



Aluna: Mariana Soares Oliveira
Matrícula: 231013663
Turma 01
30/12/2024

Relatório Experimento 5

1. Introdução

O intuito do seguinte experimento é desenvolver três somadores de palavras, cada um usando uma abordagem diferente (usual, com o operador operador '+' do pacote STD_LOGIC_ARITH e top model, respectivamente), utilizando a linguagem de descrição de hardware VHDL, e simular o seu comportamento por meio de *testbenchs* realizados no *software* ModelSim.

2. Teoria

1. Somador Completo

O somador de palavra de 4 bits é formado ao conectar vários somadores completos para trabalhar com números binários maiores. Cada somador completo soma três bits: dois de entrada e um "carry in" (vem-um). O resultado dessa soma é um bit de saída (soma) e um "carry out" (vai-um), que é passado para o próximo somador na cadeia.

Essa estrutura pode ser ampliada para somar números binários de qualquer tamanho, apenas aumentando o número de somadores conectados em sequência. Por exemplo, em um somador de 4 bits, quatro somadores completos trabalham juntos: o "carry out" de um alimenta o "carry in" do seguinte, garantindo que o cálculo da soma seja correto.

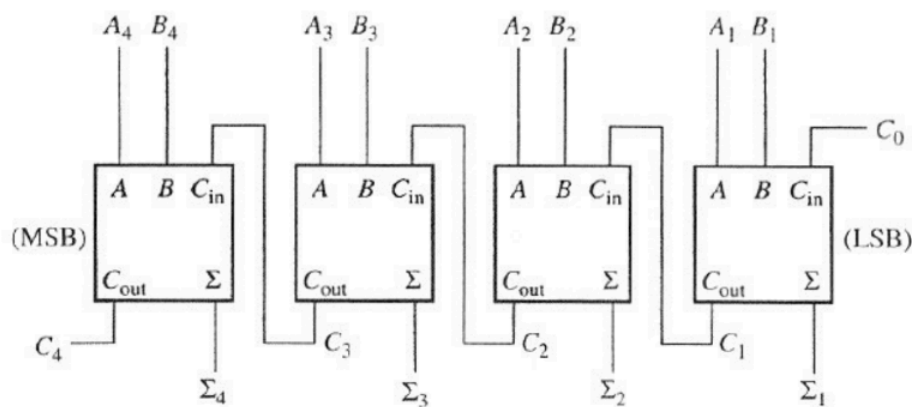


Figura 1: Somador de palavras de 4 bits.

2. Modelo em “U” invertido

O modelo U invertido é uma forma organizada de testar circuitos em VHDL. Ele é chamado assim porque lembra a letra "U" invertida, onde os testes começam em uma extremidade e os resultados são verificados na outra. Esse modelo é formado por três partes principais:

1. **Testbench:** É o arquivo de teste que gera as entradas para o circuito e verifica as saídas. Ele automatiza o processo, simulando vários casos de uso.
2. **Device Under Test (DUT):** É o circuito que está sendo avaliado. Ele é o projeto desenvolvido e que se deseja testar.
3. **Golden Model:** É o modelo de referência, ou seja, o ideal. Ele calcula os resultados esperados com base nas mesmas entradas enviadas para o DUT e serve como padrão para comparar as saídas geradas.

Essas três partes trabalham juntas no **Top Model**, que organiza os testes e faz a validação de forma automatizada. Essa estrutura é muito usada porque garante que o circuito funcione corretamente antes de ser implementado fisicamente.

