

## **LABORATÓRIO 6 – SOMADORES**

EFRAIN MARCELO PULGAR PANTALEON<sup>1</sup>  
FERNANDO LUCAS SOUSA SILVA<sup>2</sup>  
MATHEUS GOMES DINIZ ANDRADE<sup>3</sup>  
TEÓFILO VITOR DE CARVALHO CLEMENTE<sup>4</sup>

Universidade Federal do Rio Grande do Norte, Departamento de Engenharia de Computação e Automação

### **1. INTRODUÇÃO**

O presente relatório tem por objetivo consolidar os conhecimentos adquiridos a respeito dos Somadores e do seu funcionamento, dessa forma faremos o desenvolvimento e análise do comportamento dos circuitos digitais por eles compostos. Assim, apresentaremos os resultados obtidos de acordo com cada item esclarecido no roteiro, de modo a mostrar os códigos usados para a emulação dos projetos e respectivos resultados adquiridos via o software Quartus.

### **2. METODOLOGIA**

Para a realização deste laboratório foi necessário o conhecimento teórico a respeito do funcionamento dos Somadores como também o conhecimento de registradores para a sua construção, tanto os incrementadores como os decrementadores. Desse modo, para a experimentação prática foi utilizado o software Quartus II, ele possibilita a construção de diversos circuitos digitais a partir do uso de portas lógicas, Latches, Flip-Flops e outros componentes que podem ser simulados com seu uso, com isso foi possível a construção dos códigos em VHDL utilizando a lógica dos códigos vista nas aulas teóricas.

### **3. RESULTADOS**

Nesta seção serão apresentados os resultados obtidos nas simulações realizadas, como também os meios utilizados para as obter no software.

#### **3.1. SOMADOR DE 2 BITS COM DESCRIÇÃO COMPORTAMENTAL**

Para o primeiro tópico foi solicitado a implementação em código VHDL de um somador para entrada de 2 bits, que é um dispositivo capaz de somar dois números binários A e B de N bits. Como veremos na implementação a seguir do código para um somador de 2 bits e sua respectiva simulação.

```

1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3  USE ieee.std_logic_unsigned.all;
4  ENTITY somador2bits IS
5  GENERIC (
6    WIDTH : integer := 2
7  );
8  PORT (
9    CIN : IN STD_LOGIC;
10   A : IN STD_LOGIC_VECTOR(WIDTH - 1 downto 0);
11   B : IN STD_LOGIC_VECTOR(WIDTH - 1 downto 0);
12   S : OUT STD_LOGIC_VECTOR(WIDTH - 1 downto 0);
13   COUT : OUT STD_LOGIC
14 );
15 END somador2bits;
16 ARCHITECTURE behavior OF somador2bits IS
17   SIGNAL SUM: STD_LOGIC_VECTOR(WIDTH downto 0);
18 BEGIN
19   SUM <= ('0' & A) + ('0' & B) + CIN;
20   S <= SUM(WIDTH - 1 downto 0);
21   COUT <= SUM(WIDTH);
22 END behavior;

```

Figura 1 - Código VHDL para o somador de 2 bits.

Como pode-se perceber na simulação, esse código não satisfaz completamente todas as somas possíveis para entrada de 2 bits, por causa do problema de overflow e a não contemplação do carryout no resultado final da soma. Assim, uma forma de resolver seria aplicando o somador do tipo carry-ripple.

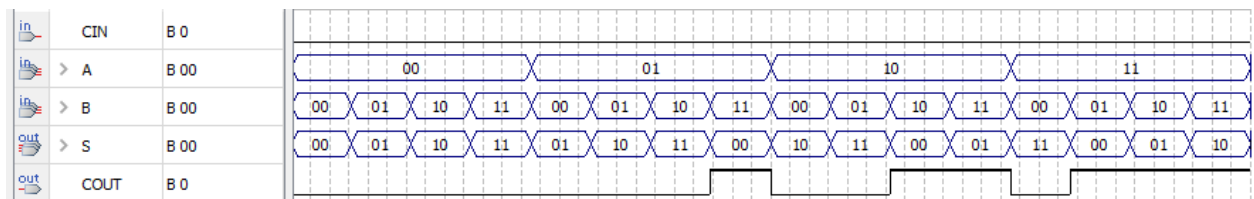


Figura 2 - Resultado da simulação para o somador de 2 bits.

