



Departamento de Engenharia Elétrica  
Universidade de Brasília  
Disciplina: Laboratório de Sistemas Digitais (Turma: 04)  
Professor: Guilherme Torres

Primeiro Semestre de 2024

## **Experimento 5**

**Datas da realização do experimento:** 25 de abril de 2024

**Aluno:** Felipe Lopes Gibin Duarte

**Matrícula:** 231025207

---

### **1. INTRODUÇÃO**

Neste experimento, vamos construir um somador de palavras binárias usando somadores completos em cascata. Além disso, vamos ter um primeiro contato com o pacote STD\_LOGIC\_ARITH, bem como desenvolver um testbench para simulação e teste de circuitos em VHDL. Usaremos técnicas de projeto modular, desenvolvendo grandes sistemas construídos a partir de sistemas menores, tudo isso usando a linguagem de descrição de hardware VHDL. Ou seja, vamos desenvolver entidades mais complexas usando sistemas anteriormente desenvolvidos, como o somador completo e o somador parcial. Desse modo, todos os procedimentos serão desenvolvidos usando o software “ModelSim”. Para alcançar tais objetivos, o seguinte foi feito.

---

### **2. DESENVOLVIMENTO**

#### **Questão 1**

Escrever em VHDL e simular no ModelSim um somador de palavras de 4 bits, construído utilizando somente somadores completos (entidade desenvolvida no visto 1 do experimento 2, utilizada aqui como “component”). A nova entidade deve ter como entrada dois vetores A e B (com 4 bits cada) e, como saída, um vetor S (com 5 bits). O somador completo está definido segundo as equações abaixo.

$$S = A \oplus B \oplus Cin \text{ e } Cout = A \cdot B + A \cdot Cin + B \cdot Cin$$

-O código documentado está no segundo arquivo

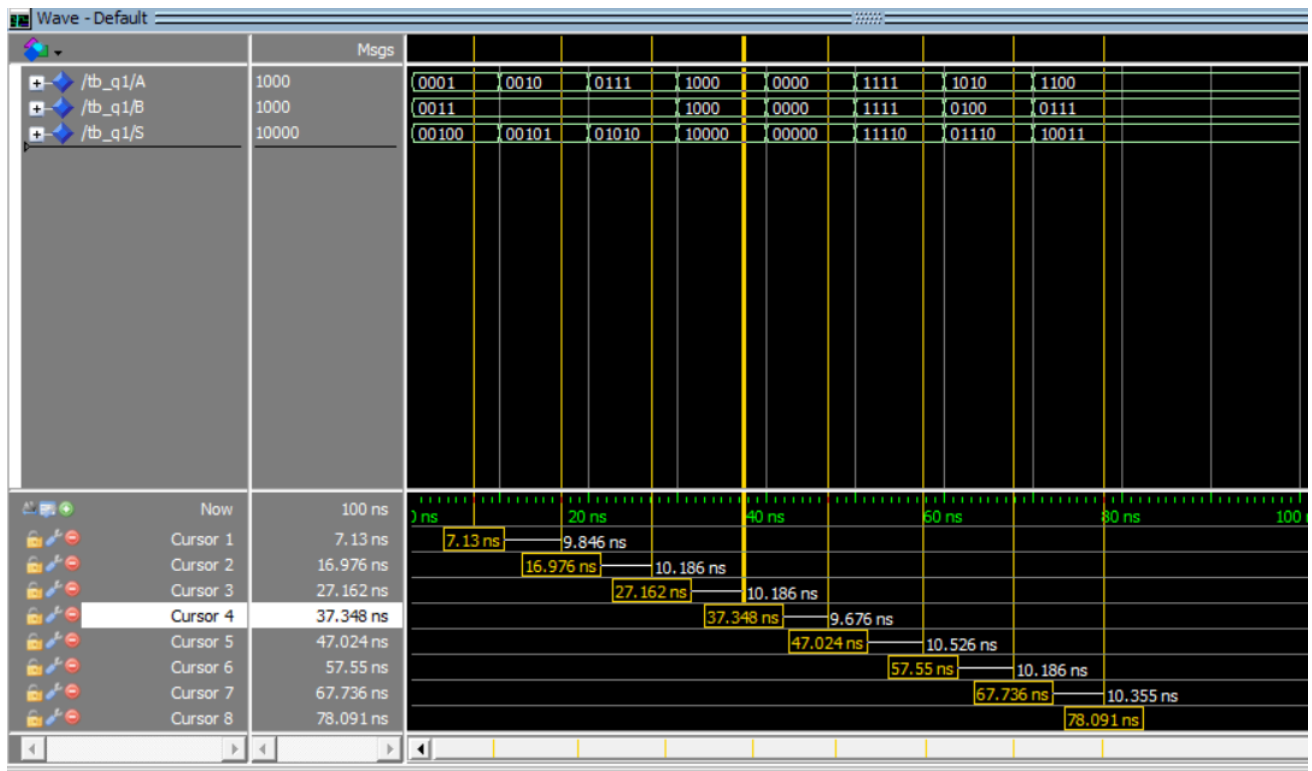
-Confirmação do código compilado:

Project - C:/Users/gibin/OneDrive/Desktop/Lab SD/Model Sim exp5/Exp5					
Name	Status	Type	Order	Modified	
Q1(5).vhd	✓	VHDL	0	04/26/2024 03:50:02 ...	
somador.vhd	✓	VHDL	5	04/25/2024 05:26:07 ...	
Q2(5).vhd	✗	VHDL	2	04/25/2024 05:09:04 ...	
tb_Q1(5).vhd	✓	VHDL	1	04/26/2024 04:18:05 ...	
Q3(5).vhd	✗	VHDL	4	04/25/2024 05:09:22 ...	
tb_Q2(5).vhd	✗	VHDL	3	04/25/2024 05:09:13 ...	

```
# Compile of tb_Q1(5).vhd was successful.  
# Compile of somador.vhd was successful.  
# Compile of Q1(5).vhd was successful.  
  
ModelSim>
```

-Simulação do código no ModelSim:



-Todos os casos foram vistos, e o seguinte foi analisado:

- Cursor 1) A='0001', B='0011', S='00100';     1+3=4
- Cursor 2) A='0010', B='0011', S='00101';     2+3=5
- Cursor 3) A='0111', B='0011', S='01010';     7+3=10
- Cursor 4) A='1000', B='1000', S='10000';     8+8=16
- Cursor 5) A='0000', B='0000', S='00000';     0+0=0
- Cursor 6) A='1111', B='1111', S='11110';     15+15=30
- Cursor 7) A='1010', B='0100', S='01110';     10+4=14
- Cursor 8) A='1100', B='0111', S='10011';     12+7=19

## 2.2) Questão 2

Escrever em VHDL e simular no ModelSim um somador de palavras de 4 bits, construído utilizando o operador '+' do pacote STD\_LOGIC\_ARITH. A nova entidade deve ter como entrada dois vetores A e B (com 4 bits cada) e, como saída, um vetor S (com 5 bits).

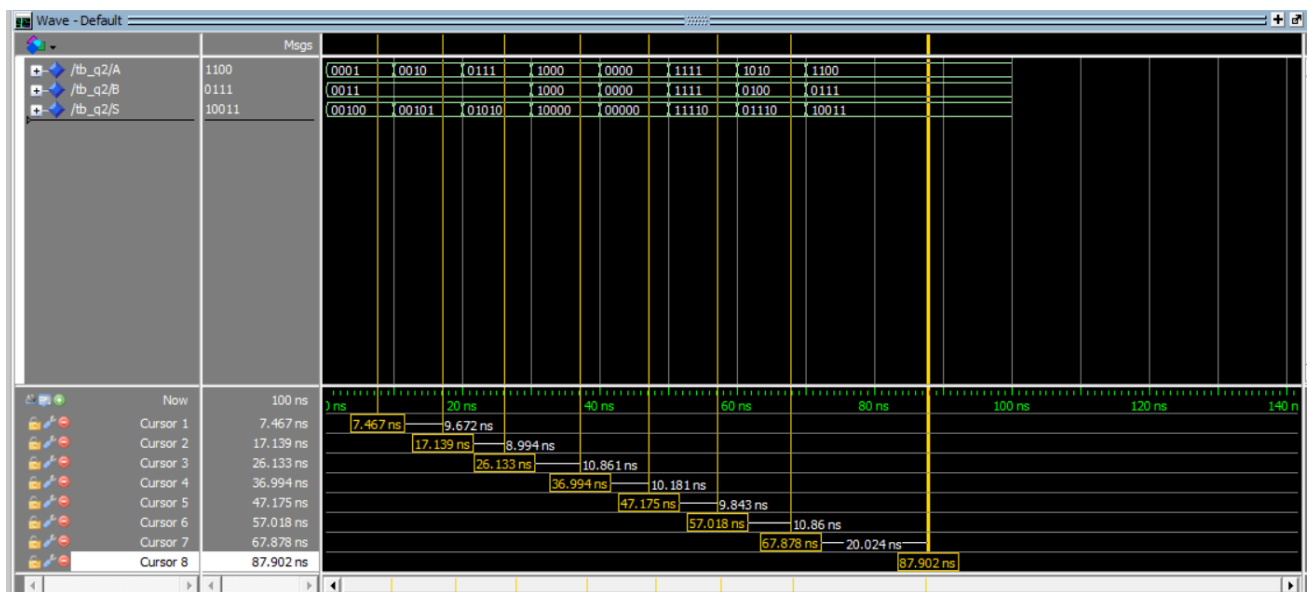
-O código documentado está no segundo arquivo

