

Nome: Gustavo Pereira Chaves
Matrícula: 19/0014113
Turma: 03A
Data: 23/04/2021

Relatório do Experimento 8

Questão 1:

- Documentação:

contador10.vhd

```
1  -- Universidade de Brasília
2  -- Laboratório de Sistemas Digitais
3  -- Autor : Gustavo Pereira Chaves
4  -- Matrícula: 19/0014113
5  -- Data : 23/04/2021
6
7  -- Implementação de um contador modulo 10, com reset e load sincronos
8
9  -- *****
10 -- Circuito: Contador modulo 100 com saída BCD
11 --          clk          Entrada 1
12 --          reset        Entrada 2
13 --          enable       Entrada 3 (ativo baixo)
14 --          rci          Entrada 3 (ativo baixo)
15 --          load         Entrada 4
16 --          D            Entrada 5
17 --          Q            Saída 1
18 --          rco          Saída 2 (ativo baixo)
19 -- *****
20
21 --Package (Pacote)
22 --constantes e bibliotecas
23
24 library IEEE;
25 use IEEE.std_logic_1164.all;
26 use ieee.numeric_std.all;
27
28 --Entity (Entidade)
29 --pinos de entrada e saída
30
31 entity contador10 is
32     port (
33         clk, reset, enable, rci, load :in std_logic;
34         D : in std_logic_vector(3 downto 0);
35         Q : out std_logic_vector(3 downto 0);
36         rco : out std_logic
37     );
38 end contador10;
39
40 --Architecture (Arquitetura)
41 --implementacoes do projeto
42
43 architecture contador10_arch of contador10 is
44
45     -- Definição de tipo state que indica os estados possiveis
46     type state is (ST0, ST1, ST2, ST3, ST4, ST5, ST6, ST7, ST8, ST9);
47
48     -- Sinais auxiliares para armazenar estado atual e mudanças de estado
49     signal current_state, next_state, load_state :state;
50
```

```

51 -- a definicao inicia por
52 begin
53
54     with D select
55         load_state <= ST0 when "0000",
56                     ST1 when "0001",
57                     ST2 when "0010",
58                     ST3 when "0011",
59                     ST4 when "0100",
60                     ST5 when "0101",
61                     ST6 when "0110",
62                     ST7 when "0111",
63                     ST8 when "1000",
64                     ST9 when "1001",
65                     ST0 when others;
66
67 -- Processo síncrono da maquina de estados
68 sync_process: process(clk)
69 begin
70     if rising_edge(clk) then
71         current_state <= next_state;
72     end if;
73 end process;
74
75 -- Processo combinacional da maquina de estados
76 comb_process: process (current_state, reset, enable, rci, load, load_state)
77 begin
78     case current_state is
79         when ST0 =>
80             Q <= "0000";
81             rco <= '1';
82             if (reset = '1') then next_state <= ST0;
83             elsif (load = '1') then next_state <= load_state;
84             elsif ((enable = '0') or (rci = '0')) then next_state <= ST1;
85             else next_state <= ST0;
86             end if;
87
88         when ST1 =>
89             Q <= "0001";
90             rco <= '1';
91             if (reset = '1') then next_state <= ST0;
92             elsif (load = '1') then next_state <= load_state;
93             elsif ((enable = '0') or (rci = '0')) then next_state <= ST2;
94             else next_state <= ST1;
95             end if;
96
97         when ST2 =>
98             Q <= "0010";
99             rco <= '1';
100            if (reset = '1') then next_state <= ST0;
101            elsif (load = '1') then next_state <= load_state;
102            elsif ((enable = '0') or (rci = '0')) then next_state <= ST3;
103            else next_state <= ST2;
104            end if;
105
106         when ST3 =>
107             Q <= "0011";
108             rco <= '1';
109             if (reset = '1') then next_state <= ST0;
110             elsif (load = '1') then next_state <= load_state;
111             elsif ((enable = '0') or (rci = '0')) then next_state <= ST4;
112             else next_state <= ST3;
113             end if;
114
115         when ST4 =>
116             Q <= "0100";
117             rco <= '1';
118             if (reset = '1') then next_state <= ST0;
119             elsif (load = '1') then next_state <= load_state;
120             elsif ((enable = '0') or (rci = '0')) then next_state <= ST5;
121             else next_state <= ST4;
122             end if;
123

```

```

123
124
125     when ST5 =>
126         Q <= "0101";
127         rco <= '1';
128         if (reset = '1') then next_state <= ST0;
129         elsif (load = '1') then next_state <= load_state;
130         elsif ((enable = '0') or (rci = '0')) then next_state <= ST6;
131         else next_state <= ST5;
132         end if;
133
134     when ST6 =>
135         Q <= "0110";
136         rco <= '1';
137         if (reset = '1') then next_state <= ST0;
138         elsif (load = '1') then next_state <= load_state;
139         elsif ((enable = '0') or (rci = '0')) then next_state <= ST7;
140         else next_state <= ST6;
141         end if;
142
143     when ST7 =>
144         Q <= "0111";
145         rco <= '1';
146         if (reset = '1') then next_state <= ST0;
147         elsif (load = '1') then next_state <= load_state;
148         elsif ((enable = '0') or (rci = '0')) then next_state <= ST8;
149         else next_state <= ST7;
150         end if;
151
152     when ST8 =>
153         Q <= "1000";
154         rco <= '1';
155         if (reset = '1') then next_state <= ST0;
156         elsif (load = '1') then next_state <= load_state;
157         elsif ((enable = '0') or (rci = '0')) then next_state <= ST9;
158         else next_state <= ST8;
159         end if;
160
161     when ST9 =>
162         Q <= "1001";
163         rco <= '0';
164         if (reset = '1') then next_state <= ST0;
165         elsif (load = '1') then next_state <= load_state;
166         elsif ((enable = '0') or (rci = '0')) then next_state <= ST0;
167         else next_state <= ST9;
168         end if;
169
170     when others =>
171         Q <= "0000";
172         rco <= '1';
173         if (reset = '1') then next_state <= ST0;
174         elsif (load = '1') then next_state <= load_state;
175         elsif ((enable = '0') or (rci = '0')) then next_state <= ST1;
176         else next_state <= ST0;
177         end if;
178
179     end case;
180 end process;
181
182 end contador10_arch;
-- a definicao termina por end