### 3.2. SOMADOR DE 2 BITS UTILIZANDO PORTAS LÓGICAS

Para o segundo tópico, foi solicitada a construção de um somador de 2 bits, utilizando portas lógicas, apresentando as equações e tabela verdade desenvolvidas no processo. A seguir será mostrada a tabela verdade, a partir dela foi possível obter as equações de S0, S1 e S2 e posteriormente a implementação em VHDL das portas lógicas e sua simulação.

a0	b0	a1	b1	s0	s1	s2
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	1	0
0	1	0	1	0	1	1
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	0	1	1
1	0	1	1	1	0	0
1	1	0	0	1	0	0
1	1	0	1	1	0	1
1	1	1	0	1	0	1
1	1	1	1	1	1	0

Tabela 1 - Tabela Verdade.

Com isso podemos obter as seguintes equações como veremos a seguir:

$$s0 = (a0 \oplus b0) a1 b1 + a0 b0$$

$$s1 = a0 \oplus b0 \oplus (a1 b1)$$

$$s2 = a1 \oplus b1$$

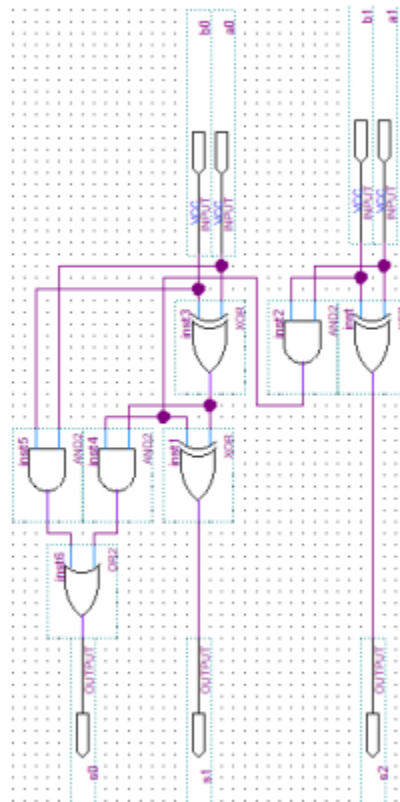


Figura 3 - Somador de 2 bits em portas lógicas.

in	> A	B 00																
in	> B	B 00																
out	> S	B 000																
			00				01				10				11			
			00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
			000	001	010	011	001	010	011	100	010	011	100	101	011	100	101	110

Figura 4 - Simulação do somador de 2 bits.

### 3.3. BLOCO MEIO SOMADOR EM ESTILO CARRY-RIPPLE

Para este tópico foi pedido a representação de um bloco meio somador em estilo carry-ripple, que consiste basicamente em um componente que recebe como entradas os bits 2 bits e gera como saída uma soma s e um carry (o “vai um”) de saída.

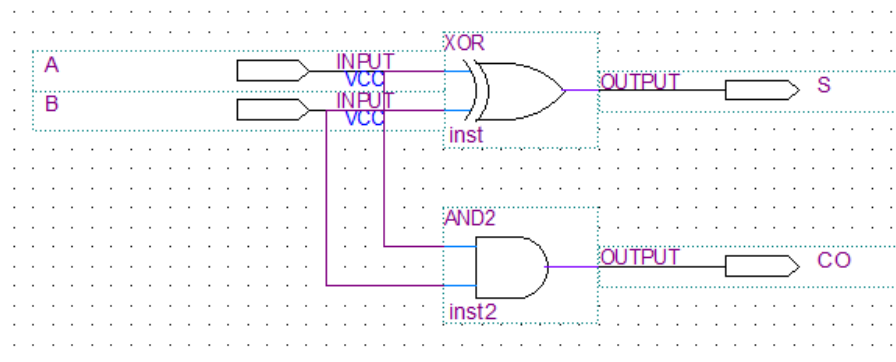


Figura 5 - Bloco meio somador com carry-ripple.

