



Universidade de Brasília
Faculdade de Tecnologia
Laboratório de Sistemas Digitais

Relatório 04

Carla de Araujo Clementino Ribeiro Mat:180030736

Professor:
Guilherme de Sousa Torres

1 Objetivos

Utilizar os multiplexadores 4 para 1, 8 para 1 e o decodificador de 4 para 16 bits desenvolvidos nos experimentos anteriores para implementar as funções Booleanas apresentadas abaixo.

2 Questões Propostas

1. Escrever em VHDL e simular no ModelSim uma entidade com 3 bits de entrada (A, B e C) e 2 bits de saída (X e Y), que implemente as funções Booleanas abaixo. Para isso, sua arquitetura deve usar somente dois multiplexadores de 4 entradas (entidade desenvolvida na Questão 02 do Experimento 2, utilizada aqui como “component”) e uma porta inversora.

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

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

2. Escrever em VHDL e implementar em FPGA uma entidade com 7 bits de entrada (A, B, C, D, E, F e G) e 1 bit de saída (S), que implemente a função Booleana abaixo. Para isso, sua arquitetura deve usar somente um decodificador de 4 para 16 (entidade desenvolvida no visto 2 do experimento 3, utilizada aqui como “component”), um multiplexador de 8 entradas (entidade desenvolvida no visto 1 do experimento 3, também utilizada aqui como “component”) e três portas OU. Associe cada um dos bits de entrada a diferentes chaves (SW0 a SW7) e a saída a um dos LEDs (LD0 a LD7). Dica: use as variáveis E, F e G como entradas de seleção do multiplexador.

$$\begin{aligned} S = & F \cdot G + \\ & \overline{A} \cdot B \cdot C \cdot D \cdot \overline{E} \cdot \overline{F} \cdot G + \\ & \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot \overline{E} \cdot \overline{F} \cdot G + \\ & A \cdot \overline{B} \cdot C \cdot \overline{E} \cdot F \cdot \overline{G} + \\ & \overline{A} \cdot B \cdot C \cdot D \cdot \overline{E} \cdot F \cdot \overline{G} + \\ & A \cdot B \cdot C \cdot D \cdot E \cdot \overline{F} \cdot \overline{G} + \\ & A \cdot \overline{B} \cdot \overline{C} \cdot D \cdot E \cdot \overline{F} \cdot \overline{G} \end{aligned} \quad (3)$$

3 Desenvolvimento Teórico

Para resolver as questões apresentadas foi utilizado o Teorema de Expansão de Shannon, a qual afirma que qualquer expressão Booleana pode ser expressa por:

$$f(w_1, \dots, w_n) = \overline{w_1} \cdot f(0, \dots, w_n) + w_1 \cdot f(1, \dots, w_n) \quad (4)$$

Dessa forma, é possível utilizar multiplexadores para implementá-las. A equação acima representa a Expansão de Shannon que nos permite implementar uma função Booleana utilizando um multiplexador de 2 para 1. É possível afirmar isso ao comparar a expressão (7) com a expressão que descreve o multiplexador:

$$Z = S \cdot I_1 + \overline{S} \cdot I_0 \quad (5)$$

Sendo assim, ao comparar as equações (7) e (5) temos que a entrada de seleção seria controlada pelo bit w_1 e, $I_1 = f(1, \dots, w_n)$ e $I_0 = f(0, \dots, w_0)$ representam as duas entradas possíveis. Isso, pode ser melhor ilustrado pelo desenho abaixo:

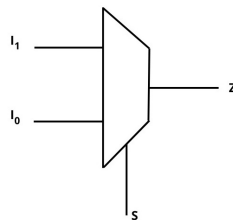


Figura 1: Multiplexador 2 para 1. Fonte: Autor.

