

Laboratório 6 – Somadores

Objetivos:

1. Experimentar a descrição em VHDL de circuitos digitais, na forma de portas lógicas e comportamental;
2. Reforçar os conceitos de somadores.
3. Pôr em prática conceitos aprendidos na disciplina teórica.

Introdução:

Nesta aula iremos rever o funcionamento e a construção de somadores. Serão apresentadas duas formas de implementação deste circuito combinacional, e o motivo pelo qual o padrão tradicional de construção do circuito não é recomendado.

Somadores:

Os somadores são um bloco combinacional de grande importância em projetos de circuitos digitais, já que a adição é uma das operações mais comuns. Um somador de largura N é responsável por somar dois números binários A e B de N bits, além de administrar o *carry* ("vai um") C de um bit.

Uma das formas de projetar este circuito é através da construção da tabela verdade, onde podemos extrair as equações para cada saída do somador. O problema para esta abordagem é que, a medida em que a largura do somador aumenta, a tabela verdade e a complexidade do circuito cresce de forma exponencial. Um somador de quatro bits, por exemplo, teria duas entradas de quatro bits, uma entrada de *carry* (opcional), quatro saídas de soma e uma saída de *carry*. Ou seja, no exemplo citado acima temos uma tabela verdade extensa, com ao menos 13 colunas.

Uma segunda abordagem para a construção dos somadores seria replicar a forma como as adições são realizadas manualmente, através de uma coluna por vez. Por exemplo, vamos considerar a soma ilustrada na Figura 1. Na primeira coluna é feita a soma dos dois bits de entrada, gerando um *carry* e um resultado de saída. Nas colunas restantes é realizada a soma dos bits de entrada juntamente com o *carry*, que é repassado para a coluna seguinte caso exista.

$$\begin{array}{r}
 1 \leftarrow \text{Carry (vai um)} \\
 1101 \\
 + 0111 \\
 \hline
 0 \leftarrow \text{Soma}
 \end{array}$$

Figura 1: Processo de soma em coluna, apresentando o resultado e o *carry*.

Ao final do processo de soma, o bit de *carry* se torna o último bit que compõe o resultado final (o MSB). O circuito somador que opera neste formato é conhecido como *carry-ripple*. A Figura 2 ilustra o processo completo de soma para o exemplo descrito anteriormente.

$$\begin{array}{r}
 111 \\
 1101 \\
 + 0111 \\
 \hline
 10100
 \end{array}$$

↑
Carry de Saída

Figura 2: Processo completo de soma, com o *carry* de saída.

A partir deste ponto podemos projetar um componente combinacional para realizar o processo de adição em colunas. A construção deste circuito digital é realizada através de meio somadores e somadores completos. A Figura 3 ilustra um bloco meio somador e sua tabela verdade. Este bloco possui como entradas os bits *a* e *b* que serão somados, o resultado *s* e o *carry* de saída *co*.

Entradas		Saídas	
a	b	co	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

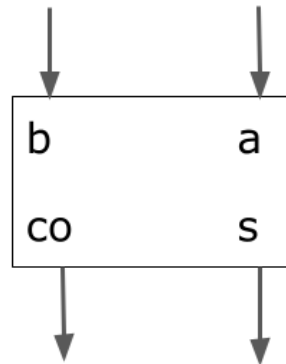


Figura 3: Representação do meio somador e tabela verdade.

A Figura 4 ilustra um bloco somador completo e sua tabela verdade. Neste caso temos como entradas os bits a , b e ci que serão somados, o resultado s e o *carry* de saída co . Como pode ser visto, a diferença entre o somador completo e o meio somador é a presença do *carry* de entrada ci .

Entradas			Saídas	
a	b	ci	co	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

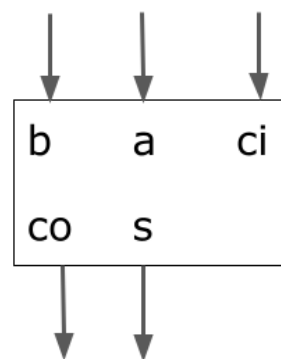


Figura 4: Representação do somador completo e tabela verdade.