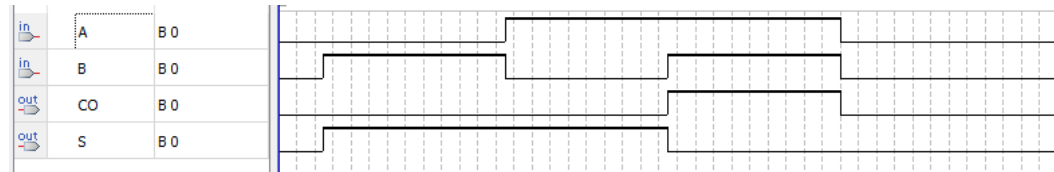


Figura 6 – Simulação para o bloco meio somador.

Como visto no esquema do bloco e na simulação, quando somamos A e B sendo 1 com 0, temos a saída sum(s) que vale 1 e o CARRY-OUT que vale 0. Agora vamos somar 1 com 1, e quando fazemos isso, a soma sum vale 0 mas o carry vale 1, ou seja, se concatenarmos o carry e o sum, temos que  $1+1 = 10$ , que era o resultado esperado da soma.

### 3.4. BLOCO SOMADOR COMPLETO EM ESTILO CARRY-RIPPLE

Neste tópico foi pedida a construção de um somador completo, ele recebe como entradas os bits a e b, e também um carry in de entrada, com isso teremos como saída uma soma s e um carry de saída. Assim, logo abaixo está a montagem dos circuitos utilizando as portas lógicas.

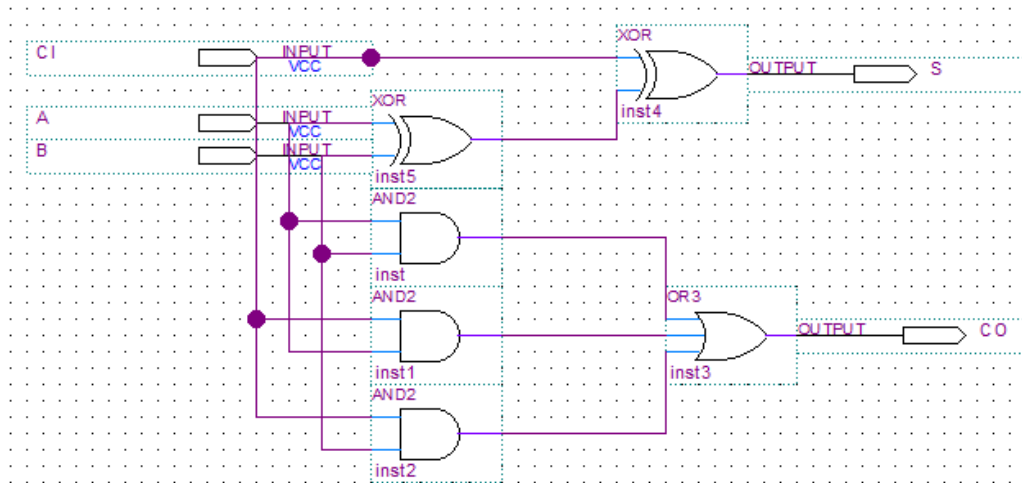


Figura 7 - Bloco somador completo com carry-ripple.

