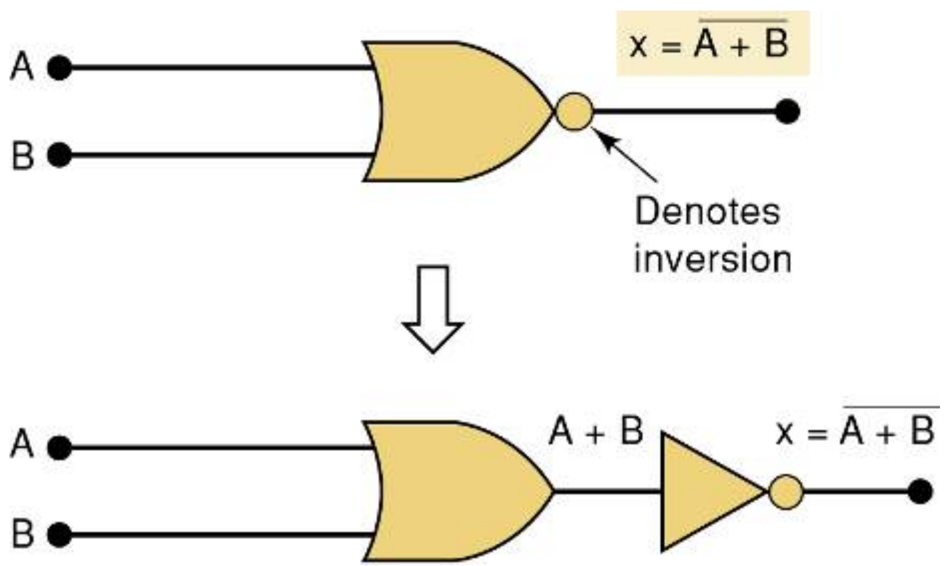
Exercício

3-9 Portas NAND e NOR

- Combinam a operações básicas **AND**, **OR**, **e NOT**.
  - Simplificando a escrita de expressões booleanas.
- A saída de NAND e NOR pode ser obtida através da determinação da saída de um AND ou OR, invertendo-a posteriormente.
- As tabelas verdade para NOR e NAND são o complemento das tabelas verdade para OU e AND.

3-9 Portas NAND e NOR

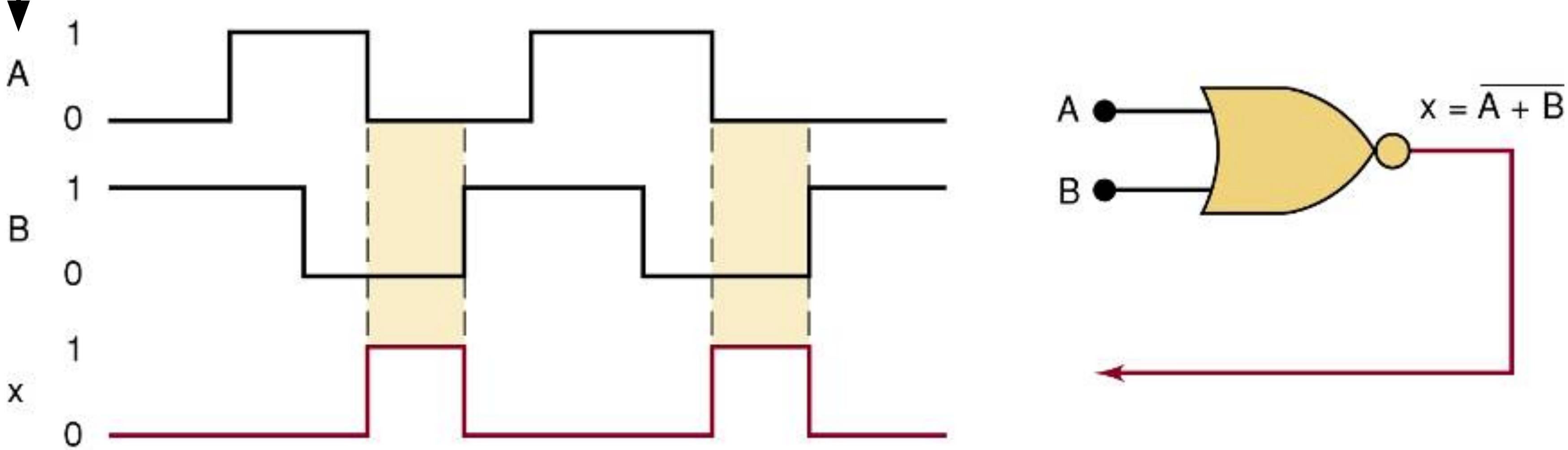
- A porta **NOR** é uma **OR** invertida.
- Um inversor ("bolha") é colocado na saída da porta OR, descrevendo a expressão de saída booleana $\boldsymbol{x} = \overline{\boldsymbol{A} + \boldsymbol{B}}$



		OR	NOR
A	B	$A + B$	$\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

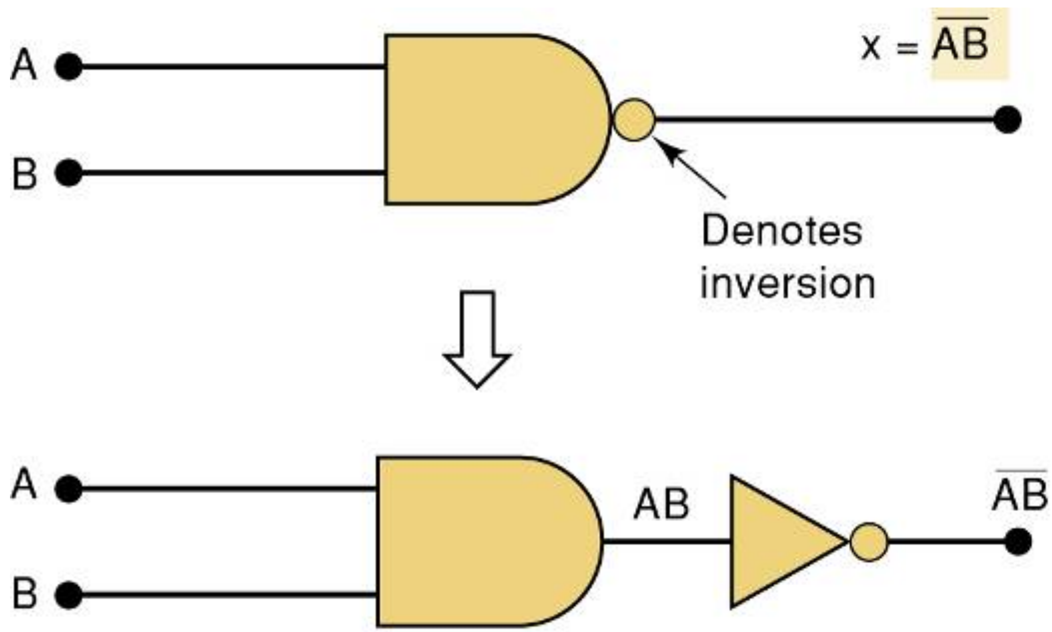
3-9 Portas NAND e NOR

Forma de onda de saída de uma porta NOR para as formas de onda de entrada mostradas aqui.



