



Aluna: Mariana Soares Oliveira  
Matrícula: 231013663  
Turma 01  
28/11/2024

## Relatório Experimento 4

### 1. Introdução

O intuito do seguinte experimento é desenvolver dois multiplexadores de 4 entradas (questão 1), um decodificador de 4 para 16 e um multiplexador de 8 entradas (questão 2) – usando como base entidades desenvolvidas em relatórios anteriores – para as funções booleanas descritas na **Teoria**, utilizando a linguagem de descrição de hardware VHDL, e simular o seu comportamento por meio de *testbenchs* realizados no *software* ModelSim.

### 2. Teoria

#### 2.1 Multiplexador

Um multiplexador é um dispositivo combinacional que seleciona uma de várias entradas de dados para transmitir como saída, com base nos sinais de controle. Um multiplexador de 4 entradas, possui 4 linhas de entrada, 2 linhas de seleção e uma linha de saída. Sua operação pode ser representada pela função:

$$Y = I_0 \cdot \overline{S_1} \cdot \overline{S_0} + I_1 \cdot \overline{S_1} \cdot S_0 + I_2 \cdot S_1 \cdot \overline{S_0} + I_3 \cdot S_1 \cdot S_0$$

Já em um multiplexador de 8 entradas, o princípio é o mesmo, mas há 8 linhas de entrada ( $I_0$  até  $I_7$ ), 3 linhas de seleção ( $S_0$ ,  $S_1$  e  $S_2$ ) e 1 linha de saída. Assim, selecionando entre as 8 combinações possíveis.

#### 2.2 Decodificador

Um decodificador é um circuito combinacional que converte uma entrada binária de  $n$  bits em até  $2^n$  saídas. No caso de um decodificador 4 para 16, as 4 entradas determinam qual das 16 saídas será ativada (nível lógico 1), enquanto as outras permanecem em nível lógico 0. Ele é utilizado principalmente para identificar combinações únicas de entrada.

#### 2.3 Funções e Tabelas Verdade das Questões 1 e 2

Na primeira parte do experimento, será desenvolvido dois multiplexadores de 4 entradas e uma porta inversora para uma entidade com 3 bits de entrada ( $A$ ,  $B$  e  $C$ ) e 2 bits de saída ( $X$  e  $Y$ ), a partir das equações e Tabela Verdade abaixo:



$$X = \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot \bar{C} + A \cdot B \quad (1)$$

$$Y = \bar{A} \cdot \bar{B} + \bar{A} \cdot B \cdot \bar{C} + A \cdot B \cdot C \quad (2)$$

A	B	C	X	Y
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	1

Tabela Verdade 1

Já na segunda parte do experimento, será desenvolvido um decodificador 4 para 16, um multiplexador de 8 entradas e três portas ou para uma entidade com 7 bits de entrada (A, B, C, D, E, F e G) e 1 bit de saída (S) a partir das equações e Tabela Verdade abaixo:

$$S = F \cdot G + A \cdot B \cdot C \cdot D \cdot \bar{E} \cdot \bar{F} \cdot G + \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} \cdot \bar{E} \cdot \bar{F} \cdot G + A \cdot \bar{B} \cdot C \cdot E \cdot F \cdot \bar{G} + \bar{A} \cdot B \cdot C \cdot D \cdot \bar{E} \cdot F \cdot \bar{G} + A \cdot B \cdot C \cdot D \cdot E \cdot \bar{F} \cdot \bar{G} + A \cdot \bar{B} \cdot \bar{C} \cdot D \cdot E \cdot \bar{F} \cdot \bar{G} \quad (3)$$

A	B	C	D	E	F	G	S
0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1
0	1	1	1	0	1	0	1
0	1	1	1	1	0	0	0
1	0	1	0	1	1	0	1
1	1	1	1	0	0	1	1
1	1	1	1	1	1	0	0
1	0	0	1	1	0	0	1

Tabela Verdade 2

**OBSERVAÇÃO:** Como a Tabela Verdade para a Questão 2 é muito grande ( $2^7 = 128$  combinações), coloquei apenas algumas para que o relatório não fique muito extenso. A tabela completa pode ser encontrada no seguinte *link*:

[Tabela Verdade Completa - Questão 2 Experimento 4](#).