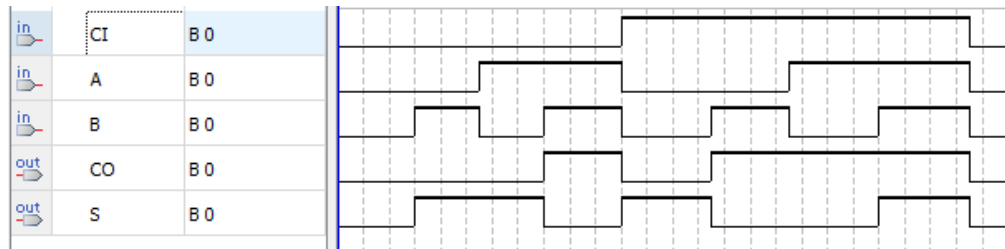


Figura 8 - Simulação para o bloco somador completo.

### 3.5. SOMADOR DE 3 BITS DO TIPO CARRY-RIPPLE

Neste penúltimo tópico foi pedida a implementação de um somador de 3 bits semelhante aos anteriores, porém para isso vamos utilizar de circuitos semelhantes aos usados nos itens anteriores e assim obter um somador que seja capaz de realizar as somas, como veremos na imagem da implementação do circuito e sua respectiva simulação.

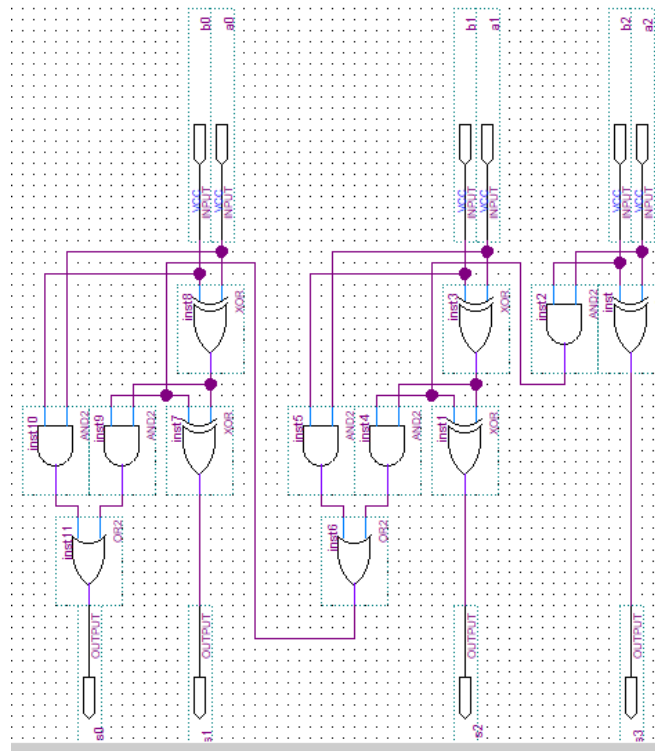


Figura 9 - Somador 3 bits carry-ripple.

in	> A	B 000	000	001	010	011	100	101	110	111	000	001	010	011	100	101	110	111	000	001	010	011	100	101	110	111	000	001	010	011	100	101	110	111
in	> B	B 000	000	001	010	011	100	101	110	111	000	001	010	011	100	101	110	111	000	001	010	011	100	101	110	111	000	001	010	011	100	101	110	111
out	> S	B 0000	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Figura 10 - Resultado da simulação somador 3 bits carry-ripple.

### 3.6. SOMADOR DE 6 BITS

Na última questão, construímos um somador de 6 bits adicionando 3 somadores completos ao projeto de portas lógicas do somador de 3 bits. Sendo assim, as imagens a seguir mostram o resultado final da implementação em portas lógicas e da simulação.

