Sistemas Computacionais

Parte 03 – Álgebra de Boole e Portas Lógicas

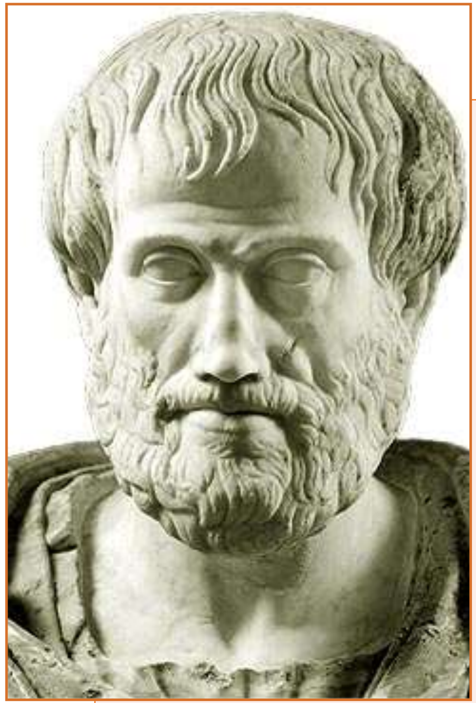
Prof. Fancisco Javier

Católica
Universidade Católica de Brasília

Álgebra de Boole

Origem

Silogismo

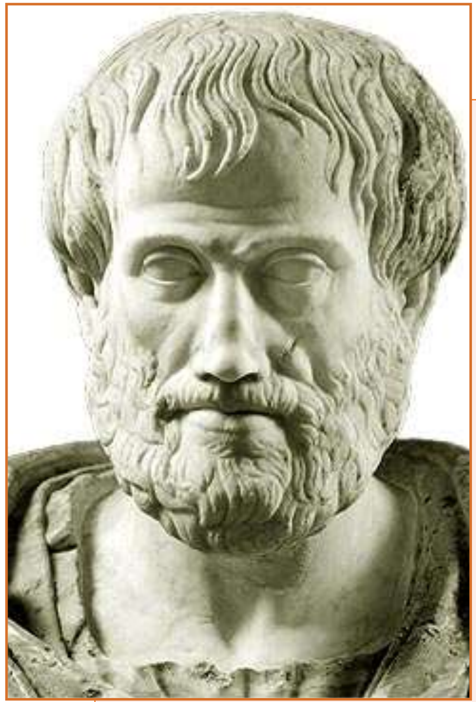


Aristóteles (384 A.C – 332 A.C.): filósofo grego da antiguidade.

- Discípulo de Platão
- Mestre de Alexandre, o Grande

- Criou a base do pensamento europeu.

Silogismo

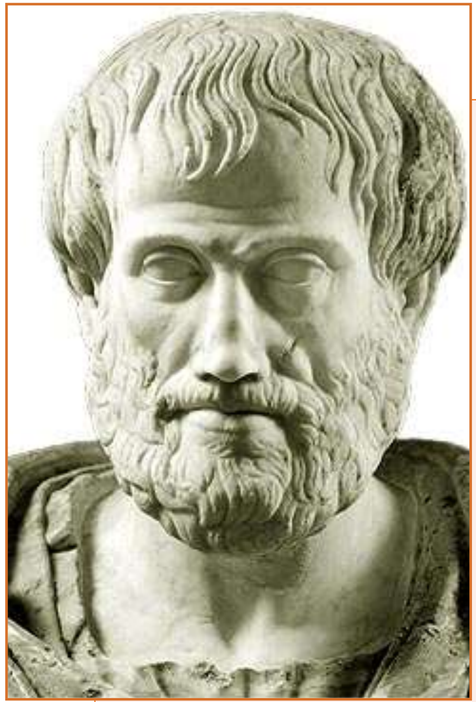


Da obra *Analíticos Anteriores*, designou a conclusão deduzida de premissas, sendo considerada a **argumentação lógica perfeita**.

- É um argumento dedutivo constituído de três proposições declarativas (**duas premissas e uma conclusão**) que se conectam de tal modo que, a partir de duas premissas, é possível deduzir uma conclusão.
- Num silogismo, as **premissas são juízos** que antecedem a conclusão e dos quais ela decorre.

Universidade Católica de Brasília

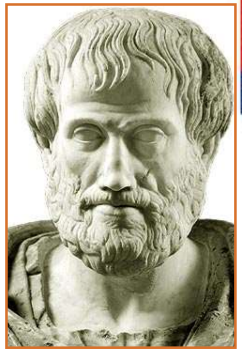
Silogismo



- Dos juízos prévios (premissas), infere-se a consequência (conclusão).
- Silogismo regular – argumento típico dedutivo, composto de três proposições declarativas: premissa maior (P), premissa menor (p) e conclusão (C) e de três termos – menor (t), maior (T) e médio (M) – compostos dois a dois.

Universidade Católica de Brasília

Silogismo



termo médio termo maior

"Todos os homens são mortais; — Premissa maior

termo menor termo médio

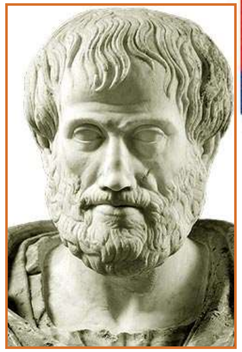
os gregos são homens; — Premissa menor

logo, os gregos são mortais." — Conclusão

- A premissa maior e a premissa menor são identificadas de acordo com a extensão dos seus termos.
- O conjunto de todos os homens é mais extenso do que o conjunto de todos os gregos.
- Logo, a premissa maior é "Todos os homens são mortais".
- Nas premissas, o termo maior (predicado da conclusão) e o termo menor (sujeito da conclusão) são comparados com o termo médio (comum às duas premissas).

Universidade Católica de Brasília

Silogismo



termo m

“Todos os ho

termo menor

os gregos sã

logo, os gregos são m

aior e a premissa
ificadas de acordo
seus termos.

os homens é
conjunto

or é “Todos
rtais“.

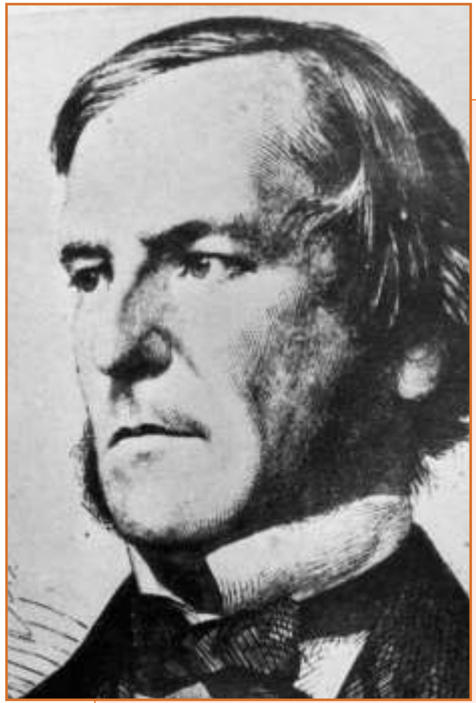
as, o termo maior
o da conclusão) e o
termo menor (sujeito da
conclusão) são comparados com o
termo médio (comum às duas
premissas).

Para que esse papo todo?????

Universidade Católica de Brasília

George Boole (1815-1864)

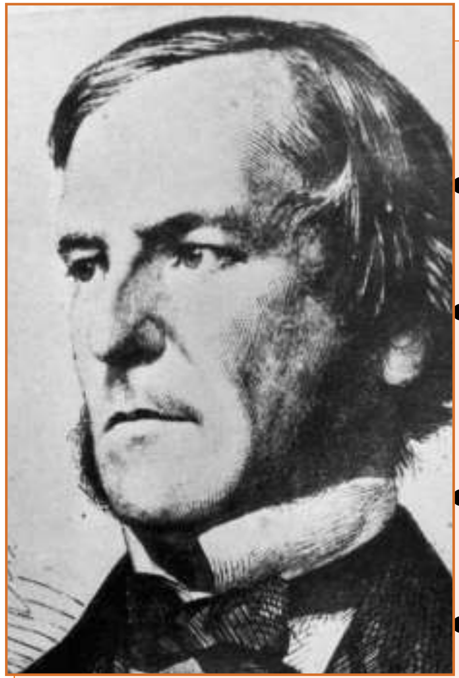
“O matemático é o poeta das ciências exatas”



