LÓGICA MATEMÁTICA

Professora: Izabel Cristina

ÁLGEBRA BOOLEANA

A **álgebra booleana** foi inventada no ano de **1854** pelo matemático inglês **George Boole**, ele declarou a ideia da álgebra de Boole em seu livro *"Uma Investigação das Leis do Pensamento"*.

A **Álgebra Booleana** usa funções e variáveis, como na álgebra convencional, que podem assumir apenas um dentre dois valores:

{Falso, Verdadeiro} - raciocínio humano

{Desligado, Ligado} - circuitos de chaveamento

{0, 1} - sistema binário

{0v, +5v} - eletrônica digital

Na **álgebra Booleana**, existem três operações ou funções básicas. São elas, operação **OU**, operação **E** e **complementação**. Todas as funções Booleanas podem ser representadas em termos destas operações básicas.

Operação OR (OU)

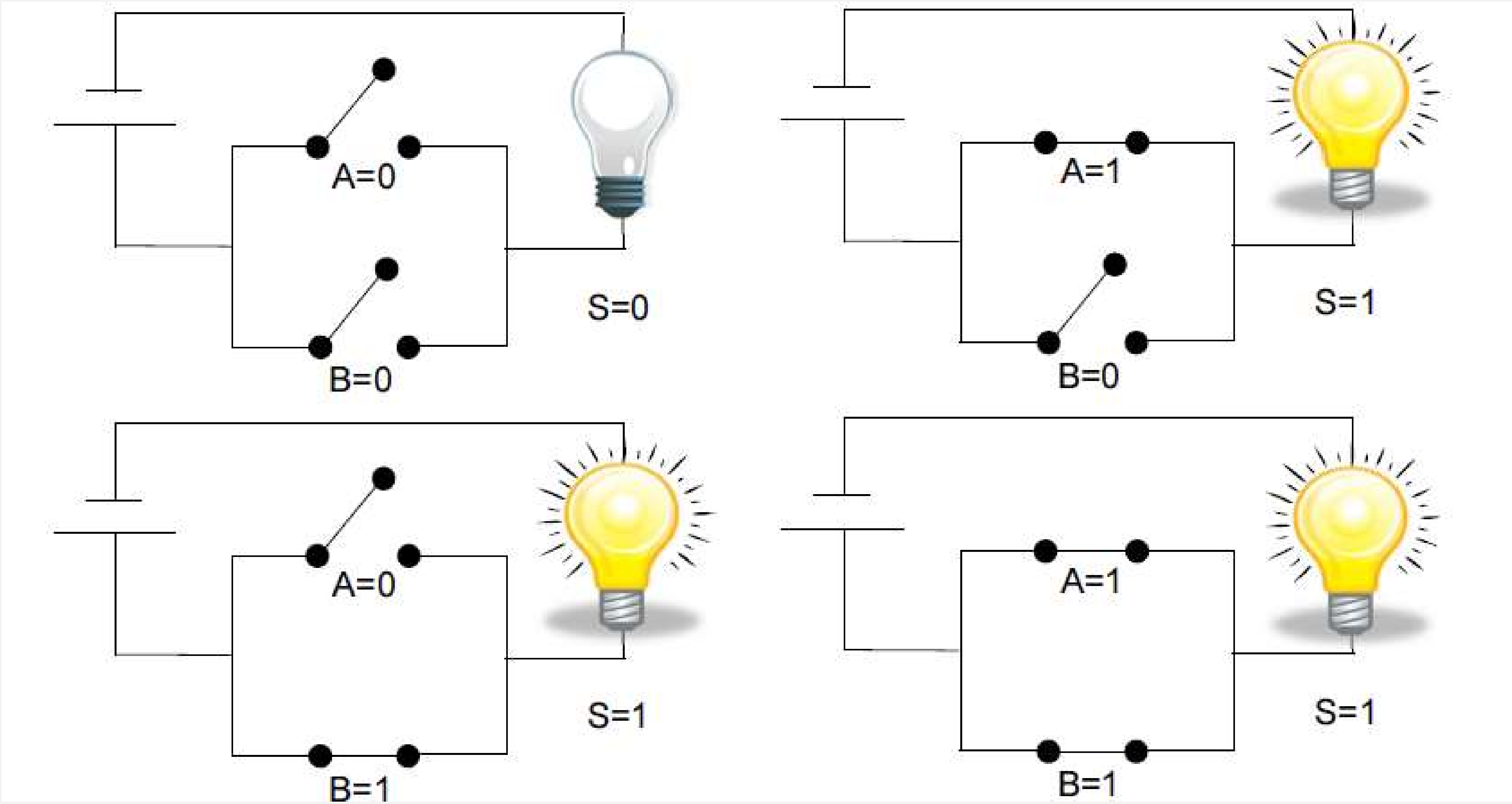
A **operação OR** é aquela que terá saída **verdadeiro** se pelo menos uma das duas condições for atendida.

Para entender vamos imaginar a situação em que temos uma lâmpada dentro de um forno e ela deve ser acesa se *o interruptor for acionado* **OU (OR)** se *a porta do forno for aberta*.

Ao fazer a combinação das duas entradas lógicas observamos que:

- **Se somente o interruptor for acionado a lâmpada é acesa;**
- **Se somente a porta do forno for aberta a lâmpada é acesa;**
- **Se o interruptor for acionado e a porta do forno for aberta a lâmpada é acesa;**
- Se nem o interruptor for acionado e nem a porta do forno for aberta a lâmpada não é acesa.

Um símbolo possível para representar a operação **OU** é "**+**", tal como o símbolo da adição algébrica (dos reais). Porém, como estamos trabalhando com variáveis Booleanas, sabemos que não se trata da adição algébrica, mas sim da adição lógica. Outro símbolo também encontrado na bibliografia é "V".

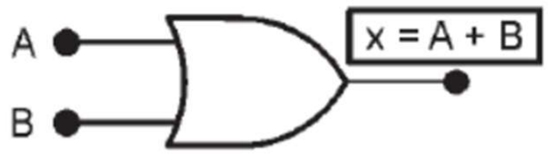

A=0
B=0
S=0
A=1
B=0
S=1
A=0
B=1
S=1
A=1
B=1
S=1

Operação OR (OU) e a Porta OR

- A operação OU de duas entradas A e B produz um nível lógico 1 como resultado se a entrada A **ou** a entrada B estiverem em nível lógico 1.