```

questao1.vhd

```
1  -- Universidade de Brasília
2  -- Laboratório de Sistemas Digitais
3  -- Autor : Gustavo Pereira Chaves
4  -- Matrícula: 19/0014113
5  -- Data : 23/04/2021
6
7  -- Implementação de um contador modulo 100 com saída BCD, com reset e load síncronos
8  -- utilizando 2 contadores de modulo 10
9
10 -- *****
11 -- Circuito: Contador modulo 100 com saída BCD
12 --          clk           Entrada 1
13 --          reset         Entrada 2
14 --          enable        Entrada 3 (ativo baixo)
15 --          load           Entrada 4
16 --          dezload        Entrada 5
17 --          uniload        Entrada 6
18 --          cont_dezena     Saída 1
19 --          cont_unidade    Saída 2
20 -- *****
21
22 --Package (Pacote)
23 --constantes e bibliotecas
24
25 library IEEE;
26 use IEEE.std_logic_1164.all;
27 use ieee.numeric_std.all;
28
29 --Entity (Entidade)
30 --pinos de entrada e saída
31
32 entity contador100 is
33     port (
34         clk, reset, enable, load: in std_logic;
35         dezload, uniload :in std_logic_vector(3 downto 0);
36         cont_dezena, cont_unidade: out std_logic_vector(3 downto 0)
37     );
38 end contador100;
39
40 --Architecture (Arquitetura)
41 --implementacoes do projeto
42
43 architecture contador100_arch of contador100 is
44     -- Declaração do componente contador10, referente a sua arquitetura descrita no arquivo contador10.vhd
45
46     component contador10 is
47         port (
48             clk, reset, enable, rci, load :in std_logic;
49             D : in std_logic_vector(3 downto 0);
50             Q : out std_logic_vector(3 downto 0);
51             rco : out std_logic
52         );
53     end component;
54
55     -- Sinal auxiliares para a interconexao ao
56     -- processo de estimulo
57
58     signal signal_rco: std_logic;
59
60     begin
61         -- Instancias do componente contador10 e interconexao dos componentes:
62         unidade: contador10 PORT MAP (clk, reset, enable, enable, load, uniload, cont_unidade, signal_rco);
63         dezena: contador10 PORT MAP (clk, reset, signal_rco, signal_rco, load, dezload, cont_dezena, open);
64
65     end contador100_arch;
66     -- a definicao termina por end
```