3-9 Portas NAND e NOR

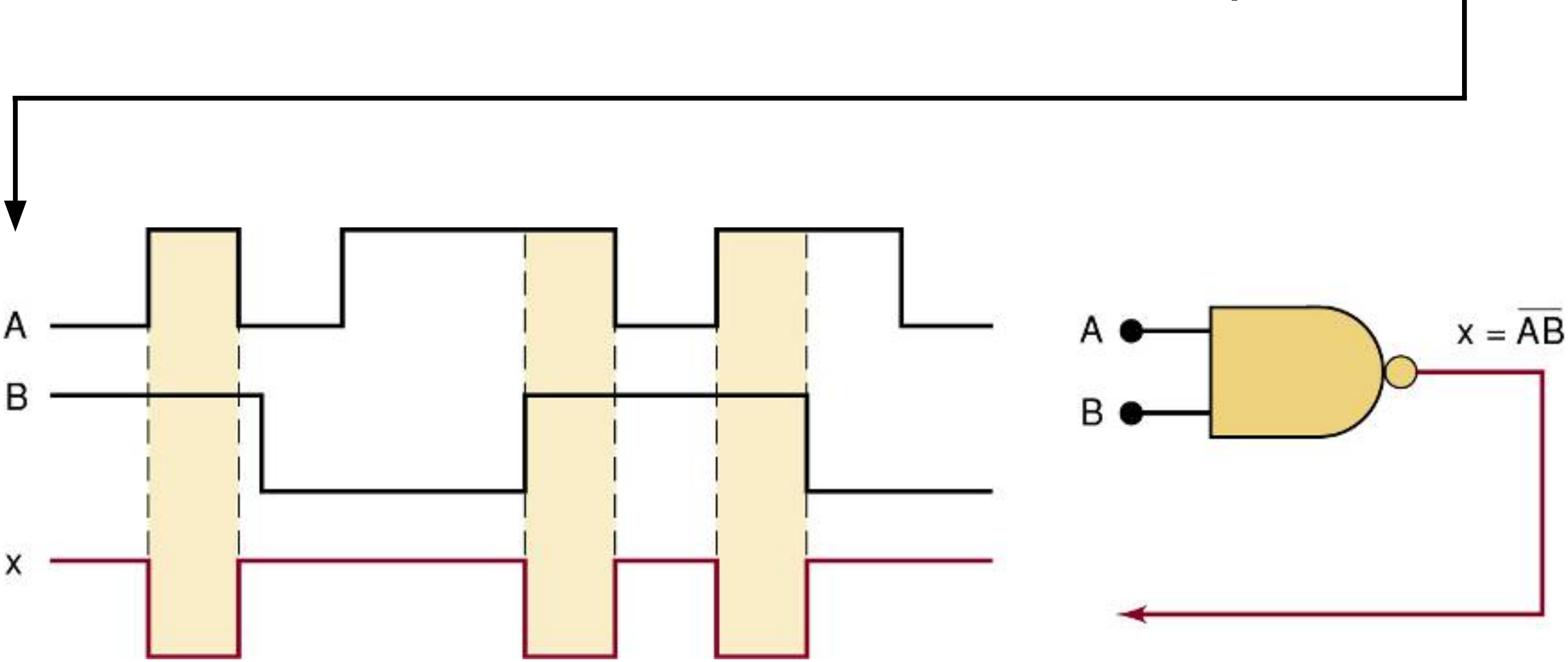
- A porta **NAND** é uma **AND** invertida.
- Um inversor é colocado na saída da porta AND, descrevendo a expressão de saída booleana ***x*** = $\overline{AB}$



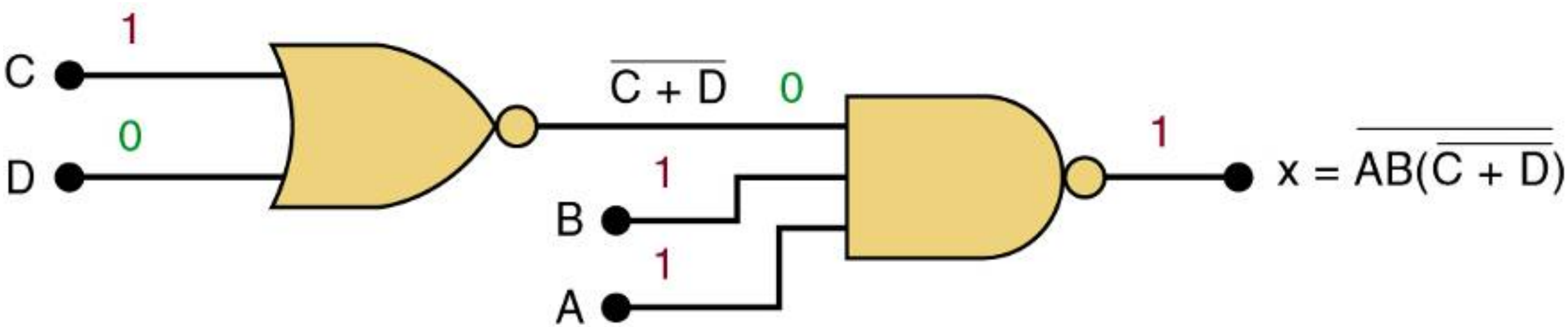
		AND	NAND
A	B	AB	$\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

3-9 Portas NAND e NOR

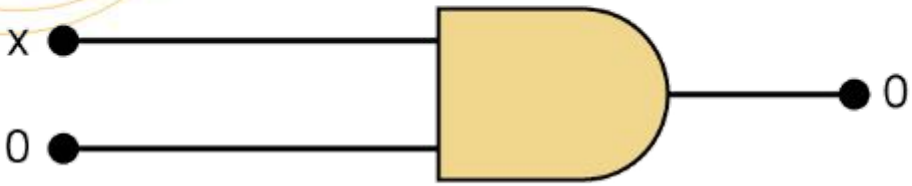
Forma de onda de saída de uma porta NAND para as formas de onda de entrada mostradas aqui



Circuito lógico para a expressão $x = \overline{AB \cdot (\overline{C + D})}$ usando somente portas **NOR** e **NAND**.



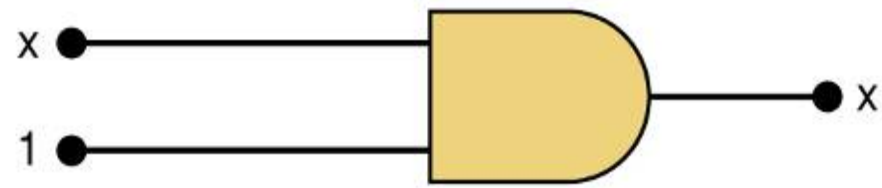
3-10 Teoremas Booleanos



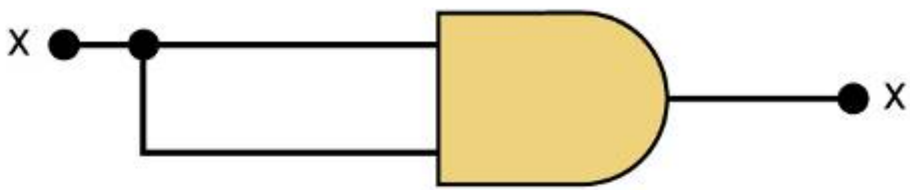
(1) $x \cdot 0 = 0$

Teorema (1): para toda a entrada juntamente com 0 numa porta AND, o resultado será 0.