OR

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



Porta OR

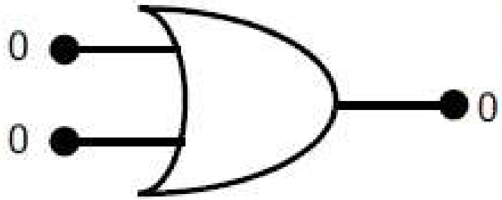
- A expressão booleana para a operação OU é:

$$x = A + B$$

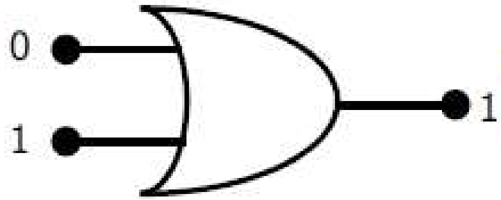
- Para três entradas, teremos:

$$x = A + B + C$$

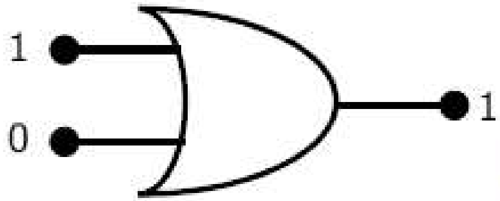
- Se A, B e C forem 1, teremos $x = 1 + 1 + 1 = 1$



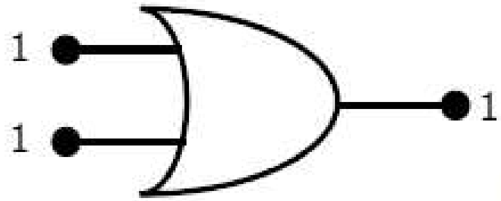
A	B	S=A+B
0	0	0
0	1	1
1	0	1
1	1	1



A	B	S=A+B
0	0	0
0	1	1
1	0	1
1	1	1



A	B	S=A+B
0	0	0
0	1	1
1	0	1
1	1	1



A	B	S=A+B
0	0	0
0	1	1
1	0	1
1	1	1

Operação AND (E)

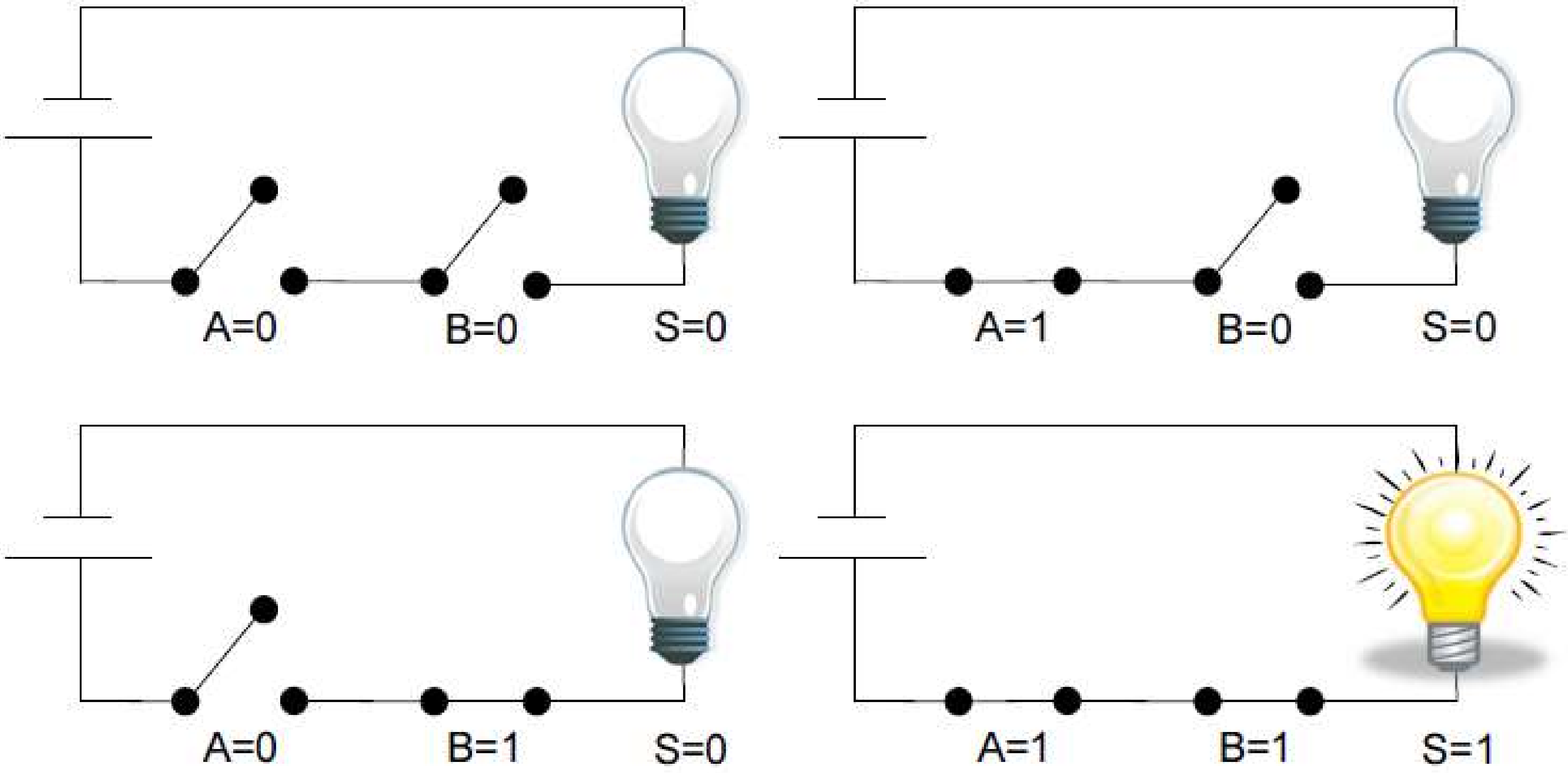
A **operação AND** é aquela que somente terá como saída **verdadeira** se **ambas** as condições forem atendidas.

Vamos considerar que uma secadora de roupas só opera se um *botão for acionado* **AND (E)** se a *porta estiver fechada*.

Ao fazer a combinação das duas entradas lógicas observamos que:

- Se somente o botão for acionado a secadora não irá secar;
- Se somente a porta estiver fechada a secadora não irá secar;
- **Se o botão for acionado e a porta estiver fechada a secadora irá secar;**
- Se nem o botão for acionado e nem a porta estiver fechada a secadora não irá secar.

- Situações possíveis:

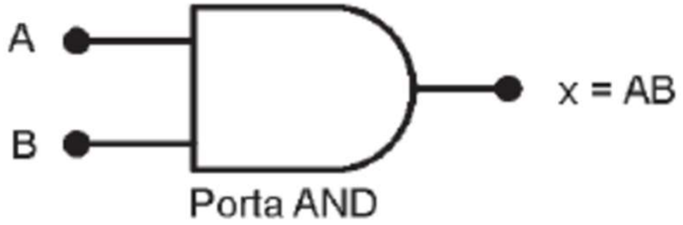


Operação AND (E) e a Porta AND

- A operação AND de duas entradas A e B produz um nível lógico 1 como resultado somente se ambas as entradas A e B estiverem em nível lógico 1.