tb_questao1.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 um contador modulo 100 com saída BCD, com reset e load síncronos
8  -- utilizando 2 contadores de modulo 10
9
10 -- *****
11 -- Testbench para simulação funcional do
12 -- Circuito: Contador modulo 100 com saída BCD
13 --
14 --          clk          Entrada 1
15 --          reset        Entrada 2
16 --          enable        Entrada 3 (ativo baixo)
17 --          load          Entrada 4
18 --          dezload       Entrada 5
19 --          uniload       Entrada 6
20 --          cont_dezena    Saída 1
21 --          cont_unidade   Saída 2
22 -- *****
23
24 -- entity (Entidade)
25 -- tb_questao1 uma entidade sem pinos de entrada e saída
26
27 entity tb_questao1 is end;
28
29 --Package (Pacote)
30 --constantes e bibliotecas
31
32 library ieee;
33 use ieee.std_logic_1164.ALL;
34 use std.textio.all;
35 use ieee.numeric_std.all;
36
37 -- Architecture (Arquitetura)
38
39 architecture tb_questao1_arch of tb_questao1 is
40
41 -- Declaração do componente contador100, referente a sua arquitetura descrita no arquivo questao1.vhd
42
43 component contador100 is
44     port (
45         clk, reset, enable, load: in std_logic;
46         dezload, uniload: in std_logic_vector(3 downto 0);
47         cont_dezena, cont_unidade: out std_logic_vector(3 downto 0)
48     );
49 end component;
50
51 -- Sinais auxiliares para a interconexão ao
52 -- processo de estímulo
53
54 signal clk :std_logic := '0';
55 signal reset, enable, load :std_logic;
56 signal dezload, uniload :std_logic_vector(3 downto 0);
57
58 -- Instância do componente contador100 e interconexão do componente ao processo de estímulo
59
60 begin
61     u0: contador100 PORT MAP (clk, reset, enable, load, dezload, uniload, open, open);
62
63 -- Implementação do clock
64 clk <= not clk after 5 ms;
65
66 -- Implementação do processo de estímulo
67 estimulo: process
68 begin
69     load <= '1'; dezload <= "0010"; uniload <= "1000";
70     wait for 10 ms;
71
72     enable <= '0'; load <= '0';
73     wait for 130 ms;
74
75     reset <= '1';
76     wait for 10 ms;
77     reset <= '0';
78     wait for 20 ms;
79
80     enable <= '1';
81     wait for 10 ms;
82     enable <= '0';
83     wait;
84 -- Fim do processo de estímulo
85 end process;
86
87 --Fim da definição da arquitetura
88 END;
```

- **Compilação:**

Project - /home/gustavo/Documents/Lab Sd/Experimento 8/questao1/questao1

Name	Status	Type	Order	Modified
questao1.vhd	✓	VHDL	0	04/21/2021 05:16:29 p...
contador10.vhd	✓	VHDL	1	04/21/2021 07:05:11 p...
tb_questao1.vhd	✓	VHDL	2	04/21/2021 07:19:21 p...

Library × Project ×

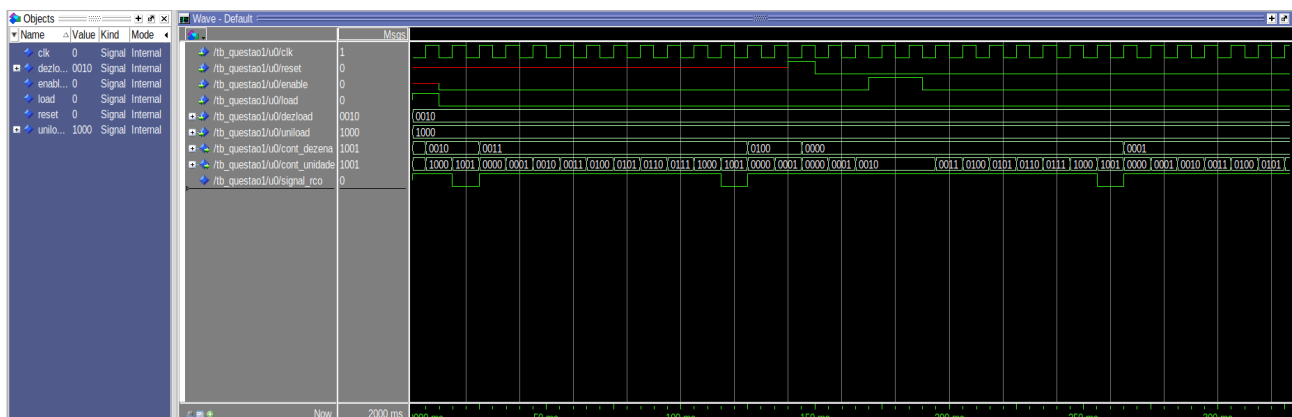
Transcript

```
# Errors: 0, Warnings: 1
# Compile of questao1.vhd was successful.
# Compile of contador10.vhd was successful.
# Compile of tb_questao1.vhd was successful.
# 3 compiles, 0 failed with no errors.
```

ModelSim>

Project : questao1 <No

- **Simulação do Código:**



Questão 2:

- Documentação:

questao2.vhd

