

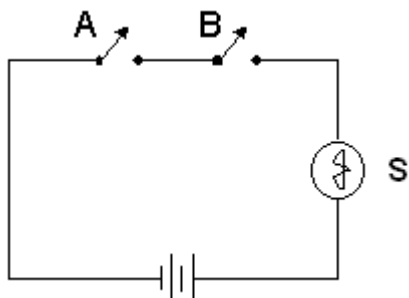
## PORTAS LÓGICAS

Em 1958 um engenheiro americano chamado Claude Elwood Shannon utilizou a Lógica Proposicional (também conhecida como Lógica de Boole ou Lógica Booleana) para a solução de problemas de circuitos, dando origem à Eletrônica Digital. Esse ramo emprega um pequeno grupo de circuitos básicos padronizados chamados **portas lógicas**. Através da utilização delas, podemos implementar qualquer expressão da lógica proposicional. Para mais facilmente relacionarmos os conhecimentos anteriores com a notação nova, tabelas comparativas serão apresentadas a cada novo conceito:

Notação Lógica	Notação Eletrônica
Valor lógico	Estado
Verdadeiro ou Verdade	"1" ou "ligado" ou "chave fechada" ou "lâmpada acesa"
Falso ou Falsidade	"0" ou "desligado" ou "chave aberta" ou "lâmpada apagada"
Operação Lógica	Função Lógica

### Funções Lógicas:

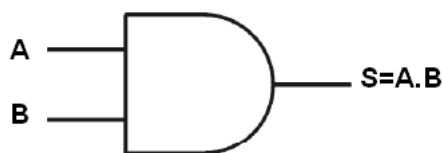
**FUNÇÃO E ou AND:** Executa a operação lógica E ( $\wedge$ ) na qual só circulará corrente (estado "1") caso todas as chaves do circuito estejam fechadas. É a representação do circuito em série. A notação é **A.B** que se lê "A e B".



Circuito Eletrônico E (AND)

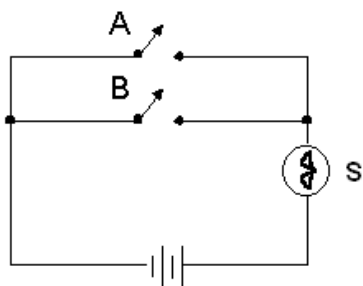
A	B	S=A.B
0	0	0
0	1	0
1	0	0
1	1	1

Tabela Verdade E (AND)



Porta Lógica E (AND)

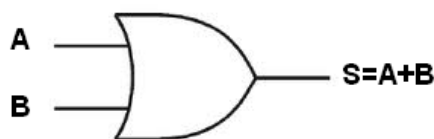
**FUNÇÃO OU ou OR:** Executa a operação lógica OU ( $\vee$ ) na qual circulará corrente (estado "1") quando pelo menos uma das chaves do circuito estiver fechada. Para isso, o circuito é apresentado em paralelo. A notação é **A + B** que se lê "A ou B".



Circuito Eletrônico OU (OR)

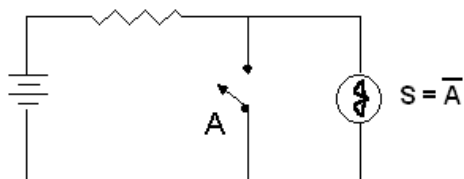
A	B	S=A+B
0	0	0
0	1	1
1	0	1
1	1	1

Tabela Verdade OU (OR)



Porta Lógica OU (OR)

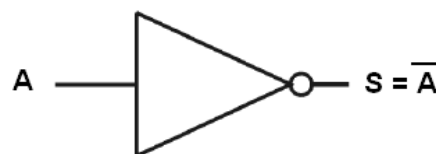
**FUNÇÃO NÃO ou NOT:** Executa a operação lógica NÃO ( $\sim$ ) que é aquela que inverte ou complementa o estado da variável. A notação é  **$\bar{A}$**  que se lê "não A" ou "A barrado".



Circuito Eletrônico NÃO (NOT)

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

Tabela Verdade NÃO (NOT)



Porta Lógica NÃO (NOT)

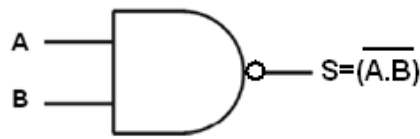
**\*INVERSOR:** O inversor (representado ao lado) pode executar a função NOT em qualquer bloco lógico, podendo vir antes ou depois deste dependendo da operação que se deseja obter. A seguir, alguns blocos obtidos com o uso do inversor.



