



Teoria de Eletrônica Digital 2 - 1/2017

Teste 3 - 23/03/2017

Prazo: Dia 28/03/17, às 09:55h

Descreva, em VHDL, os modelos de Moore e de Mealy de uma máquina de estados síncrona com uma entrada **A** e uma saída **Z** que será igual a **1** se ocorrer a sequência **1,0,0,1** na entrada; caso contrário, **Z = 0**. Considere que o final da palavra anterior pode ser considerado como o início da próxima palavra da sequência.

Para simular, utilize um *testbench* para gerar uma sequência de bits correspondente aos dois últimos dígitos do seu número de matrícula. Por exemplo, para a matrícula **01/234567**, os dois últimos dígitos são **6 (0110)** e **7 (0111)** e a sequência, portanto, é **0, 1, 1, 0, 0, 1, 1, 1**.

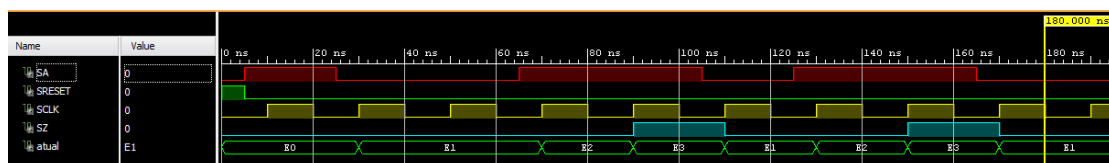
Enviar, via Moodle, um arquivo PDF contendo um *screenshot* das formas de onda mostrando as entradas de *clock* e *reset*, a saída de Moore e o estado atual, e outro *screenshot* mostrando as entradas de *clock* e *reset*, a saída de Mealy e o estado atual. Incluir uma breve descrição textual dos resultados, o nome e o número de matrícula.

O exemplo a seguir mostra as descrições e simulações do detector da sequência **0, 1, 1**, mostrado em aula.



```
25 entity detectorfsm is
26     Port ( A : in STD_LOGIC;
27           CLK : in STD_LOGIC;
28           RESET: in STD_LOGIC;
29           Z : out STD_LOGIC);
30 end detectorfsm;
31
32 architecture Behavioral of detectorfsm is
33
34     type estado is (E0, E1, E2, E3);
35     signal atual, proximo: estado;
36
37 begin
38
39     armazena_estado: process(CLK, RESET)
40     begin
41         if RESET = '1' then
42             atual <= E0;
43         elsif rising_edge(CLK) then
44             atual <= proximo;
45         end if;
46     end process;
47
48     transicao: process(atual, A)
49     begin
50         case atual is
51             when E0 =>
52                 Z <= '0';
53                 if A = '0' then
54                     proximo <= E1;
55                 else
56                     proximo <= E0;
57                 end if;
58             when E1 =>
59                 Z <= '0';
60                 if A = '0' then
61                     proximo <= E1;
62                 else
63                     proximo <= E2;
64                 end if;
65             when E2 =>
66                 Z <= '0';
67                 if A = '0' then
68                     proximo <= E1;
69                 else
70                     proximo <= E3;
71                 end if;
72             when E3 =>
73                 Z <= '1';
74                 if A = '0' then
75                     proximo <= E1;
76                 else
77                     proximo <= E0;
78                 end if;
79         end case;
80     end process;
81 end Behavioral;
```

```
34 entity detectorfsm_tb is
35 end detectorfsm_tb;
36
37 architecture Behavioral of detectorfsm_tb is
38
39     component detectorfsm
40     Port ( A : in STD_LOGIC;
41           CLK : in STD_LOGIC;
42           RESET: in STD_LOGIC;
43           Z : out STD_LOGIC);
44     end component;
45
46     signal SA, SRESET, SCLK, SZ: STD_LOGIC;
47
48 begin
49
50     uut: detectorfsm port map ( A => SA,
51                                CLK => SCLK,
52                                RESET => SRESET,
53                                Z => SZ);
54
55     clk: process
56     begin
57         SCLK <= '0';
58         wait for 10 ns;
59         SCLK <= '1';
60         wait for 10 ns;
61     end process;
62
63     stim: process
64     begin
65         SRESET <= '1'; SA <= '0';
66         wait for 5 ns;
67         SRESET <= '0'; SA <= '1';
68         wait for 20 ns;
69         SA <= '0';
70         wait for 40 ns;
71         SA <= '1';
72         wait for 40 ns;
73         SA <= '0';
74         wait for 20 ns;
75         SA <= '1';
76         wait for 40 ns;
77         SA <= '0';
78         wait;
79     end process;
80
81 end Behavioral;
```