AND

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

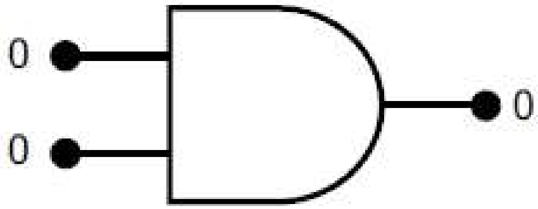


- A expressão booleana para a operação AND é:

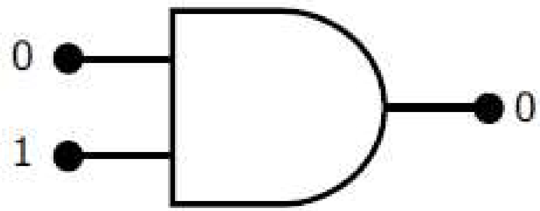
$$x = A.B = AB$$

- Para três entradas, teremos:

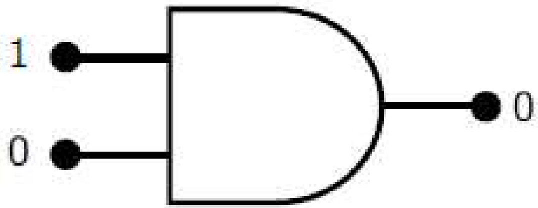
$$x = A.B.C = ABC$$



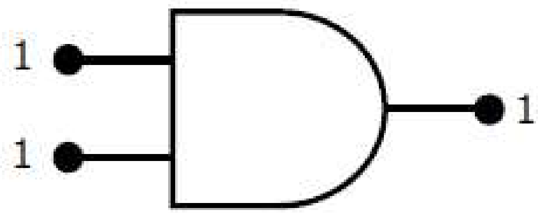
A	B	S=A.B
0	0	0
0	1	0
1	0	0
1	1	1



A	B	S=A.B
0	0	0
0	1	0
1	0	0
1	1	1



A	B	S=A.B
0	0	0
0	1	0
1	0	0
1	1	1



A	B	S=A.B
0	0	0
0	1	0
1	0	0
1	1	1

Operação NOT (NÃO)

A operação **NOT** é aquela que terá como saída o inverso da sua entrada.

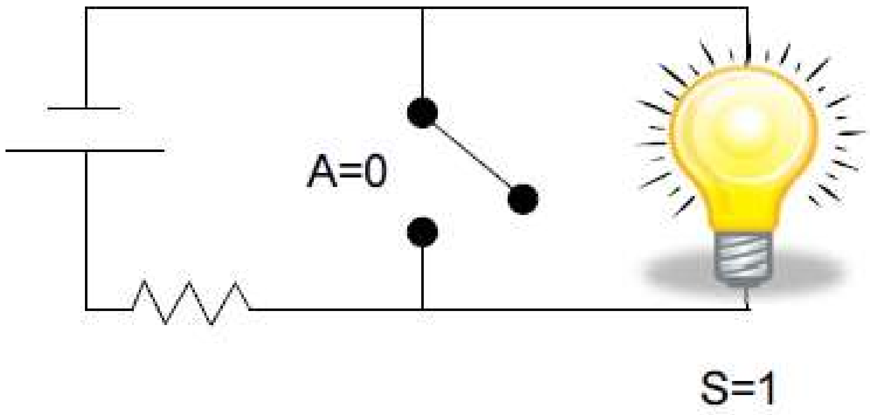
Diferente das operações **OR** e **AND**, essa operação é realizada sobre uma única entrada. Então, se a variável **A** invertida, o resultado **x** pode ser escrito como:

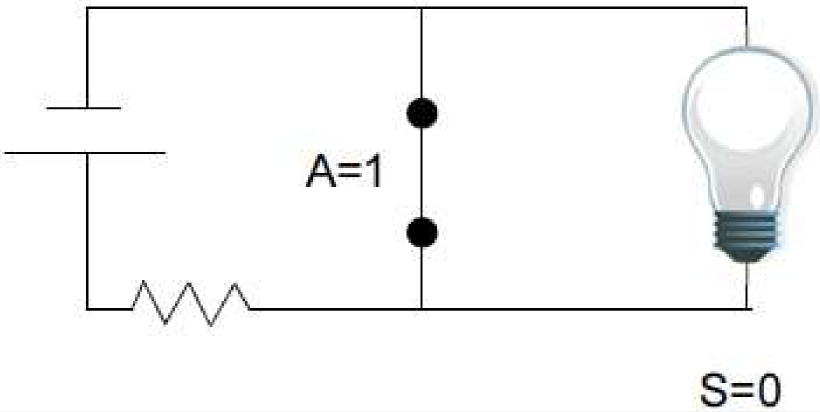
$$X = \overline{A}$$

Assim, lemos essa expressão como "**x é igual a A negado**", o "**x é igual ao inverso de A**" ou "**x é igual a A barrado**".

Todas as expressões indicam que o valor lógico de **x** é o oposto do valor lógico de **A**.

- Usando as mesmas convenções dos circuitos anteriores, tem-se que:
  - Quando a chave A está aberta (A=0), passará corrente pela lâmpada e ela acenderá (S=1)
  - Quando a chave A está fechada (A=1), a lâmpada estará em curto-circuito e não passará corrente por ela, ficando apagada (S=0)

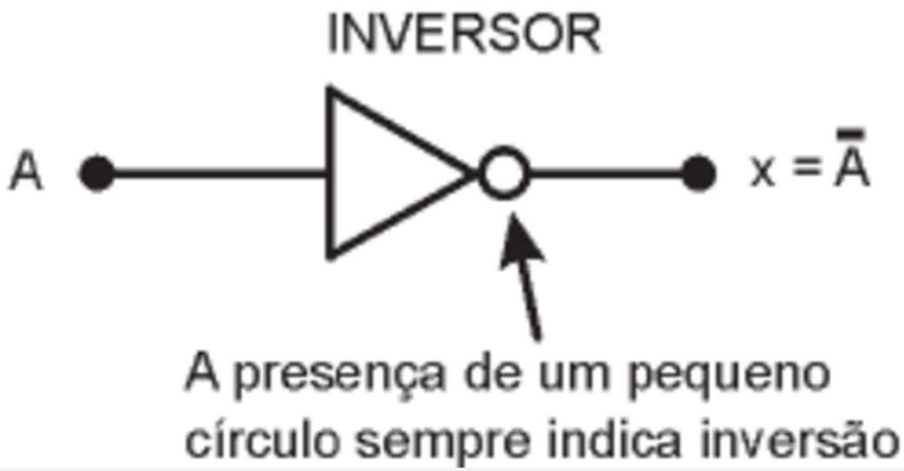


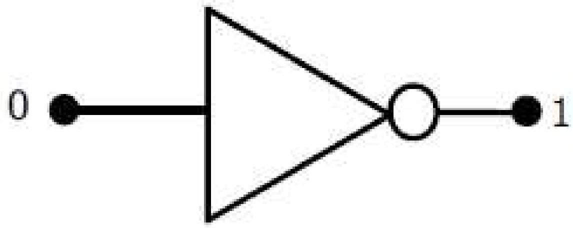


Os símbolos utilizados para representar a operação complementação (ou Negação, ou Inversão) sobre uma variável Booleana A são **$\bar{A}$**, **~A** e **A'**.

