Aula 05 - Lógica digital

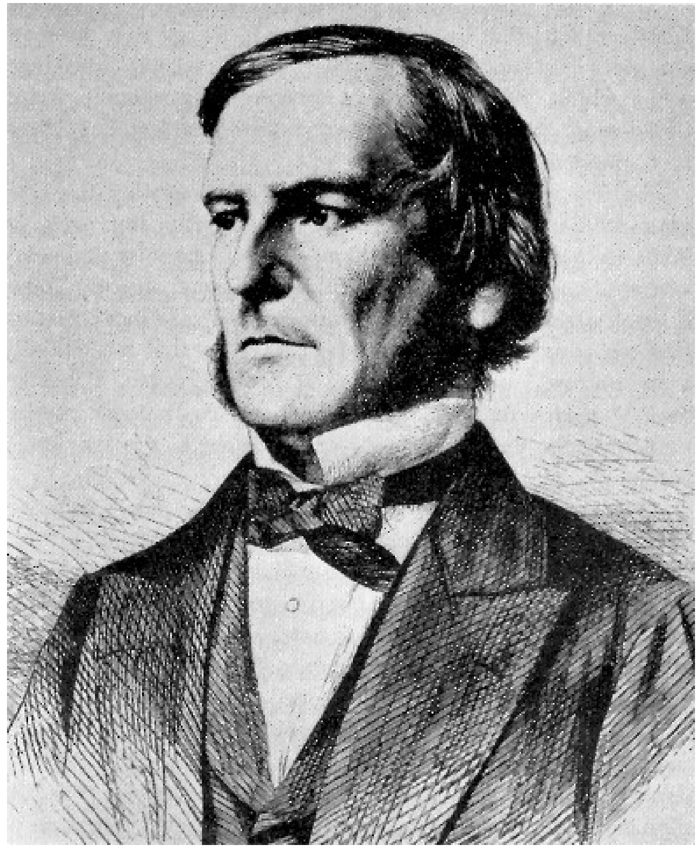
Sumário

- Lógica digital
- Portas lógicas
- Funções lógicas
- Exercícios

Lógica Digital

- **Álgebra booleana**
  - **George Boole (1864)**
    - Álgebra na qual há apenas dois valores válidos: falso ou verdadeiro.
  - **Claude Shannon (1938)**
    - Álgebra booleana para projetos de circuitos de comutação de relés
    - As técnicas sugeridas por Shannon foram subsequentemente utilizadas para projetos de circuitos eletrônicos digitais

Lógica Digital



George Boole (1815 - 1864). Matemático e filósofo inglês, “pai” da lógica digital moderna.

Lógica Digital

- Álgebra booleana
  - **Análise**: forma econômica de descrever um circuito digital

- Álgebra booleana
  - **Análise**: forma econômica de descrever um circuito digital
  - **Projeto**: dada uma função a ser implementada, a álgebra booleana pode ser usada para simplificar essa função

Lógica Digital

- Álgebra booleana
  - Variáveis
    - 1 (verdadeiro)
    - 0 (falso)

Lógica Digital

- Álgebra booleana
  - Variáveis
    - 1 (verdadeiro)
    - 0 (falso)
  - Operações básicas
    - AND (E)
    - OR (OU)
    - NOT (NÃO)

Lógica Digital

- Álgebra booleana
  - Variáveis
    - 1 (verdadeiro)
    - 0 (falso)
  - Operações básicas
    - AND (E)
    - OR (OU)
    - NOT (NÃO)

- Representação simbólica
  - A ***AND*** B = $A \cdot B$
  - A ***OR*** B = $A + B$
  - ***NOT*** A = $\bar{A}$, A'

- Assim como na álgebra comum, o resultado de uma operação booleana é obtido por meio de uma tabuada.

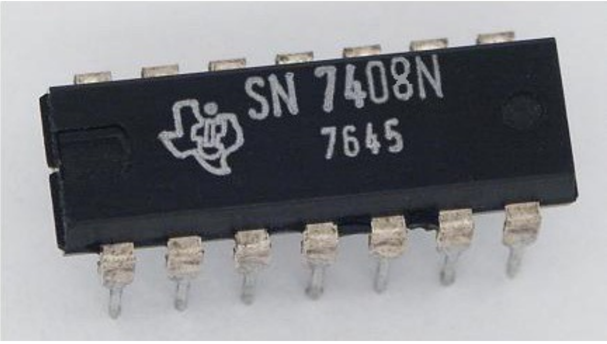
Lógica Digital

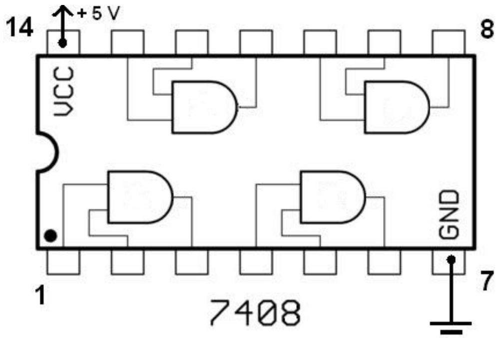
- Assim como na álgebra comum, o resultado de uma operação booleana é obtido por meio de uma tabuada.

- Na álgebra booleana, as tabuadas são chamada de **tabelas verdade**.

