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

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

Circuito Somador/Subtrator Completo

Relatório 6

BELO HORIZONTE
2021

Sumário

1. Objetivos

O objetivo principal deste relatório é continuar o trabalho desenvolvido na atividade anterior. Para tanto, além do circuito somador desenvolvido anteriormente conseguir somar dois números de quatro bits positivos, será possível realizar a soma de um número positivo A com um número negativo B. Isso será feito utilizando complemento de dois e um sinal *enable*.

2. Desenvolvimento

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

2.1 Tabela verdade

A primeira parte do projeto foi desenvolvida no trabalho anterior. Trata-se da implementação de um circuito somador completo para dois números (A e B) de um bit.

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. Além disso, é importante destacar que o método utilizado será a estrutura hierárquica.

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:

Imagem 01: Mapa da primeira saída (S)

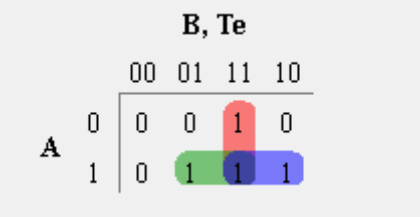


Imagem 02: Mapa da segunda saída (Ts)

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:

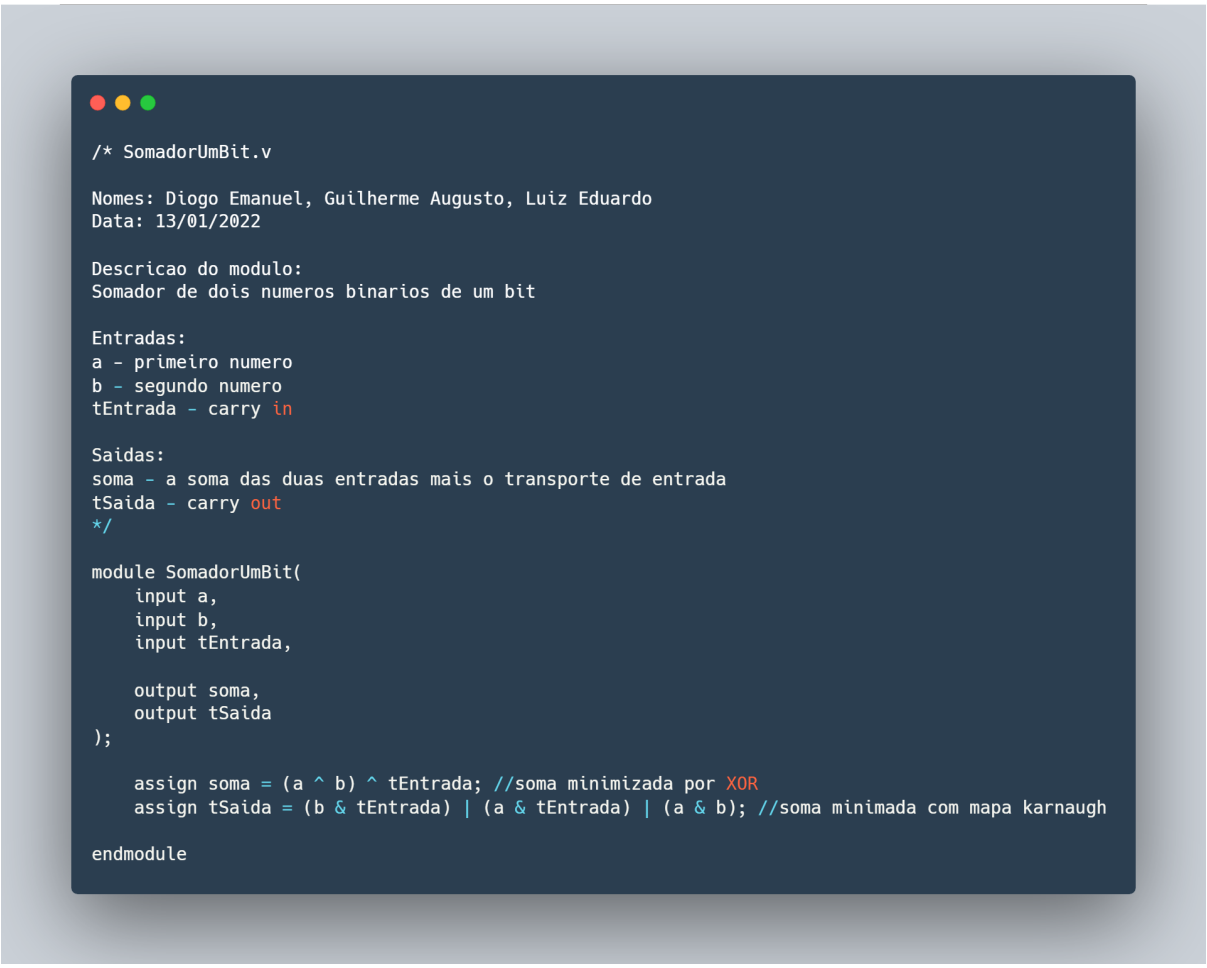
$$S = A\ xor\ B\ xor\ Te$$

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



```
/* SomadorUmBit.v

Nomes: Diogo Emanuel, Guilherme Augusto, Luiz Eduardo
Data: 13/01/2022

Descricao do modulo:
Somador de dois numeros binarios de um bit

Entradas:
a - primeiro numero
b - segundo numero
tEntrada - carry in

Saidas:
soma - a soma das duas entradas mais o transporte de entrada
tSaida - carry out
*/

module SomadorUmBit(
    input a,
    input b,
    input tEntrada,

    output soma,
    output tSaida
);

    assign soma = (a ^ b) ^ tEntrada; //soma minimizada por XOR
    assign tSaida = (b & tEntrada) | (a & tEntrada) | (a & b); //soma minimada com mapa karnaugh

endmodule
```

Imagem 03: Representação em Verilog HDL do módulo somador de um bit

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.

```
/* SomadorSubtratorCompleto.v

Nomes: Diogo Emanuel, Guilherme Augusto, Luiz Eduardo
Data: 13/01/2022

Descricao do modulo:
Realiza a soma ou a subtração de dois numeros de quatro bits.

Entradas:
A - primeiro numero de quatro bits
B - segundo numero de quatro bits
M - sinal enable para indicar soma caso 0 ou subtração caso 1

Saidas:
resultado - resultado da soma ou subtracao com cinco bits, levando em conta a possibilidade de overflow
*/

module SomadorSubtratorCompleto(
  input [3:0]A,
  input [3:0]B,
  input M,

  output [4:0]resultado
);

  wire [4:0]soma;
  wire [4:0]tEntrada;

  SomadorUmBit somadorUm(
    A[0],
    B[0] ^ M,
    M,
    soma[0],
    tEntrada[0]
  );

  SomadorUmBit somadorDois(
    A[1],
    B[1] ^ M,
    tEntrada[0],
    soma[1],
    tEntrada[1]
  );

  SomadorUmBit somadorTres(
    A[2],
    B[2] ^ M,
    tEntrada[1],
    soma[2],
    tEntrada[2]
  );

  SomadorUmBit somadorQuatro(
    A[3],
    B[3] ^ M,
    tEntrada[2],
    soma[3],
    tEntrada[3]
  );

  // utilizado para verificar overflow durante a soma
  SomadorUmBit somadorCinco(
    0,
    M,
    tEntrada[3],
    soma[4],
    tEntrada[4]
  );

  assign resultado[0] = soma[0];
  assign resultado[1] = soma[1];
  assign resultado[2] = soma[2];
  assign resultado[3] = soma[3];
  assign resultado[4] = soma[4];

endmodule
```

Imagem 04: Descrição do módulo somador/subtrator de quatro bits

Conforme solicitado, a implementação do módulo principal envolve a utilização da estrutura hierárquica. Basicamente, o circuito aceita duas entradas A e B de quatro bits e o input de sinal M, utilizado para identificar a operação matemática. Além disso, é preciso retornar o resultado da respectiva operação.

O sistema conta também com dois *wires* para guardar os dados transitados dentro do circuito. O primeiro deles é o wire de quatro bits *soma* que tem o objetivo de guardar o resultado retornado por cada módulo do circuito somador de um bit. O segundo deles, também de quatro bits, tem o objetivo de servir como o transporte de entrada dos módulos de um bit. Percebe-se que a saída de um se tornará a entrada do próximo. Observa-se que caso M seja igual a 0, o circuito flui com os valores de B originais e o transporte de entrada inicial é 0, realizando a soma. Caso M seja igual a 1, o circuito troca os valores dos bits de B e o transporte de entrada se torna 1, ou seja, realiza-se complemento de dois, com o objetivo de realizar a subtração.

