

## 5o Projeto - Somador completo:

**Integrantes do grupo:** Lara Gama Santos, Mateus Ribeiro Ferraz, Sulamita Ester Costa.

**Objetivos:** Implementar, usando Verilog HDL, um circuito digital somador completo de 1 bit e, logo após, usar estrutura hierárquica para criar um somador completo de 4 bits.

### Primeira parte: Somador 1 bit

1-Tabela-verdade:

TE = transporte de entrada

TS = transporte de saída

Entradas: A,B e TE.

Saídas: Soma, TS.

A	B	TE	SOMA	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

2- Expressões booleanas:

No bloco 0 foi utilizado um  $\frac{1}{2}$  somador, construído a partir das seguintes expressões booleanas:

SOMA:  $A \oplus B$

TS:  $A.B$

Para encontrar as demais expressões booleanas nós utilizamos o mapa de Karnaugh.

Mapas de Karnaugh:

Utilizando a saída SOMA:

	$\sim A \sim B$	$\sim AB$	$AB$	$A \sim B$
$\sim TE$	0	1	0	1
$TE$	1	0	1	0

Temos:  $SOMA = \sim A. \sim B. Te + \sim A. B. \sim Te + A. B. Te + A. \sim B. \sim Te$

Utilizando a saída TS:

	$\sim A \sim B$	$\sim AB$	$AB$	$A \sim B$
$\sim TE$	0	0	1	0
$TE$	0	1	1	1

Temos:  $TS = B. Te + A. Te + AB$

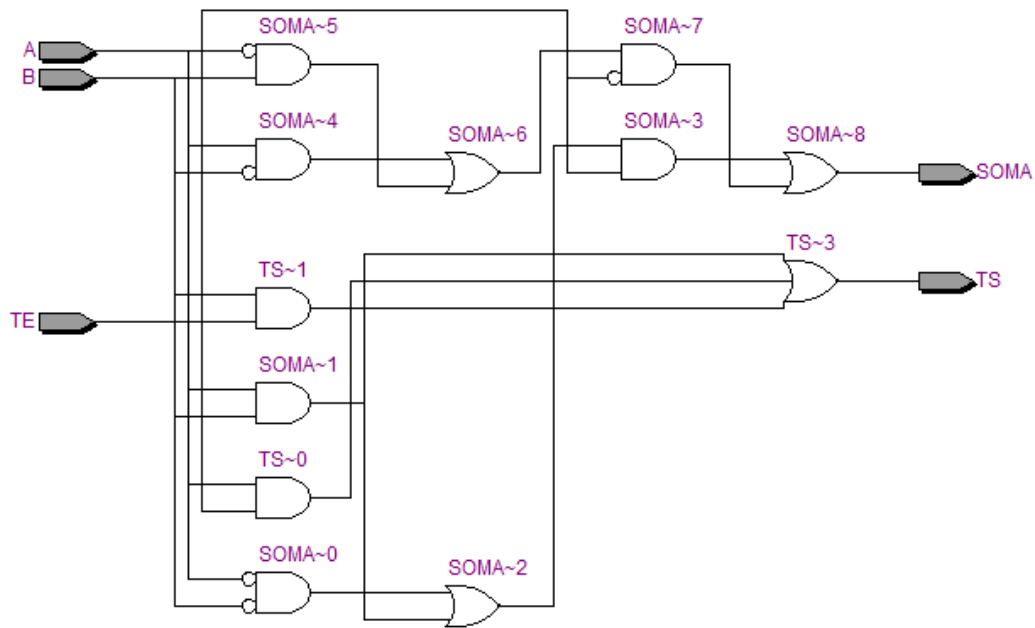
**3-** O circuito é sequencial ou combinacional?

Ele é combinacional, pois suas saídas dependem única e exclusivamente das combinações entre as variáveis de entrada.

