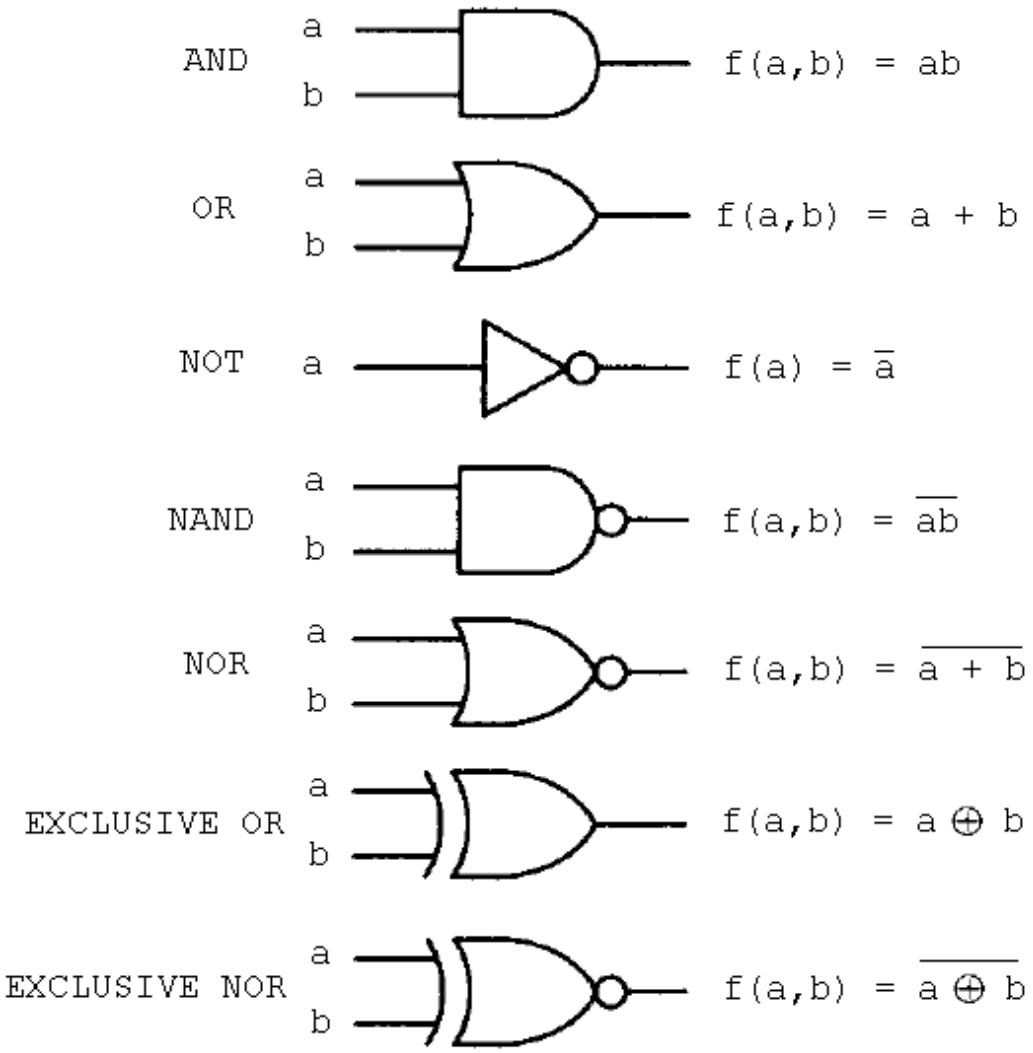
ELE 0316 / ELE 0937– Eletrônica Básica

Departamento de Engenharia Elétrica
FEIS - UNESP

Aula 07 : Portas Lógicas e Álgebra Booleana

7.1 – Portas Lógicas e Expressões Algébricas

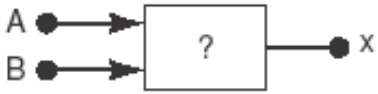


7.1 – Portas Lógicas e Tabela-Verdade

Saída

Entradas

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



(a)

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

(b)

A	B	C	D	x
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
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	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

(c)

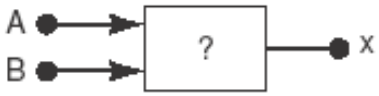
FIGURA 3.1
Exemplos de tabelas-verdade para circuitos de: (a) duas entradas, (b) três entradas e (c) quatro entradas.

7.1 – Portas Lógicas e Tabela-Verdade

Saída

Entradas

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



(a)

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

(b)

A	B	C	D	x
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
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	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

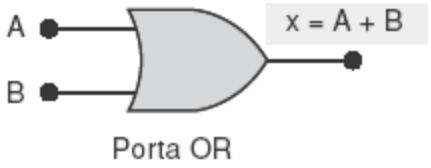
(c)

FIGURA 3.1
Exemplos de tabelas-verdade para circuitos de: (a) duas entradas, (b) três entradas e (c) quatro entradas.

OR

A	B	x = A + B
0	0	0
0	1	1
1	0	1
1	1	1

(a)

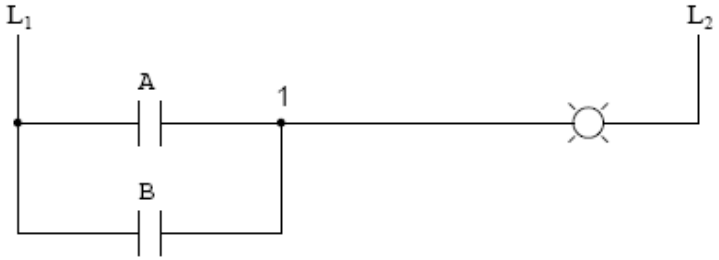


Porta OR

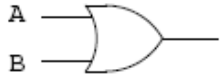
(b)

FIGURA 3.2
(a) Tabela-verdade que define a operação OR; (b) Símbolo de uma porta OR de duas entradas.

7.2 – Porta Lógica OR (OU ou ≥1)



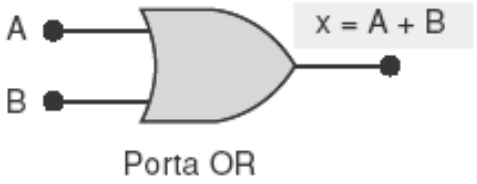
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1



OR

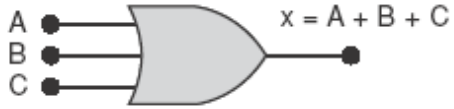
A	B	x = A + B
0	0	0
0	1	1
1	0	1
1	1	1

(a)



(b)

FIGURA 3.2
(a) Tabela-verdade que define a operação OR; (b) Símbolo de uma porta OR de duas entradas.



A	B	C	x = A + B + C
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

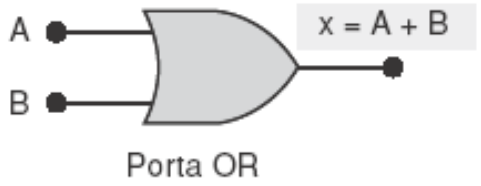
FIGURA 3.3
Símbolo e tabela-verdade para uma porta OR de três entradas.

7.2 – Porta Lógica OR (OU ou ≥1)

OR

A	B	x = A + B
0	0	0
0	1	1
1	0	1
1	1	1

(a)



Porta OR

(b)

FIGURA 3.2
(a) Tabela-verdade que define a operação OR; (b) Símbolo de uma porta OR de duas entradas.

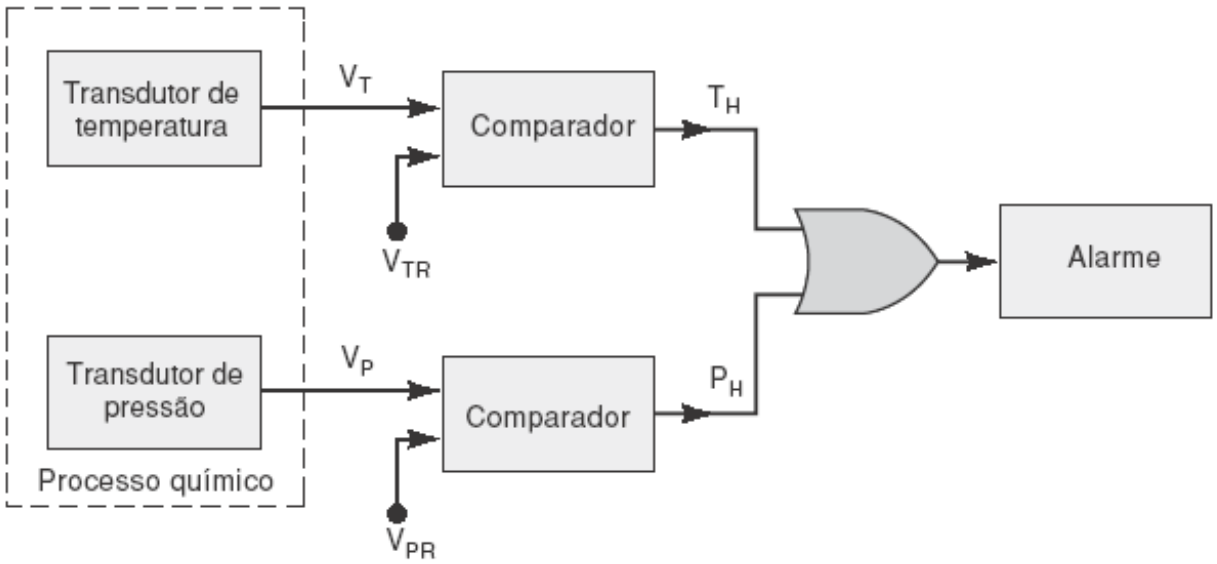
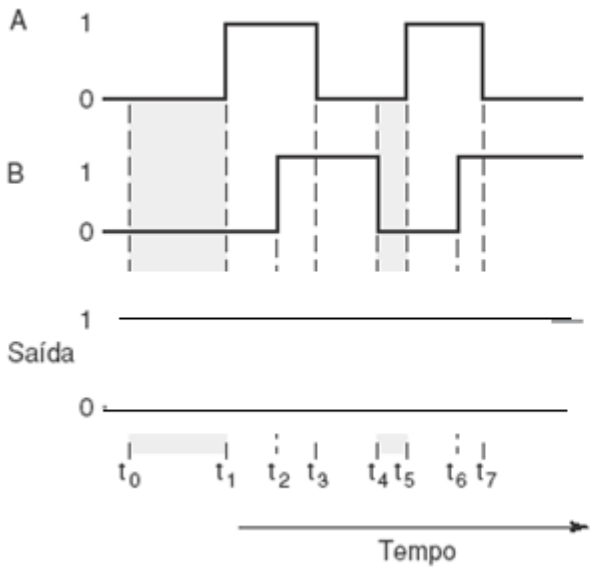


FIGURA 3.4 Exemplo do uso de uma porta OR em um sistema de alarme.

7.2 – Porta Lógica OR (OU ou ≥1)

Diagrama de temporização: 2 Entradas



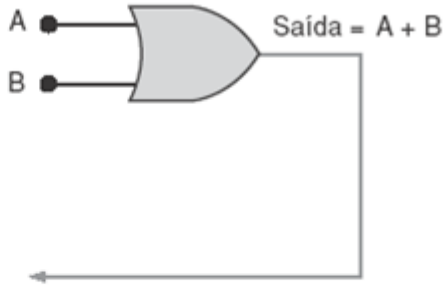
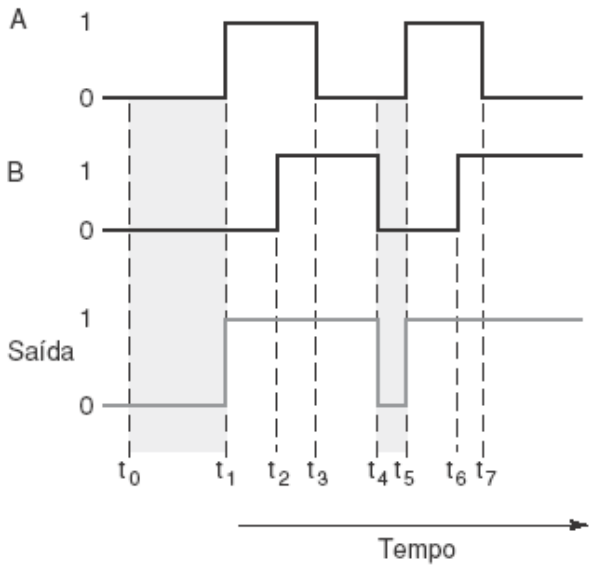


FIGURA 3.5
Exemplo 3.2.

7.2 – Porta Lógica OR (OU ou ≥1)

Diagrama de temporização: 2 Entradas



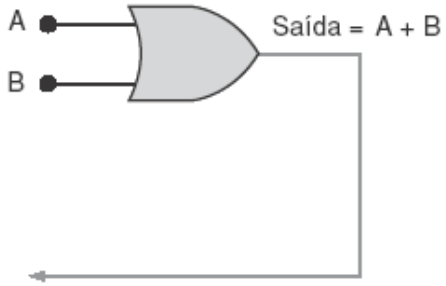
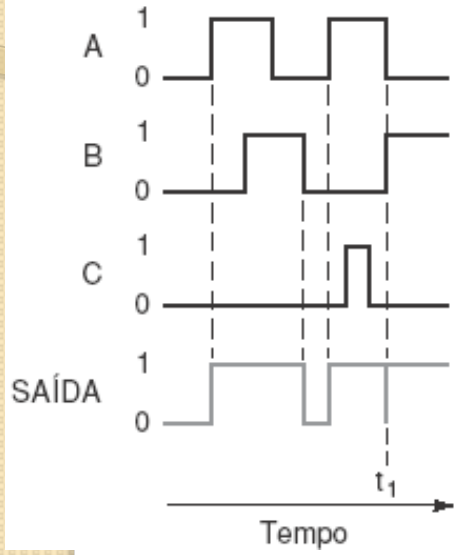


FIGURA 3.5
Exemplo 3.2.

7.2 – Porta Lógica OR (OU ou ≥1)

Diagrama de temporização: 3 Entradas



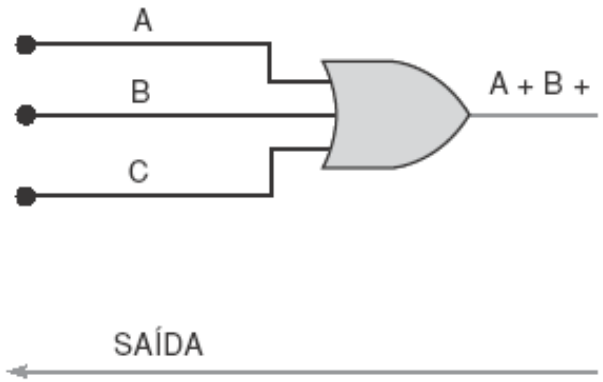


FIGURA 3.6
Exemplos 3.3A e 3.3B.

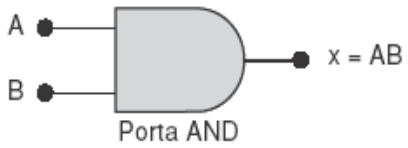
7.3 – Porta Lógica AND (E ou &)

FIGURA 3.7
(a) Tabela-verdade para a operação AND; (b) símbolo da porta AND.

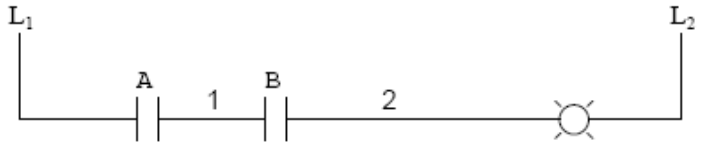
AND

A	B	x = A · B
0	0	0
0	1	0
1	0	0
1	1	1

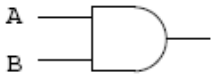
(a)



(b)



A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1



A	B	C	x = ABC
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

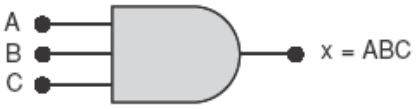


FIGURA 3.8
Tabela-verdade e símbolo para uma porta AND de três entradas.

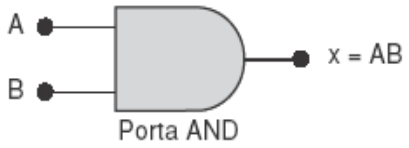
7.3 – Porta Lógica AND (E ou &)

FIGURA 3.7
(a) Tabela-verdade para a operação AND; (b) símbolo da porta AND.

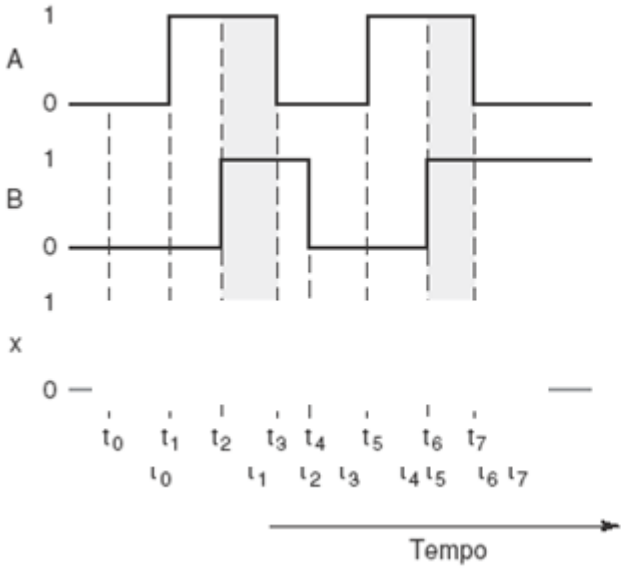
AND

A	B	x = A · B
0	0	0
0	1	0
1	0	0
1	1	1

(a)



(b)



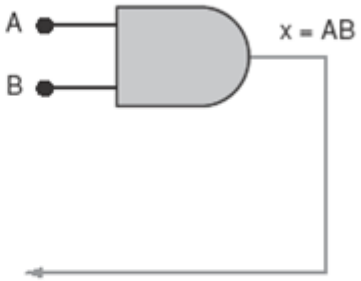


FIGURA 3.9
Exemplo 3.4.

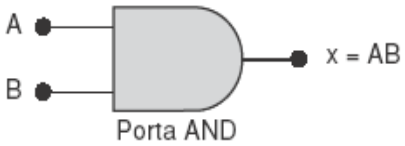
7.3 – Porta Lógica AND (E ou &)

FIGURA 3.7
(a) Tabela-verdade para a operação AND; (b) símbolo da porta AND.

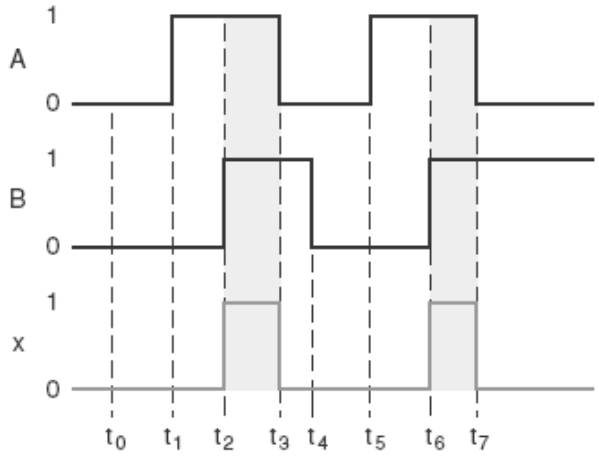
AND

A	B	x = A · B
0	0	0
0	1	0
1	0	0
1	1	1

(a)



(b)



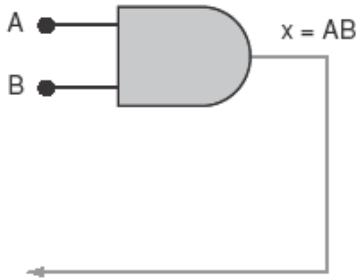


FIGURA 3.9
Exemplo 3.4.

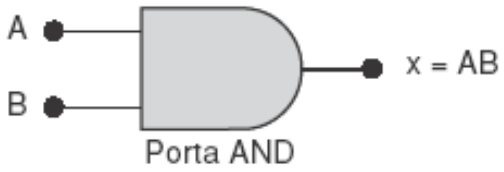
7.3 – Porta Lógica AND (E ou &)

FIGURA 3.7
(a) Tabela-verdade para a operação AND; (b) símbolo da porta AND.

AND

A	B	x = A · B
0	0	0
0	1	0
1	0	0
1	1	1

(a)



(b)

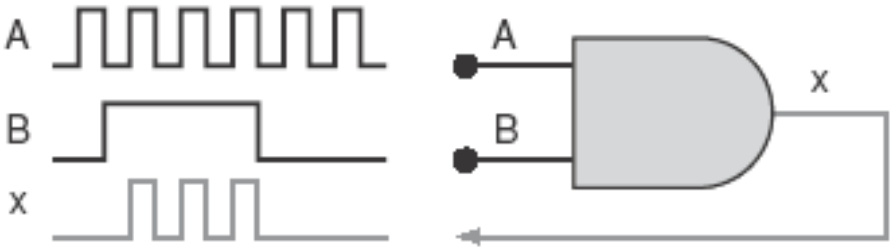


FIGURA 3.10
Exemplos 3.5A e 3.5B.

7.4 – Porta Lógica NOT (NÃO ou 1)

NOT

A	$x = \overline{A}$
0	1
1	0

(a)

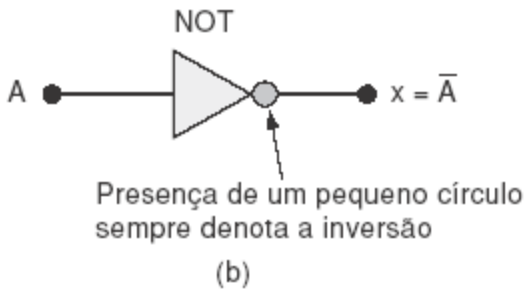


FIGURA 3.11
(a) Tabela-verdade; (b) Símbolo para o INVERSOR (circuito NOT); (c) Exemplos de formas de ondas.

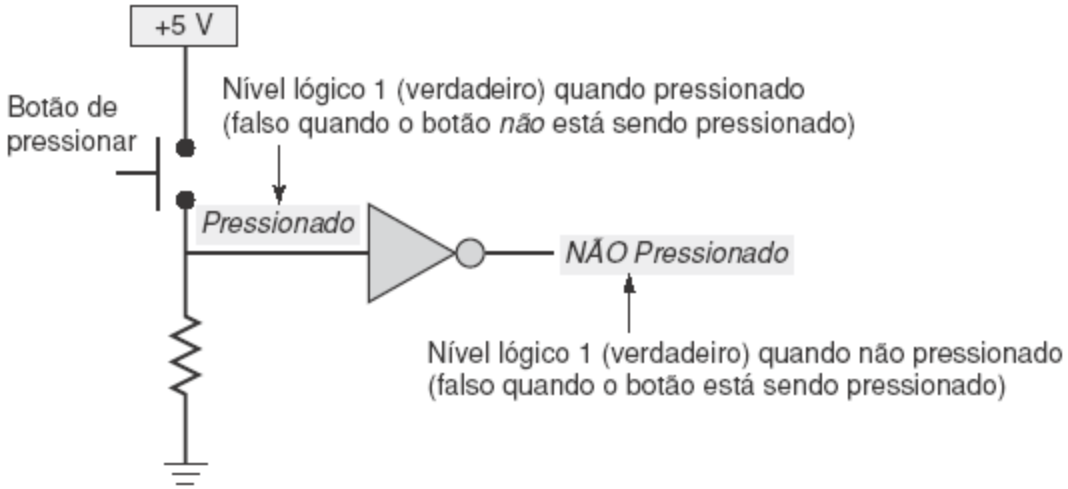


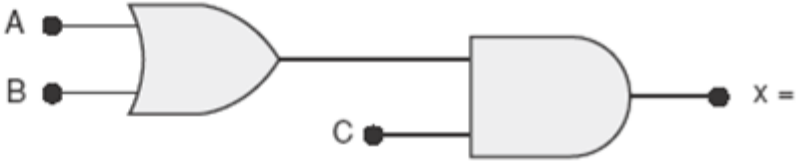
FIGURA 3.12
Uma porta NOT indicando que um botão *não* está pressionado quando a saída é verdadeira.

7.5 – Circuitos Lógicos e Expressões Booleanas

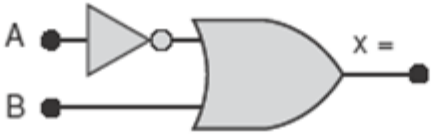
Obtenha a expressão de X.

(a)

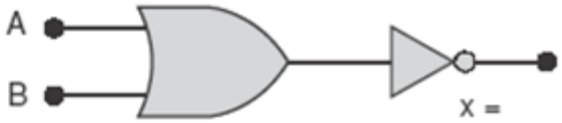
FIGURA 3.13
(a) Um circuito lógico e suas expressões booleanas;
(b) Circuito lógico com uma expressão que requer parênteses.



