Circuitos Digitais

Álgebra Booleana

Prof. Marcelo Grandi Mandelli
mgmandelli@unb.br

Circuitos Digitais

- Circuitos digitais podem ser descritos através de funções matemáticas:

$$s(e) = \text{LÓGICA}$$

SAÍDA

ENTRADA(s)

Exemplo: e + e

Circuitos Digitais

- Circuitos digitais operam de modo binário :
  - cada tensão de entrada ou saída tem valor 0 ou 1 → intervalos de tensão predefinidos

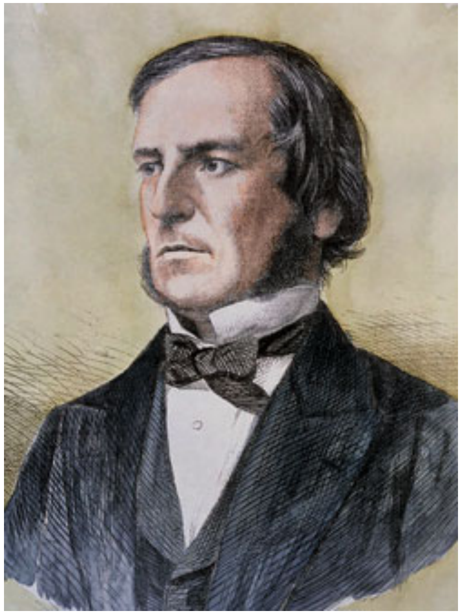
- Essa característica dos circuitos digitais nos permite utilizar a **álgebra booleana**

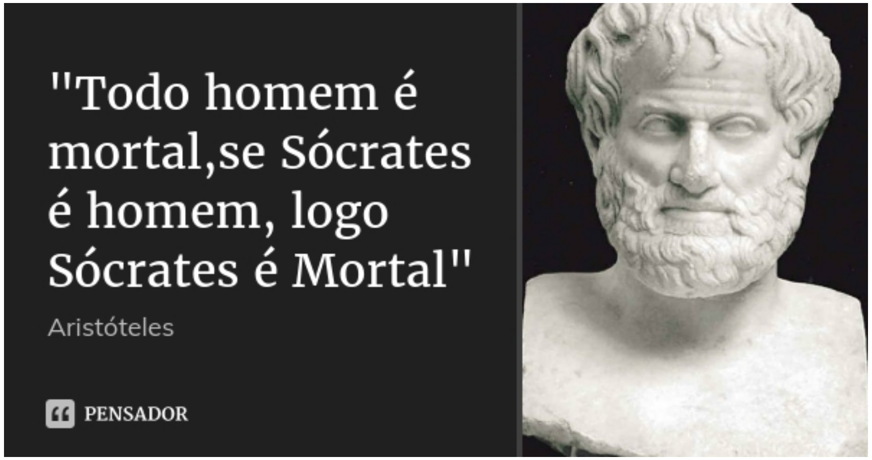
Álgebra Booleana

- Criada por George Boole

- George Boole foi um matemático do século XIX (1815-1864)
  - ele nunca viu um circuito elétrico digital (1950s)
  - nem mesmo a lâmpada elétrica (1879).

- Boole trabalhou com as idéias de Aristóteles → Lógica





Formalizar o pensamento humano

Álgebra Booleana

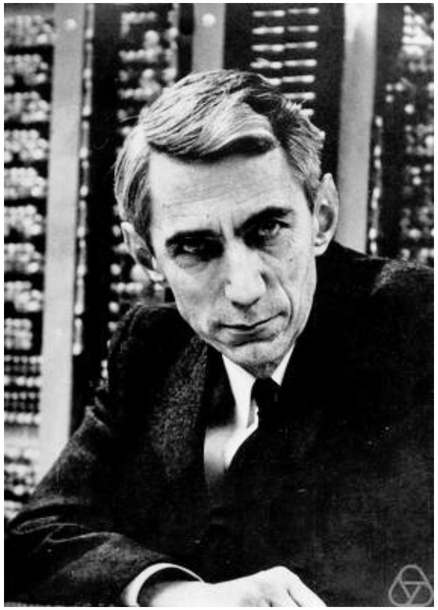
Formalizar o pensamento humano

- Exemplo:
  - Vou sair amanhã se eu tiver dinheiro e se fizer sol
    - X = Vou sair amanhã
      - X = V → Vou sair
      - X = F → Não vou sair
    - D = Eu tiver dinheiro
      - D = V → Eu tenho dinheiro
      - D = F → Eu não tenho dinheiro
    - S = Fizer sol
      - S = V → Faz sol
      - S = F → Não faz sol

X = S e D

Álgebra Booleana

Em 1938, Claude Shannon mostrou que a Álgebra Booleana podia ser usada para projetar circuitos elétricos



Álgebra Booleana

- Nos permite descrever a relação entre a(s) entradas e saída(s) de circuitos digitais através de uma equação → função booleana

- Diferentemente da Álgebra dos Reais, Constantes e Variáveis assumem apenas dois valores **0 (falso)** e **1 (verdadeiro)**

- variáveis lógicas
  - representada por um símbolo → letras do alfabeto
  - Podem representar entradas ou saídas de um circuito
  - Ex.: A = 0 , A = 1