```
1  -- Universidade de Brasília
2  -- Laboratório de Sistemas Digitais
3  -- Autor : Gustavo Pereira Chaves
4  -- Matrícula: 19/0014113
5  -- Data : 23/04/2021
6
7  -- Implementação do sistema de temporizacao do controle de semaforos indicando
8  -- que já se passaram 5 segundos (T5), 6 segundos (T6), 20 segundos (T20), 60 segundos (T60)
9
10 -- *****
11 -- Circuito: Temporizador do semaforo
12 --          clk          Entrada 1
13 --          reset        Entrada 2
14 --          T5           Saida 1
15 --          T6           Saida 2
16 --          T20          Saida 3
17 --          T60          Saida 4
18 --          cont_dezena   Saida 5
19 --          cont_unidade  Saida 6
20 -- *****
21
22 --Package (Pacote)
23 --constantes e bibliotecas
24
25 library IEEE;
26 use IEEE.std_logic_1164.all;
27 use ieee.numeric_std.all;
28
29 --Entity (Entidade)
30 --pinos de entrada e saída
31
32 entity questao2 is
33     port (
34         clk, reset : in std_logic;
35         cont_dezena, cont_unidade : out std_logic_vector(3 downto 0);
36         T5, T6, T20, T60 : out std_logic
37     );
38 end questao2;
39
40 --Architecture (Arquitetura)
41 --implementacoes do projeto
42
43 architecture questao2_arch of questao2 is
44     -- Declaracao do componente contador100, referente a sua arquitetura descrita no arquivo questao1.vhd
45
46     component contador100 is
47         port (
48             clk, reset, enable, load: in std_logic;
49             dezload, uniload :in std_logic_vector(3 downto 0);
50             cont_dezena, cont_unidade: out std_logic_vector(3 downto 0)
51         );
52     end component;
53
54     -- Declaracao do componente timeflags, referente a sua arquitetura descrita no arquivo timeflags.vhd
55
56     component timeflags is
57         port(
58             cont_dezena, cont_unidade :in std_logic_vector(3 downto 0);
59             T5, T6, T20, T60 :out std_logic
60         );
61     end component;
62
63     -- Sinais auxiliares para a interconexao ao
64     -- processo de estimulo
65
66     signal s_dezena, s_unidade: std_logic_vector(3 downto 0);
67
68
69 begin
70     -- Instancias do componente contador100 e timeflags e interconexao dos componentes:
71     u0: contador100 PORT MAP (clk, reset, '0', '0', "0000", "0000", s_dezena, s_unidade);
72     u1: timeflags PORT MAP (s_dezena, s_unidade, T5, T6, T20, T60);
73
74     cont_dezena <= s_dezena;
75     cont_unidade <= s_unidade;
76
77 end questao2_arch;
78 -- a definicao termina por end
79
80
81
```

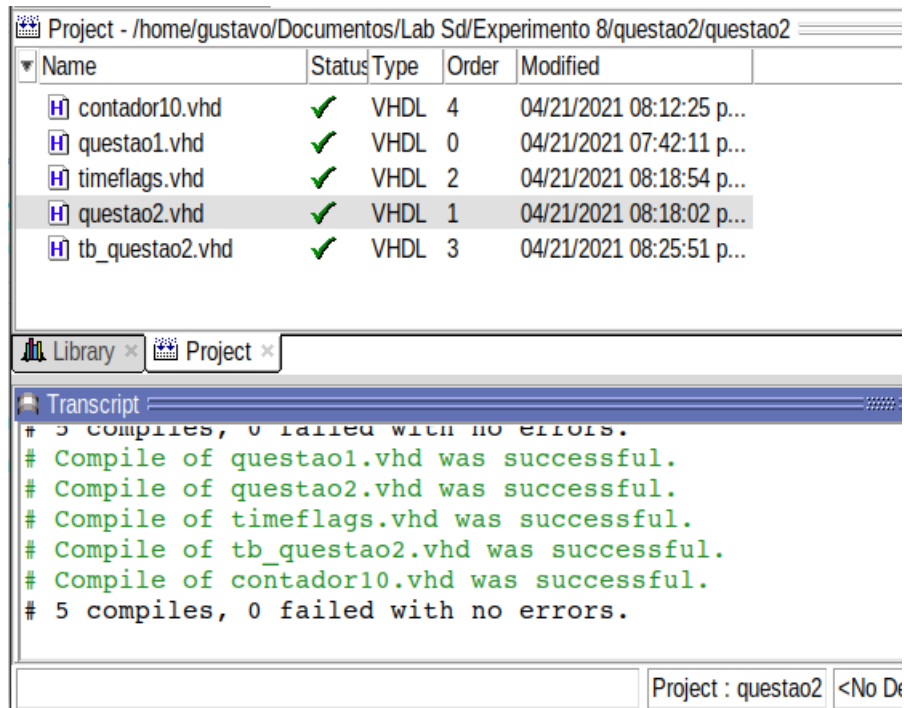

timeflags.vhd