(b)



(a)



(b)

FIGURA 3.14
Circuitos com INVERSORES.

7.5 – Circuitos Lógicos e Expressões Booleanas

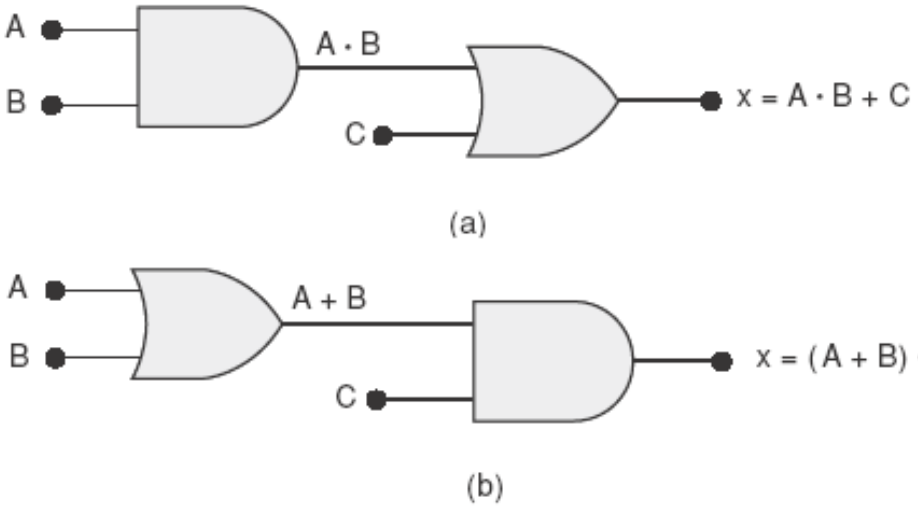


FIGURA 3.13
(a) Um circuito lógico e suas expressões booleanas;
(b) Circuito lógico com uma expressão que requer parênteses.

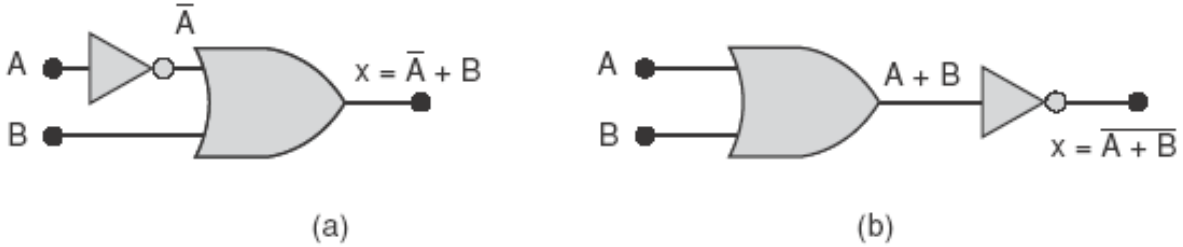
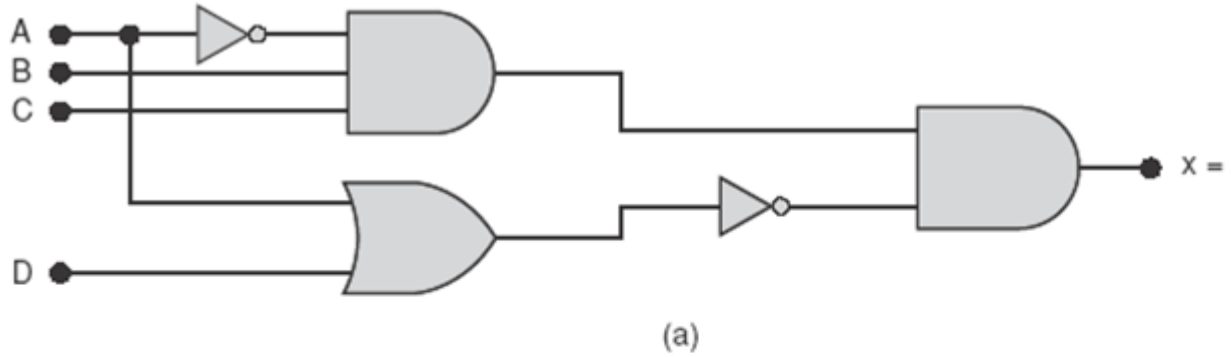


FIGURA 3.14
Circuitos com INVERSORES.

7.5 – Circuitos Lógicos e Expressões Booleanas



(a)

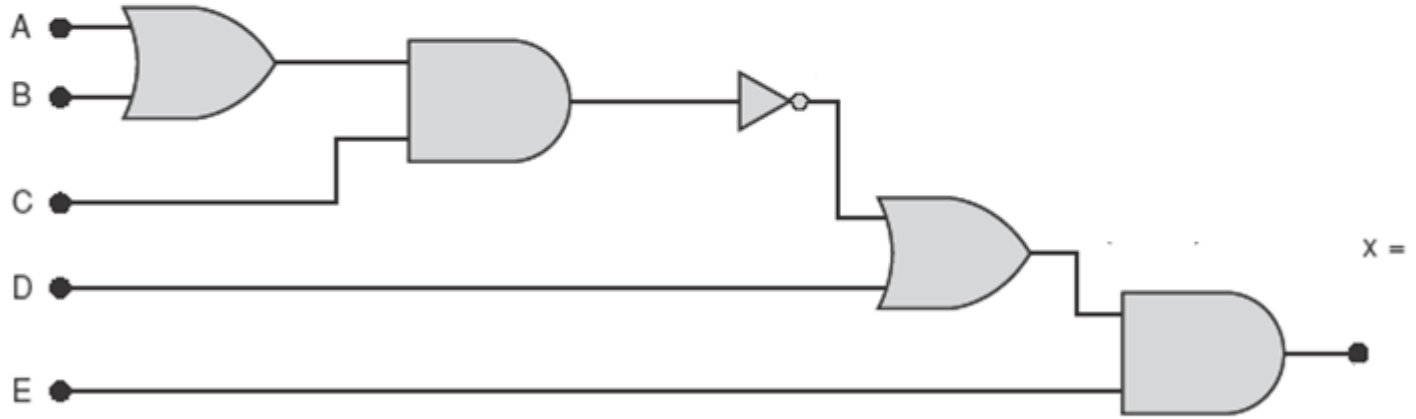
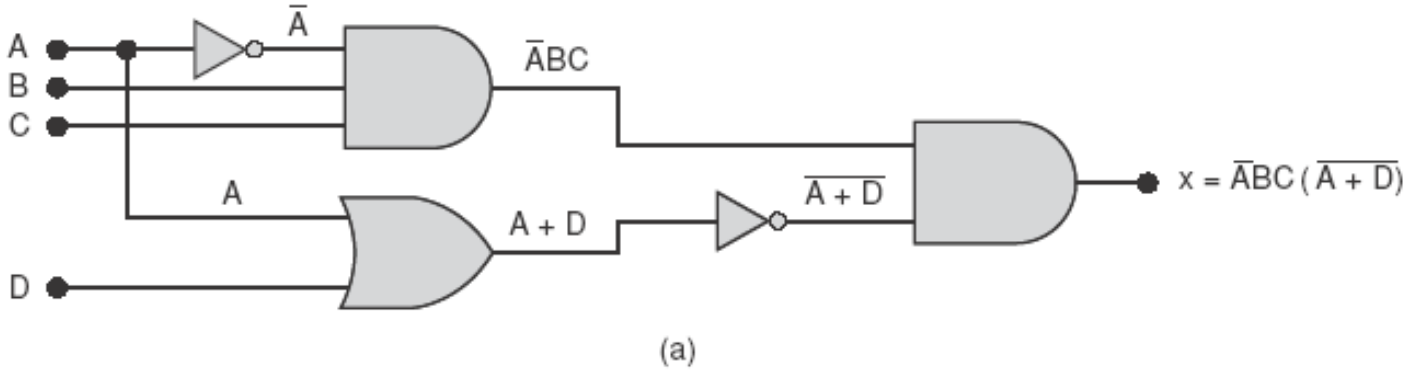
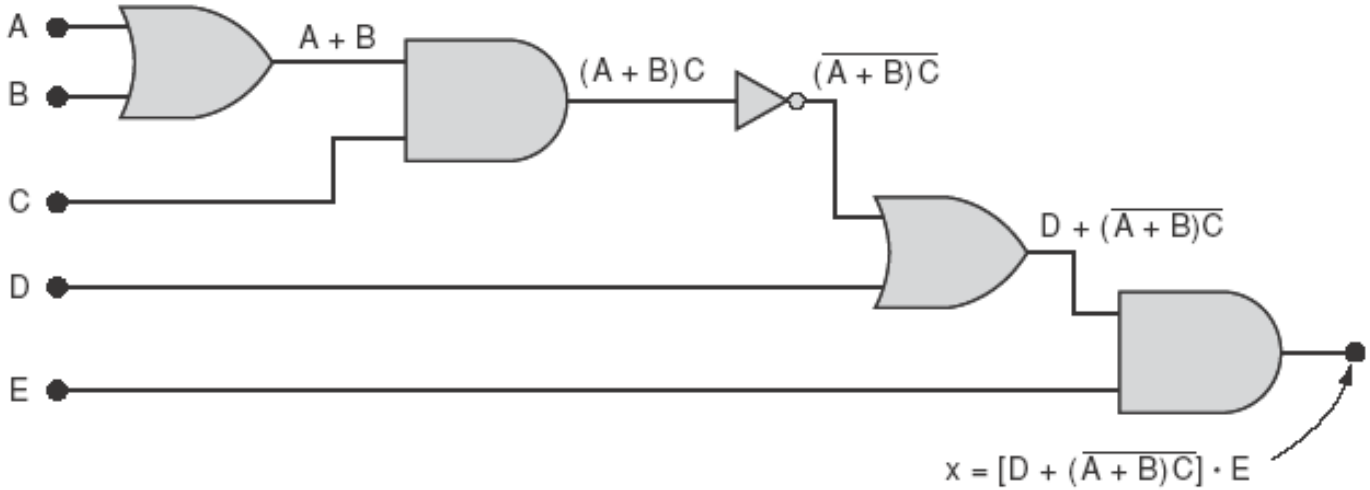


FIGURA 3.15 (b)

7.5 – Circuitos Lógicos e Expressões Booleanas



(a)



(b)

FIGURA 3.15
Mais exemplos.

7.6 – Resumo Geral das Principais Portas Lógicas

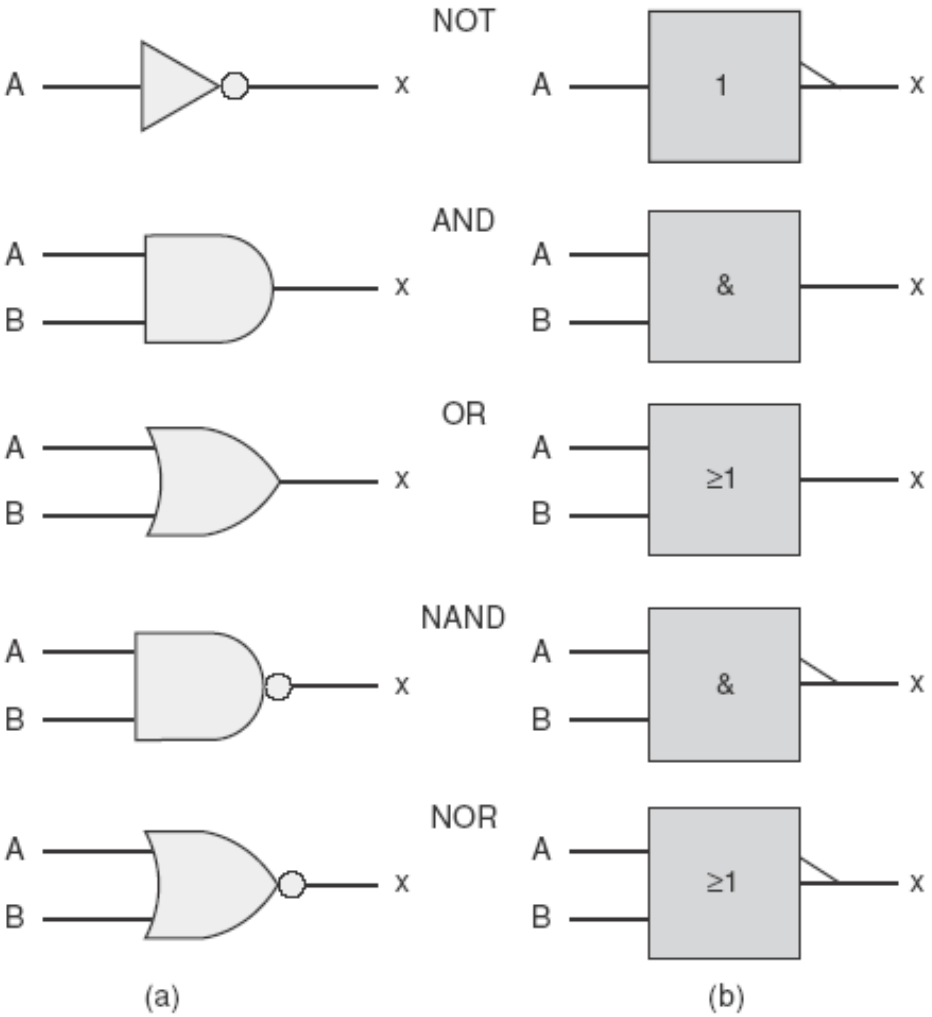
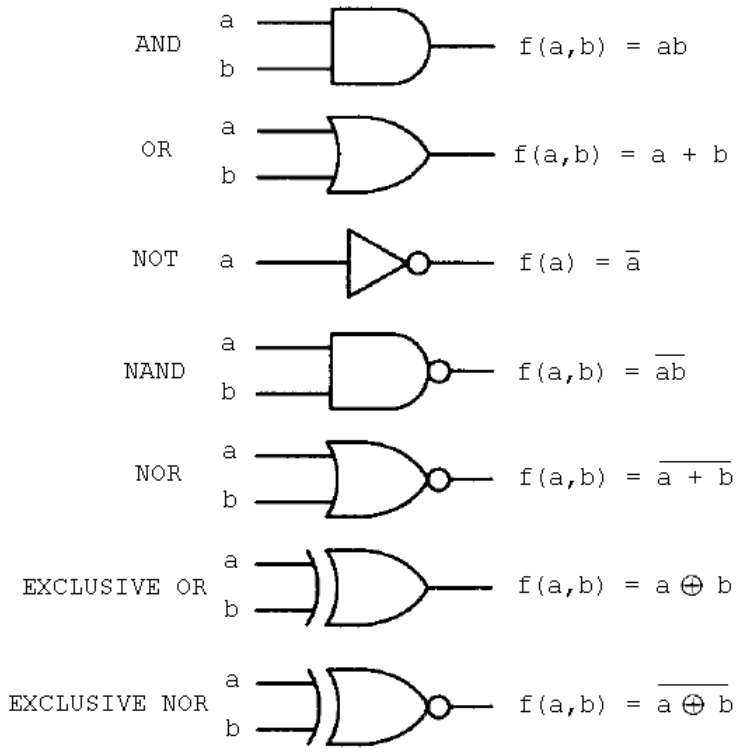
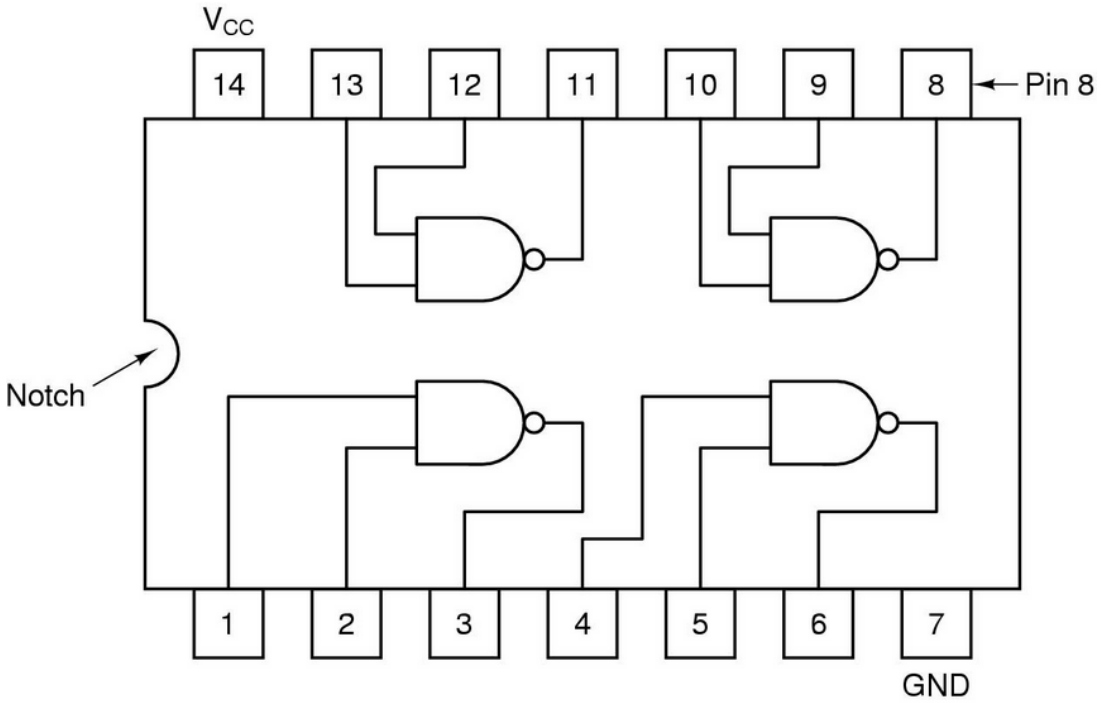


FIGURA 3.41
Símbolos lógicos-padrão:
(a) tradicional; (b) IEEE/ANSI.

7.7 – Portas Lógicas e Principais Circuitos Integrados



NAND – 7400 – Família TTL



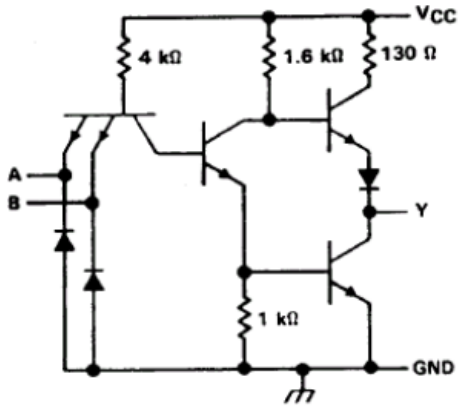
7.7 – Portas Lógicas e Principais Circuitos Integrados

SN5400, SN54LS00, SN54S00
SN7400, SN74LS00, SN74S00
QUADRUPLE 2-INPUT POSITIVE-NAND GATES

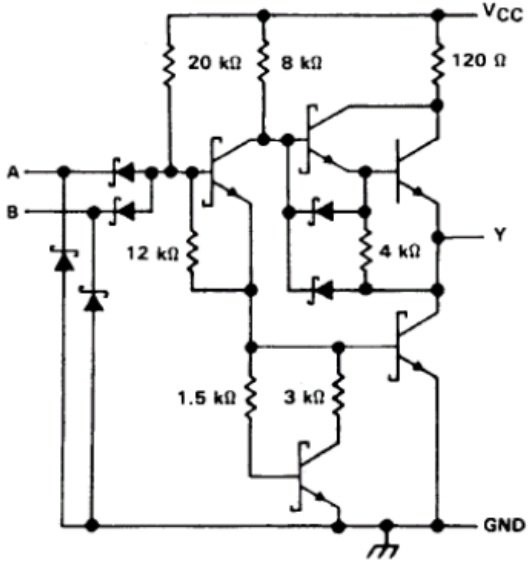
SDLS025 – DECEMBER 1983 – REVISED MARCH 1988

schematics (each gate)

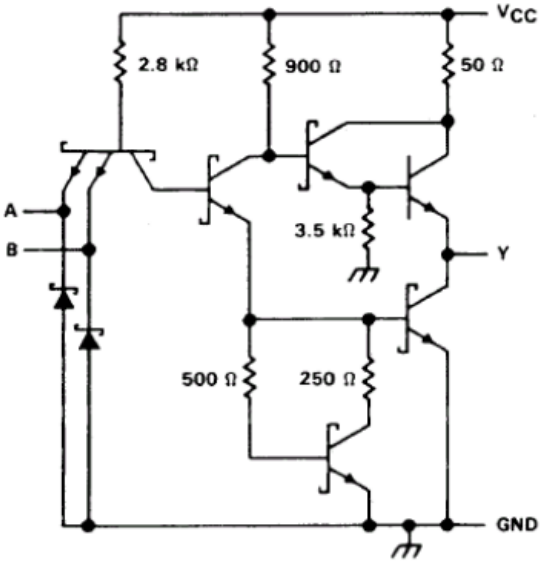
'00



'LS00

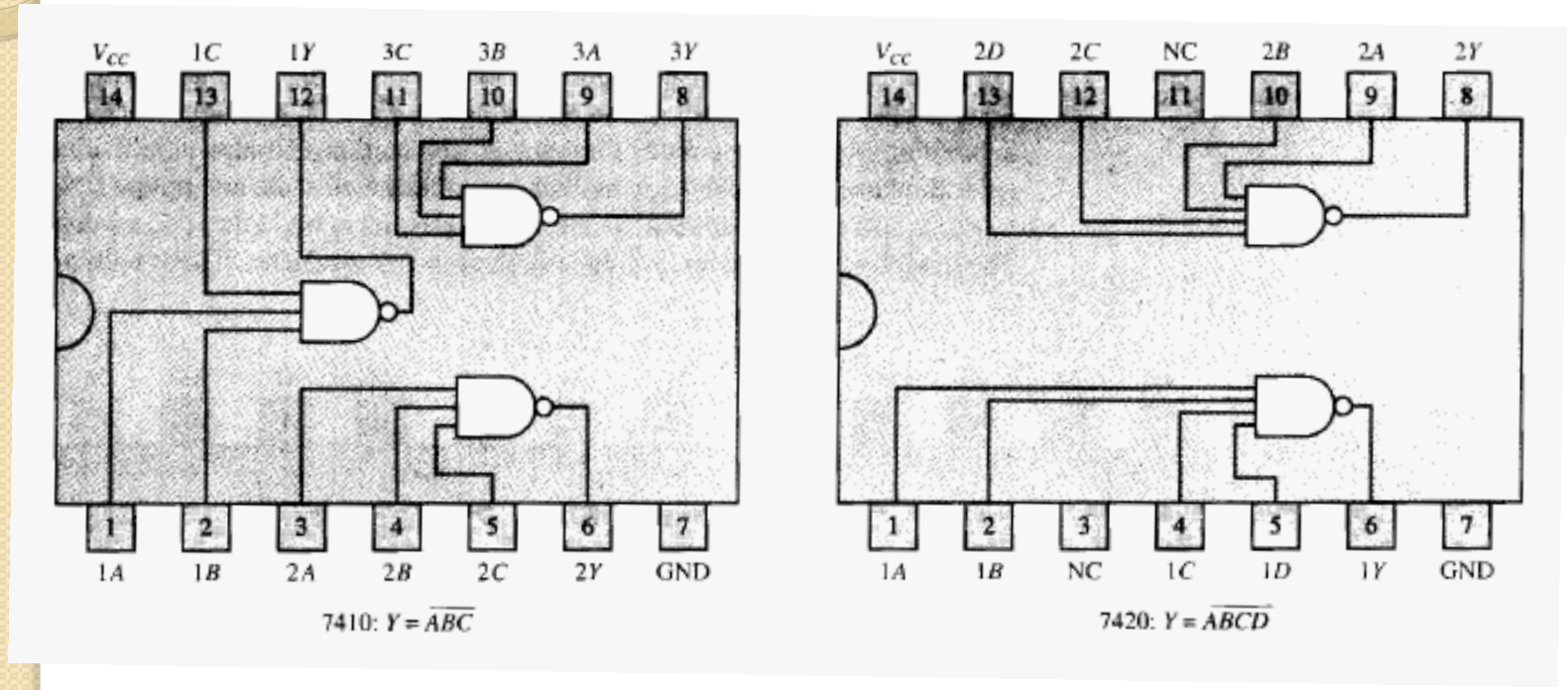


'S00

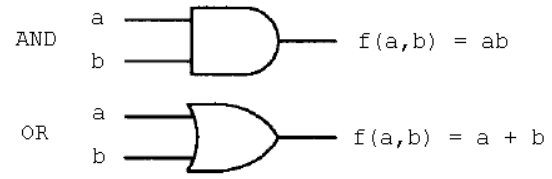


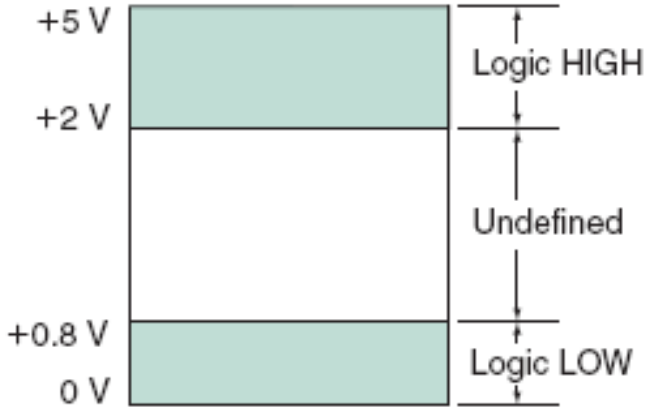
7.9 – Portas Lógicas e Principais Circuitos Integrados