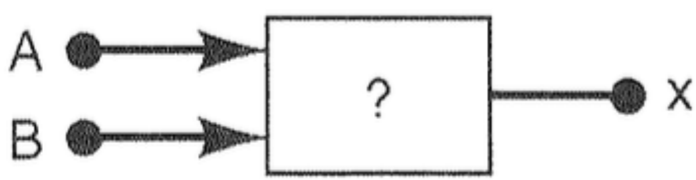
- operações básicas (operações lógicas)
  - OR (OU), AND (E) e NOT (NÃO)

Tabelas-Verdade

- Representam o comportamento da uma função booleana através de uma forma tabular

- Listam **todas as combinações de valores que as variáveis de entrada** podem assumir e os correspondentes valores de saídas

Tabela verdade de 2 entradas



entradas

saída

A	B	X
0	0	1
0	1	0
1	0	1
1	1	0

Quando as entradas A = 0 e B = 1,
a saída X tem valor 0

Tabelas-Verdade

- O número de combinações em uma tabela-verdade será igual a 2^n, onde **n é o número de entradas**

- Lista de combinações segue a sequência de contagem binária

Tabela verdade de 2 entradas

00 : 0 em binário
01 : 1 em binário
10 : 2 em binário
11 : 3 em binário

A	B	X
0	0	1
0	1	0
1	0	1
1	1	0

2^2 = **4** combinações

Tabelas-Verdade

Tabela verdade de 3 entradas

A	B	C	X
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$2^3 = 8$

Intercala quatro 0s e quatro 1s

Intercala dois 0s e dois 1s

Intercala um 0 e um 1

Tabelas-Verdade

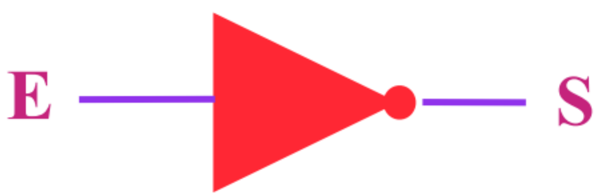
Tabela verdade de 4 entradas

$2^4 = 16$

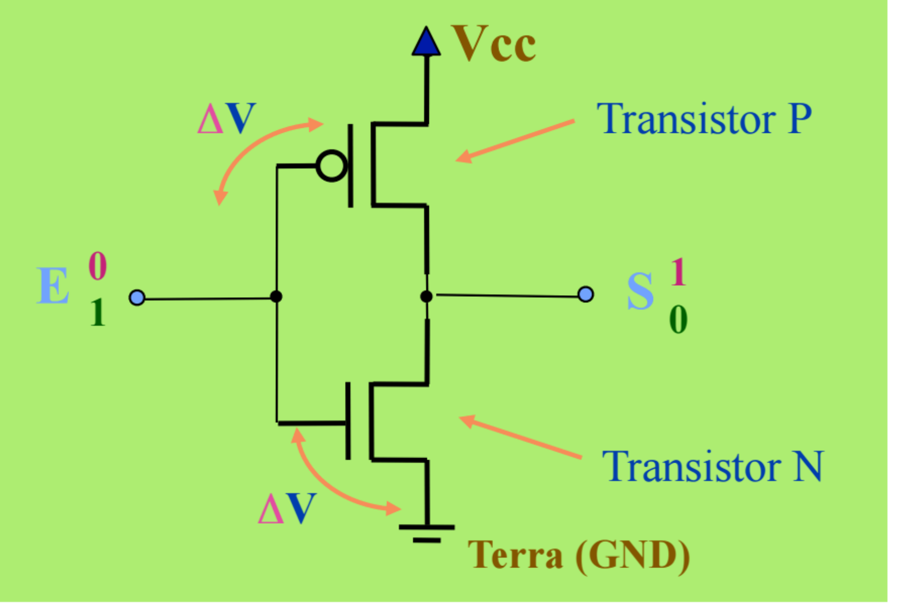
A	B	C	D	X
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Álgebra Booleana

- Portas lógicas
  - Circuitos digitais que implementam operadores lógicos
  - as saídas são resultado de uma operação lógica das entradas
  - Construídas a partir de transistores, resistores e diodos



PORTA NOT

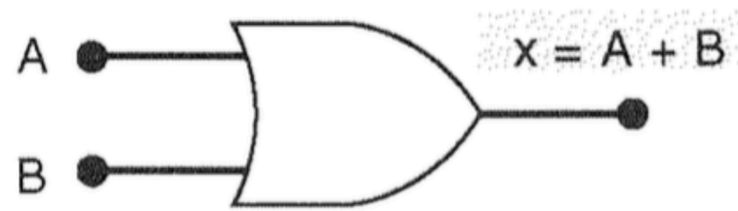


Operações Booleanas

- Operação OR (OU, Adição Lógica)
  - A operação **OR** resulta **1** se pelo menos uma das variáveis de entrada vale **1**
  - Representação:
    - $A + B$, A v B, A | B, A or B

OR

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

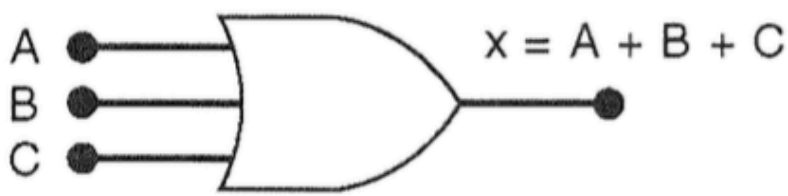


Porta OR

Operações Booleanas

- Operação OR (OU, Adição Lógica)