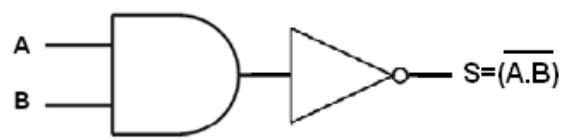
**FUNÇÃO NÃO E, NE ou NAND:** Executa a composição das operações de negação ( $\sim$ ) e a operação lógica E ( $\wedge$ ). É a inversão da função E, representada logicamente por  $\sim(A \wedge B)$ . Nela, só não circulará corrente (estado "0") caso todas as chaves do circuito estejam fechadas. A notação é  $(\overline{A.B})$  que se lê "A e B barrados" ou "não é verdade que A e B".

A	B	$S = (\overline{A.B})$
0	0	1
0	1	1
1	0	1
1	1	0

Tabela Verdade NE (NAND)



Porta Lógica NE (NAND)

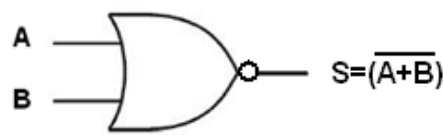


Composição das portas E e NOT formando o NE

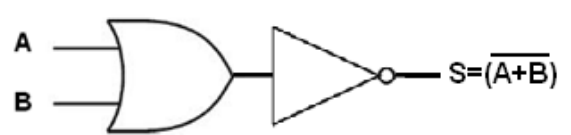
**FUNÇÃO NÃO OU, NOU ou NOR:** Executa a composição das operações de negação ( $\sim$ ) e a operação lógica OU ( $\vee$ ). É a inversão da função OU, representada logicamente por  $\sim(A \vee B)$ . Nela, só circulará corrente (estado "1") caso todas as chaves do circuito estejam abertas. A notação é  $(\overline{A+B})$  que se lê "A ou B barrados" ou "não é verdade que A ou B".

A	B	$S = (\overline{A+B})$
0	0	1
0	1	0
1	0	0
1	1	0

Tabela Verdade NOU (NOR)



Porta Lógica NOU (NOR)



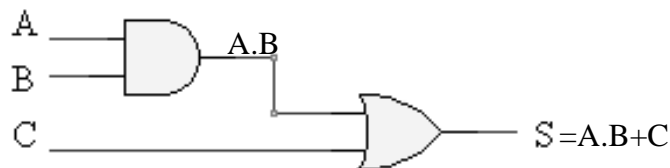
Composição das portas OU e NOT formando o NOU

### Expressões booleanas obtidas a partir de circuitos lógicos:

Todo circuito lógico executa uma expressão booleana (uma proposição lógica) simples ou composta, sendo sempre formado pela combinação e interligação das portas lógicas básicas. Eis um exemplo:



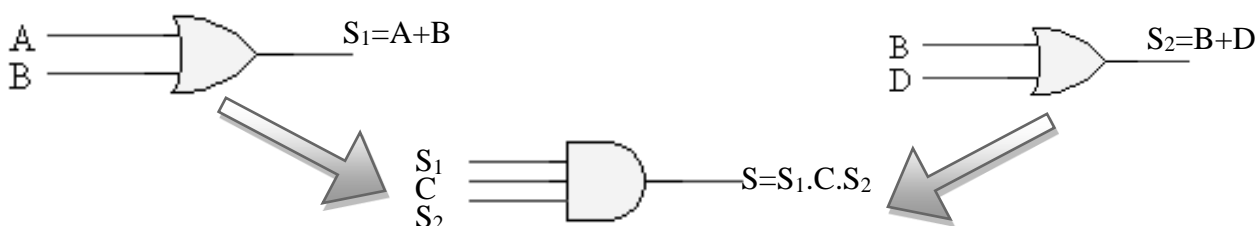
A saída  $S_1$  é dada por  $A.B$ . Como  $S_1$  é a entrada da porta OU, temos que  $S = S_1 + C$ . Substituindo a expressão de  $S_1$ , temos:



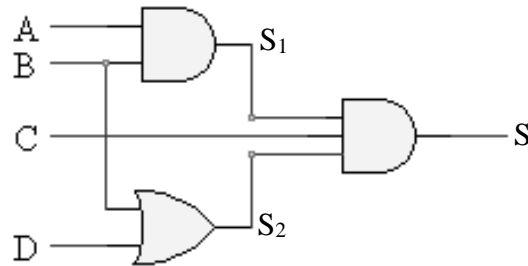
*Obs.: podemos considerar a prioridade entre as operações AND e OR a mesma entre as operações de produto e soma da aritmética, portanto, dispensam-se os parênteses.*

### Circuitos Lógicos obtidos a partir de expressões booleanas:

É possível desenhar o circuito lógico que executa uma expressão booleana qualquer verificando as prioridades e hierarquias estabelecidas na expressão. Eis um exemplo: obter o circuito que executa  $S = (A+B).C.(B+D)$



Inserindo os blocos de  $S_1$  e  $S_2$  no bloco AND predominante na expressão, temos:



### Tabelas-verdade obtidas a partir de expressões booleanas:

Extraír a tabela verdade de uma expressão booleana segue o mesmo princípio da construção de tabelas-verdade para proposições lógicas:

1. Montar o referencial de possibilidades
2. Montar as colunas para os grupos das expressões (saídas parciais)
3. Preencher os resultados parciais
4. Montar a coluna da saída
5. Preencher o resultado da saída do circuito

*Obs<sub>1</sub>.*: podemos abstrair colunas mais simples, como as da operação de inversão, considerando a habilidade pre0existente no preenchimento das tabelas-verdade proposicionais

*Obs<sub>2</sub>.*: lembre-se de considerar agora uma **prioridade natural** do AND em relação ao OR

Ex.:  $A \cdot \bar{B} \cdot C + A \cdot \bar{D} + \bar{A} \cdot B \cdot D$

Depois dos passos 1 e 2 apresentados acima, podemos conferir as possibilidades de cada saída parcial sem precisar construir uma coluna para as operações NOT, assim:

- (a) Saída  $S_1 = A \cdot \bar{B} \cdot C$ : como esta é uma porta AND, ela só terá saída 1 quando  $A=1$ ,  $B=0$  e  $C=1$
- (b) Saída  $S_2 = A \cdot \bar{D}$ : também uma porta AND, logo terá saída 1 quando  $A=1$  e  $D=0$
- (c) Saída  $S_3 = \bar{A} \cdot B \cdot D$ : nesta porta AND, saída será 1 quando  $A=0$ ,  $B=1$  e  $D=1$

	A	B	C	D	$S_1 = A \cdot \bar{B} \cdot C$	$S_2 = A \cdot \bar{D}$	$S_3 = \bar{A} \cdot B \cdot D$	$S = S_1 + S_2 + S_3$
	0	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	0
	0	1	0	0	0	0	0	0
(c)	0	1	0	1	0	0	1	1
	0	1	1	0	0	0	0	0
(c)	0	1	1	1	0	0	1	1
(b)	1	0	0	0	0	1	0	1
	1	0	0	1	0	0	0	0
(a) e (b)	1	0	1	0	1	1	0	1
(a)	1	0	1	1	1	0	0	1
(b)	1	1	0	0	0	1	0	1
	1	1	0	1	0	0	0	0
(b)	1	1	1	0	0	1	0	1
	1	1	1	1	0	0	0	0

É possível também preencher a coluna final de uma vez, observando os casos particulares que notavelmente terão resposta 1 (ou 0) a partir de uma análise simples das propriedades da Álgebra Booleana ou Álgebra das Proposições (Notação: Tautologia representa-se por 1 e Contradição por 0).

Ex.:  $\bar{A} + B + A \cdot \bar{B} \cdot \bar{C}$ .

- (1) Nos casos em que  $A = 0$  (ou seja,  $\bar{A} = 1$ ),  $S = 1$ , pois  $1 + X = 1$ .
- (2) Pelo mesmo motivo, nos casos em que  $B = 1$ ,  $S = 1$ .
- (3) O termo  $A \cdot \bar{B} \cdot \bar{C}$  será 1 somente no caso 100
- (4) Por exclusão, ou por atribuição de valores, o caso remanescente (101) será 0

A	B	C	S
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

*Obs.*: para analisar o comportamento (tabela-verdade) de um circuito, basta extrair primeiramente sua expressão e aplicar um dos procedimentos acima.