Também é possível realizá-la para outros termos:

$$f(w_1, \dots, w_k, \dots, w_2) = \overline{w_k} \cdot f(w_1, \dots, 0, w_n) + w_k \cdot f(w_1, \dots, 1, w_n) \quad (6)$$

E para mais de um termo:

$$f(w_1, w_2, \dots, w_2n) = \overline{w_1} \cdot \overline{w_2} \cdot f(0, 0, \dots, w_n) + \overline{w_1} \cdot w_2 \cdot f(0, 1, \dots, w_n) + w_1 \cdot \overline{w_2} \cdot f(1, 0, \dots, w_n) + w_1 \cdot w_2 \cdot f(1, 1, \dots, w_n) \quad (7)$$

3.1 Primeira Questão

Para a primeira questão, podemos reescrever a equação (1) por meio da Expansão de Shannon, de forma a adaptá-la para um multiplexador de 4 para 1 da seguinte forma:

$$\begin{aligned} f(A, B, C) &= \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot \overline{C} + A \cdot B \\ &= \overline{A} \cdot \overline{B} \cdot f(0, 0, C) + \overline{A} \cdot B \cdot f(0, 1, C) + A \cdot \overline{B} \cdot f(1, 0, C) + A \cdot B \cdot f(1, 1, C) \end{aligned} \quad (8)$$

Logo, resolvendo cada uma das funções separadamente temos:

$$f(0, 0, C) = I_0 = \overline{0} \cdot 0 \cdot C + 0 \cdot \overline{0} \cdot \overline{C} + 0 \cdot 0 = 0 \quad (9)$$

$$f(0, 1, C) = I_1 = \overline{0} \cdot 1 \cdot C + 0 \cdot \overline{1} \cdot \overline{C} + 0 \cdot 1 = C \quad (10)$$

$$f(1, 0, C) = I_2 = \overline{1} \cdot 0 \cdot C + 1 \cdot \overline{0} \cdot \overline{C} + 1 \cdot 0 = \overline{C} \quad (11)$$

$$f(1, 1, C) = I_3 = \overline{1} \cdot 1 \cdot C + 1 \cdot \overline{1} \cdot \overline{C} + 1 \cdot 1 = 1 \quad (12)$$

E as variáveis de seleção são dadas por:

$$S_1 = A \text{ e } S_0 = B \quad (13)$$

E finalmente, obtemos o seguinte multiplexador:

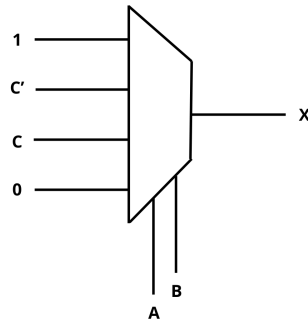


Figura 2: Multiplexador da Equação (1). Fonte: Autor.

Em seguida, faremos a mesma coisa para a equação (2):

$$\begin{aligned} f(A, B, C) &= \overline{A} \cdot \overline{B} + \overline{A} \cdot B \cdot \overline{C} + A \cdot B \cdot C \\ &= \overline{A} \cdot \overline{B} \cdot f(0, 0, C) + \overline{A} \cdot B \cdot f(0, 1, C) + A \cdot \overline{B} \cdot f(1, 0, C) + A \cdot B \cdot f(1, 1, C) \end{aligned} \quad (14)$$

Resolvendo cada função separadamente:

$$f(0, 0, C) = I_0 = \overline{0} \cdot 0 \cdot C + 0 \cdot \overline{0} \cdot \overline{C} + 0 \cdot 0 = 1 \quad (15)$$

$$f(0, 1, C) = I_1 = \overline{0} \cdot 1 \cdot C + 0 \cdot \overline{1} \cdot \overline{C} + 0 \cdot 1 = \overline{C} \quad (16)$$

$$f(1, 0, C) = I_2 = \overline{1} \cdot 0 \cdot C + 1 \cdot \overline{0} \cdot \overline{C} + 1 \cdot 0 = 0 \quad (17)$$

$$f(1, 1, C) = I_3 = \overline{1} \cdot 1 \cdot C + 1 \cdot \overline{1} \cdot \overline{C} + 1 \cdot 1 = \overline{C} \quad (18)$$

E as variáveis de seleção são dadas por:

$$S_1 = A \text{ e } S_0 = B \quad (19)$$

E representando a equação (2) na forma de um multiplexador:

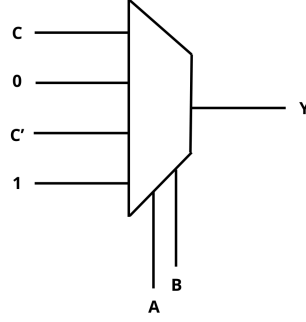


Figura 3: Multiplexador da Equação (2). Fonte: Autor

3.2 Segunda Questão

Para a segunda questão, também usaremos a Expansão de Shannon para expressar a equação (21) por meio de um multiplexador, porém, juntamente com um decodificador de 4 para 16 para auxiliar nas possíveis entradas. Realizando a expansão para as entradas E, F e G obtemos:

$$\begin{aligned} f(A, B, C, D, E, F, G) = & F \cdot G + \\ & \overline{A} \cdot B \cdot C \cdot D \cdot \overline{E} \cdot \overline{F} \cdot G + \\ & \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot \overline{E} \cdot \overline{F} \cdot G + \\ & A \cdot \overline{B} \cdot C \cdot \overline{E} \cdot F \cdot \overline{G} + \\ & \overline{A} \cdot B \cdot C \cdot D \cdot \overline{E} \cdot F \cdot \overline{G} + \\ & A \cdot B \cdot C \cdot D \cdot E \cdot \overline{F} \cdot \overline{G} + \\ & A \cdot \overline{B} \cdot \overline{C} \cdot D \cdot E \cdot \overline{F} \cdot \overline{G} \end{aligned} \quad (20)$$

$$\begin{aligned} f(A, B, C, D, E, F, G) = & \overline{E} \cdot \overline{F} \cdot \overline{G} \cdot f(A, B, C, D, 0, 0, 0) + \\ & \overline{E} \cdot \overline{F} \cdot G \cdot f(A, B, C, D, 0, 0, 1) + \\ & \overline{E} \cdot F \cdot \overline{G} \cdot f(A, B, C, D, 0, 1, 0) + \\ & \overline{E} \cdot F \cdot G \cdot f(A, B, C, D, 0, 1, 1) + \\ & E \cdot \overline{F} \cdot \overline{G} \cdot f(A, B, C, D, 1, 0, 0) + \\ & E \cdot \overline{F} \cdot G \cdot f(A, B, C, D, 1, 0, 1) + \\ & E \cdot F \cdot \overline{G} \cdot f(A, B, C, D, 1, 1, 0) + \\ & E \cdot F \cdot G \cdot f(A, B, C, D, 1, 1, 1) \end{aligned} \quad (21)$$

Resolvendo cada função separadamente:

$$f(A, B, C, D, 0, 0, 0) = I_0 = 0 \quad (22)$$

$$f(A, B, C, D, 0, 0, 1) = I_1 = A \cdot B \cdot C \cdot D + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \quad (23)$$

$$f(A, B, C, D, 0, 1, 0) = I_2 = \overline{A} \cdot B \cdot C \cdot D \quad (24)$$

$$f(A, B, C, D, 0, 1, 1) = I_3 = 1 \quad (25)$$

$$f(A, B, C, D, 1, 0, 0) = I_4 = A \cdot B \cdot C \cdot D + A \cdot \overline{B} \cdot \overline{C} \cdot D \quad (26)$$

$$f(A, B, C, D, 1, 0, 1) = I_5 = 0 \quad (27)$$

$$f(A, B, C, D, 1, 1, 0) = I_6 = A \cdot \overline{B} \cdot C \cdot X \quad (28)$$

$$f(A, B, C, D, 1, 1, 1) = I_7 = 1 \quad (29)$$

* X representa um don't care nesse caso.

E as variáveis de seleção são dadas por:

$$S_2 = E \text{ e } S_1 = F \text{ e } S_0 = G \quad (30)$$

Representando no formato de multiplexador:

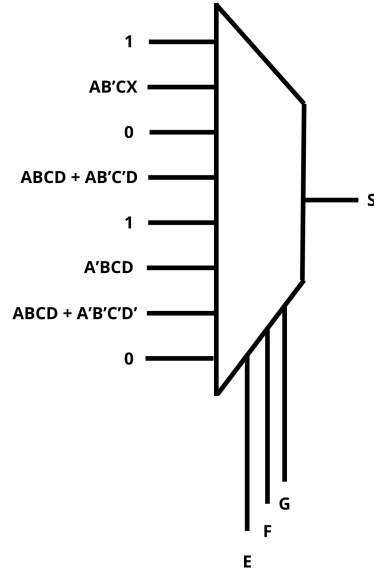


Figura 4: Multiplexador da Equação (21). Fonte: Autor.