Mas antes... Portas Lógicas

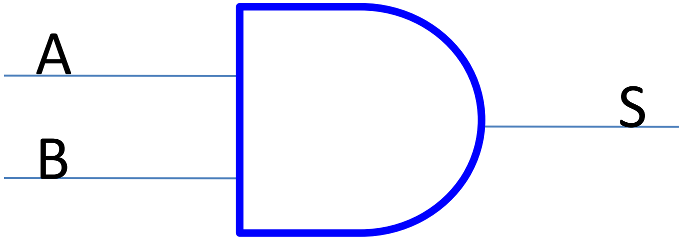
- Portas lógicas são:
  - Os blocos fundamentais dos circuitos lógicos digitais.
  - Circuitos eletrônicos que produzem um sinal de saída que é o resultado de uma ***operação booleana*** entre os sinais de entrada.





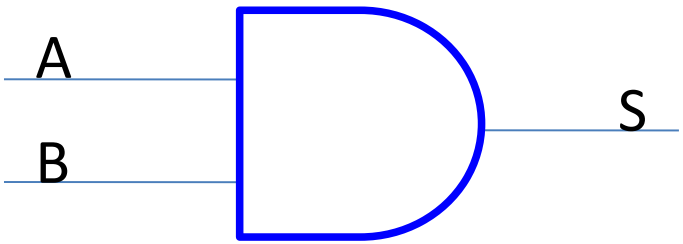
Lógica Digital

- Operação ***AND***
  - O resultado da operação é verdadeiro (valor binário 1) se e somente se todas as entradas forem verdadeiras (1)



S = A ***AND*** B = A · B

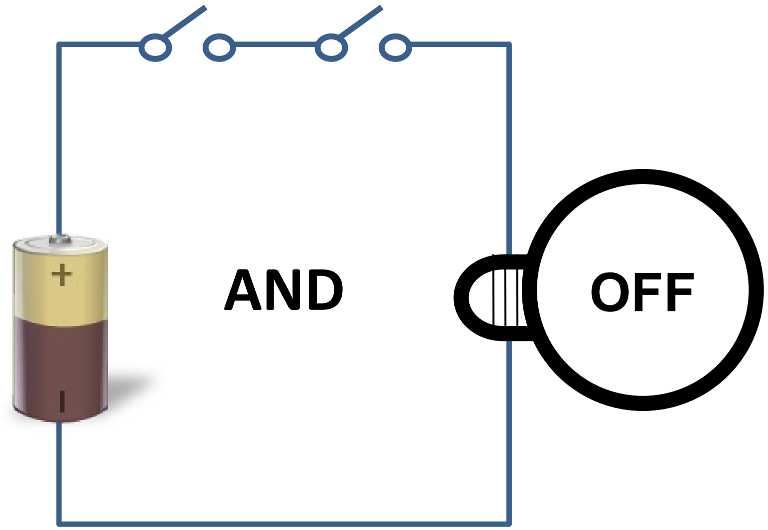
- Operação ***AND***
  - O resultado da operação é verdadeiro (valor binário 1) se e somente se todas as entradas forem verdadeiras (1)