-Confirmação do código compilado:

Project - C:/Users/gibin/OneDrive/Desktop/Lab SD/Model Sim exp5/Exp5					
Name	Status	Type	Order	Modified	
Q1(5).vhd	✓	VHDL	0	04/26/2024 03:50:02 ...	
tb_Q1(5).vhd	✓	VHDL	1	04/28/2024 02:32:54 ...	
Q2(5).vhd	✓	VHDL	2	04/28/2024 02:30:45 ...	
somador.vhd	✓	VHDL	5	04/25/2024 05:26:07 ...	
tb_Q2(5).vhd	✓	VHDL	3	04/28/2024 02:35:55 ...	
Q3(5).vhd	✗	VHDL	4	04/25/2024 05:09:22 ...	

```
# Compile of Q1(5).vhd was successful.
# Compile of tb_Q1(5).vhd was successful.
# Compile of Q2(5).vhd was successful.
# Compile of tb_Q2(5).vhd was successful.
# Compile of Q3(5).vhd failed with 1 errors.
# Compile of somador.vhd was successful.
# 6 compiles, 1 failed with 1 error.
```

-Simulação do código no ModelSim:



-OBS: Para facilitar a verificação da correção, o testbench da questão 2 foi desenvolvido de maneira idêntica ao da questão 1. Portanto, a análise dos resultados é igual.

-Todos os casos foram vistos, e o seguinte foi analisado:

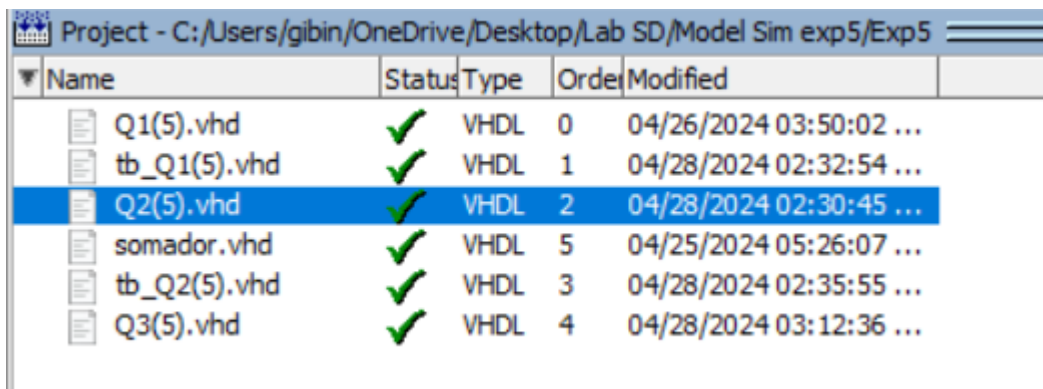
Cursor 1)	A='0001', B='0011', S='00100';	1+3=4
Cursor 2)	A='0010', B='0011', S='00101';	2+3=5
Cursor 3)	A='0111', B='0011', S='01010';	7+3=10
Cursor 4)	A='1000', B='1000', S='10000';	8+8=16
Cursor 5)	A='0000', B='0000', S='00000';	0+0=0
Cursor 6)	A='1111', B='1111', S='11110';	15+15=30
Cursor 7)	A='1010', B='0100', S='01110';	10+4=14
Cursor 8)	A='1100', B='0111', S='10011';	12+7=19