```
1  -- Universidade de Brasilia
2  -- Laboratório de Sistemas Digitais
3  -- Autor : Gustavo Pereira Chaves
4  -- Matrícula: 19/0014113
5  -- Data : 23/04/2021
6
7  -- Implementação do sistema que verifica se já se passaram 5 segundos (T5), 6 segundos (T6), 20 segundos (T20), 60 segundos (T60)
8
9  -- *****
10 -- Circuito: Verificador de tempo
11 --          cont_dezena      Entrada 1
12 --          cont_unidade     Entrada 2
13 --          T5               Saida 1
14 --          T6               Saida 2
15 --          T20              Saida 3
16 --          T60              Saida 4
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 timeflags is
30     port(
31         cont_dezena, cont_unidade :in std_logic_vector(3 downto 0);
32         T5, T6, T20, T60 :out std_logic
33     );
34 end timeflags;
35
36 --Architecture (Arquitetura)
37 --implementacoes do projeto
38
39 architecture timeflags_arch of timeflags is
40
41     -- Sinais auxiliares para a interconexao ao
42     -- processo de estimulo
43
44     signal contador: std_logic_vector(7 downto 0);
45
46
47 begin
48     contador <= cont_dezena & cont_unidade;
49
50     T5 <= '1' when contador >= x"05" else '0';
51     T6 <= '1' when contador >= x"06" else '0';
52     T20 <= '1' when contador >= x"20" else '0';
53     T60 <= '1' when contador >= x"60" else '0';
54
55 end timeflags_arch;
56 -- a definicao termina por end
```

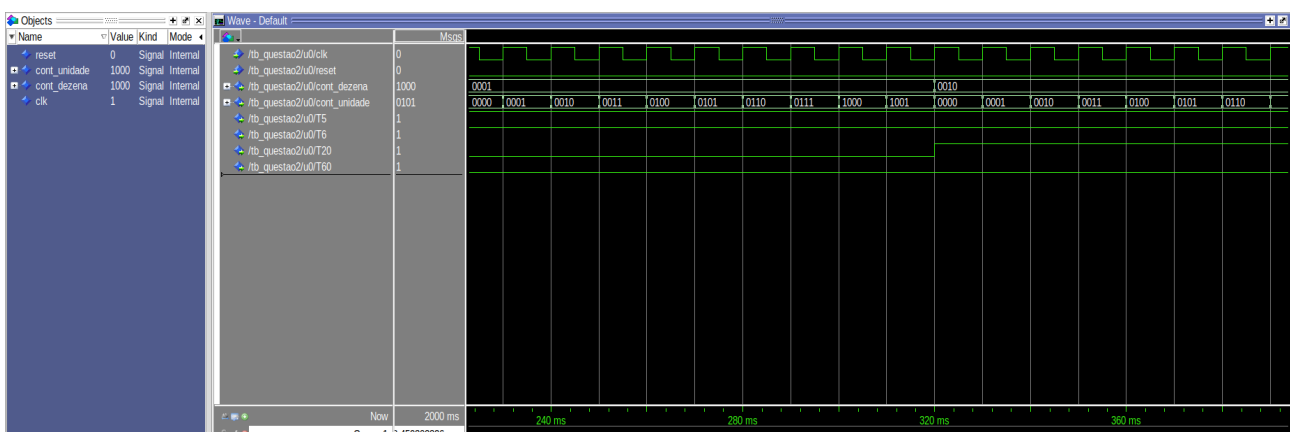
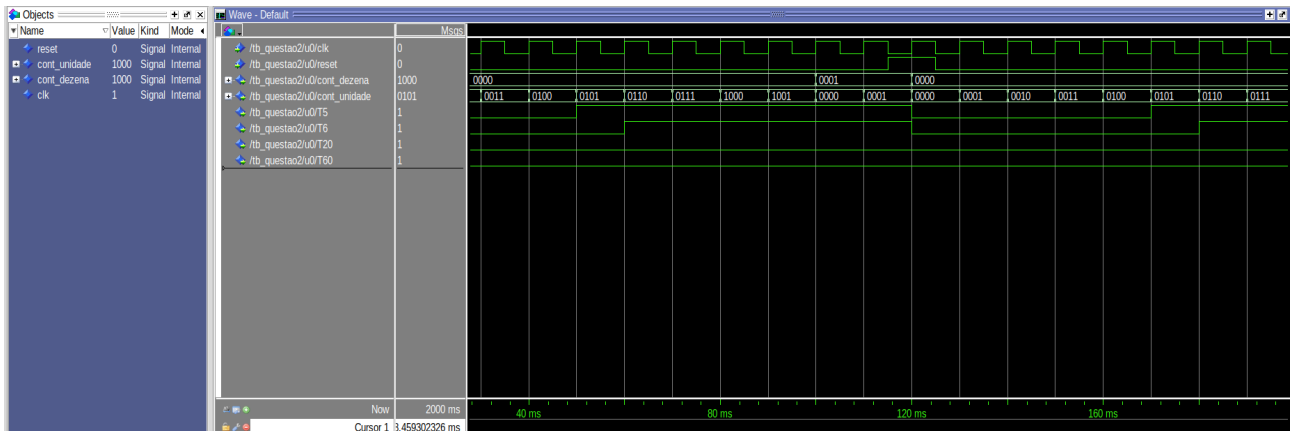

tb_questao2.vhd

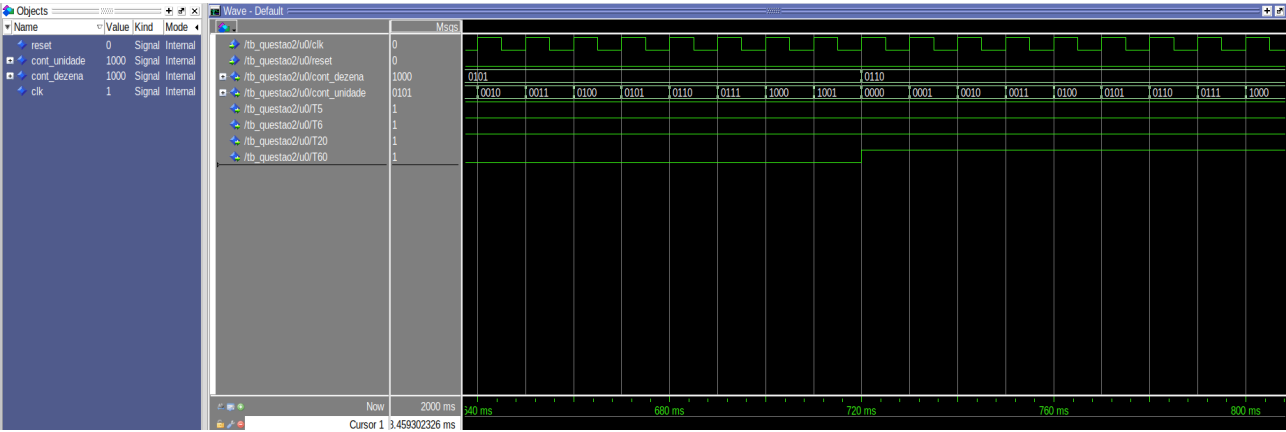
```
1  -- Universidade de Brasília
2  -- Laboratório de Sistemas Digitais
3  -- Autor : Gustavo Pereira Chaves
4  -- Matrícula: 19/0014113
5  -- Data : 16/04/2021
6
7  -- Implementação do sistema de temporizacao do controle de semaforos indicando
8  -- que já se passaram 5 segundos (T5), 6 segundos (T6), 20 segundos (T20), 60 segundos (T60)
9
10 -- *****
11 -- Testbench para simulação funcional do
12 -- Circuito: Temporizador do semaforo
13 --          clk          Entrada 1
14 --          reset        Entrada 2
15 --          T5            Saida 1
16 --          T6            Saida 2
17 --          T20           Saida 3
18 --          T60           Saida 4
19 --          cont_dezena   Saida 5
20 --          cont_unidade  Saida 6
21 -- *****
22
23 -- entity (Entidade)
24 -- tb_questao2 uma entidade sem pinos de entrada e saida
25
26 entity tb_questao2 is end;
27
28 --Package (Pacote)
29 --constantes e bibliotecas
30
31 library ieee;
32 use ieee.std_logic_1164.ALL;
33 use std.textio.all;
34 use ieee.numeric_std.all;
35
36 -- Architecture (Arquitetura)
37
38 architecture tb_questao2_arch of tb_questao2 is
39
40 -- Declaracao do componente questao2, referente a sua arquitetura descrita no arquivo questao2.vhd
41
42     component questao2 is
43     port (
44         clk, reset : in std_logic;
45         cont_dezena, cont_unidade : out std_logic_vector(3 downto 0);
46         T5, T6, T20, T60: out std_logic
47     );
48     end component;
49
50 -- Sinais auxiliares para a interconexao ao
51 -- processo de estimulo
52
53     signal clk :std_logic := '1';
54     signal reset :std_logic;
55     signal cont_dezena, cont_unidade :std_logic_vector(3 downto 0);
56
57 -- Instancia do componente contador100 e interconexao do componente ao processo de estimulo
58
59 begin
60     u0: questao2 PORT MAP (clk, reset, cont_dezena, cont_unidade, open, open, open, open);
61
62 -- Implementação do clock
63 clk <= not clk after 5 ms;
64
65 -- Implementacao do processo de estimulo
66 estimulo: process
67 begin
68     reset <= '0';
69     wait for 115 ms;
70     reset <= '1';
71     wait for 10 ms;
72     reset <= '0';
73     wait;
74 -- Fim do processo de estimulo
75     end process;
76
77 --Fim da definição da arquitetura
78 END;
```

- **Compilação:**



- **Simulação do Código:**





Questão 3:

- Documentação:

questao3.vhd