Teorema (2): também pode ser comparado a uma multiplicação.



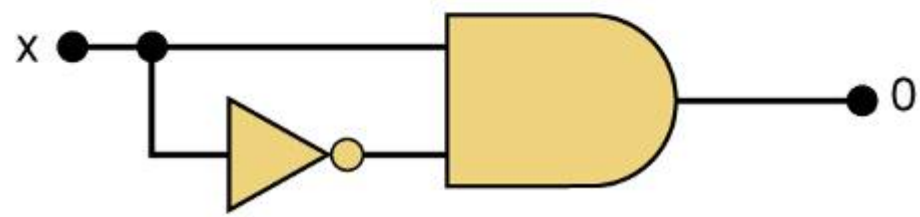
(2) $x \cdot 1 = x$



(3) $x \cdot x = x$

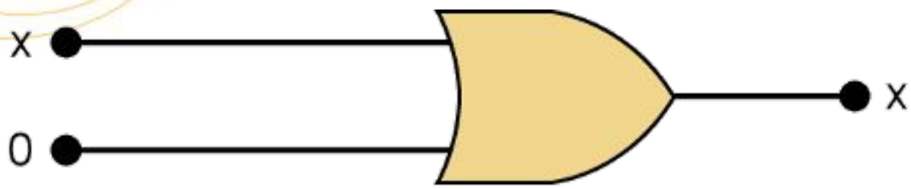
Teorema (3): pode ser provado testando-se caso a caso.
Se *x* = 0, então 0 • 0 = 0
Se *x* = 1, então 1 • 1 = 1
Assim, *x* • x = x

Teorema (4): pode ser provado da mesma forma.



(4) $x \cdot \overline{x} = 0$

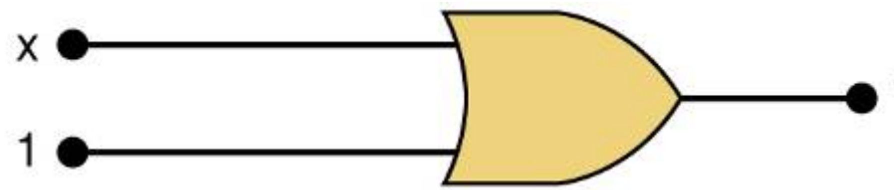
3-10 Teoremas Booleanos



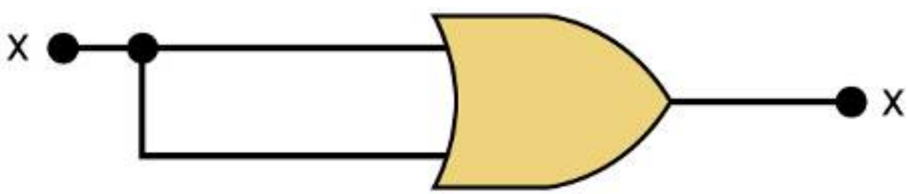
(5) $x + 0 = x$

Teorema (5): o 0 *adicionado* a qualquer valor não afeta este valor, seja numa adição ou numa operação OR.

Teorema (6): para qualquer variável numa operação OR com 1, o resultado será 1. Verifique: 0 + 1 = 1 e 1 + 1 = 1.



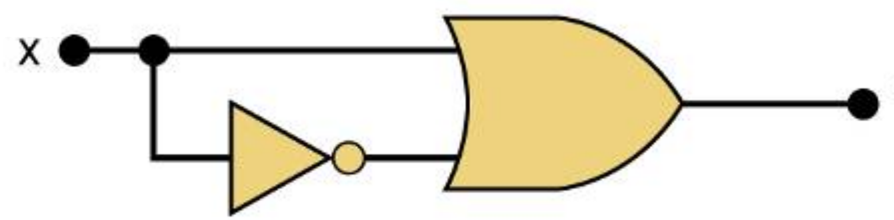
(6) $x + 1 = 1$



(7) $x + x = x$

Teorema (7): pode ser provado verificando-se os dois valores de x. 0 + 0 = 0 e 1 + 1 = 1.

Teorema (8): pode ser verificado de forma similar.



(8) $x + \overline{x} = 1$

Teoremas Multivariáveis

Leis comutativas

(9) $x + y = y + x$

(10) $x \cdot y = y \cdot x$

Leis Associativas

(11) $x + (y + z) = (x + y) + z = x + y + z$

(12) $x(yz) = (xy)z = xyz$

Lei Distributiva

(13*a*) $x(y + z) = xy + xz$

(13*b*) $(w + x)(y + z) = wy + xy + wz + xz$

Teoremas Multivariáveis

Teoremas (14) e (15) não possuem similares na álgebra tradicional. Podem ser provados verificando-se todos os possíveis valores para *x* e *y*.

$$(14) \quad x + xy = x$$
$$(15a) \quad x + \bar{x}y = x + y$$
$$(15b) \quad \bar{x} + xy = \bar{x} + y$$

Tabela de análise para o Teorema (14)

$$\begin{aligned} x + xy &= x(1 + y) \\ &= x \cdot 1 && \text{[using theorem (6)]} \\ &= x && \text{[using theorem (2)]} \end{aligned}$$

x	y	xy	x + xy
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

3-11 Teoremas de DeMorgan

Teoremas de DeMorgan são extremamente úteis na simplificação expressões em que um produto ou soma das variáveis está invertido.

$$(16) \quad \overline{(x + y)} = \bar{x} \cdot \bar{y}$$

Teorema (16) diz que a inversão da soma OR de duas variáveis é o mesmo que inverter cada variável individualmente e aplicar AND nas variáveis invertidas.

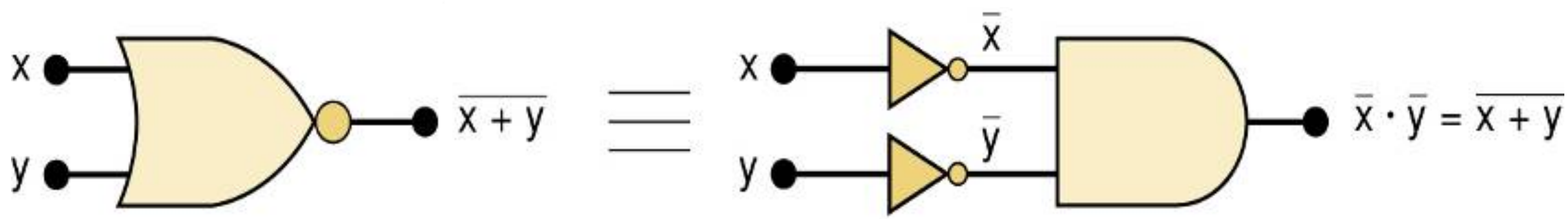
$$(17) \quad \overline{(x \cdot y)} = \bar{x} + \bar{y}$$

Teorema (17) diz que a inversão do produto E de duas variáveis é o mesmo que inverter cada variável individualmente e, em seguida, reuni-las num OR.

