Introdução aos circuitos lógicos elementares

Tópicos

- Introdução à álgebra de Boole
- Introdução à ferramenta DesignWorks
- Níveis lógicos
- Portas lógicas elementares
- Desenho e simulação de circuitos lógicos simples com DesignWorks

Definições

Circuitos lógicos podem ser classificados em dois tipos: circuitos *combinatórios* e circuitos *sequenciais*. Num circuito lógico *combinatório* as saídas dependem unicamente dos valores atuais das entradas.

O dispositivo lógico mais básico chama-se *porta lógica* (*gate*, em inglês). Uma porta lógica tem uma ou mais entradas e produz uma saída que é a função dos valores atuais da(s) entrada(s). Os valores das entradas e saídas são quantidades analógicas (tais como voltagem, corrente, etc.). Entretanto, estas quantidades podem ser modeladas por dois valores discretos: 0 e 1.

Os símbolos gráficos das portas lógicas mais conhecidas estão ilustrados na Figura 1.

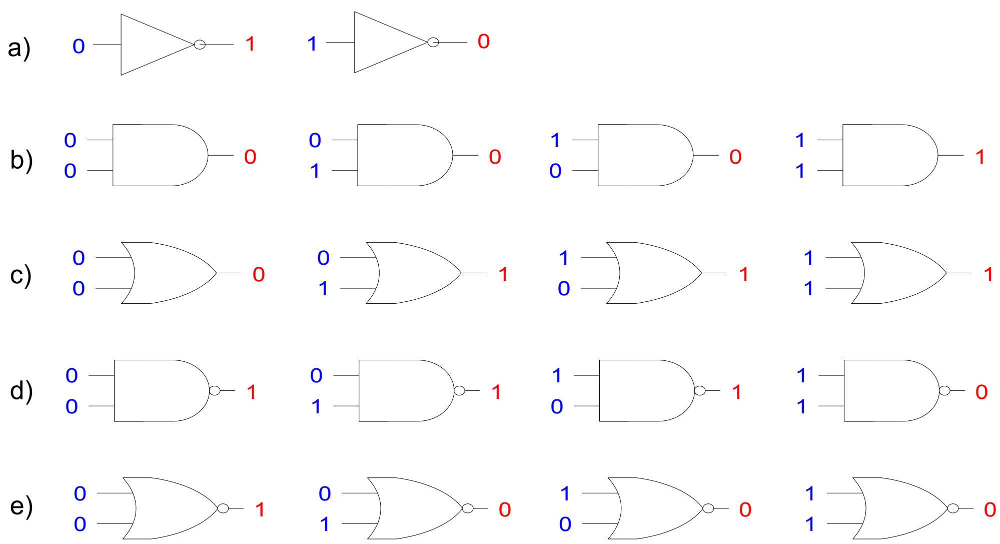

Figura 1. Portas lógicas: a) **NOT**, b) **AND**, c) **OR**, d) **NAND**, e) **NOR**.

Todas as combinações possíveis dos valores de entradas e as saídas respetivas estão ilustradas na Figura 1. A porta lógica **NOT** (inversor) produz na saída o valor que é o valor inverso da entrada. A porta lógica **AND** produz na saída o valor 1 só quando todas as entradas estão a 1, e gera 0 em todos os casos restantes. A porta lógica **OR** produz na saída o valor 0 só quando todas as entradas estão a 0, e gera 1 em todos os casos restantes. A porta **NAND**, em termos lógicos, é equivalente a uma **AND** seguida dum inversor. A porta **NOR**, em termos lógicos, é equivalente a uma **OR** seguida dum inversor.

Com as portas lógicas **AND**, **OR** e **NOT** é possível implementar qualquer função lógica. Para descrever funções lógicas recorreremos à álgebra de Boole.

Na **álgebra de Boole** são usadas variáveis simbólicas (x, y, etc.) para representar valores de sinais lógicos, modelando os com apenas dois valores possíveis: 0 e 1. O traço por cima duma variável ($\bar{x}$) é um operador algébrico que descreve a função do inversor (da porta **NOT**), o · é o operador de multiplicação (conjunção) lógica que descreve a função da porta **AND**, e + é o operador de soma (disjunção) lógica que descreve a função da porta **OR**.

A maneira mais simples de representar uma função booleana é através de uma *tabela de verdade*. Nesta tabela são indicados os valores da função para todas as combinações possíveis de entradas. Por exemplo, podemos descrever as funções lógicas de portas **NOT**, **OR** e **AND** de duas entradas com as tabelas seguintes:

x	*y*	*f*
0	0	0
0	1	0
1	0	0
1	1	1

x	*y*	*g*
0	0	0
0	1	1
1	0	1
1	1	1

x	*h*
0	1
1	0

Ou, utilizando expressões algébricas: $f = x.y, g = x + y, h = x'$

Uma das maneiras de deduzir a expressão algébrica (na forma de soma de produtos) a partir da tabela de verdade é a seguinte:

a) identificar todas as linhas (combinações de entradas) que produzem 1 na saída;

b) para cada uma destas combinações criar um produto que será a multiplicação lógica das variáveis de entrada respetivas (invertidas se nesta linha da tabela de verdade estão a 0, ou não invertidas, caso estejam a 1);

c) criar uma soma lógica de todos os produtos obtidos na alínea anterior.

Exemplo: deduza a função booleana representada com a tabela de verdade seguinte:

x	*y*	*f*
0	0	0
0	1	0
1	0	1
1	1	1

$f(x, y) = xy' + xy$

Exercícios

1 Analise e repita exemplos 1 e 2 do *tutorial* de introdução à ferramenta DesignWorks. O *tutorial* está disponível na página da disciplina.

2 Construa as tabelas de verdade que descrevem o comportamento das portas NAND e NOR de duas entradas. Simule o comportamento destas duas portas lógicas com a ferramenta DesignWorks e verifique se corresponde às tabelas derivadas.

3 Determine o resultado das operações lógicas seguintes:

a)	11110000	OR	01011011
b)	11110000	AND	01011011
c)	11110000	NAND	01011011
d)	11110000	NOR	01011011

4 Determine o resultado das expressões seguintes quando $x = 1, y = 0, z = 0$

a) $(xyz + xyz' + x'y'z')'$
b) $[(x + y + z).(x + y + z').(x' + y' + z')]'$

5 Considere a seguinte função booleana: $f(a,b,c) = a'b'c + a'bc' + ab'c' + abc$

a) Construa a tabela de verdade que define $f(a,b,c)$
b) Construa no DesignWorks o circuito que implementa a função $f(a,b,c)$
c) Simule o circuito e certifique-se que funciona de acordo com a especificação

6 Deduza a função booleana representada com a tabela de verdade seguinte:

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

a) Construa no DesignWorks o circuito que implementa a função $f(x_1, x_2, x_3)$
b) Simule o circuito e certifique-se que funciona de acordo com a especificação

7 Analise e repita o exemplo 3 do *tutorial* de introdução ao DesignWorks (disponível na página da disciplina).