A	B	S = A AND B
0	0	0
0	1	0
1	0	0
1	1	1

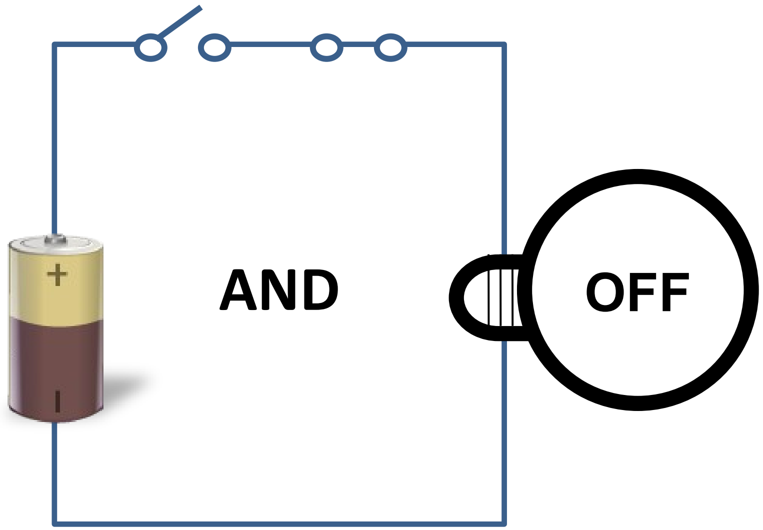
S = A ***AND*** B = A · B

AND – Analogia da Lampada



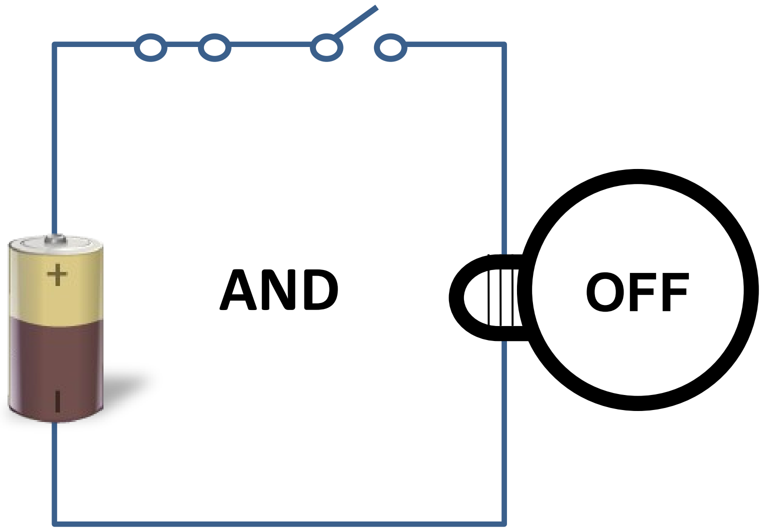
A	B	S = A AND B
0	0	0
0	1	0
1	0	0
1	1	1

AND – Analogia da Lampada



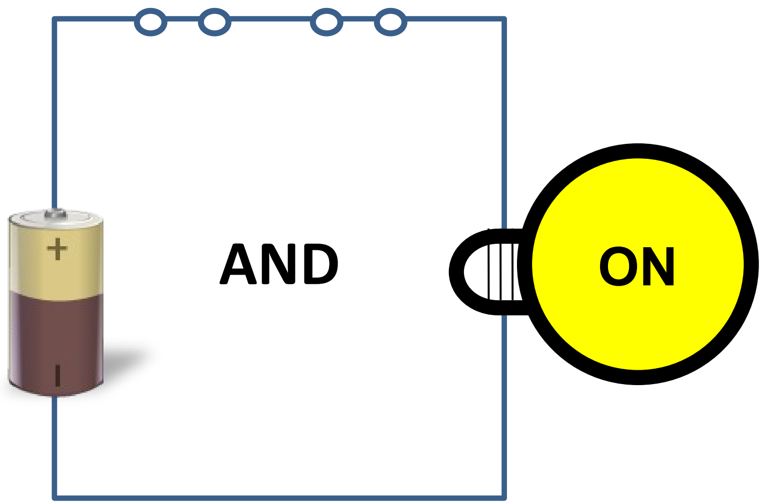
A	B	S = A AND B
0	0	0
0	1	0
1	0	0
1	1	1

AND – Analogia da Lampada



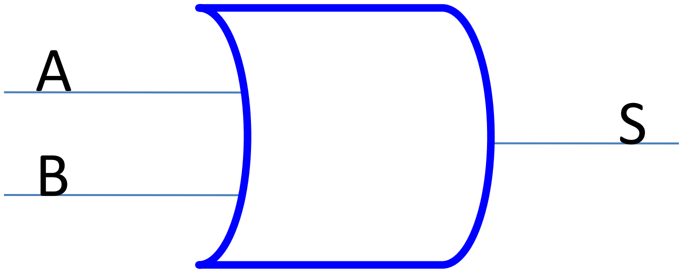
A	B	S = A AND B
0	0	0
0	1	0
1	0	0
1	1	1

AND – Analogia da Lampada



A	B	S = A AND B
0	0	0
0	1	0
1	0	0
1	1	1

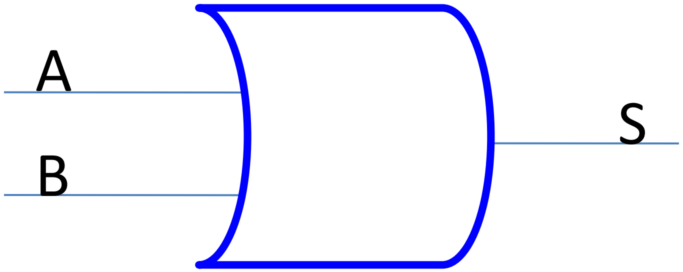
- Operação ***OR***
  - O resultado da operação é verdadeiro (valor binário 1) se qualquer uma das entradas, ou ambas, forem verdadeiras



S = A ***OR*** B = A + B

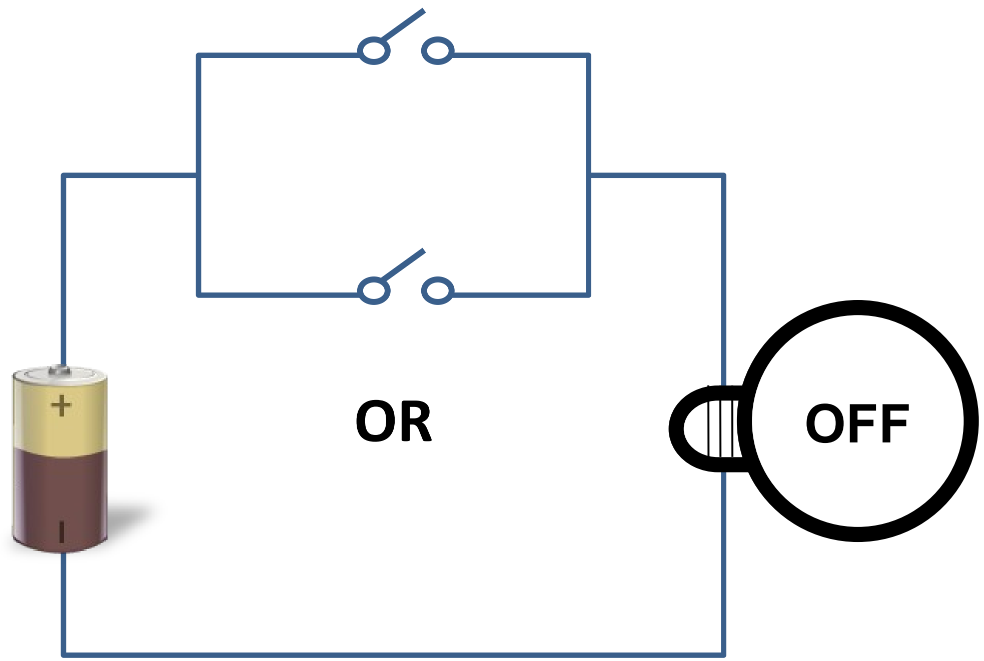
Lógica Digital

- Operação ***OR***
  - O resultado da operação é verdadeiro (valor binário 1) se qualquer uma das entradas, ou ambas, forem verdadeiras