NAND – 7410 e 7420 – Família TTL



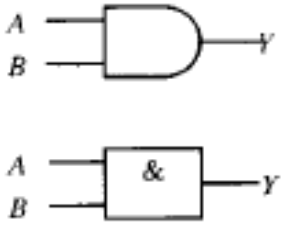
7.10 – Portas Lógicas e Níveis de Tensão





a	b	$f_{AND}(a, b) = ab$
0	0	0
0	1	0
1	0	0
1	1	1

A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

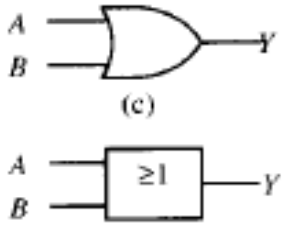


NOTE

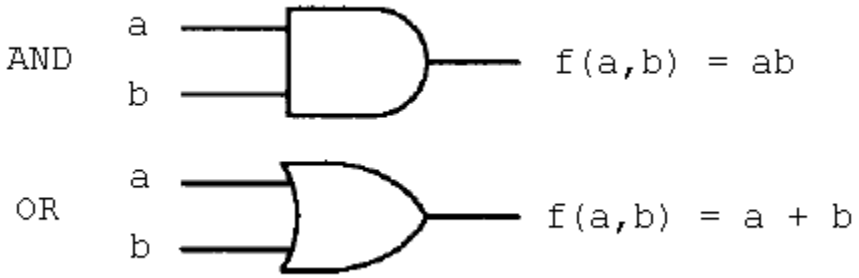
+5 V = Logic HIGH = 1
0 V = Logic LOW = 0

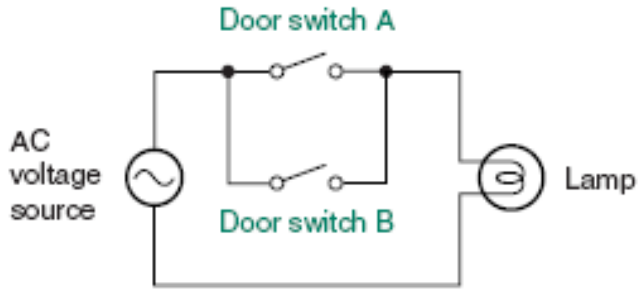
a	b	$f_{OR}(a, b) = a + b$
0	0	0
0	1	1
1	0	1
1	1	1

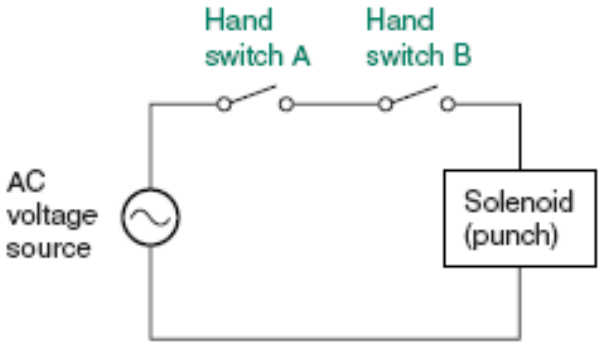
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	H



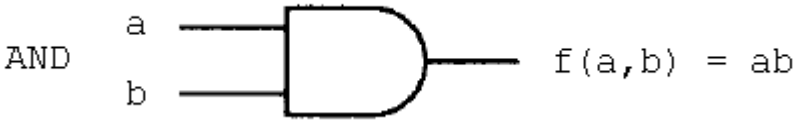
7.10 – Portas Lógicas e Níveis de Tensão

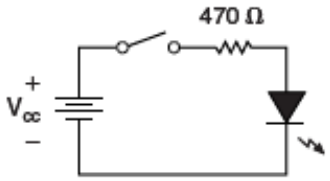


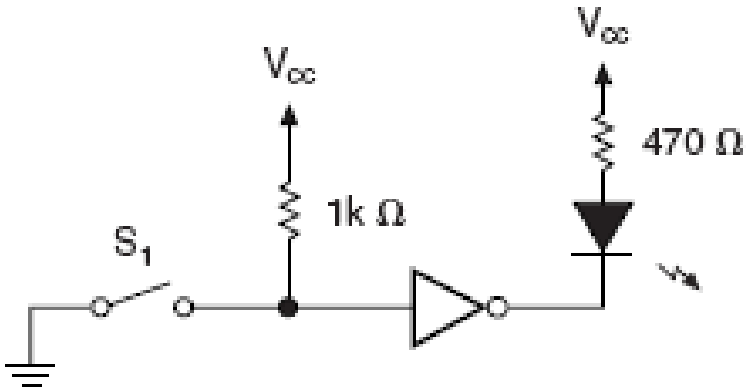


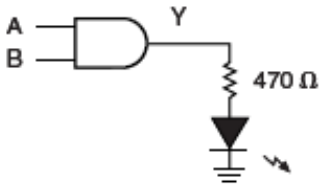


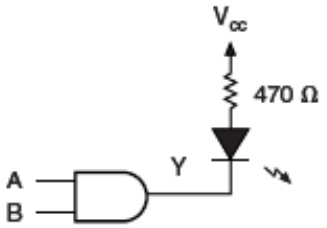
7.10 – Portas Lógicas e Níveis de Tensão











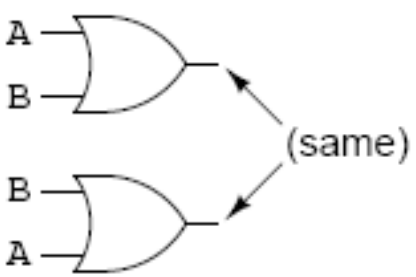
7.11 - Identidades Booleanas => Resumo

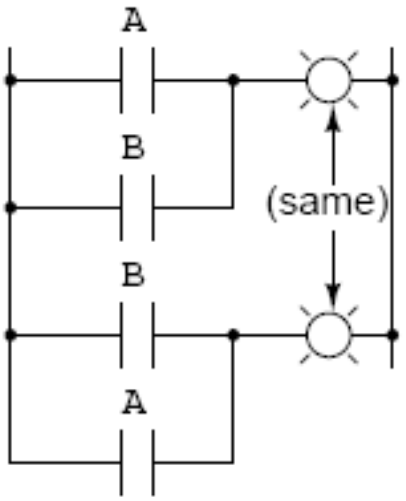
Additive	Multiplicative
$A + 0 = A$	$0A = 0$
$A + 1 = 1$	$1A = A$
$A + A = A$	$AA = A$
$A + \overline{A} = 1$	$A\overline{A} = 0$

7.12 – Propriedade Comutativa - OR

Commutative property of addition

A + B = B + A





Generic switch contact designation

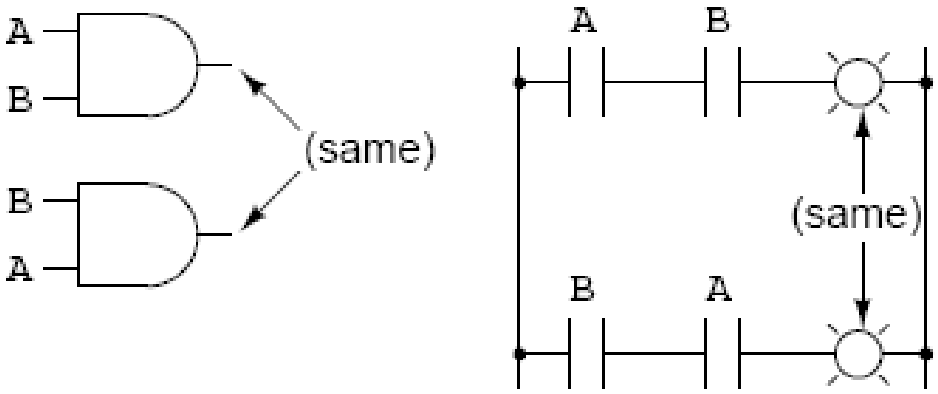
Normally-open

Normally-closed

7.12 – Propriedade Comutativa - AND

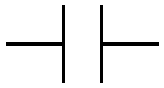
Commutative property of multiplication

AB = BA



Generic switch contact designation

Normally-open

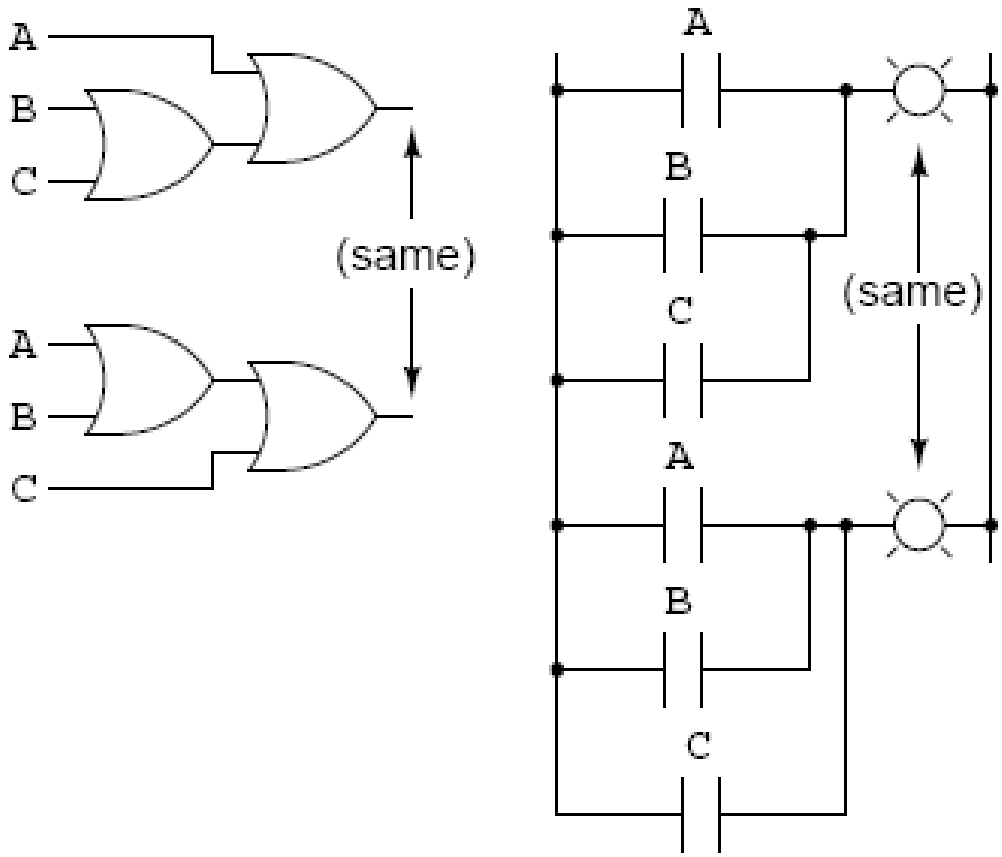


Normally-closed

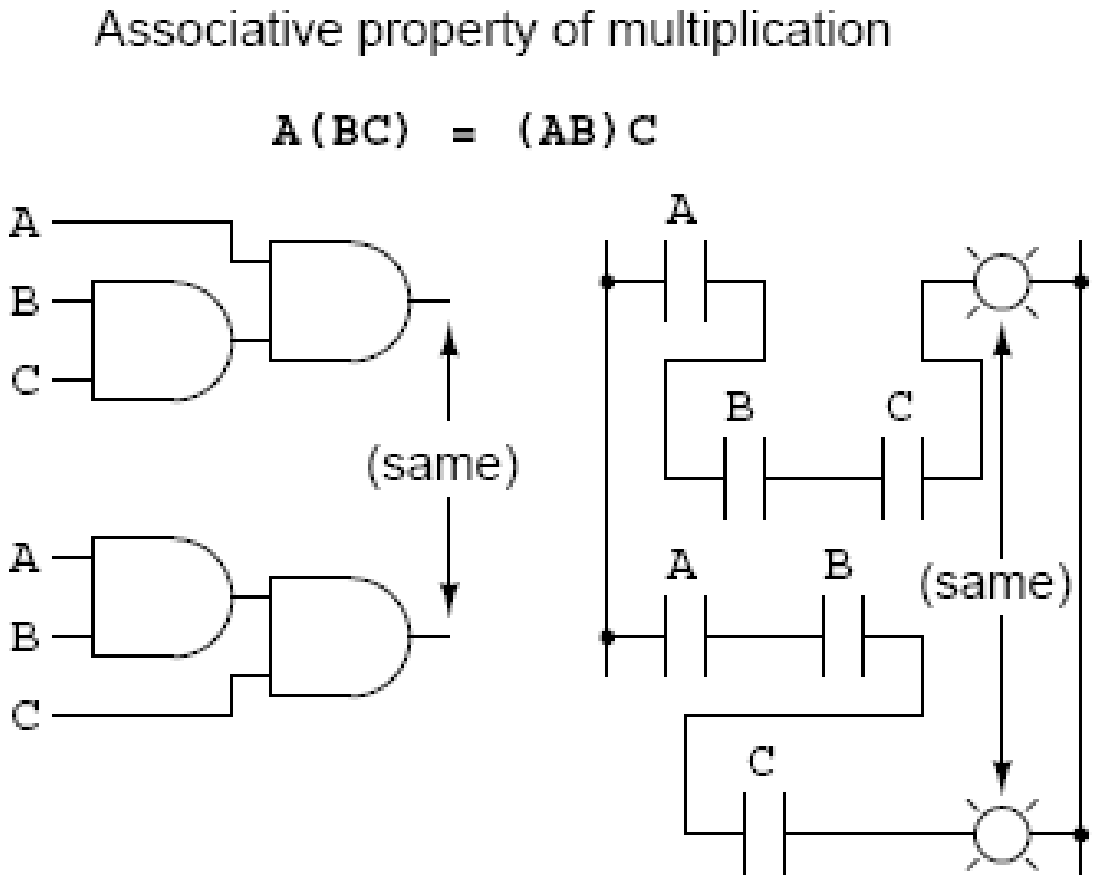
7.13 – Propriedade Associativa - OR

Associative property of addition

$$A + (B + C) = (A + B) + C$$



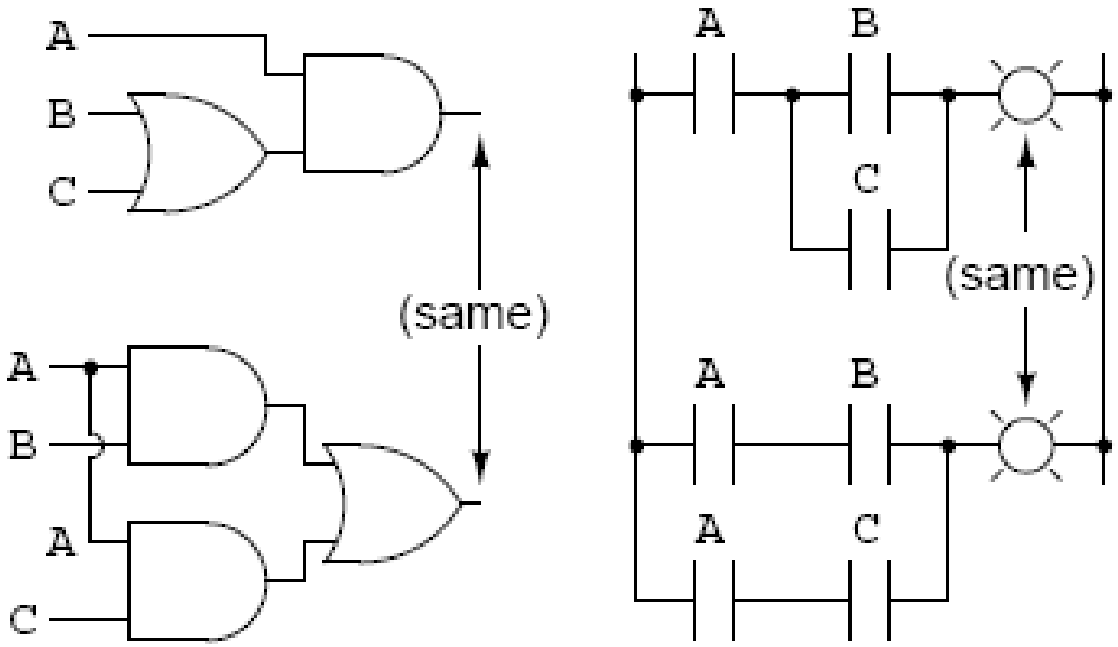
7.13 – Propriedade Associativa - AND



7.14 – Propriedade Distributiva

Distributive property

$$A(B + C) = AB + AC$$



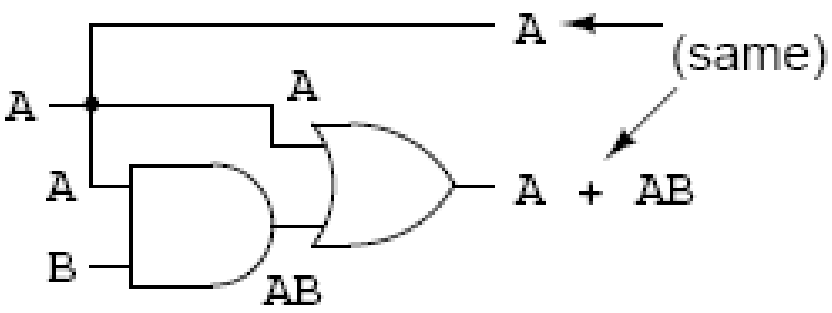
7.15 – Propriedades: Resumo

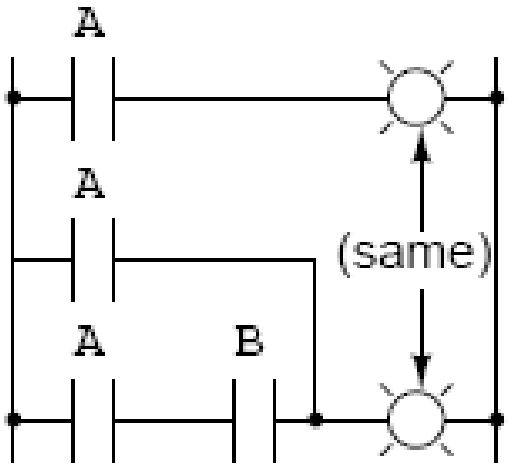
Basic Boolean algebraic properties

Additive	Multiplicative
$A + B = B + A$	$AB = BA$
$A + (B + C) = (A + B) + C$	$A(BC) = (AB)C$
$A(B + C) = AB + AC$	

7.16 – Propriedades para Simplificação

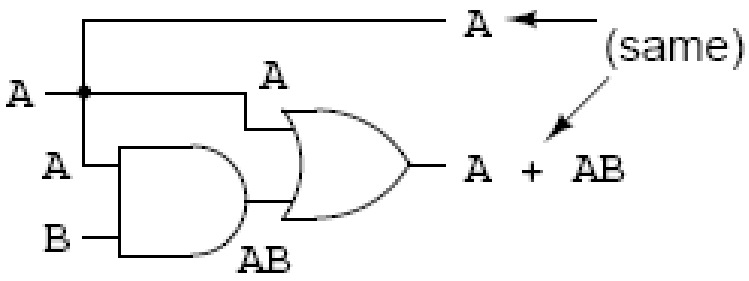
$$A + AB = A$$

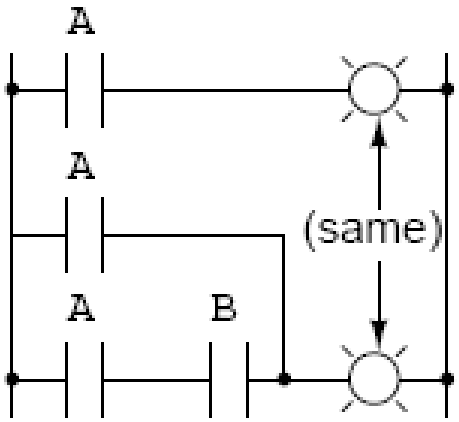




7.16 – Propriedades para Simplificação

A + AB = A





A + AB

↓ Factoring **A** out of both terms

A(1 + B)

↓ Applying identity **A + 1 = 1**

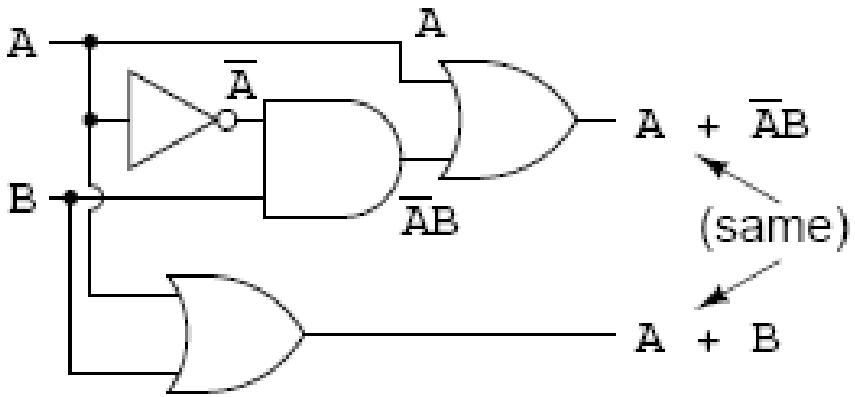
A(1)