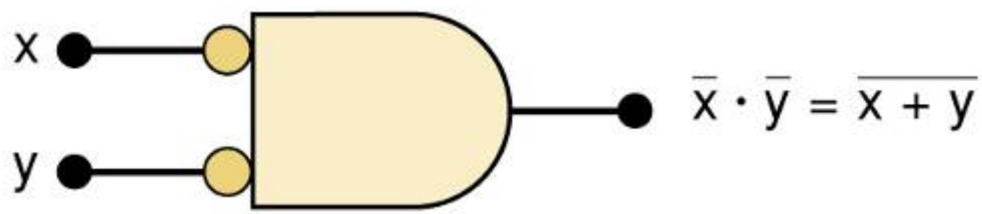
Cada teorema de DeMorgan pode ser facilmente comprovado pela verificação de todas as combinações possíveis de x e y.

Circuitos equivalentes pelo Teorema (16)

$$(16) \quad \overline{(x + y)} = \bar{x} \cdot \bar{y}$$

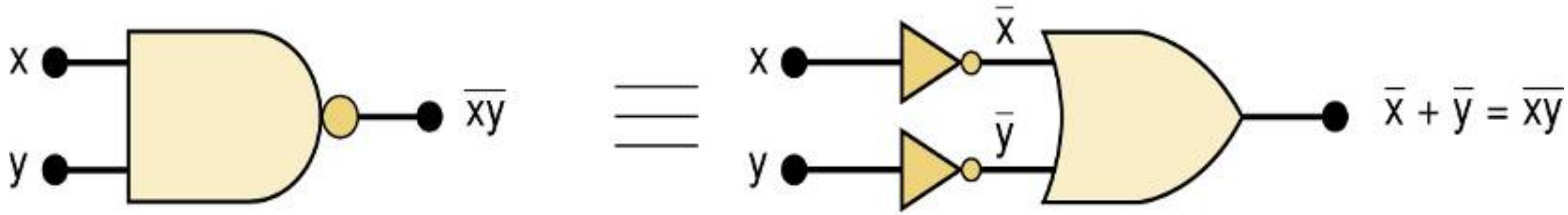


O símbolo alternativo para a função NOR.

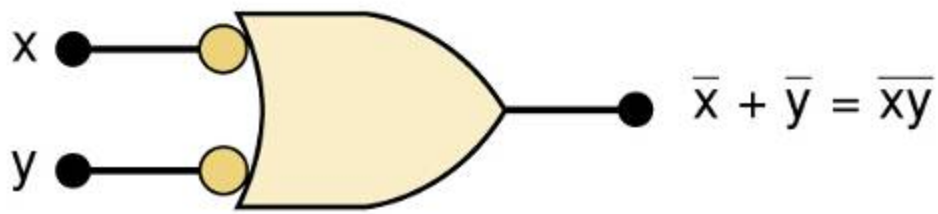


Circuitos equivalentes pelo Teorema (17)

$$(17) \quad \overline{(x \cdot y)} = \bar{x} + \bar{y}$$



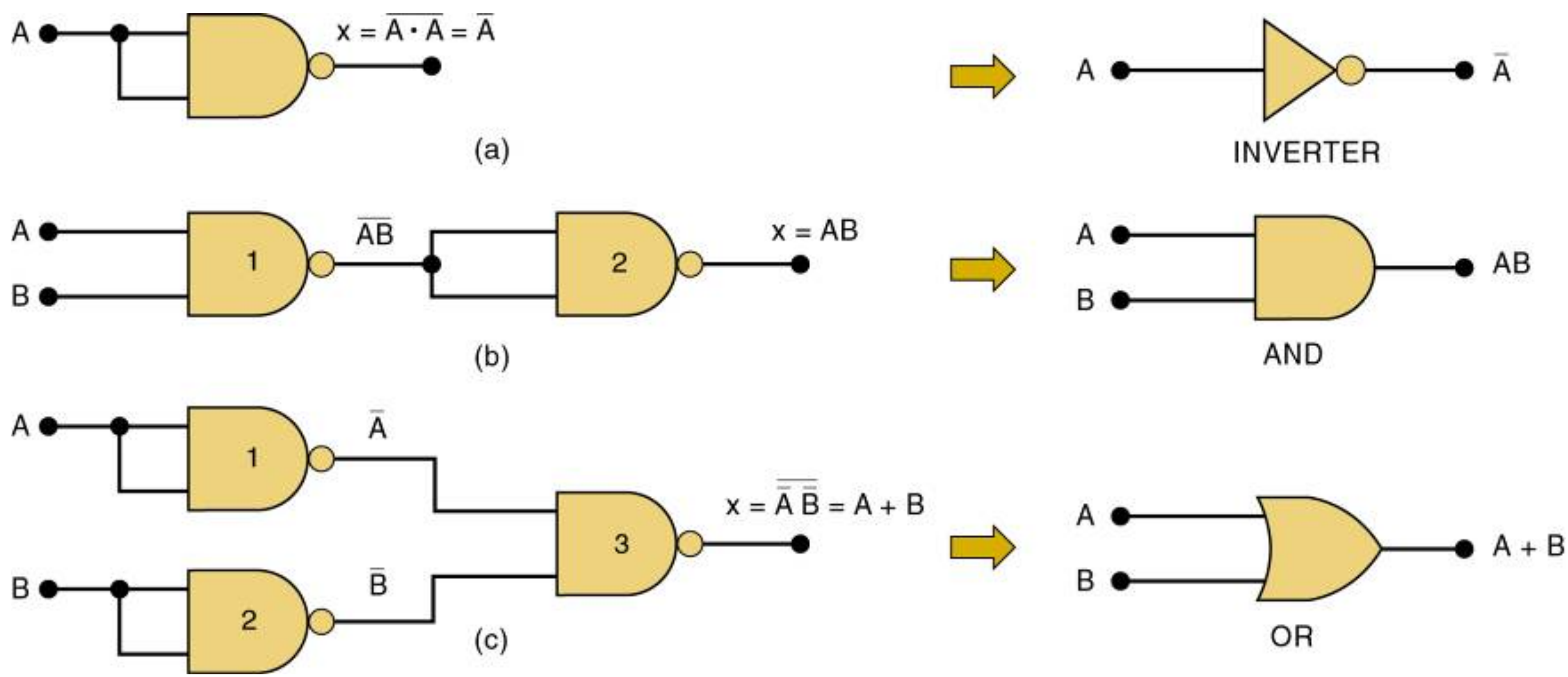
O símbolo alternativo para a função NAND.



