

Nome: Gustavo Pereira Chaves
Matrícula: 19/0014113
Turma: 03A
Data: 16/04/2021

Relatório do Experimento 6

Questão 1:

- Documentação:

MqEstRefri.vhd

```
1  -- Universidade de Brasília
2  -- Laboratório de Sistemas Digitais
3  -- Autor : Gustavo Pereira Chaves
4  -- Matrícula: 19/0014113
5  -- Data : 16/04/2021
6
7  -- Implementação de uma máquina de refrigerantes utilizando uma máquina de estados de Moore
8
9  -- *****
10 -- Circuito: Máquina de Refrigerante de Moore:
11 --          clk  Entrada 1
12 --          reset Entrada 2
13 --          A    Entrada 3
14 --          Q    Saida 1 - Moeda de 25 centavos
15 --          H    Saida 2 - Moeda de 50 centavos
16 --          R    Saida 3 - Refrigerante
17 -- *****
18
19 --Package (Pacote)
20 --constantes e bibliotecas
21
22 library IEEE;
23 use IEEE.std_logic_1164.all;
24 use ieee.numeric_std.all;
25
26 --Entity (Entidade)
27 --pinos de entrada e saída
28
29 entity MqEstRefri is
30     port (
31         clk, reset :in std_logic;
32         A : in std_logic_vector(1 downto 0);
33         Q, H, R: out std_logic
34     );
35 end MqEstRefri;
36
37 --Architecture (Arquitetura)
38 --implementacoes do projeto
39
40 architecture MqEstRefri_arch of MqEstRefri is
41
42     -- Definição de tipo state que indica os estados possiveis
43     type state is (Init, One, Two, Three, Four, Five, Cancel_One, Cancel_Two, Cancel_Three);
44
45     -- Sinais auxiliares para armazenar o estado atual e o próxima estado respectivamente
46     signal current_state :state := Init;
47     signal next_state :state;
48
```

```

49  -- a definicao inicia por
50  begin
51
52  -- Processo síncrono da maquina de estados
53  sync_process: process (clk, reset)
54  begin
55      if (reset = '1') then
56          current_state <= Init;
57      elsif rising_edge(clk) then
58          current_state <= next_state;
59      end if;
60  end process;
61
62  -- Processo combinacional da maquina de estados
63  comb_process: process (current_state, A)
64  begin
65      Q <= '0'; H <= '0'; R <= '0';
66      case current_state is
67          when Init =>
68              case A is
69                  when "00" => next_state <= Init;
70                  when "01" => next_state <= One;
71                  when "10" => next_state <= Two;
72                  when others => next_state <= Init;
73              end case;
74          when One =>
75              case A is
76                  when "00" => next_state <= One;
77                  when "01" => next_state <= Two;
78                  when "10" => next_state <= Three;
79                  when others => next_state <= Cancel_One;
80              end case;
81          when Two =>
82              case A is
83                  when "00" => next_state <= Two;
84                  when "01" => next_state <= Three;
85                  when "10" => next_state <= Four;
86                  when others => next_state <= Cancel_Two;
87              end case;
88          when Three =>
89              case A is
90                  when "00" => next_state <= Three;
91                  when "01" => next_state <= Four;
92                  when "10" => next_state <= Five;
93                  when others => next_state <= Cancel_Three;
94              end case;
95          when Four =>
96              case A is
97                  when "00" => next_state <= Init;
98                  when "01" => next_state <= One;
99                  when "10" => next_state <= Two;
100             when others => next_state <= Init;
101             end case;
102             R <= '1';