### QUESTÃO 03

Escrever em VHDL e simular no ModelSim um testbench para simular e testar o somador de palavras de 4 bits desenvolvido no visto 1. Esse testbench deve gerar todas as 256 combinações possíveis de valores para A e B, aguardando 500 ns entre combinações e, para cada combinação, comparar a saída do somador do visto 1 (utilizado aqui como device under test ou DUT) com a saída do somador do visto 2 (utilizado aqui como golden model), imprimindo uma mensagem de erro se as saídas não concordarem

**-O código documentado está no segundo arquivo**

**-Confirmação do código compilado:**



Name	Status	Type	Order	Modified
Q1(5).vhd	✓	VHDL	0	04/26/2024 03:50:02 ...
tb_Q1(5).vhd	✓	VHDL	1	04/28/2024 02:32:54 ...
Q2(5).vhd	✓	VHDL	2	04/28/2024 02:30:45 ...
somador.vhd	✓	VHDL	5	04/25/2024 05:26:07 ...
tb_Q2(5).vhd	✓	VHDL	3	04/28/2024 02:35:55 ...
Q3(5).vhd	✓	VHDL	4	04/28/2024 03:12:36 ...

```
# Compile of Q3(5).vhd was successful.
# Compile of Q1(5).vhd was successful.
# Compile of tb_Q1(5).vhd was successful.
# Compile of Q2(5).vhd was successful.
# Compile of tb_Q2(5).vhd was successful.
# Compile of Q3(5).vhd was successful.
# Compile of somador.vhd was successful.
# 6 compiles, 0 failed with no errors.
```

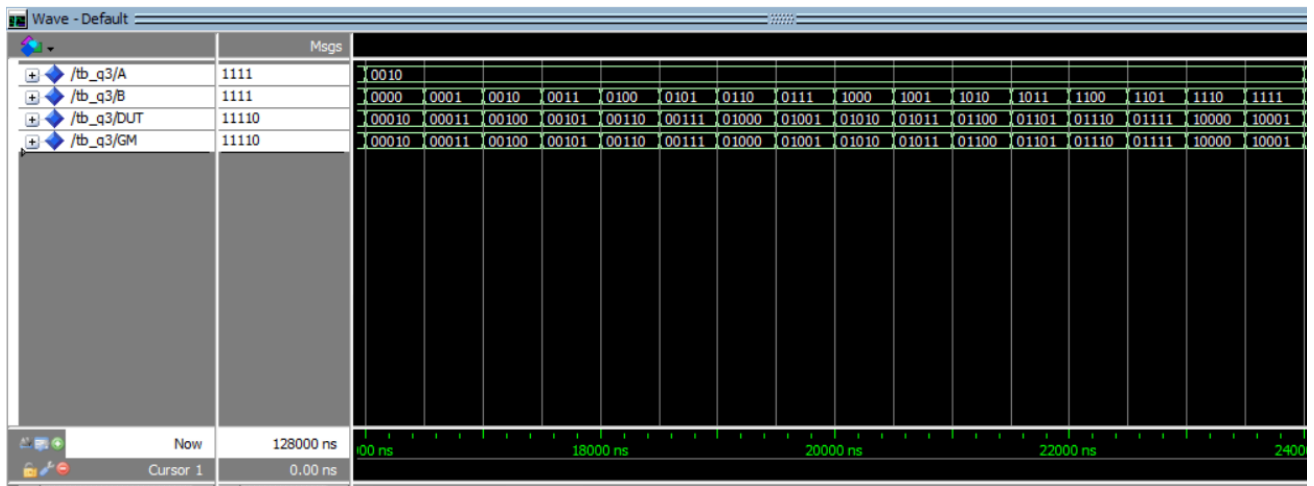
### -Simulação do código no ModelSim:



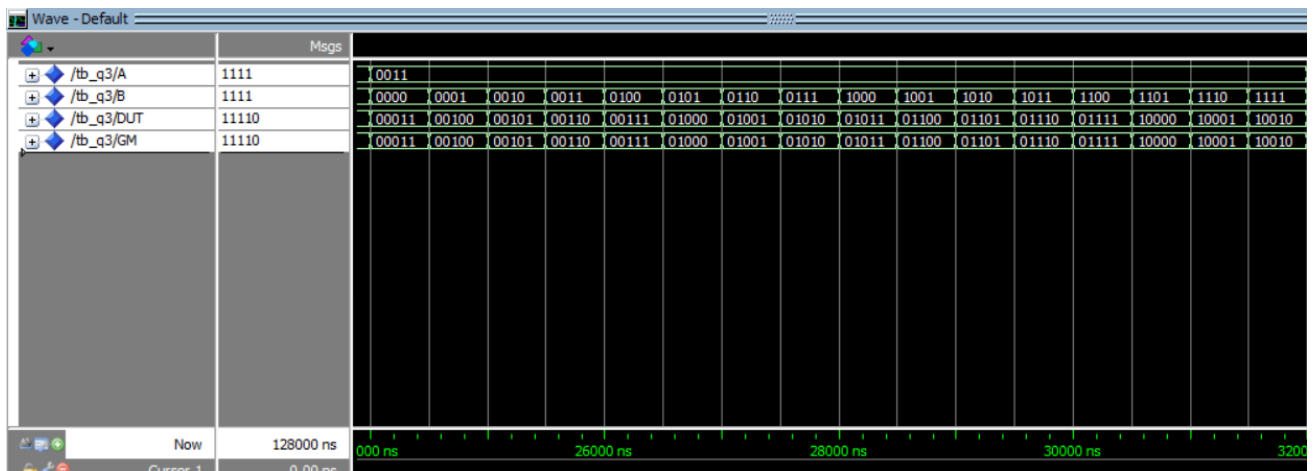
Trecho referente às combinações onde A= '0000' e B varia



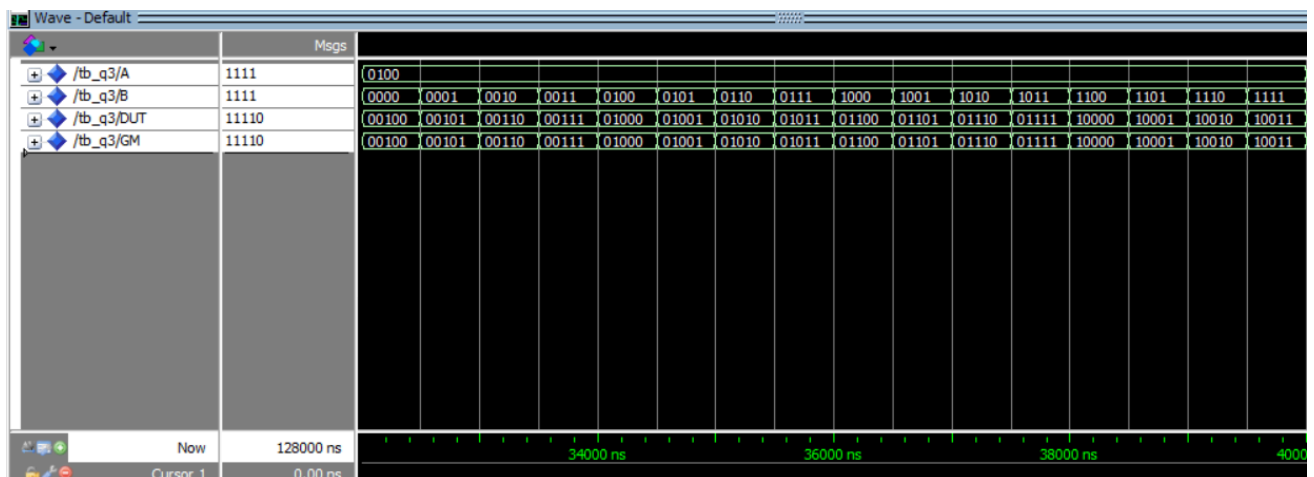
Trecho referente às combinações onde A= '0001' e B varia