## Expressões booleanas obtidas a partir de tabelas verdade:

Em projetos práticos, é mais comum precisarmos extrair a expressão booleana de um comportamento desejado, portanto, a partir de uma tabela-verdade, é possível obter a expressão (e consequentemente o circuito lógico) seguindo o procedimento descrito abaixo:

1. Identificar na tabela verdade os casos onde a saída  $S=1$
2. Montar os termos de cada caso usando a operação AND
3. Somá-las usando o OR

Note-se que este método permite obter uma expressão padrão de qualquer tabela verdade formada pela **soma de produtos**. Ex.:

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

→ Caso 00:  $S=1$  quando  $A=0$  e  $B=0$  (ou seja,  $\bar{A} = 1$  e  $\bar{B} = 1$ )  $\Rightarrow \bar{A} \cdot \bar{B}$

→ Caso 10:  $S=1$  quando  $A=1$  e  $B=0$  (ou seja,  $A=1$  e  $\bar{B} = 1$ )  $\Rightarrow A \cdot \bar{B}$

→ Caso 11:  $S=1$  quando  $A=1$  e  $B=1$   $\Rightarrow A \cdot B$

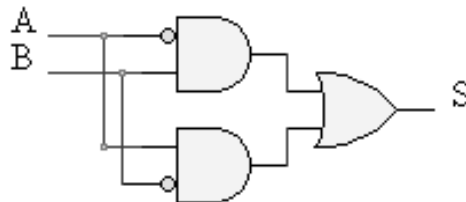
$$\text{Logo } S = \bar{A} \cdot \bar{B} + A \cdot \bar{B} + A \cdot B$$

## Outras Funções Lógicas:

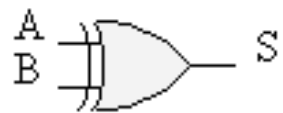
**Bloco OU EXCLUSIVO ou EXCLUSIVE OR ou XOR:** Executa a operação lógica OU EXCLUSIVO ( $\underline{\vee}$ ) na qual circulará corrente (estado "1") quando os valores de entrada forem diferentes. É representada logicamente por  $A \underline{\vee} B$ . A notação é  $A \oplus B$  que se lê "ou A ou B". Da tabela verdade abaixo obtêm a expressão que traduz o XOR através das portas AND e OR e inversão já vistas:  $A \oplus B = \bar{A} \cdot B + \bar{B} \cdot A$

A	B	$S=A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Tabela Verdade Ou Exclusivo (XOR)



Circuito para obtenção do XOR



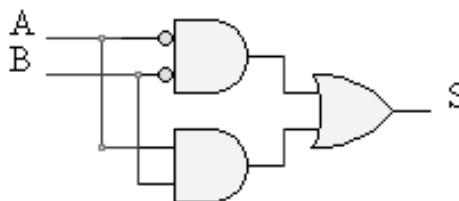
Porta Lógica XOR

Obs.: ao contrário das demais portas, o XOR só pode ter duas entradas, devido sua definição básica.

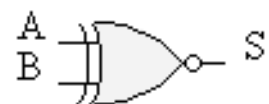
**Bloco COINCIDÊNCIA ou EXCLUSIVE NOR ou XNOR:** Executa a operação lógica BICONDICIONAL ( $\leftrightarrow$ ) na qual circulará corrente (estado "1") quando os valores de entrada forem iguais. A notação é  $A \odot B$  que se lê "A coincidência B" ou "A se, e somente se, B". Da tabela verdade abaixo obtêm a expressão que traduz o XNOR através das portas AND e OR e inversão já vistas:  $A \odot B = \bar{A} \cdot \bar{B} + A \cdot B$

A	B	$S=A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1

Tabela Verdade Coincidência (XNOR)



Circuito para obtenção do XNOR



Porta Lógica XNOR

Obs.: analogamente ao XOR, o XNOR só pode ter duas entradas.

**\*ORDEM DE PRIORIDADE:** Adota-se na eletrônica a seguinte ordem natural de prioridade: NOT, AND, XOR, OR, XNOR. Lembrando que a ordem natural é afetada pela presença de parênteses.