Relatório 7

Nome: Gabriel Cruz Vaz Santos
Matrícula: 200049038
Turma: C

- ## Questão 1

Introdução

Foi solicitado a implementação de uma máquina de estados síncrona do tipo Moore para controlar uma máquina de refrigerantes que aceita moedas de R$0,25 e R$0,50.

Teoria

A máquina terá 3 entradas diferentes: as portas lógicas clock e reset e o vetor lógico A de tamanho 2. A entrada A é referente a qual operação foi realizada, da seguinte forma:

- A = “00” => nenhuma solicitação foi feita
- A =”01” => inserido uma moeda de 25 centavos
- A = “10” => inserido moeda de 50 centavos
- A = “11” => cancelamento

A máquina consta como saída “000”, cada caractere indicando respectivamente quantos refrigerantes, moedas de 50 centavos e moedas de 25 centavos foram retornados pela máquina.

Código

Os códigos foram programados no ambiente de desenvolvimento integrado Visual Studio Code, na linguagem VHDL e compilados pelo software do ModelSim. As imagens 1, 2, 3 e 4 são da implementação da entidade da máquina de estado e as demais do testbench.

Relatorio7 > refrigerante.vhd

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity maquinaEstados is
    port(
        A: in std_logic_vector (1 downto 0);
        clock, reset: in std_logic;
        Y: out std_logic_vector (2 downto 0)
    );

end maquinaEstados;

architecture maquinaEstados_arch of maquinaEstados is

    type estado is (init, c25, c50, c75, c100, c125, d25, d50, d75);
    signal currentState, nextState : estado;

begin

    sync_proc: process(clock, reset)
    begin
        if reset = '1' then
            currentState <= init;
        elsif rising_edge(clock) then
            currentState <= nextState;
        end if;
    end process sync_proc;

    comb_proc: process(currentState, A)
    begin
        case currentState is
```

```
32              when init =>
33                  Y <= "000";
34                  if(A = "01") then nextState <= c25;
35                  elsif(A = "10") then nextState <= c50;
36                  elsif(A = "11") then nextState <= init;
37                  else nextState <= init;
38                  end if;
39              when c25 =>
40                  Y <= "000";
41                  if(A = "01") then nextState <= c50;
42                  elsif(A = "10") then nextState <= c75;
43                  elsif(A = "11") then nextState <= d25;
44                  else nextState <= c25;
45                  end if;
46              when c50 =>
47                  Y <= "000";
48                  if(A = "01") then nextState <= c75;
49                  elsif(A = "10") then nextState <= c100;
50                  elsif(A = "11") then nextState <= d50;
51                  else nextState <= c50;
52                  end if;
53              when c75 =>
54                  Y <= "000";
55                  if(A = "01") then nextState <= c100;
56                  elsif(A = "10") then nextState <= c125;
57                  elsif(A = "11") then nextState <= d75;
58                  else nextState <= c75;
59                  end if;
60              when c100 =>
61                  Y <= "100";
```

```
62                  if(A = "01") then nextState <= c25;
63                  elsif(A = "10") then nextState <= c50;
64                  elsif(A = "11") then nextState <= init;
65                  else nextState <= init;
66                  end if;
67              when c125 =>
68                  Y <= "110";
69                  if(A = "01") then nextState <= c25;
70                  elsif(A = "10") then nextState <= c50;
71                  elsif(A = "11") then nextState <= init;
72                  else nextState <= init;
73                  end if;
74              when d25 =>
75                  Y <= "010";
76                  if(A = "01") then nextState <= c25;
77                  elsif(A = "10") then nextState <= c50;
78                  elsif(A = "11") then nextState <= init;
79                  else nextState <= init;
80                  end if;
81              when d50 =>
82                  Y <= "001";
83                  if(A = "01") then nextState <= c25;
84                  elsif(A = "10") then nextState <= c50;
85                  elsif(A = "11") then nextState <= init;
86                  else nextState <= init;
87                  end if;
88              when d75 =>
89                  Y <= "011";
90                  if(A = "01") then nextState <= c25;
91                  elsif(A = "10") then nextState <= c50;
```