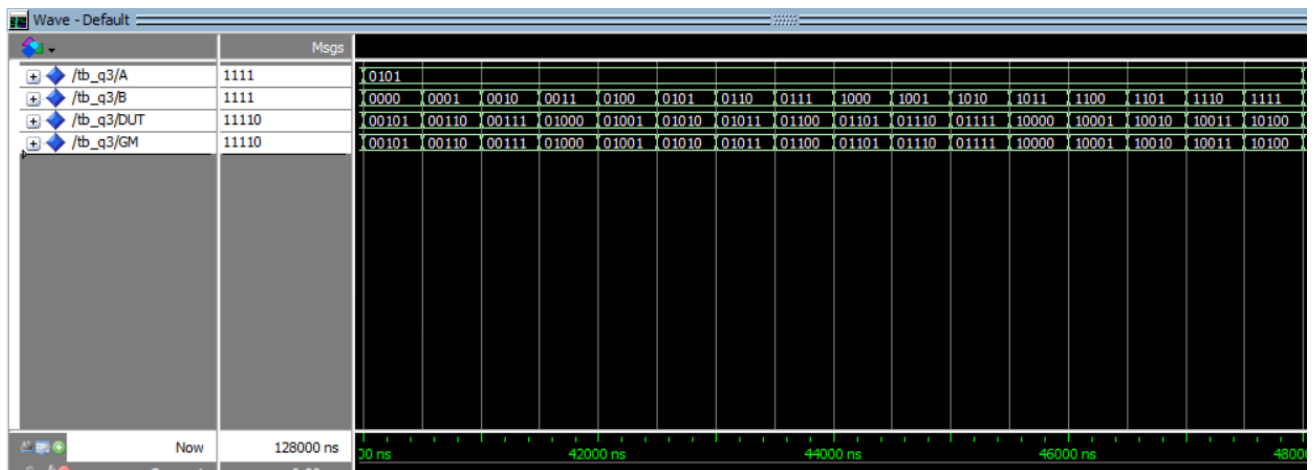
Trecho referente às combinações onde A= '0010' e B varia



Trecho referente às combinações onde A= '0011' e B varia



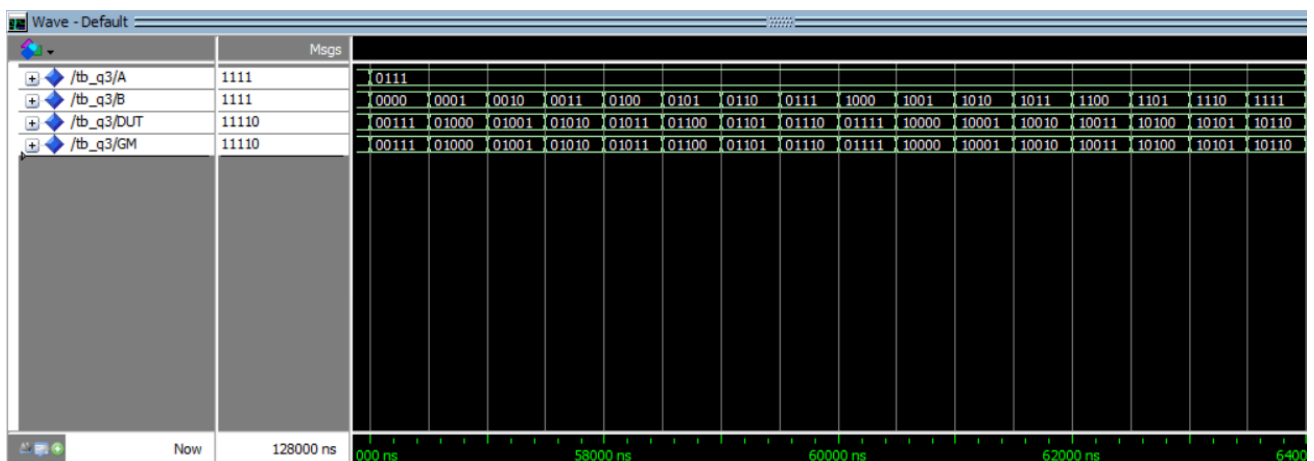
Trecho referente às combinações onde A= '0100' e B varia