```

```

103         when Five =>
104             case A is
105                 when "00" => next_state <= Init;
106                 when "01" => next_state <= One;
107                 when "10" => next_state <= Two;
108                 when others => next_state <= Init;
109             end case;
110             Q <= '1'; R <= '1';
111         when Cancel_One =>
112             case A is
113                 when "00" => next_state <= Init;
114                 when "01" => next_state <= One;
115                 when "10" => next_state <= Two;
116                 when others => next_state <= Init;
117             end case;
118             Q <= '1';
119         when Cancel_Two =>
120             case A is
121                 when "00" => next_state <= Init;
122                 when "01" => next_state <= One;
123                 when "10" => next_state <= Two;
124                 when others => next_state <= Init;
125             end case;
126             H <= '1';
127         when Cancel_Three =>
128             case A is
129                 when "00" => next_state <= Init;
130                 when "01" => next_state <= One;
131                 when "10" => next_state <= Two;
132                 when others => next_state <= Init;
133             end case;
134             Q <= '1'; H <= '1';
135         end case;
136     end process;
137
138 end MqEstRefri_arch;
139 -- a definicao termina por end
140

```

tb_MqEstRefri.vhd

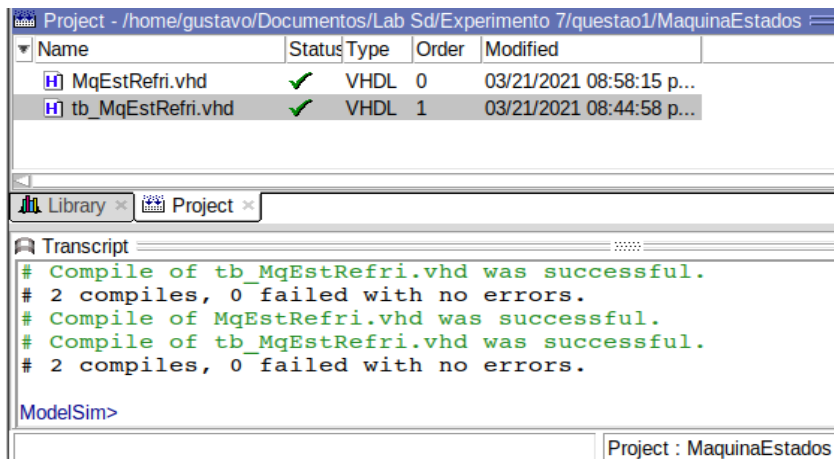
```
1  -- Universidade de Brasília
2  -- Laboratório de Sistemas Digitais
3  -- Autor : Gustavo Pereira Chaves
4  -- Matrícula: 19/0014113
5  -- Data : 16/04/2021
6
7  -- Implementação de uma máquina de refrigerantes utilizando uma máquina de estados de Moore
8
9  -- *****
10 -- Testbench para simulação funcional do
11 -- Circuito: Máquina de Refrigerante de Moore:
12 --           clk  Entrada 1
13 --           reset Entrada 2
14 --           Q     Saida 1 - Moeda de 25 centavos
15 --           H     Saida 2 - Moeda de 50 centavos
16 --           R     Saida 3 - Refrigerante
17 --           R     Saida 3
18 -- *****
19
20 -- entity (Entidade)
21 -- tb_MqEstRefri uma entidade sem pinos de entrada e saida
22
23 entity tb_MqEstRefri is end;
24
25 --Package (Pacote)
26 --constantes e bibliotecas
27
28 library ieee;
29 use ieee.std_logic_1164.ALL;
30 use std.textio.all;
31 use ieee.numeric_std.all;
32
33 -- Architecture (Arquitetura)
34
35 architecture tb_MqEstRefri_arch of tb_MqEstRefri is
36