Aos 12 anos traduziu as obras de Horácio para o inglês

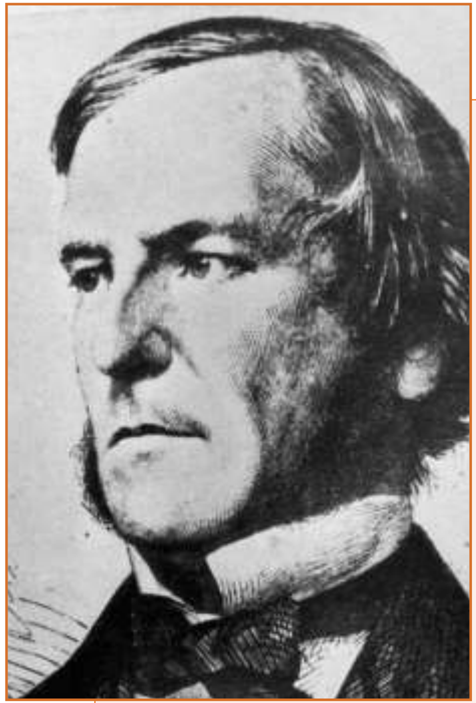
- Matemático e filósofo britânico
- Estudos de grego e latim
- “Uma Investigação das Leis do Pensamento” (1854)
- Formulação de uma **linguagem simbólica do pensamento**
- Resolução de uma equação não leva a uma resposta numérica, mas a uma conclusão lógica

George Boole (1815-1864)



- Uso de símbolos algébricos, como x, y, z, para denotar palavras, frases ou proposições
- As **variáveis** podem assumir **apenas valores de caráter lógico** (0 ou 1, F ou V, S ou N) e os operadores retornam também apenas valores de caráter lógico (0 ou 1, F ou V, S ou N)
- Com a utilização de símbolos, proposições podem ser reduzidas à forma de equações
- Uma conclusão silogística para duas premissas pode ser obtida por meio de regras algébricas comuns, que permitem alcançar a solução da equação

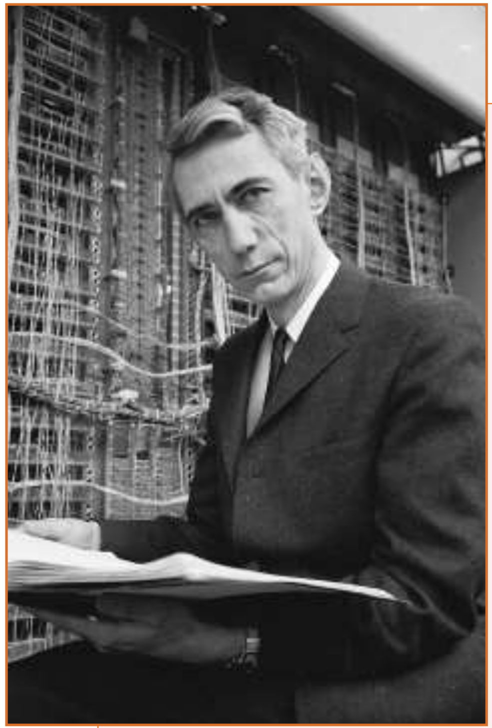
George Boole (1815-1864)



Exemplo:

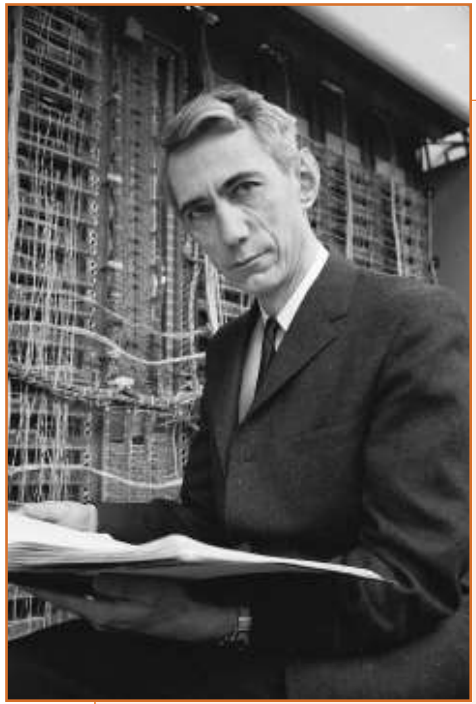
- X = jovem
- Y = faz o curso de Ciência da Computação
- (1 – X) = tudo que NÃO é jovem
- XY = tudo que é jovem E que faz o curso de Ciência da Computação
- X+Y = tudo que é jovem OU que faz o curso de Ciência da Computação
- (1-X)(1-Y) = tudo que NÃO é jovem E que NÃO faz o curso de Ciência da Computação

Claude Shannon (1916-2001)



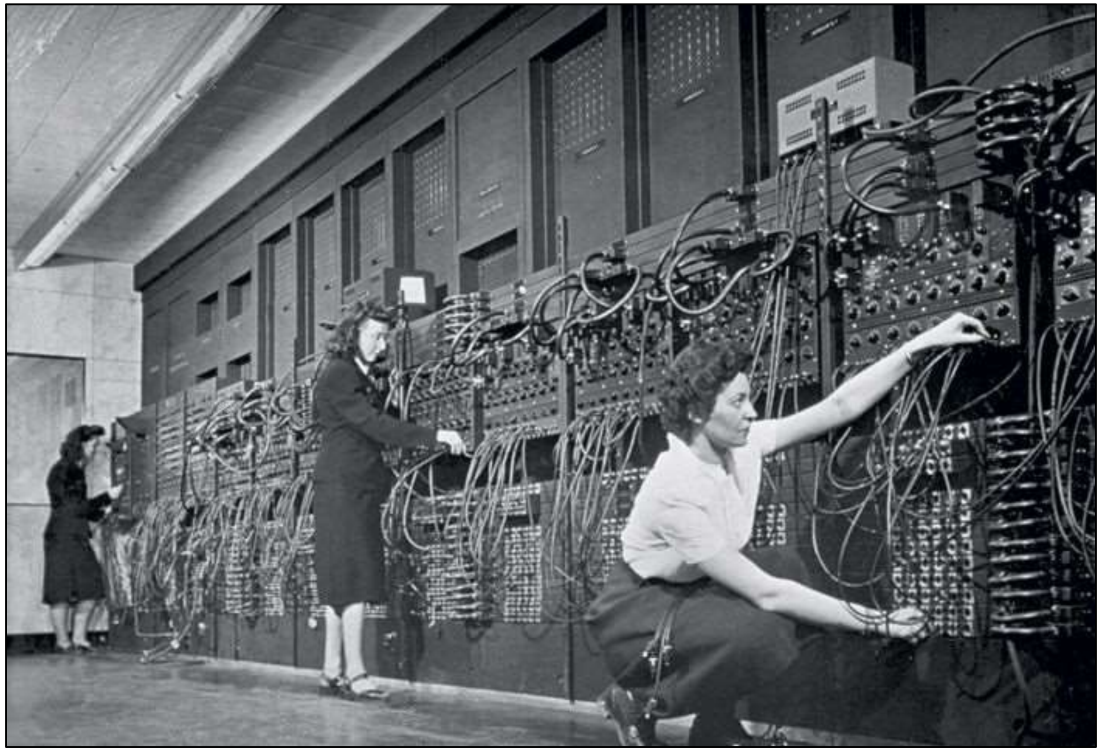
- Matemático e engenheiro eletricista estadunidense.
- Pai da Teoria da Informação (1948).
- Em 1938, em sua tese de mestrado, "Uma Análise Simbólica de Relés e Circuitos de Chaveamento", no *Massachusetts Institute of Tecnology* (MIT), aplicou a álgebra de Boole para mostrar que as propriedades de circuitos elétricos de chaveamento podem ser representadas por uma álgebra booleana com dois valores.

Claude Shannon (1916-2001)



- Simplificou o arranjo de relés eletromecânicos utilizados em comutadores para roteamento telefônico.
- Demonstrou que uma aplicação elétrica utilizando álgebra booleana e aritmética binária poderia resolver qualquer problema de lógica.