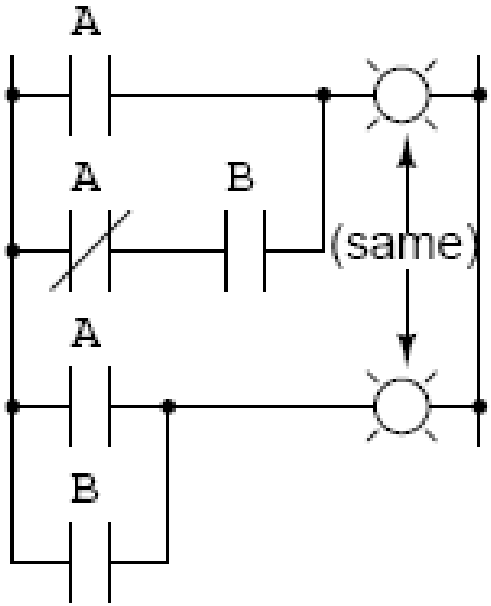
↓ Applying identity **1A = A**

A

7.16 – Propriedades para Simplificação

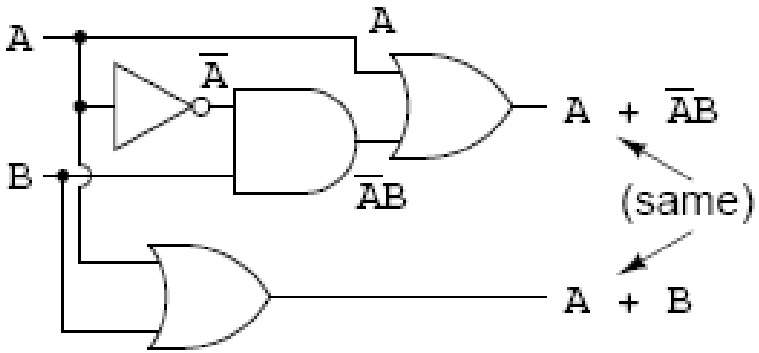
$$A + \overline{A}B = A + B$$

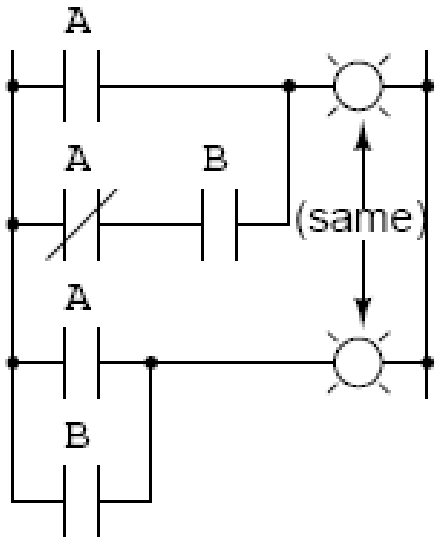




7.16 – Propriedades para Simplificação

$$A + \overline{A}B = A + B$$





$A + \overline{A}B$

↓ Applying the previous rule to expand **A** term

$A + AB = A$

$A + AB + \overline{A}B$

↓ Factoring **B** out of 2nd and 3rd terms

$A + B(A + \overline{A})$

↓ Applying identity $A + \overline{A} = 1$

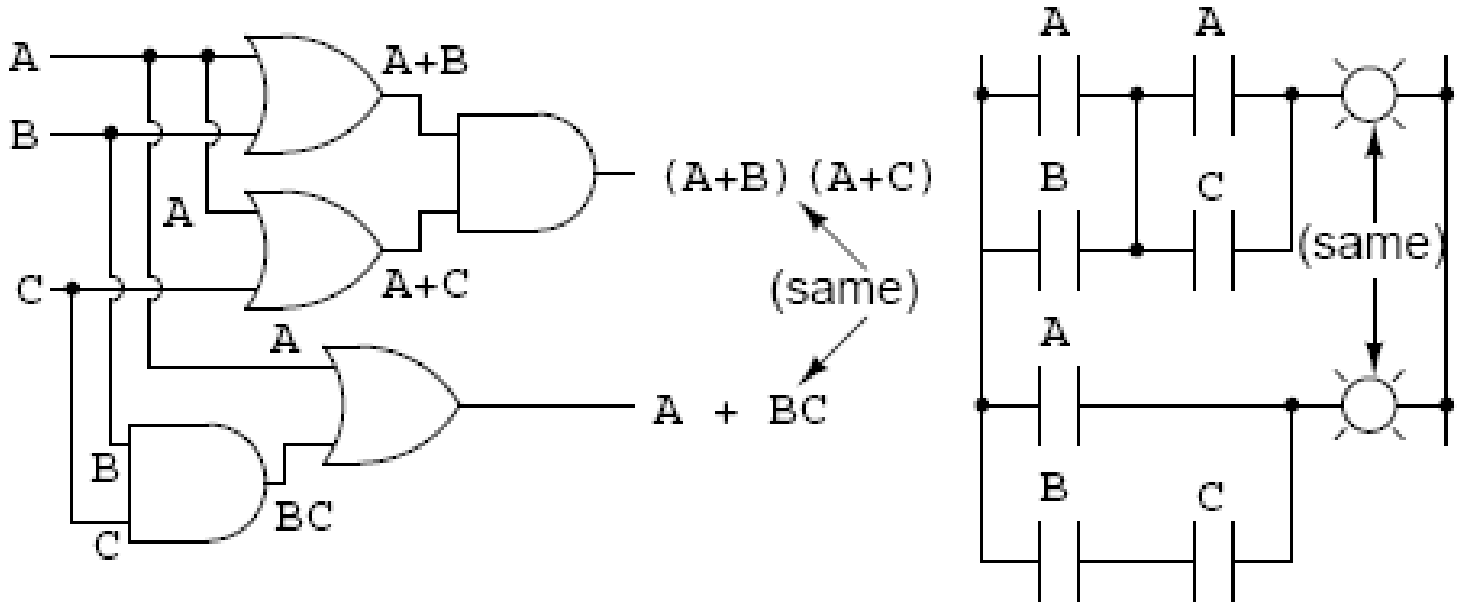
$A + B(1)$

↓ Applying identity $1A = A$

$A + B$

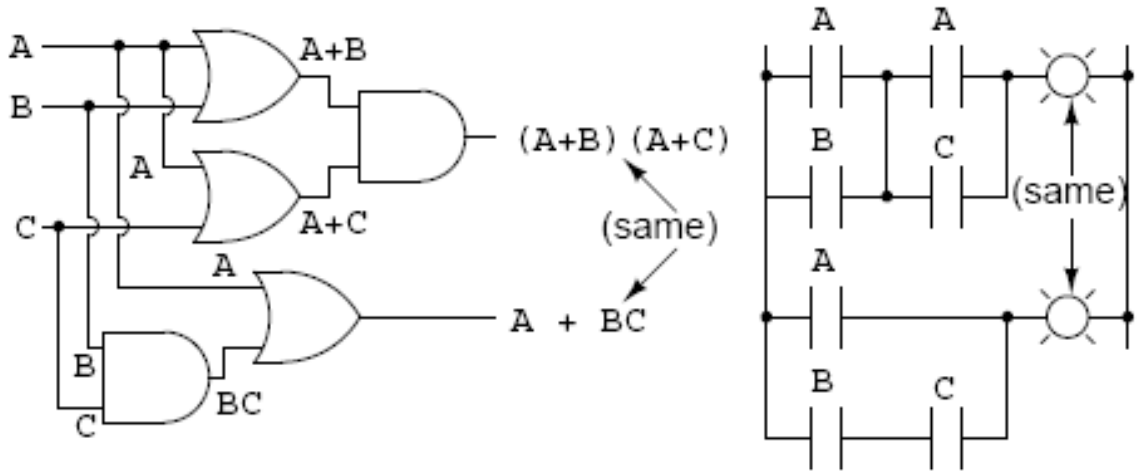
7.17 – Propriedades para Simplificação

$$(A + B)(A + C) = A + BC$$



7.17 – Propriedades para Simplificação

$$(A + B)(A + C) = A + BC$$



$(A + B)(A + C)$

↓ Distributing terms

$AA + AC + AB + BC$

↓ Applying identity $AA = A$

$A + AC + AB + BC$

↓ Applying rule $A + AB = A$ to the $A + AC$ term

$A + AB + BC$

↓ Applying rule $A + AB = A$ to the $A + AB$ term

$A + BC$

7.18 – Propriedades para Simplificação

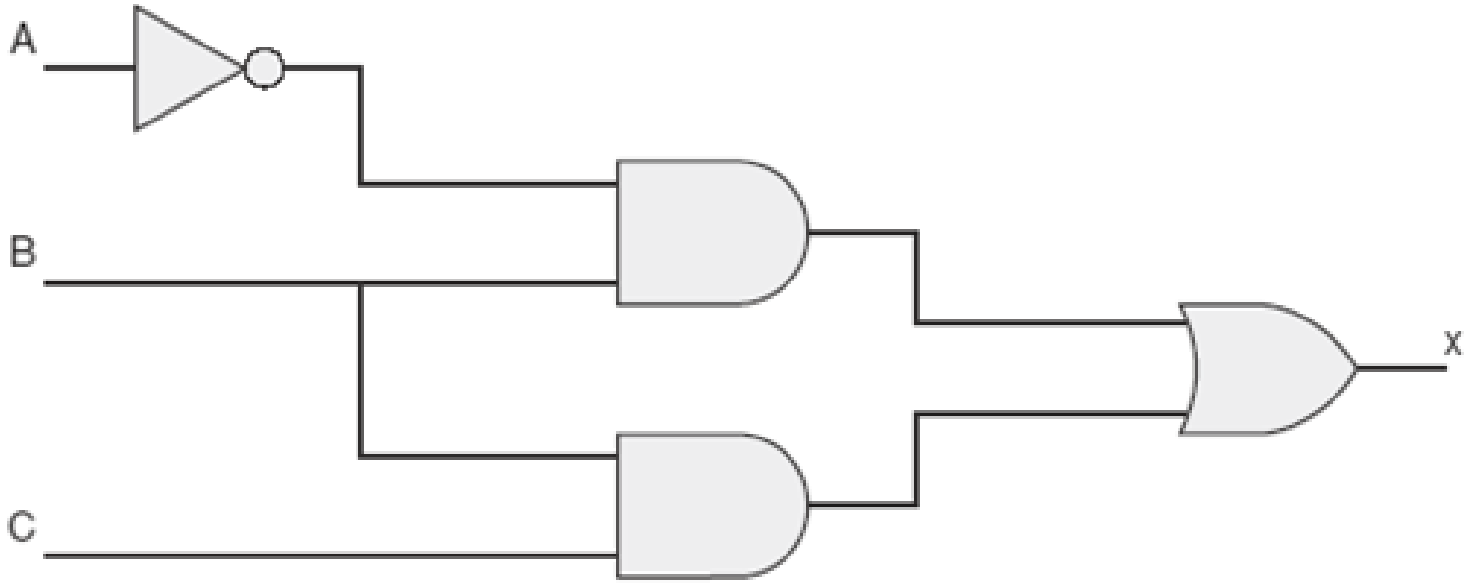
Useful Boolean rules for simplification

$$A + AB = A$$

$$A + \overline{A}B = A + B$$

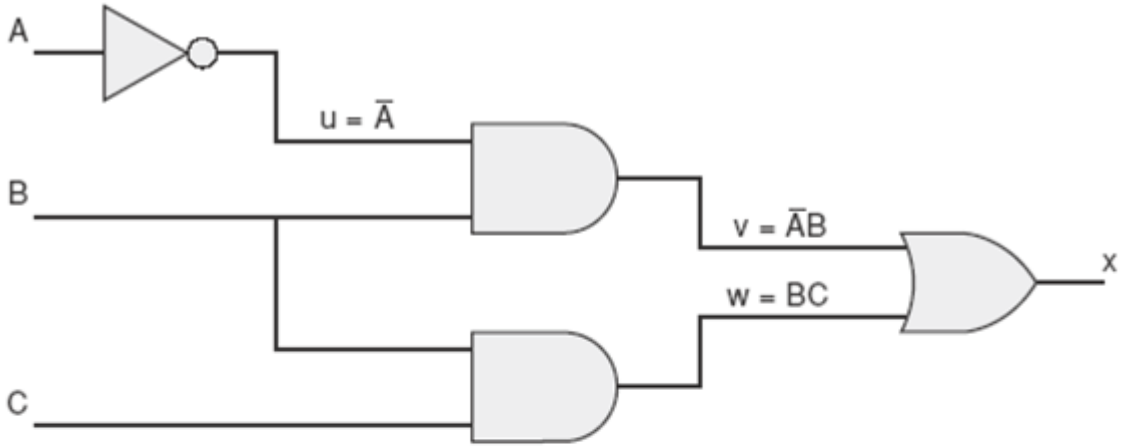
$$(A + B)(A + C) = A + BC$$

Exercicio: Obter a tabela-verdade do circuito



(a)

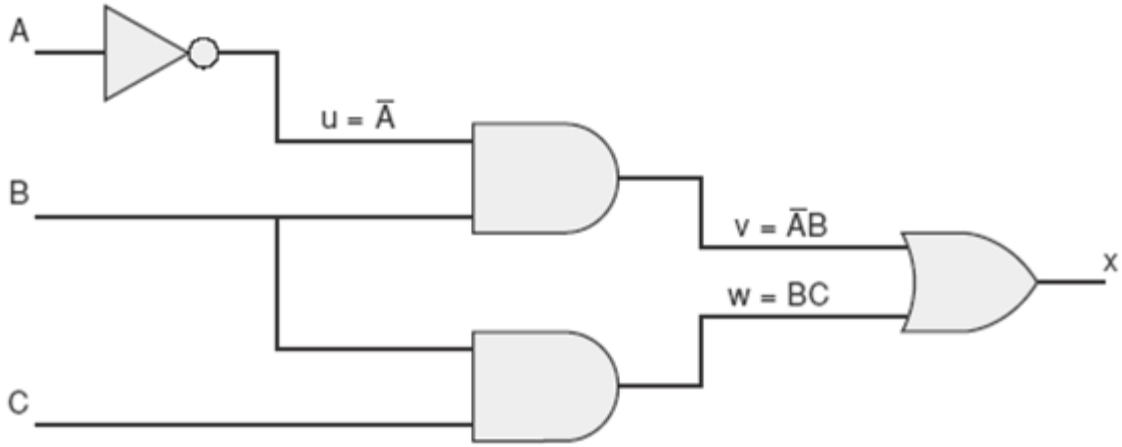
Exercício: Obter a tabela-verdade do circuito



(a)

A	B	C	u= Ā	v= ĀB	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			

(b)



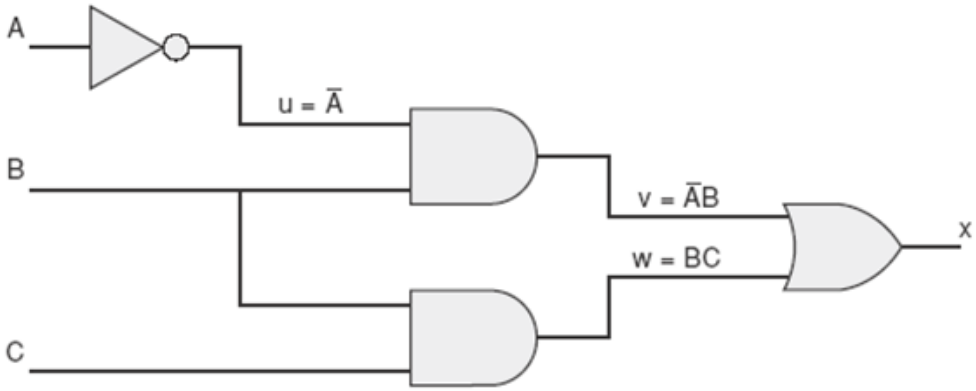
(a)

A	B	C	u= $\bar{A}$	v= $\bar{A}$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			

(b)

A	B	C	u= $\bar{A}$	v= $\bar{A}$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**		

(c)



(a)

A	B	C	u= Ā	v= ĀB	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			

(b)

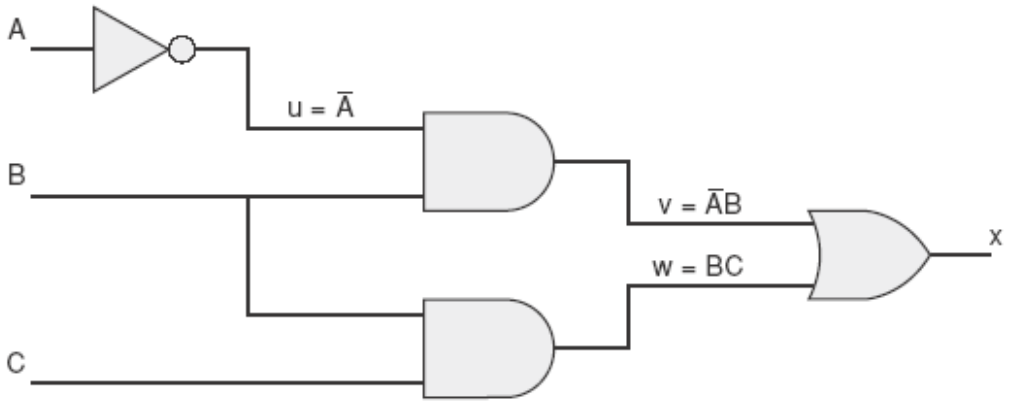
A	B	C	u= Ā	v= ĀB	w= BC	x= v+w
0	**0**	0	1	**0**		
0	**0**	1	1	**0**		
0	1	0	1	1		
0	1	1	1	1		
1	0	0	0	**0**		
1	0	1	0	**0**		
1	1	0	**0**	**0**		
1	1	1	**0**	**0**		

(c)

A	B	C	u= Ā	v= ĀB	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	

(d)

FIGURA 3.16
Análise de um circuito lógico usando tabelas-verdade.



(a)

A	B	C	u= Ā	v= ĀB	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			

(b)

A	B	C	u= Ā	v= ĀB	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**		

(c)

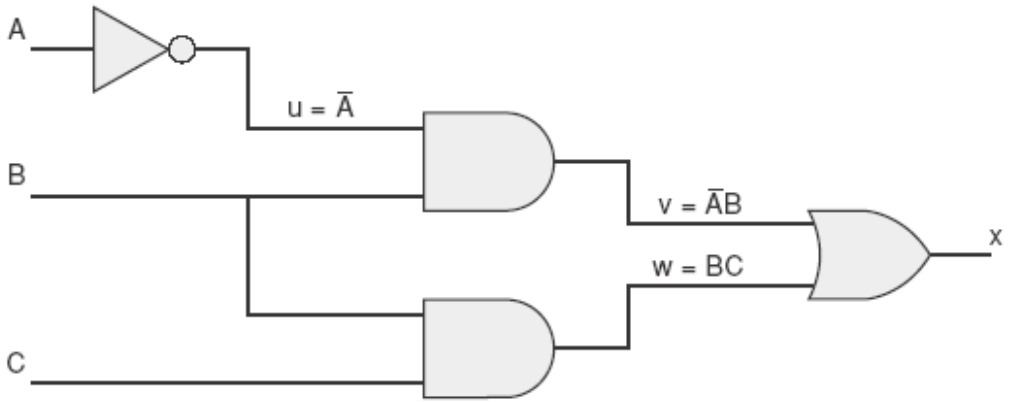
A	B	C	u= Ā	v= ĀB	w= BC	x= v+w
0	**0**	**0**	1	0	**0**	
0	**0**	1	1	0	**0**	
0	1	**0**	1	1	**0**	
0	1	1	1	1	1	
1	**0**	**0**	0	0	**0**	
1	**0**	1	0	0	**0**	
1	1	**0**	0	0	**0**	
1	1	1	0	0	1	

(d)

A	B	C	u= Ā	v= ĀB	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

(e)

FIGURA 3.16
Análise de um circuito lógico usando tabelas-verdade.



(a)

A	B	C	u= Ā	v= ĀB	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			

(b)

A	B	C	u= Ā	v= ĀB	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**		

(c)

A	B	C	u= Ā	v= ĀB	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	

(d)

A	B	C	u= Ā	v= ĀB	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

(e)

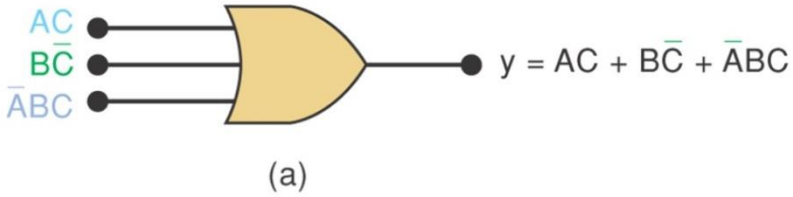
FIGURA 3.16
Análise de um circuito lógico usando tabelas-verdade.

Exemplo 1: Construindo um Circuito Lógico a partir da expressão

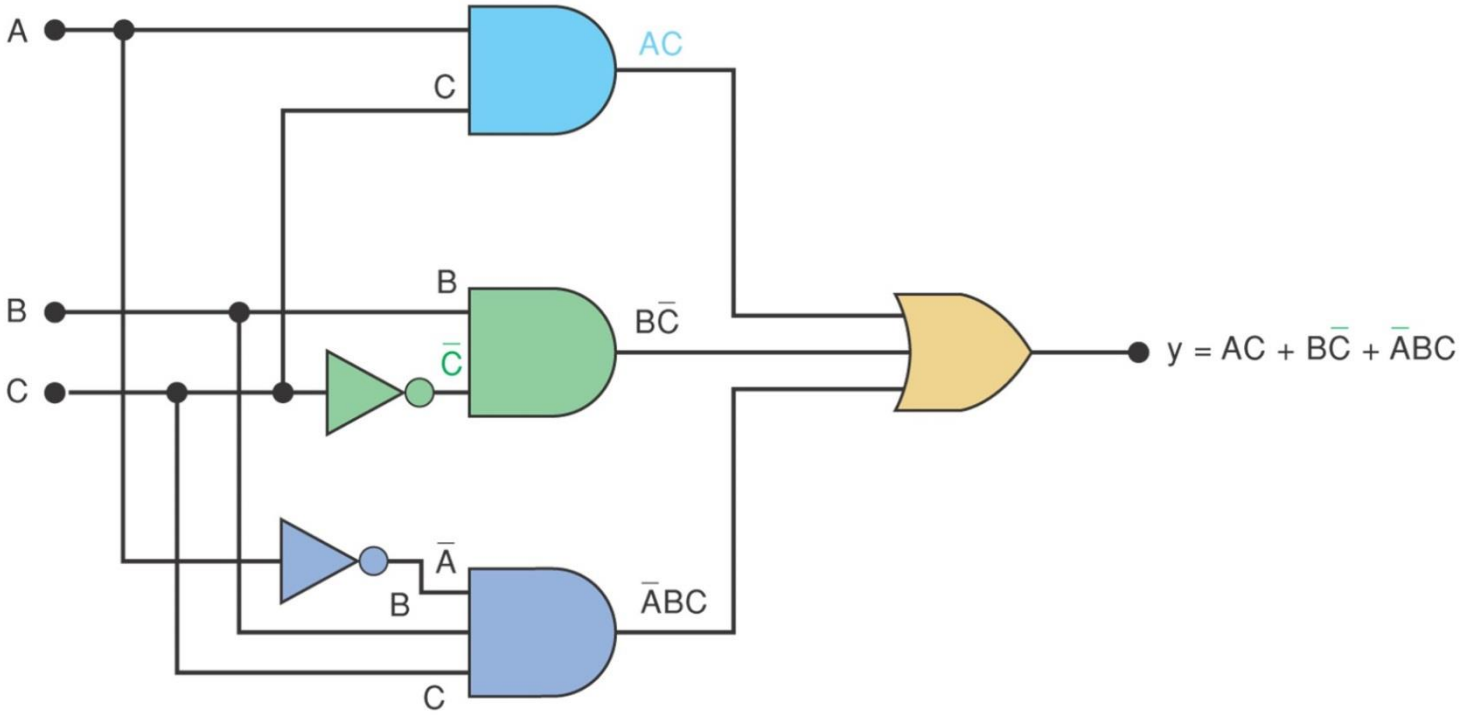
$$y = AC + B\overline{C} + \overline{A}BC$$

Exemplo 1: Construindo um Circuito Lógico a partir da expressão

$$y = AC + B\overline{C} + \overline{A}BC$$



(a)



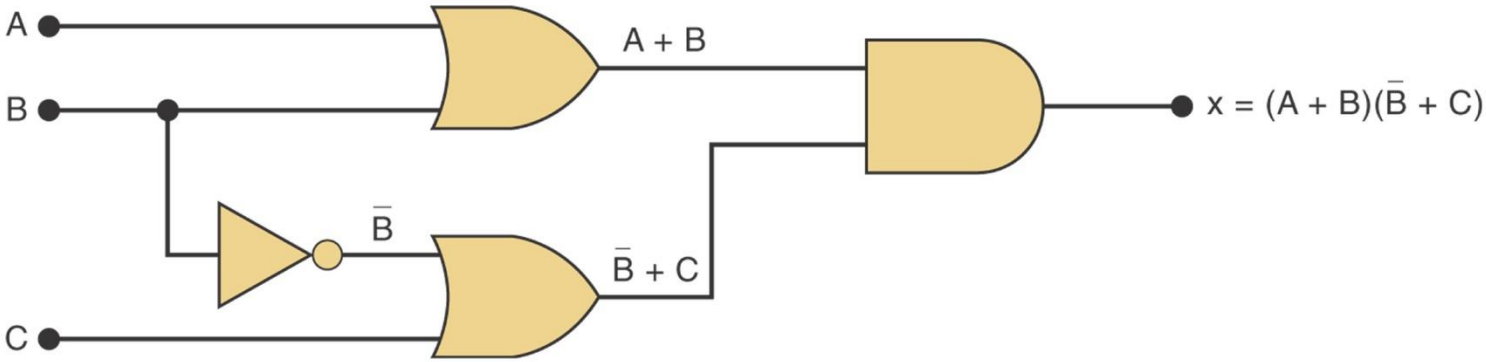
(b)

Exemplo 2: Construindo um Circuito Lógico a partir da expressão

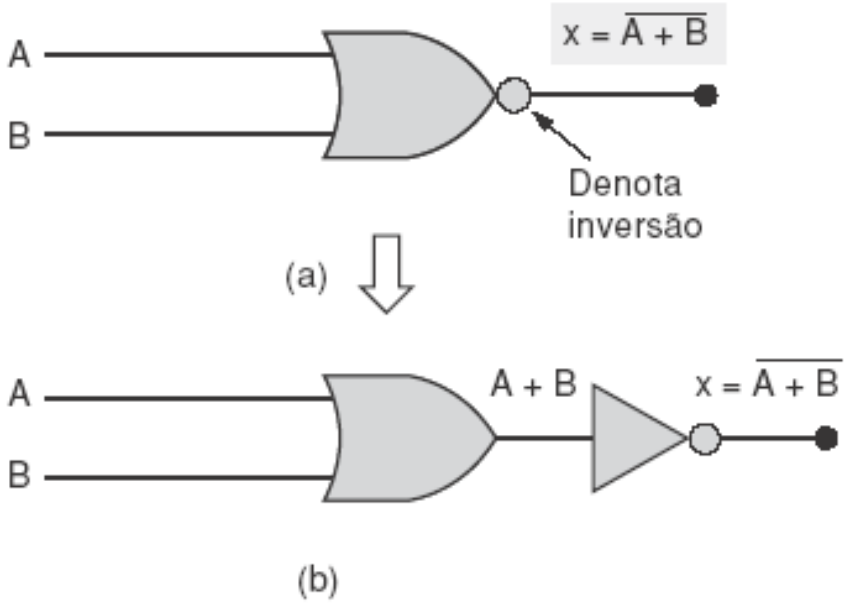
$$X = (A + B)(\overline{B} + C)$$

Exemplo 2: Construindo um Circuito Lógico a partir da expressão

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



Porta NOR: Not OR



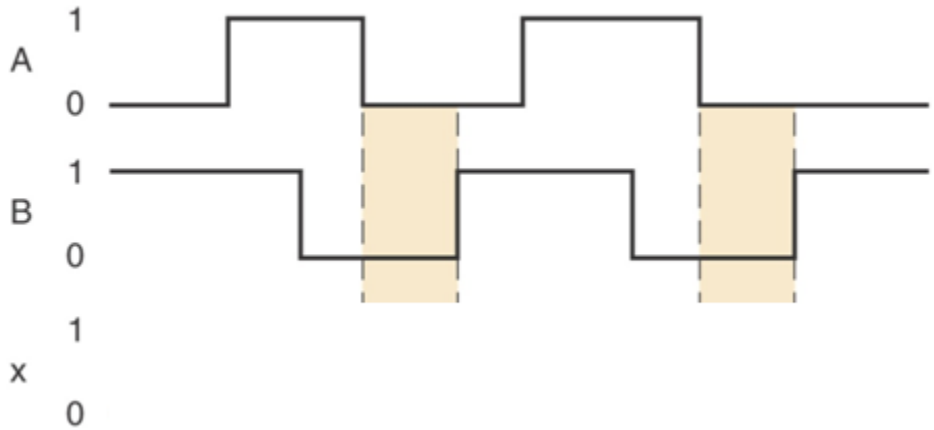
		OR	NOR
A	B	A + B	$\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

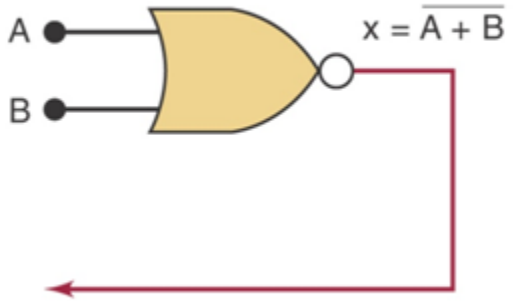
(c)

FIGURA 3.19
(a) Símbolo da porta NOR;
(b) Circuito equivalente;
(c) Tabela-verdade.

