Lógica Digital (1001351)

Funções e Circuitos Lógicos

Prof. Edilson Kato
kato@ufscar.br

Prof. Maurício Figueiredo
mauricio@ufscar.br

Prof. Ricardo Menotti
menotti@ufscar.br

Prof. Roberto Inoue
rsinoue@ufscar.br

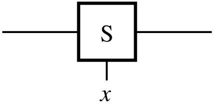
Departamento de Computação
Universidade Federal de São Carlos

Atualizado em: 27 de fevereiro de 2019

Departamento de Computação

Uma variável binária

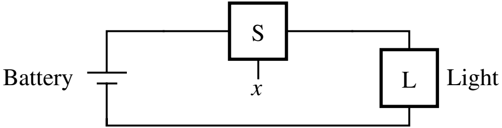
(a) Two states of a switch



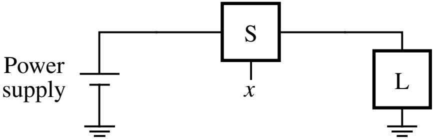
(b) Symbol for a switch

Figure 2.1 A binary switch.

Uma variável binária



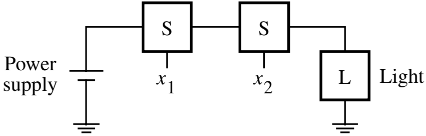
(a) Simple connection to a battery



(b) Using a ground connection as the return path

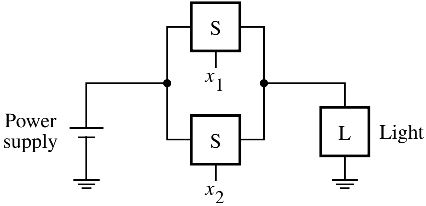
Figure 2.2 A light controlled by a switch.

Funções lógicas E (série) e OU (paralelo)



(a) The logical AND function (series connection)

$L(x_1, x_2) = x_1.x_2$
onde
$L = 1$ se $x_1 = 1$ **E** $x_2 = 1$,
$L = 0$ *caso contrário.*



(b) The logical OR function (parallel connection)

$L(x_1, x_2) = x_1 + x_2$
onde
$L = 1$ se $x_1 = 1$ **OU**
$x_2 = 1$ **OU** $x_1 = x_2 = 1$,
$L = 0$ se $x_1 = x_2 = 0$.

Figure 2.3 Two basic functions.

Combinando as funções

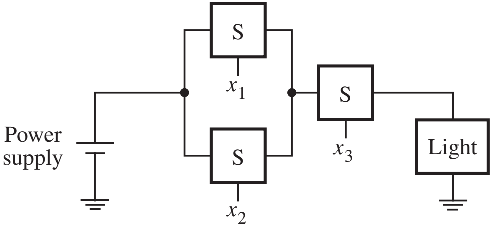
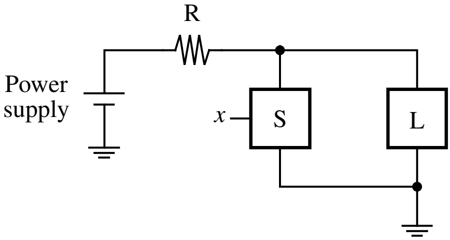


Figure 2.4 A series-parallel connection.

$$L(x_1, x_2, x_3) = (x_1 + x_2).x_3$$

Função lógica NÃO (inversão ou complemento)



$L(x) = \overline{x}$
onde
$L = 1$ se $x = 0$,
$L = 0$ se $x = 1$.

Figure 2.5 An inverting circuit.

representações possíveis: $\overline{x} = x' = !x = \sim x =$ NOTx

para $f(x_1, x_2) = x_1 + x_2$, temos seu complemento

$$\overline{f}(x_1, x_2) = \overline{x_1 + x_2}$$

$$= (x_1 + x_2)' = !(x_1 + x_2) = \sim (x_1 + x_2) = \text{NOT}(x_1 + x_2)$$

Tabela Verdade

x_1	x_2	$x_1.x_2$	$x_1 + x_2$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Tabela Verdade para as funções lógicas AND e OR

Lógica Proposicional

proposições		*negação*	*conjunção*	*disjunção*	*implicação*	*equivalência*
p	q	$\sim p$	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \Leftrightarrow q$
0	0		0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1		1	1	1	1

Tabela Verdade para os conectivos lógicos

Tabela Verdade

x_1	x_2	x_3	$x_1.x_2.x_3$	$x_1 + x_2 + x_3$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Funções lógicas AND e OR com três entradas

Tabela Verdade

x_1	x_2	$x_1 + x_2$	$\overline{x_1 + x_2}$	$\overline{x_1}$	$\overline{x_2}$	$\overline{x_1} + \overline{x_2}$
0	0	0	1			
0	1	1	0			
1	0	1	0			
1	1	1	0			

Provando que $\overline{f}(x_1, x_2) = \overline{x_1 + x_2} \neq \overline{x_1} + \overline{x_2}$

