

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS –
CEFET/MG**

**DIOGO EMANUEL ANTUNES SANTOS
GUILHERME AUGUSTO DE OLIVEIRA
LUIZ EDUARDO LEROY SOUZA**

Circuito Somador Completo
Relatório 5

**BELO HORIZONTE
2021**

Sumário

Objetivos	2
Desenvolvimento	2
2.1 Tabela verdade	2
2.2 Expressões lógicas	3
2.3 Descrição em Verilog HDL	4
2.4 Circuito lógico	6
2.5 Simulação do circuito no ModelSim:	8
Análise dos resultados:	9

1. Objetivos

O relatório possui duas questões a serem respondidas. O primeiro tópico diz respeito ao projeto de um somador completo que funciona apenas com um bit. O segundo objetivo é implementar um circuito somador completo de quatro bits de dados, utilizando o módulo criado na primeira questão.

2. Desenvolvimento

Nesta parte será apresentado o passo a passo utilizado para desenvolver o projeto.

2.1 Tabela verdade

A tabela verdade da primeira parte tem como entradas dois números A e B, além do transporte de entrada Te, tendo em vista que trata-se de um somador completo. Como saída, apresenta o resultado da soma S e o transporte de saída Ts.

A	B	Te	S	Ts
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Tabela 01: Tabela verdade da primeira parte

A tabela verdade da segunda questão não foi desenvolvida, tendo em vista que, devido ao grande número de entradas, a tabela teria muitas linhas.

2.2 Expressões lógicas

A expressão booleana do somador completo de um bit é dado por meio dos respectivos diagramas de Veitch-Karnaugh, desenvolvidos com auxílio do software Logisim:

		B, Te			
		00	01	11	10
A	0	0	1	0	1
	1	1	0	1	0

Imagem 01: Mapa da primeira saída (S)

		B, Te			
		00	01	11	10
A	0	0	0	1	0
	1	0	1	1	1

Imagem 02: Mapa da segunda saída (Ts)

Como é possível observar pelos mapas, a saída S não possui simplificação por este método. De modo geral, é possível escrever a expressão final como:

$$S = A \text{ xor } B \text{ xor } Te$$

Por outro lado, a expressão inicial da segunda saída pode ser simplificada como:

$$Ts = BTe + ATe + AB$$

Novamente, no caso da segunda questão, não é possível detalhar uma expressão lógica para representar o mesmo, tendo em vista que será formado por meio da estrutura hierárquica.