Em seguida, analisando a tabela verdade de um decodificador 4 para 16 e utilizando suas possíveis saídas para controlar as entradas do multiplexador acima, podemos reescrevê-lo da seguinte forma:

A	B	C	D	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_{12}	O_{13}	O_{14}	O_{15}
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Tabela 1: Tabela Verdade do Decodificador 4 para 16. Fonte: Autor.

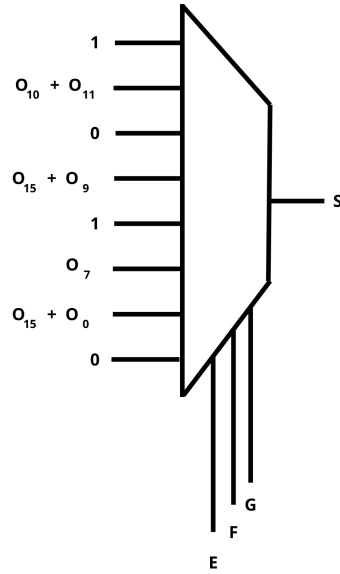


Figura 5: Multiplexador da Equação (21) com Decodificador. Fonte: Autor.

4 Simulação

4.1 Primeira Questão

Considerando a questão proposta, foi utilizado o multiplexador 4 para 1 implementado no experimento anterior, o qual pode ser visto abaixo:

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity mux is
5  Port (
6      S : in STD_LOGIC_VECTOR (1 downto 0); -- vetor de 2 bits (entrada)
7      D : in STD_LOGIC_VECTOR (3 downto 0); -- vetor de 4 bits (entrada)
8      Y : out STD_LOGIC -- (saida)
9  );
10 end mux;
11
12 architecture Behavioral of mux is
13
14 begin
15
16 -- Y = D0.S1'.S2' + D1.S1'.S0 + D2.S1.S0' + D3.S1.S2
17 Y <= (D(0) and not S(1) and not S(0)) or (D(1) and not S(1) and S(0)) or (D(2) and S(1) and not S(0)) or (D(3) and S(1) and S(0));
18
19 end Behavioral;
20

```

Figura 6: Multiplexador 4 para 1. Fonte: Autor.

Em seguida, criamos uma entidade para que fosse possível realizar simulações do nosso circuito por meio de sinais controlados. É possível ver a utilização dos dois multiplexadores:

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity master is
5  Port (
6      SW: in STD_LOGIC_VECTOR (2 downto 0);
7      LD1: out STD_LOGIC;
8      LD2: out STD_LOGIC
9  );
10 end master;
11
12 architecture Behavioral of master is
13
14     component mux
15     Port (
16         S : in STD_LOGIC_VECTOR (1 downto 0); -- vetor de 2 bits (entrada)
17         D : in STD_LOGIC_VECTOR (3 downto 0); -- vetor de 4 bits (entrada)
18         Y : out STD_LOGIC -- (saida)
19     );
20 end component;
21
22 signal u0: STD_LOGIC_VECTOR(3 downto 0); --entradas_x
23 signal u1: STD_LOGIC_VECTOR(3 downto 0); --entradas_y
24 signal u2: STD_LOGIC_VECTOR(1 downto 0); --chaves
25
26 begin
27
28     u2 <= SW(2 downto 1);
29     u0 <= "1" & not SW(0) & SW(0) & "0";
30     u1 <= SW(0) & "0" & not SW(0) & "1";
31
32     mux_x: mux port map(S => u2, D => u0, Y => LD1);
33     mux_y: mux port map(S => u2, D => u1, Y => LD2);
34
35 end Behavioral;

```

Figura 7: Entidade - Primeira Questão. Fonte: Autor.

Cada um dos bits de entrada são associados a uma chave (sw0 a sw7) e os bits de saída a diferentes LEDs, como especificado na questão.