**4-** A descrição do circuito em verilog HDL:

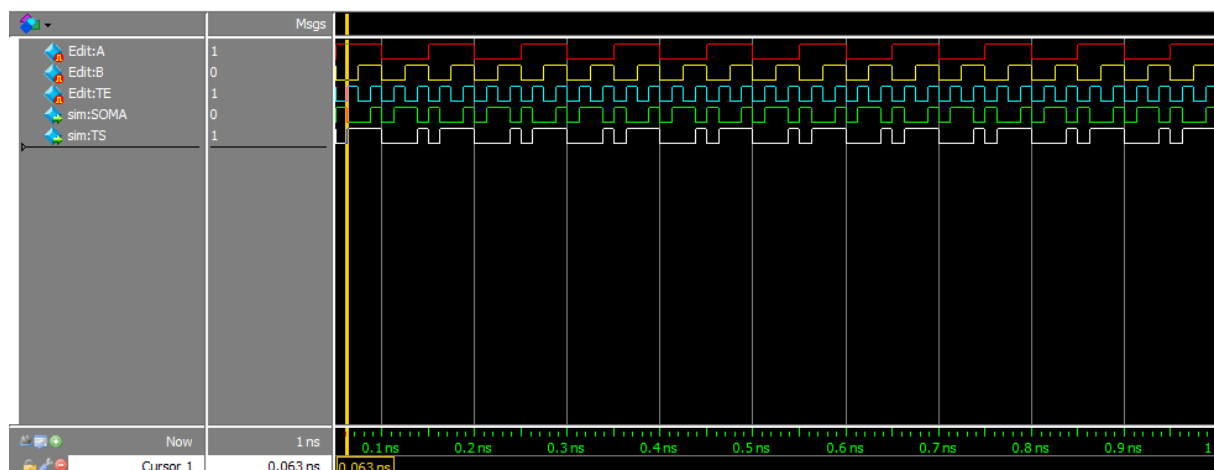
```
module somador_1bit(A, B, TE, SOMA, TS);  
  
  input A, B, TE;  
  output SOMA, TS;  
  
  assign SOMA = ( $\sim A \& \sim B \& TE$ ) | ( $\sim A \& B \& \sim TE$ ) | ( $A \& B \& TE$ ) | ( $A \& \sim B \& \sim TE$ );  
  assign TS = ( $A \& TE$ ) | ( $B \& TE$ ) | ( $A \& B$ );  
  
endmodule
```

**5-** Esquemático do circuito:



**6- a-** Nós utilizamos os períodos de 100, de 50 e de 25, pois eles nos permitem visualizar bem as diferenças nos resultados em cada valor lógico das entradas, já que o período de B é a metade de A, e o período de TE é a metade B.

**b-** A figura gerada na simulação:



**7- Comente e analise os resultados obtidos:**

Para desenvolver esse código, utilizamos a lógica de circuitos somadores estudados na parte teórica. Com o desenvolvimento desse projeto, observamos que quando somamos números binários há a ocorrência de bits de carry in e carry out,

isso permite que somadores completos de mais bits sejam construídos de maneira hierárquica utilizando como base o somador completo de 1 bit.

## **Segunda parte: Somador 4 bits:**

### **1-Tabela-verdade:**

A tabela verdade do somador de 4 bits é inviável de fazer manualmente pois é muito grande.

### **2- Expressões booleanas:**

Como utilizaremos a estrutura hierárquica para criar o somador 4 bits a partir do somador 1 bit, então, iremos implementar as mesmas expressões booleanas utilizadas na primeira parte.

No bloco 0 foi utilizado um  $\frac{1}{2}$  somador, construído a partir das seguintes expressões booleanas:

SOMA:  $A \oplus B$

T:  $A.B$

Os demais blocos são somadores completos, construídos a partir das seguintes expressões booleanas

$\sim A.\sim B.T + \sim A.B.\sim T + A.B.T + A.\sim B.\sim T$

$B.T + A.T + AB$

### **3- O circuito é sequencial ou combinacional?**

Ele é combinacional, pois suas saídas dependem única e exclusivamente das combinações entre as variáveis de entrada.