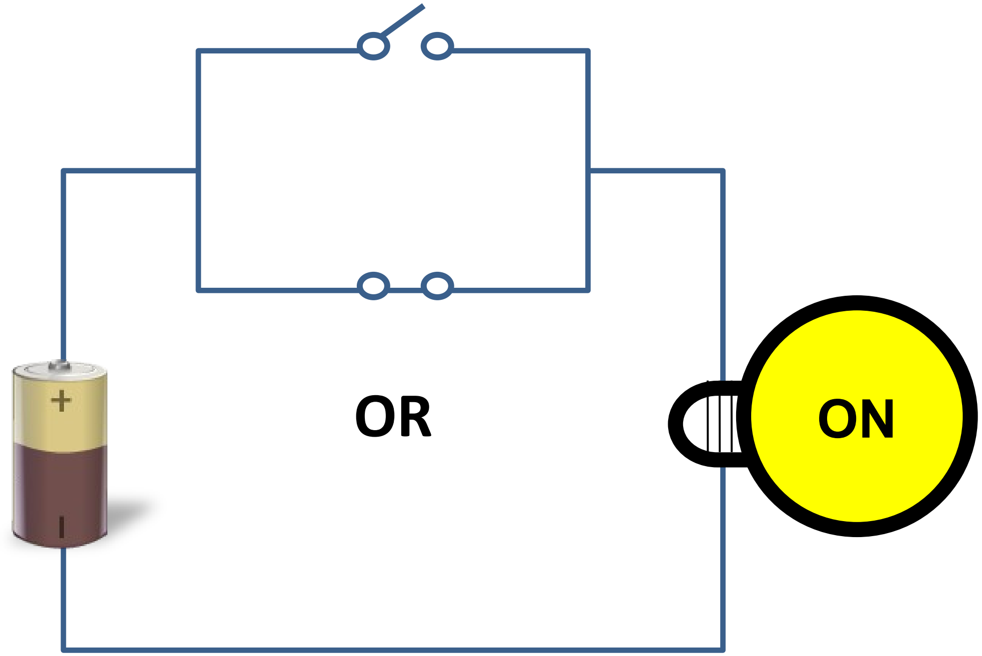
S = A ***OR*** B = A + B

OR – Analogia da Lampada



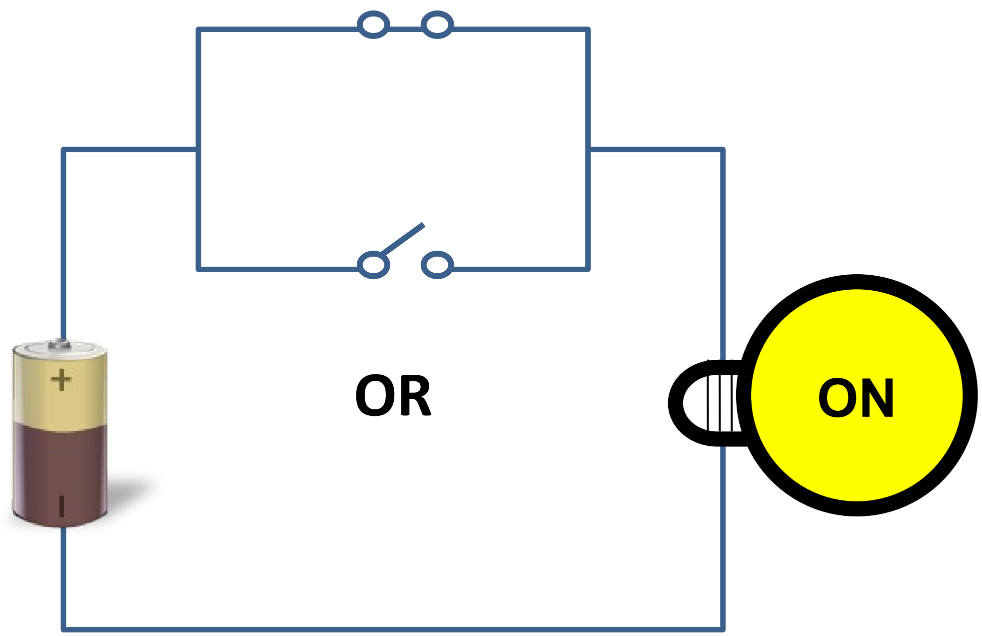
A	B	S = A OR B
0	0	0
0	1	1
1	0	1
1	1	1

OR – Analogia da Lampada



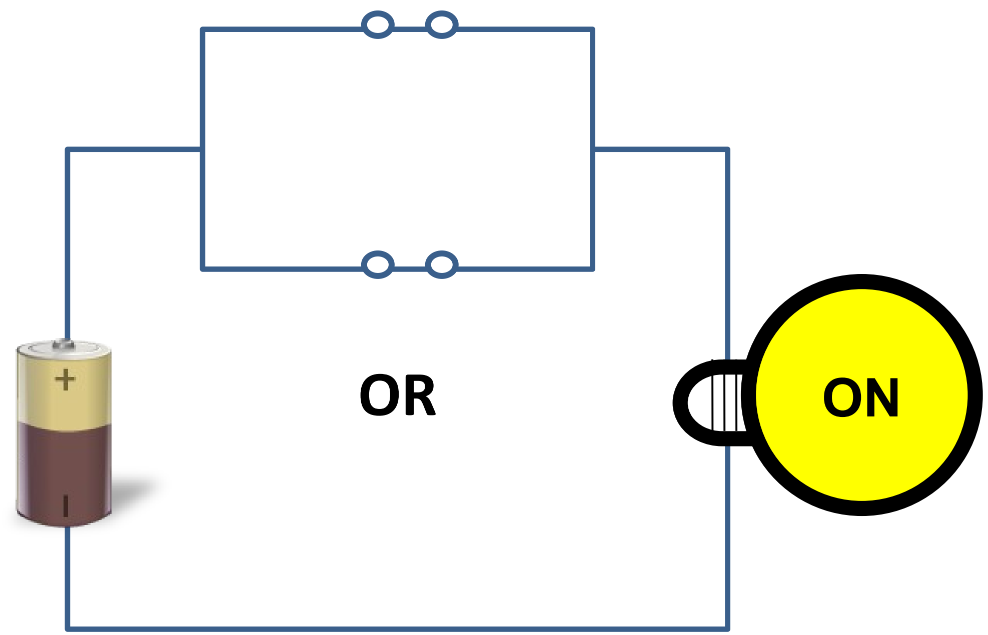
A	B	S = A OR B
0	0	0
0	1	1
1	0	1
1	1	1

OR – Analogia da Lampada



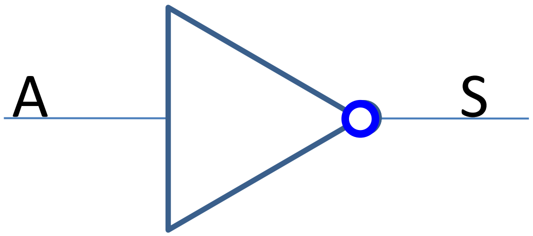
A	B	S = A OR B
0	0	0
0	1	1
1	0	1
1	1	1

OR – Analogia da Lampada



A	B	S = A OR B
0	0	0
0	1	1
1	0	1
1	1	1

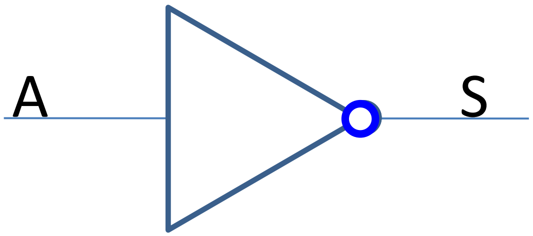
- Operação ***NOT***
  - Operação unária
  - Inverte o valor do entrada



$S = \textbf{\textit{NOT}}\ A = \bar{A}$