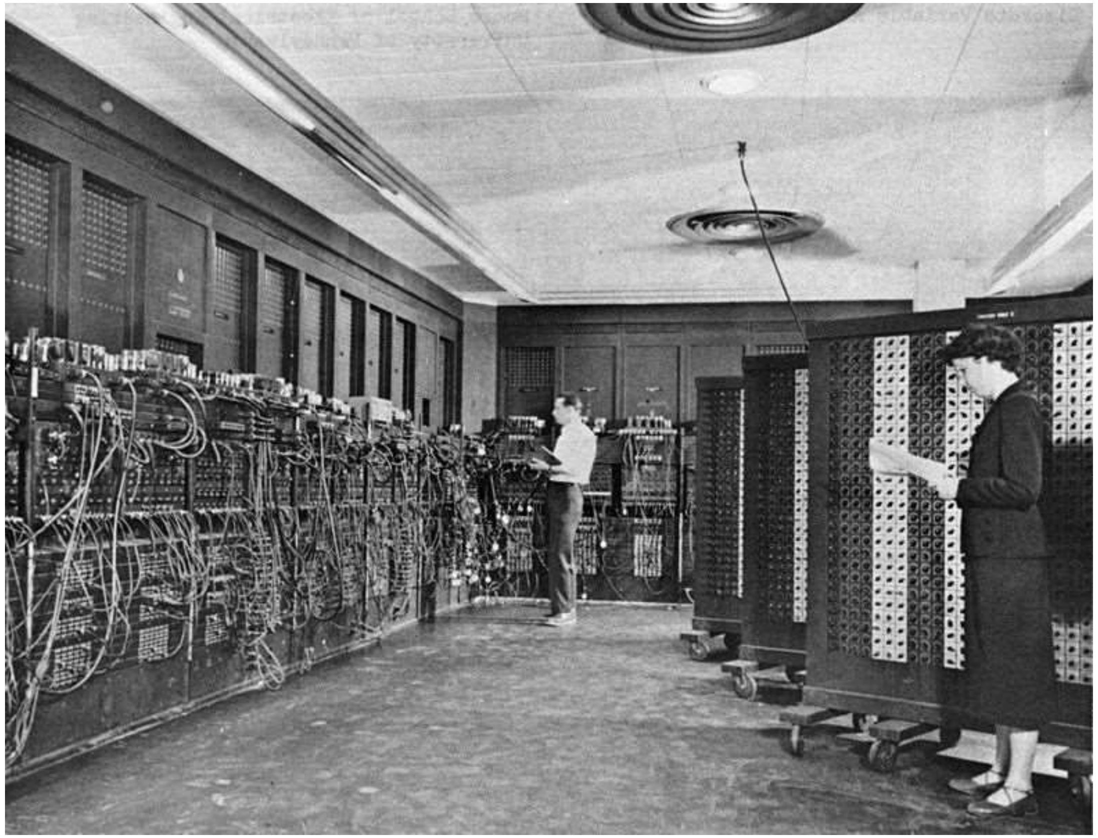
ENIAC - *Electronic Numerical Integrator and Calculator*



- Desenvolvido entre 1943 e 1946, a pedido do exército dos Estados Unidos para estudos de balísticas, pela Universidade da Pensilvânia.
- Continha 18 mil válvulas e 70 mil resistores que consumiam 200 KWh de energia, ocupava 180 m^2 e pesava 30 toneladas

ENIAC - *Electronic Numerical Integrator and Calculator*

- Utilizava o **sistema decimal** e sua memória consistia de 20 acumuladores, cada um capaz de manter um número decimal de 10 dígitos.
- Um anel de 10 válvulas representava cada dígito. A qualquer momento, somente uma válvula do anel estava no estado ligado, representando um dos 10 dígitos.



Católica
Universidade Católica de Brasília

John von Neumann (1903-1957)

A “Arquitetura de Neumann” continua sendo empregada na maioria dos computadores atuais

- Matemático americano, nascido na Hungria.
- Realizou trabalhos em mecânica quântica, teoria dos conjuntos, teoria dos jogos.
- Seus estudos foram **base para a construção do IAS**, o primeiro computador binário do mundo, na Universidade de Princeton, empregando uma arquitetura onde os **dados são processados na memória principal do computador**, tornando-o muito mais rápido.

Católica
Universidade Católica de Brasília

Álgebra de Boole

Constantes e variáveis booleanas

Tabela-verdade

Operações lógicas básicas

Variáveis e constantes booleanas

- Na álgebra convencional as variáveis são descritas por *x* e *y*, enquanto na álgebra booleana são *A* e *B*.
- A álgebra booleana considera apenas dois valores (0 e 1).

Lógico 0	Lógico 1
Falso	Verdadeiro
Desligado	Ligado
Baixo	Alto
Não	Sim
Aberto	Fechado

Tabela-verdade

- **É uma técnica para documentar os resultados possíveis de uma função booleana.**
- A relação entre a quantidade de entradas (e) e de combinações possíveis (c) é:

$$c = 2^e$$

Então:

- se temos duas entradas, teremos 4 combinações possíveis;
- se temos 3 entradas, teremos 8 combinações possíveis;
- E assim por diante.

No caso a comparação entre A e B leva a saídas tais que se são diferentes é verdadeiro (1).

Saída

Entradas

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

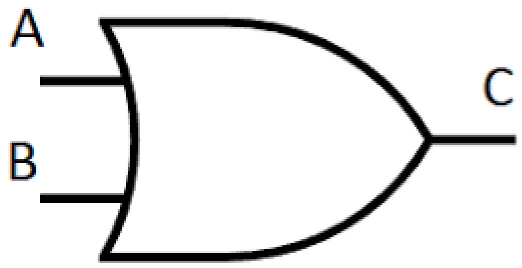
Operações lógicas básicas

Operação OR (“OU”)

Porta OR

- C = A + B

OR		
A	**B**	**C**
0	0	0
0	1	1
1	0	1
1	1	1



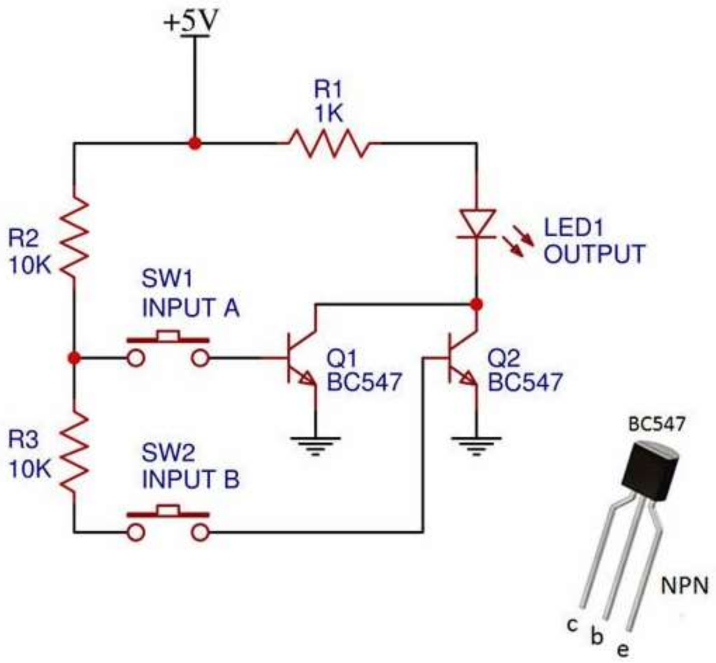
(padrão 91-1984 da ANSI/IEEE)

- “Se A ou B é verdadeiro, então C é verdadeiro”