Tabela Verdade

x_1	x_2	$x_1 + x_2$	$\overline{x_1 + x_2}$	$\overline{x_1}$	$\overline{x_2}$	$\overline{x_1} + \overline{x_2}$
0	0	0	1	1	1	
0	1	1	0	1	0	
1	0	1	0	0	1	
1	1	1	0	0	0	

Provando que $\overline{f}(x_1, x_2) = \overline{x_1 + x_2} \neq \overline{x_1} + \overline{x_2}$

Tabela Verdade

x_1	x_2	$x_1 + x_2$	$\overline{x_1 + x_2}$	$\overline{x_1}$	$\overline{x_2}$	$\overline{x_1} + \overline{x_2}$
0	0	0	1	1	1	1
0	1	1	0	1	0	1
1	0	1	0	0	1	1
1	1	1	0	0	0	0

Provando que $\overline{f}(x_1, x_2) = \overline{x_1 + x_2} \neq \overline{x_1} + \overline{x_2}$

Portas Lógicas e Circuitos Lógicos

- As funções lógicas AND, OR e NOT podem ser usadas para implementar funções lógicas de qualquer complexidade;

Portas Lógicas e Circuitos Lógicos

- As funções lógicas AND, OR e NOT podem ser usadas para implementar funções lógicas de qualquer complexidade;
- Uma função complexa pode exigir muitas dessas operações básicas para sua implementação;

Portas Lógicas e Circuitos Lógicos

- As funções lógicas AND, OR e NOT podem ser usadas para implementar funções lógicas de qualquer complexidade;
- Uma função complexa pode exigir muitas dessas operações básicas para sua implementação;
- Cada operação lógica pode ser implementada eletronicamente com transistores, resultando em um elemento de circuito chamado de porta lógica;

Portas Lógicas e Circuitos Lógicos

- As funções lógicas AND, OR e NOT podem ser usadas para implementar funções lógicas de qualquer complexidade;
- Uma função complexa pode exigir muitas dessas operações básicas para sua implementação;
- Cada operação lógica pode ser implementada eletronicamente com transistores, resultando em um elemento de circuito chamado de porta lógica;
- Uma porta lógica tem uma ou mais entradas e uma saída que é uma função de suas entradas;

Portas Lógicas e Circuitos Lógicos

- As funções lógicas AND, OR e NOT podem ser usadas para implementar funções lógicas de qualquer complexidade;
- Uma função complexa pode exigir muitas dessas operações básicas para sua implementação;
- Cada operação lógica pode ser implementada eletronicamente com transistores, resultando em um elemento de circuito chamado de porta lógica;
- Uma porta lógica tem uma ou mais entradas e uma saída que é uma função de suas entradas;
- Podemos projetar um circuito lógico desenhando um esquemático, consistindo de símbolos gráficos representando as portas lógicas.

Portas Lógicas e Circuitos Lógicos

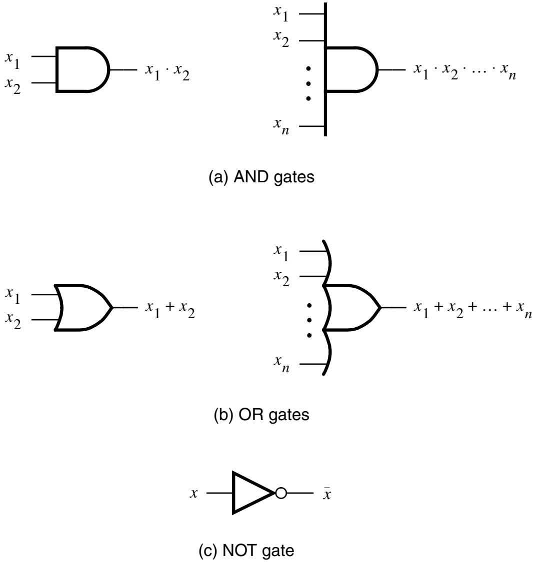


Figure 2.8 The basic gates.

Portas Lógicas e Circuitos Lógicos

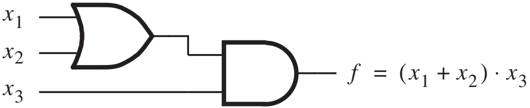
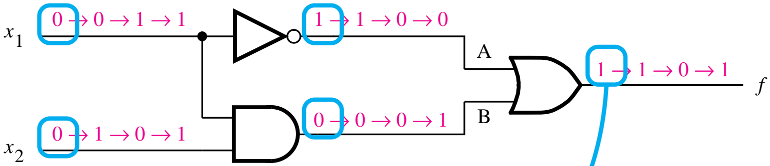


Figure 2.9 The function from Figure 2.4.