Descrição VHDL do modelo de Moore e *testbench*

Simulação do modelo de Moore



```

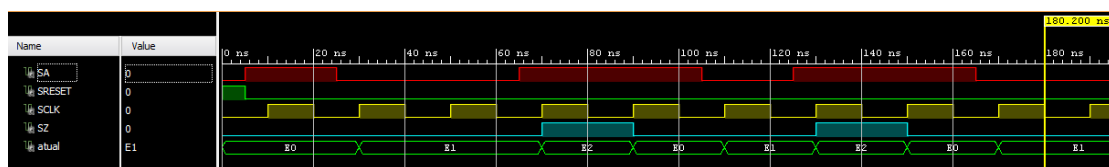
34 entity detectorfsmmealy is
35     Port ( A : in STD_LOGIC;
36           CLK : in STD_LOGIC;
37           RESET: in STD_LOGIC;
38           Z : out STD_LOGIC);
39 end detectorfsmmealy;
40
41 architecture Behavioral of detectorfsmmealy is
42
43     type estado is (E0, E1, E2);
44     signal atual, proximo: estado;
45
46 begin
47
48     armazena_estado: process (CLK, RESET)
49     begin
50         if RESET = '1' then
51             atual <= E0;
52         elsif rising_edge (CLK) then
53             atual <= proximo;
54         end if;
55     end process;
56
57     transicao: process (atual, A)
58     begin
59         case atual is
60             when E0 =>
61                 if A = '0' then
62                     proximo <= E1;
63                     Z <= '0';
64                 else
65                     proximo <= E0;
66                     Z <= '0';
67                 end if;
68             when E1 =>
69                 if A = '0' then
70                     proximo <= E1;
71                     Z <= '0';
72                 else
73                     proximo <= E2;
74                     Z <= '0';
75                 end if;
76             when E2 =>
77                 if A = '0' then
78                     proximo <= E1;
79                     Z <= '0';
80                 else
81                     proximo <= E0;
82                     Z <= '1';
83                 end if;
84             end case;
85         end process;
86 end Behavioral;

```

```

34 entity detectorfsmmealy_tb is
35 end detectorfsmmealy_tb;
36
37 architecture Behavioral of detectorfsmmealy_tb is
38
39     component detectorfsmmealy
40     Port ( A : in STD_LOGIC;
41           CLK : in STD_LOGIC;
42           RESET: in STD_LOGIC;
43           Z : out STD_LOGIC);
44     end component;
45
46     signal SA, SRESET, SCLK, SZ: STD_LOGIC;
47
48 begin
49
50     uut: detectorfsmmealy port map ( A => SA,
51                                     CLK => SCLK,
52                                     RESET => SRESET,
53                                     Z => SZ);
54
55     clk: process
56     begin
57         SCLK <= '0';
58         wait for 10 ns;
59         SCLK <= '1';
60         wait for 10 ns;
61     end process;
62
63     stim: process
64     begin
65         SRESET <= '1'; SA <= '0';
66         wait for 5 ns;
67         SRESET <= '0'; SA <= '1';
68         wait for 20 ns;
69         SA <= '0';
70         wait for 40 ns;
71         SA <= '1';
72         wait for 40 ns;
73         SA <= '0';
74         wait for 20 ns;
75         SA <= '1';
76         wait for 40 ns;
77         SA <= '0';
78         wait;
79     end process;
80
81
82 end Behavioral;
83

```

Descrição VHDL do modelo de Mealy e *testbench*

Simulação do modelo de Mealy