kahoot.it

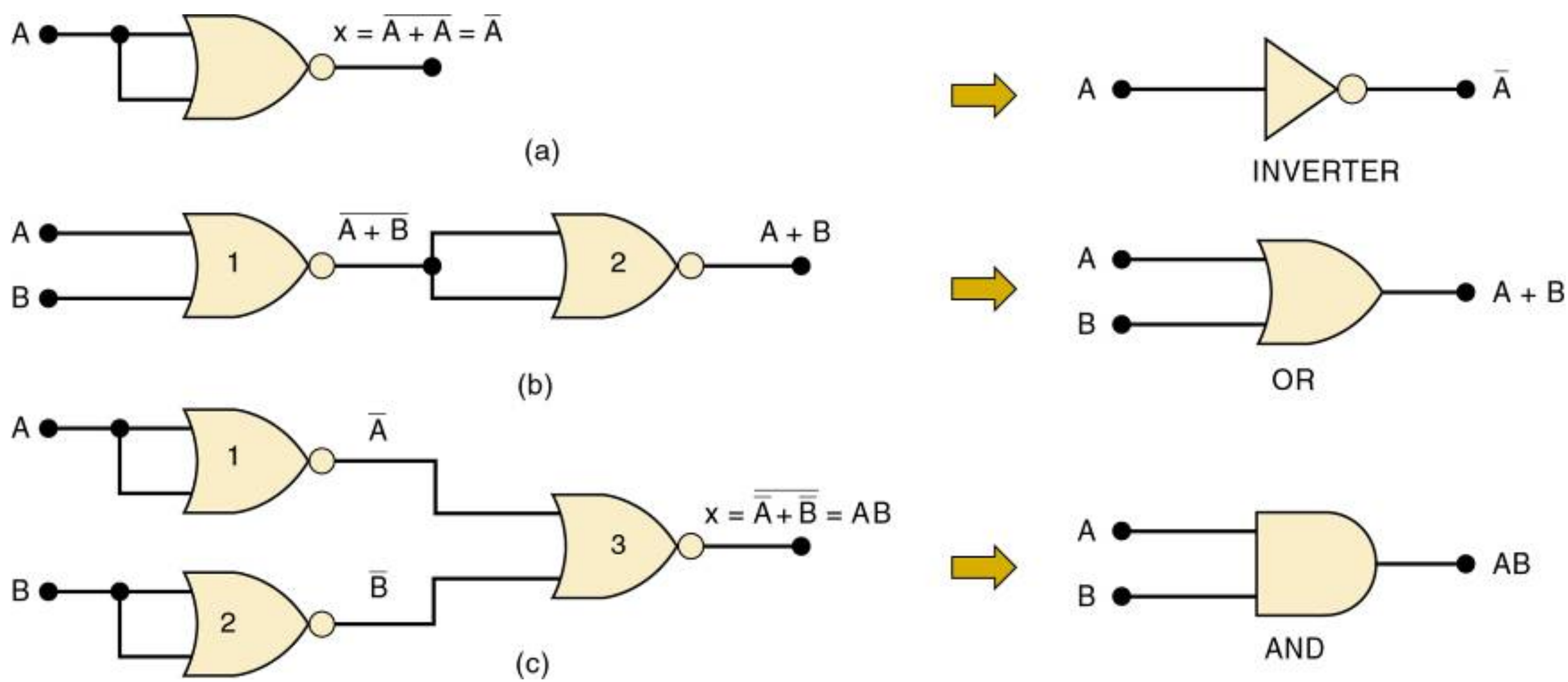
- Portas NAND e NOR podem ser usadas para criar as três expressões lógicas básicas.
  - OR, AND, e NOT (inversão).
    - Fornecem flexibilidade - muito útil no projeto de circuito lógicos.

Como combinações de NANDs ou NORs são utilizadas para criar as três funções lógicas.



É possível, portanto, implementar qualquer expressão lógica utilizando apenas portas NAND e nenhum outro tipo de porta, como mostrado.

Como combinações de NANDs ou NORs são utilizadas para criar as três funções lógicas.



NOR pode ser usada para implementar quaisquer das operações booleanas, como mostrado.

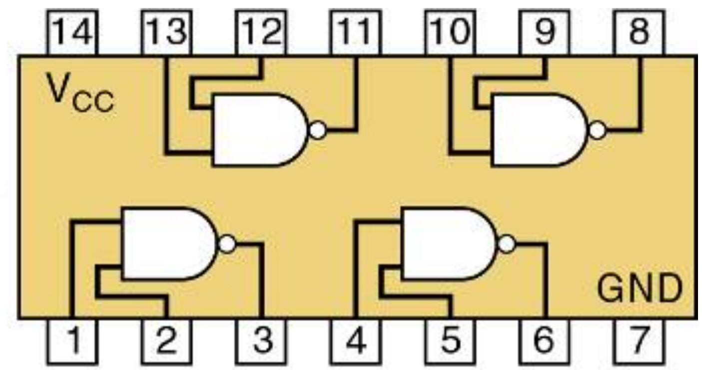
3-12 Universalidade das portas NAND e NOR

Um circuito lógico para gerar um sinal *x*, que será alto sempre que as condições A e B existirem simultaneamente, ou sempre que as condições C e D existirem simultaneamente.

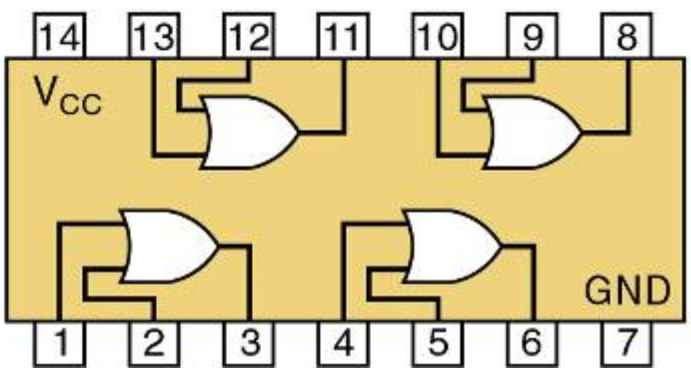
Expressão lógica: $x = AB + CD$.

Cada um dos CIs TTL mostrados aqui será usado para esta função. Cada CI é um *quad*, com quatro portas idênticas em um único *chip*