Por fim, foi criado um arquivo de teste para simular todas as possíveis entradas do circuito:

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity test_bench is
5  end test_bench;
6
7  architecture Behavioral of test_bench is
8      signal A : STD_LOGIC_VECTOR(2 downto 0) := "000";
9      signal LED_X : STD_LOGIC;
10     signal LED_Y : STD_LOGIC;
11
12     component master is
13     Port (
14         SW: in STD_LOGIC_VECTOR (2 downto 0);
15         LD1: out STD_LOGIC;
16         LD2: out STD_LOGIC
17     );
18     end component;
19
20 begin
21
22     UUT: master port map (SW => A, LD1 => LED_X, LD2 => LED_Y);
23
24     A(0) <= not A(0) after 2ns;
25     A(1) <= not A(1) after 4ns;
26     A(2) <= not A(2) after 8ns;
27
28 end Behavioral;

```

Figura 8: Casos de Teste - Primeira Questão. Fonte: Autor.

Realizando a simulação e comparando o resultado com as tabelas verdade das equações (1) e (2) apresentadas abaixo, obtemos os seguintes resultados:

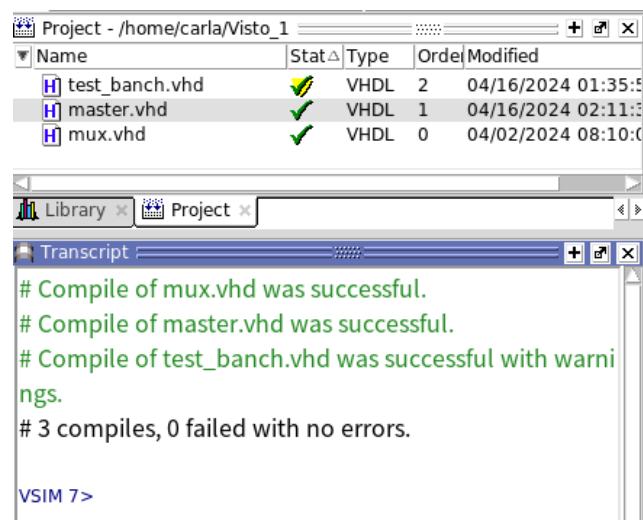


Figura 9: Compilação dos arquivos - Primeira Questão. Fonte: Autor.

Msgs									
+ /test_bench/UUT/SW /test_bench/UUT/LD1 /test_bench/UUT/LD2	-No Data-	000	001	010	011	100	101	110	111
	-No Data-								
	-No Data-								

Figura 10: Simulação da Primeira Questão. Fonte: Autor.

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

Tabela 2: Tabela Verdade de X. Fonte: Autor.

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

Tabela 3: Tabela Verdade de Y. Fonte: Autor.

4.2 Segunda Questão

Considerando a questão proposta, foi utilizado o multiplexador 8 para 1 e o decodificador 4 para 16 implementados nos experimentos anteriores, os quais podem ser vistos abaixo:

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity mux_8 is
5  Port (
6      S : in STD_LOGIC_VECTOR (2 downto 0);
7      D : in STD_LOGIC_VECTOR (7 downto 0);
8      Y : out STD_LOGIC
9  );
10 end mux_8;
11
12 architecture Behavioral of mux_8 is
13
14     signal entrada: STD_LOGIC_VECTOR(2 downto 0);
15
16     begin
17
18         entrada <= S(2) & S(1) & S(0);
19
20         Y <= D(0) when (entrada = "000") else
21             D(1) when (entrada = "001") else
22             D(2) when (entrada = "010") else
23             D(3) when (entrada = "011") else
24             D(4) when (entrada = "100") else
25             D(5) when (entrada = "101") else
26             D(6) when (entrada = "110") else
27             D(7);
28
29     end Behavioral;
30