37 -- Declaracao do componente MqEstRefri, referente a sua arquitetura descrita no arquivo MqEstRefri.vhd
38
39 component MqEstRefri is
40     port (
41         clk, reset :in std_logic;
42         A : in std_logic_vector(1 downto 0);
43         Q, H, R: out std_logic
44     );
45 end component;
46
47 -- Sinais auxiliares para a interconexao ao
48 -- processo de estimulo
49
50 signal clk :std_logic := '0';
51 signal reset:std_logic;
52 signal A :std_logic_vector(1 downto 0);
53
54 -- Instancia do componente MqEstRefri e interconexao do componente ao processo de estimulo
55
56 begin
57     u0: MqEstRefri port map (clk, reset, A, open, open, open);
58
59 -- Implementação do clock
60 clk <= not clk after 5 ns;
61
62 -- Implementacao do processo de estimulo
63 estimulo: process
64     begin
65         -- Testar colocando apenas moedas de 25 centavos
66         for i in 0 to 3 loop
67             A <= "00"; -- Mantém Estado
68             wait for 10 ns;
69             A <= "01"; -- Muda Estado
70             wait for 10 ns;
71         end loop;
```

```

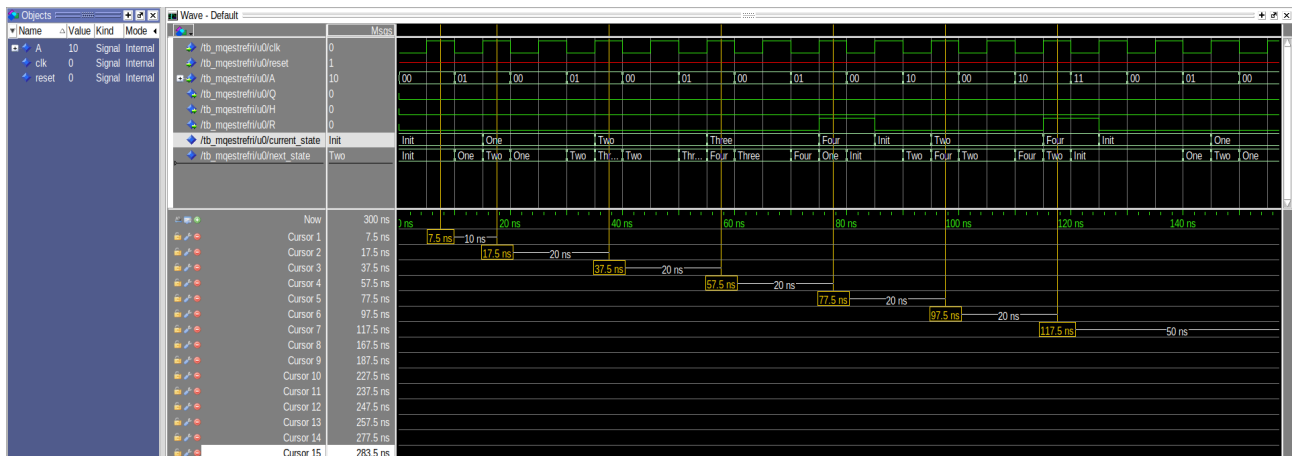
72
73 -- Testar colocando apenas moedas de 50 centavos
74 for i in 0 to 1 loop
75     A <= "00"; -- Mantém Estado
76     wait for 10 ns;
77     A <= "10"; -- Muda Estado
78     wait for 10 ns;
79 end loop;
80 A <= "11"; -- Cancelando
81 wait for 10 ns;
82
83 -- Testar colocando 1 moeda de 25 e 2 de 50
84 A <= "00"; -- Mantém Estado
85 wait for 10 ns;
86 A <= "01"; -- Muda Estado
87 wait for 10 ns;
88
89 for i in 0 to 1 loop
90     A <= "00"; -- Mantém Estado
91     wait for 10 ns;
92     A <= "10"; -- Muda Estado
93     wait for 10 ns;
94 end loop;
95 A <= "11"; -- Cancelando
96 wait for 10 ns;
97
98 -- Testando cancelar em cada um dos 4 primeiros estados
99 -- Initial
100 A <= "11";
101 wait for 10 ns;
102
103 -- One
104 A <= "01"; -- Mudando para Estado One
105 wait for 10 ns;
106 A <= "11"; -- Mudando para Estado Cancel_One
107 wait for 10 ns;
108
109 -- Two
110 A <= "10"; -- Mudando para Estado Two
111 wait for 10 ns;
112 A <= "11"; -- Mudando para Estado Cancel_Two
113 wait for 10 ns;
114
115 -- Three
116 A <= "01"; -- Mudando para Estado One
117 wait for 10 ns;
118 A <= "10"; -- Mudando para Estado Three
119 wait for 10 ns;
120 A <= "11"; -- Mudando para Estado Cancel_Three
121 wait for 10 ns;
122
123 -- Testando Reset Assíncrono
124 A <= "10";
125 wait for 2.5 ns;
126 reset <= '1';
127 wait for 5 ns;
128 reset <= '0';
129
130 wait;
131 -- Fim do processo de estímulo
132 end process;
133
134 --Fim da definição da arquitetura
135 END;

