Aula 7: Portas Lógicas: AND, OR, NOT e XOR

Conforme foi discutido na última aula, cada operação lógica possui sua própria tabela verdade. A seguir será apresentado o conjunto básico de portas lógicas e suas respectivas tabelas verdade.

Porta AND

Trata-se de uma operação que aceita dois operandos ou duas entradas (A e B), conforme mostra a Figura 1. Os operandos são binários simples (0 e 1).

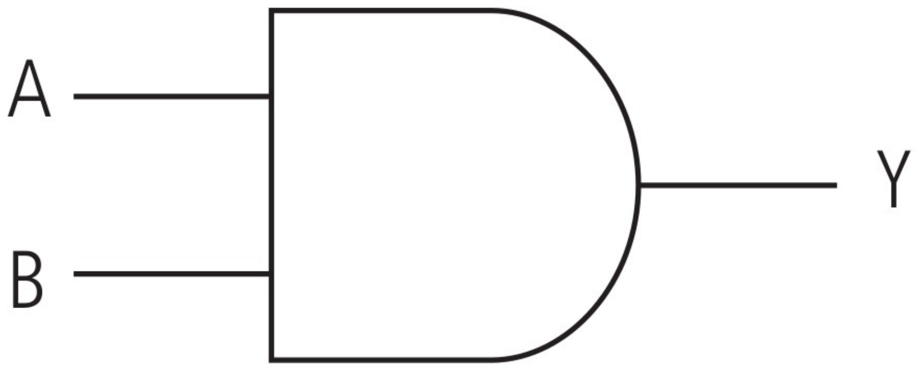


Figura 1: Porta Lógica AND (E).

Pode-se dizer que a operação AND simula uma multiplicação binária, permitindo os possíveis resultados conforme mostra a Tabela 1.

Entrada		Saída
A	**B**	**X = A.B**
0	0	0
0	1	0
1	0	0
1	1	1

Tabela 1: Tabela verdade da porta lógica AND.

Conforme é possível observar, a regra é: “se o primeiro operando é 1 e o segundo operando é 1, o resultado é 1 (Verdadeiro), senão o resultado é 0 (Falso)”.

Um exemplo de aplicação de uma porta AND seria na composição de um circuito para a transferência de bits de dados de um local para outro (ex.: da memória para a CPU). Nesse caso, a finalidade seria a de garantir que um bit de origem seja o mesmo bit de destino.

Porta OR

Trata-se de uma operação que aceita dois operandos ou duas entradas (A e B), conforme mostra a Figura 2. Os operandos são binários simples (0 e 1).

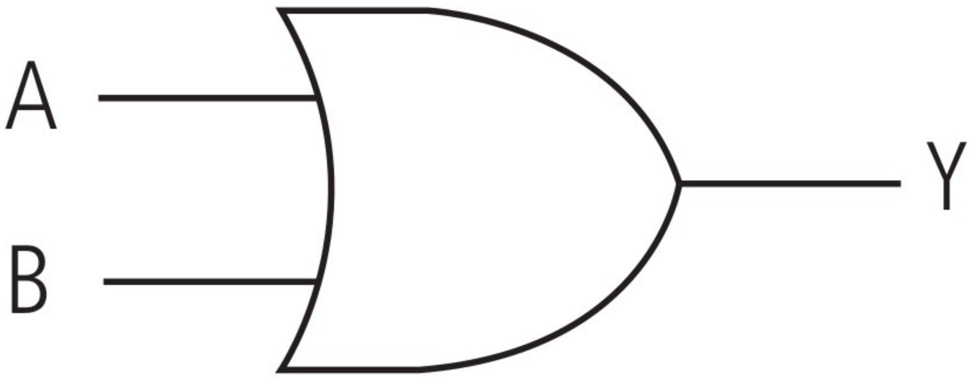


Figura 2: Porta lógica OR (OU).

Podemos dizer que a operação OR simula uma soma de binários, permitindo os possíveis resultados conforme mostra a Tabela 2.

Entrada		Saída
A	**B**	**X = A+B**
0	0	0
0	1	1
1	0	1
1	1	1

Tabela 2: Tabela verdade da porta lógica OR.