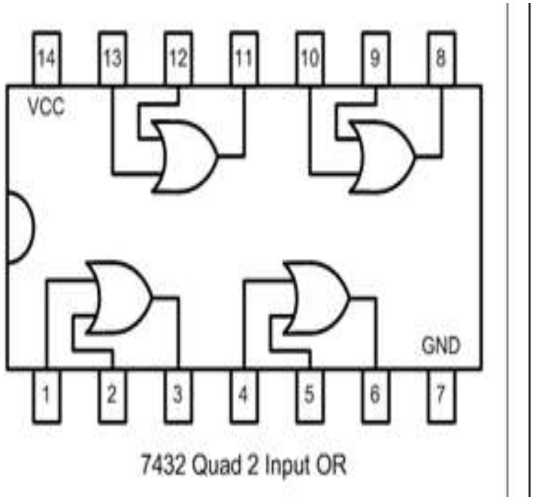
Universidade Católica de Brasília

Operações lógicas básicas

Implementando a porta OR

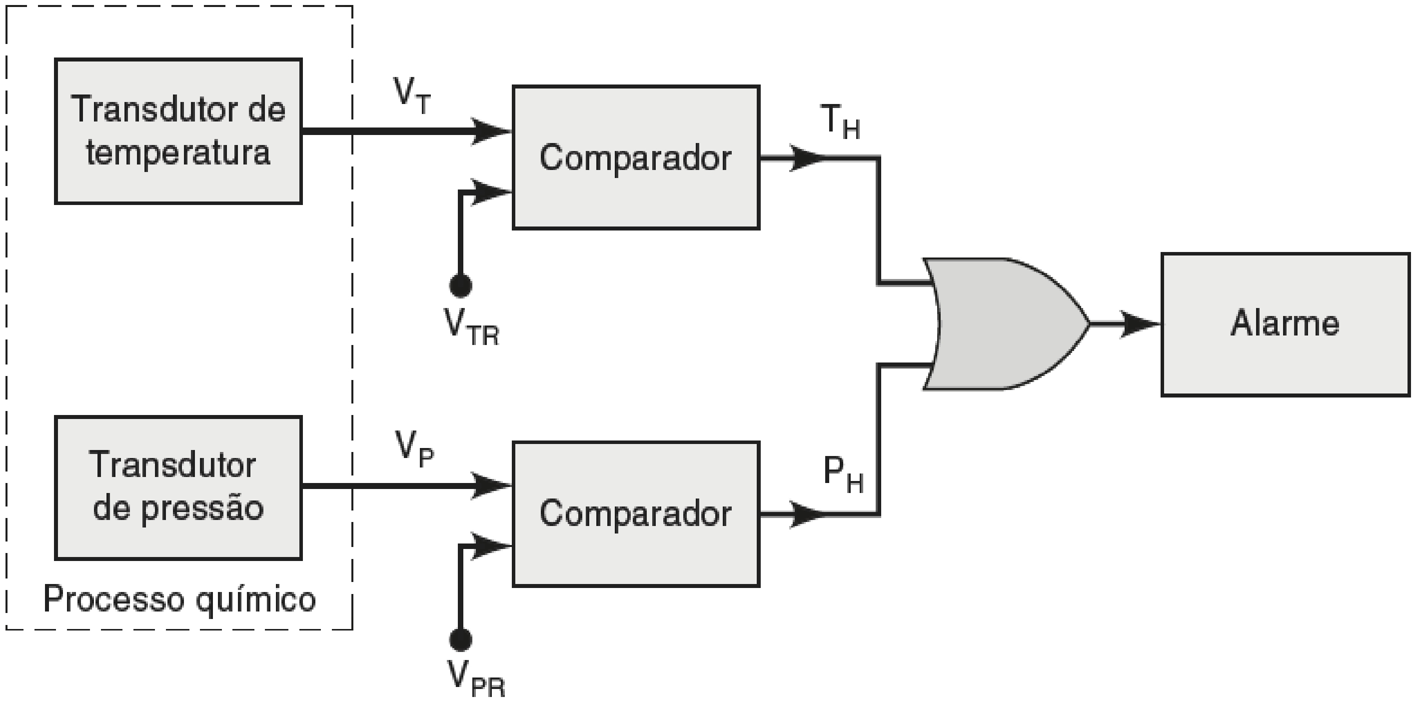


Construindo portas OR



Operações lógicas básicas

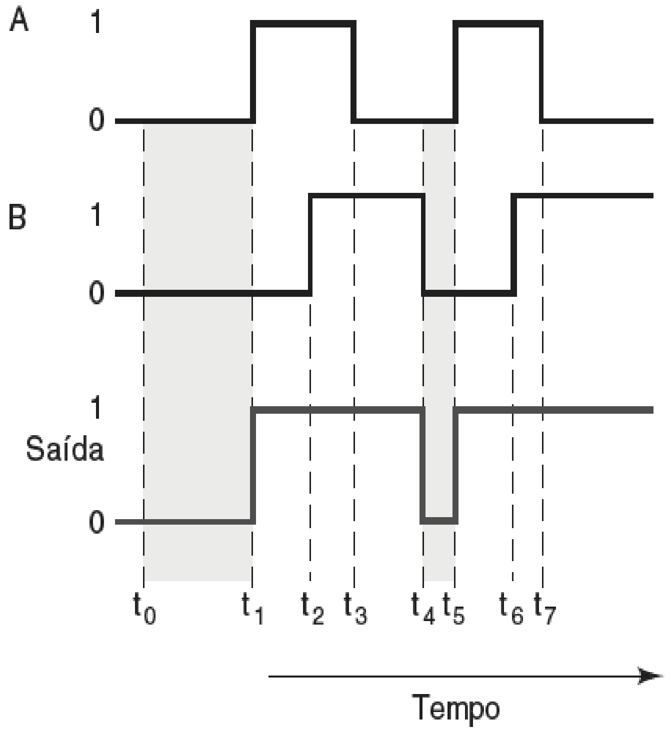
- Exemplo 1:

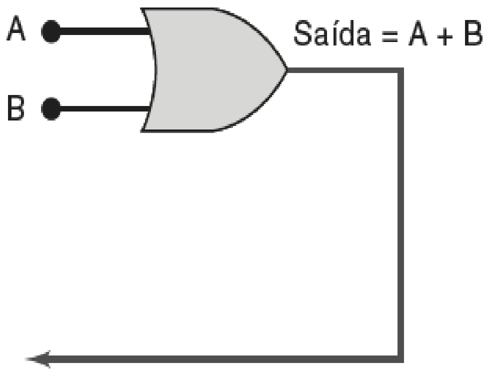


H –High -> alta; T = Temperatura; P = Pressão; V = Tensão; TR = Temperatura de Referência; PR = Pressão de Referência

Operações lógicas básicas

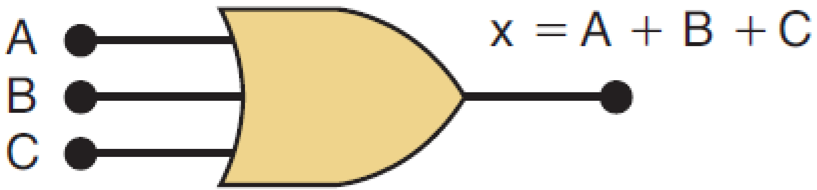
- Exemplo 2:





• Operação OR(OU) com porta OR e 3 entradas

- Em circuitos digitais, uma **porta OR** é um circuito que tem duas ou mais entradas e cuja saída é igual à combinação OR das entradas.
- Símbolo e tabela-verdade para uma 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

Note que a operação **OU** só pode ser definida se houver, pelo menos, duas variáveis envolvidas. Ou seja, não é possível realizar a operação sobre somente uma variável. Devido a isso, o operador "+" (**OU**) é dito **binário**.

Operações lógicas básicas

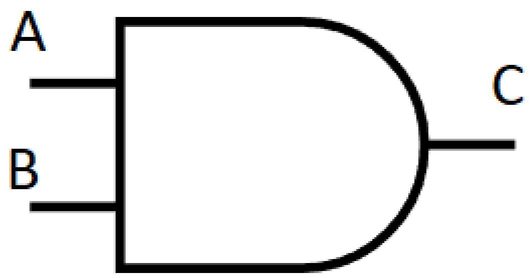
Operação AND (“E”)

- C = A . B

AND		
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

- “Se A e B são verdadeiros, então C é verdadeiro”

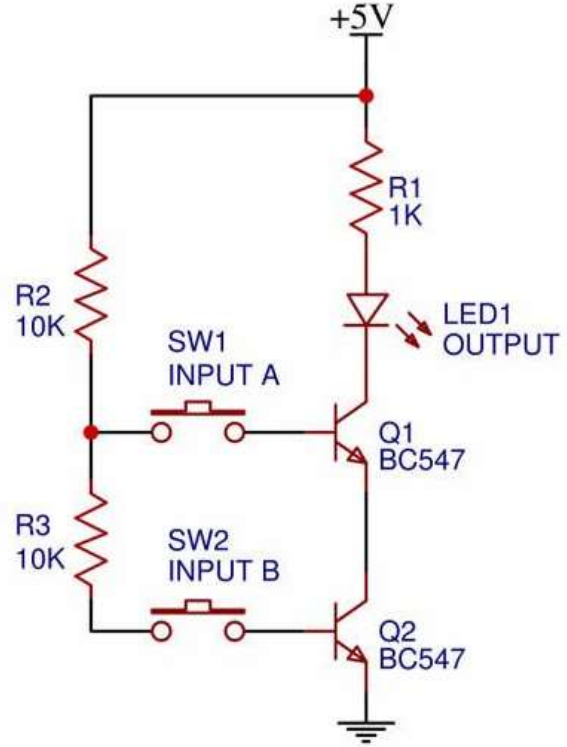
Porta AND

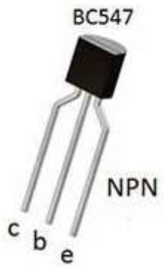


(padrão 91-1984 da ANSI/IEEE)