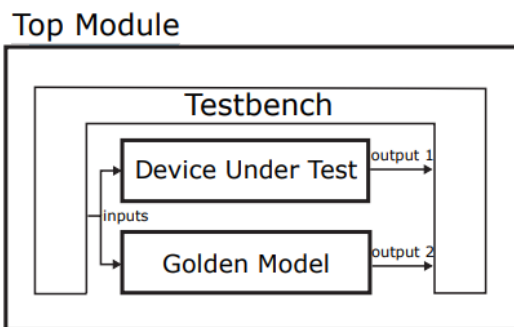


Figura 2: Organização do modelo U invertido

3. Códigos

Neste experimento utilizamos a linguagem de descrição de hardware VHDL por meio do software Modelsim para desenvolver três somadores de palavras conforme as figuras 3 até 6. Posteriormente, foi desenvolvido um código auxiliar chamado *testbench* para cada circuito, descrito nas figuras 7, 8 e 9.

```
C:\Users\maria\Desktop\231013663_Projeto5\questao1\questao1.vhd - Default
Ln#
1  -- biblioteca
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4  -- entidade
5  entity questao1 is
6  Port (
7      A : in STD_LOGIC_VECTOR (3 downto 0);
8      B : in STD_LOGIC_VECTOR (3 downto 0);
9      S : out STD_LOGIC_VECTOR (4 downto 0)
10 );
11 end questao1;
12 -- arquitetura
13 architecture rtl of questao1 is
14 component ent_somador
15 Port (
16     A, B, Cin : in STD_LOGIC;
17     S, Cout : out STD_LOGIC;
18 )
19 end component;
20 signal aux : STD_LOGIC_VECTOR (2 downto 0) := "000";
21 begin
22     sc1: ent_somador port map (A => A(0), B => B(0), Cin => '0', S => S(0), Cout => aux(0));
23     sc2: ent_somador port map (A => A(1), B => B(1), Cin => aux(0), S => S(1), Cout => aux(1));
24     sc3: ent_somador port map (A => A(2), B => B(2), Cin => aux(1), S => S(2), Cout => aux(2));
25     sc4: ent_somador port map (A => A(3), B => B(3), Cin => aux(2), S => S(3), Cout => S(4));
26 end rtl;
```

Figura 3. Codificação da questão 1



```
C:/Users/maria/Desktop/231013663_Projeto5/questao1/somador.vhd - Default
Ln#
1  -- biblioteca
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4
5  --entidade
6  entity ent_somador is
7  Port (
8      A : in STD_LOGIC;
9      B : in STD_LOGIC;
10     Cin : in STD_LOGIC;
11     S : out STD_LOGIC;
12     Cout : out STD_LOGIC
13 );
14 end ent_somador;
15
16 -- arquitetura
17 architecture arch_somador of ent_somador is
18 begin
19     S <= A xor B xor Cin;
20     Cout <= (A and B) or (A and Cin) or (B and Cin);
21 end arch_somador;
22
23
```

Figura 4. Codificação do Somador Completo

```
C:/Users/maria/Desktop/231013663_Projeto5/questao2/questao2.vhd - Default
Ln#
1  -- biblioteca
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4  use IEEE.STD_LOGIC_ARITH.ALL;
5  -- entidade
6  entity questao2 is
7  port (
8      A, B : in std_logic_vector (3 downto 0);
9      S : out std_logic_vector (4 downto 0)
10 );
11 end questao2;
12 -- arquitetura
13 architecture rtl of questao2 is
14 begin
15     S <= unsigned ('0' & A) + unsigned ('0' & B);
16 end rtl;
```

Figura 5. Codificação da questão 2



```
C:/Users/maria/Desktop/231013663_Projeto5/questao3/top_module.vhd - Default
Ln#
1  -- biblioteca
2  library ieee;
3  use ieee.std_logic_1164.all;
4  use ieee.numeric_std.all;
5  -- entidade
6  entity top_module is end;
7  -- arquitetura
8  architecture tp_arch of top_module is
9      -- componentes
10     component questao1 is
11     port (
12         A : in std_logic_vector(3 downto 0);
13         B : in std_logic_vector(3 downto 0);
14         S : out std_logic_vector(4 downto 0)
15     );
16     end component;
17     component questao2 is
18     port (
19         A : in std_logic_vector(3 downto 0);
20         B : in std_logic_vector(3 downto 0);
21         S : out std_logic_vector(4 downto 0)
22     );
23     end component;
24     component tb_questao3 is
25     port (
26         A : out std_logic_vector(3 downto 0);
27         B : out std_logic_vector(3 downto 0);
28         dut : in std_logic_vector(4 downto 0);
29         gm : in std_logic_vector(4 downto 0)
30     );
31     end component;
32     -- sinais auxiliares
33     signal A, B :std_logic_vector(3 downto 0);
34     signal S_dut, S_gm :std_logic_vector(4 downto 0);
35     begin
36         U0: questao1 PORT MAP(A, B, S_dut);
37         U1: questao2 PORT MAP(A, B, S_gm);
38         U3: tb_questao3 PORT MAP(A, B, S_dut, S_gm);
39     end tp_arch;
40
```