### 3. Códigos

Neste experimento utilizamos a linguagem de descrição de hardware VHDL por meio do software Modelsim para desenvolver dois multiplexadores de 4 entradas (questão 1), um decodificador de 4 para 16 e um multiplexador de 8 entradas (questão 2) conforme as figuras 1, 2, 3, 4 e 5. Posteriormente, foi desenvolvido um código auxiliar chamado *testbench* para cada circuito, descrito nas figuras 6 e 7.

```
C:/Users/maria/Desktop/231013663_Projeto4/questao1/mux4.vhd - Default
Ln#
1      ---- biblioteca
2      library IEEE;
3      use IEEE.STD_LOGIC_1164.ALL;
4
5      ---- entidade
6      entity ent_mux4 is
7      port (
8          D : in std_logic_vector (3 downto 0);
9          S : in std_logic_vector (1 downto 0);
10         Y : out std_logic
11     );
12 end ent_mux4;
13
14 ---- arquitetura
15 architecture arch_mux4 of ent_mux4 is
16 begin
17     process(D,S)
18     begin
19         if S = "00" then
20             Y <= D(0);
21         elsif S = "01" then
22             Y <= D(1);
23         elsif S = "10" then
24             Y <= D(2);
25         elsif S = "11" then
26             Y <= D(3);
27         end if;
28     end process;
29 end arch_mux4;
30
```

Figura 1. Codificação do Multiplexador 4x1



```
C:/Users/maria/Desktop/231013663_Projeto4/questao1/q1.vhd - Default
Ln#
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4  -- entidade
5  entity q1 is port (
6      A, B, C : in STD_LOGIC;
7      X, Y : out STD_LOGIC);
8  end q1;
9  -- arquitetura
10 architecture rtl of q1 is
11     component ent_mux4 is
12     port (
13         D : in std_logic_vector (3 downto 0);
14         S : in std_logic_vector (1 downto 0);
15         Y : out std_logic
16     );
17 end component;
18 signal aux_dx, aux_dy : std_logic_vector (3 downto 0);
19 signal aux_s : std_logic_vector (1 downto 0);
20 begin
21     cx : ent_mux4 port map (aux_dx, aux_s, X);
22     aux_dx(3) <= '1';
23     aux_dx(2) <= not C;
24     aux_dx(1) <= C;
25     aux_dx(0) <= '0';
26
27     aux_s(1) <= A;
28     aux_s(0) <= B;
29
30     cy : ent_mux4 port map (aux_dy, aux_s, Y);
31     aux_dy(3) <= C;
32     aux_dy(2) <= '0';
33     aux_dy(1) <= not C;
34     aux_dy(0) <= '1';
35 end rtl;
36
```

Figura 2. Codificação da Questão 1



```
C:/Users/maria/Desktop/231013663_Projeto4/questao2/deco4x16.vhd (/tb_q2/cs/u0) - Default :
Ln#
1      ---- biblioteca
2      library IEEE;
3      use IEEE.STD_LOGIC_1164.all;
4      ---- entidade
5      entity ent_deco4x16 is
6      port (
7          A: in STD_LOGIC_VECTOR (3 downto 0);
8          Y: out STD_LOGIC_VECTOR (15 downto 0)
9      );
10     end ent_deco4x16;
11     ---- arquitetura
12     architecture arch_deco4x16 of ent_deco4x16 is
13     begin
14         with A select
15             Y <= "0000000000000001" when "0000",
16                  "0000000000000010" when "0001",
17                  "0000000000000100" when "0010",
18                  "00000000000001000" when "0011",
19                  "0000000000010000" when "0100",
20                  "0000000000100000" when "0101",
21                  "0000000001000000" when "0110",
22                  "0000000010000000" when "0111",
23                  "0000000100000000" when "1000",
24                  "0000001000000000" when "1001",
25                  "0000010000000000" when "1010",
26                  "0000100000000000" when "1011",
27                  "0001000000000000" when "1100",
28                  "0010000000000000" when "1101",
29                  "0100000000000000" when "1110",
30                  "1000000000000000" when "1111",
31                  "0000000000000000" when others;
32     end arch_deco4x16;
```

