

# **Relatório de Sistemas Digitais L1**

## **Funções Combinatórias**

João Oliveira  
Tomás A. Reis

Instituto Superior Técnico  
Universidade de Lisboa

21 de Março de 2014  
Quinta-Feira LSD1

# 1 Introdução

O objectivo deste trabalho é concepção de um circuito que recebendo um número  $A$  no intervalo  $[0;3]$  realiza sobre ele uma operação indicada por um número  $B$  também no intervalo  $[0;3]$  segundo o seguinte código:

B	Operação
0	Deslocamento à direita
1	Deslocamento à esquerda
2	Operação indefinida
3	Identidade

Tabela 1: Operações codificadas

Retorna assim um número  $S$  também no intervalo  $[0;3]$  e um *bit*  $C_0$  (“*Carry Out*”) que representa o dígito que é perdido no deslocamento de  $A$ .

Na operação “Deslocamento à direita” cada *bit* é movido para a posição à sua direita. O *bit* mais à direita é assim “perdido” sendo o retorno de  $C_0$ , já o *bit* mais à esquerda passa a 0. Neste caso, isto resum-se a que  $S_0$  é igual a  $A_1$ ,  $C_0$  a  $A_0$  e  $S_1$  é 0.

A operação “Deslocamento à esquerda” é semelhante, sendo que em cada *bit* é movido para a posição à sua esquerda. O *bit* mais à esquerda é assim “perdido” sendo o retorno de  $C_0$ , já o *bit* mais à direita passa a 0. Neste caso, isto resum-se a que  $S_1$  é igual a  $A_0$ ,  $C_0$  a  $A_1$  e  $S_0$  é 0.

A “Operação indefinida” não tem interesse, e como tal o seu retorno é escolhido consoante o mais útil à economização de portas lógicas. A operação “Identidade” retorna o mesmo número, sendo que, neste caso, o “*Carry Out*” não faz sentido, visto não existir deslocamento. Considerámos assim que nesta situação o retorno de  $C_0$  é uma indeterminação (“X”). Para esta operação, então,  $S_1$  será idêntico a  $S_1$  e  $S_0$  idêntico a  $A_0$ .

Tem-se também como alvo conseguir este circuito utilizando o mínimo de recursos.

## 2 Projecto

### 2.1 Entradas e Saídas

Estando  $A$ ,  $B$  e  $S$  no intervalo  $[0;3]$  cada um será representado por dois bits, enquanto  $C_0$  apenas necessitará de um. A codificação de  $A, B$  e  $S$  seguirão a conversão habitual de binário para decimal, como apresentado na seguintes tabela:

A <sub>0</sub>	A <sub>1</sub>	Valor de A	B <sub>0</sub>	B <sub>1</sub>	Valor de B	S <sub>0</sub>	S <sub>1</sub>	Valor de S
0	0	0	0	0	0	0	0	0
0	1	1	0	1	1	0	1	1
1	0	2	1	0	2	1	0	2
1	1	3	1	1	3	1	1	3

Tabela 2: Codificação das entradas

## 2.2 Tabela de verdade

A <sub>1</sub>	A <sub>0</sub>	B <sub>1</sub>	B <sub>0</sub>	S <sub>1</sub>	S <sub>0</sub>	C <sub>0</sub>
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	X	X	X
0	0	1	1	0	0	X
0	1	0	0	0	0	1
0	1	0	1	1	0	0
0	1	1	0	X	X	X
0	1	1	1	0	1	X
1	0	0	0	0	1	0
1	0	0	1	0	0	1
1	0	1	0	X	X	X
1	0	1	1	1	0	X
1	1	0	0	0	1	1
1	1	0	1	1	0	1
1	1	1	0	X	X	X
1	1	1	1	1	1	X

Tabela 3: Tabela de Verdade

## 2.3 Simplificação das funções algébricas

Segundo os quadros de Karnaugh apresentados (Figura 1.) com os implicantes assinalados, podemos exprimir as funções na forma de soma de produtos como:

$$S_1(A_1, A_0, B_1, B_0) = A_0 \overline{B_1} B_0 + A_1 B_1 \quad (1)$$

$$S_0(A_1, A_0, B_1, B_0) = A_1 \overline{B_0} B_1 + A_0 B_1 \quad (2)$$

$$C_0(A_1, A_0, B_1, B_0) = A_0 \overline{B_0} + A_1 B_0 \quad (3)$$

Figura 1: Quadros de Karnaugh

Função S1

$\begin{smallmatrix} B_1, B_0 \\ A_1, A_0 \end{smallmatrix}$	00	01	11	10
00	0	0	0	X
01	0	1	0	X
11	0	1	1	X
10	0	0	1	X