  - Porta OR de três entradas



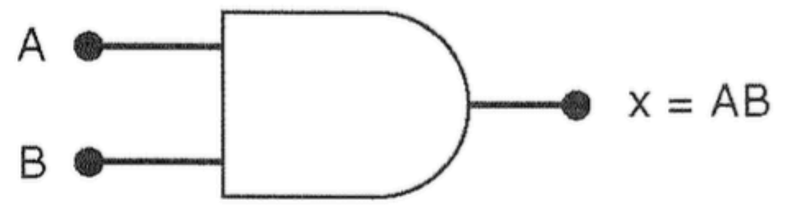
A	B	C	x = A + B + C
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Operações Booleanas

- Operação AND (E, Multiplicação Lógica)
  - A operação **E** resulta **0** se pelo menos uma das variáveis de entrada vale **0**
  - Representação:
    - $A \cdot B$, AB, $A \wedge B$, A & B, A and B

AND

A	B	$x = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

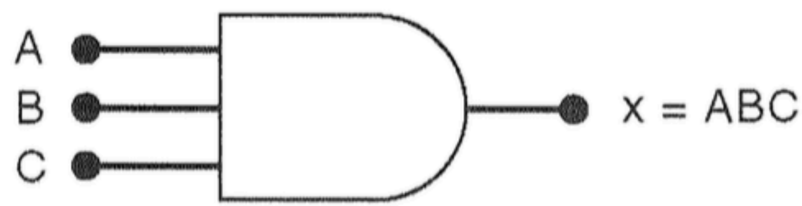


Porta AND

Operações Booleanas

- Operação AND (E, Multiplicação Lógica)
  - Porta AND de três entradas

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

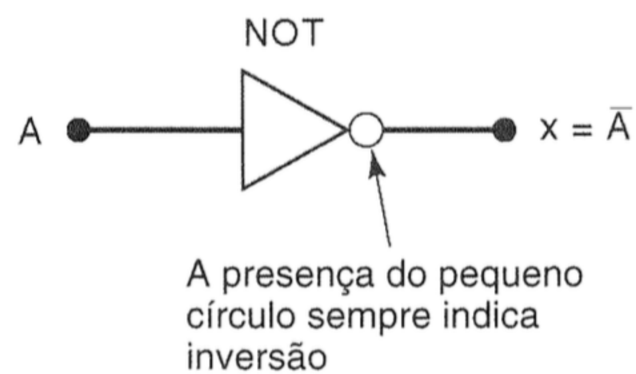


Operações Booleanas

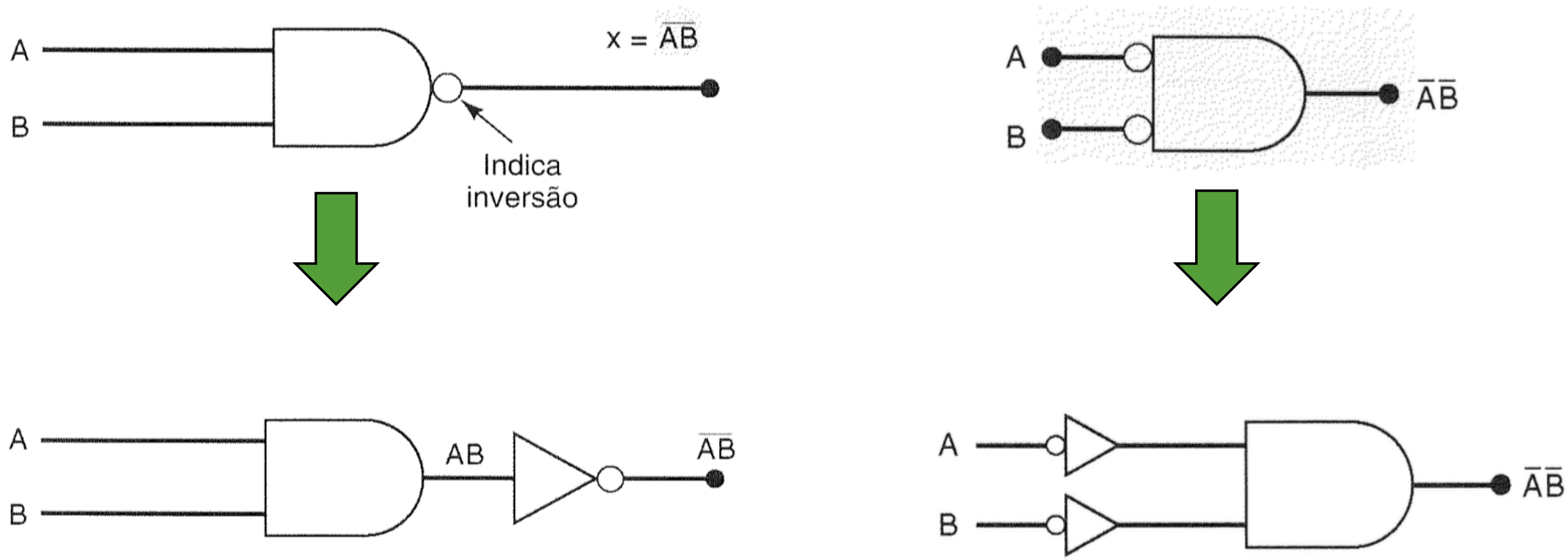
- Operação NOT (NÃO, Inversor, Complemento)
  - Esta operação resulta no valor complementar ao que uma variável apresenta:
    - o valor complementar será **1** se a variável vale **0** e será **0** se a variável vale **1**
  - Representação:
    - **A′**, $\overline{\mathbf{A}}$, ¬A, $\overline{(A+B)}$, (A+B)′, !(A+B)

NOT

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



Operações Booleanas



$\overline{AB}$ é diferente de $\overline{A}\,\overline{B}$!!