```
92                  elsif(A = "11") then nextState <= init;
93                  else nextState <= init;
94                  end if;
95              when others => nextState <= nextState;
96          end case;
97      end process comb_proc;
98
99  end maquinaEstados_arch ; -- maquinaEstados_arch
```

Relatorio7 > refrigerante_tb.vhd

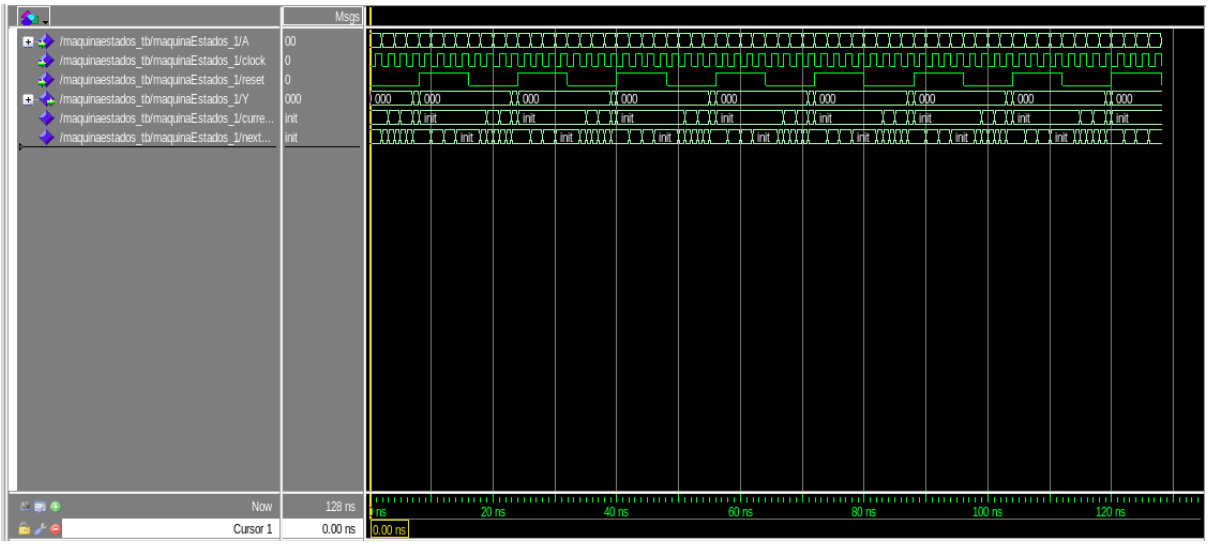
```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity maquinaEstados_tb is end;

architecture maquinaEstados_arch of maquinaEstados_tb is
    component maquinaEstados is
        port(
            A: in std_logic_vector (1 downto 0);
            clock, reset: in std_logic;
            Y: out std_logic_vector (2 downto 0)
        );
    end component;

    signal entrance: std_logic_vector (3 downto 0) := "0000";

    constant time_0: time := 2 ns;
    constant time_1: time := 4 ns;
    constant time_2: time := 8 ns;
    constant time_3: time := 16 ns;


begin
    maquinaEstados_1: maquinaEstados port map (clock => entrance(0), A(0) => entrance(1), A(1) => entrance(2), reset => entrance(3