Porta NOR: Not OR
Diagrama de Temporização

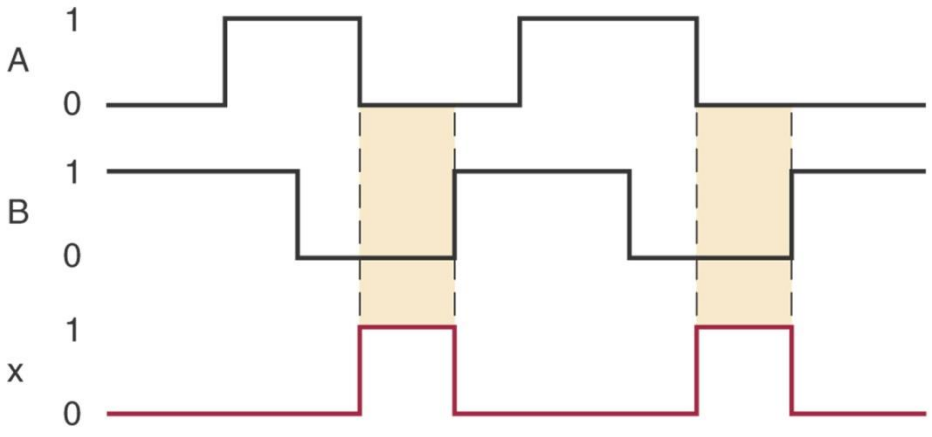
FIGURA 3.20
Exemplo 3.8.

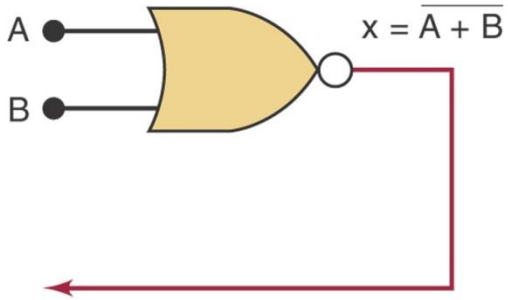




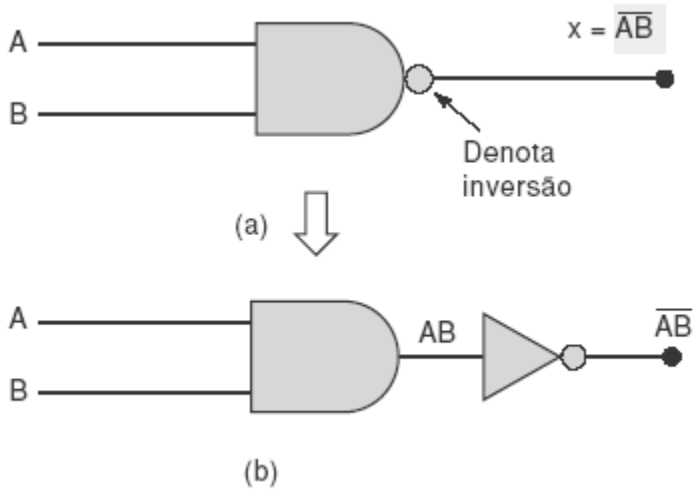
Porta NOR: Not OR
Diagrama de Temporização

FIGURA 3.20
Exemplo 3.8.





Porta NAND: Not AND



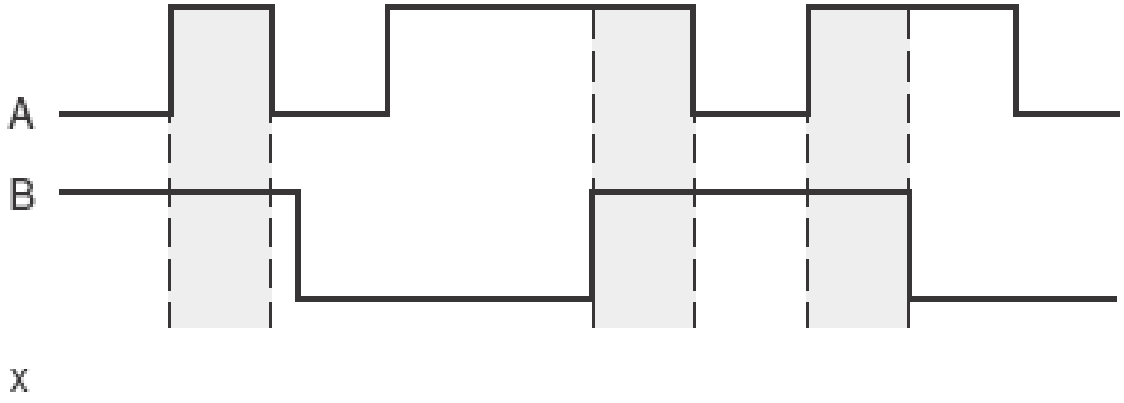
		AND	NAND
A	B	AB	$\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

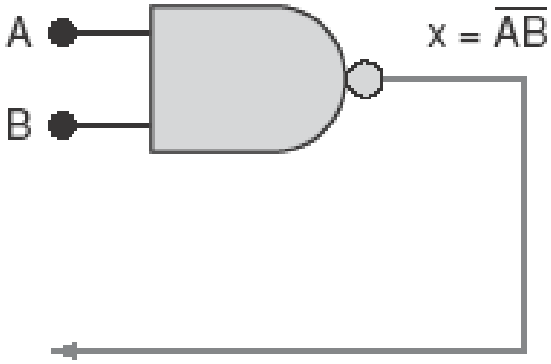
(c)

FIGURA 3.22
(a) Símbolo da porta NAND;
(b) Circuito equivalente;
(c) Tabela-verdade.

Porta NAND: Not AND

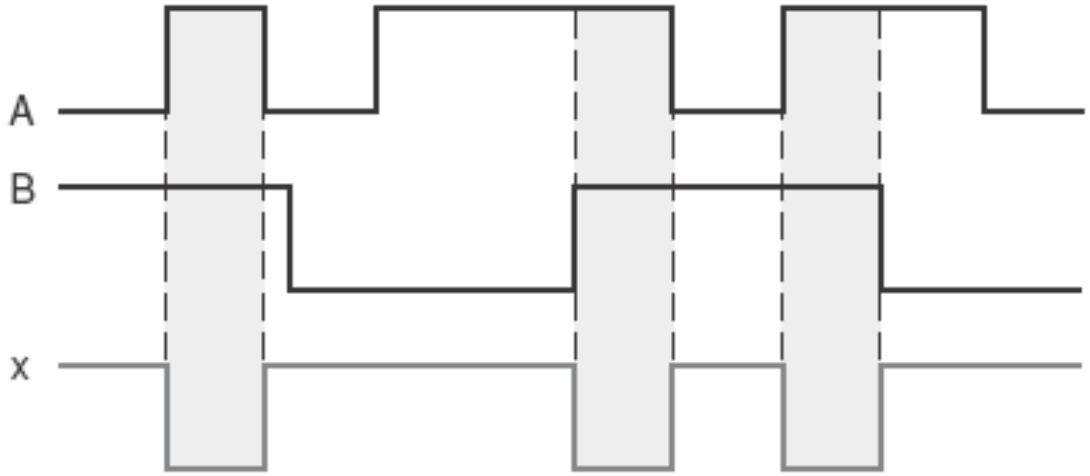
Diagrama de Temporização: Obtenha x.

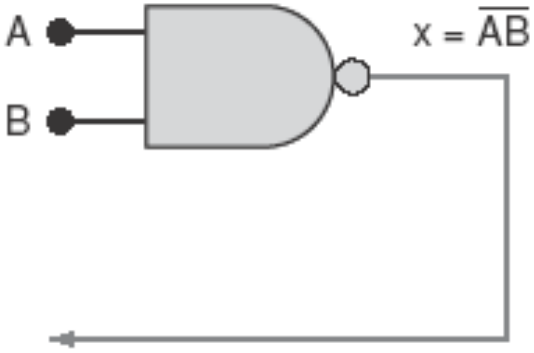




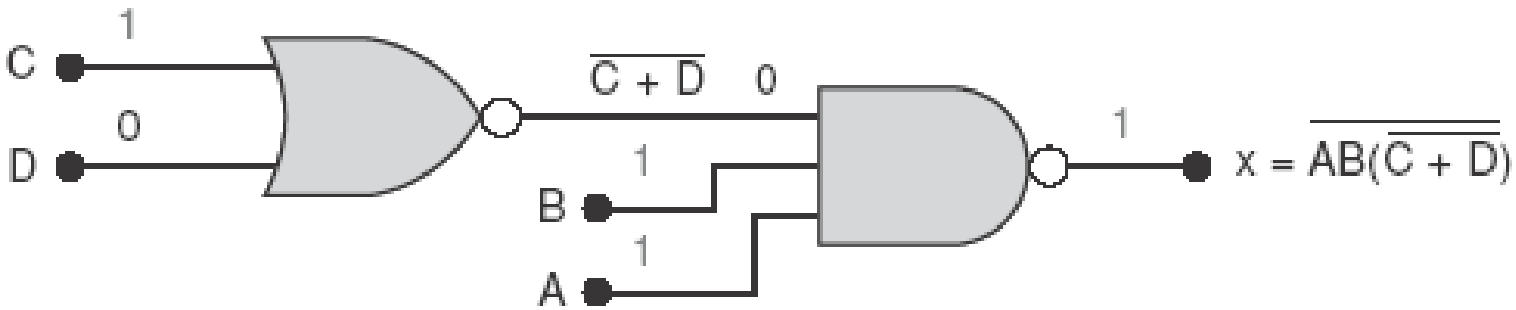
Exemplo 3.10.

Porta NAND: Not AND
Diagrama de Temporização





Exemplo 3.10.



Obtenha a Tabela-Verdade.

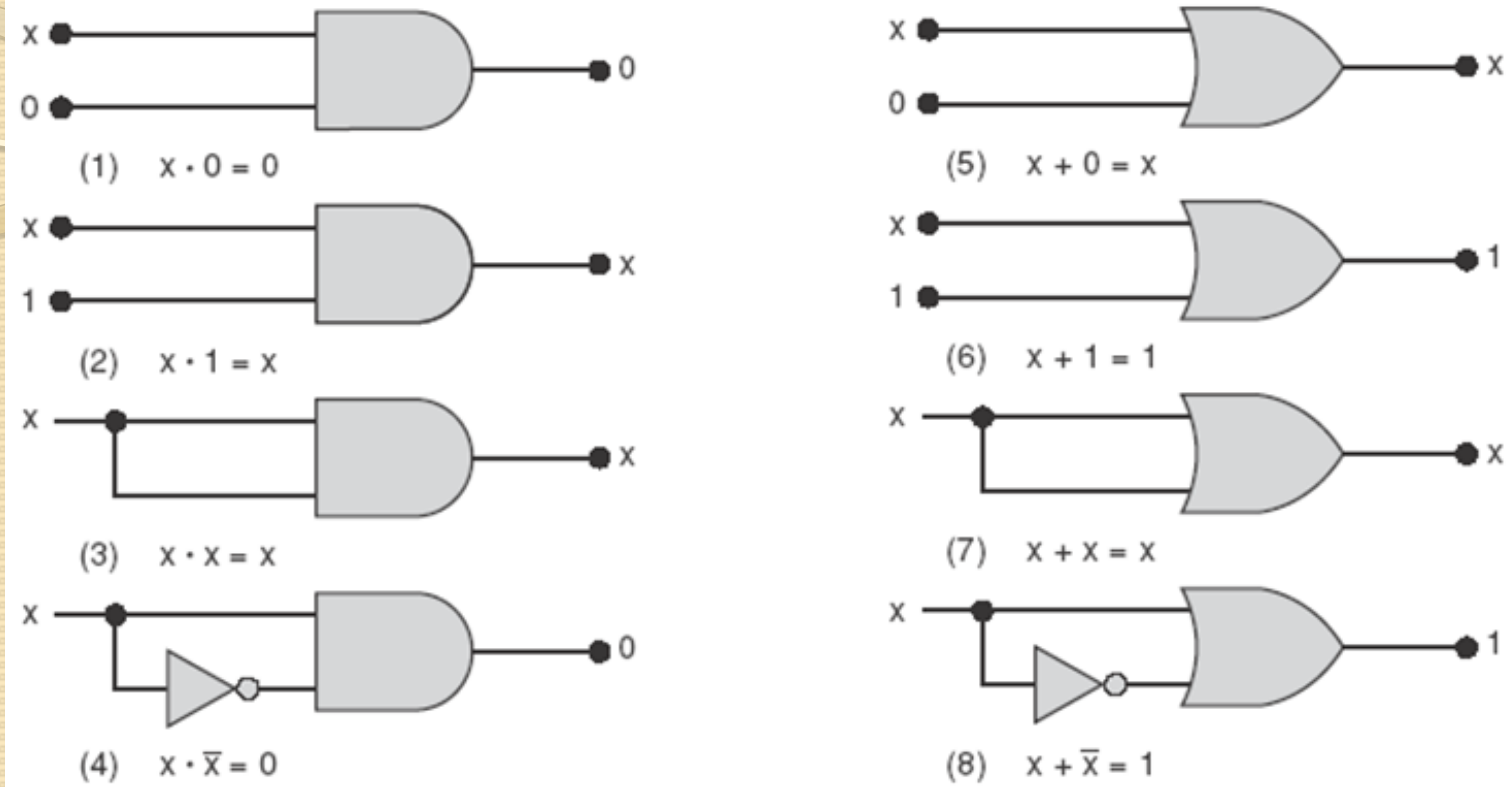
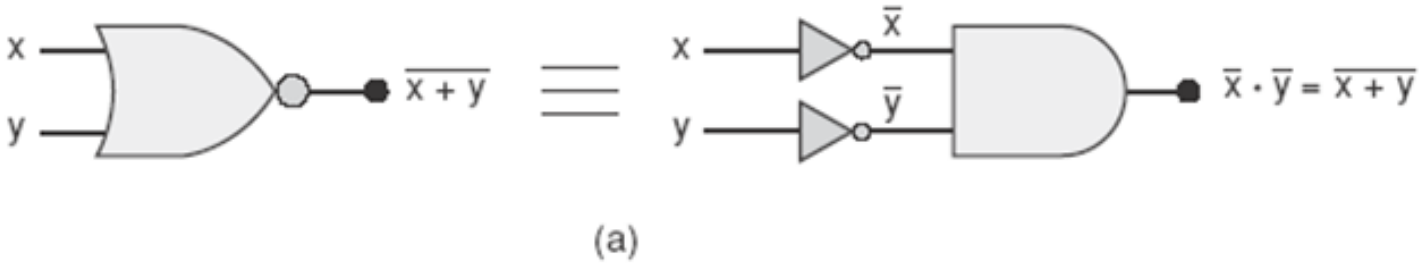
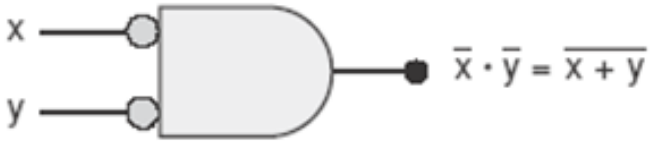


FIGURA 3.25
Teoremas para uma única variável.



(a)



(b)

FIGURA 3.26
(a) Circuitos equivalentes relativos ao teorema (16); (b) Símbolo alternativo para a função NOR.

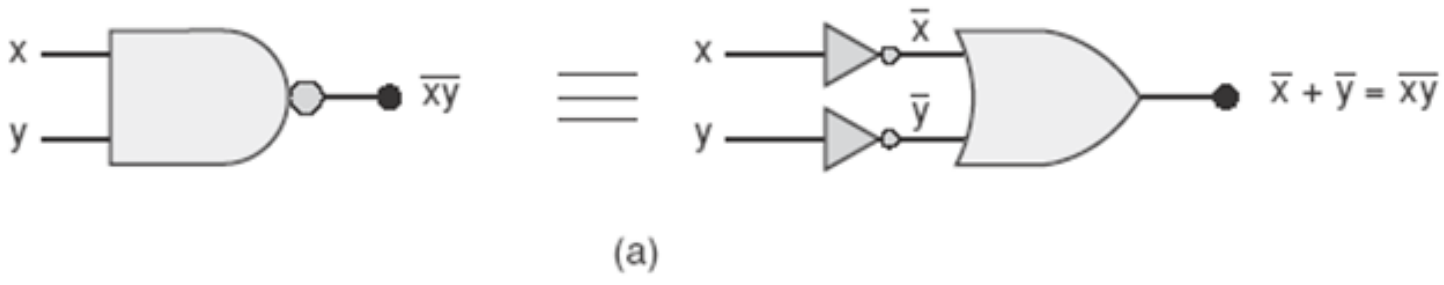
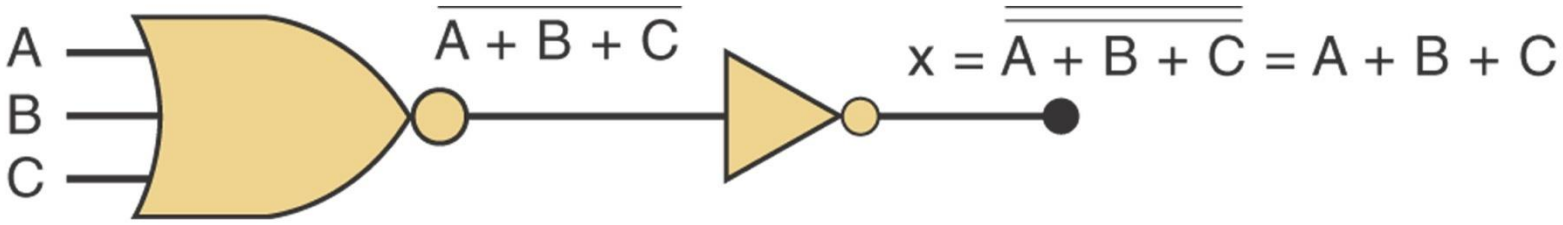


FIGURA 3.27
(a) Circuitos equivalentes relativos ao teorema (17); (b) Símbolo alternativo para a função NAND.

FIGURA 3.21
Exemplo 3.9.



Obtenha a expressão do circuito abaixo e simplifique

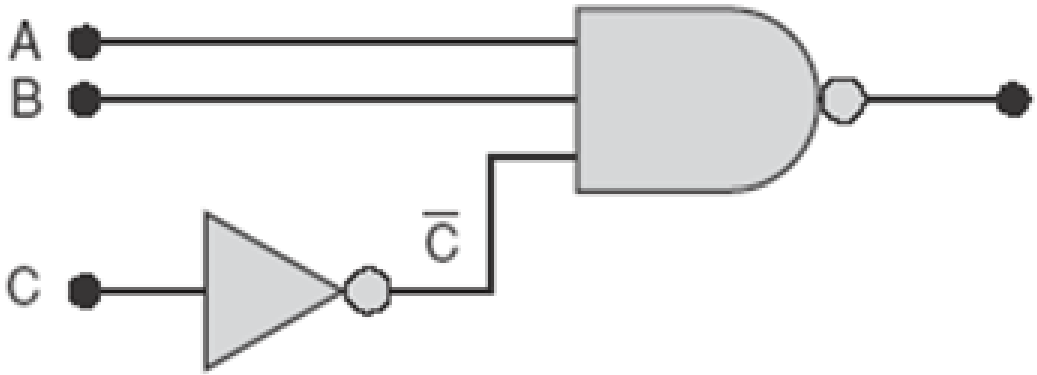


FIGURA 3.28
Exemplo 3.17.

Obtenha a expressão do circuito abaixo e simplifique

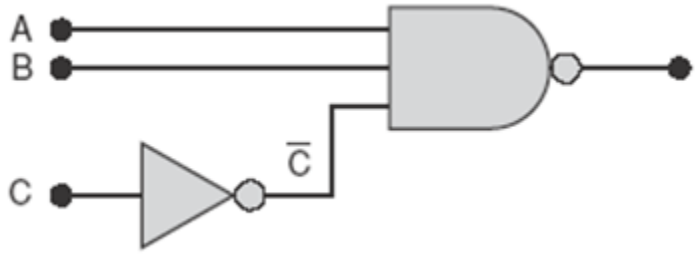
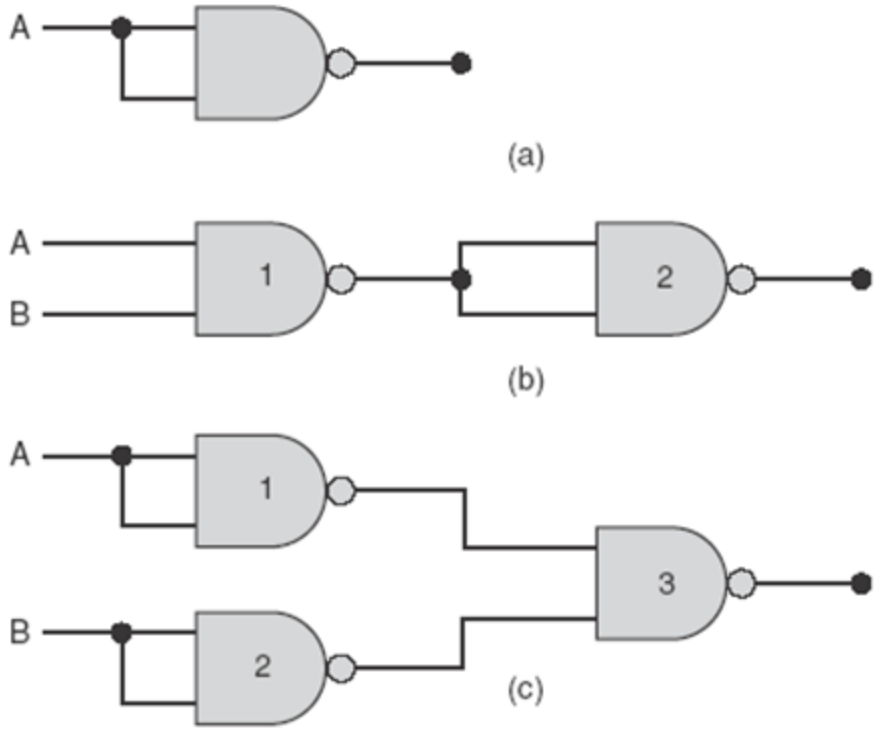


FIGURA 3.28
Exemplo 3.17.

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

Porta NAND

Obtenha as expressões de saída.