Figura 3. Codificação do Decodificador 4x16

```
C:/Users/maria/Desktop/231013663_Projeto4/questao2/mux8x1.vhd (/tb_q2/cs/u1) - Default :
Ln#
1      ---- biblioteca
2      library IEEE;
3      use IEEE.STD_LOGIC_1164.all;
4      ---- entidade
5      entity ent_mux8x1 is port (
6          D: in STD_LOGIC_VECTOR (7 downto 0);
7          S: in STD_LOGIC_VECTOR (2 downto 0);
8          Y: out STD_LOGIC );
9      end ent_mux8x1;
10     ---- arquitetura
11     architecture arch_mux8x1 of ent_mux8x1 is
12     begin
13         Y <= D(0) when S = "000" else
14              D(1) when S = "001" else
15              D(2) when S = "010" else
16              D(3) when S = "011" else
17              D(4) when S = "100" else
18              D(5) when S = "101" else
19              D(6) when S = "110" else
20              D(7);
21     end arch_mux8x1;
```

Figura 4. Codificação do Multiplexador 8x1



```
C:/Users/maria/Desktop/231013663_Projeto4/questao2/q2.vhd (/tb_q2/cs) - Default
Ln#
1  ---- biblioteca
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4  ---- entidade
5  entity q2 is port (
6  | A, B, C, D, E, F, G: in STD_LOGIC;
7  | S: out STD_LOGIC);
8  | end q2;
9  ---- arquitetura
10 architecture rtl of q2 is
11 | --- declaração do mux8x1 (q1 exp 3)
12 | component ent_mux8x1 is port (
13 | | D: in STD_LOGIC_VECTOR (7 downto 0);
14 | | S: in STD_LOGIC_VECTOR (2 downto 0);
15 | | Y: out STD_LOGIC );
16 | | end component;
17 | --- declaração do decodificador (q2 exp 3)
18 | component ent_deco4x16 is
19 | | port (
20 | | | A: in STD_LOGIC_VECTOR (3 downto 0);
21 | | | Y: out STD_LOGIC_VECTOR (15 downto 0)
22 | | );
23 | | end component;
24 | signal aux_d : std_logic_vector (7 downto 0) := "00000000";
25 | signal aux_a : std_logic_vector (3 downto 0) := "0000";
26 | signal aux_y : std_logic_vector (15 downto 0) := x"0000";
27 | signal aux_s : std_logic_vector (2 downto 0) := "000";
28
29 | begin
30
31 | u0 : ent_deco4x16 port map (A => aux_a, Y => aux_y);
32 | u1 : ent_mux8x1 port map (D => aux_d, S => aux_s, Y => S);
33
34 | aux_a(3) <= A;
35 | aux_a(2) <= B;
36 | aux_a(1) <= C;
37 | aux_a(0) <= D;
38
39 | aux_d(7) <= '1';
40 | aux_d(6) <= aux_y(11) or aux_y(10);
41 | aux_d(5) <= '0';
42 | aux_d(4) <= aux_y(15) or aux_y(9);
43 | aux_d(3) <= '1';
44 | aux_d(2) <= aux_y(7);
45 | aux_d(1) <= aux_y(15) or aux_y(0);
46 | aux_d(0) <= '0';
47
48 | aux_s(2) <= E;
49 | aux_s(1) <= F;
50 | aux_s(0) <= G;
51
52 | end rtl;
```