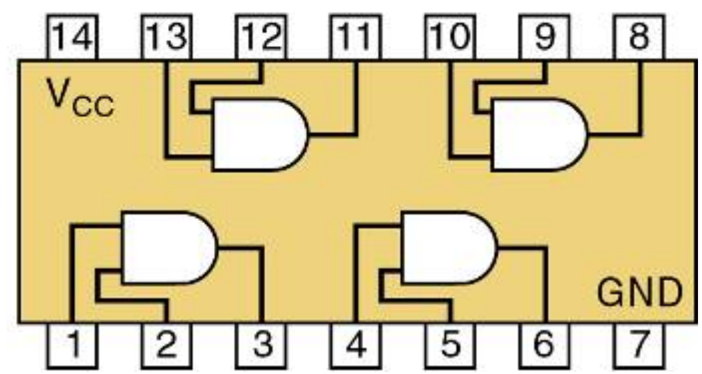
74LS00



74LS32

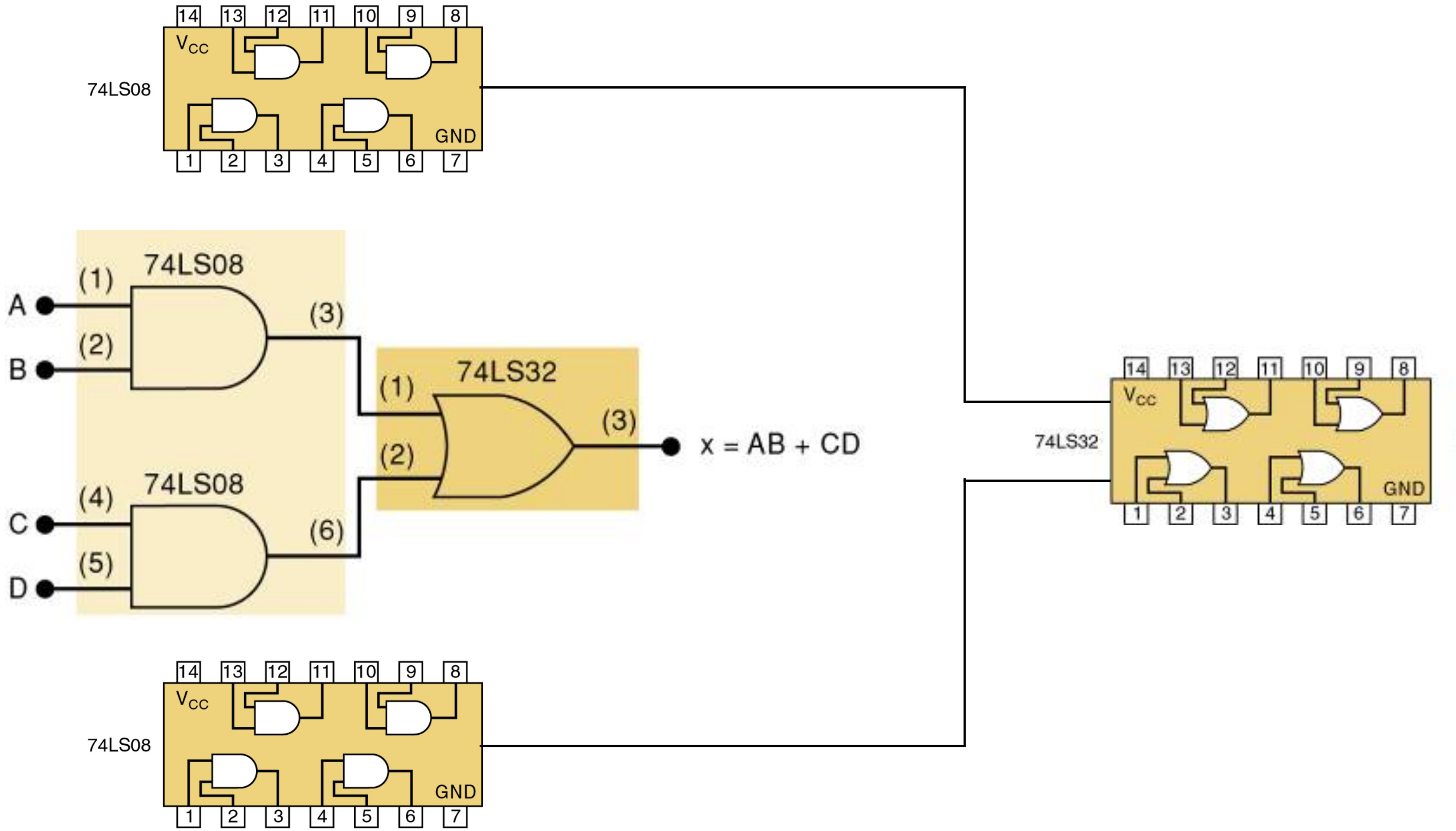


74LS08



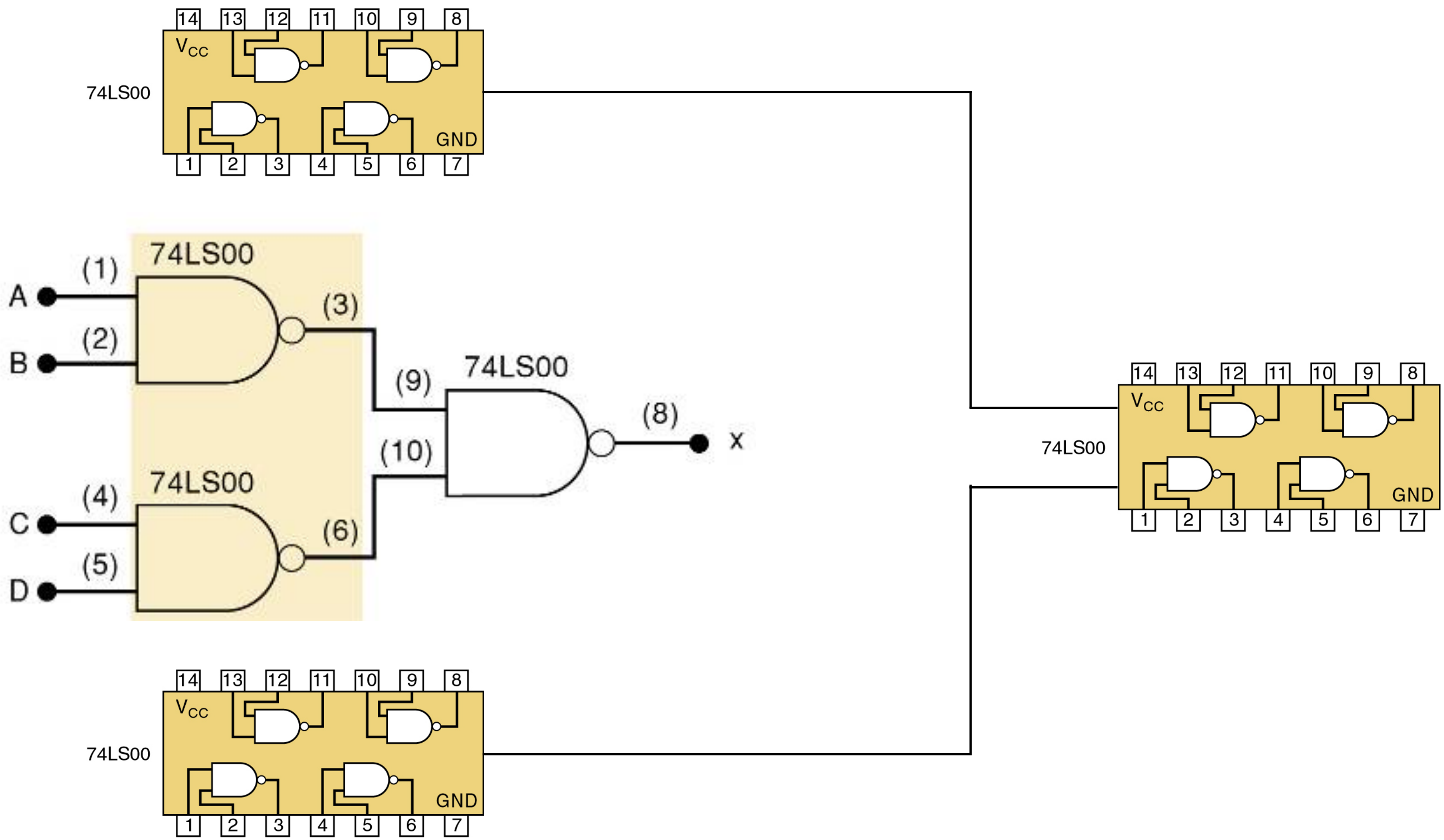
3-12 Universalidade das portas NAND e NOR