### **4- A descrição do circuito em verilog HDL:**

```

module somador_4bits(A, B , SOMA);
input  [3:0] A;
input  [3:0] B;

wire  [2:0] T;

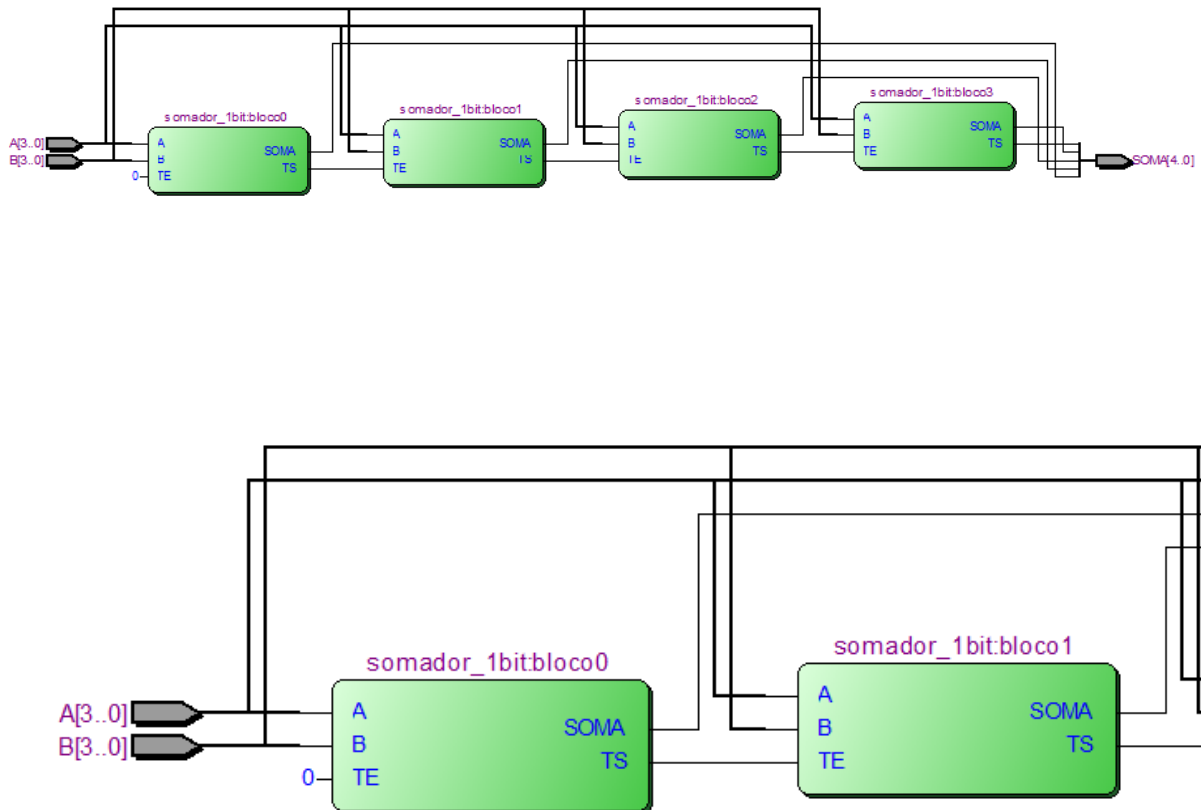
output [4:0] SOMA;

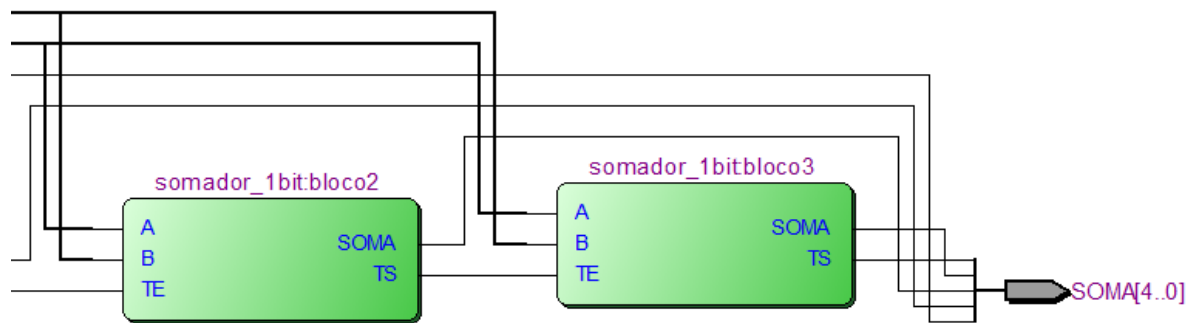
somador_1bit bloco0 (A[0], B[0], 0, SOMA[0], T[0]);
somador_1bit bloco1 (A[1], B[1], T[0], SOMA[1], T[1]);
somador_1bit bloco2 (A[2], B[2], T[1], SOMA[2], T[2]);
somador_1bit bloco3 (A[3], B[3], T[2], SOMA[3], SOMA[4]);

endmodule

```

## 5- Esquemático do circuito:





**6- a-** As imagens abaixo ilustram quais foram os tipos de sinal e períodos dos sinais de entrada. É possível observar que foram escolhidos períodos e ciclos diferentes para que pudessem ser comparadas diferentes possibilidades de saída.