```
1  -- Universidade de Brasília
2  -- Laboratório de Sistemas Digitais
3  -- Autor : Gustavo Pereira Chaves
4  -- Matrícula: 19/0014113
5  -- Data : 23/04/2021
6
7  -- Implementação de um semáforo utilizando uma máquina de estados de Mealy
8
9  -- *****
10 -- Circuito: Semáforo utilizando máquina de estados de MEaly:
11 --          clk           Entrada 1
12 --          ligadesliga   Entrada 2
13 --          sensorA       Entrada 3
14 --          sensorB       Entrada 4
15 --          cont_dezena   Entrada 5
16 --          cont_unidade  Entrada 6
17 --          semaforoA     Saída 1
18 --          semaforoB     Saída 2
19 -- *****
20
21 --Package (Pacote)
22 --constantes e bibliotecas
23
24 library IEEE;
25 use IEEE.std_logic_1164.all;
26 use ieee.numeric_std.all;
27
28 --Entity (Entidade)
29 --pinos de entrada e saída
30
31 entity questao3 is
32     port (
33         clk, ligadesliga, sensorA, sensorB: in std_logic;
34         cont_dezena, cont_unidade: out std_logic_vector(3 downto 0);
35         semaforoA, semaforoB : out std_logic_vector(2 downto 0)
36     );
37 end questao3;
38
39 --Architecture (Arquitetura)
40 --implementacoes do projeto
41
42 architecture questao3_arch of questao3 is
43     -- Declaracao do componente MaqEstados, referente a sua arquitetura descrita no arquivo maqestados.vhd
44
45     component MaqEstados is
46         port (
47             clk, ligadesliga, sensorA, sensorB, T5, T6, T20, T60 :in std_logic;
48             semaforoA, semaforoB :out std_logic_vector(2 downto 0);
49             resetcounter :out std_logic
50         );
51     end component;
52
53     -- Declaracao do componente questao2, referente a sua arquitetura descrita no arquivo questao2.vhd
54
55     component questao2 is
56         port (
57             clk, reset : in std_logic;
58             cont_dezena, cont_unidade :out std_logic_vector(3 downto 0);
59             T5, T6, T20, T60: out std_logic
60         );
61     end component;
62
63     -- Sinal auxiliares para a interconexao ao
64     -- processo de estimulo
65
66     signal T5, T6, T20, T60, resetcounter: std_logic;
67
68     begin
69         -- Instancias dos componentes e interconexao dos mesmos:
70         u0: MaqEstados PORT MAP (clk, ligadesliga, sensorA, sensorB, T5, T6, T20, T60, semaforoA, semaforoB, resetcounter);
71         u1: questao2 PORT MAP (clk, resetcounter, cont_dezena, cont_unidade, T5, T6, T20, T60);
72
73     end questao3_arch;
74     -- a definicao termina por end
```

maestados.vhd