Figura 5. Codificação da Questão 2



```
C:/Users/maria/Desktop/231013663_Projeto4/questao1/tb_q1.vhd - Default
Ln#
1  -- biblioteca
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4  -- entidade
5  entity tb_q1 is end;
6  -- arquitetura
7  architecture rtl of tb_q1 is
8      component q1 is
9          port (
10             A, B, C : in STD_LOGIC;
11             X, Y : out STD_LOGIC
12          );
13      end component;
14
15      signal aux_a, aux_b, aux_c : STD_LOGIC := '0';
16      signal aux_x, aux_y : STD_LOGIC;
17  begin
18      uut: q1 port map (
19          A => aux_a,
20          B => aux_b,
21          C => aux_c,
22          X => aux_x,
23          Y => aux_y
24      );
25
26      stimulus: process
27      begin
28          -- estímulos de entrada
29          aux_c <= '0'; aux_b <= '0'; aux_a <= '0';
30          wait for 1 ns;
31          aux_c <= '1'; aux_b <= '0'; aux_a <= '0';
32          wait for 2 ns;
33          aux_c <= '0'; aux_b <= '1'; aux_a <= '0';
34          wait for 4 ns;
35          aux_c <= '1'; aux_b <= '1'; aux_a <= '0';
36          wait for 8 ns;
37          aux_c <= '0'; aux_b <= '0'; aux_a <= '1';
38          wait for 16 ns;
39          aux_c <= '1'; aux_b <= '0'; aux_a <= '1';
40          wait for 32 ns;
41
42          aux_c <= '0'; aux_b <= '1'; aux_a <= '1';
43          wait for 64 ns;
44          aux_c <= '1'; aux_b <= '1'; aux_a <= '1';
45          wait for 128 ns;
46          wait;
47      end process;
48  end rtl;
```

Figura 6. Testbench da Questão 1



```
C:/Users/maria/Desktop/231013663_Projeto4/questao2/tb_q2.vhd (/tb_q2) - Default
Ln#
1  ---- biblioteca
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4  use IEEE.NUMERIC_STD.ALL;
5  ---- entidade
6  entity tb_q2 is
7  end tb_q2;
8  -- arquitetura
9  architecture rtl of tb_q2 is
10     -- Declaração do componente
11     component q2 is
12     port (
13         A, B, C, D, E, F, G: in STD_LOGIC;
14         S: out STD_LOGIC
15     );
16     end component;
17     -- sinais de estímulo
18     signal aux: STD_LOGIC_VECTOR(6 downto 0) := "0000000";
19     signal S: STD_LOGIC;
20     begin
21     -- instância da UUT
22     cs: q2
23     port map (
24         A => aux(6),
25         B => aux(5),
26         C => aux(4),
27         D => aux(3),
28         E => aux(2),
29         F => aux(1),
30         G => aux(0),
31         S => S
32     );
33     -- geração de estímulos
34     estimulo: process
35     begin
36         for i in 0 to 127 loop
37             wait for 5 ns;
38             aux <= std_logic_vector(to_unsigned(i, 7));
39         end loop;
40         wait;
41     end process;
42 end rtl;
```

Figura 7. Testbench da Questão 2

## 4. Compilação

Os códigos gerados anteriormente foram submetidos a uma compilação com o intuito de garantir seu funcionamento, como mostrado nas figuras 8 e 9 ambos os códigos não apresentam erros de sintaxe.