Lógica Digital

- Operação ***NOT***
  - Operação unária
  - Inverte o valor do entrada



A	S = NOT Ā
0	1
1	0

S = ***NOT*** A = Ā

Lógica Digital

- Como na álgebra comum, podemos combinar as operações, formando **expressões lógicas**.

- O resultado de uma expressão lógica pode ser calculado aplicando-se cada operação lógica, consultando-se as tabelas verdade correspondentes.

- Para indicar a ordem de aplicação das operações, usam-se parênteses como na álgebra comum.

Observações

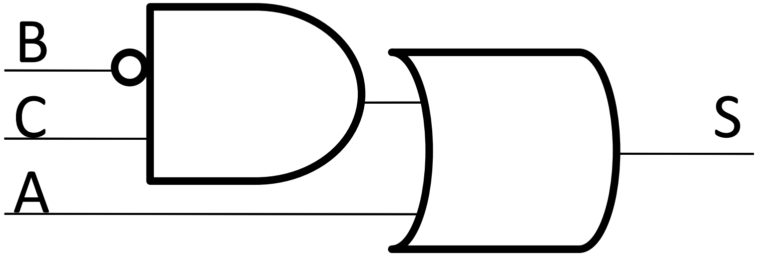
- A operação AND tem precedência sobre a operação OR