$\begin{smallmatrix} B_1, B_0 \\ A_1, A_0 \end{smallmatrix}$	00	01	11	10
00	0	0	0	X
01	0	1	0	X
11	0	1	1	X
10	0	0	1	X

Função S0

$\begin{smallmatrix} B_1, B_0 \\ A_1, A_0 \end{smallmatrix}$	00	01	11	10
00	0	0	0	X
01	0	0	1	X
11	1	0	1	X
10	1	0	0	X

$\begin{smallmatrix} B_1, B_0 \\ A_1, A_0 \end{smallmatrix}$	00	01	11	10
00	0	0	0	X
01	0	0	1	X
11	1	0	1	X
10	1	0	0	X

Função C0

$\begin{smallmatrix} B_1, B_0 \\ A_1, A_0 \end{smallmatrix}$	00	01	11	10
00	0	0	X	X
01	1	0	X	X
11	1	1	X	X
10	0	1	X	X

$\begin{smallmatrix} B_1, B_0 \\ A_1, A_0 \end{smallmatrix}$	00	01	11	10
00	0	0	X	X
01	1	0	X	X
11	1	1	X	X
10	0	1	X	X

Alternativamente, podemos expressá-las como produto de somas tendo em conta os quadros com implicados marcados, obtendo:

$$S_1(A_1, A_0, B_1, B_0) = (B_0)(A_1 + \overline{B_1})(A_0 + B_1) \quad (4)$$

$$S_0(A_1, A_0, B_1, B_0) = (B_1 + \overline{B_0})(A_1 + \overline{B_0})(A_1 + B_0) \quad (5)$$

$$C_0(A_1, A_0, B_1, B_0) = (A_0 + B_0)(A_1 + \overline{B_0}) \quad (6)$$

## 2.4 Funções a construir

Sendo os nossos números 79144 e 78811 respetivamente então:

$$78811 + 79144 = 157955 \quad (7)$$

$$157955/3 = 52651 + 2/3 \quad (8)$$

Sendo o último algarismo de um número inteiro em base três dado pelo resto da primeira divisão desse número por três, podemos ver na operação aritmética apresentada em (8) que o algarismo menos significativo da soma dos nossos números é 2. Logo, apenas realizaremos a função  $S_1$  e  $C_0$ .

## 2.5 Transformação das expressões algébricas

### 2.5.1 De forma a serem concretizadas com portas NAND-2, NAND-3 e NOT

1. A partir da forma disjuntiva

$$\begin{aligned} S_1(A_1, A_0, B_1, B_0) &= A_0 \overline{B_1} B_0 + A_1 B_1 \\ &= \overline{\overline{(A_0 \overline{B_1} B_0)} \overline{(A_1 B_1)}} \end{aligned} \quad (9)$$

$$\begin{aligned} C_0(A_1, A_0, B_1, B_0) &= A_0 \overline{B_0} + A_1 B_0 \\ &= \overline{\overline{(A_0 \overline{B_0})} \overline{(A_1 B_0)}} \end{aligned} \quad (10)$$

Requisitos de implementação:

- (a) 1x NAND-3
- (b) 5x NAND-2
- (c) 2x NOT

2. A partir da forma conjutiva

$$\begin{aligned} S_1(A_1, A_0, B_1, B_0) &= B_0 (A_0 + B_1) (\overline{B_1} + A_1) \\ &= \overline{\overline{\overline{B_0} \overline{(A_0 + B_1)} \overline{(B_1 + A_1)}}} \end{aligned} \quad (11)$$

$$\begin{aligned} C_0(A_1, A_0, B_1, B_0) &= (A_0 + B_0) (A_1 + \overline{B_0}) \\ &= \overline{\overline{\overline{(A_0 + B_0)} \overline{(A_1 + \overline{B_0})}}} \end{aligned} \quad (12)$$