Possível implementação # 1



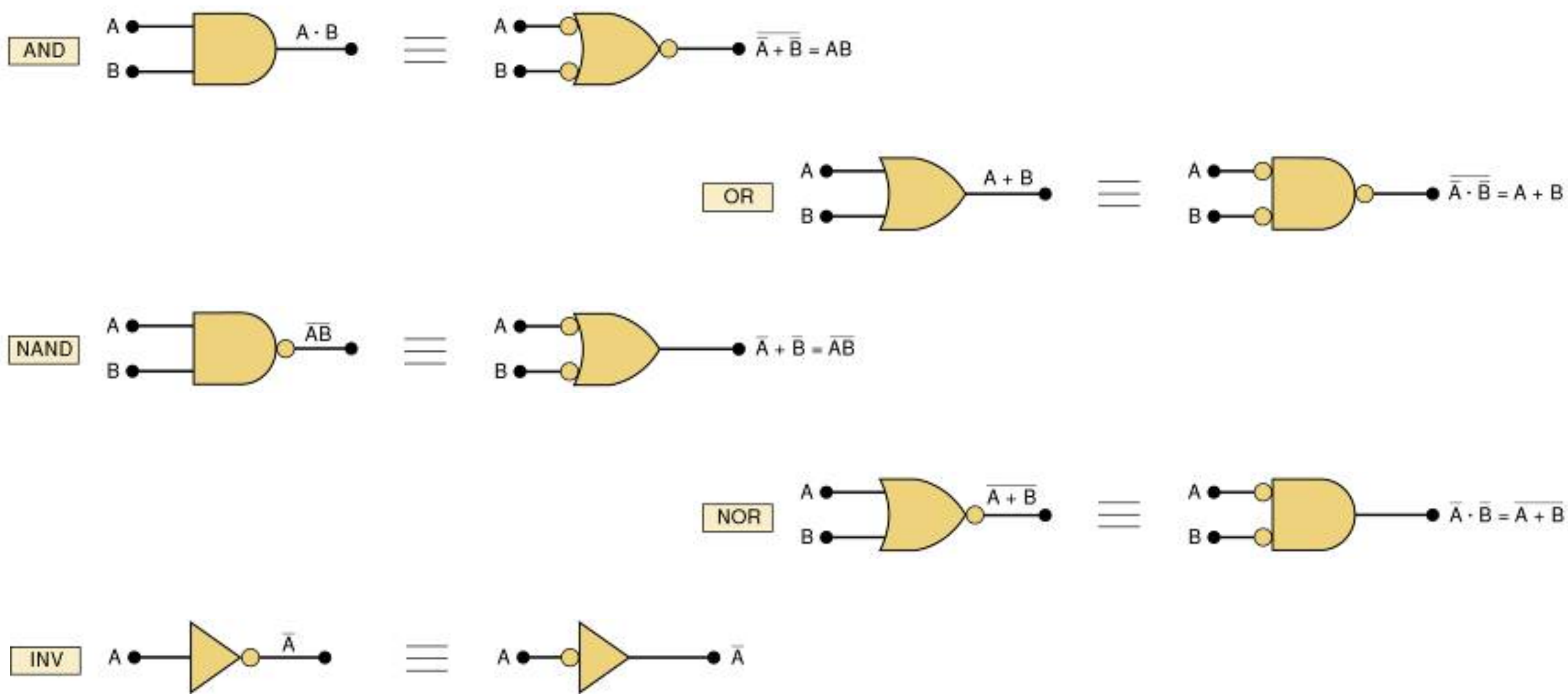
3-12 Universalidade das portas NAND e NOR

Possível implementação #2



3-13 Representações alternativas para as portas lógicas

- Para converter um símbolo padrão em um alternativo:
    - Inverter cada **entrada** e **saída** dos símbolos padrão.
        - Adicionar uma bolha de inversão, onde não houver.
        - Remover as bolhas de onde elas existirem.



3-13 Representações alternativas para as portas lógicas

Pontos relativos às equivalências dos símbolos lógicos:

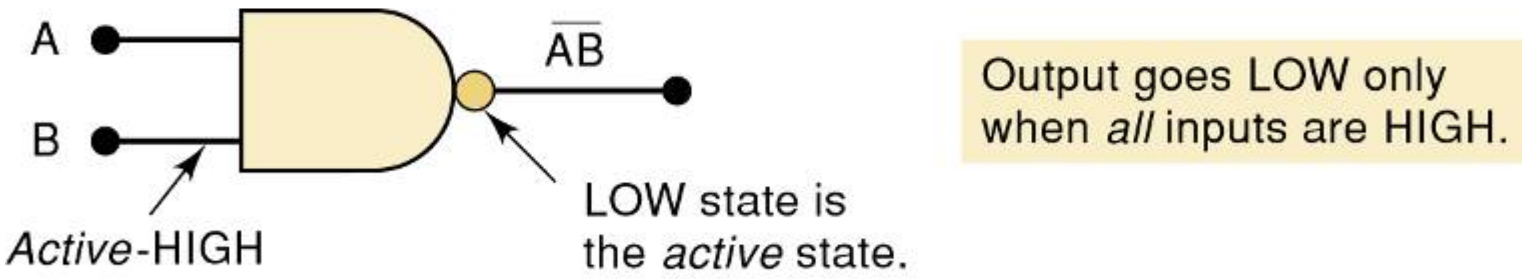
- As equivalências podem ser generalizadas para portas com qualquer número de entradas.
- Nenhum dos símbolos padrão têm bolhas em suas entradas, e todos os símbolos alternativos têm.
- Símbolos alternativos e padrão para cada porta representam o mesmo circuito físico.
- Portas **NAND** e **NOR** são portas inversoras.
  - Tanto os símbolos padrão quanto os alternativos para cada porta terão uma bolha na entrada **ou** na saída.
- Portas **AND** e **OR** são não-inversoras.
  - Os símbolos alternativos para cada porta terão bolhas em ambas as entradas e saída.

3-13 Representações alternativas para as portas lógicas

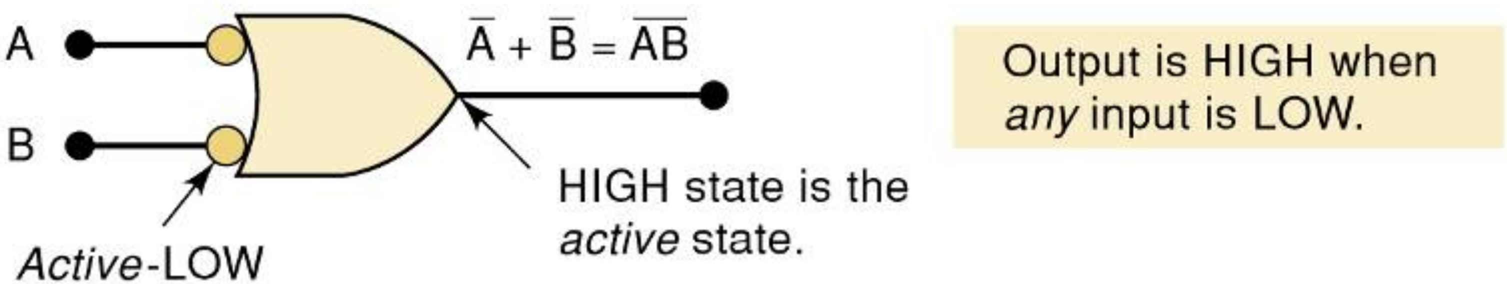
- Ativa em ALTA (*Active-HIGH*) – uma entrada/saída que **não possui** bolha inversora.

- Ativa em BAIXA (*Active-LOW*) – uma entrada/saída que **possui** bolha inversora.