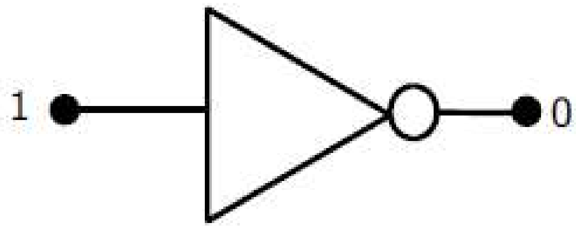
INVERSOR

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





A	S=Ā
0	1
1	0



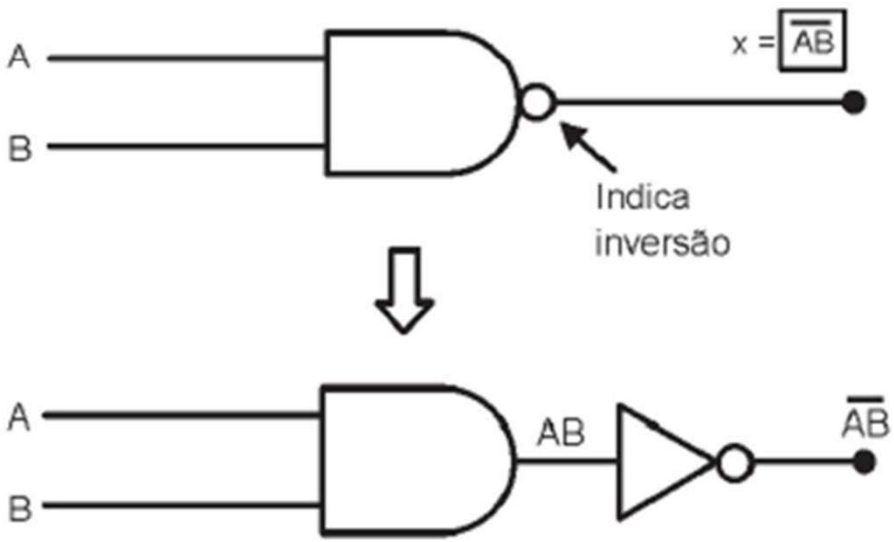
A	S=Ā
0	1
1	0

RESUMO DAS OPERAÇÕES LÓGICAS BOOLEANAS

OR	AND	NOT
$0 + 0 = 0$	$0 \cdot 0 = 0$	$\bar{0} = 1$
$0 + 1 = 1$	$0 \cdot 1 = 0$	$\bar{1} = 0$
$1 + 0 = 1$	$1 \cdot 0 = 0$	
$1 + 1 = 1$	$1 \cdot 1 = 1$	

NÃO E (NAND)

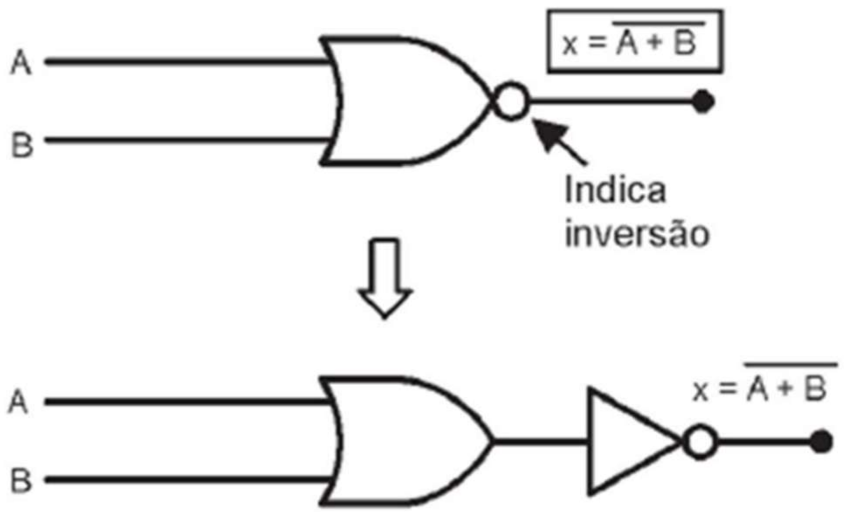
Equivale a uma porta **AND** seguida por uma porta **NOT**. Ela produz uma saída que é o inverso da saída produzida pela porta **AND**.



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

NÃO OU (NOR)

Equivale a uma porta **OR** seguida por uma porta **NOT**. Ela produz uma saída que é o inverso da saída produzida pela porta **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

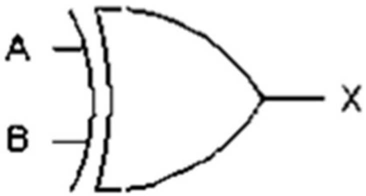
OU EXCLUSIVO (XOR)

A porta **XOR** produz **0** na saída quando todos os bits de entrada são iguais e **1** quando pelo menos um dos bits de entrada é diferente dos demais.

- **Função Booleana**

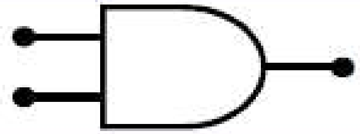
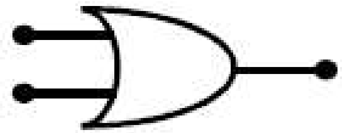
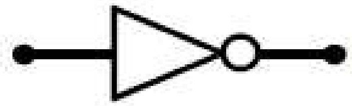
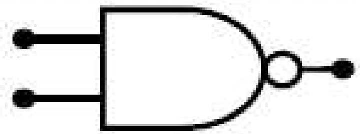
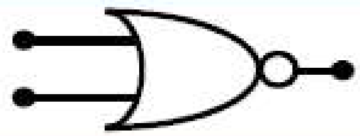
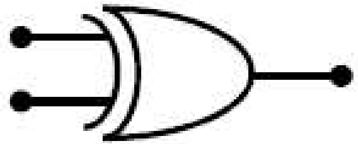
$$X = A \oplus B$$