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

Observações

- A operação AND tem precedência sobre a operação OR

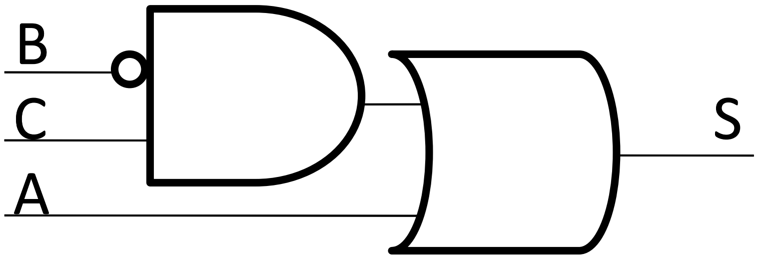
$S = A + \overline{B} \cdot C$



Observações

- A operação AND tem precedência sobre a operação OR

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

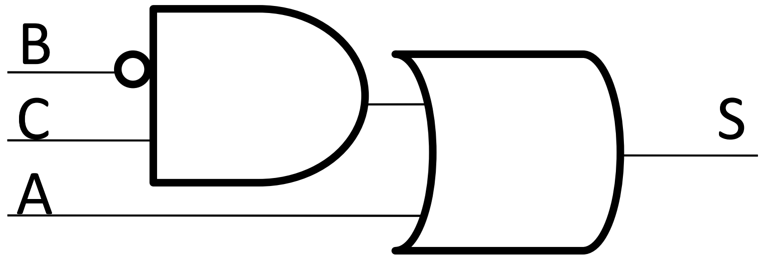


$S = (A + \overline{B}) \cdot C$

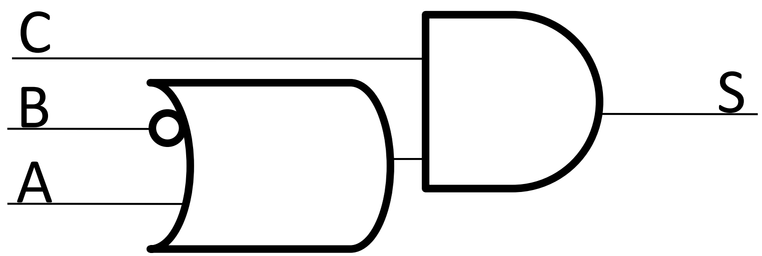
Observações

- A operação AND tem precedência sobre a operação OR

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



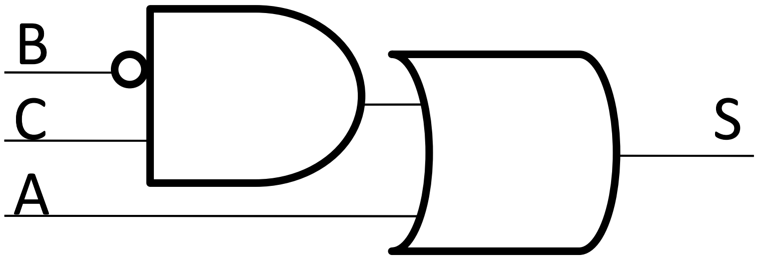
$S = (A + \overline{B}) \cdot C$



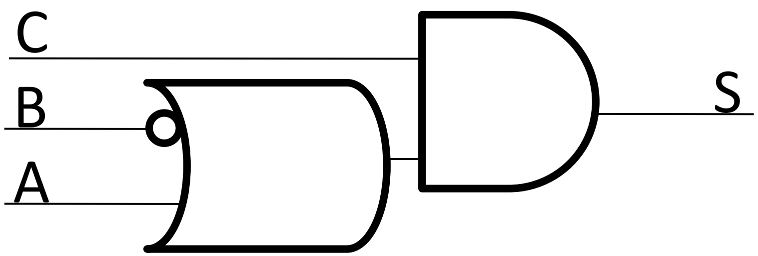
Observações

- A operação AND tem precedência sobre a operação OR

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



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



- A operação AND pode ser representada pela concatenação dos operandos: $A \cdot B = AB$

Identidades básicas da álgebra booleana

Postulados Básicos		
$A \cdot B = B \cdot A$	$A + B = B + A$	Leis da comutatividade
$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$	$A + (B \cdot C) = (A + B) \cdot (A + C)$	Leis da distributividade
$1 \cdot A = A$	$0 + A = A$	Elemento identidade
$A \cdot \bar{A} = 0$	$A + \bar{A} = 1$	Elemento inverso

Identidades básicas da álgebra booleana

Outras Identidades		
$0 \cdot A = 0$	$1 + A = 1$	
$A \cdot A = A$	$A + A = A$	
$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$	Leis de associatividade
$\overline{A \cdot B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \cdot \overline{B}$	Teorema de DeMorgan

NAND, NOR e XOR

- ***Outras operações lógicas importantes***
  - ***NAND*** *- Complemento (NOT) da função AND*
    - A NAND B = NOT(A AND B) = $\overline{AB}$

NAND, NOR e XOR

- ***Outras operações lógicas importantes***
  - ***NAND*** - *Complemento (NOT) da função AND*
    - A NAND B = NOT(A AND B) = $\overline{AB}$
  - ***NOR*** - *Complemento (NOT) da Função OR*
    - A NOR B = NOT (A OR B) = $\overline{A + B}$

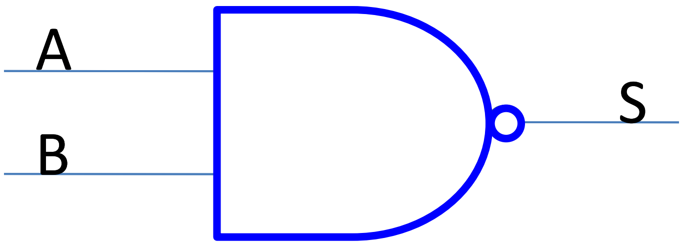
Prof. Clausius Duque Reis

NAND, NOR e XOR

- ***Outras operações lógicas importantes***
  - **_NAND_** - *Complemento (NOT) da função AND*
    - A NAND B = NOT(A AND B) = $\overline{AB}$
  - **_NOR_** - *Complemento (NOT) da Função OR*
    - A NOR B = NOT (A OR B) = $\overline{A + B}$
  - **_XOR_** – *Ou Exclusivo*
    - *A XOR B* = $A \oplus B$

Lógica Digital

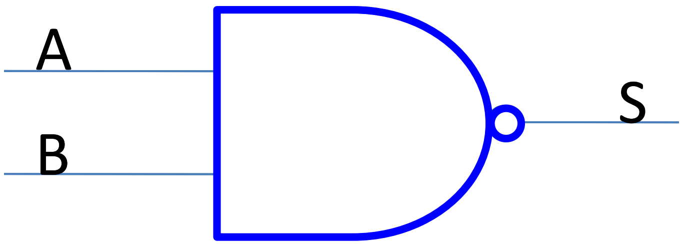
- Operação ***NAND***
  - O resultado da operação é o complemento (NOT) da função AND.