```

1  -- Universidade de Brasília
2  -- Laboratório de Sistemas Digitais
3  -- Autor : Gustavo Pereira Chaves
4  -- Matrícula: 19/0014113
5  -- Data : 23/04/2021
6
7  -- Implementação da máquina de estados que será utilizada no semáforo
8
9  -- *****
10 -- Circuito: Máquina de Estados de Mealy do Semáforo:
11 --      clk      Entrada 1
12 --      ligadesliga Entrada 2
13 --      sensorA   Entrada 3
14 --      sensorB   Entrada 4
15 --      T5        Entrada 5
16 --      T6        Entrada 6
17 --      T20       Entrada 7
18 --      T60       Entrada 8
19 --      semaforoA Saída 1
20 --      semaforoB Saída 2
21 --      resetcounter Saída 3
22 -- *****
23
24 --Package (Pacote)
25 --constantes e bibliotecas
26
27 library IEEE;
28 use IEEE.std_logic_1164.all;
29 use ieee.numeric_std.all;
30
31 --Entity (Entidade)
32 --pinos de entrada e saída
33
34 entity MaqEstados is
35     port (
36         clk, ligadesliga, sensorA, sensorB, T5, T6, T20, T60 :in std_logic;
37         semaforoA, semaforoB :out std_logic_vector(2 downto 0);
38         resetcounter :out std_logic
39     );
40 end MaqEstados;
41
42 --Architecture (Arquitetura)
43 --implementacoes do projeto
44
45 architecture MaqEstados_arch of MaqEstados is
46 -- Definição de tipo state que indica os estados possíveis
47     type state is (vermelhoA, verdeA, trocaA, vermelhoB, verdeB, trocaB, desligado_vermelho, desligado_amarelo);
48
49 -- Sinais auxiliares para armazenar o estado atual e o próxima estado respectivamente
50     signal current_state :state := vermelhoA;
51     signal next_state :state;
52
53 -- a definicao inicia por
54 begin
55
56 -- Processo síncrono da maquina de estados
57     sync_process: process (clk, ligadesliga)
58     begin
59         if rising_edge(clk) or rising_edge(ligadesliga) or falling_edge(ligadesliga) then
60             current_state <= next_state;
61             resetcounter <= '0';
62         end if;
63     end process;
64
65 -- Processo combinacional da maquina de estados
66     comb_process: process (current_state, T5, T6, T20, T60, ligadesliga)
67     begin
68         case current_state is
69             when vermelhoA =>
70                 if ligadesliga = '0' then
71                     next_state <= desligado_vermelho;
72                     resetcounter <= '1';
73                 else
74                     semaforoA <= "100"; semaforoB <= "100";
75                     if T5 = '1' then
76                         next_state <= verdeA;
77                         resetcounter <= '1';
78                     else
79                         next_state <= current_state;
80                     end if;
81                 end if;
82             when verdeA =>
83                 if ligadesliga = '0' then
84                     next_state <= desligado_vermelho;
85                     resetcounter <= '1';
86                 else
87                     semaforoA <= "001"; semaforoB <= "100";
88                     if T60 = '1' then
89                         next_state <= trocaA;
90                         resetcounter <= '1';
91                     elsif T20 = '1' then
92                         if sensorA = '1' then next_state <= current_state;
93                         elsif sensorB = '1' then
94                             next_state <= trocaA;
95                             resetcounter <= '1';
96                         end if;
97                     else
98                         next_state <= current_state;
99                     end if;
100                 end if;
101         end if;

```

```

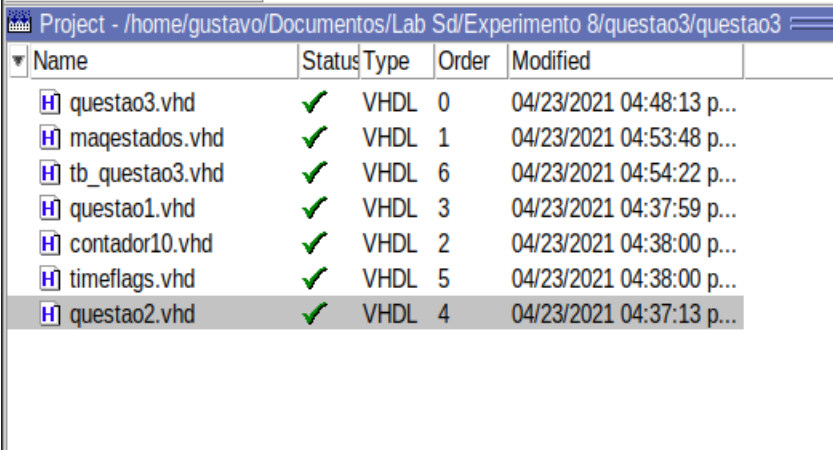
102
103
104     when trocaA =>
105         if ligadesliga = '0' then
106             next_state <= desligado_vermelho;
107             resetcounter <= '1';
108         else
109             semaforoA <= "110"; semaforoB <= "100";
110             if T6 = '1' then
111                 next_state <= vermelhoB;
112                 resetcounter <= '1';
113             else
114                 next_state <= current_state;
115             end if;
116         end if;
117
118     when vermelhoB =>
119         if ligadesliga = '0' then
120             next_state <= desligado_vermelho;
121             resetcounter <= '1';
122         else
123             semaforoA <= "100"; semaforoB <= "100";
124             if T5 = '1' then
125                 next_state <= verdeB;
126                 resetcounter <= '1';
127             else
128                 next_state <= current_state;
129             end if;
130         end if;
131
132     when verdeB =>
133         if ligadesliga = '0' then
134             next_state <= desligado_vermelho;
135             resetcounter <= '1';
136         else
137             semaforoA <= "100"; semaforoB <= "001";
138             if T60 = '1' then
139                 next_state <= trocaB;
140                 resetcounter <= '1';
141             elsif T20 = '1' then
142                 if sensorB = '1' then next_state <= current_state;
143                 elsif sensorA = '1' then
144                     next_state <= trocaB;
145                     resetcounter <= '1';
146                 end if;
147             else
148                 next_state <= current_state;
149             end if;
150         end if;
151
152     when trocaB =>
153         if ligadesliga = '0' then
154             next_state <= desligado_vermelho;
155             resetcounter <= '1';
156         else
157             semaforoA <= "100"; semaforoB <= "110";
158             if T6 = '1' then
159                 next_state <= vermelhoA;
160                 resetcounter <= '1';
161             else
162                 next_state <= current_state;
163             end if;
164         end if;
165
166     when desligado_vermelho =>
167         semaforoA <= "100"; semaforoB <= "100";
168         if ligadesliga = '0' then
169             next_state <= vermelhoA;
170             resetcounter <= '1';
171         else
172             next_state <= desligado_amarelo;
173             resetcounter <= '1';
174         end if;
175
176     when desligado_amarelo =>
177         semaforoA <= "010"; semaforoB <= "010";
178         if ligadesliga = '0' then
179             next_state <= vermelhoA;
180             resetcounter <= '1';
181         else
182             next_state <= desligado_vermelho;
183             resetcounter <= '1';
184         end if;
185     end case;
186 end process;
187
188 end MaqEstados_arch;
189 -- a definicao termina por end
190

```

tb_questao3.vhd