- **Representação gráfica**



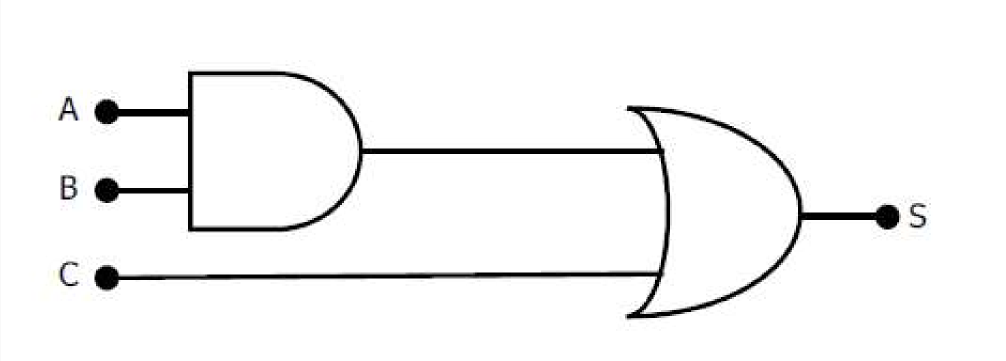
- **Tabela Verdade**

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

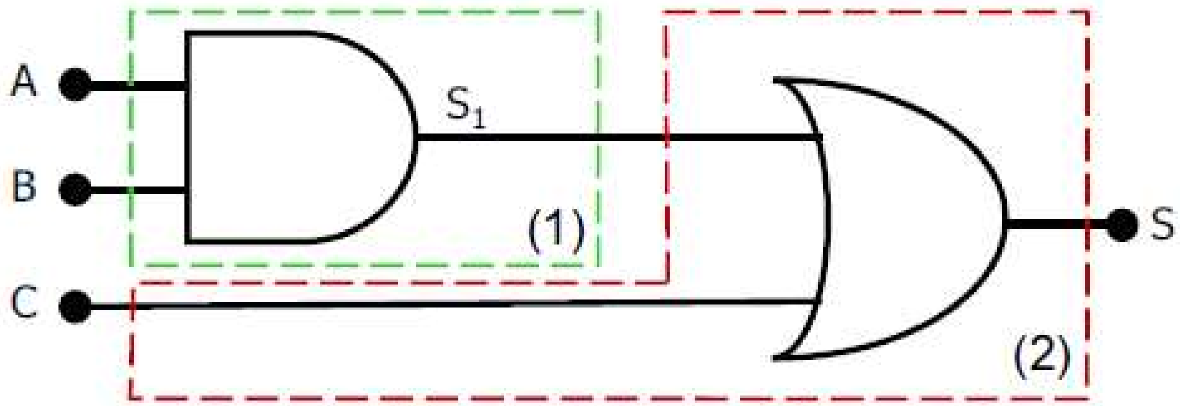
Nome	Símbolo Gráfico	Função Algébrica	Tabela Verdade
E (AND)		S=A.B S=AB	A B | S=A.B 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OU (OR)		S=A+B	A B | S=A+B 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
NÃO (NOT) Inversor		S=Ā S=A' S= ¬ A	A | S=Ā 0 | 1 1 | 0
NE (NAND)		$S=\overline{A.B}$ S=(A.B)' S= ¬(A.B)	A B | $S=\overline{A.B}$ 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOU (NOR)		$S=\overline{A+B}$ S=(A+B)' S= ¬(A+B)	A B | $S=\overline{A+B}$ 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
XOR		S=A⊕B	A B | S=A⊕B 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0

EXPRESSÕES BOOLEANAS GERADAS POR CIRCUITOS LÓGICOS

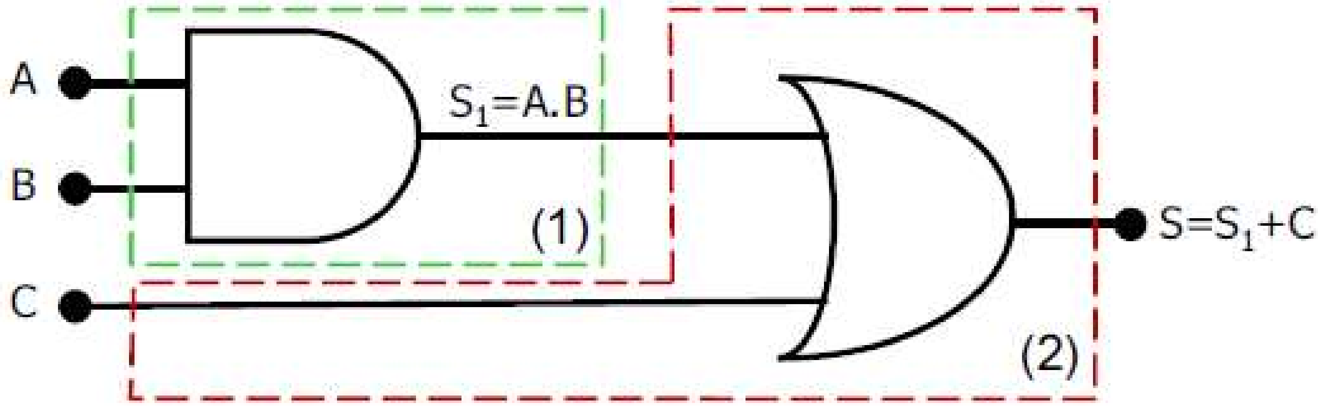
- Todo circuito lógico executa uma expressão booleana.
- Um circuito, por mais complexo que seja, é composto pela interligação dos blocos lógicos básicos.
- Seja o circuito a seguir:



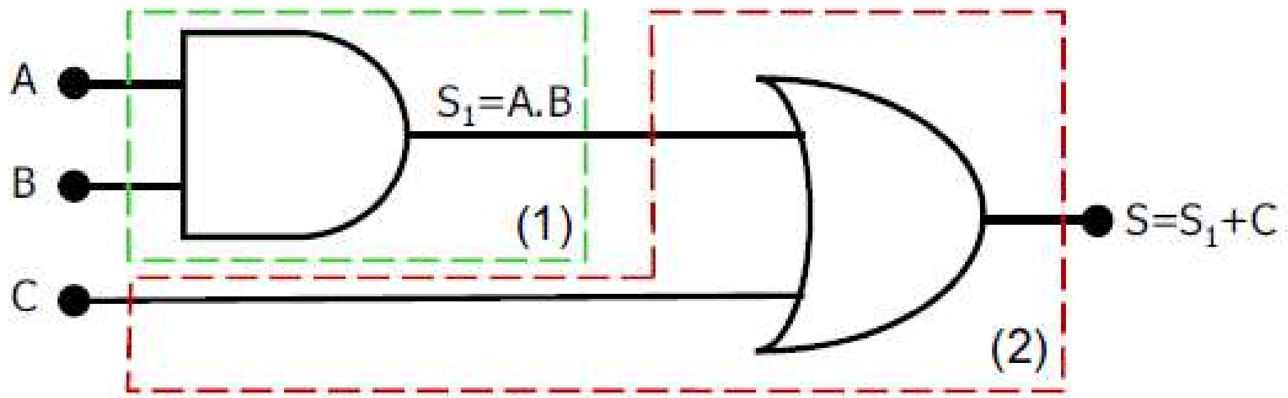
- Vamos dividi-lo em duas partes (1) e (2)
  - No circuito (1), a saída S_1 contém o produto A.B, já que o bloco é uma porta **E**
  - Portanto, S_1 = A.B



- No circuito (2), note que a saída S_1 é utilizada como uma das entradas da porta **OU**
- A outra entrada da porta **OU** corresponde à variável C, o que nos leva à:
  - $S = S_1 + C$