    entrance(0) <= not entrance(0) after time_0/2;
    entrance(1) <= not entrance(1) after time_1/2;
    entrance(2) <= not entrance(2) after time_2/2;
    entrance(3) <= not entrance(3) after time_3/2;
end maquinaEstados_arch ; -- maquinaEstados_arch
```

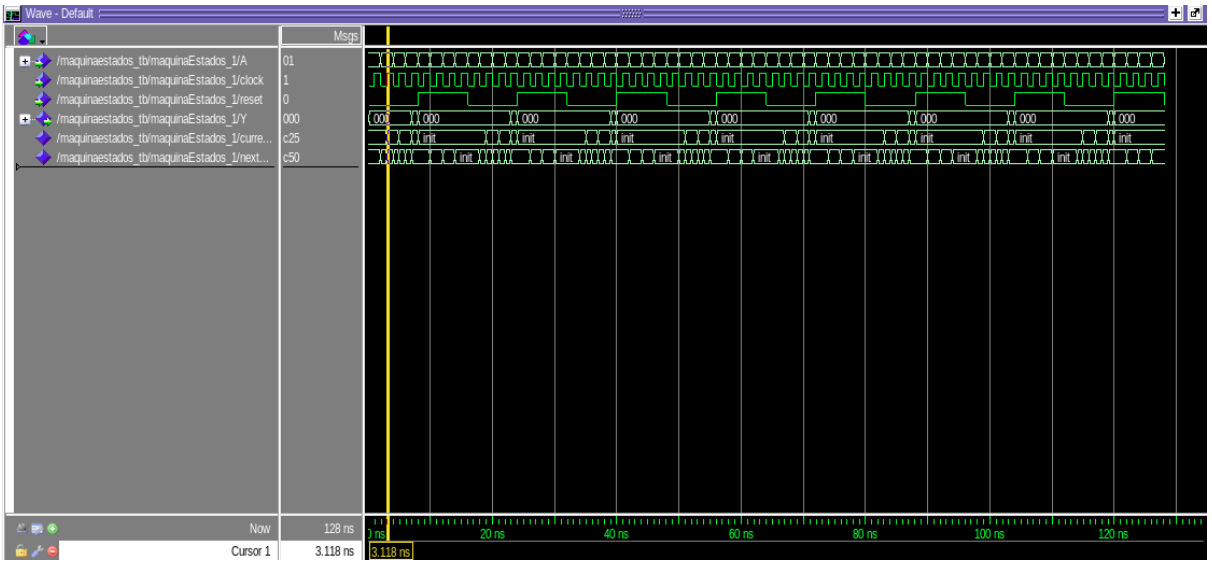
Compilador

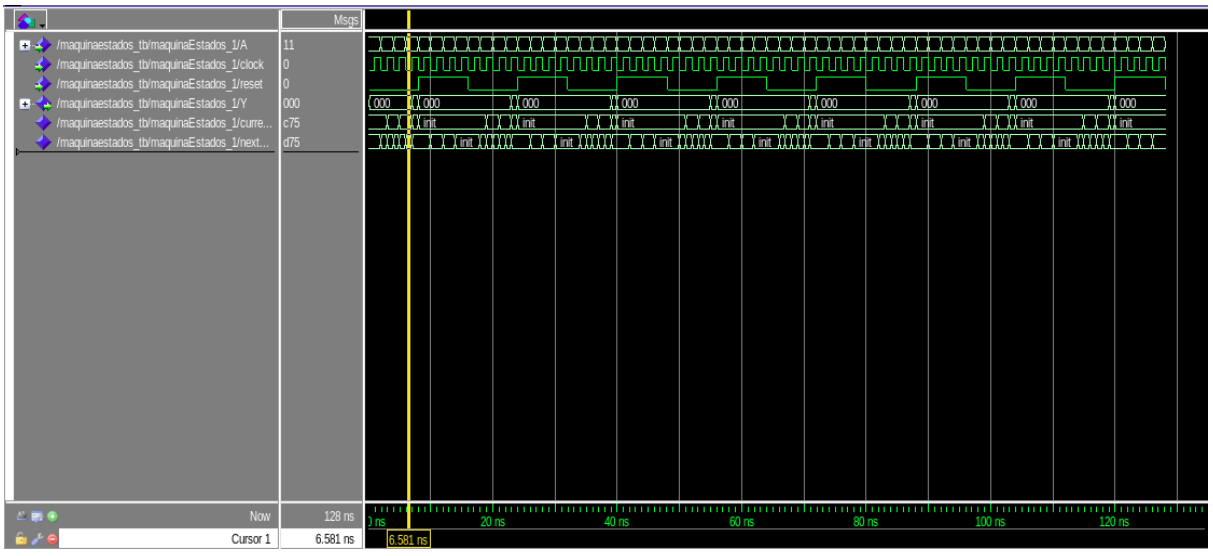
Os códigos acima foram compilados para garantir seu funcionamento, através do compilador do Modelsim. Não apresentam erros de sintaxe;

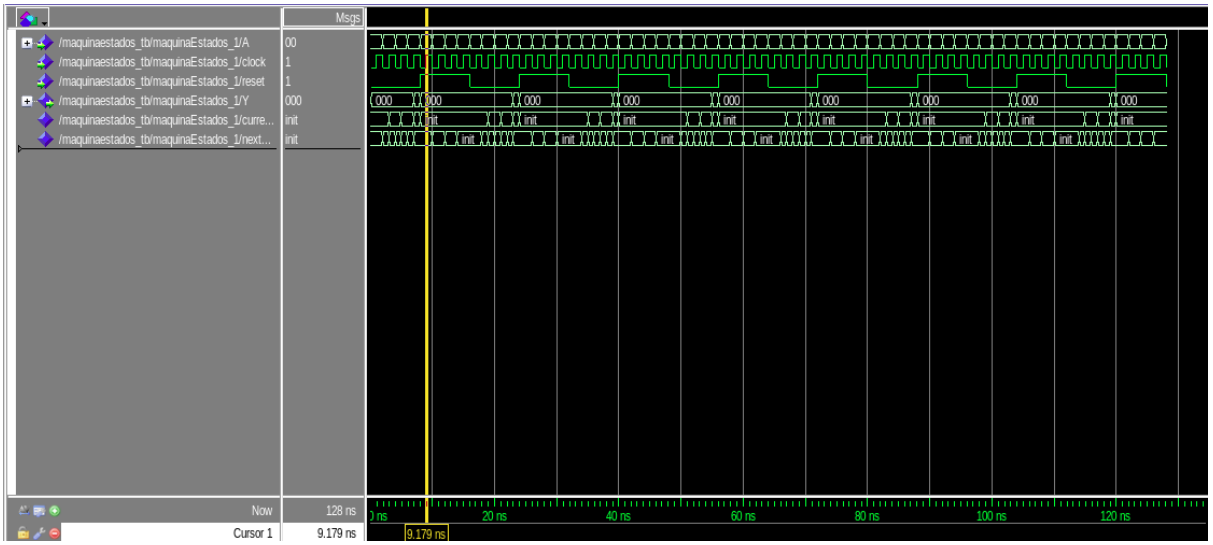
```
# Compile of refrigerante.vhd was successful.
# Compile of refrigerante_tb.vhd was successful.
```

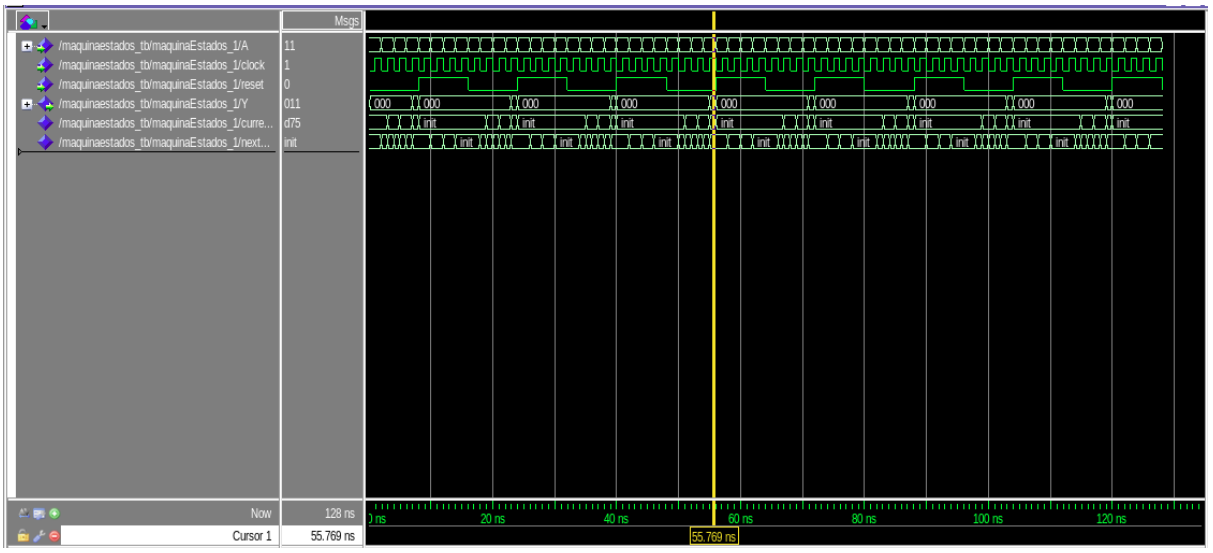
Simulação

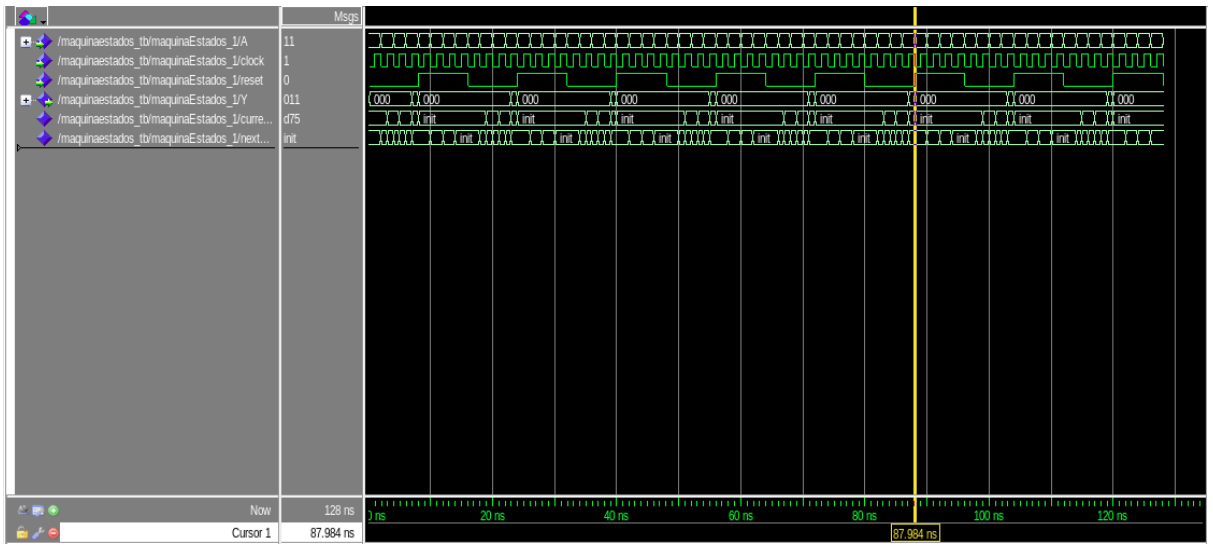

Msgs
/maquinaestados_tb/maquinaEstados_1/A
/maquinaestados_tb/maquinaEstados_1/clock
/maquinaestados_tb/maquinaEstados_1/reset
/maquinaestados_tb/maquinaEstados_1/Y
/maquinaestados_tb/maquinaEstados_1/curre...