Figura 6. Codificação do *Top Model* (questão 3)



```
C:/Users/maria/Desktop/231013663_Projeto5/questao1/tb_questao1.vhd - Default
Ln#
1  -- biblioteca
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4  -- entidade
5  entity tb_questao1 is
6  end tb_questao1;
7  -- arquitetura
8  architecture ul of tb_questao1 is
9      component questao1
10         Port (
11             A : in STD_LOGIC_VECTOR (3 downto 0);
12             B : in STD_LOGIC_VECTOR (3 downto 0);
13             S : out STD_LOGIC_VECTOR (4 downto 0)
14         );
15     end component;
16     signal as : STD_LOGIC_VECTOR (3 downto 0) := "0000";
17     signal bs : STD_LOGIC_VECTOR (3 downto 0) := "0000";
18     signal s_out : STD_LOGIC_VECTOR(4 downto 0);
19 begin
20     ql : questao1 port map (A => as, B => bs, S => s_out);
21     as(0) <= not as(0) after 1 ns;
22     as(1) <= not as(1) after 2 ns;
23     as(2) <= not as(2) after 4 ns;
24     as(3) <= not as(3) after 8 ns;
25     bs(0) <= not bs(0) after 16 ns;
26     bs(1) <= not bs(1) after 32 ns;
27     bs(2) <= not bs(2) after 64 ns;
28     bs(3) <= not bs(3) after 128 ns;
29 end ul;
30
```

Figura 7. *Testbench* da questão 1



```
C:/Users/maria/Desktop/231013663_Projeto5/questao2/tb_questao2.vhd - Default
Ln#
1  |-- biblioteca
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4  -- entidade
5  entity tb_questao2 is end;
6  -- arquitetura
7  architecture ul of tb_questao2 is
8      component questao2
9      Port (
10         A : in STD_LOGIC_VECTOR (3 downto 0);
11         B : in STD_LOGIC_VECTOR (3 downto 0);
12         S : out STD_LOGIC_VECTOR (4 downto 0)
13     );
14     end component;
15
16     signal as : STD_LOGIC_VECTOR (3 downto 0) := "0000";
17     signal bs : STD_LOGIC_VECTOR (3 downto 0) := "0000";
18     signal y_out : STD_LOGIC_VECTOR (4 downto 0);
19
20 begin
21
22     q2 : questao2 port map (A => as, B => bs, S => y_out);
23
24     as(0) <= not as(0) after 1 ns;
25     as(1) <= not as(1) after 2 ns;
26     as(2) <= not as(2) after 4 ns;
27     as(3) <= not as(3) after 8 ns;
28     bs(0) <= not bs(0) after 16 ns;
29     bs(1) <= not bs(1) after 32 ns;
30     bs(2) <= not bs(2) after 64 ns;
31     bs(3) <= not bs(3) after 128 ns;
32
33 end ul;
34
```

Figura 8. *Testbench* da questão 2



```
1  -- biblioteca
2  library ieee;
3  use ieee.std_logic_1164.all;
4  use ieee.numeric_std.all;
5  use std.textio.all;
6  -- entidade
7  entity tb_questao3 is
8  port (
9      A : out std_logic_vector(3 downto 0);
10     B : out std_logic_vector(3 downto 0);
11     dut : in std_logic_vector(4 downto 0);
12     gm : in std_logic_vector(4 downto 0)
13 );
14 end tb_questao3;
15 -- arquitetura
16 architecture testbench_arch of tb_questao3 is
17     -- sinais para monitorar entradas e saídas
18     signal A_reg, B_reg : std_logic_vector(3 downto 0);
19     signal S_dut_reg, S_gm_reg : std_logic_vector(4 downto 0);
20     -- função auxiliar para conversão de std_logic_vector para string
21     function to_string(signal_value: std_logic_vector) return string is
22         variable result: string(1 to signal_value'length);
23     begin
24         for i in signal_value'range loop
25             result(i + 1) := character'value(std_ulogic'image(signal_value(i)));
26         end loop;
27         return result;
28     end function;
29 begin
30     -- mapeamento dos sinais
31     A <= A_reg;
32     B <= B_reg;
33     S_dut_reg <= dut;
34     S_gm_reg <= gm;
35     -- processo de teste
36     process
37     begin
38         report "Iniciando teste..." severity NOTE;
39         for i in 0 to 15 loop
40             A_reg <= std_logic_vector(to_unsigned(i, 4));
41             for j in 0 to 15 loop
42                 B_reg <= std_logic_vector(to_unsigned(j, 4));
43                 wait for 500 ns;
44                 -- comparação dos resultados
45                 assert (S_dut_reg = S_gm_reg)
46                     report "Erro: A=" & integer'image(to_integer(unsigned(A_reg))) &
47                         " B=" & integer'image(to_integer(unsigned(B_reg))) &
48                         " DUT=" & to_string(S_dut_reg) & " GM=" & to_string(S_gm_reg)
49                     severity ERROR;
50             end loop;
51         end loop;
52         report "Teste finalizado!" severity NOTE;
53         wait;
54     end process;
55 end testbench_arch;
```

Figura 9. Testbench da questão 3

4. Compilação

Os códigos gerados anteriormente foram submetidos a uma compilação com o intuito de garantir seu funcionamento, como mostrado nas figuras 5 e 6 ambos os códigos não apresentam erros de sintaxe.