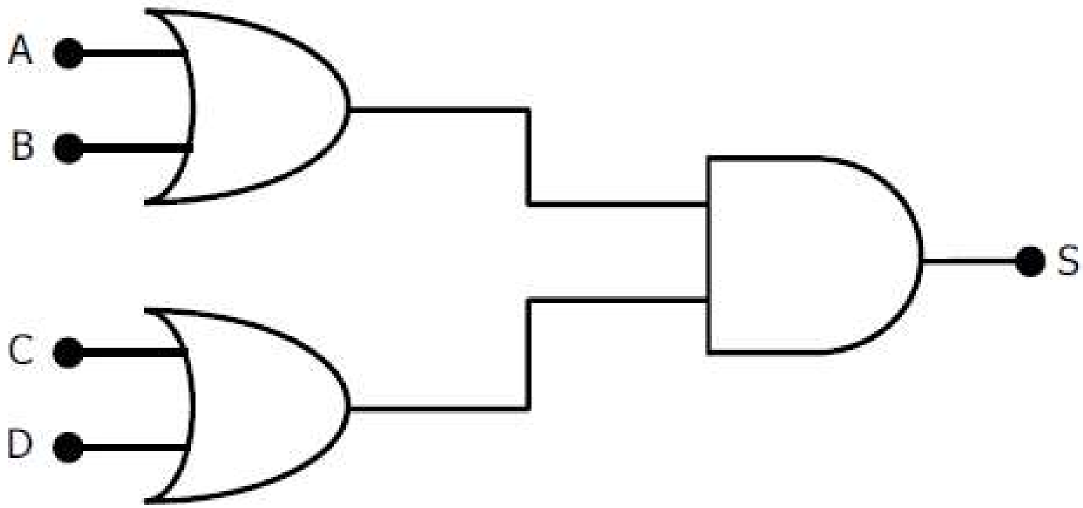


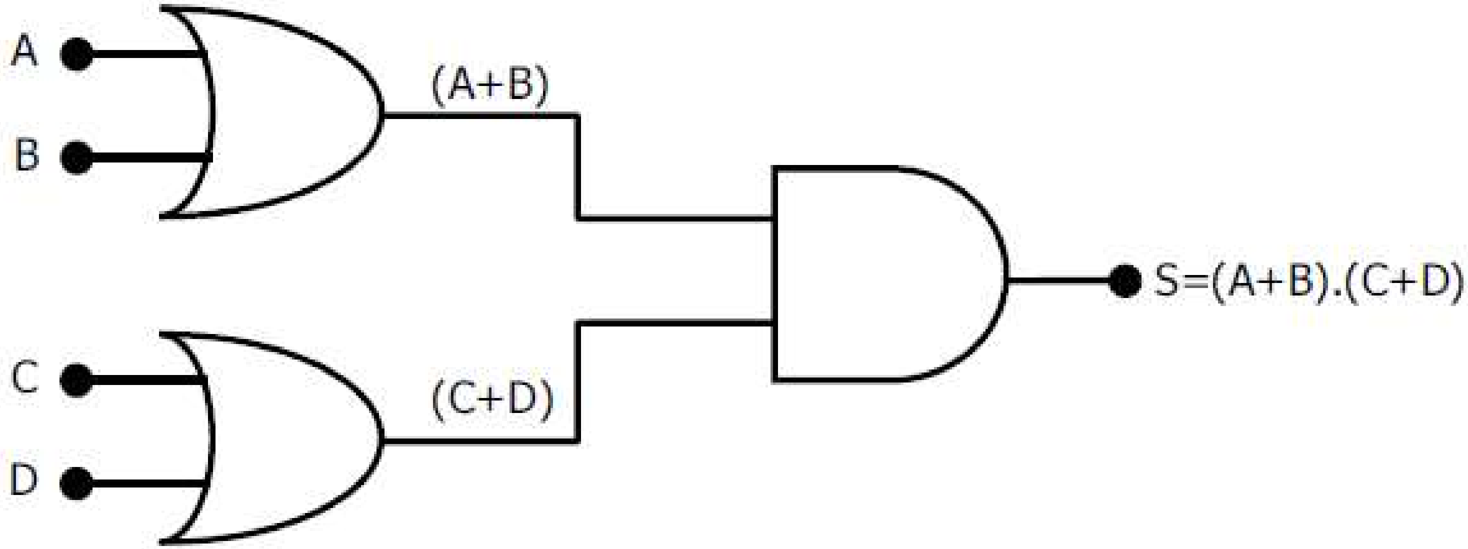
- Para obter a expressão final em relação às entradas A, B e C basta substituir a expressão S_1 na expressão de S, ou seja:
  - (1) $S_1 = A.B$
  - (2) $S = S_1 + C$
  - Obtém-se $S = S_1 + C = (A.B) + C$



Exercício

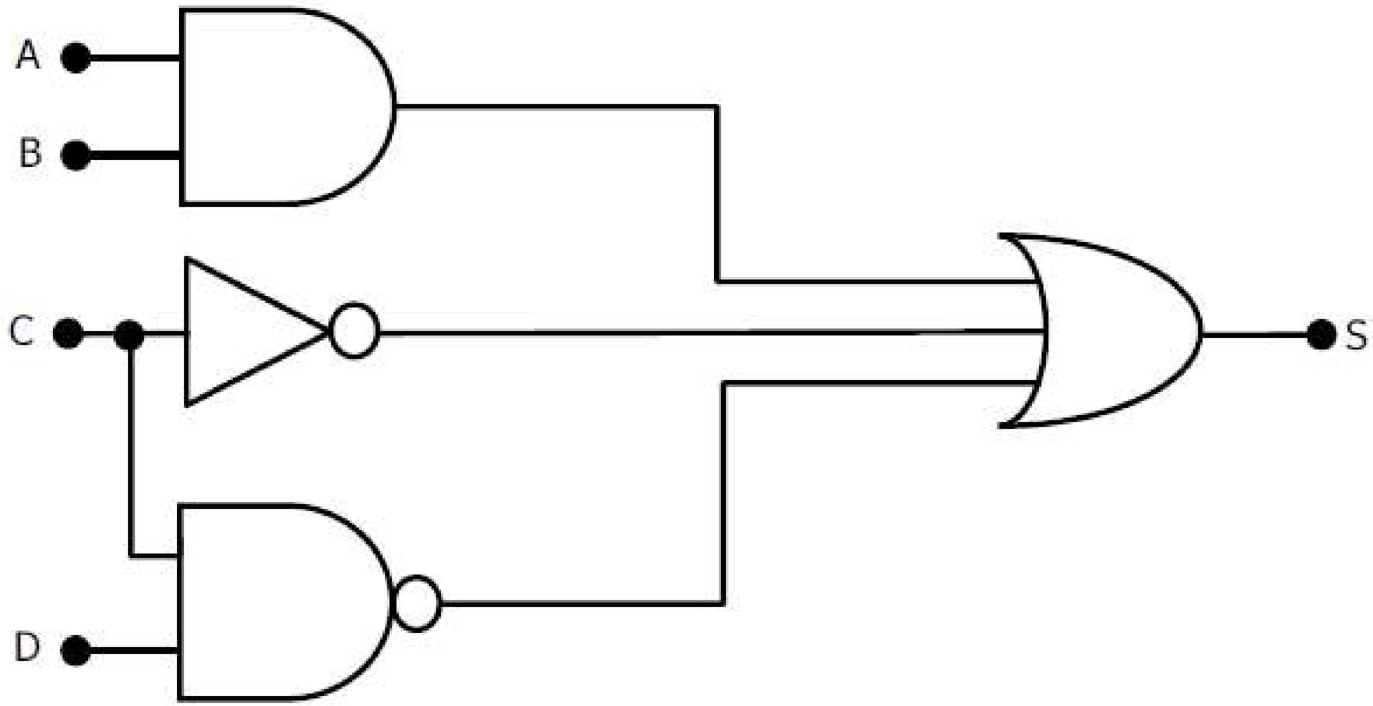
- Escreva a expressão booleana executada pelo circuito