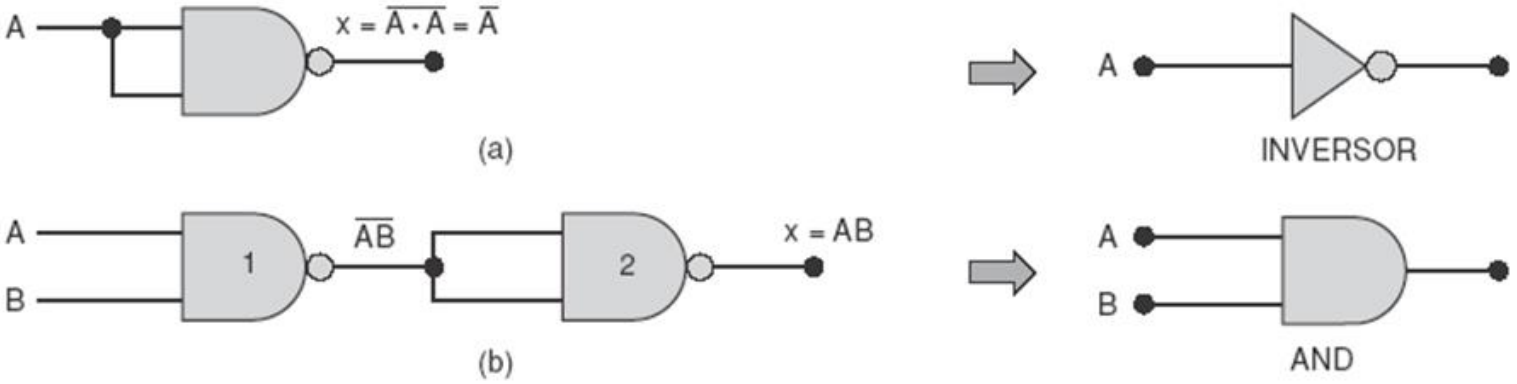


Porta NAND

Obtenha as expressões de saída.

Porta NAND

Obtenha as expressões de saída.



Porta NAND

Obtenha as expressões de saída.

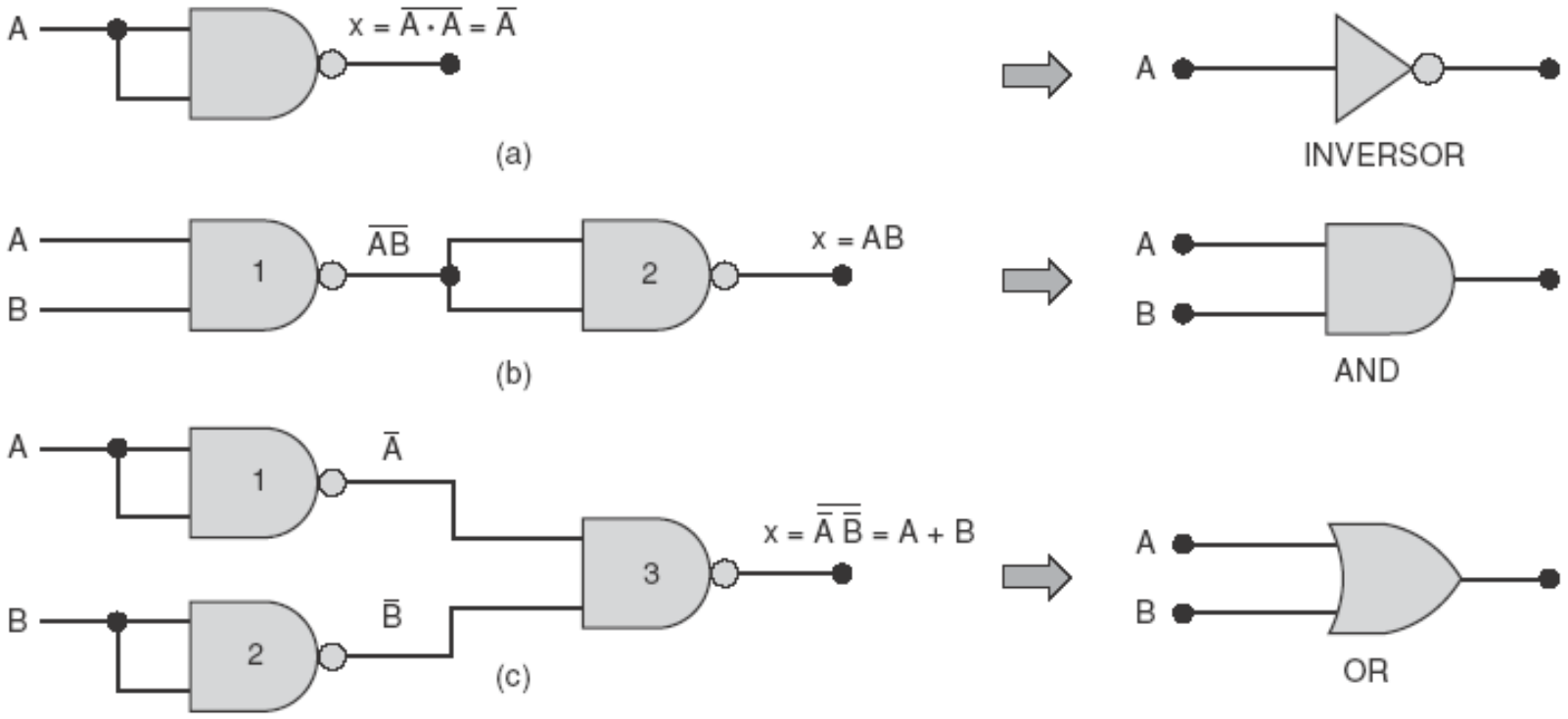
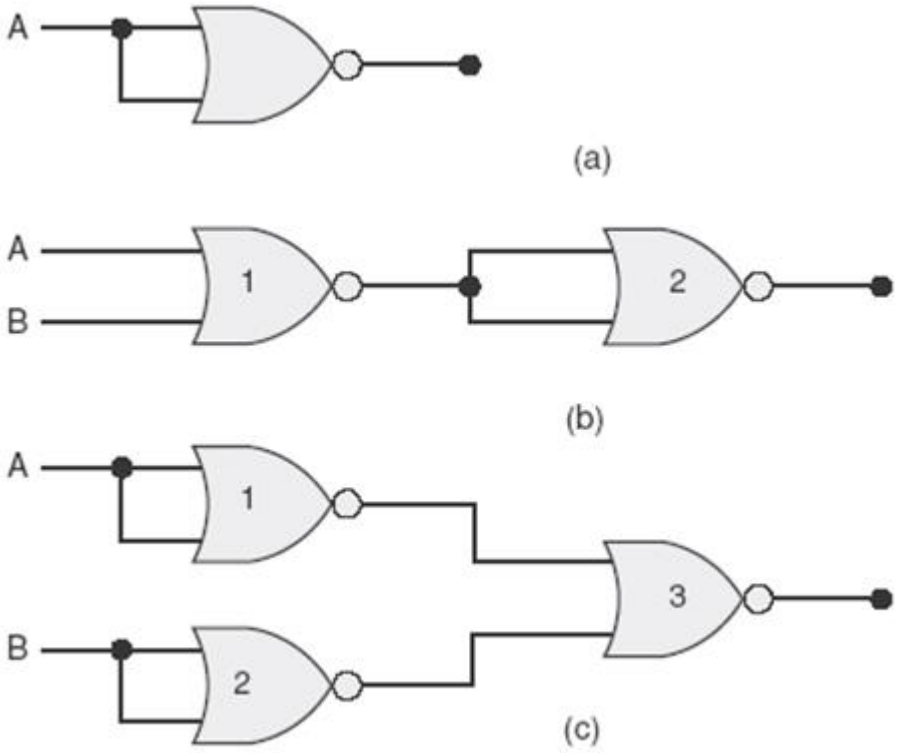


FIGURA 3.29
As portas NAND podem ser usadas para implementar qualquer função booleana.

Porta NOR

Obtenha as expressões de saída.



Porta NOR

Obtenha as expressões de saída.

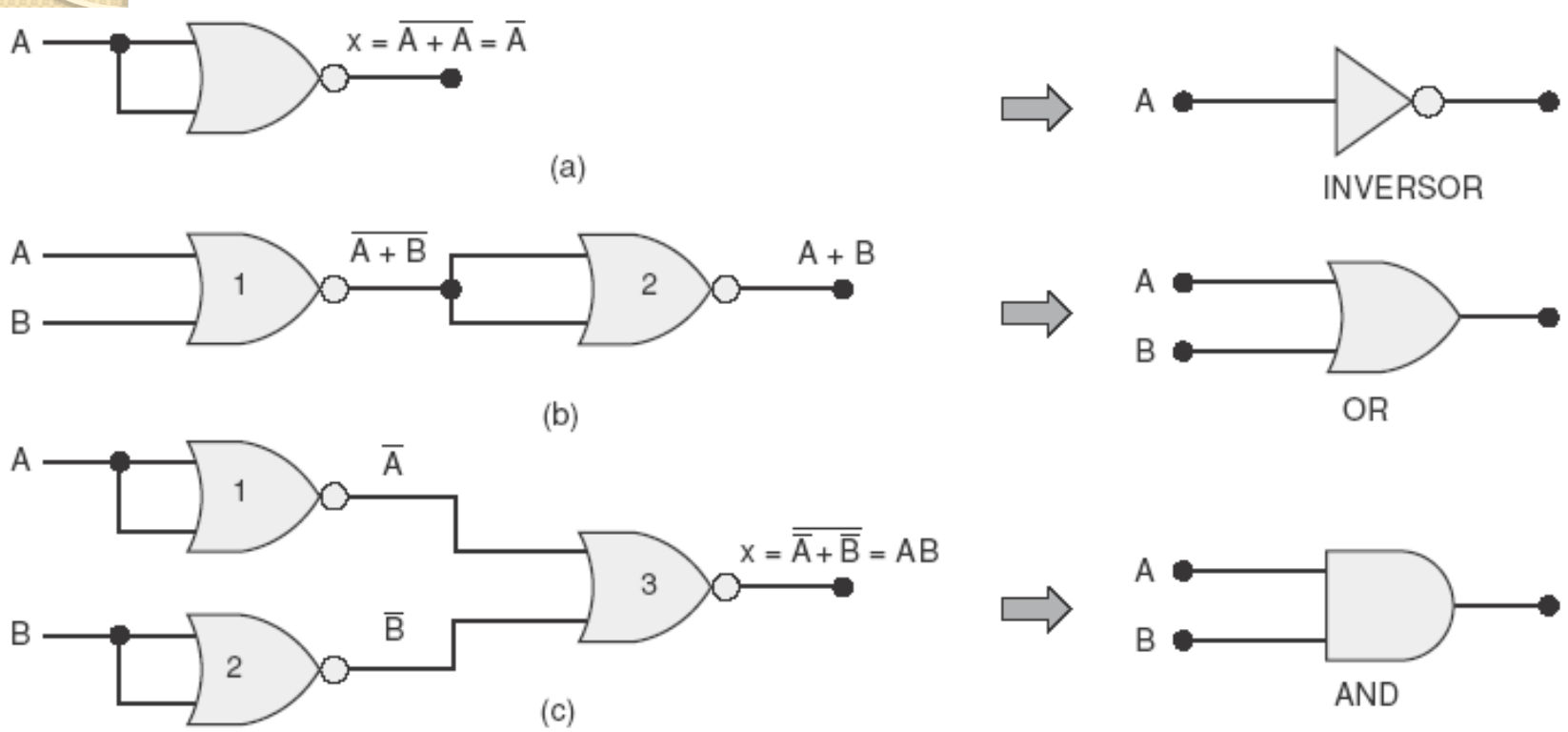
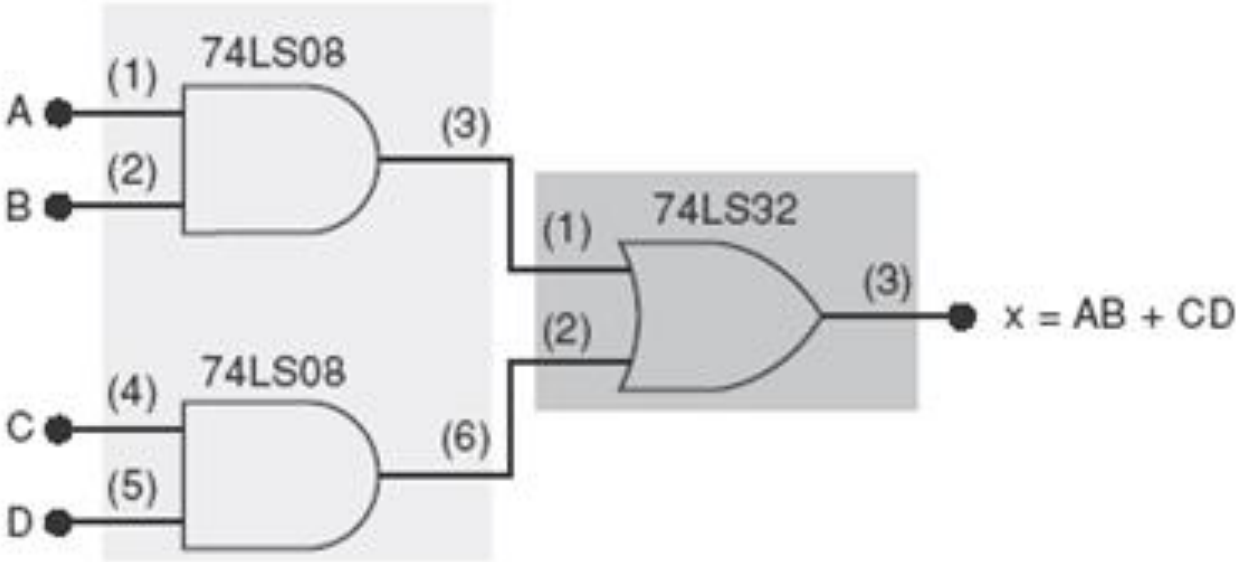


FIGURA 3.30
As portas NOR podem ser usadas para implementar qualquer operação booleana.

Exercício: Implementar o seguinte circuito usando apenas portas NAND



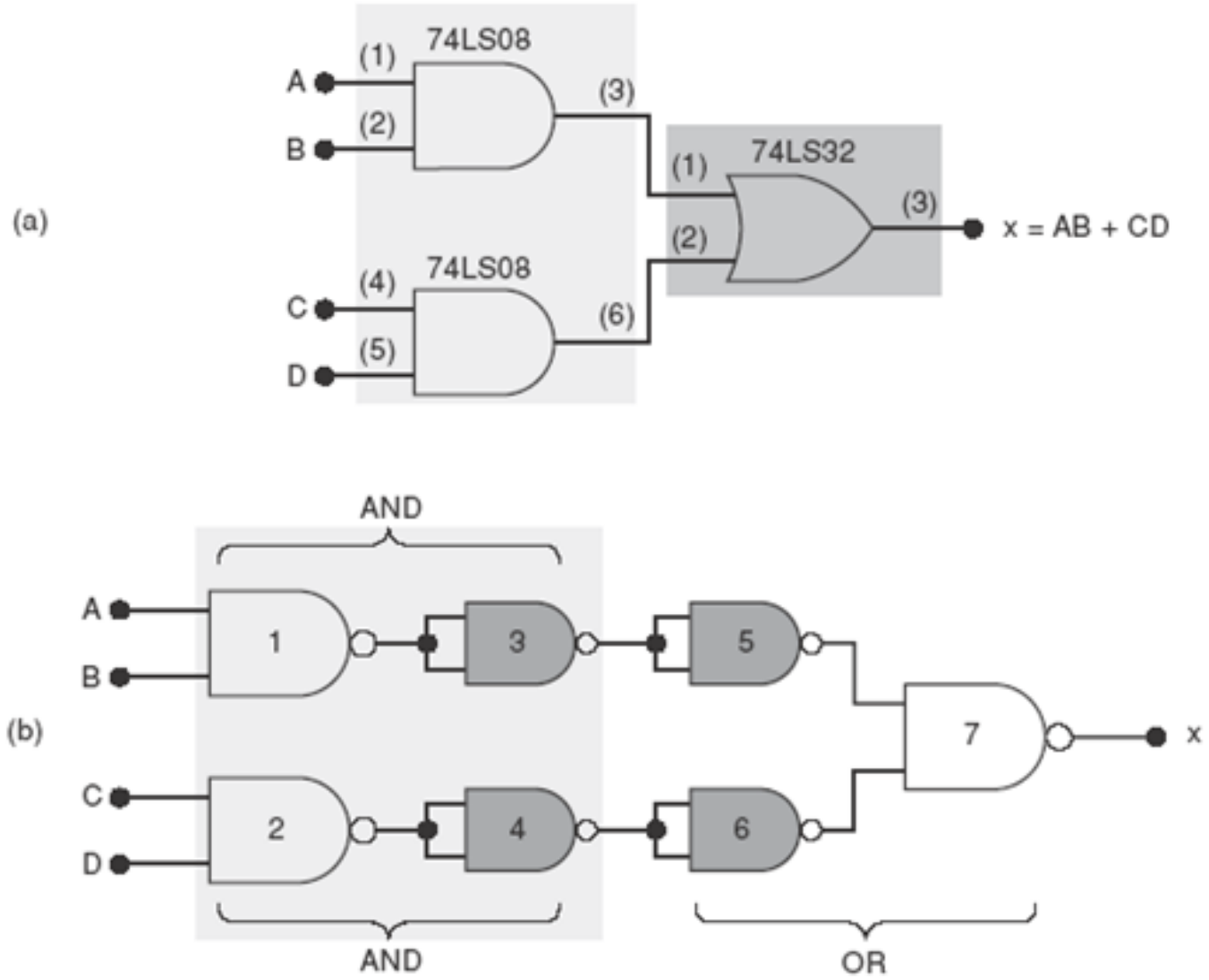


FIGURA 3.32
Implementações possíveis para o Exemplo 3.18.

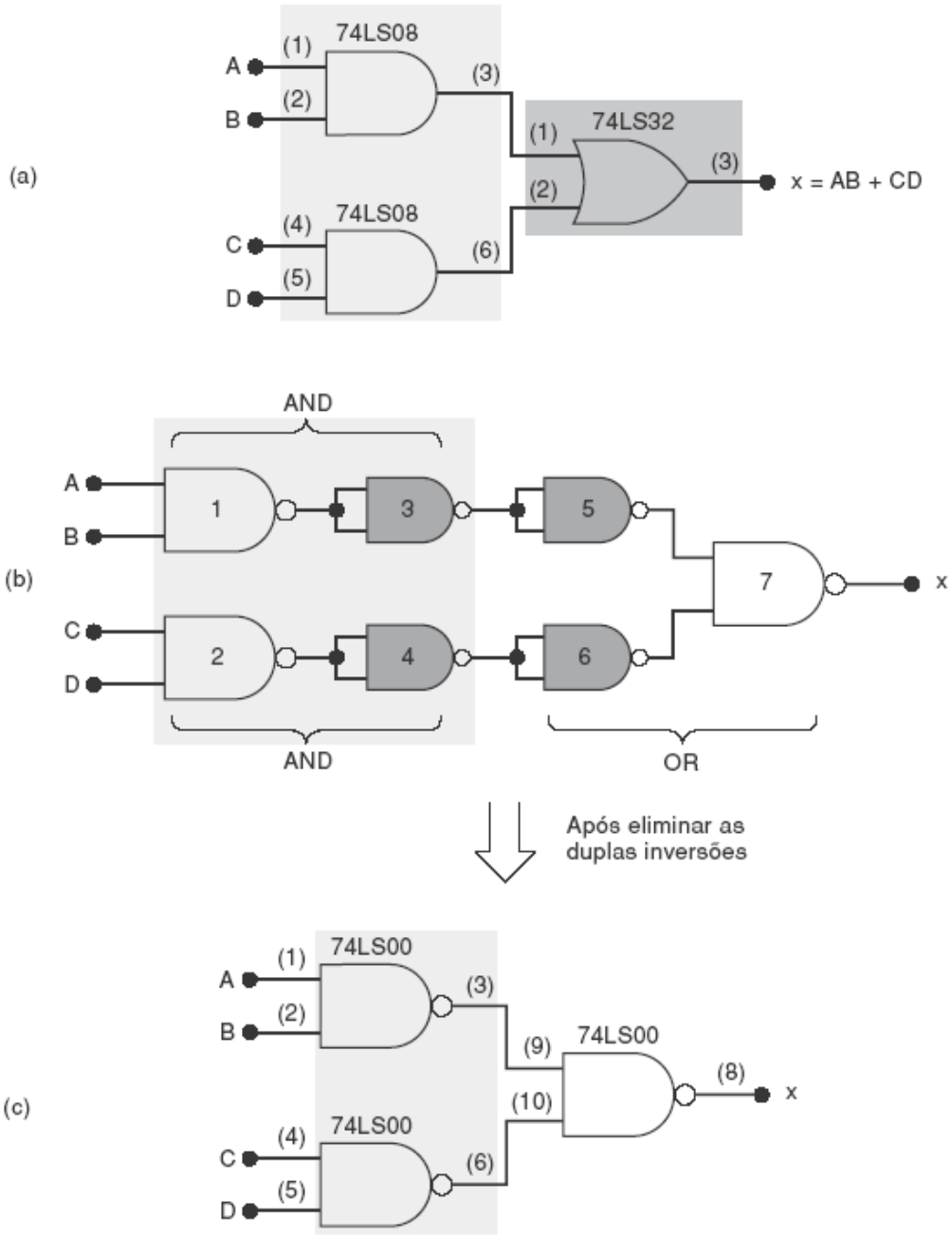
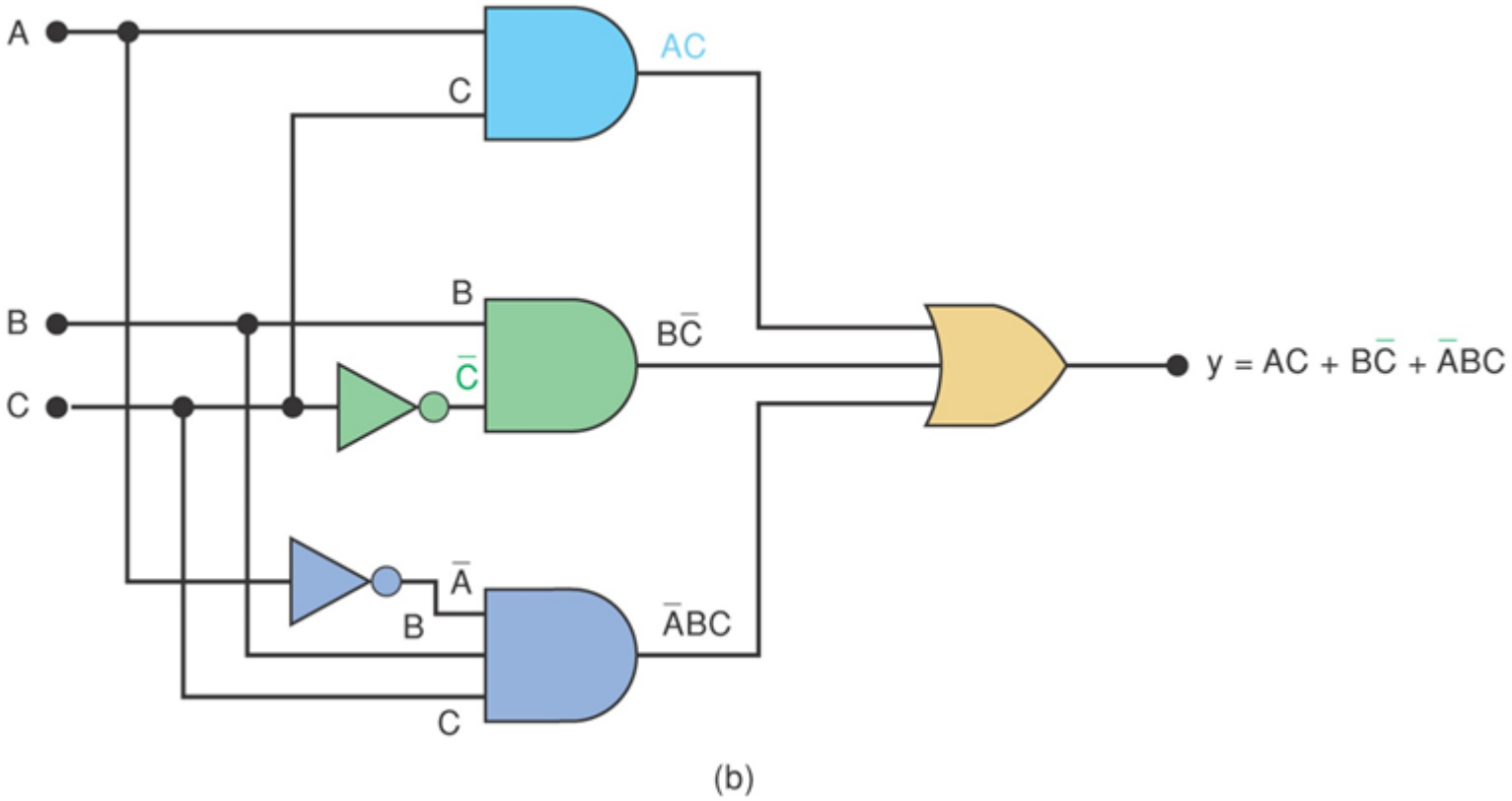


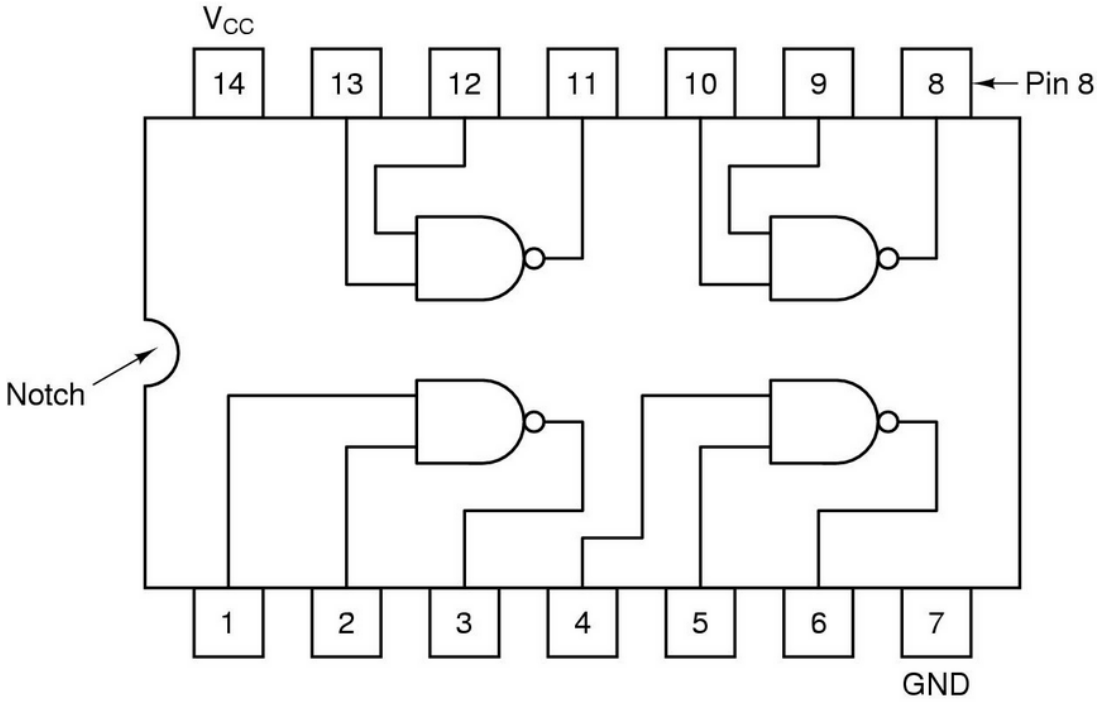
FIGURA 3.32
Implementações possíveis para o Exemplo 3.18.