2.3 Descrição em Verilog HDL

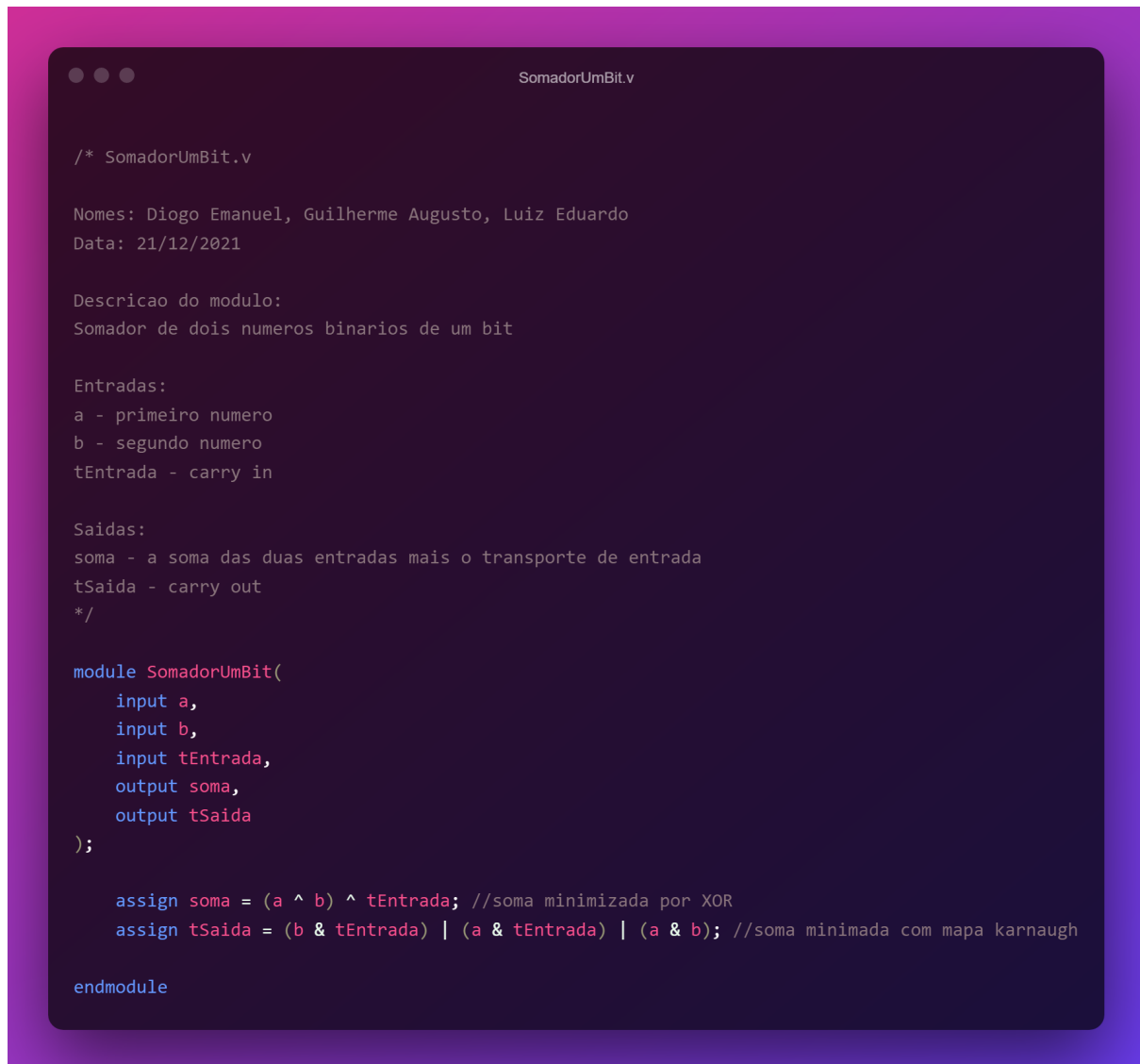


Imagem 03: Representação em Verilog HDL do módulo 1

Na imagem anterior é possível entender o funcionamento do módulo somador de um bit completo que foi desenvolvido utilizando a linguagem Verilog HDL. No código são declaradas as mesmas entradas e saídas exemplificadas na tabela verdade. Basta ao final do código aplicar as duas expressões lógicas encontradas.

```
SomadorCompleto.v

/* SomadorCompleto.v

Nomes: Diogo Emanuel, Guilherme Augusto, Luiz Eduardo
Data: 21/12/2021

Descricao do modulo:
Modulo que implementa um somador completo de dois numeros de quatro bits utilizando estrutura
hierarquica.

Entradas:
[3:0]a - primeiro numero
[3:0]b - segundo numero

Saídas:
soma[4:0] - a soma das duas entradas
*/

module SomadorCompleto(
    input [3:0]a,
    input [3:0]b,

    output [4:0]soma
);

    wire [2:0]fio;

    SomadorUmBit somadorUm(
        a[0],
        b[0],
        0,
        soma[0],
        fio[0]
    );

    SomadorUmBit somadorDois(
        a[1],
        b[1],
        fio[0],
        soma[1],
        fio[1]
    );

    SomadorUmBit somadorTres(
        a[2],
        b[2],
        fio[1],
        soma[2],
        fio[2]
    );

    SomadorUmBit somadorQuatro(
        a[3],
        b[3],
        fio[2],
        soma[3],
        soma[4]
    );

endmodule
```

Imagem 04: Descrição da segunda parte

A implementação do circuito completo, referente à questão 2, envolve a utilização do módulo desenvolvido na parte anterior. Utilizando da estrutura hierárquica, foram declarados quatro instâncias do somador de um bit, tendo em vista que o objetivo é somar quatro bits.

Assim, as variáveis presentes no circuito final são os dois inputs de 4 bits (A e B), mais a soma resultante. Além disso, foi preciso declarar uma *wire* de 3 bits para conectar o transporte de saída de um dos módulos ao transporte de entrada do módulo seguinte. Observe ainda que, ao final, o transporte de saída proveniente do último módulo é inserido no número

que representa a soma. Isso é feito para evitar casos de overflow, uma vez que a soma de dois números de quatro bits pode facilmente resultar em um número de 5 bits.