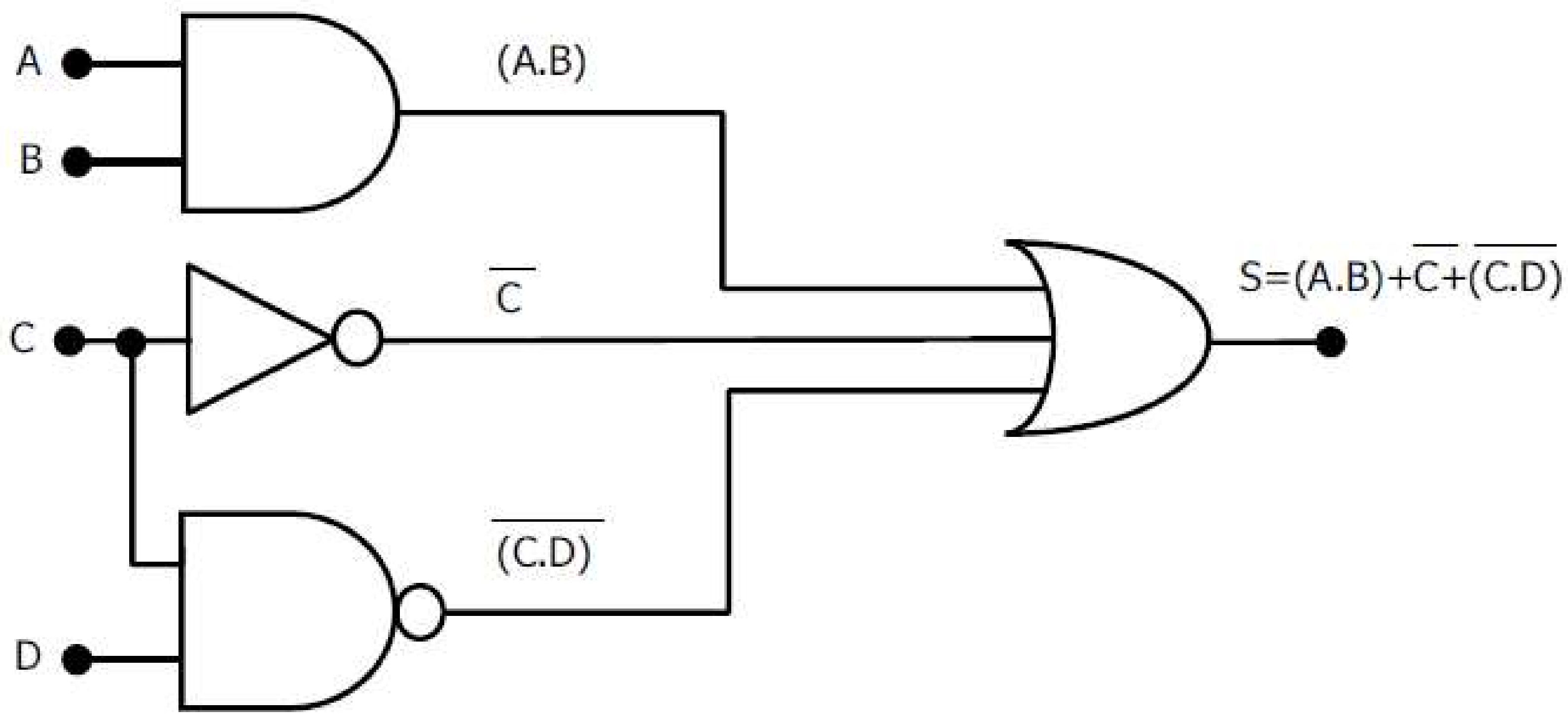



A
B
(A+B)
C
D
(C+D)
S=(A+B).(C+D)

Exercício

- Determinar a expressão booleana característica do circuito

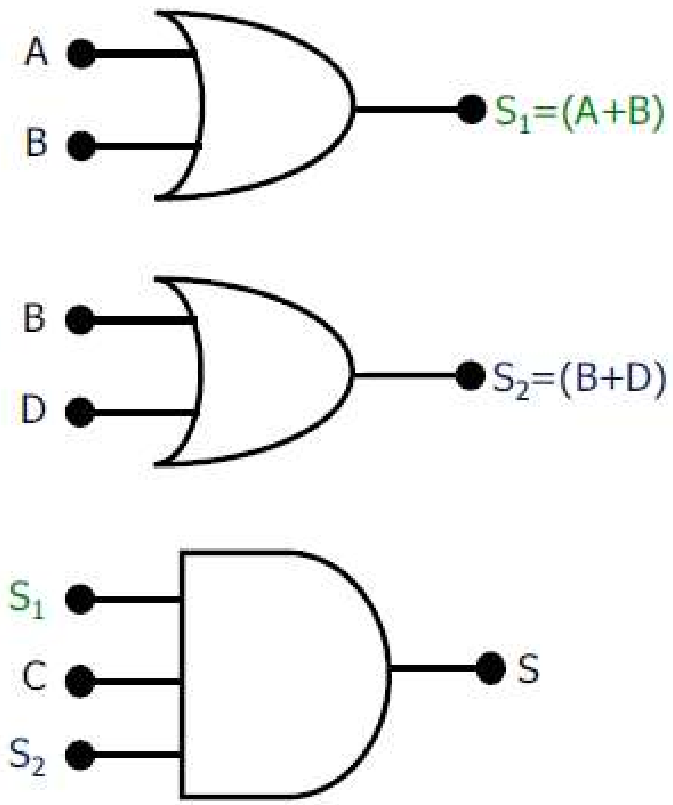



A
B
C
D
(A.B)
$\overline{C}$
$\overline{(C.D)}$
$S=(A.B)+\overline{C}+\overline{(C.D)}$