Conforme podemos observar, a regra é: “se o primeiro operando é 1 ou o segundo operando é 1, ou se os dois operandos forem 1, o resultado é 1, senão o resultado é 0”.

As operações lógicas OR são muito utilizadas em lógica digital ou mesmo em comandos de decisão de algumas linguagens de programação (ex.: Se (X=1 OU Y=1) Então Executa uma ação).

Porta NOT

A porta NOT representa um inversor. Essa operação aceita apenas um operando ou uma entradas (A), conforme mostra a Figura 3. O operando pode ser um dígito binário (0 ou 1).

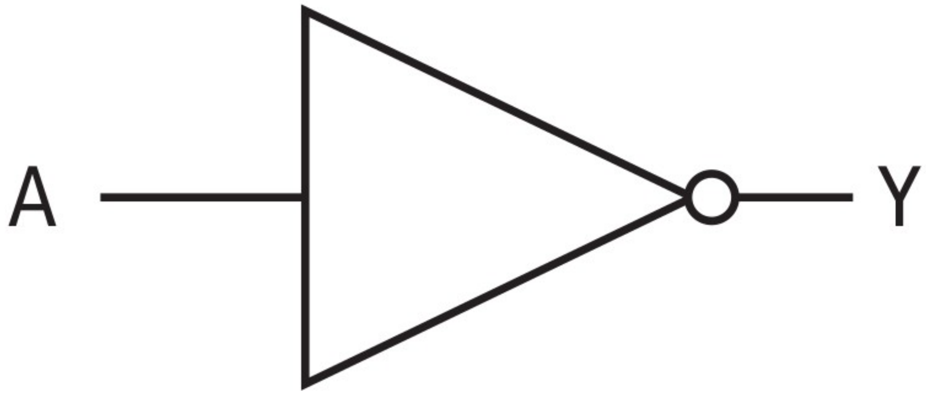


Figura 3: Porta Lógica NOT (NÃO).

Pode-se dizer que a operação NOT realiza a inversão de um dígito binário, permitindo os possíveis resultados conforme mostra a Tabela 3.

Entrada	Saída
A	**$X = \bar{A}$**
0	1
1	0

Tabela 3: Tabela verdade da porta lógica NOT.

Conforme é possível observar, a regra é: “se o operando for 1, o resultado é 0, senão o resultado é 1”.

Entre as principais aplicações dos circuitos inversores está a representação de números negativos no formato binário, quando se usa o que é chamado de complemento a 1 ou complemento a 2, fazendo-se necessária a inversão de um grupo de bits representativos de um número negativo.

Porta Lógica XOR

A denominação XOR é a abreviação do termo EXCLUSIVE OR. Trata-se de uma operação que aceita dois operandos ou duas entradas (A e B), conforme mostra a Figura 4. Os operandos são binários simples (0 e 1).

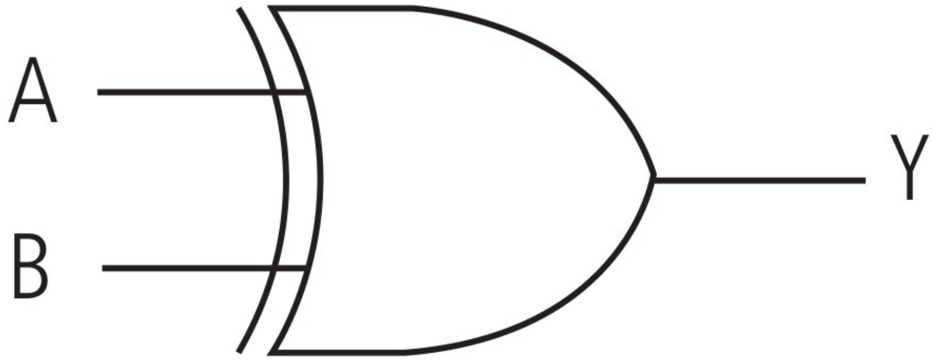


Figura 4: Porta lógica XOR (OU EXCLUSIVO).