```

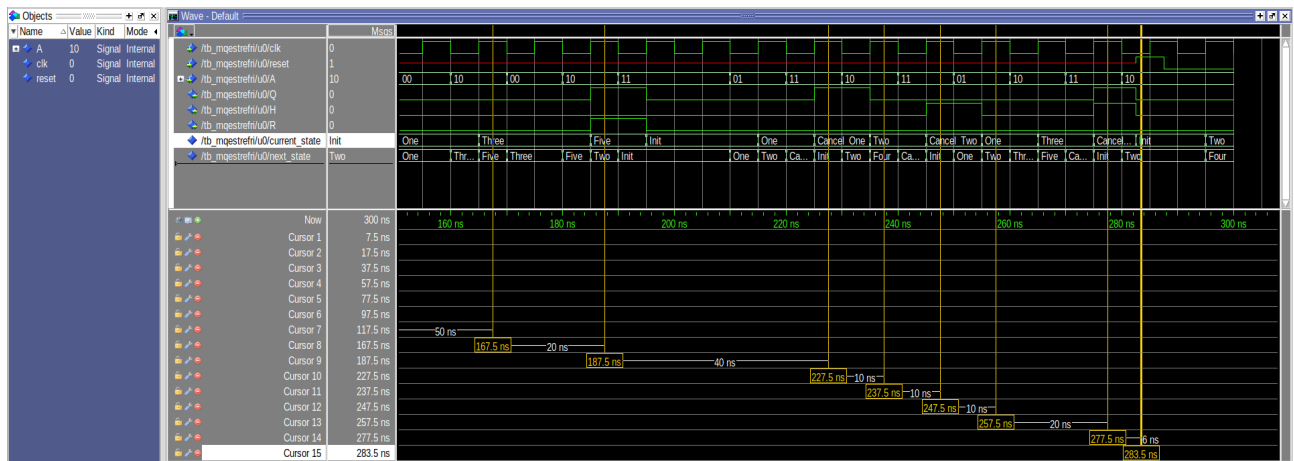
- **Compilação:**



- **Simulação do Código:**



Cursor 1: Em 7,5 ns, clk = 1, A = 00, Q = 0, H = 0, R = 0, current_state = Init
 Cursor 2: Em 17,5 ns, clk = 1, A = 01, Q = 0, H = 0, R = 0, current_state = One
 Cursor 3: Em 37,5 ns, clk = 1, A = 01, Q = 0, H = 0, R = 0, current_state = Two
 Cursor 4: Em 57,5 ns, clk = 1, A = 01, Q = 0, H = 0, R = 0, current_state = Three
 Cursor 5: Em 77,5 ns, clk = 1, A = 01, Q = 0, H = 0, R = 1, current_state = Four
 Cursor 6: Em 97,5 ns, clk = 1, A = 10, Q = 0, H = 0, R = 0, current_state = Two
 Cursor 7: Em 117,5 ns, clk = 1, A = 10, Q = 0, H = 0, R = 1, current_state = Four



Cursor 8: Em 167,5 ns, clk = 1, A = 10, Q = 0, H = 0, R = 0, current_state = Three
Cursor 9: Em 187,5 ns, clk = 1, A = 10, Q = 1, H = 0, R = 1, current_state = Five
Cursor 10: Em 227,5 ns, clk = 1, A = 11, Q = 1, H = 0, R = 0, current_state = Cancel_One
Cursor 11: Em 237,5 ns, clk = 1, A = 10, Q = 0, H = 0, R = 0, current_state = Two
Cursor 12: Em 247,5 ns, clk = 1, A = 11, Q = 0, H = 1, R = 0, current_state = Cancel_Two
Cursor 13: Em 257,5 ns, clk = 1, A = 01, Q = 0, H = 0, R = 0, current_state = One
Cursor 14: Em 117,5 ns, clk = 0, reset = 1, Q = 0, H = 0, R = 0, current_state = Init