Interpretação para os dois símbolos para a porta **NAND**.

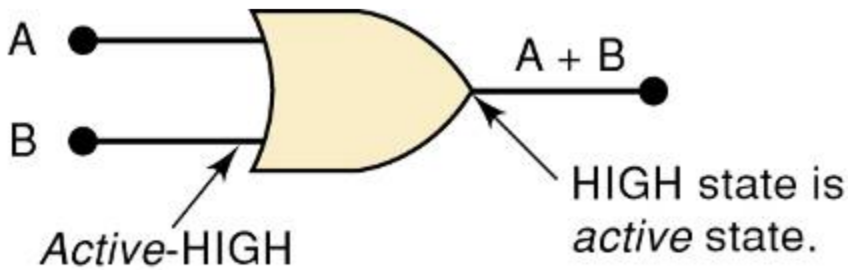


(a)

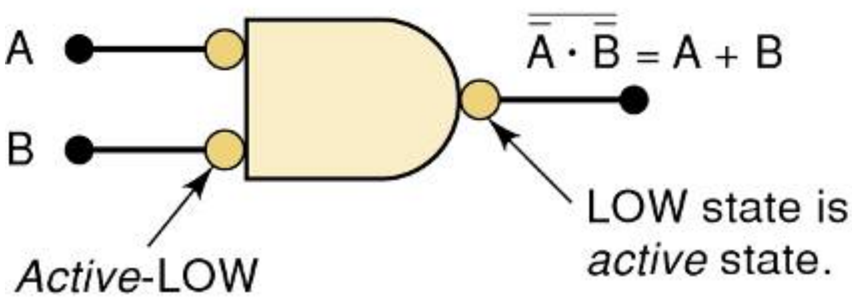


(b)

Interpretação para os dois símbolos para a porta **OR**.



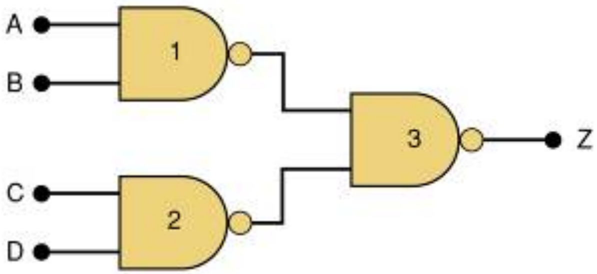
Output goes HIGH when *any* input is HIGH.



Output goes LOW only when *all* inputs are LOW.

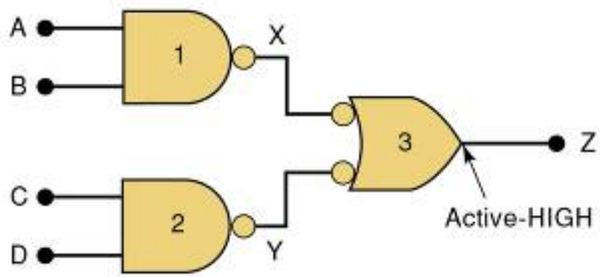
3-14 Qual representação usar

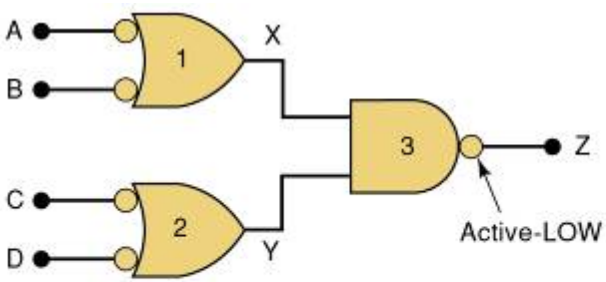
O uso adequado dos símbolos alternativos no esquema de circuitos pode deixar a operação do circuito muito mais clara.



Circuito original usando símbolos **NAND** padrão.

Representação equivalente onde a saída Z é ativa-em-ALTA





Representação equivalente onde a saída Z é ativa-em-BAIXA.

A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

3-14 Qual representação usar

- Quando possível, escolha símbolos de portas tal que saídas com inversores (bolhas) estão conectadas à entrada com inversores.
  - E saídas sem inversores ligadas a entradas sem inversores.

3-14 Qual representação usar

Uma barra sobre um sinal significa que ele á ativo em BAIXA.	$\overline{\boldsymbol{RD}}$	A ausência de barra indica que ele é ativo em ALTA.	$\boldsymbol{RD}$

3-14 Qual representação usar

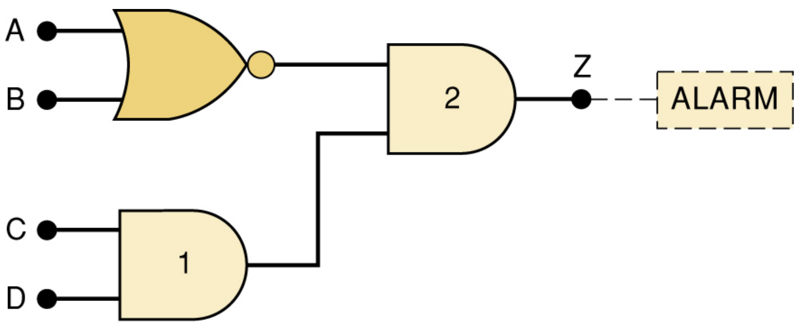
- Um sinal de saída pode ter dois estados ativos, com uma função importante no estado ALTO, e outra no estado BAIXO.
- É costume rotular esses sinais para que ambos os estados ativos sejam aparentes.

Um exemplo comum é o sinal de leitura/escrita.

$$RD/\overline{WR}$$

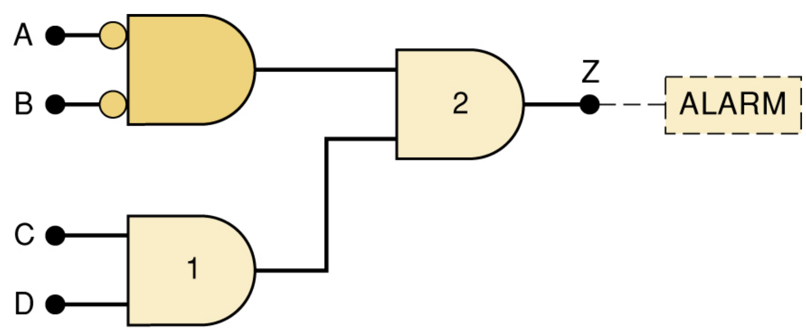
Quando este sinal é alto, a operação de leitura (RD) é realizada; quando ela está baixa, a operação de escrita ($\overline{WR}$) é executada.

O circuito lógico mostrado ativa um alarme quando a saída Z vai para nível ALTO.



Modifique o diagrama do circuito para que represente seu funcionamento de forma mais clara.

O símbolo NOR deve ser alterado para o símbolo alternativo com uma saída sem inversor (ativo-ALTA) para coincidir com a entrada da porta AND 2 sem inversor.



O circuito tem agora saídas sem inversão ligadas às entradas sem inversão da porta 2.

END

ELEVENTH EDITION

Digital Systems

Principles and Applications

Ronald J. Tocci
Monroe Community College

Neal S. Widmer
Purdue University

Gregory L. Moss
Purdue University

PEARSON