Análise de um circuito lógico

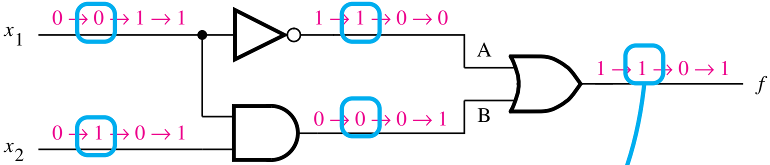


(a) Network that implements $f = \bar{x}_1 + x_1 \cdot x_2$

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	1
1	0	0
1	1	1

(b) Truth table for f

Análise de um circuito lógico

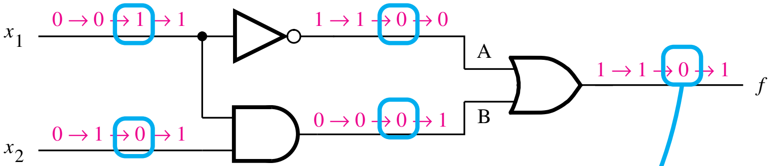


(a) Network that implements $f = \bar{x}_1 + x_1 \cdot x_2$

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	1
1	0	0
1	1	1

(b) Truth table for f

Análise de um circuito lógico

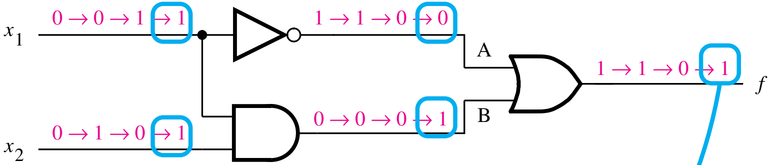


(a) Network that implements $f = \bar{x}_1 + x_1 \cdot x_2$

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	1
1	0	$\boxed{0}$
1	1	1

(b) Truth table for f

Análise de um circuito lógico

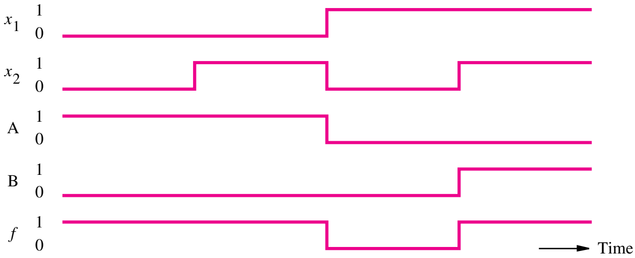


(a) Network that implements $f = \bar{x}_1 + x_1 \cdot x_2$

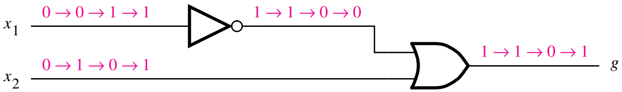
x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	1
1	0	0
1	1	$\boxed{1}$

(b) Truth table for f

Síntese de uma função lógica



(c) Timing diagram

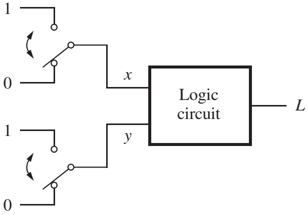


(d) Network that implements $g = \bar{x}_1 + x_2$

Funções lógicas equivalentes

- Em geral, uma função lógica pode ser implementada com uma variedade de circuitos com diferentes custos;
- As funções lógicas vistas anteriormente são funcionalmente equivalentes;
- É possível notar a equivalência a partir da análise dos circuitos e contrução das tabelas verdade;
- 0 mesmo resultado pode ser alcançado através da manipulação algébrica de expressões lógicas, que fornece a base para técnicas modernas de projeto.

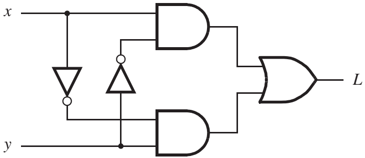
Função XOR



(a) Two switches that control a light

x	y	L
0	0	0
0	1	1
1	0	1
1	1	0

(b) Truth table



(c) Logic network

(d) XOR gate symbol

Figure 2.11 An example of a logic circuit.

$$L = f(x, y) = x \oplus y$$

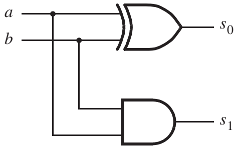
Aplicação: meio somador

a	0	0	1	1
$+\,b$	$+\,0$	$+\,1$	$+\,0$	$+\,1$
$s_1\,s_0$	0 0	0 1	0 1	1 0

(a) Evaluation of $S = a + b$

a	b	s_1	s_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

(b) Truth table



(c) Logic network

Figure 2.12 Addition of binary numbers.

$$S_0 = f(a,b) = a \oplus b$$
$$S_1 = f(a,b) = a.b$$

Bibliografia

- Brown, S. & Vranesic, Z. - Fundamentals of Digital Logic with Verilog Design, 3rd Ed., Mc Graw Hill, 2009
- http://ecalculo.if.usp.br/ferramentas/logica/logica.htm

Lógica Digital (1001351)

Funções e Circuitos Lógicos

Prof. Edilson Kato
kato@ufscar.br

Prof. Maurício Figueiredo
mauricio@ufscar.br

Prof. Ricardo Menotti
menotti@ufscar.br

Prof. Roberto Inoue
rsinoue@ufscar.br

Departamento de Computação
Universidade Federal de São Carlos

Atualizado em: 27 de fevereiro de 2019