É importante destacar também a presença de um quinto somador no código. Isso ocorre para garantir que em casos de overflow durante a soma (A+B), seja adicionado o valor correto no resultado. Por fim, são realizados os comandos de *assign* para guardar o valor final na variável de saída.

2.4 Circuito lógico

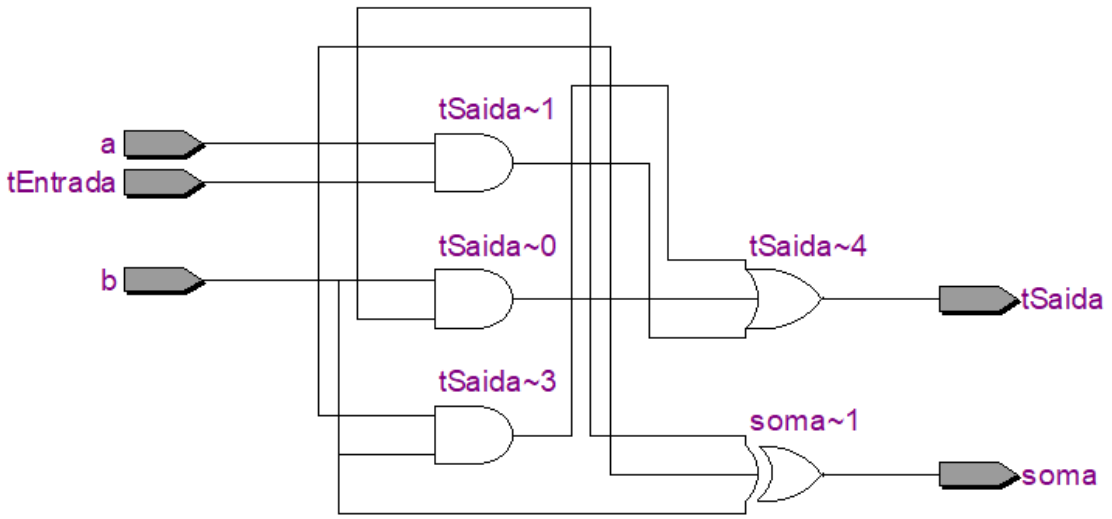


Imagem 05: Circuito da primeira parte

Descrição do circuito lógico:

No circuito lógico acima, que compreende o circuito de um bloco somador/subtrator de um bit que recebe os bits a e b, cada um se referindo 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) da soma anterior.

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 tranporte de saída que num próximo bloco somador/subtrator será o transporte de entrada(tEntrada).

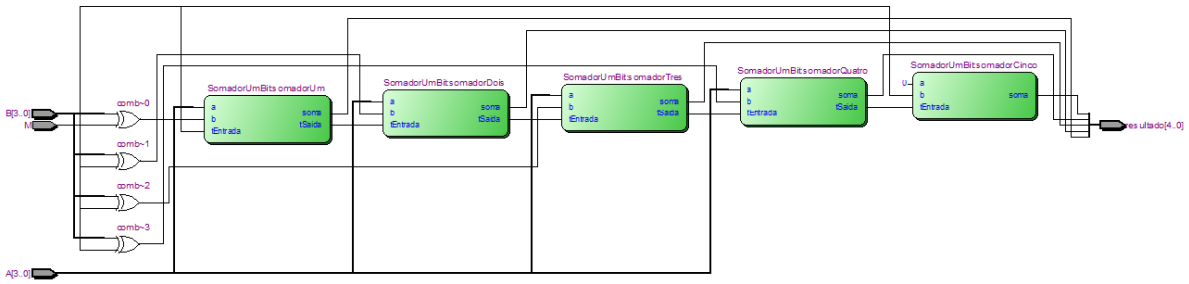


Imagem 06: Circuito da segunda parte

Descrição do circuito lógico

Acima temos o circuito formado por 5 somadores/subtratores completos de um bit , dessa forma é possível somar ou subtrair dois números de 4 bits, sendo cada bit do menos significativo ao mais significativo de cada número e também o transporte de entrada(tEntrada), que como explicado anteriormente funciona como o bit de controle para a

operação que será feito. Assim, o transporte de Saída(tSaída) de cada bloco corresponde ao bit tEntrada do bloco seguinte.