Ordem de precedência

$$\{\,\},\ [\,],\ (\,),\ \overbrace{\text{NOT},\ \text{AND},\ \text{OR},\ \text{NOT}}^{\text{avaliados da esquerda para a direita}}$$

sobre uma variável (NOT vermelho) — sobre termo(s) (NOT azul)

- Exemplos:

$F = (A + B') \cdot C + D'$

1. B'
2. $A + B'$
3. $(A + B') \cdot C$
4. $(A + B') \cdot C + D'$

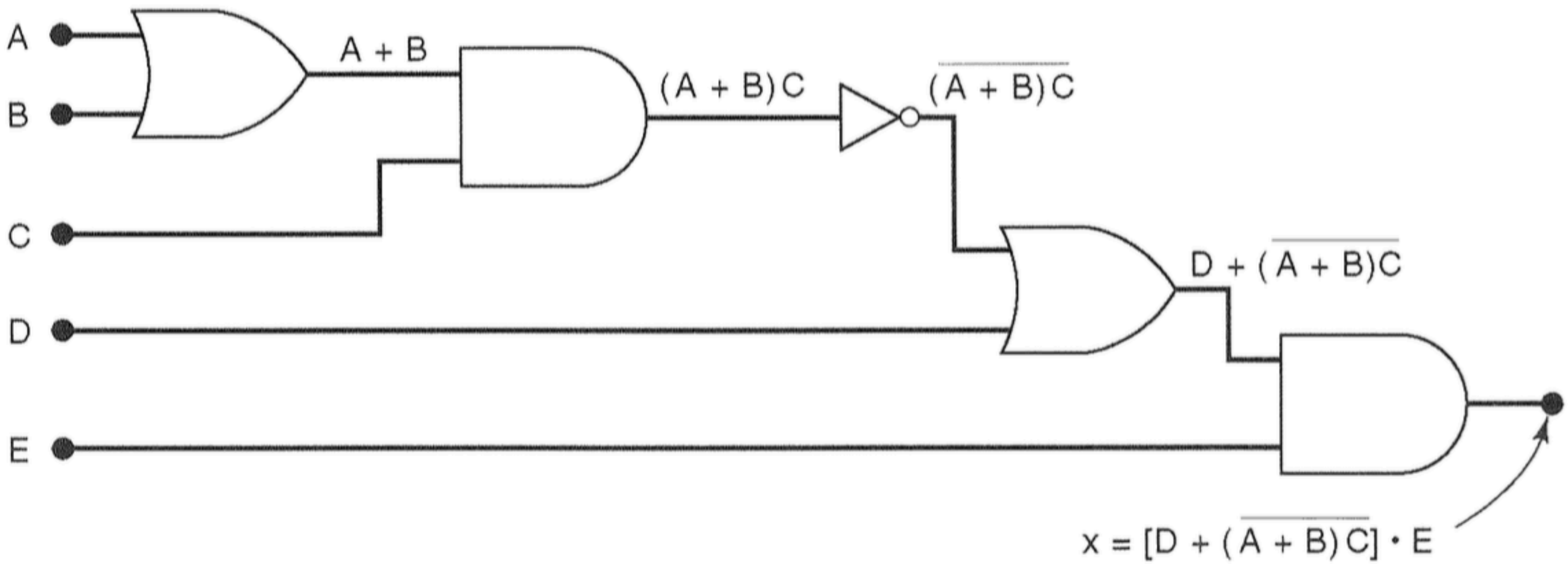
$F = \overline{A}BC(\overline{A + D})$

1. $A+D$
2. $(\overline{A+D})$
3. $\overline{A}$
4. $\overline{A}BC$
5. $\overline{A}BC(\overline{A + D})$