$$S = A\ \boldsymbol{\mathit{NAND}}\ B = \overline{A \cdot B}$$

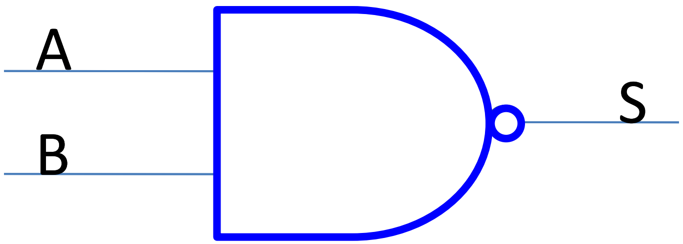
Lógica Digital

- Operação ***NAND***
  - O resultado da operação é o complemento (NOT) da função AND.
  - Ou seja, o resultado é falso (valor binário 0) se e somente se todas as entradas forem verdadeiras.



$$S = A\ \boldsymbol{NAND}\ B = \overline{A \cdot B}$$

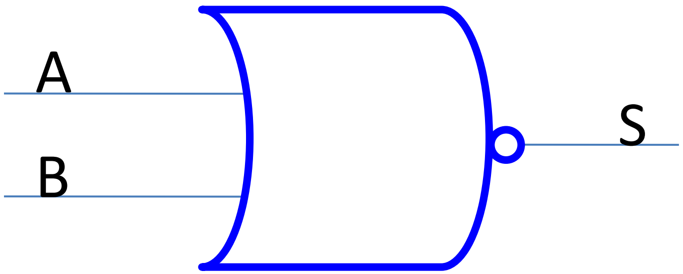
- Operação ***NAND***
  - O resultado da operação é o complemento (NOT) da função AND.
  - Ou seja, o resultado é falso (valor binário 0) se e somente se todas as entradas forem verdadeiras.



$S = A\ \textbf{\textit{NAND}}\ B = \overline{A \cdot B}$

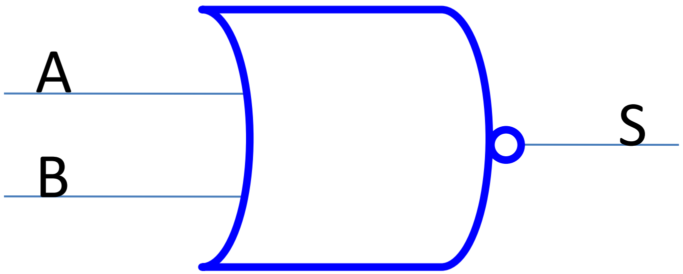
A	B	S = A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

- Operação ***NOR***
  - O resultado da operação é o complemento (NOT) da função OR.



$$S = A \text{ } \boldsymbol{\mathit{NOR}} \text{ } B = \overline{A + B}$$

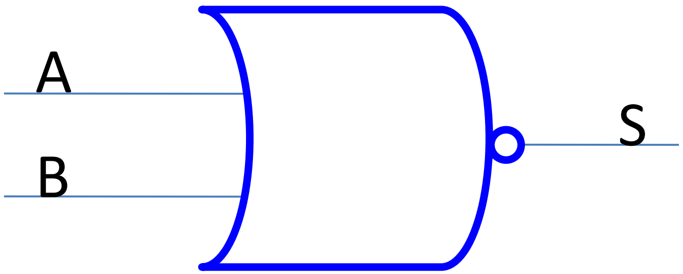
- Operação ***NOR***
  - O resultado da operação é o complemento (NOT) da função OR.
  - Ou seja, o resultado é falso (valor binário 0) se qualquer uma das entradas, ou ambas, forem verdadeiras.



$$S = A\ \textbf{\textit{NOR}}\ B = \overline{A + B}$$

Lógica Digital

- Operação ***NOR***
  - O resultado da operação é o complemento (NOT) da função OR.
  - Ou seja, o resultado é falso (valor binário 0) se qualquer uma das entradas, ou ambas, forem verdadeiras.

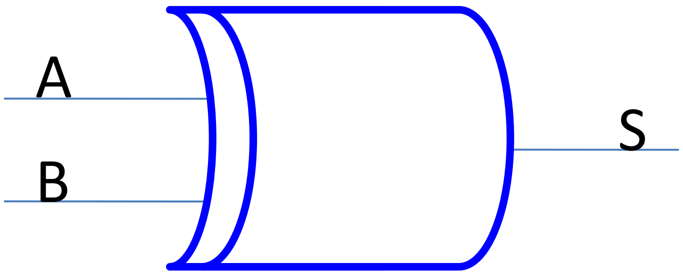


$$S = A \ \boldsymbol{NOR} \ B = \overline{A + B}$$

A	B	S = A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

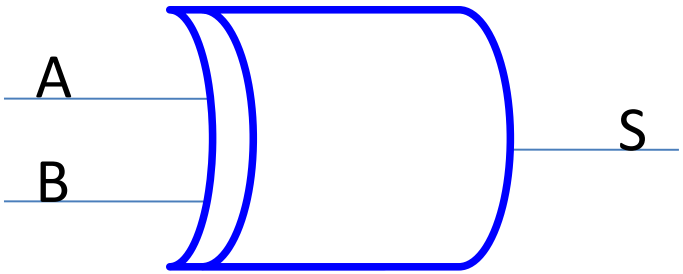
Lógica Digital

- Operação ***XOR*** (OU - Exclusivo)
  - O resultado da operação é verdadeiro (valor binário 1) se e somente se exatamente um dos operandos tem valor 1.



S = A ***XOR*** B = A $\oplus$ B

- Operação ***XOR*** (OU - Exclusivo)
  - O resultado da operação é verdadeiro (valor binário 1) se e somente se exatamente um dos operandos tem valor 1.