```

Figura 11: Multiplexador 8 para 1. Fonte: Autor.

```

1
2 library IEEE;
3 use IEEE.STD_LOGIC_1164.ALL;
4
5 entity dec_4_to_16 is
6   Port (
7     A : in STD_LOGIC_VECTOR (3 downto 0);
8     Y : out STD_LOGIC_VECTOR (15 downto 0)
9   );
10 end dec_4_to_16;
11
12 architecture Behavioral of dec_4_to_16 is
13 begin
14   with A select
15     Y <= "0000000000000001" when "0000",
16         "0000000000000010" when "0001",
17         "0000000000000100" when "0010",
18         "0000000000001000" when "0011",
19         "0000000000010000" when "0100",
20         "0000000001000000" when "0101",
21         "0000000010000000" when "0110",
22         "0000000100000000" when "0111",
23         "0000001000000000" when "1000",
24         "0000010000000000" when "1001",
25         "0000100000000000" when "1010",
26         "0001000000000000" when "1011",
27         "0010000000000000" when "1100",
28         "0100000000000000" when "1101",
29         "1000000000000000" when "1110",
30         "1000000000000000" when others;
31 end Behavioral;
32

```

Figura 12: Decodificador 4 para 16. Fonte: Autor.

Em seguida, criamos uma entidade para que fosse possível realizar simulações do nosso circuito por meio de sinais controlados. É possível ver a utilização do multiplexador e do decodificador:

```

1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity master is
5   Port (
6     SW: in STD_LOGIC_VECTOR (6 downto 0);
7     LD1: out STD_LOGIC
8   );
9 end master;
10
11 architecture Behavioral of master is
12
13   component mux_8
14     Port (
15       S : in STD_LOGIC_VECTOR (2 downto 0);
16       D : in STD_LOGIC_VECTOR (7 downto 0);
17       Y : out STD_LOGIC
18     );
19   end component;
20
21   component dec_4_to_16
22     Port (
23       A : in STD_LOGIC_VECTOR (3 downto 0);
24       Y : out STD_LOGIC_VECTOR (15 downto 0)
25     );
26   end component;
27
28   signal u0: STD_LOGIC_VECTOR(15 downto 0); --saida do dec
29   signal u1: STD_LOGIC_VECTOR(3 downto 0); --entradas do dec
30   signal u2: STD_LOGIC_VECTOR(2 downto 0); --chaves
31   signal u3: STD_LOGIC_VECTOR(7 downto 0); --entradas do mux
32
33 begin
34
35   u1 <= SW(6 downto 3);
36
37   decoder: dec_4_to_16 port map(A => u1, Y => u0);
38
39   u2 <= SW(2 downto 0);
40
41   u3(0) <= '0';
42   u3(1) <= u0(0) or u0(15);
43   u3(2) <= u0(7);
44   u3(3) <= '1';
45   u3(4) <= u0(15) or u0(9);
46   u3(5) <= '0';
47   u3(6) <= u0(10) or u0(11); --don't care
48   u3(7) <= '1';
49
50   mux: mux_8 port map(S => u2, D => u3, Y => LD1);
51
52 end Behavioral;

```

Figura 13: Entidade - Segunda Questão. Fonte: Autor.

Cada um dos bits de entrada são associados a uma chave (sw0 a sw7) e a saída a um LED, como especificado na questão.

Por fim, foi criado um arquivo de teste para simular todas as possíveis entradas do circuito:

```
1
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4
5  entity test_bench is
6  end test_bench;
7
8  architecture Behavioral of test_bench is
9      signal U0 : STD_LOGIC_VECTOR(6 downto 0) := "0000000";
10     signal S : STD_LOGIC;
11
12     component master is
13     Port (
14         SW: in STD_LOGIC_VECTOR (6 downto 0);
15         LD1: out STD_LOGIC
16     );
17     end component;
18
19     begin
20
21     UUT: master port map (
22         SW => U0,
23         LD1 => S
24     );
25
26     U0(0) <= not U0(0) after 2ns;
27     U0(1) <= not U0(1) after 4ns;
28     U0(2) <= not U0(2) after 8ns;
29     U0(3) <= not U0(3) after 16ns;
30     U0(4) <= not U0(4) after 32ns;
31     U0(5) <= not U0(5) after 64ns;
32     U0(6) <= not U0(6) after 128ns;
33
34     end Behavioral;
```

Figura 14: Casos de Teste - Segunda Questão. Fonte: Autor.