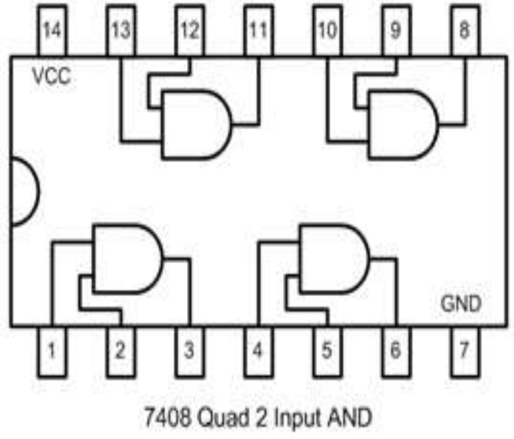
Operações lógicas básicas

Implementando a porta AND



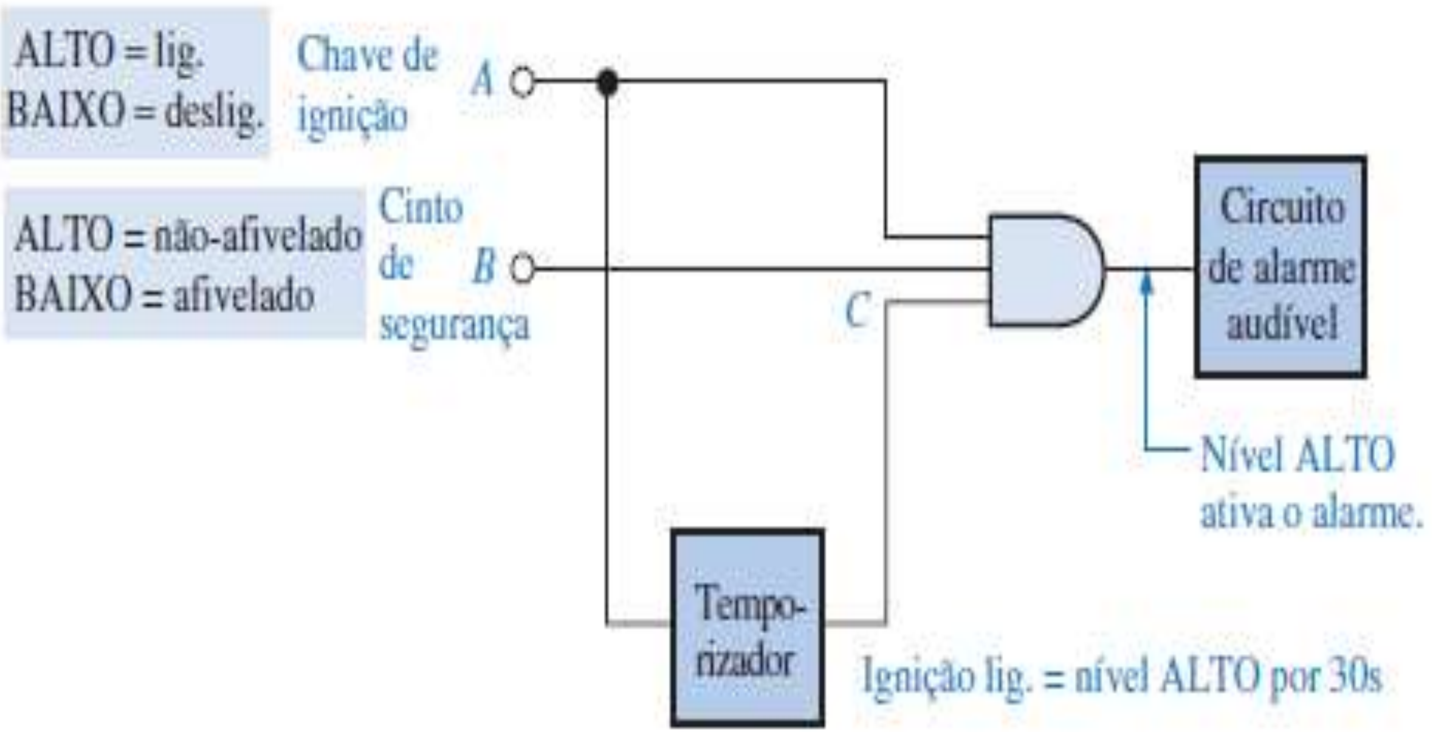


Construindo portas AND



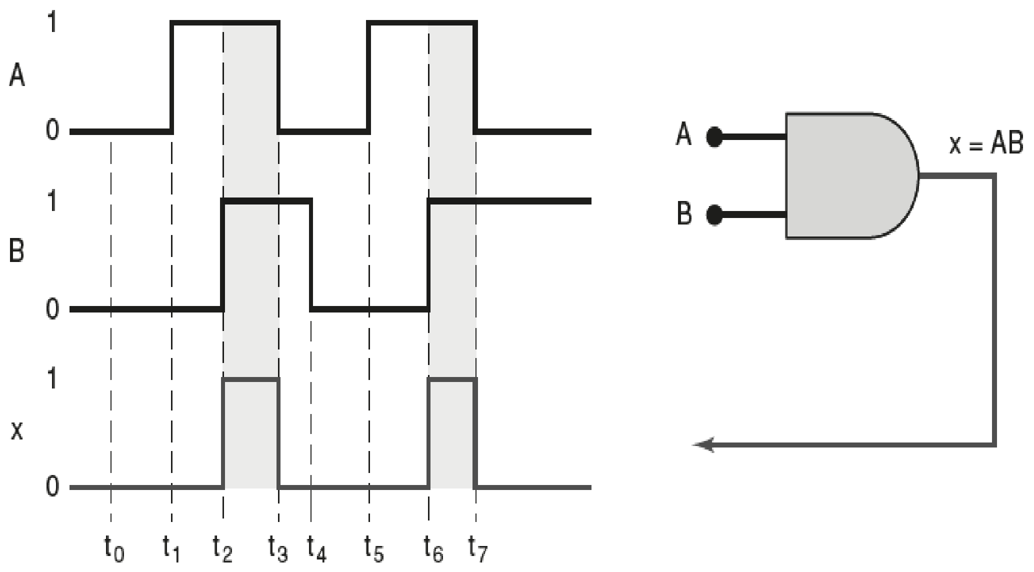
Operações lógicas básicas

- Exemplo 1: Detector de cinto afivelado ou não



Operações lógicas básicas

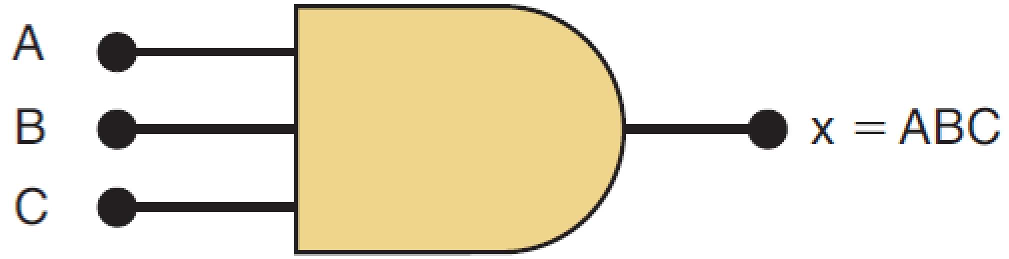
- Exemplo 2:



• Operação AND (E) com porta AND e 3 entradas

- Tabela-verdade e símbolo para uma 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



- **Operação AND (E) com porta AND**

▪ Multiplicação Lógica:

A operação **E**, ou **multiplicação** lógica, pode ser definida da seguinte forma:

"A operação **E** resulta **0** se pelo menos uma das variáveis de entrada vale **0**".

Pela definição dada, pode-se deduzir que o resultado da operação **E** será **1** se, e somente se, todas as entradas valerem **1**.

O símbolo usualmente utilizado na operação **E** é "·", porém outra notação possível é "∧". Podemos, também, listar as possibilidades de combinações entre dois valores Booleanos e os respectivos resultados, para a operação **E**:

$$
\begin{aligned}
0 \cdot 0 &= 0 \\
0 \cdot 1 &= 0 \\
1 \cdot 0 &= 0 \\
1 \cdot 1 &= 1
\end{aligned}
$$

- **Operação AND (E) com porta AND**

- Multiplicação Lógica:

Também para a operação **E** valem as propriedades **associativa** e **comutativa**. Então, a equação A·BC pode ainda ser avaliada tomando-se as variáveis aos pares, em qualquer ordem. Veja a tabela verdade a seguir e compare os resultados.

