1. Conceitos de Álgebra Booleana

2. Portas Lógicas e Inversores

Álgebra Booleana

- George Boole (1815-1864)

  1848: *The Calculus of Logic*

  Aplicação da matemática às operações mentais do raciocínio humano - definição da “álgebra booleana”

- Claude Shannon (1916-2001)

  ±1938: Tese de mestrado: *A Symbolic Analysis of Relay and Switching Circuits*

  Aplicação da álgebra booleana ao estudo e projeto de circuitos

Álgebra Booleana

- Conjunto de valores:

  {Falso, Verdadeiro} - raciocínio humano

  {Desligado, Ligado} - circuitos de chaveamento

  {0, 1} - sistema binário

  {0V, +5V} - eletrônica digital

- Conjunto de Operações:

  - complementação

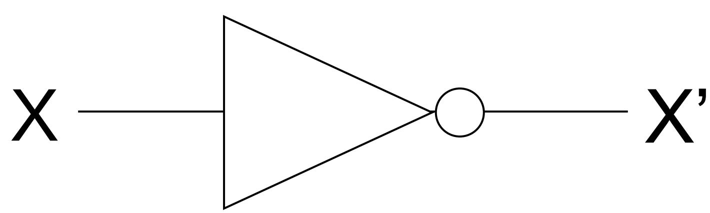
  - multiplicação lógica

  - adição lógica

Complementação (NOT)

X	X’
0	1
1	0

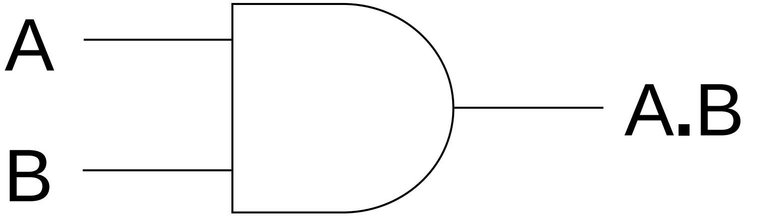
Componente: inversor ou porta NOT (inverter)



Multiplicação Lógica (E, AND)

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

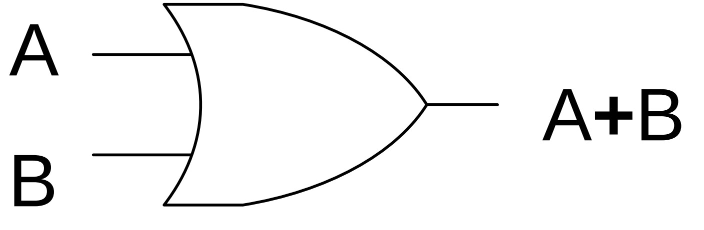
Componente: porta E (AND gate)



Adição Lógica (OU, OR)

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

Componente: porta OU (OR gate)



Precedência das Operações

1 - ( )

2 - NOT

3 - AND

4 - OR

Exemplos:

$A \,.\, B + C'$

$(A \,.\, B + C)'$

$A \,.\, (B + C)'$

$A \,.\, (B + C')$

Expressões Booleanas x Circuitos

A + B . C'

Exercício: desenhar o circuito

Construção da tabela-verdade - considerar a precedência !

A	B	C	C'	B.C'	A+B.C'
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Efeito da Precedência das Operações

1 - ( )

2 - NOT

3 - AND

4 - OR

Exemplos:

$A \cdot B + C'$

$(A \cdot B + C)'$

$A \cdot (B + C)'$

$A \cdot (B + C')$

A	B	C	C’	A.B	A.B+C’
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Efeito da Precedência das Operações

1 - ( )

2 - NOT

3 - AND

4 - OR

Exemplos:

A . B + C′

(A . B + C)′

A . (B + C)′

A . (B + C′)

A	B	C	A.B	A.B+C	(A.B+C)’
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Efeito da Precedência das Operações

1 - ( )

2 - NOT

3 - AND

4 - OR

Exemplos:

A . B + C ′

(A . B + C)′

$\boxed{\text{A . (B + C)′}}$

A . (B + C′)

A	B	C	B+C	(B+C)’	A.(B+C)’
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Efeito da Precedência das Operações

1 - ( )

2 - NOT

3 - AND

4 - OR

Exemplos:

$A \cdot B + C'$

$(A \cdot B + C)'$

$A \cdot (B + C)'$

$\boxed{A \cdot (B + C')}$

A	B	C	C’	B+C’	A.(B+C’)
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Efeito da Precedência das Operações

1 - ( )

2 - NOT

3 - AND

4 - OR

Comparando as saídas dos quatro circuitos:

Exemplos:

A . B + C ′

(A . B + C)′

A . (B + C)′

A . (B + C′)

A	B	C	A.B+C’	(A.B+C)’	A.(B+C)’	A.(B+C’)
0	0	0	1	1	0	0
0	0	1	0	0	0	0
0	1	0	1	1	0	0
0	1	1	0	0	0	0
1	0	0	1	1	1	1
1	0	1	0	0	0	0
1	1	0	1	0	0	1
1	1	1	1	0	0	1

Expressões Booleanas x Circuitos

A + B . (A' + B')

Exercício: desenhar o circuito

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

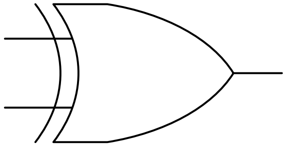
Conclusão: o mesmo resultado pode ser obtido apenas com A+B

Conceito importante: "minimizar" a expressão booleana

Portas mais complexas (1)

Porta XOR
(2 entradas)

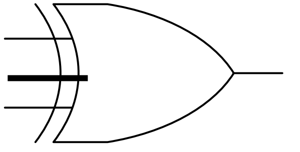
A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0



- ou exclusivo
- função “não iguais”

Porta XOR
(mais de 2 entradas)

A	B	C	(A ⊕ B ⊕ C)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

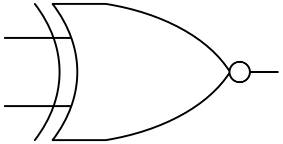


- função “ímpar”

Portas mais complexas (2)

Porta XNOR
(2 entradas)

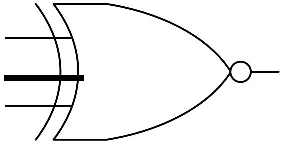
A	B	(A ⊕ B)’
0	0	1
0	1	0
1	0	0
1	1	1



- não ou exclusivo
- função “iguais”

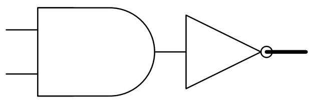
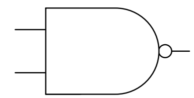
Porta XNOR
(mais de 2 entradas)

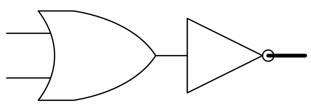
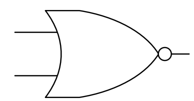
A	B	C	(A ⊕ B ⊕ C)’
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

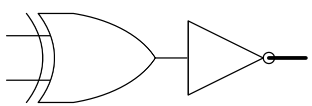
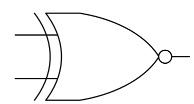


- função “par”

Portas mais complexas (3)

 é equivalente a  (NAND)

 é equivalente a  (NOR)

 é equivalente a  (XNOR)