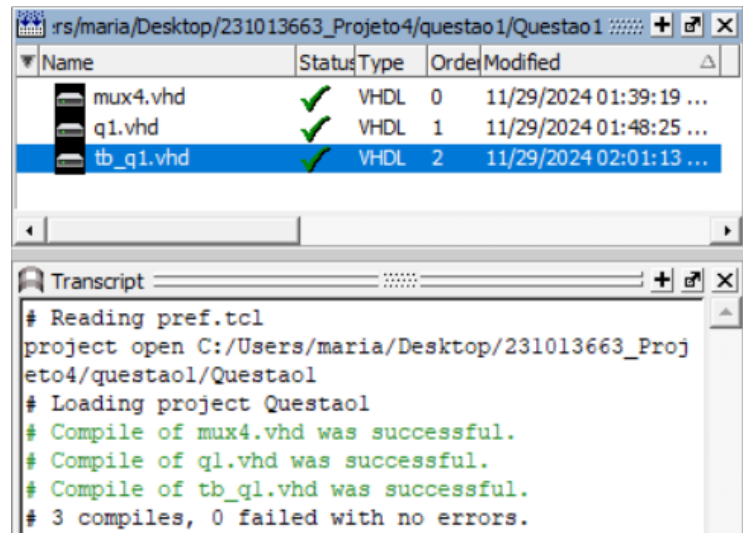


Figura 8. Compilação da Questão 1

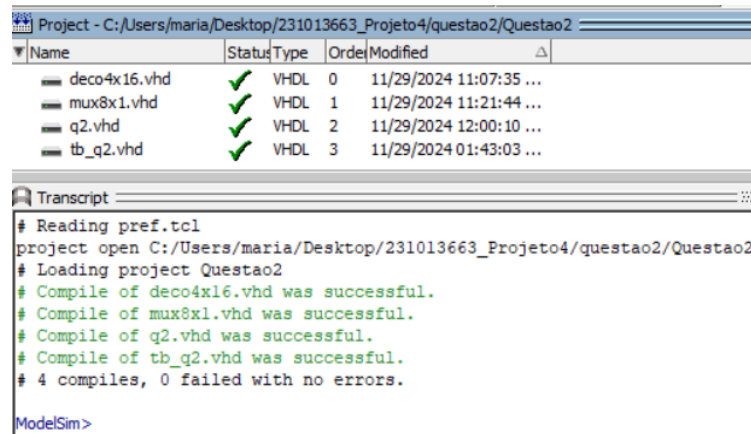


Figura 9. Compilação da Questão 2

## 5. Simulação

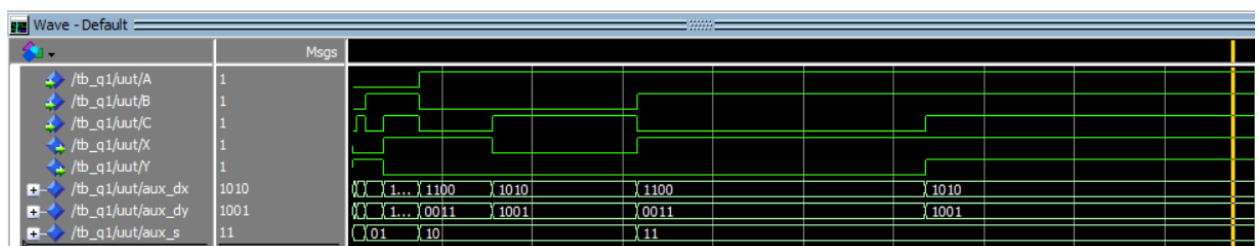


Figura 10. Simulação de onda do banco de testes da questão 1

- Cursor 1 (em 0.00 ns): A = 0 , B = 0 , C = 0 , X = 0 , Y = 0
- Cursor 2 (em 1.457 ns): A = 0 , B = 0 , C = 1 , X = 0 , Y = 1



- **Cursor 3 (em 3.547 ns):** A = 0 , B = 1 , C = 0 , X = 0 , Y = 1
- **Cursor 4 (em 8.833 ns):** A = 0 , B = 1 , C = 1 , X = 1 , Y = 0
- **Cursor 5 (em 16.992 ns):** A = 1 , B = 0 , C = 0 , X = 1 , Y = 0
- **Cursor 6 (em 32.174 ns):** A = 1 , B = 0 , C = 1 , X = 0 , Y = 0
- **Cursor 7 (em 64.986 ns):** A = 1 , B = 1 , C = 0 , X = 1 , Y = 0
- **Cursor 8 (em 128.944 ns):** A = 1 , B = 1 , C = 1 , X = 1 , Y = 1