The screenshot shows the ModelSim interface for a project named 'Questao1'. The top window displays a file list with the following data:

Name	Status	Type	Order	Modified
somador.vhd	✓	VHDL	2	12/30/2024 09:04:03 ...
tb_questao1.vhd	✓	VHDL	1	01/02/2025 08:37:36 ...
questao1.vhd	✓	VHDL	0	01/02/2025 08:38:20 ...

The bottom window, titled 'Transcript', shows the following text:

```
# Reading pref.tcl
project open C:/Users/maria/Desktop/231013663_Projeto5/questao1/Questao1
# Loading project Questao1
# Compile of questao1.vhd was successful.
# Compile of tb_questao1.vhd was successful.
# Compile of somador.vhd was successful.
# 3 compiles, 0 failed with no errors.

ModelSim>
```

Figura 10. Compilação da questão 1

The screenshot shows the ModelSim interface for a project named 'Questao2'. The top window displays a file list with the following data:

Name	Status	Type	Order	Modified
questao2.vhd	✓	VHDL	0	12/30/2024 10:4...
tb_questao2.vhd	✓	VHDL	1	12/30/2024 10:4...

The bottom window, titled 'Transcript', shows the following text:

```
# Reading pref.tcl
project open C:/Users/maria/Desktop/231013663_Projeto5/questao2/Questao2
# Loading project Questao2
# Compile of questao2.vhd was successful.
# Compile of tb_questao2.vhd was successful.
# 2 compiles, 0 failed with no errors.

ModelSim>
```

Figura 11. Compilação da questão 2

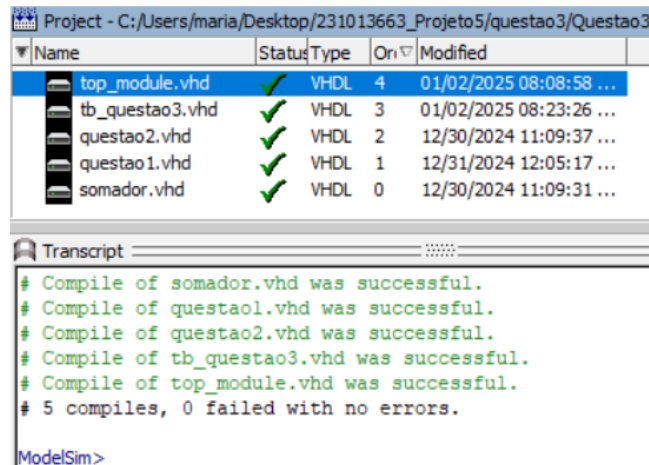


Figura 12. Compilação da questão 3

5. Simulação

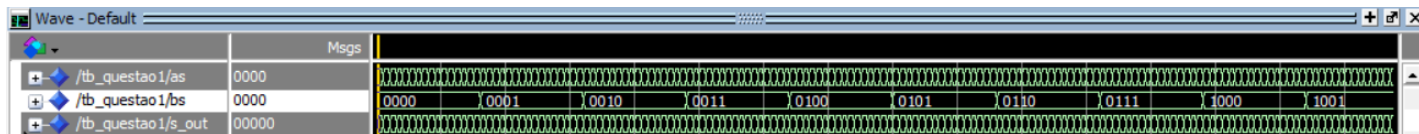


Figura 13. Simulação de onda do banco de testes da questão 1

- Cursor 1 (em 0.00 ns): A = 0000 , B = 0000 , S = 00000 , aux = 000
- Cursor 2 (em 4.00 ns): A = 0100 , B = 0000 , S = 00100 , aux = 000
- Cursor 3 (em 18.00 ns): A = 0010 , B = 0001 , S = 00011 , aux = 000
- Cursor 4 (em 27.00 ns): A = 1011 , B = 0001 , S = 01100 , aux = 011
- Cursor 5 (em 39.00 ns): A = 0111 , B = 0010 , S = 01001 , aux = 110
- Cursor 6 (em 43.00 ns): A = 1011 , B = 0010 , S = 01101 , aux = 010
- Cursor 7 (em 79.601 ns): A = 1111 , B = 0100 , S = 10011 , aux = 100
- Cursor 8 (em 95.00 ns): A = 1111 , B = 0101 , S = 10100 , aux = 111