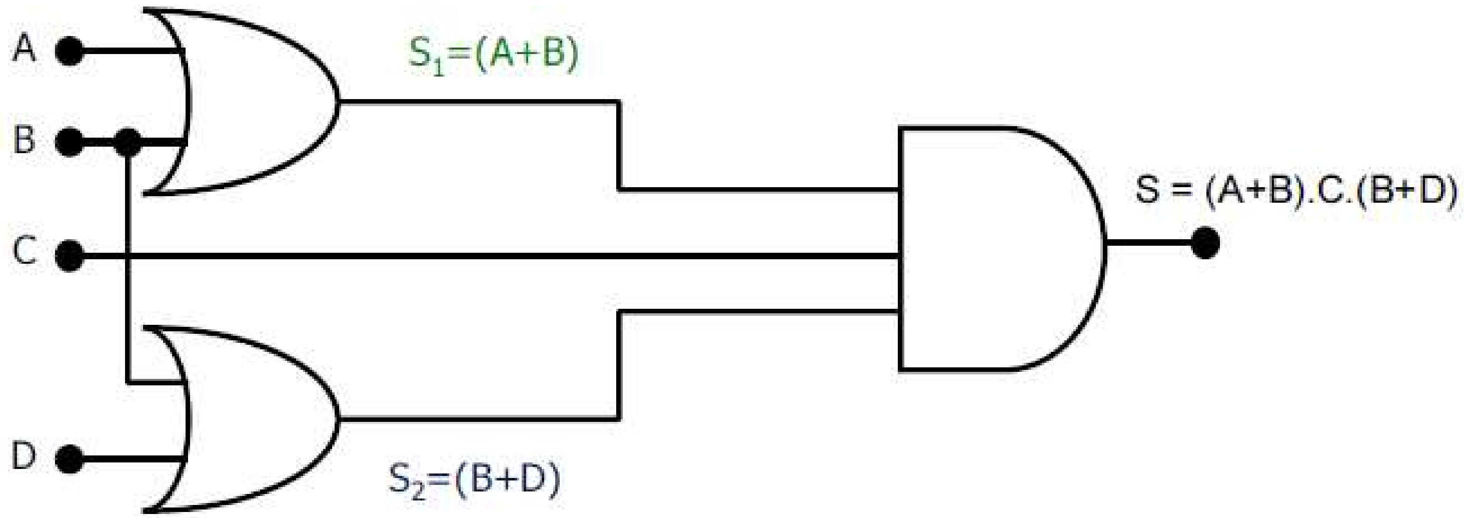
CIRCUITOS GERADOS POR EXPRESSÕES BOOLEANAS

- Até o momento, vimos como obter uma expressão característica a partir de um circuito.
- Também é possível obter um circuito lógico, dada uma expressão booleana.
- Nesse caso, como na aritmética elementar, parênteses têm maior prioridade, seguidos pela multiplicação (**função E**) e, por último, pela soma (**função OU**).

- Seja a expressão
  - S = (A+B).C.(B+D)
- Vamos separar as subfórmulas da expressão, ou seja:
  - S = (A+B) . C . (B+D)
- Dentro do primeiro parêntese temos a soma booleana S_1=(A+B), portanto o circuito que executa esse parêntese será uma porta **OU**
- Dentro do segundo parêntese temos a soma booleana S_2=(B+D). Novamente, o circuito que executa esse parêntese será uma porta **OU**
- Portanto, temos:
  - S = S_1 . C . S_2
- Agora temos uma multiplicação booleana e o circuito que a executa é uma porta **E**



- O circuito completo é:

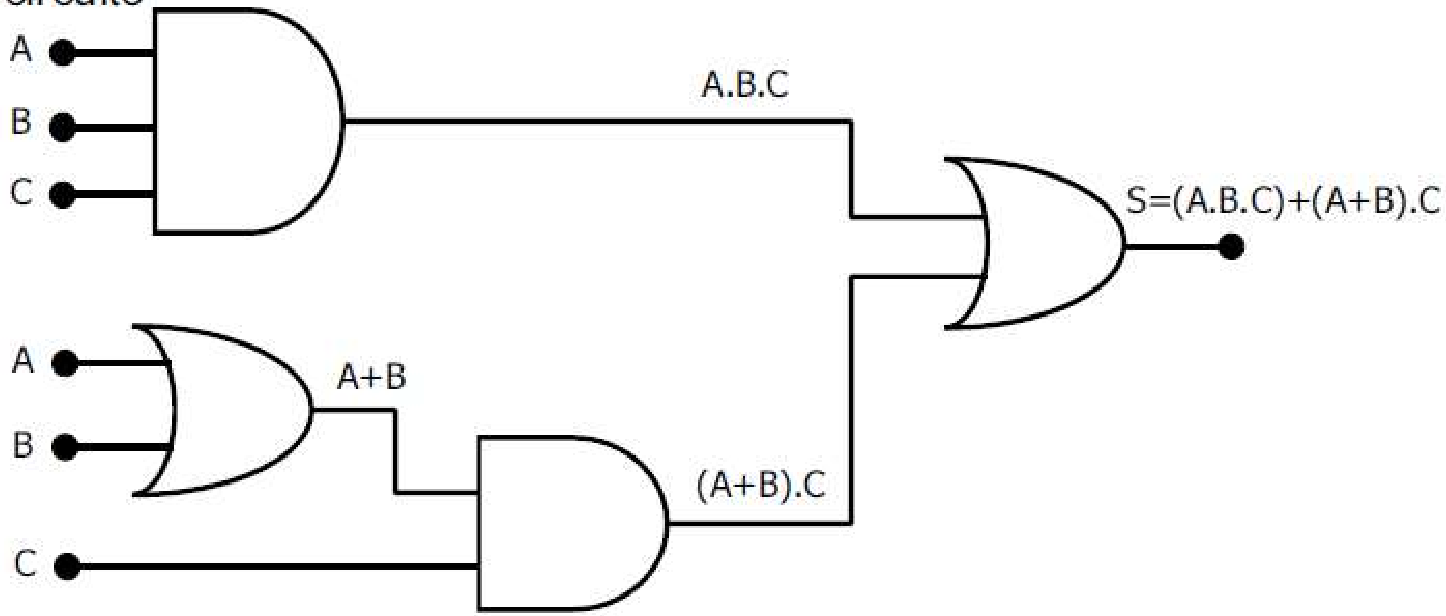


Exercício

- Desenhe o circuito lógico que executa a seguinte expressão booleana
  - $S = (A.B.C) + (A+B).C$

Solução

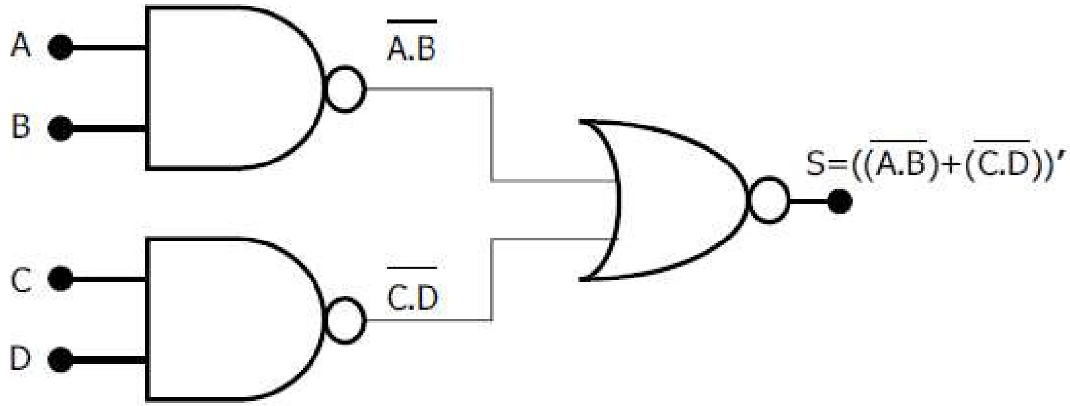
- É importante lembrar que as entradas que representam a mesma variável estão interligadas
- Contudo o desenho sem interligações facilita a interpretação do circuito



Exercício

- Desenhe o circuito lógico cuja expressão característica é
  - $S = (\overline{A.B} + \overline{C.D})'$

Solução



OBRIGADO!