Pode-se dizer que a operação XOR possui como principal função a verificação de igualdade, permitindo os possíveis resultados conforme mostra a Tabela 4.

Entrada		Saída
A	**B**	**$X = A \oplus B$**
0	0	0
0	1	1
1	0	1
1	1	0

Tabela 4: Tabela verdade da porta lógica XOR.

Conforme é possível observar, a regra é: “se o primeiro operando ou o segundo operando for igual a 1, o resultado é 1; senão, o resultado é 0”. Ou seja, para entradas iguais a saída será 0 e para entradas diferentes a saída será 1.

A porta XOR possui inúmeras aplicações, sendo um elemento lógico bastante versátil, permitindo, por exemplo, a fabricação de um testador de igualdade entre valores, para testar, de modo rápido, se duas palavras de dados são iguais.

Outras portas lógicas e circuitos combinatórios

Existem outras portas lógicas derivadas das portas lógicas apresentadas acima, tais como as portas NAND (porta AND invertida) e a porta NOR (porta OR invertida). Ambas são apresentadas nas Figuras 5 e 6, respectivamente.

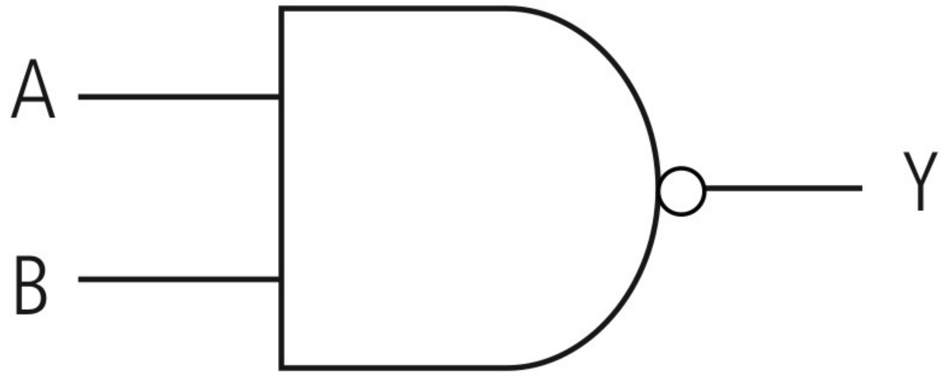


Figura 5: Porta lógica NAND (NÃO E).

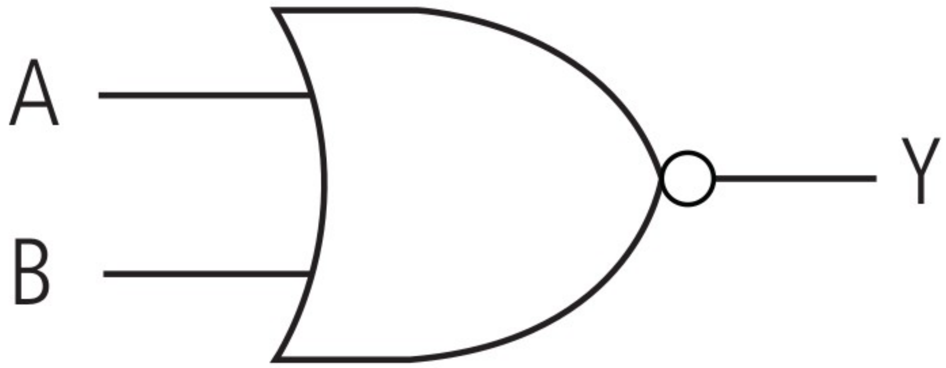


Figura 6: Porta lógica NOR (NÃO OU).

As operações lógicas são realizadas em dois passos: primeiro a operação AND ou OR e, em seguida, o seu resultado é invertido. Esse tipo de portas lógicas também possui diversas aplicações, sendo utilizado para reduzir a complexidade e a quantidade de portas lógicas necessárias a um determinado circuito lógico.

Não esqueça de que um circuito lógico pode possuir diversas portas lógicas e, portanto, suas tabelas verdade poderão ter inúmeras entradas e inúmeras saídas (as quais poderão ser representadas por suas respectivas equações booleanas). A Tabela 5 mostra o resumo dos símbolos gráficos e matemáticos (equação booleana) de portas lógicas.