A:

Counter Attributes

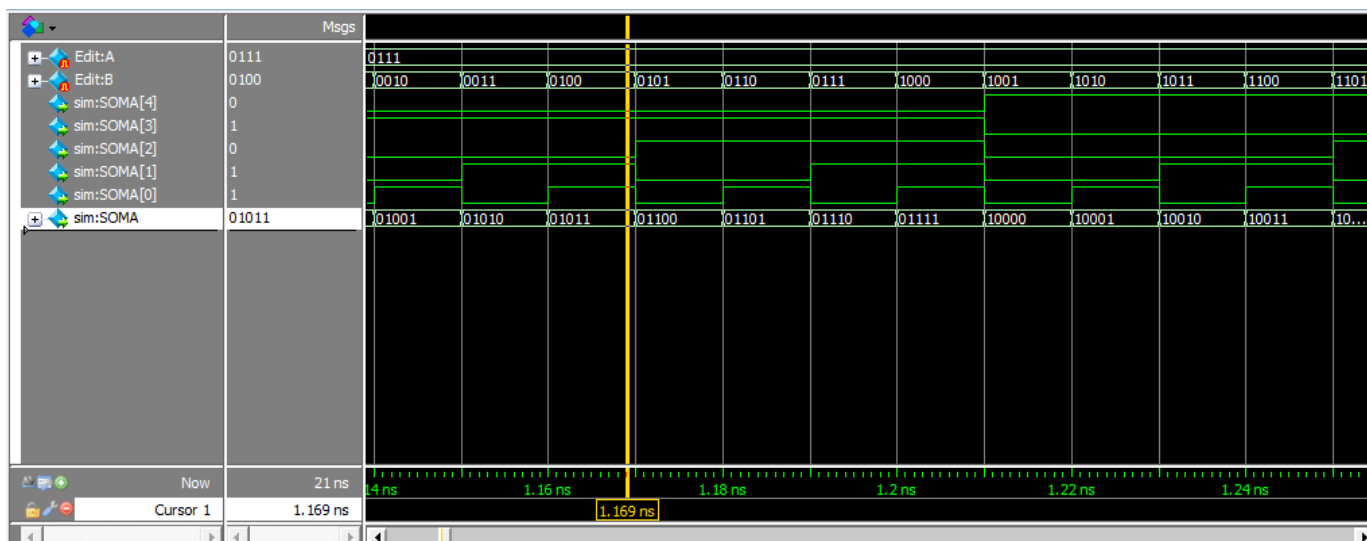
Start Value	End Value
0000	1111
Time Period	Time Unit
160	ps
Counter Type	Count Direction
Binary	Up
Step Count	
1	
Repeat	
<input checked="" type="radio"/> Forever <input type="radio"/> Never <input type="radio"/> Count	
Times	

B:

Counter Attributes

Start Value	End Value
0000	1111
Time Period	Time Unit
10	ps
Counter Type	Count Direction
Binary	Up
Step Count	
1	
Repeat	
<input checked="" type="radio"/> Forever <input type="radio"/> Never <input type="radio"/> Count Times <input type="text"/>	

b- A figura gerada na simulação:



7) Comente e analise os resultados obtidos:

Para criar o somador completo de 4 bits usamos estrutura hierárquica para implementar o somador 1 bit em blocos para resultar em um somador 4 bits. Nesse projeto, utilizamos quatro blocos somadores, em que o primeiro é um  $\frac{1}{2}$  somador e os demais são somadores completos. Além disso, observamos que quando somamos dois números binários de 4 bits, utilizando o somador, resulta em um número de 5 bits em que o bit mais à esquerda é um bit gerado por overflow.