Refazer o circuito seguinte utilizando apenas portas NAND de duas entradas (CI 7400).

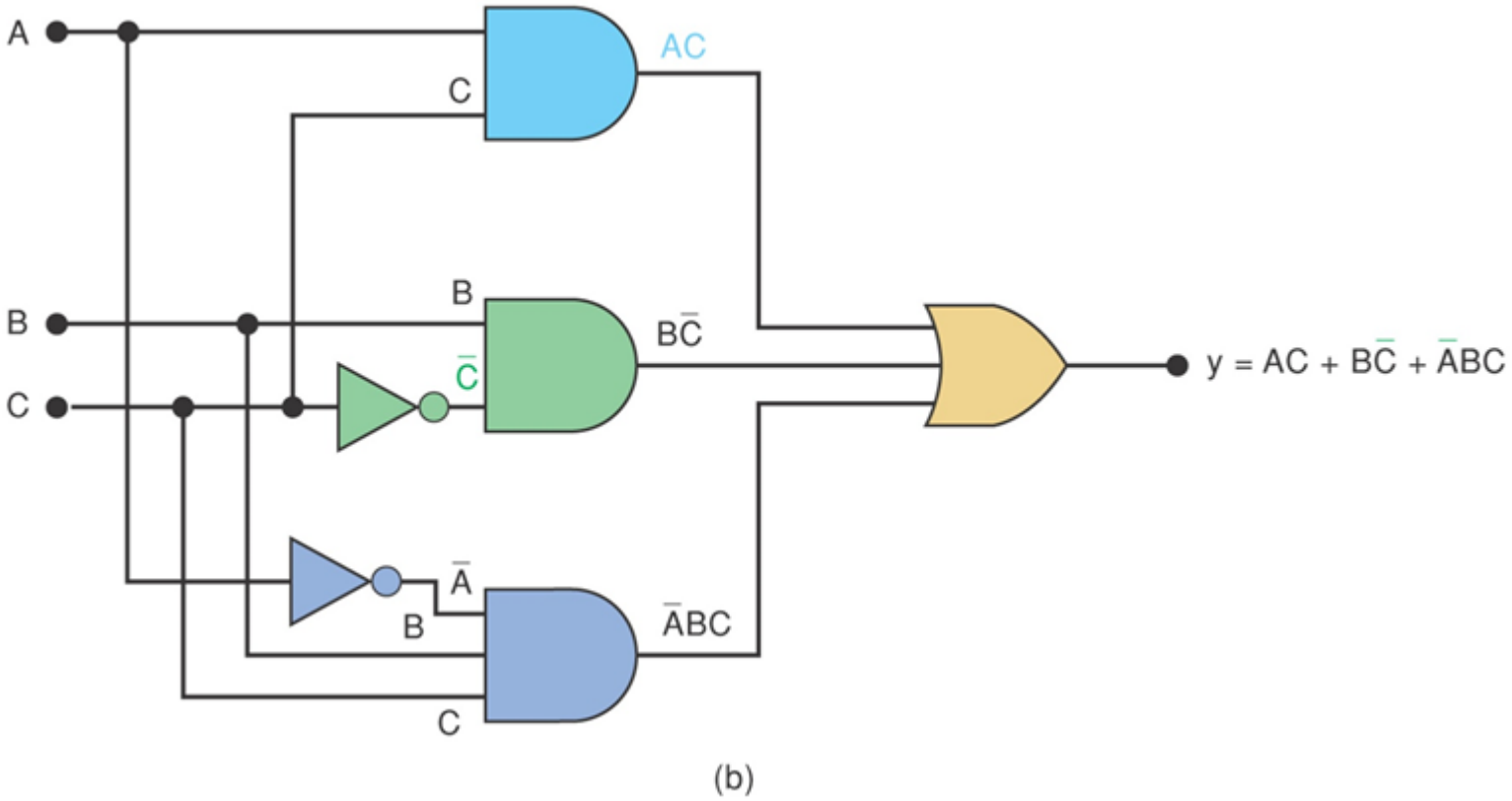


7.7 – Portas Lógicas e Principais Circuitos Integrados

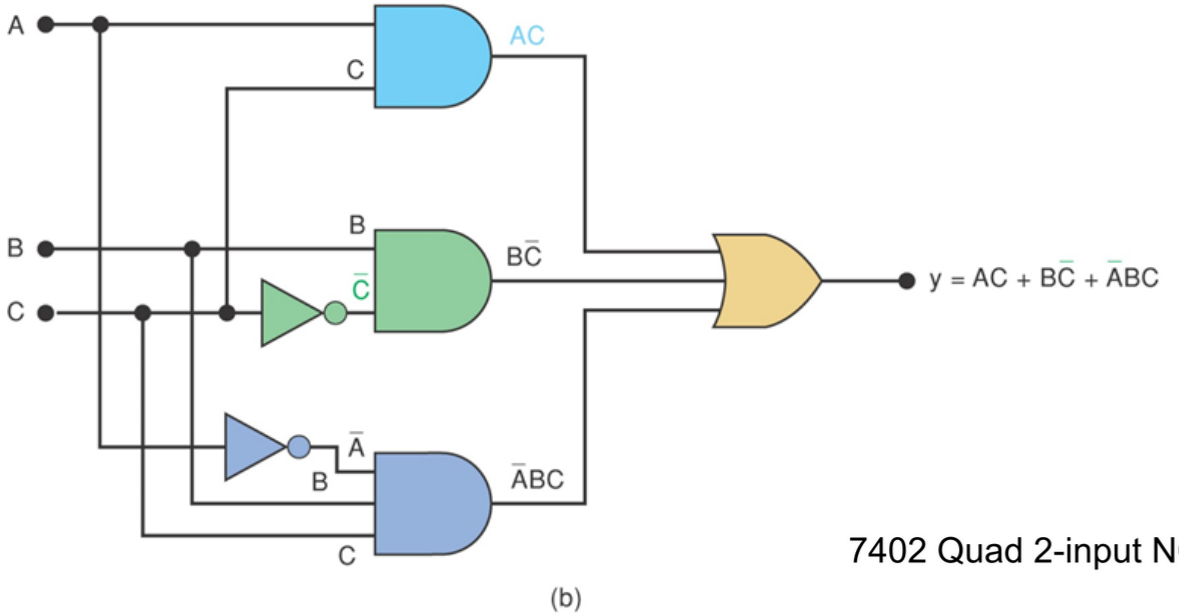
NAND – 7400 – Família TTL



Refazer o circuito seguinte utilizando apenas portas NOR de duas entradas (CI 7400).

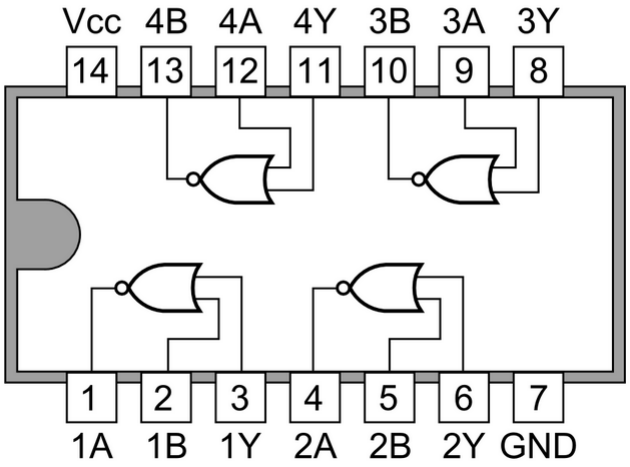


Refazer o circuito seguinte utilizando apenas portas NOR de duas entradas (CI 7400).



(b)

7402 Quad 2-input NOR Gates



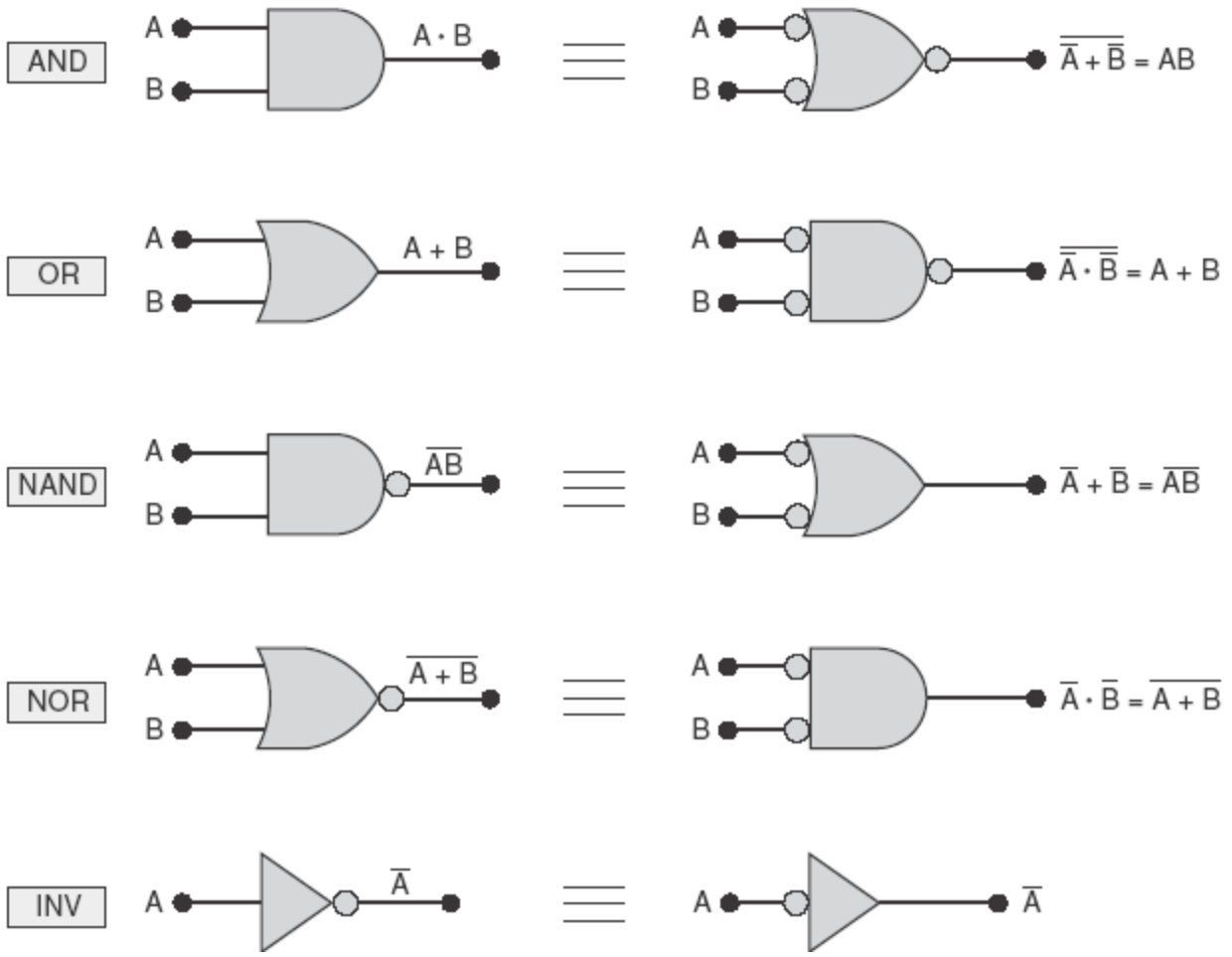
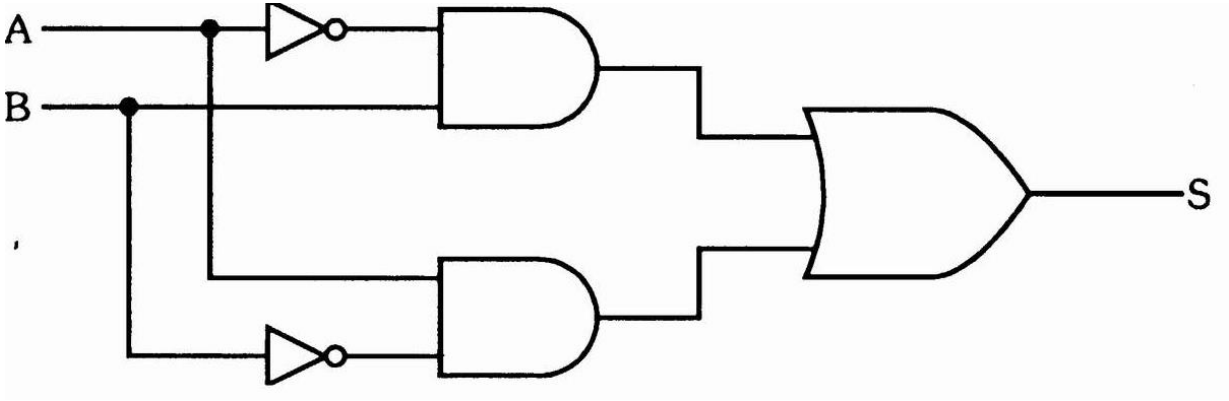
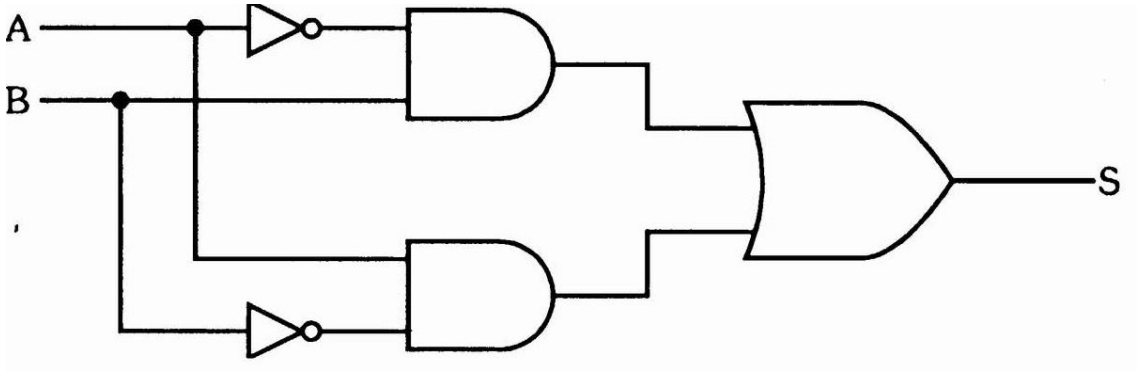


FIGURA 3.33
Símbolos-padrão e alternativos para várias portas lógicas e para o inversor.

Obtenha a tabela-verdade do seguinte circuito:

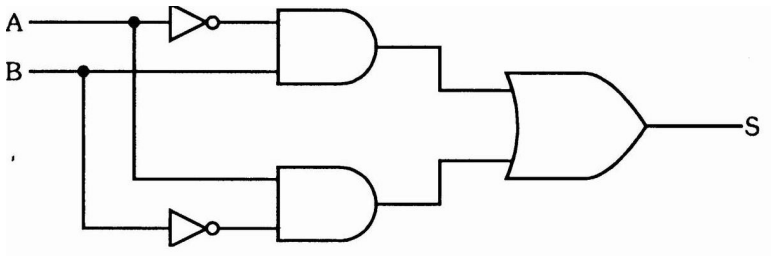


Obtenha a tabela-verdade do seguinte circuito:

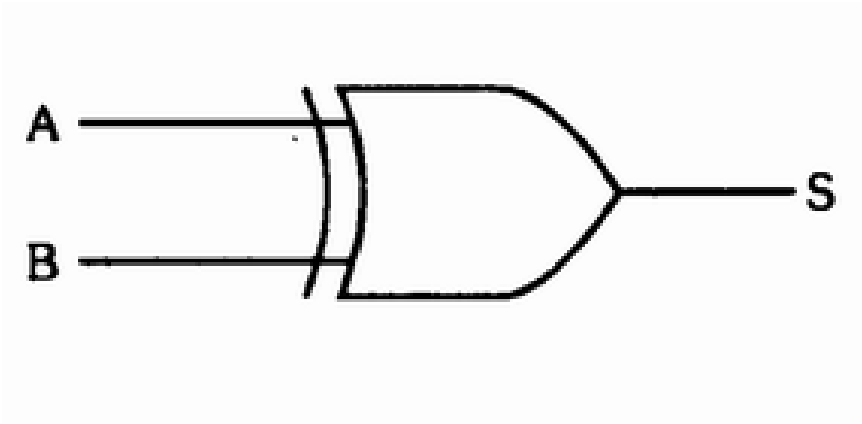


A	B	S
0	0	
0	1	
1	0	
1	1	

Obtenha a tabela-verdade do seguinte circuito:

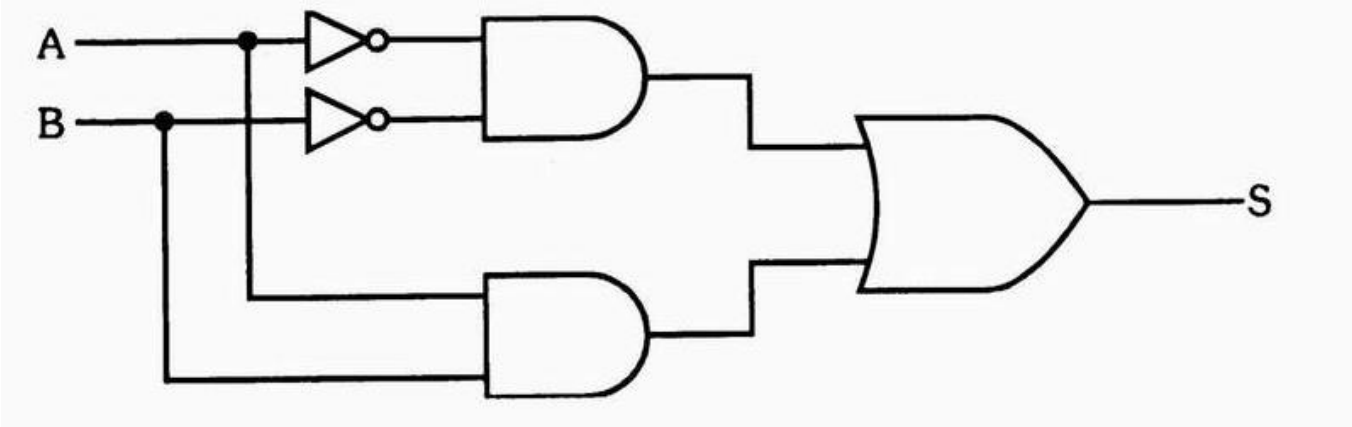


A	B	S
0	0	0
0	1	1
1	0	1
1	1	0



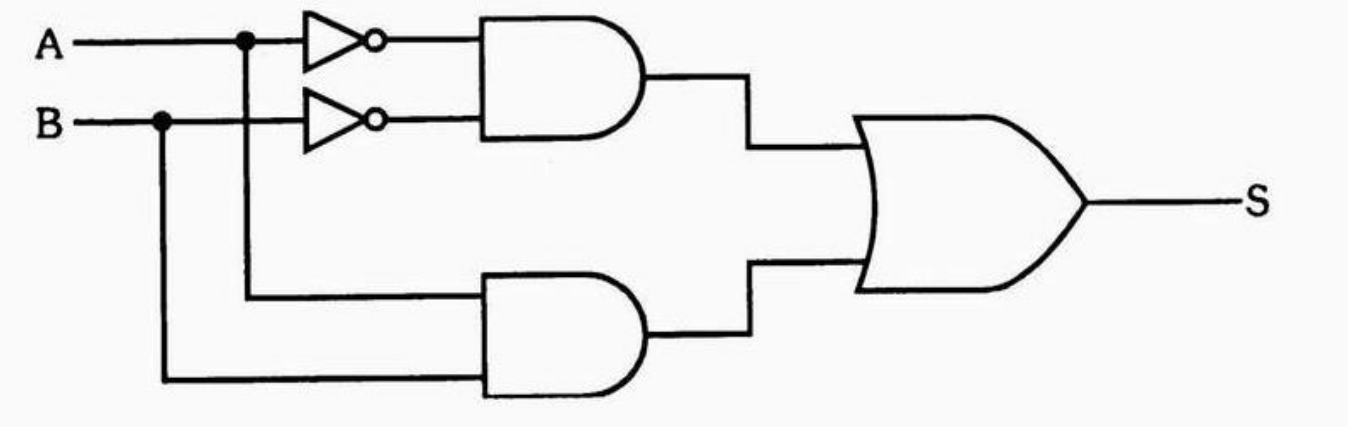
Porta Or-Exclusivo ou XOR

Obtenha a tabela-verdade do seguinte circuito:



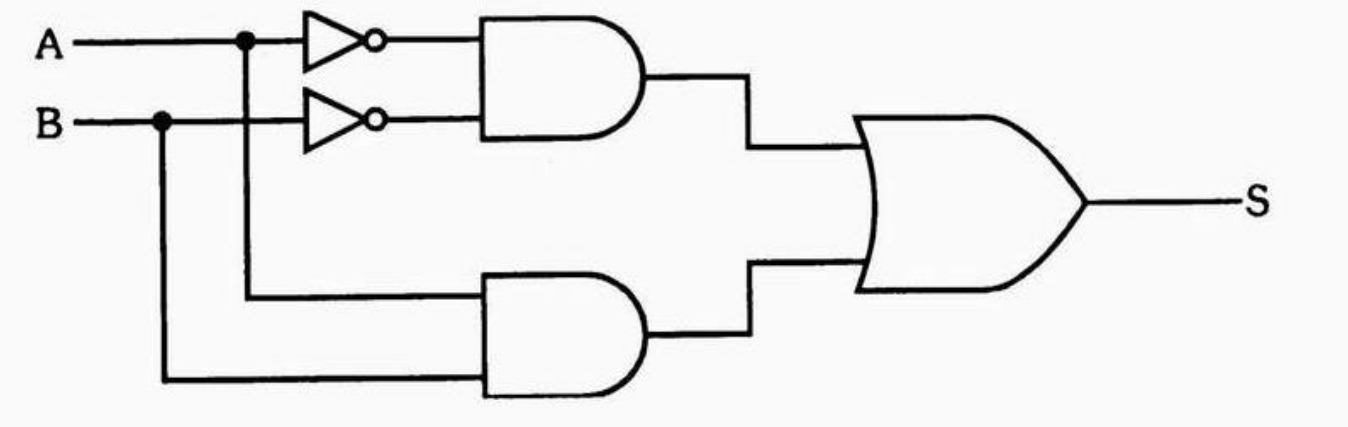
A	B	S
0	0	
0	1	
1	0	
1	1	

Obtenha a tabela-verdade do seguinte circuito:



A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

Obtenha a tabela-verdade do seguinte circuito:

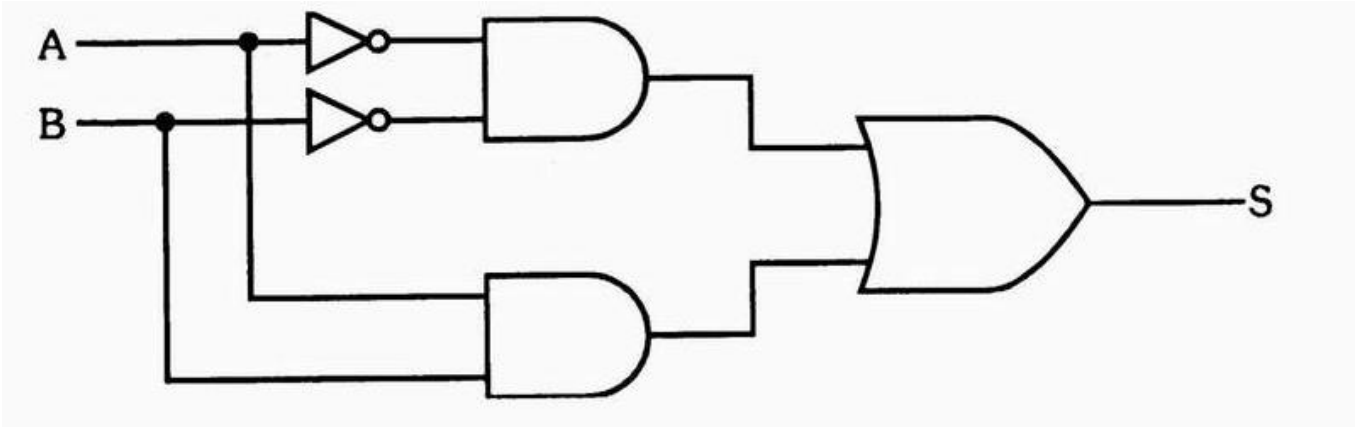


A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

Tabela-Verdade XOR

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

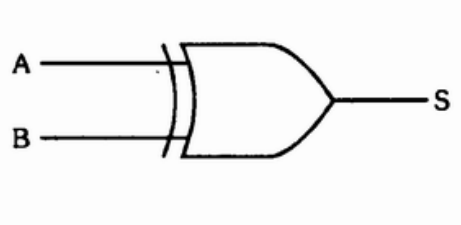
Obtenha a tabela-verdade do seguinte circuito:



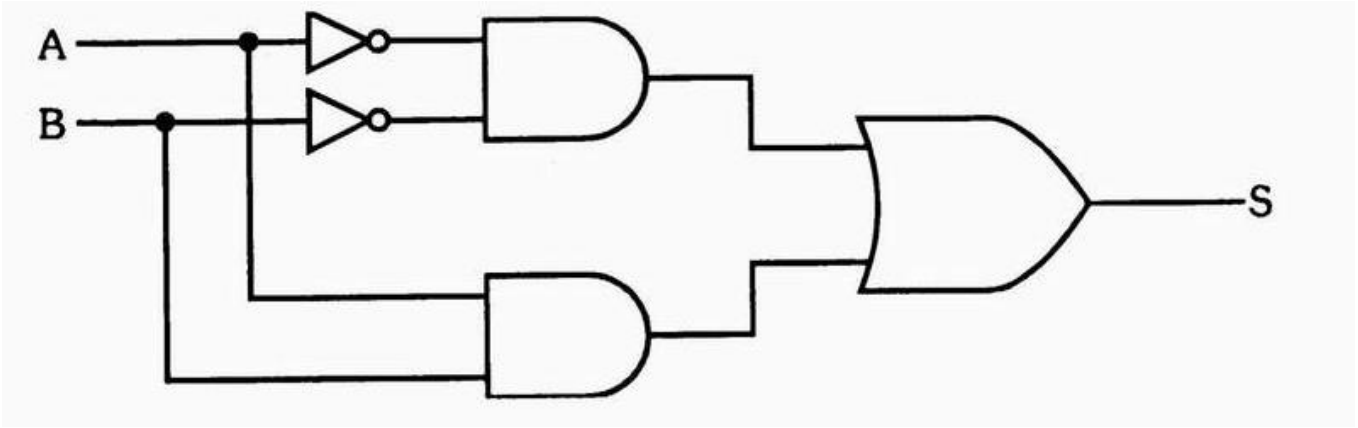
A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

Tabela-Verdade XOR

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0



Obtenha a tabela-verdade do seguinte circuito:



NXOR ou Coincidencia

A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

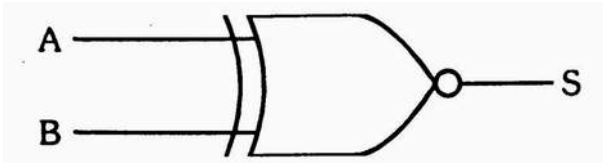
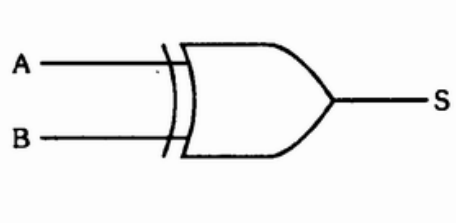


Tabela-Verdade XOR

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0



Equivalencia entre portas

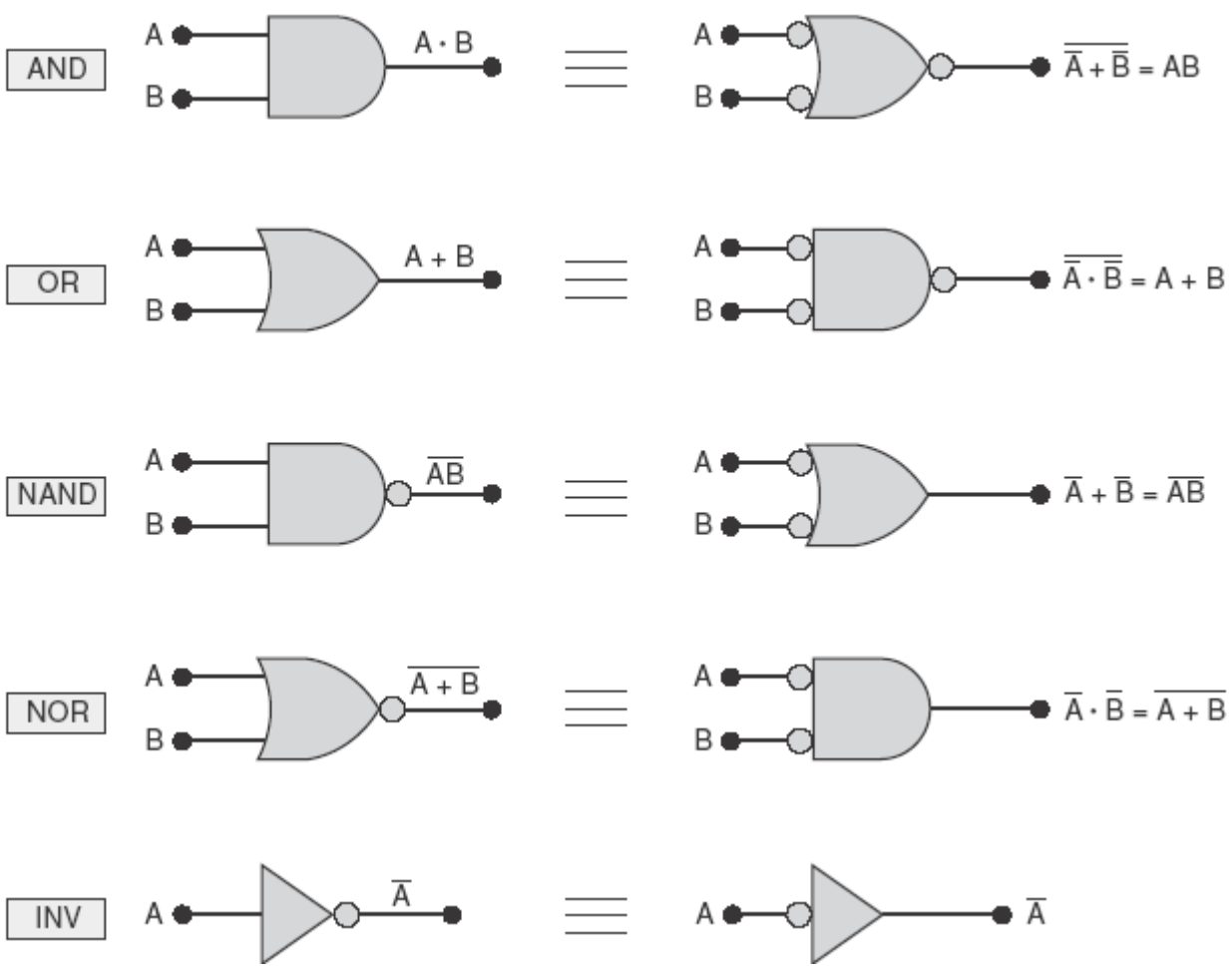
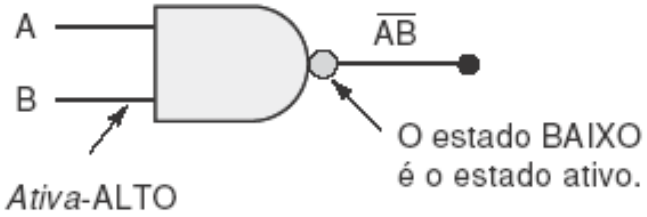


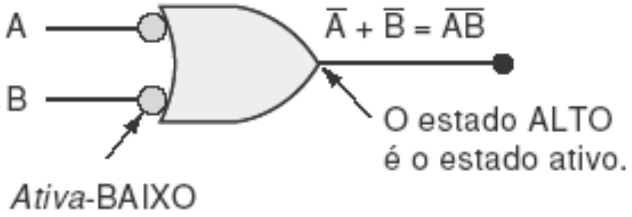
FIGURA 3.33
Símbolos-padrão e alternativos para várias portas lógicas e para o inversor.

Entrada / Saída Ativa – Nível Alto / Nível Baixo



(a)

A saída vai para o nível BAIXO somente quando *todas* as entradas forem para o nivel ALTO.

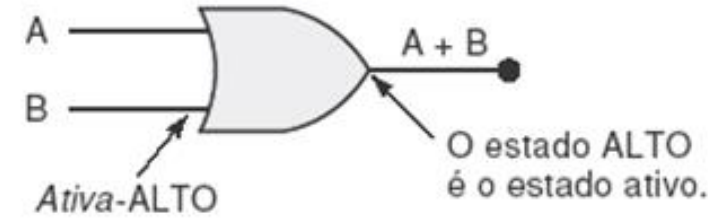


(b)

A saída vai para o nível ALTO quando *qualquer* entrada for para o nivel BAIXO.

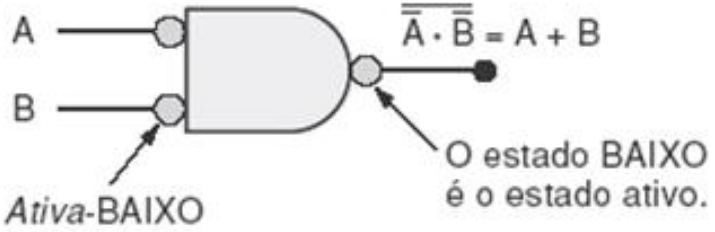
FIGURA 3.34
Interpretação dos dois símbolos da porta NAND.

Entrada / Saída Ativa – Nível Alto / Nível Baixo



A saída vai para o nível ALTO quando *qualquer* entrada for para o nivel ALTO.

(a)

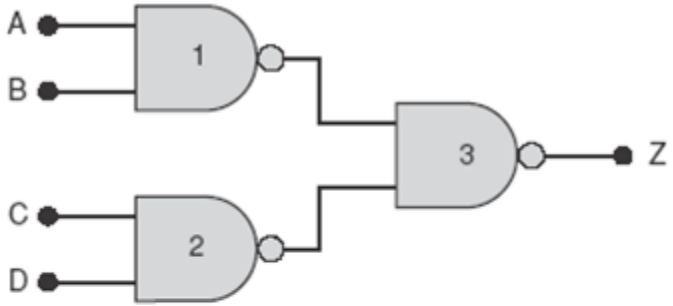


A saída vai para o nível BAIXO somente quando *todas* as entradas forem para o nivel BAIXO.

(b)

FIGURA 3.35
Interpretação dos dois símbolos da porta OR.

FIGURA 3.36
(a) Circuito original usando símbolos-padrão NAND;
(b) Representação equivalente em que a saída *Z* é ativa-ALTO;
(c) Representação equivalente em que a saída *Z* é ativa-BAIXO;
(d) Tabela-verdade.

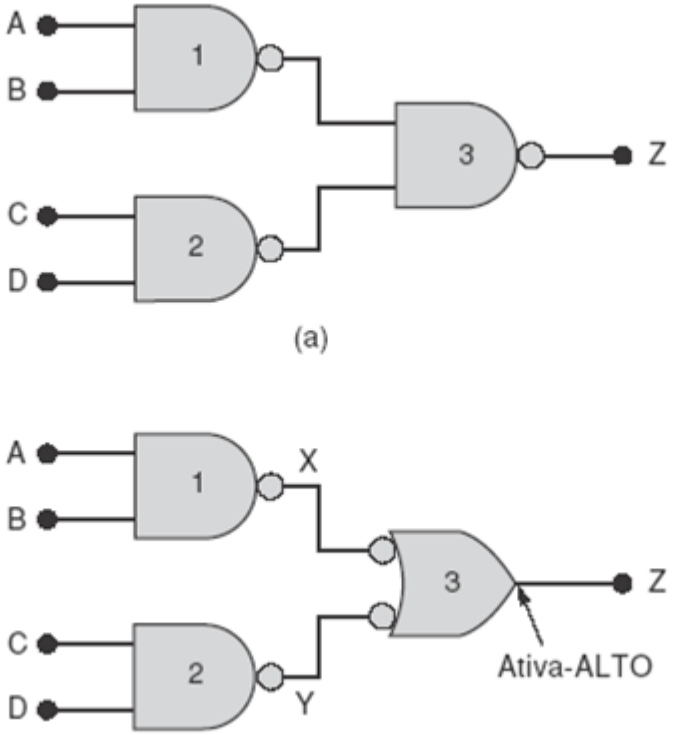


(a)

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

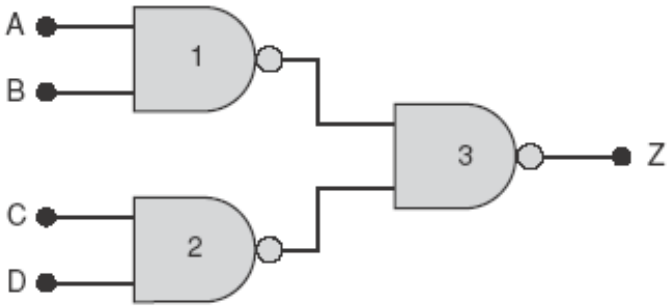
(d)

FIGURA 3.36
(a) Circuito original usando símbolos-padrão NAND;
(b) Representação equivalente em que a saída *Z* é ativa-ALTO;
(c) Representação equivalente em que a saída *Z* é ativa-BAIXO;
(d) Tabela-verdade.

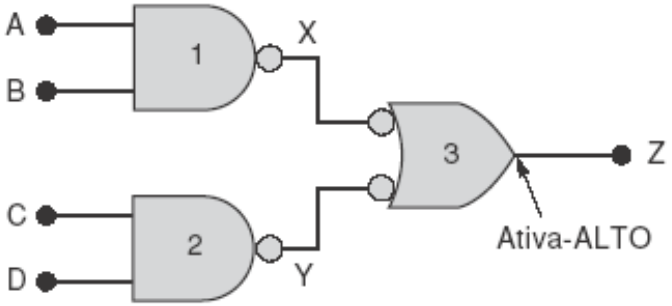


A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

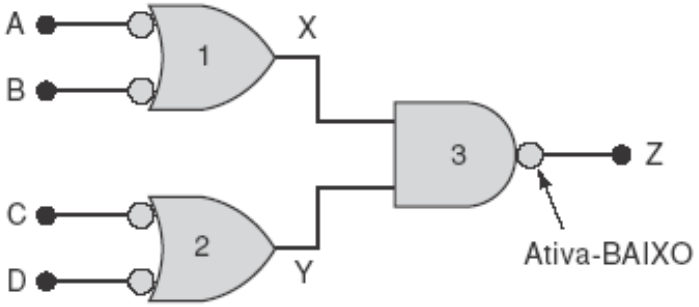
(d)



(a)



(b)

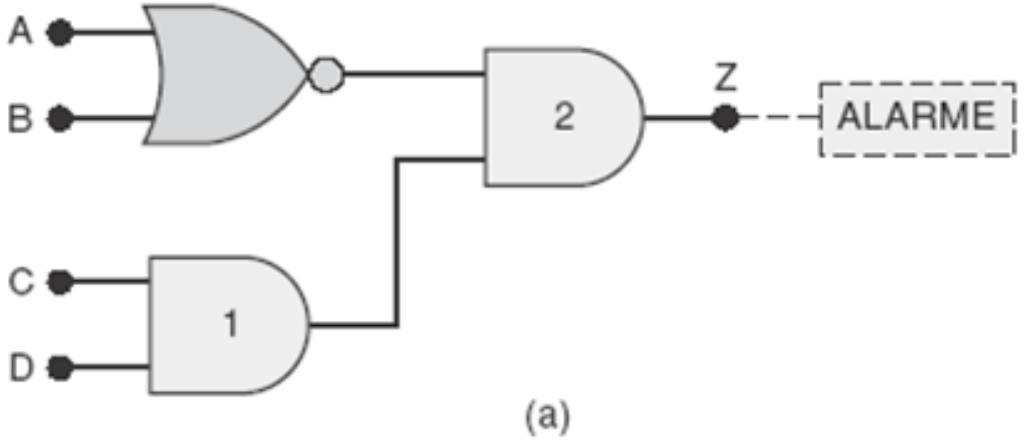


(c)

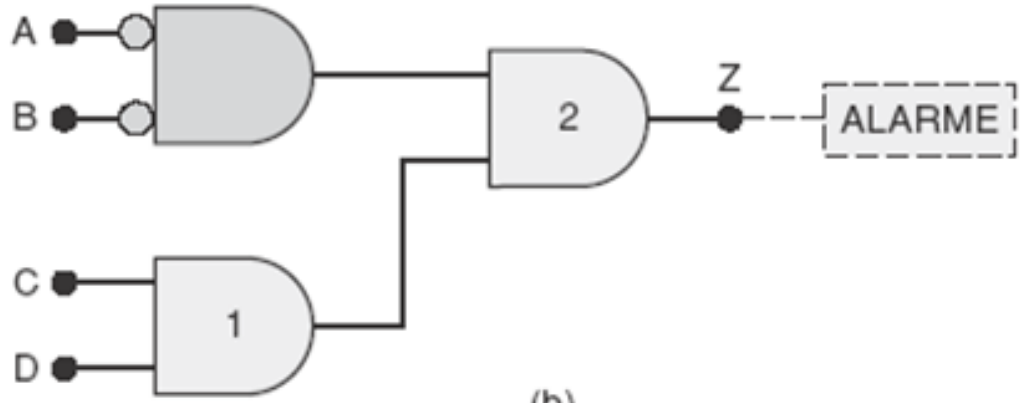
A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

(d)

FIGURA 3.36
(a) Circuito original usando símbolos-padrão NAND; (b) Representação equivalente em que a saída *Z* é ativa-ALTO; (c) Representação equivalente em que a saída *Z* é ativa-BAIXO; (d) Tabela-verdade.



(a)



(b)

FIGURA 3.37
Exemplo 3.20.

Quais são as entradas para MEM = 0 ?

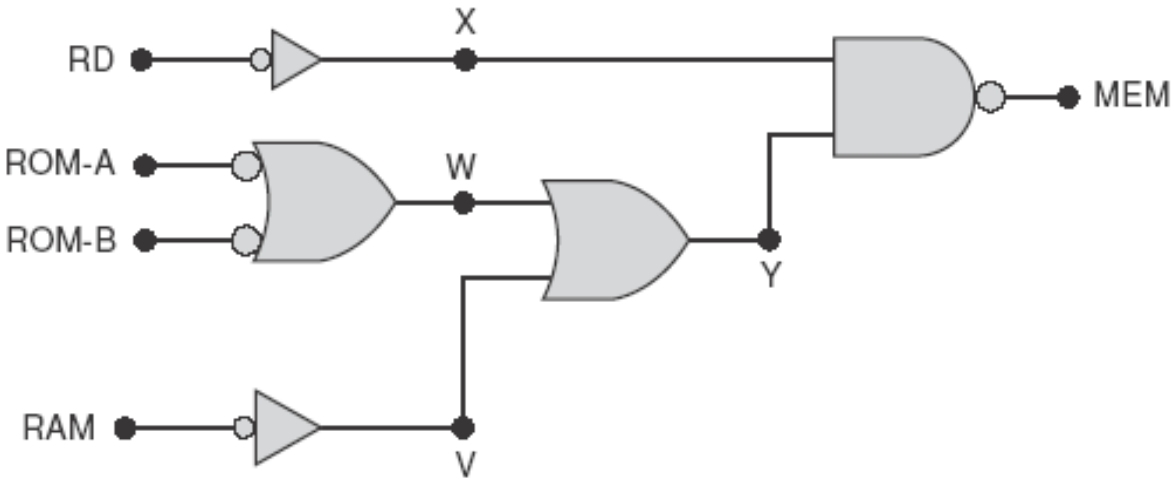


FIGURA 3.39
Exemplo 3.22.

Quais são as entradas para DRIVE =1 ?

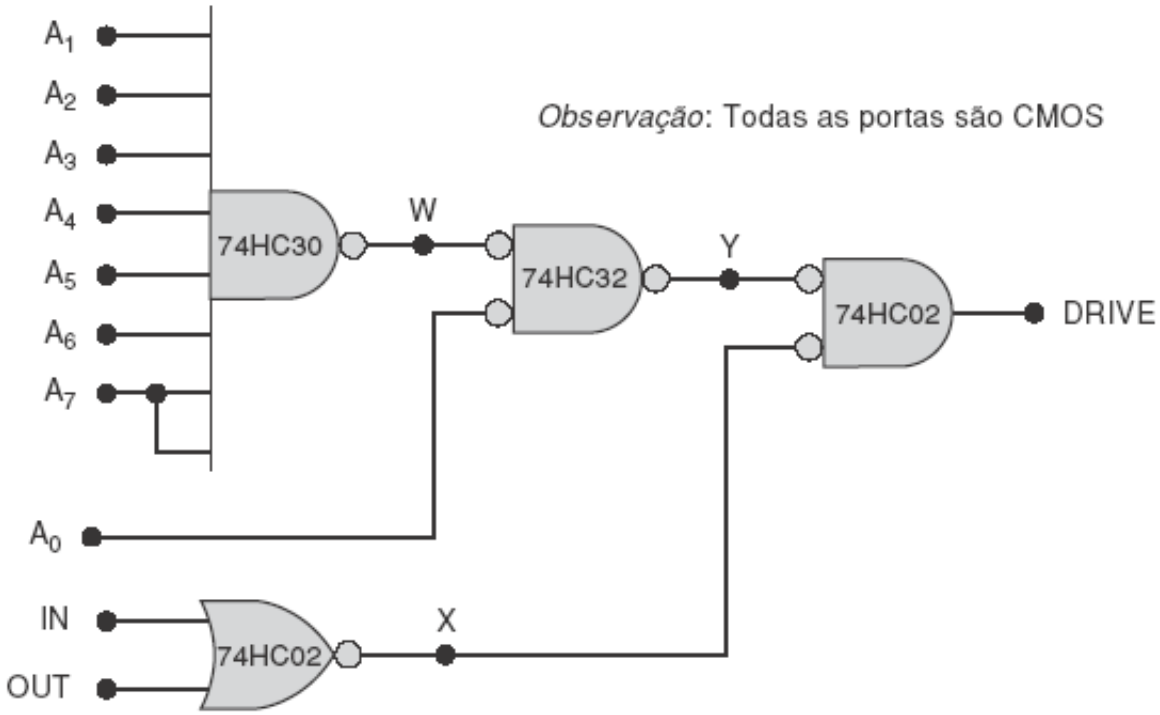


FIGURA 3.40
Exemplo 3.23.