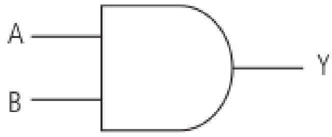
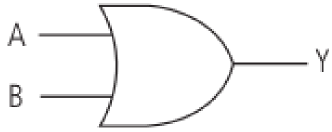
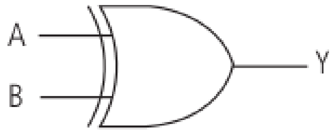
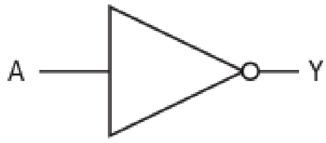
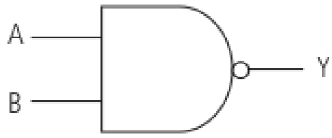
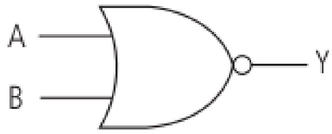
Função Lógica Básica	Símbolo Gráfico da Porta	Equação Booleana
AND		$Y = A.B$
OR		$Y = A+B$
XOR		$Y = A \oplus B$
NOT		$Y = \bar{A}$
NAND		$Y = \overline{A.B}$
NOR		$Y = \overline{A+B}$

Tabela 5: Símbolos gráficos e equações booleanas de portas lógicas.

A partir das portas lógicas básicas, é possível interligar diversas de suas unidades, de modo a construir redes lógicas, também chamadas de circuitos combinatórios. Um circuito combinatório é definido como um conjunto de portas lógicas cuja saída em qualquer instante de tempo é função somente das entradas.

Existe outra categoria de circuitos que combina portas lógicas, denominada circuitos sequenciais, os quais, além de possuir portas, contêm elementos de armazenamento (uma espécie de memória). A Figura 7 exemplifica um circuito combinatório implementado para uma determinada função.