Trecho referente às combinações onde A= '0101' e B varia

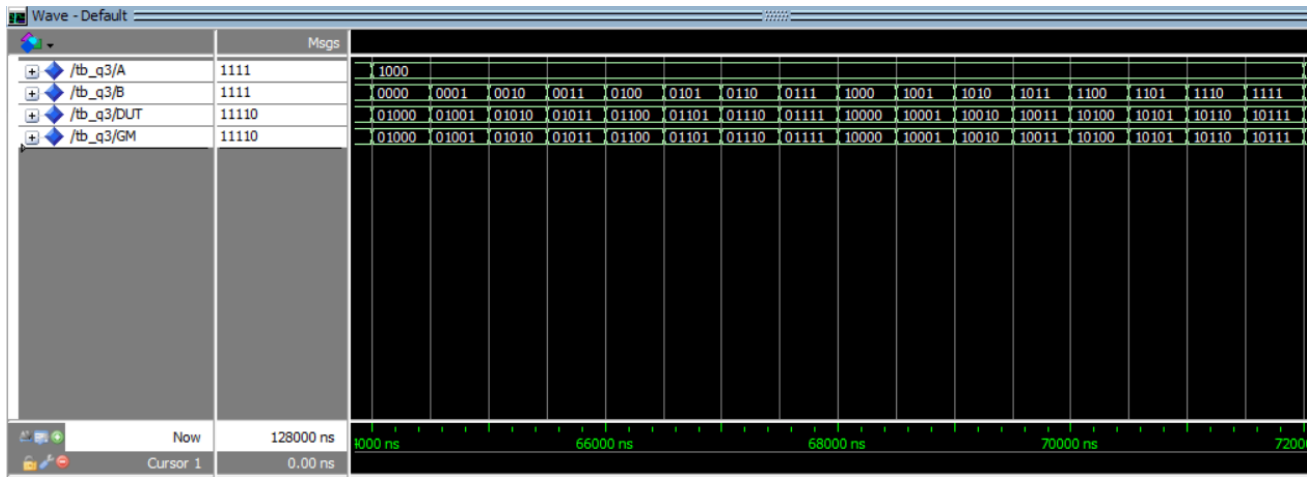


Trecho referente às combinações onde A= '0110' e B varia



Trecho referente às combinações onde A= '0111' e B varia





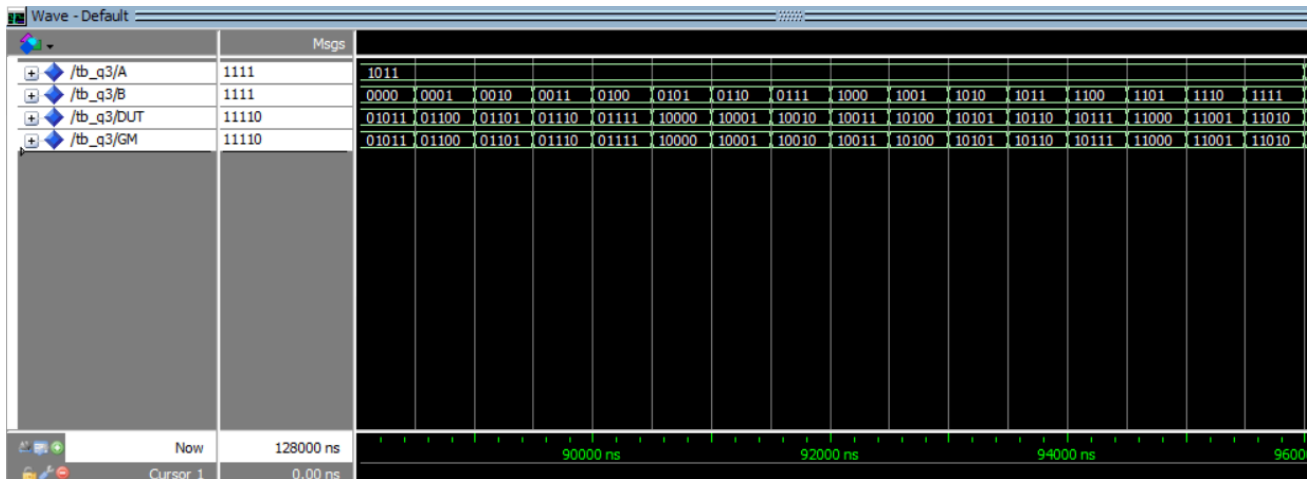
Trecho referente às combinações onde A= '1000' e B varia