Circuitos a partir de expressões booleanas

{ }, [ ], ( ), NOT, AND, OR, NOT

- Exemplo : $X = [D + \overline{(A+B)C}] \cdot E$



LEIS E REGRAS DA ÁLGEBRA BOOLEANA

1. Lei comutativa

a) $A+B = B+A$

b) $AB = BA$

2. Lei associativa

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

b) $A(BC) = (AB)C$

3. Lei distributiva

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

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

LEIS E REGRAS DA ÁLGEBRA BOOLEANA

4. $A + 0 = A$

A	B	X
0	**0**	**0**
0	1	1
1	**0**	**1**
1	1	1

5. $A \cdot 0 = 0$

A	B	X
0	**0**	**0**
0	1	0
1	**0**	**0**
1	1	1

6. $A + 1 = 1$

A	B	X
0	0	0
0	**1**	**1**
1	0	1
1	**1**	**1**

7. $A \cdot 1 = A$

A	B	X
0	0	0
0	**1**	**0**
1	0	0
1	**1**	**1**

LEIS E REGRAS DA ÁLGEBRA BOOLEANA

8. $A + A = A$

A	B	X
0	**0**	**0**
0	1	1
1	0	1
1	**1**	**1**

9. $A \cdot A = A$

A	B	X
0	**0**	**0**
0	1	0
1	0	0
1	**1**	**1**

10. $A + A' = 1$

A	B	X
0	0	0
0	**1**	**1**
1	**0**	**1**
1	1	1

11. $A \cdot A' = 0$

A	B	X
0	0	0
0	**1**	**0**
1	**0**	**0**
1	1	1

LEIS E REGRAS DA ÁLGEBRA BOOLEANA

12. $(A')' = A$

A	X
0	1
1	0

13. $A + AB = A$

$A(1+B)$ → Lei distributiva (3)

$A(1)$ → Regra 6 ($A + 1 = 1$)

A → Regra 7 ($A \cdot 1 = A$)

14. $A(A+B) = A$

$AA + AB$ → Lei distributiva (3)

$A + AB$ → Regra 9 ($AA = A$)

A → Regra 13 ($A + AB = A$)

LEIS E REGRAS DA ÁLGEBRA BOOLEANA

15. $A + A'B = A + B$

$(A+A')(A+B)$ → Lei distributiva (3)

$1 \cdot (A+B)$ → Regra 10 ($A + A' = 1$)

$A + B$ → Regra 7 ($A \cdot 1 = A$)

16. $A(A'+B) = AB$

$AA' + AB$ → Lei distributiva (3)

$0 + AB$ → Regra 11 ($A \cdot A' = 0$)

AB → Regra 4 ($A + 0 = A$)

LEIS E REGRAS DA ÁLGEBRA BOOLEANA

17. Teorema do Consenso

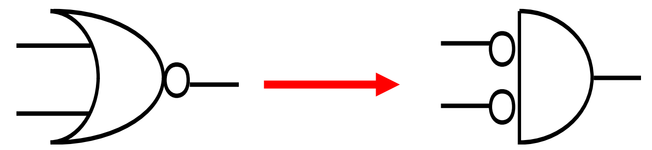
a) $AB + A'C + BC = AB + A'C$

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

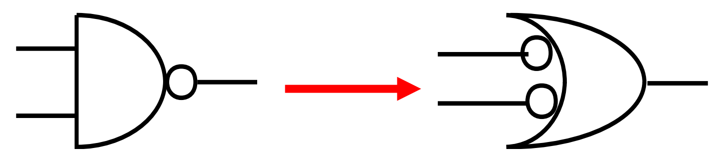
LEIS E REGRAS DA ÁLGEBRA BOOLEANA

18. De Morgan

a) $\overline{A + B} = \overline{A}\,\overline{B}$



b) $\overline{AB} = \overline{A} + \overline{B}$



Avaliação de Funções Booleanas

- Avaliação da Função

Exemplo: DeMorgan $(A + B)' = A' \cdot B'$

(A + B)'

A	B	A + B	(A+B)'
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

A' . B'

A	B	A'	B'	A' .B'
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

=

LEIS E REGRAS DA ÁLGEBRA BOOLEANA

- Em qualquer uma das Leis e Regras, uma variável qualquer pode ser substituída por uma expressão qualquer:

- Exemplo:
  - X + 1 = 1 → (substituo X por AB + C +EF)
  - AB + C +EF + 1 = 1

LEIS E REGRAS DA ÁLGEBRA BOOLEANA

Dualidade

- As leis e regras da álgebra booleana são definidas em pares: para a operação OU e para a operação E
- Se soubermos um dos componetes do par, podemos obter o outro a partir das seguintes substituições:
  - 0 → 1
  - 1 → 0
  - $\cdot$ → +
  - + → $\cdot$
- Exemplo:

$$A + (A' \cdot B) = A + B$$

$$A \cdot (A' + B) = A \cdot B$$

$$A + 1 = 1$$

$$A \cdot 0 = 0$$

Exemplo 1 - Simplicação

- $\mathbf{S = AB\overline{C} + A\overline{B}\,\overline{C}}$
- $S = A\overline{C}(B + \overline{B})$ → Lei distributiva (3)
- $S = A\overline{C}(1)$ → Regra 10 (A + A' = 1)
- $\mathbf{S = A\overline{C}}$ → Regra 7 (A • 1 = A)

Exemplo 2 - Simplicação

- $\mathbf{S = \left(\overline{A} + B\right) \cdot (A + B)}$
- $S = \overline{A}A + \overline{A}B + BA + BB$ → Lei distributiva (3)
- $S = 0 + \overline{A}B + BA + BB$ → Regra 11 (A • A' = 0)
- $S = \overline{A}B + BA + BB$ → Regra 4 (A + 0 = A)
- $S = \overline{A}B + BA + B$ → Regra 9 (A • A = A)
- $S = B(\overline{A} + A + 1)$ → Lei distributiva (3)
- $S = B(1)$ → Regra 6 (A + 1 = 1)
- $\mathbf{S = B}$ → Regra 7 (A • 1 = A)