A B C	A·B·C	A·B	(A·B)·C	B·C	A·(B·C)
0 0 0	0	0	0	0	0
0 0 1	0	0	0	0	0
0 1 0	0	0	0	0	0
0 1 1	0	0	0	1	0
1 0 0	0	0	0	0	0
1 0 1	0	0	0	0	1
1 1 0	0	1	0	0	0
1 1 1	1	1	1	1	1

Operações lógicas básicas

O círculo diante do símbolo sempre denota uma INVERSÃO

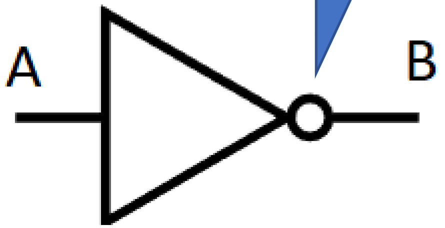
Operação NOT ("Não" ou "Inversão")

Porta NOT

- A = B

$$A = \bar{A}$$

NOT	
A	B
0	1
1	0

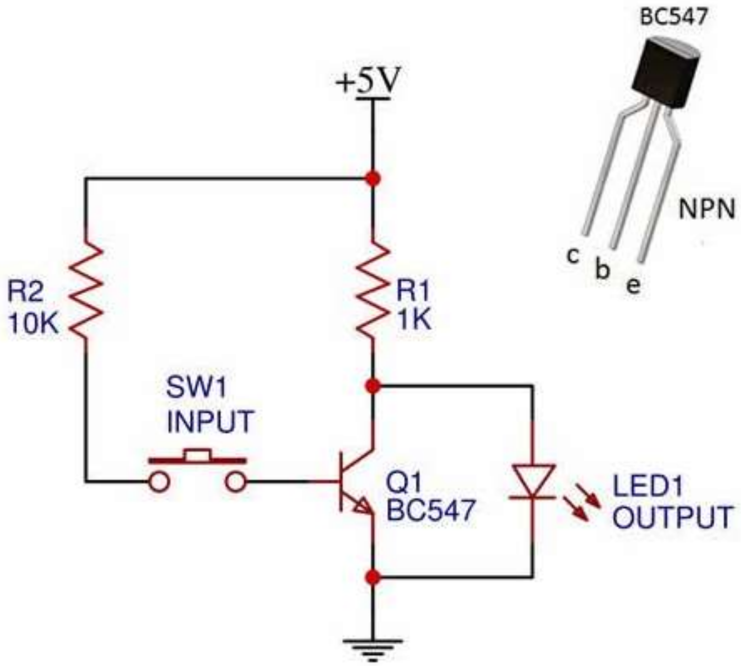


- Se A é verdadeiro, então B é falso.

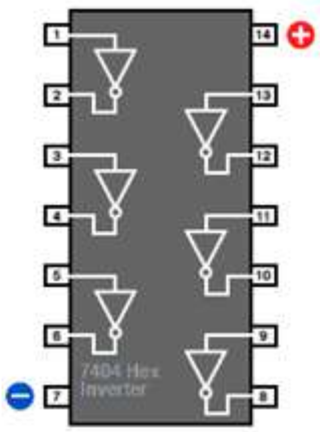
(padrão 91-1984 da ANSI/IEEE)

Operações lógicas básicas

Implementando a porta NOT



Construindo portas NOT



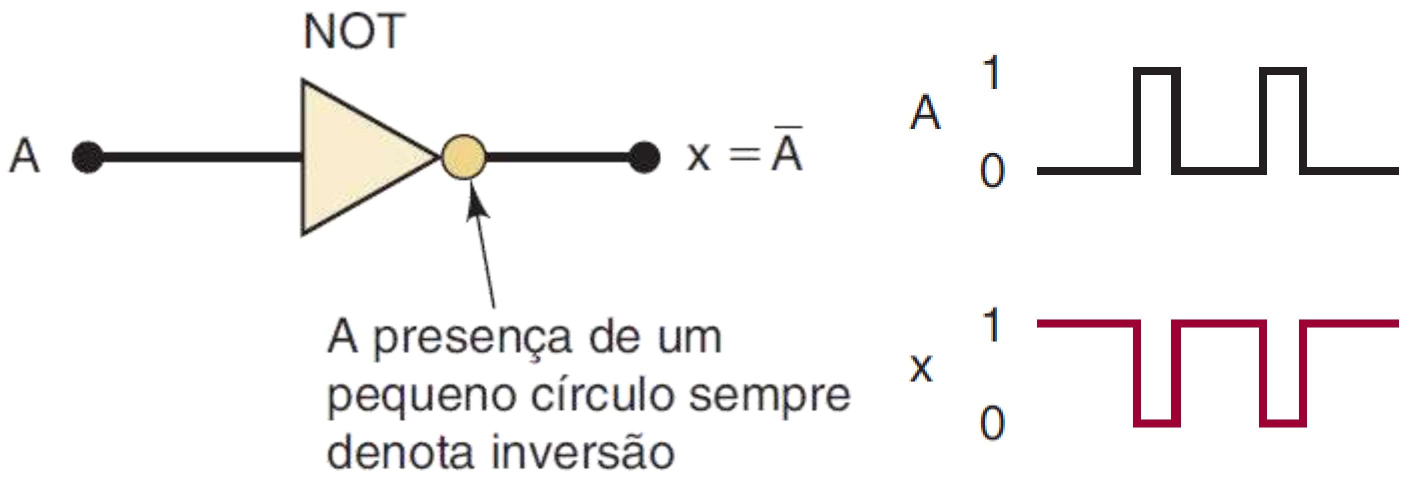
• Operação NOT

- A **operação NOT** é diferente das operações OR e AND pelo fato de poder ser realizada sobre uma única variável de entrada.

- Tabela-verdade; símbolo para o INVERSOR (circuito NOT) e amostras de formas de ondas:

NOT

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



• Operação NOT

- Complementação, negação

A operação **complementação** dispensa uma definição. É a operação cujo resultado é simplesmente o valor complementar ao que a variável apresenta. Também devido ao fato de uma variável Booleana poder assumir um entre somente dois valores, o valor complementar será **1** se a variável vale **0** e será **0** se a variável vale **1**.

Os símbolos utilizados para representar a operação complementação sobre uma variável Booleana A são $\overline{A}$, ~A e A' (lê-se A negado). Nesta disciplina, adotaremos o primeiro símbolo. O resultado da operação complementação pode ser listado:

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

E a tabela verdade para $\overline{A}$ é:

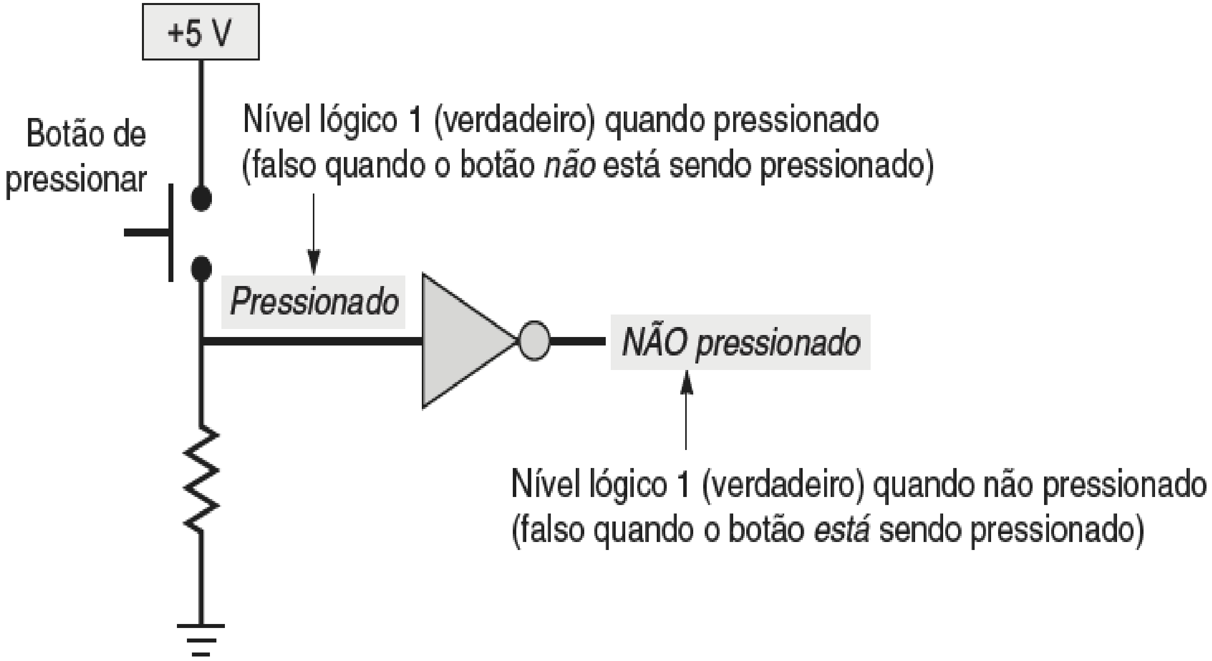
A	$\overline{A}$
0	1
1	0

Great Place To Work. Certificada

Católica
Universidade Católica de Brasília

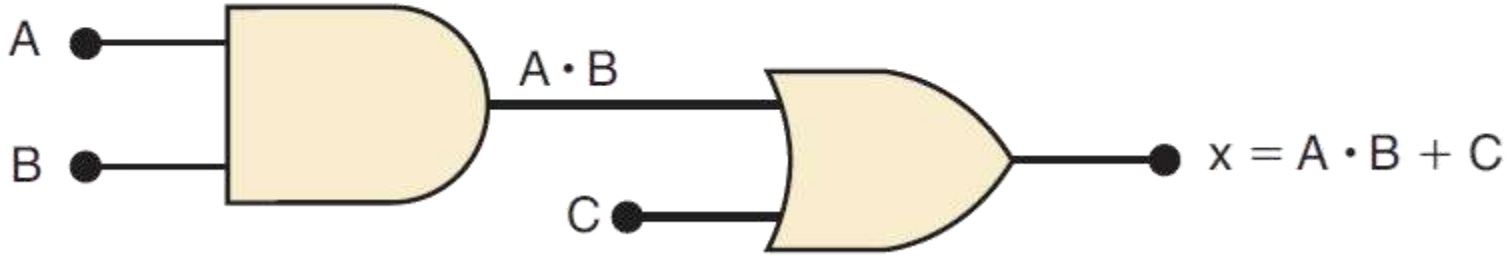
Operações lógicas básicas

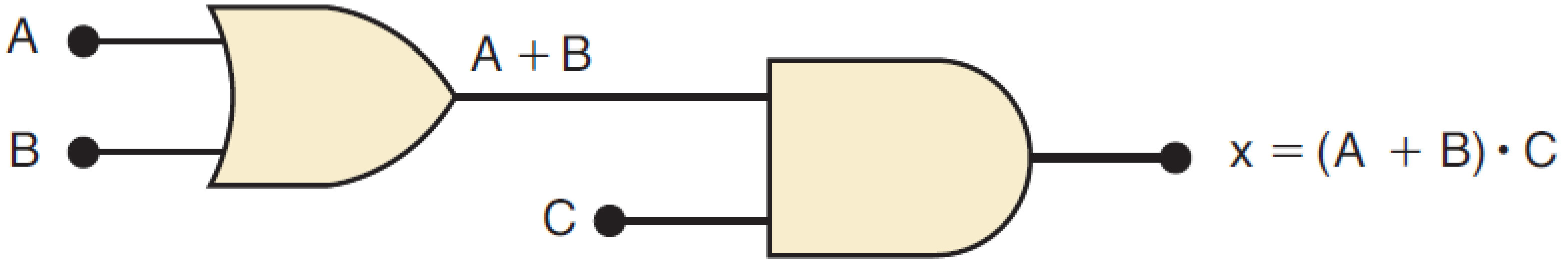
- Exemplo 1:



- **Descrição dos circuitos lógicos algebricamente**

- Circuito lógico e suas expressões booleanas; circuito lógico com expressão que requer parênteses:

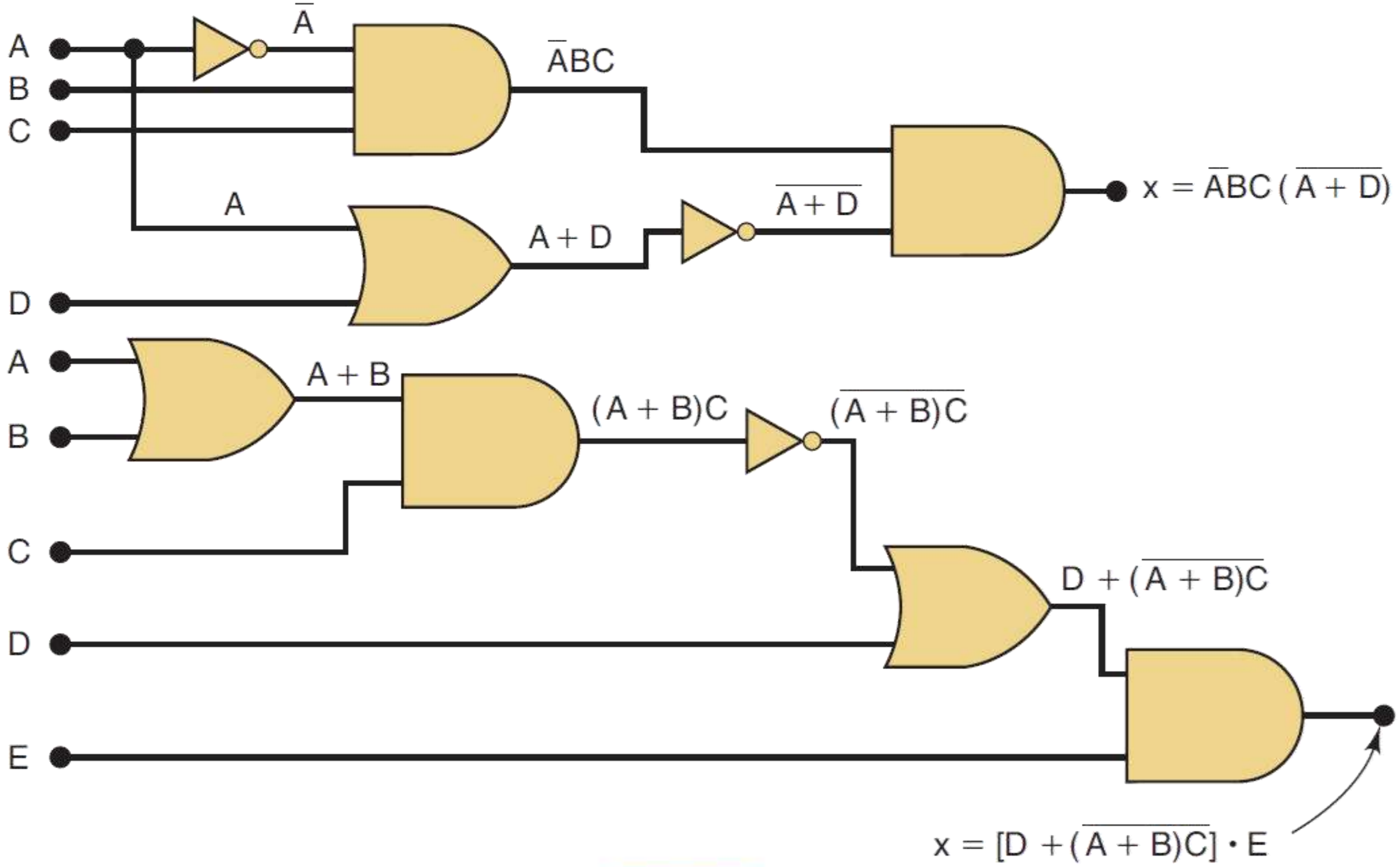




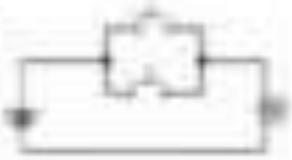
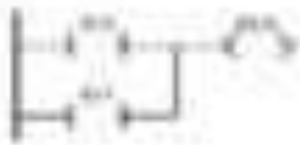
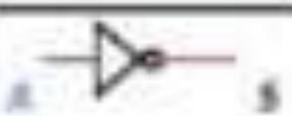
Católica
Universidade Católica de Brasília

- **Descrição dos circuitos lógicos algebricamente**

- Mais exemplos:



- **Eletrônica Digital e Circuitos Lógicos**

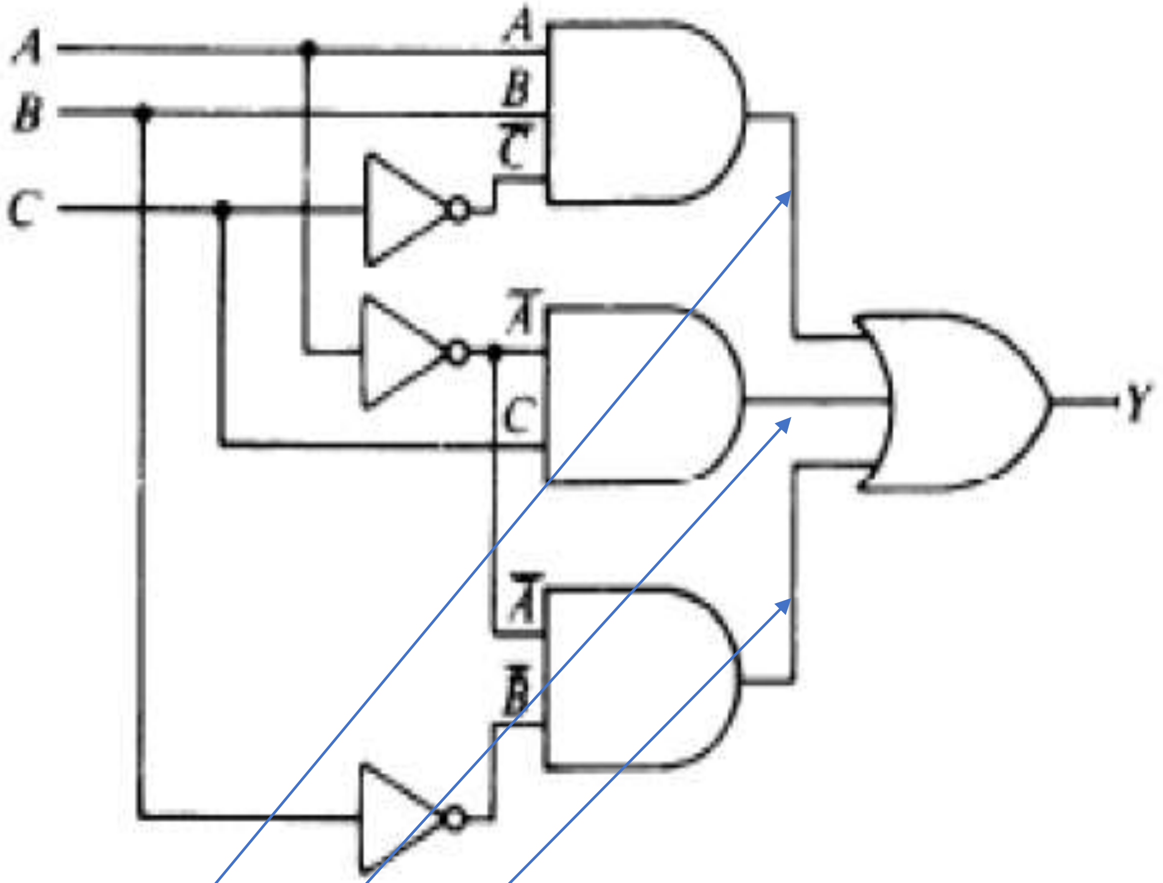
FUNÇÃO LÓGICA BÁSICA	SÍMBOLO GRÁFICO DA PORTA LÓGICA	EQUAÇÃO BOOLEANA	TABELA VERDADE	CIRCUITO ELÉTRICO EQUIVALENTE	CIRCUITO LADDER EQUIVALENTE
AND		$S = A \times B$	A B S 0 0 0 0 1 0 1 0 0 1 1 1		
OR		$S = A + B$	A B S 0 0 0 0 1 1 1 0 1 1 1 1		
NOT		$S = \overline{A}$	A S 0 1 1 0		

Católica
Universidade Católica de Brasília

- **Portas Lógicas x Equações Boleanas**

Função Lógica Básica	Símbolo Gráfico da Porta	Equação Booleana
AND		$Y = A \cdot B$
OR		$Y = A + B$
XOR		$Y = A \oplus B$
NOT		$Y = \overline{A}$
NAND		$Y = \overline{A \cdot B}$
NOR		$Y = \overline{A + B}$
XNOR		$Y = \overline{A \oplus B}$

Encontre o valor da saída Y



$$Y = A.B.\overline{C} + \overline{A}.C + \overline{A}.\overline{B}$$

Porta NOR

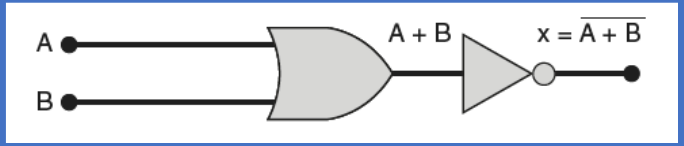
Operação NOR ("NÃO-OU")

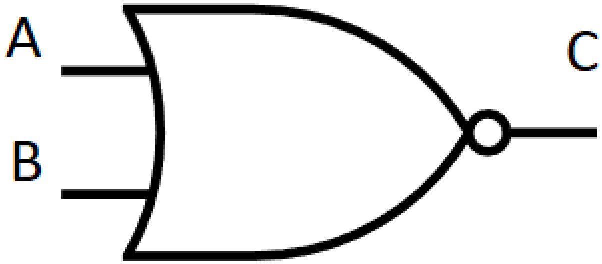
$$C = \overline{A + B}$$

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

- "Se A ou B é verdadeiro, então C é falso"

Porta NOR

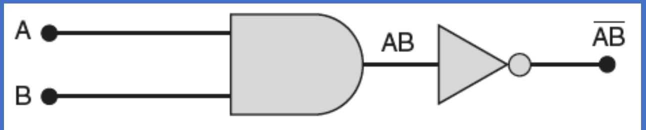




(padrão 91-1984 da ANSI/IEEE)

Universidade Católica de Brasília

Porta NAND

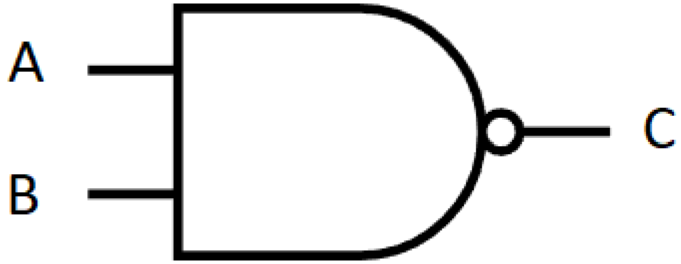


Operação NAND ("NÃO-E")

Porta NAND

$$C = \overline{A.B}$$

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



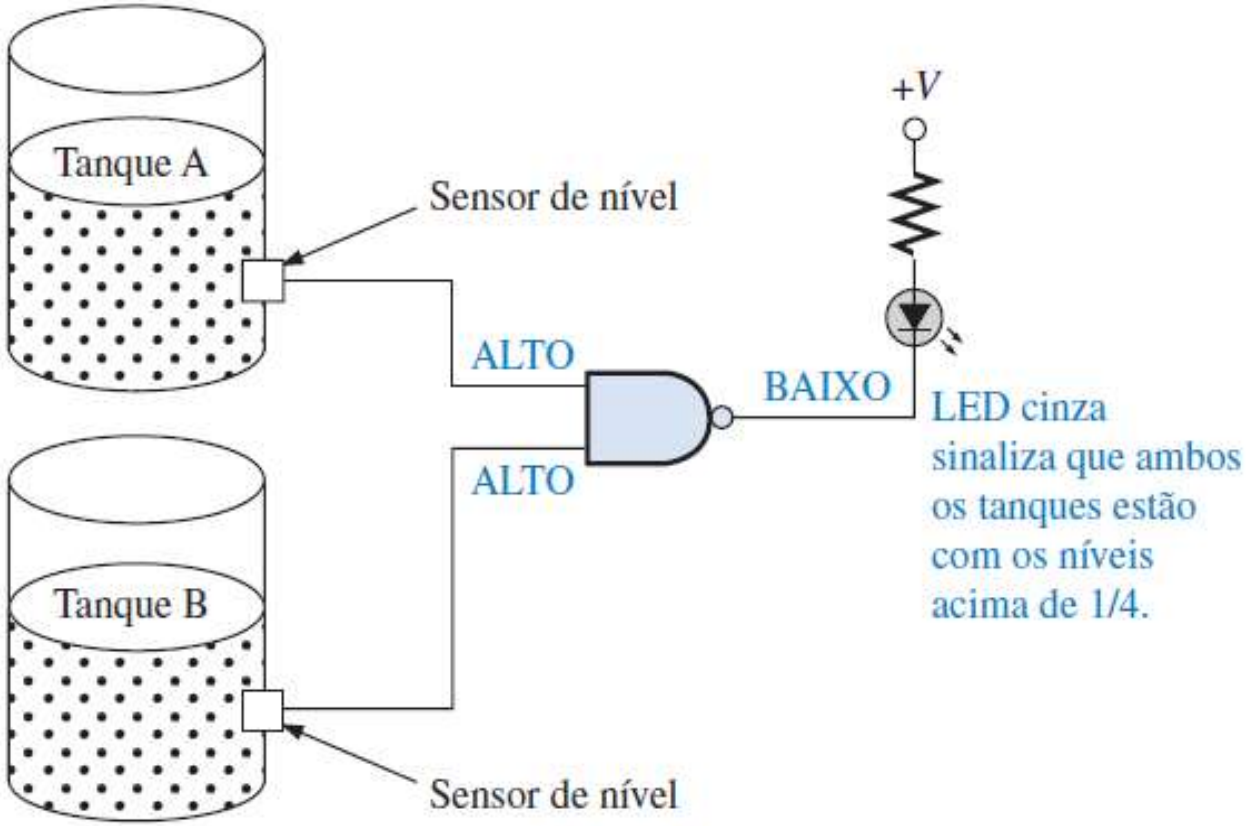
- "Se A e B é verdadeiro, então C é falso"

(padrão 91-1984 da ANSI/IEEE)

Universidade Católica de Brasília

Porta NAND

Exemplo:



Álgebra de Boole

Exercício

Questionário

Próxima aula

- Circuitos lógicos
  - Circuitos lógicos descritos algebricamente
  - Postulados / Axiomas
  - Teoremas booleanos

Dúvidas?