2.4 Circuito lógico

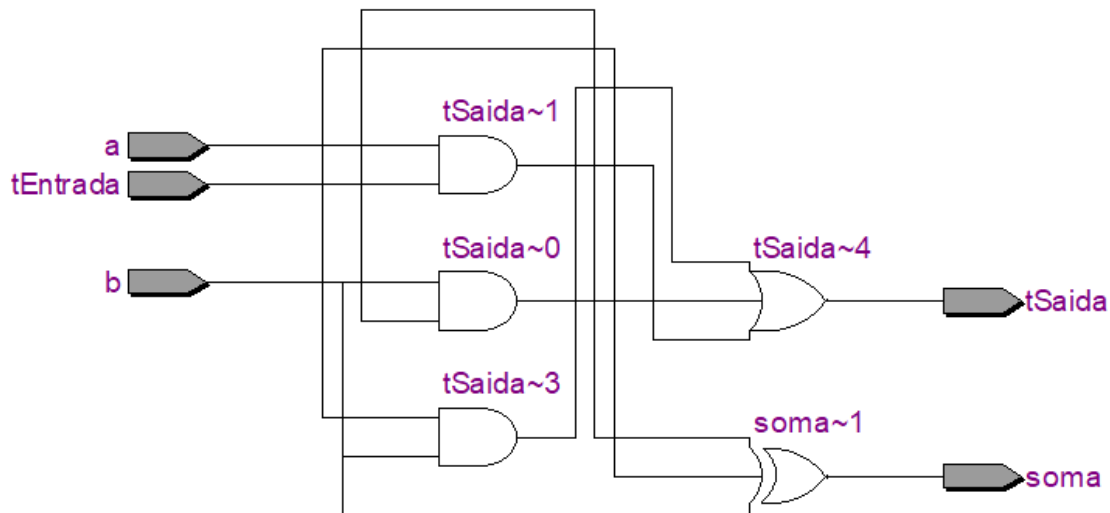


Imagem 05: Circuito da primeira parte

Descrição do circuito lógico:

No circuito lógico acima que compreende o circuito de um bloco somador de um bit que recebe os bits *a* e *b*, cada um se referendo a números diferentes em binário que serão somados, além do bit *tEntrada* que corresponde ao transporte de entrada recebido de um transporte de saída(*tSaída*) de uma soma anterior e, para o bit menos significativo *tSaída* recebe nível lógico 0.

Com relação às portas utilizadas para a construção do circuito somador acima foi utilizado uma porta XOR que recebe como entrada os bits *a*, *b* e *tEntrada* e tem como saída a soma desses bits. Além dessa, temos três portas AND e uma porta OR que controlam a parte do circuito que resulta na saída *tSaída*, o transporte de saída. Dessa forma, a porta *tSaída~1* recebe como entrada os bits *a* e *tEntrada*, a porta *tSaída~3* recebe como entrada os bits *a* e *b* e a entrada *tSaída~0* recebe como entrada os bits *b* e *tEntrada*. Já a porta OR *tSaída~4* recebe como entrada as saídas provenientes das portas AND *tSaída~0*, *tSaída~1* e *tSaída~3* e tem

como saída tSaída que corresponde ao transporte de saída que num próximo bloco somador será o transporte de entrada(tEntrada).

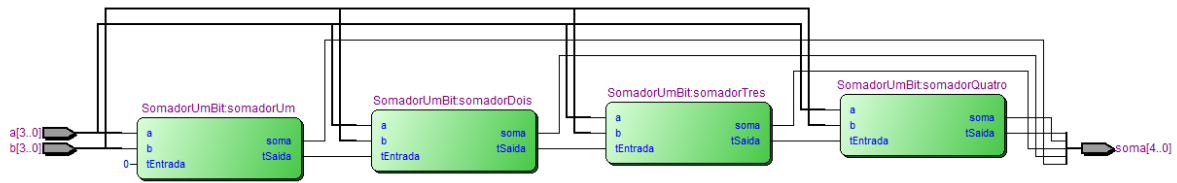


Imagem 06: Circuito da segunda parte

Descrição do circuito lógico

Acima temos o circuito formado por 4 somadores completos de um bit , dessa forma é possível somar dois números de 4 bits , sendo cada bit do menos significativo ao mais significativo de cada número e, ainda, o transporte de entrada(tEntrada), que na soma dos bits menos significativos recebe nível lógico baixo, também é somado, o mesmo que corresponde ao transporte de saída(tSaída) da soma anterior nos próximos blocos somadores. Por fim, o transporte de saída do último bloco somador corresponde ao overflow no caso da soma dos dois números de 4 bits resultar em um número de 5 bits.