Exemplo 3 - Simplicação

- $\mathbf{S} = \overline{(\overline{\mathbf{A}} + \mathbf{C}) \cdot (\overline{\mathbf{A}} + \overline{\mathbf{C}})}$
- $S = \overline{(\overline{A} + C)} + \overline{(\overline{A} + \overline{C})}$ → De Morgan
- $S = A\overline{C} + AC$ → De Morgan
- $S = A(\overline{C} + C)$ → De Morgan
- $S = A(1)$ → Regra 10 (A + A' = 1)
- $\mathbf{S} = \mathbf{A}$ → Regra 7 (A • 1 = A)

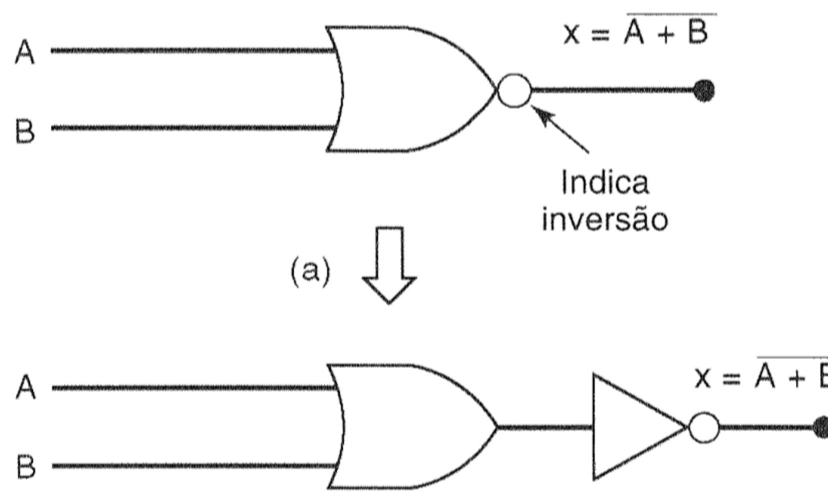
Exemplo 4 - Simplicação

- $\mathbf{S = ABC + A\overline{C} + A\overline{B}}$
- $S = A(BC + \overline{C} + \overline{B})$ → Lei distributiva (3)
- $S = A(BC + \overline{B} + \overline{C})$ → Lei comutativa (1)
- $S = A\left(BC + \overline{\overline{\overline{B} + \overline{C}}}\right)$ → Regra 12 ((A')' = A)
- $S = A(BC + \overline{BC})$ → De Morgan
- $S = A(1)$ → Regra 10 (A + A' = 1)
- $\mathbf{S = A}$ → Regra 7 (A • 1 = A)

Outras portas derivadas

- **Porta NOR**
  - Complemento da porta OR
  - Porta OR seguida de um inversor
  - Representação:
    - (A + B)’, $\overline{A + B}$