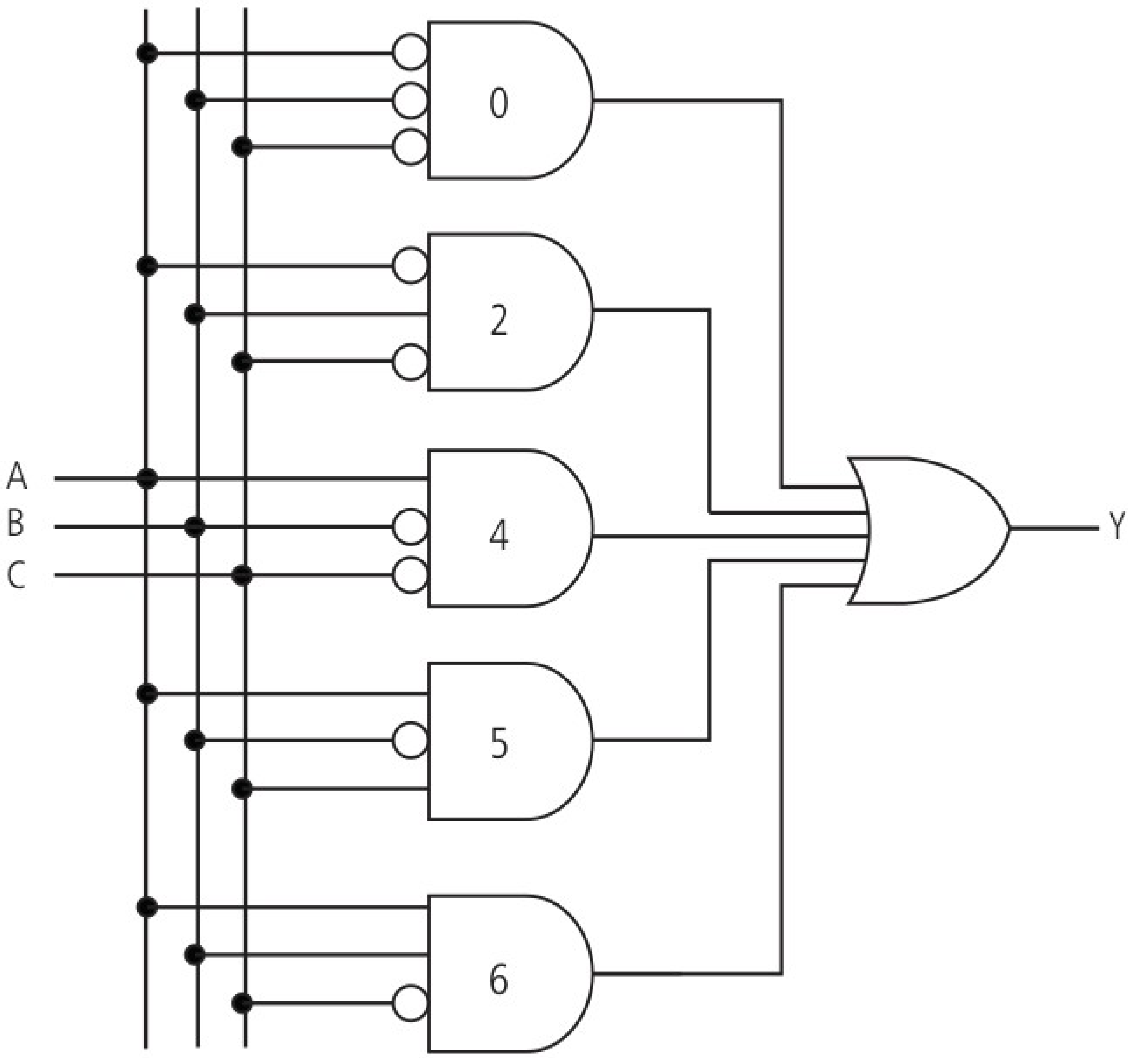


Figura 7: Circuito combinatório.

Dentre os exemplos de circuitos combinatórios, destacamos o das funções essenciais da maioria dos computadores e sistemas digitais, ou seja, a realização de operações aritméticas tais como: adição, subtração, multiplicação e divisão. Essas operações são realizadas na Unidade Lógica Aritmética (ULA) destes sistemas digitais, onde uma série de portas lógicas são combinadas para adicionar, subtrair, multiplicar ou dividir números binários. No caso das operações de multiplicação e divisão, além das portas lógicas, há a necessidade de circuitos sequenciais.

Exercícios

1. Desenhe os circuitos que implementam as seguintes expressões booleanas.

a) $S = \overline{A\overline{B} + \overline{CD}}$

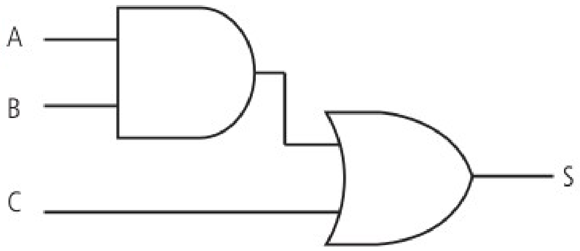
b) $S = \overline{(AB\overline{C} + \overline{CD}) \oplus D}$

c) $S = A\ B + (CD\ E)$

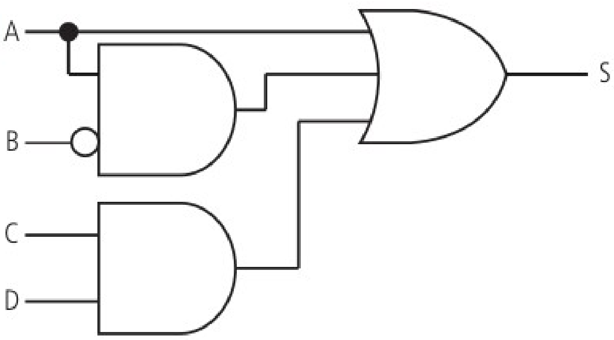
d) $S = A + (B + C\ D)(B + A)$

2. Escreva a expressão booleana executada pelos circuitos abaixo:

a)



b)



c)