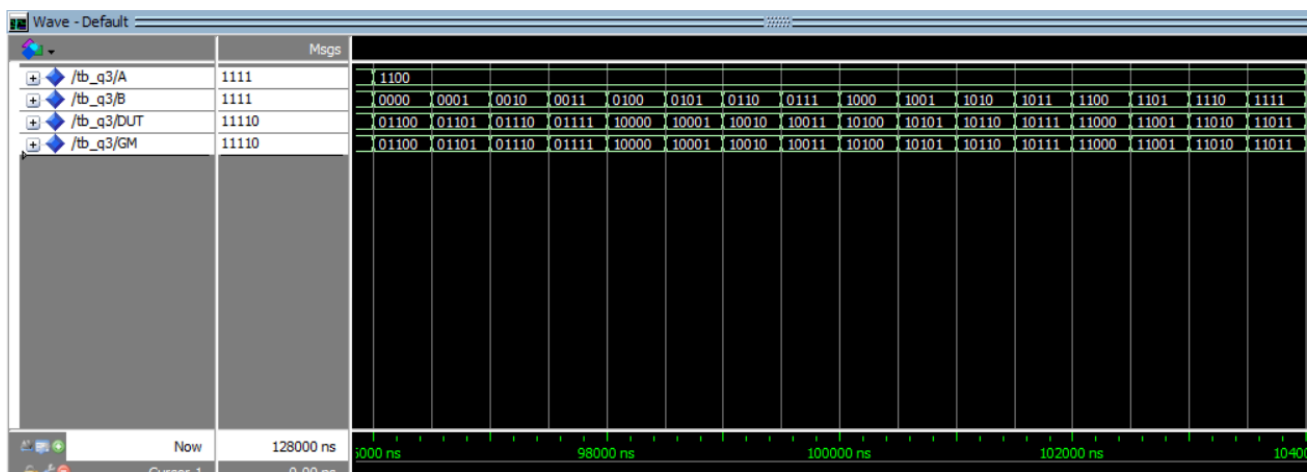
Trecho referente às combinações onde A= '1001' e B varia



Trecho referente às combinações onde A= '1010' e B varia



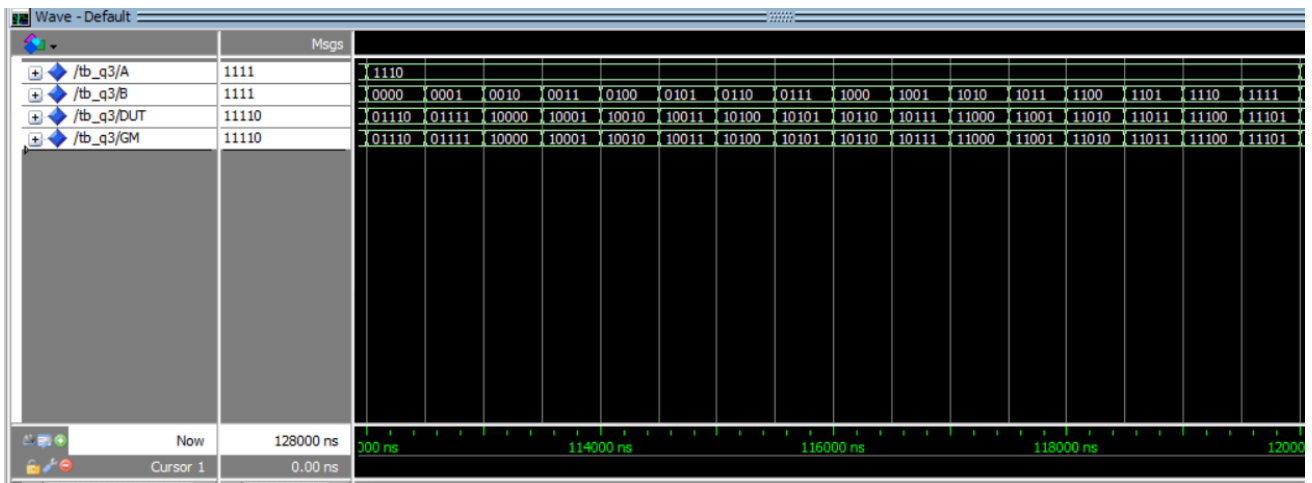
Trecho referente às combinações onde A= '1011' e B varia



Trecho referente às combinações onde A= '1100' e B varia



Trecho referente às combinações onde A= '1101' e B varia



Trecho referente às combinações onde A= '1110' e B varia



Trecho referente às combinações onde A= '1111' e B varia

-Portanto, vemos que em todas as combinações DUT e GM tiveram o mesmo resultado.

### 3. CONCLUSÃO

Após todos os procedimentos, houve uma compreensão mais sólida acerca dos conceitos básicos de design digital usando VHDL. Nesse experimento em particular, construímos um somador completo de 2 jeitos e depois os comparamos. Ficou mais claro como um componente deve ser usado, tal que a maior dificuldade do experimento passado foi solucionada. O maior obstáculo encontrado no experimento foi como fazer todas as combinações possíveis para a questão 3, bem como realizar a comparação entre duas saídas no testbench. Por fim, a experiência foi proveitosa ao aprofundar os componentes que foram introduzidos no experimento passado, dessa vez com o foco no somador completo.