Quando o bit tEntrada em nível lógico 0, é feita a adição dos bits correspondentes ao números a e b, quando em nível lógico 1, o contrário acontece e é feita a adição dos bits dos número a com os bits do número b em complemento de 2, logo, se torna uma subtração. Dessa forma, os quatro primeiros blocos somador/subtrator do circuito são responsáveis por somar os bits de cada número e, o quinto e último bloco somador/subtrator, é responsável por somar os bits de overflow no caso da soma de números de 4 bits que resultam em números de 5 bits.

2.5 Simulação do circuito no ModelSim:

Seguem os sinais de onda gerados no software:

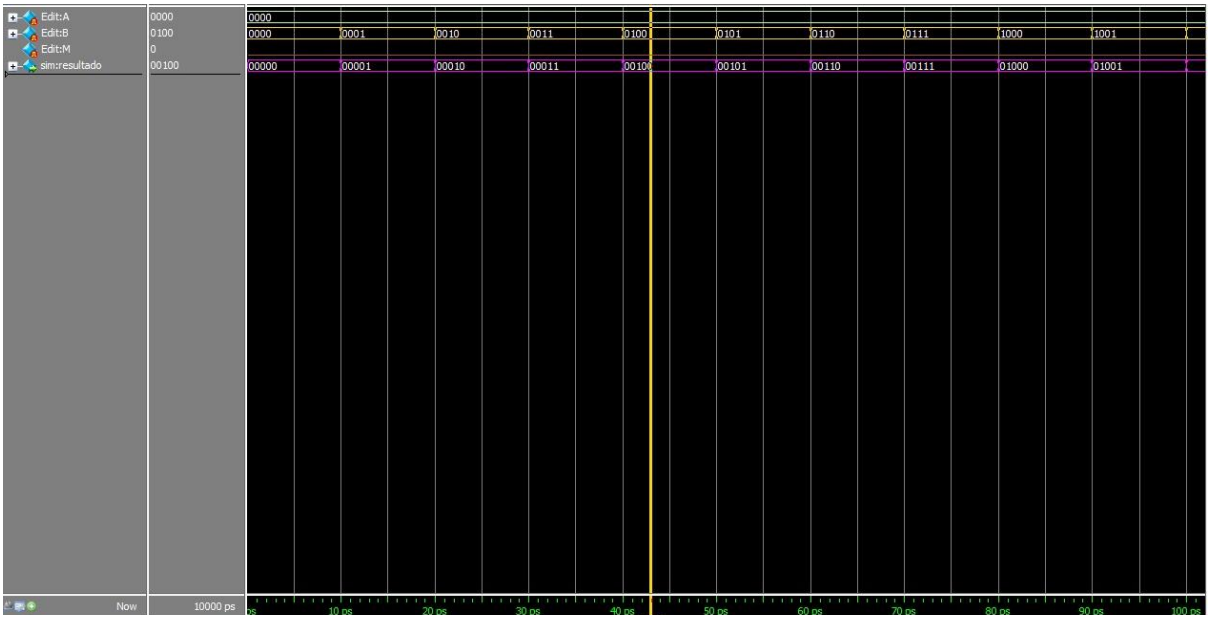


Imagem 07: Painel de ondas do Modelsim Altera

Para 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;
- Para a entrada M, foi dividido metade do gráfico para soma e metade para subtração, a fim de testar todos os valores.

O tempo máximo da simulação foi definido para 10000ps. A saída apresentada relaciona corretamente a soma ou subtração dos números binários A e B. Os valores de teste e o gráfico correspondente foram salvos no seu respectivo arquivo "*wave.do*".

3. Análise dos resultados:

Neste relatório demonstramos que foi desenvolvido um circuito digital em Verilog HDL de um somador completo de 1 bit levando em consideração o transporte de entrada. Após isso, foi chamado o módulo criado 5 vezes para criação de um somador completo de 4 bits, implementado por meio da estrutura hierárquica levando em consideração um bit de sinal que indicava a operação desejada. 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 transporte de entrada e um bit de sinal o que confirma que o código feito no Quartus II foi executado com sucesso, somando corretamente números positivos e números positivo + negativo.

Através da imagem das ondas do tópico anterior é possível analisar algumas das possibilidades de soma. No entanto, todas as combinações podem ser vistas utilizando o software Modelsim Altera e executando o arquivo "*wave.do"* fornecido.