$S = A \; \textbf{\textit{XOR}} \; B = A \oplus B$

A	B	S = A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Tabela Verdade

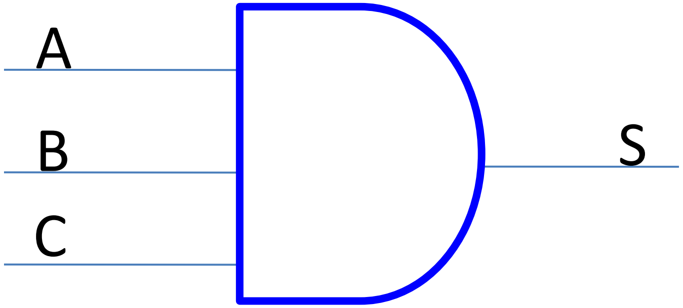
P	Q	P AND Q	P OR Q	NOT P	P NAND Q	P NOR Q	P XOR Q
0	0						
0	1						
1	0						
1	1						

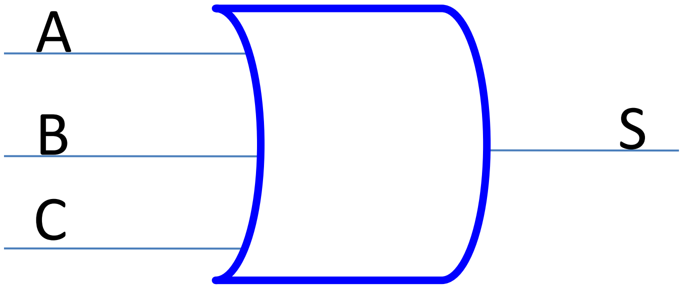
Tabela Verdade

P	Q	P AND Q	P OR Q	NOT P	P NAND Q	P NOR Q	P XOR Q
0	0	0	0	1	1	1	0
0	1	0	1	1	1	0	1
1	0	0	1	0	1	0	1
1	1	1	1	0	0	0	0

Portas Lógicas

- Portas lógicas podem ter mais de 2 entradas (2, 3, 4, ...).





Funções Lógicas

Função lógica: associação que nos “leva” de um conjunto de *n* variáveis booleanas, ao conjunto {0,1}.

$$F : \{0,1\}^n \rightarrow \{0,1\}$$

$$X_1, X_2, \ldots, X_n \rightarrow Y = F(X_1, X_2, \ldots, X_n)$$

Podemos descrever uma função lógica por meio de uma expressão ou pela sua tabela verdade.

Funções Lógicas

Ex.: Construa a tabela verdade da função

$$F(A,B,C) = A + \overline{B} \cdot C$$

Funções Lógicas

Ex.: Construa a tabela verdade da função

$$F(A,B,C) = A + \overline{B} \cdot C$$

A	B	C	$\overline{B} \cdot C$	$F(A,B,C) = A+\overline{B} \cdot C$
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Funções Lógicas

Ex.: Construa a tabela verdade da função

$F(A,B,C) = A + \overline{B} \cdot C$

A	B	C	$\overline{B} \cdot C$	$F(A,B,C) = A+\overline{B} \cdot C$
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Funções Lógicas

Ex.: Construa a tabela verdade da função

$$F(A,B,C) = A + \overline{B} \cdot C$$

A	B	C	$\overline{B} \cdot C$	$F(A,B,C) = A+\overline{B} \cdot C$
0	0	0	0	
0	0	1	1	
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	1	
1	1	0	0	
1	1	1	0	

Funções Lógicas

Ex.: Construa a tabela verdade da função

$$F(A,B,C) = A + \overline{B} \cdot C$$

A	B	C	$\overline{B} \cdot C$	$F(A,B,C) = A+\overline{B} \cdot C$
0	0	0	0	
0	0	1	1	
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	1	
1	1	0	0	
1	1	1	0	

Funções Lógicas

Ex.: Construa a tabela verdade da função

$F(A,B,C) = A + \overline{B} \cdot C$

A	B	C	$\overline{B} \cdot C$	$F(A,B,C) = A+\overline{B} \cdot C$
0	0	0	0	
0	0	1	1	
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	1	
1	1	0	0	
1	1	1	0	

Funções Lógicas

Ex.: Construa a tabela verdade da função

$F(A,B,C) = A + \overline{B} \cdot C$

A	B	C	$\overline{B} \cdot C$	$F(A,B,C) = A+\overline{B} \cdot C$
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	0	1
1	0	1	1	1
1	1	0	0	1
1	1	1	0	1

Exercícios

1) Considere as seguintes funções booleanas:

- (a) $F(A,B,C) = A + (B' \cdot C)$
- (b) $F(A,B,C) = (B + C)' \cdot A \cdot C' + A' \cdot B'$
- (c) $F(A,B,C) = A' + A' + B \cdot B' \cdot C' + C$

2) Calcule F para:

- $A = 1, B = 1, C = 0$
- $A = 1, B = 0, C = 0$
- $A = 0, B = 1, C = 1$
- $A = 0, B = 0, C = 1$

- Construa as tabelas verdade para as funções a, b e c do exercício 1.

Exercícios

- Desenhe o circuito lógico digital equivalente a função: **$D = A + (B \cdot C')$**
- Construa a tabela verdade para as seguintes funções booleanas:
  - A) $f(A,B,C) = ABC + A'B'C'$
  - B) $f(A,B,C) = A(BC' + B'C)$
- Construa os circuitos lógicos dessas funções boleanas utilizando a representação gráfica.