		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

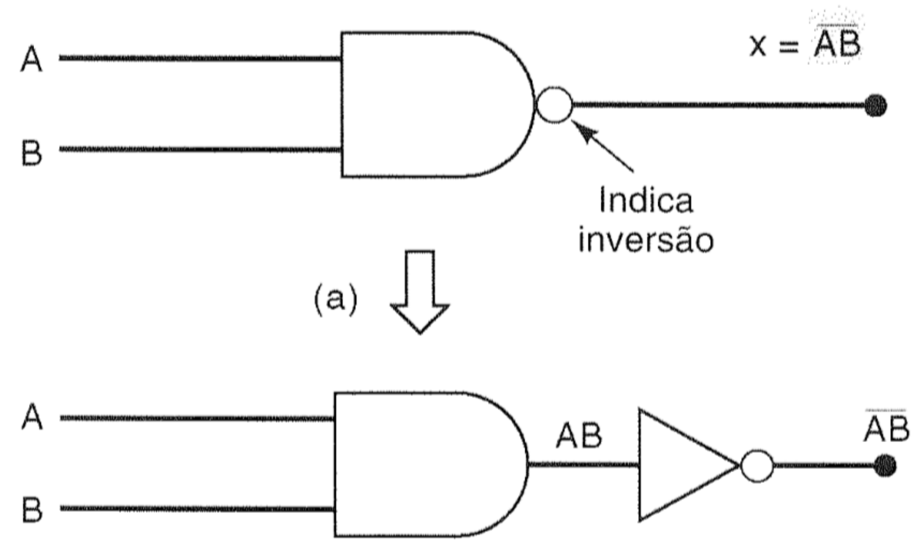


Outras portas derivadas

Porta NAND

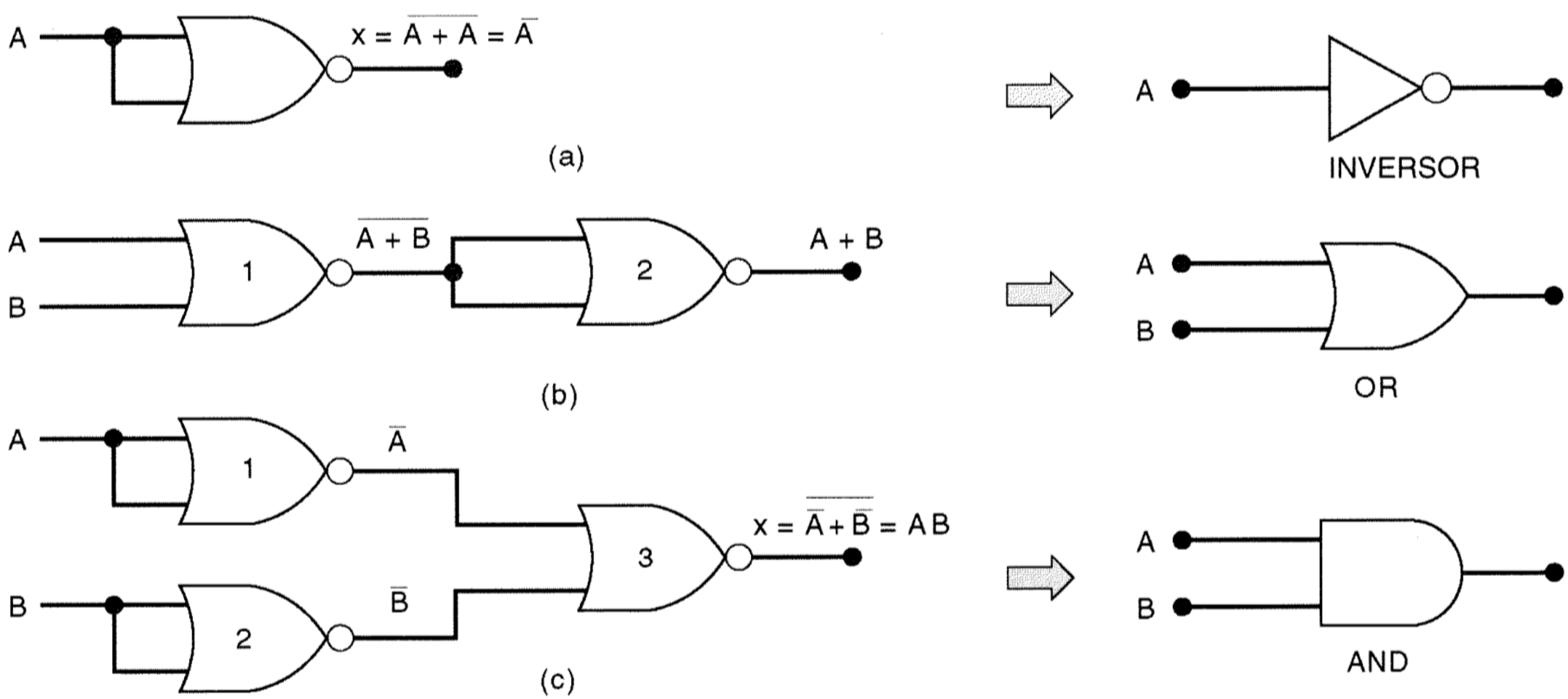
- Complemento da porta AND
- Porta AND seguida de um inversor
- Representação:
  - (AB)′, $(A \cdot B)'$, $\overline{\mathrm{AB}}$, $\overline{\mathrm{A \cdot B}}$