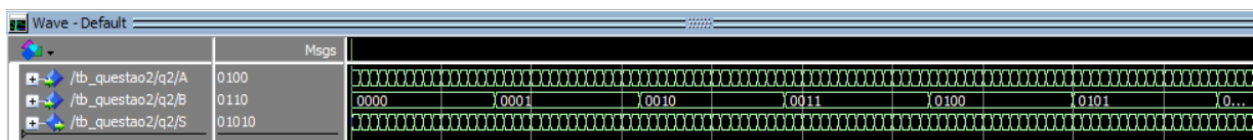


Figura 14. Simulação de onda do banco de testes da questão 2

- Cursor 1 (em 0.00 ns): A = 0000 , B = 0000 , S = 00000
- Cursor 2 (em 8.00 ns): A = 1000 , B = 0000 , S = 01000
- Cursor 3 (em 17.00 ns): A = 0001 , B = 0001 , S = 00010
- Cursor 4 (em 36.00 ns): A = 0100 , B = 0010 , S = 00110
- Cursor 5 (em 54.00 ns): A = 0110 , B = 0011 , S = 01001



- **Cursor 6 (em 64.00 ns): A = 0000 , B = 0100 , S = 00100**
- **Cursor 7 (em 81.00 ns): A = 0001 , B = 0101 , S = 00110**
- **Cursor 8 (em 123.662 ns): A = 1011 , B = 0111 , S = 10010**

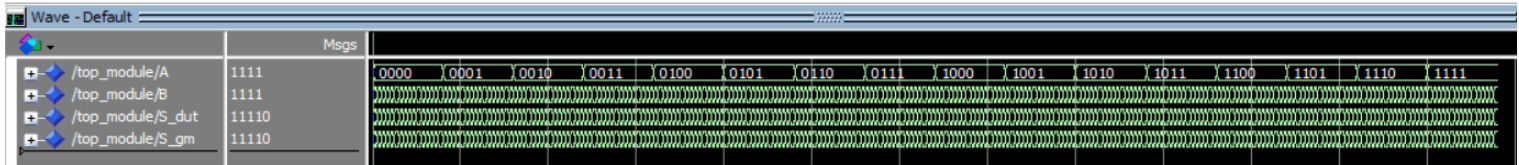


Figura 15. Simulação de onda do banco de testes da questão 3

```
VSIM 4> run -all
# ** Note: Iniciando teste...
#   Time: 0 ps  Iteration: 0  Instance: /top_module/U3
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#   Time: 0 ps  Iteration: 0  Instance: /top_module/U1
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#   Time: 0 ps  Iteration: 0  Instance: /top_module/U1
# ** Note: Teste finalizado!
#   Time: 128 us  Iteration: 0  Instance: /top_module/U3
```

Figura 16. Transcript da questão 3

- **Cursor 1 (em 0.00 ns): A = 0000 , B = 0000 , S_dut = 00000, S_gm = 00000**
- **Cursor 2 (em 5000.00 ns): A = 0000 , B = 1010 , S_dut = 01010, S_gm = 01010**
- **Cursor 3 (em 11000.00 ns): A = 0001 , B = 0110 , S_dut = 00111, S_gm = 00111**
- **Cursor 4 (em 18000.00 ns): A = 0010 , B = 0100 , S_dut = 00110, S_gm = 00110**
- **Cursor 5 (em 55500.00 ns): A = 0110 , B = 1111, S_dut = 10101, S_gm = 10101**
- **Cursor 6 (em 66500.00 ns): A = 1000 , B = 0101 , S_dut = 01101, S_gm = 01101**
- **Cursor 7 (em 91000.00 ns): A = 1011 , B = 0110 , S_dut = 10001, S_gm = 10001**
- **Cursor 8 (em 123000.00 ns): A = 1111 , B = 0110 , S_dut = 10101, S_gm = 10101**

6. Análise

Neste experimento, foram analisados três somadores de palavras, com três abordagens diferentes. Com base nas figuras 1 e 2 e na simulação de de onda apresentada nas Figuras 13, 14, 15 e 16, é possível compreender o funcionamento esperado de todos os circuitos. Dessa forma, pode-se afirmar que os códigos desenvolvidos correspondem ao comportamento esperado.



7. Conclusão

No experimento, foi possível descrever o comportamento das estruturas propostas e entender suas características. As simulações geraram os dados esperados. Não houveram erros ou divergências observados durante a realização do experimento.