2.5 Simulação do circuito no ModelSim:

Seguem os sinais de onda gerados no software:

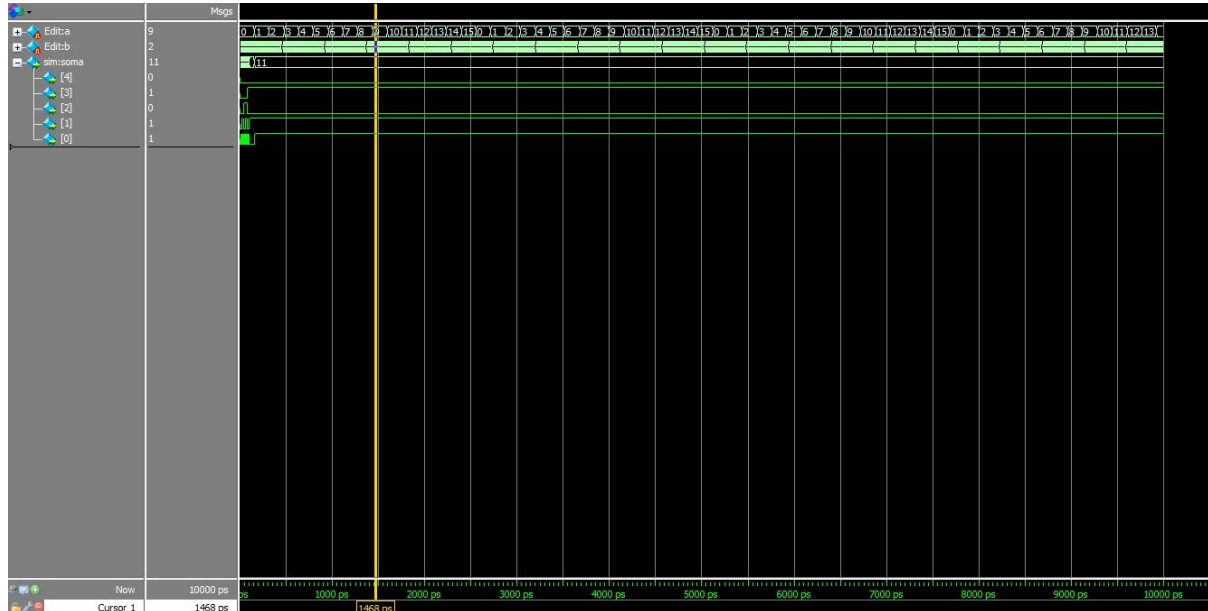


Imagem 07: Painel de ondas do Modelsim Altera para o módulo 2

Para realizar essa simulação os seguintes sinais de onda foram gerados:

- para a entrada A foi emitido uma onda de 160ps de período, variando por todos os valores possíveis (0000 - 1111);
- para a entrada B foi emitido uma onda de 10ps de período, também variando por todas as entradas possíveis. É interessante notar que esta variação dos períodos garante que todas as possibilidades serão testadas;
- o gráfico não apresenta valores para o T_e , apresentando sempre transporte de entrada inicial igual a 0.

O tempo máximo da simulação foi definido para 10000ps. A saída apresentada relaciona corretamente a soma dos números binários A e B.

3. Análise dos resultados:

Neste relatório foi desenvolvido um circuito digital em Verilog HDL de um somador completo de 1 bit. Após isso, foi utilizado o módulo criado como parte de um somador de 4 bits, implementado por meio da estrutura hierárquica. Dessa maneira, os resultados simulados no ModelSim Altera demonstram de forma correta os valores somados de A e de B, e a saída respeita a estrutura hierárquica, o que confirma que o código feito no Quartus II foi executado com sucesso.

Neste código percebemos que os somatórios feitos anteriormente são verdadeiros. Com isso, analisamos os valores A, B e a saída S, uma vez que o resultado deve obedecer o valor do somatório deles. Como no exemplo acima, quando A=0000, B=0001, então a saída deve ser o valor de 0001. O que comprova o que foi demonstrado na simulação do ModelSim Altera.