/maquinaestados_tb/maquinaEstados_1/next...
00
0
0
000
init
init
Now
128 ns
Cursor 1
0.00 ns
20 ns
40 ns
60 ns
80 ns
100 ns
120 ns


Wave - Default
Msgs
/maquinaestados_tb/maquinaEstados_1/A
/maquinaestados_tb/maquinaEstados_1/clock
/maquinaestados_tb/maquinaEstados_1/reset
/maquinaestados_tb/maquinaEstados_1/Y
/maquinaestados_tb/maquinaEstados_1/curre...
/maquinaestados_tb/maquinaEstados_1/next...
01
1
0
000
c25
c50
Now
128 ns
Cursor 1
3.118 ns
20 ns
40 ns
60 ns
80 ns
100 ns
120 ns


Msgs
/maquinaestados_tb/maquinaEstados_1/A
/maquinaestados_tb/maquinaEstados_1/clock
/maquinaestados_tb/maquinaEstados_1/reset
/maquinaestados_tb/maquinaEstados_1/Y
/maquinaestados_tb/maquinaEstados_1/curre...
/maquinaestados_tb/maquinaEstados_1/next...
11
0
0
000
c75
d75
Now
128 ns
Cursor 1
6.581 ns
20 ns
40 ns
60 ns
80 ns
100 ns
120 ns







Análise

O processo síncrono da nossa máquina de refrigerantes garante que o estado atual (current_state) da máquina sempre será inicial (init) quando reset =’1’ e

estará recebendo o próximo estado (next_state) quando reset =’0’ e estamos na borda de subida de clock.

A imagem 2 mostra o caso no qual a máquina já possui 25 centavos e estão inserindo mais 25 centavos, como mostra a entrada A (01). Nesse caso o nextState foi para (c50), indicando que a máquina já tem 50 centavos.

A imagem 3 mostra o caso que a máquina já tem 75 centavos e estão querendo cancelar o pedido, como mostra a entrada A (11). Nesse caso, o nextState foi para ‘d75’, e saída agora será “011”, devolvendo uma moeda de 50 e outra de 25 centavos.

As últimas imagens mostram que caso o estado da máquina seja de devolver dinheiro e seja solicitado cancelamento (A = “11”), o próximo estado da máquina será para o inicial.

Conclusão

Nesse experimento conseguimos com êxito implementar em vhdl uma máquina de estados Moore. As simulações se comportaram como esperado e de acordo com a tabela disponibilizada no exercício.