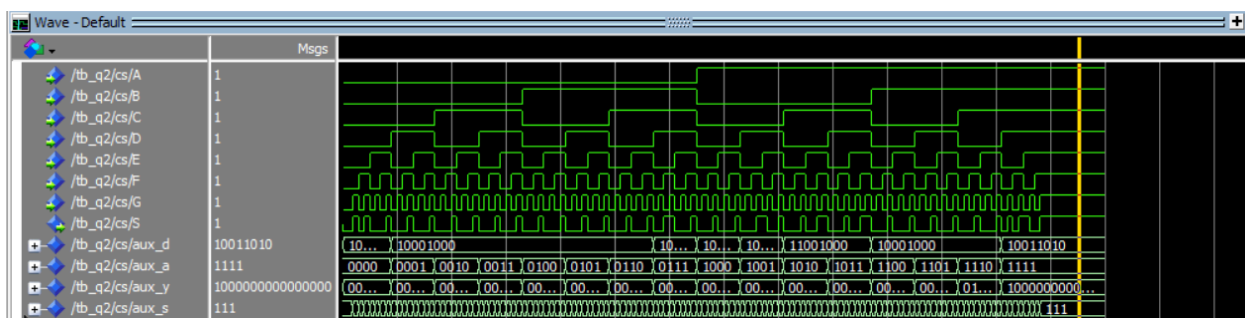


Figura 11. Simulação de onda do banco de testes da questão 2

- **Cursor 1 (em 4.541 ns):** A = 0 , B = 0 , C = 0 , D = 0 , E = 0 , F = 0 , G = 0 , S = 0;
- **Cursor 2 (em 13.196 ns):** A = 0 , B = 0 , C = 0 , D = 0 , E = 0 , F = 0 , G = 1 , S = 1;
- **Cursor 3 (em 297.063 ns):** A = 0 , B = 1 , C = 1 , D = 1 , E = 0 , F = 1 , G = 0 , S = 1;
- **Cursor 4 (em 307.649 ns):** A = 0 , B = 1 , C = 1 , D = 1 , E = 1 , F = 0 , G = 0 , S = 0;
- **Cursor 5 (em 436.405 ns):** A = 1 , B = 0 , C = 1 , D = 0 , E = 1 , F = 1 , G = 0 , S = 1;
- **Cursor 6 (em 610.497 ns):** A = 1 , B = 1 , C = 1 , D = 1 , E = 0 , F = 0 , G = 1 , S = 1;
- **Cursor 7 (em 637 ns):** A = 1 , B = 1 , C = 1 , D = 1 , E = 1 , F = 1 , G = 0 , S = 0;
- **Cursor 8 (em 675.878 ns):** A = 1 , B = 1 , C = 1 , D = 1 , E = 1 , F = 1 , G = 1 , S = 1;

## 6. Análise

Neste experimento, foram analisadas as seguintes estruturas: dois multiplexadores de 4 entradas (questão 1), um decodificador de 4 para 16 e um multiplexador de 8 entradas (questão 2), além das portas ou e inversora. Com base nas Tabelas Verdade 1 e 2 e na simulação de onda apresentada nas Figuras 10 e 11, é possível compreender o funcionamento esperado de ambos os circuitos. Dessa forma, pode-se afirmar que os códigos desenvolvidos correspondem ao



comportamento esperado, uma vez que os valores obtidos coincidem com aqueles descritos nas tabelas verdade

## **7. Conclusão**

No experimento, foi possível descrever o comportamento das estruturas propostas e entender suas características. As simulações geraram os dados esperados, que foram comparados com as tabelas verdade dos circuitos (Tabelas 1 e 2). Não houveram erros ou divergências observados durante a realização do experimento.