Realizando a simulação, obtemos os seguintes resultados:

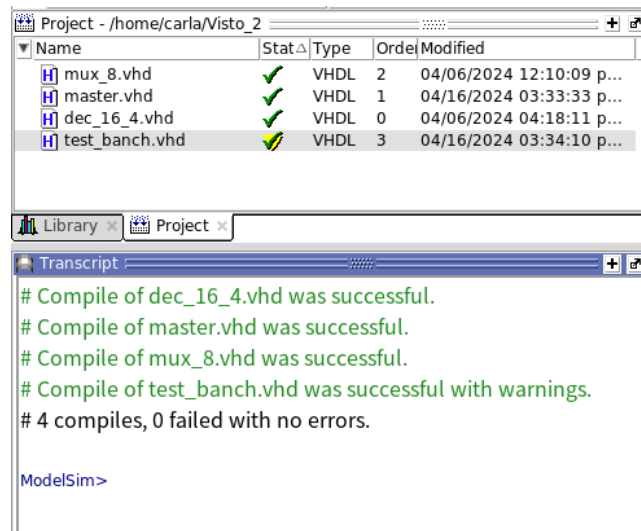


Figura 15: Compilação dos arquivos - Primeira Questão. Fonte: Autor.

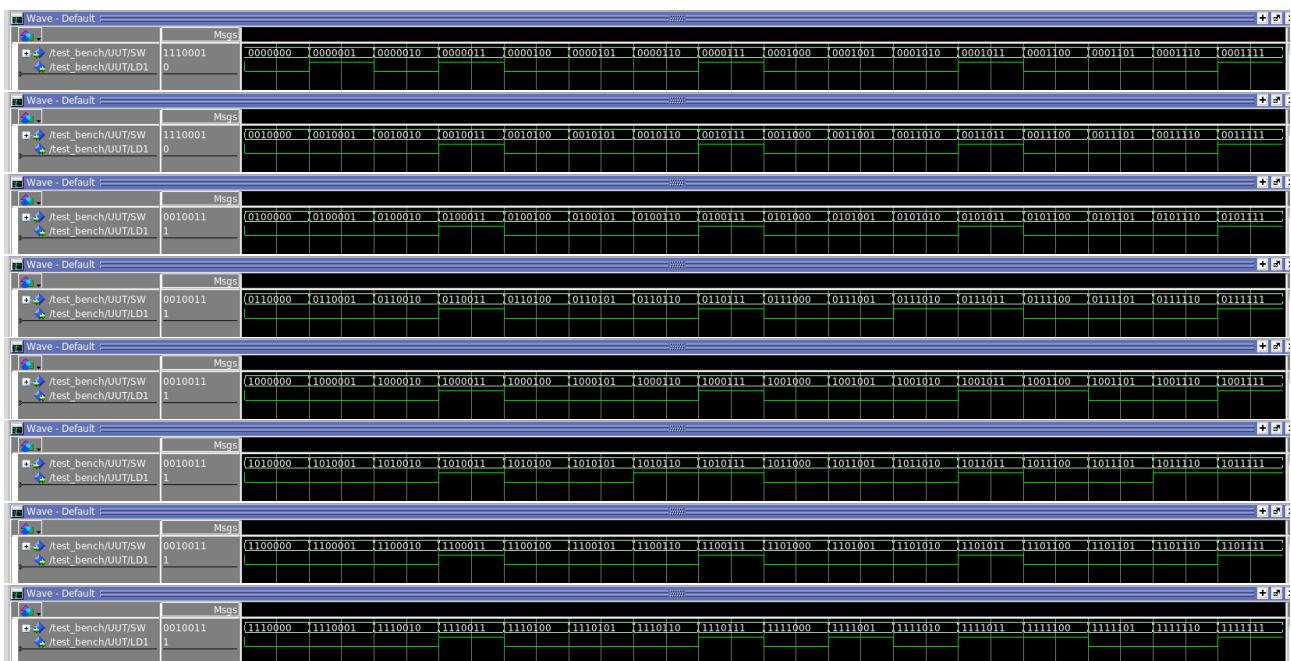


Figura 16: Simulação da Segunda Questão. Fonte: Autor.

5 Referências

[1] Pedro Souza. (2020, julho 4). [CIRCUITOS DIGITAIS] Aula 47 - Teorema da Expansão de Shannon [Vídeo]. Disponível em: <https://www.youtube.com/watch?v=Kjtvfyf6e-9U>. Acesso em: 14 de abril de 2024.