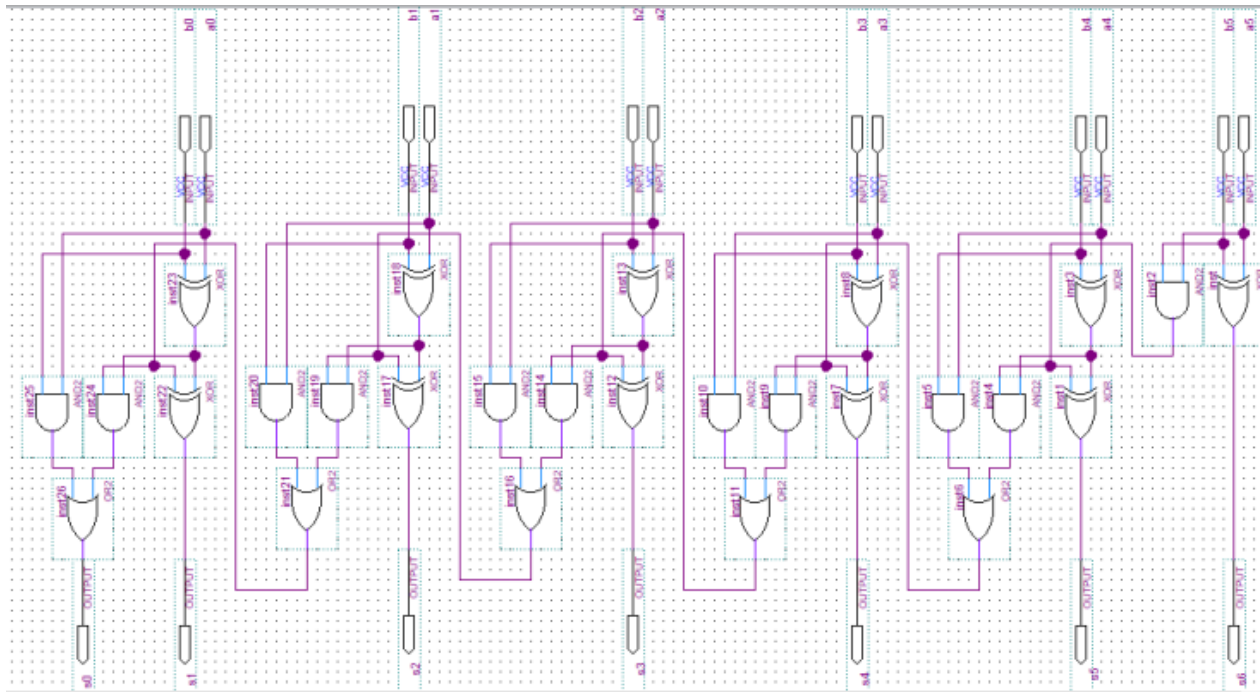


Figura 11 - Somador de 6 bits.

<div> <div></div> <div></div> <div></div> </div>	<div> <div>&gt; A</div> <div>8 000000</div> </div>	000000	000001	000010	000011	000100	000101	000110	000111	001000	001001	001010	001011	001100	001101	001110	001111
		000011	000100	000101	000110	000111	001000	001001	001010	001011	001100	001101	001110	001111	010000	010001	010010
		000011	000011	000011	000011	000011	000011	000011	000011	000011	000011	000011	000011	000011	000011	000011	000011

Figura 12 - Resultado da simulação do somador de 6 bits.

## 4. CONCLUSÕES

A realização deste laboratório permitiu o melhor entendimento a respeito dos assuntos vistos nas aulas teóricas, a partir do desenvolvimento dos Somadores, tanto através da descrição comportamental em linguagem VHDL, como também a montagem do circuito utilizando portas lógicas, foi possível observar de forma prática o funcionamento deles. Para tal, utilizamos da linguagem VHDL e simulações do ModelSim através do Quartus II e os conceitos adequados para obtermos os códigos necessários para a execução.

## 5. REFERÊNCIAS

[1] QUARTUS II. Software de simulação.

[2] Vahid, Frank. Digital Design with RTL Design, VHDL, and Verilog Solution Manual. 2º Edição.2010.