Requisitos de implementação:

- (a) 1x NAND-3
- (b) 5x NAND-2
- (c) 6x NOT

### 2.5.2 De forma a serem concretizadas com portas NOR-2, NOR-3 e NOT

1. A partir da forma disjuntiva

$$\begin{aligned} S_1(A_1, A_0, B_1, B_0) &= A_0 \overline{B_1} B_0 + A_1 B_1 \\ &= \overline{\overline{A_0 + B_1 + B_0}} + \overline{\overline{A_1 + B_1}} \end{aligned} \quad (13)$$

$$\begin{aligned} C_0(A_1, A_0, B_1, B_0) &= A_0 \overline{B_0} + A_1 B_0 \\ &= \overline{\overline{A_0 + B_0}} + \overline{\overline{A_1 + B_0}} \end{aligned} \quad (14)$$

- (a) 1x NOR-3
- (b) 5x NOR-2
- (c) 6x NOT

2. A partir da forma conjutiva

$$\begin{aligned} S_1(A_1, A_0, B_1, B_0) &= (B_0)(A_1 + \overline{B_1})(A_0 + B_1) \\ &= \overline{\overline{B_0} + \overline{(A_1 + \overline{B_1})} + \overline{(A_0 + B_1)}} \end{aligned} \quad (15)$$

$$\begin{aligned} C_0(A_1, A_0, B_1, B_0) &= (A_0 + B_0)(A_1 + \overline{B_0}) \\ &= \overline{\overline{(A_0 + B_0)} + \overline{(A_1 + \overline{B_0})}} \end{aligned} \quad (16)$$

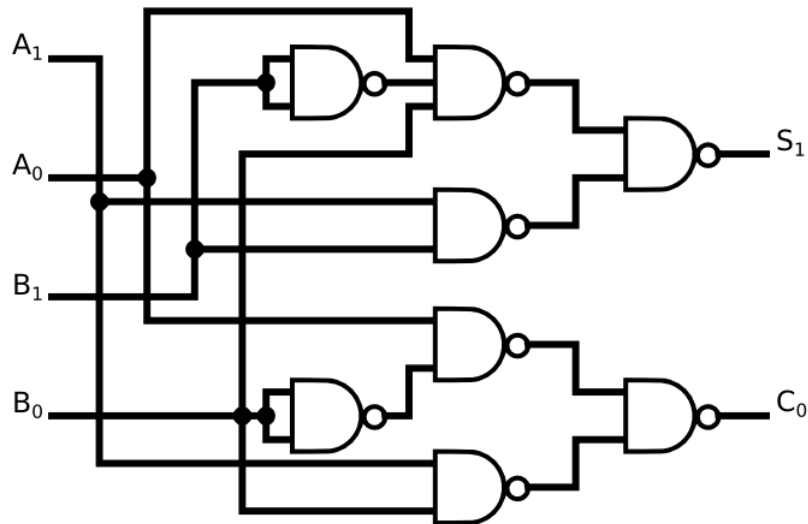
Requisitos de implementação:

- (a) 1x NOR-3
- (b) 5x NOR-2
- (c) 2x NOT

## 2.6 Diagrama Lógico

Das várias opções analisadas em cima, optámos pela implementação com portas NAND a partir da forma disjuntiva, necessitando apenas de duas portas NOT, cinco NAND-2 e um NAND-3. Seguiremos assim o diagrama apresentado na Figura 2.

Figura 2: Diagrama Lógico



## 2.7 Valor Lógico não especificado

Segundo as expressões optadas, na situação  $A=1$  e  $B=2$ , de valor não determinado, terão valores lógicos:

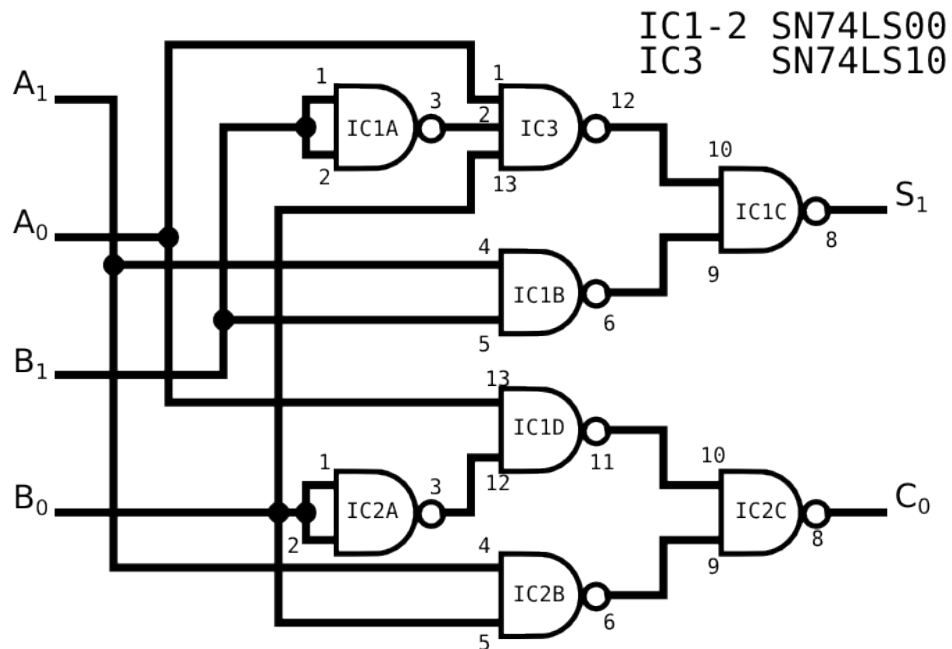
$$\begin{aligned} S_1(0, 1, 1, 0) &= 1 \cdot \bar{1} \cdot 0 + 0 \cdot 1 \\ &= 0 + 0 = 0 \end{aligned} \quad (17)$$

$$\begin{aligned} C_0(0, 1, 1, 0) &= 1 \cdot \bar{0} + 0 \cdot 0 \\ &= 1 \cdot 1 + 0 = 1 \end{aligned} \quad (18)$$

## 2.8 Esquema Eléctrico

Para a realização deste circuito segundo o logigrama previamente apresentado, serão necessários três circuitos integrados: um SN74LS10, com portas NAND-3, e dois SN74LS00. De forma a necessitarmos apenas destes três CIs, concretizaremos cada porta NOT juntando as entradas de uma porta NAND-2. Isto é necessário pois precisamos de 5 portas NAND-2. Caso utilizássemos portas NOT, a não ser que utilizássemos uma porta NAND-3 como NAND-2, utilizaríamos 4 CIs, pois precisaríamos de dois SN74LS00 de qualquer forma; porém considerámos mais simples a solução apresentada.

Figura 3: Esquema Eléctrico



### 3 Montagem e Teste

#### 3.1 Montagem

Montou-se o circuito na *breadboard* utilizando os circuitos requisitados.

#### 3.2 Utilização da Ponta de Prova



Valores de entrada				Valores Esperados		Valores de Saída	
$A_1$	$A_0$	$B_1$	$B_0$	$S_1$	$C_0$	$S_1$	$C_0$
0	0	0	0	L	L		
0	0	0	1	L	L		
0	0	1	0	X	X		
0	0	1	1	L	X		
0	1	0	0	L	H		
0	1	0	1	H	L		
0	1	1	0	X	X		
0	1	1	1	L	X		
1	0	0	0	L	L		
1	0	0	1	L	H		
1	0	1	0	X	X		
1	0	1	1	H	X		
1	1	0	0	L	H		
1	1	0	1	H	H		
1	1	1	0	X	X		
1	1	1	1	H	X		

Tabela 4: Tabela de Teste

### 3.3 Teste do circuito

#### 3.3.1 Comentário dos resultados

## 4 Conclusão

Com este trabalho teve-se como objectivo a concepção e concretização de um circuito que executa duas funções combinatórias. Para tal utilizou-se os Mapas de Karnaugh para obter as funções como soma de produtos e também como produto de somas para que depois fossem facilmente convertidas para expressões com, exclusivamente portas NAND e NOT ou NOR e NOT. A partir destas expressões escolhemos a mais económica e eficiente de implementar, tendo elaborado o respectivo diagrama lógico e esquema eléctrico após selecção dos circuitos integrados a usar.