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



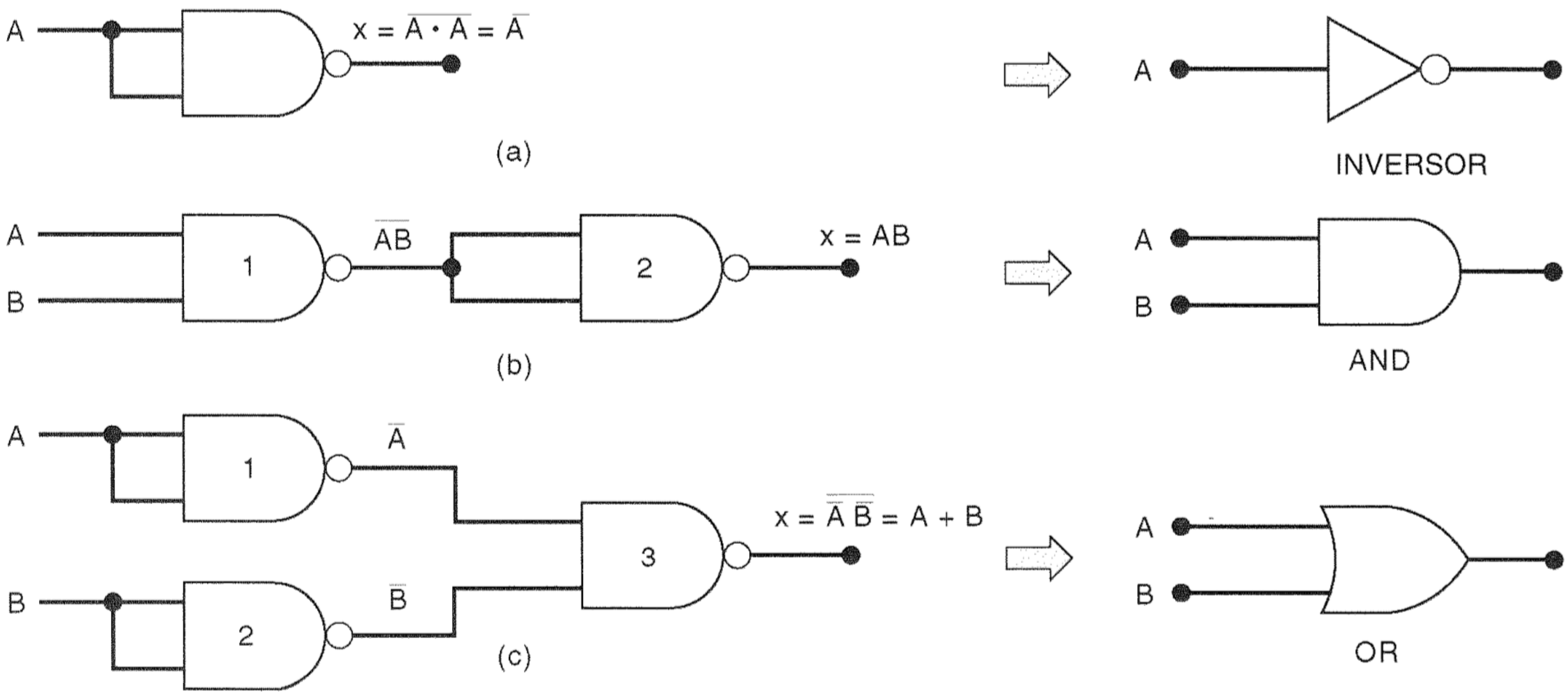
Outras portas derivadas

- Portas NOR e NAND → **portas universais**
  - podem ser usadas em combinação para implementar qualquer operação booleana



Outras portas derivadas

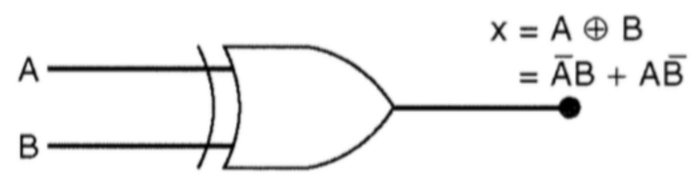
- Portas NOR e NAND → **portas universais**
  - podem ser usadas em combinação para implementar qualquer operação booleana



Outras portas derivadas

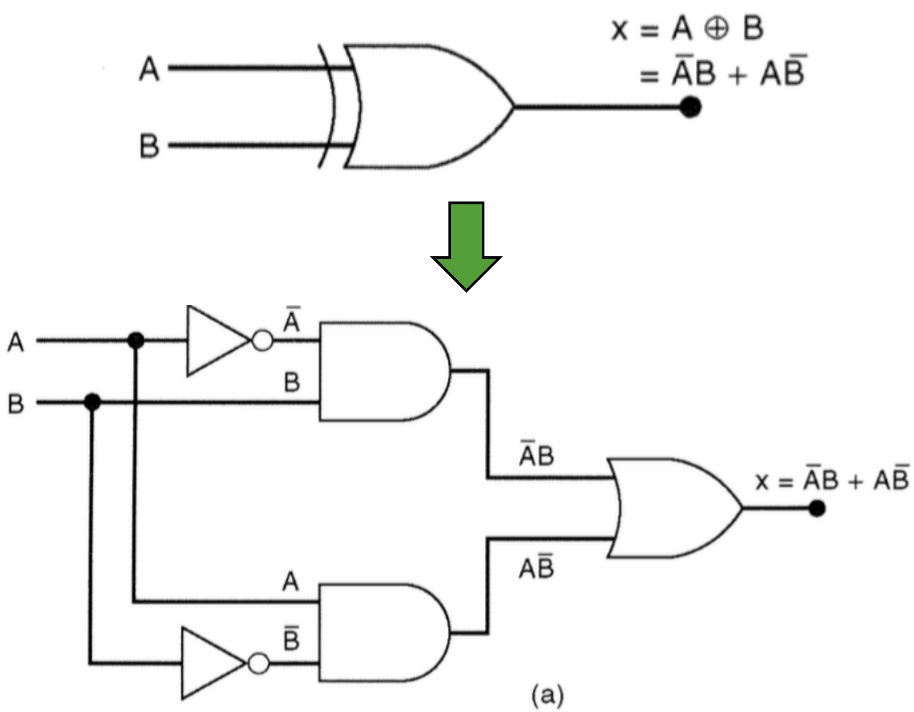
- **Porta XOR (OU exclusivo)**
  - A porta XOR de duas entradas **compara dois bits** e a saída será **1 se** e somente se eles forem **diferentes**
  - Uma porta XOR de **várias entradas** terá a saída igual a 1 se tiver um **número ímpar de 1's** nas entradas
  - Representação:
    - $A \oplus B$, A XOR B

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



Outras portas derivadas

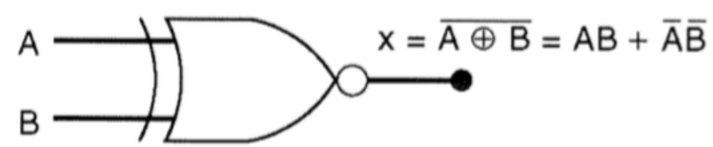
- Porta XOR (OU exclusivo)
  - $A \oplus B = \overline{A}B + A\overline{B}$



Outras portas derivadas

- Porta XNOR
  - A porta XNOR **compara dois bits** e a saída será **1 se** e somente se eles forem **iguais**
  - No caso de **várias entradas** a saída só será 1 se houver um **número par de 1's** nas entradas
  - Esta porta é também conhecida como porta comparadora
  - Representação:
    - $\overline{A \oplus B}$, A XNOR B

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



Outras portas derivadas

- Porta XNOR
  - $\overline{A \oplus B} = AB + \overline{A}\,\overline{B}$