```
1  -- Universidade de Brasília
2  -- Laboratório de Sistemas Digitais
3  -- Autor : Gustavo Pereira Chaves
4  -- Matrícula: 19/0014113
5  -- Data : 23/04/2021
6
7  -- Implementação de um semáforo utilizando uma máquina de estados de Mealy
8
9  -- *****
10 -- Testbench para simulação funcional do
11 -- Circuito: Semáforo utilizando máquina de estados de MEaly:
12 --      clk      Entrada 1
13 --      ligadesliga  Entrada 2
14 --      sensorA    Entrada 3
15 --      sensorB    Entrada 4
16 --      cont_dezena Entrada 5
17 --      cont_unidade Entrada 6
18 --      semaforoA  Saida 1
19 --      semaforoB  Saida 2
20 -- *****
21
22 -- entity (Entidade)
23 -- tb_questao2 uma entidade sem pinos de entrada e saída
24
25 entity tb_questao3 is end;
26
27 --Package (Pacote)
28 --constantes e bibliotecas
29
30 library ieee;
31 use ieee.std_logic_1164.ALL;
32 use std.textio.all;
33 use ieee.numeric_std.all;
34
35 -- Architecture (Arquitetura)
36
37 architecture tb_questao3_arch of tb_questao3 is
38
39 -- Declaracao do componente questao3, referente a sua arquitetura descrita no arquivo questao3.vhd
40
41     component questao3 is
42     port (
43         clk, ligadesliga, sensorA, sensorB: in std_logic;
44         cont_dezena, cont_unidade: out std_logic_vector(3 downto 0);
45         semaforoA, semaforoB : out std_logic_vector(2 downto 0)
46     );
47     end component;
48
49 -- Sinais auxiliares para a interconexao ao
50 -- processo de estimulo
51
52     signal clk :std_logic := '1';
53     signal ligadesliga, sensorA, sensorB :std_logic;
54
55 -- Instancia do componente contador100 e interconexao do componente ao processo de estimulo
56
57 begin
58     u0: questao3 PORT MAP (clk, ligadesliga, sensorA, sensorB, open, open, open, open);
59
60 -- Implementação do clock
61 clk <= not clk after 0.5 sec;
62
63 -- Implementacao do processo de estimulo
64 estimulo: process
65 begin
66     ligadesliga <= '1';
67     sensorA <= '1'; sensorB <= '1';
68     wait for 30 sec;
69
70     ligadesliga <= '0';
71     wait for 10 sec;
72
73     ligadesliga <= '1';
74     sensorB <= '0';
75     wait for 30 sec;
76
77     sensorA <= '0';
78     wait for 60 sec;
79
80     wait;
81 -- Fim do processo de estimulo
82 end process;
83
84 --Fim da definição da arquitetura
85 END;
```


- **Compilação:**



Project - /home/gustavo/Documents/Lab Sd/Experimento 8/questao3/questao3

Name	Status	Type	Order	Modified
questao3.vhd	✓	VHDL	0	04/23/2021 04:48:13 p...
maquestados.vhd	✓	VHDL	1	04/23/2021 04:53:48 p...
tb_questao3.vhd	✓	VHDL	6	04/23/2021 04:54:22 p...
questao1.vhd	✓	VHDL	3	04/23/2021 04:37:59 p...
contador10.vhd	✓	VHDL	2	04/23/2021 04:38:00 p...
timeflags.vhd	✓	VHDL	5	04/23/2021 04:38:00 p...
questao2.vhd	✓	VHDL	4	04/23/2021 04:37:13 p...

Library x Project x

Transcript

```
# Compile of tb_questao3.vhd was successful.
# Compile of questao3.vhd was successful.
# Compile of maquestados.vhd was successful.
# Compile of contador10.vhd was successful.
# Compile of questao1.vhd was successful.
# Compile of questao2.vhd was successful.
# Compile of timeflags.vhd was successful.
# Compile of tb_questao3.vhd was successful.
# 7 compiles, 0 failed with no errors.
```

- **Simulação do Código:**