É importante notar que estes circuitos realizam somas de um bit. Para somar um número binário de largura N maior que um, é possível combinar os blocos somadores através de instancição. Esta configuração permite que cada bloco receba uma coluna dos bits a e b que serão somados, e o *carry* de saída é propagado para o *carry* de entrada do bloco seguinte. A Figura 5 ilustra um somador de quatro bits, criado a partir de um meio somador e três somadores completos.

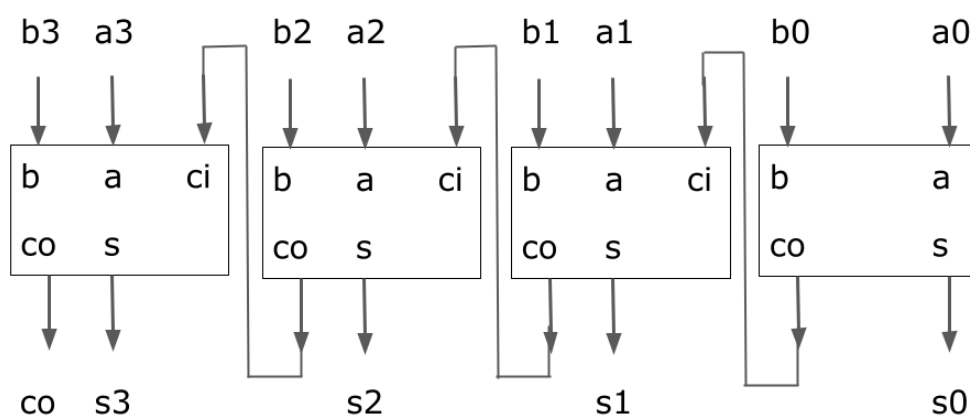


Figura 5: Representação de uma soma de quatro bits utilizando meio somador e somadores completos.

Descrição Comportamental do VHDL:

A linguagem de descrição de hardware VHDL permite a utilização da operação de soma de forma comportamental. Nesse tipo de descrição, não é necessário preocupar-se com portas lógicas. Porém, durante o *design*, é importante gerenciar o comportamento do *carry* corretamente, além de evitar problemas de *overflow*. Este erro acontece quando não existem bits suficientes para computar o resultado da soma. No quadro 1, a funcionalidade do somador completo é implementada por meio de descrição comportamental.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

ENTITY somador IS
GENERIC (
    WIDTH : integer := 16
);
PORT (
    CIN    : IN  STD_LOGIC;
    A      : IN  STD_LOGIC_VECTOR(WIDTH - 1 downto 0);
    B      : IN  STD_LOGIC_VECTOR(WIDTH - 1 downto 0);
    S      : OUT STD_LOGIC_VECTOR(WIDTH - 1 downto 0);
```

```
COUT      : OUT STD_LOGIC
);
END somador;

ARCHITECTURE behavior OF somador IS

    SIGNAL SUM: STD_LOGIC_VECTOR(WIDTH downto 0);

BEGIN

    SUM <= ('0' & A) + ('0' & B) + CIN;
    S <= SUM(WIDTH - 1 downto 0);
    COUT <= SUM(WIDTH);

END behavior;
```

Quadro 1: Descrição comportamental de um somador completo de 16 bits.

Hora de Praticar:

1. Implemente em VHDL, utilizando a descrição comportamental, um somador para entradas de 2 bits;
2. Implemente em VHDL, utilizando portas lógicas, um somador para entradas de 2 bits. Apresente a tabela verdade e as equações lógicas;
3. Implemente em VHDL, utilizando portas lógicas, um bloco meio somador que compõe o estilo *carry-ripple*;
4. Implemente em VHDL, utilizando portas lógicas, um bloco somador completo que compõe o estilo *carry-ripple*;
5. Implemente em VHDL, um somador para entradas de 3 bits do tipo *carry-ripple*, utilizando os circuitos criados nas questões 3 e 4;
6. Implemente em VHDL, um somador para entradas de 6 bits utilizando o somador da questão 5;
7. Entregue um relatório descrevendo a execução dos itens 1 a 6.