UNIVERSIDADE FEDERAL DA FRONTEIRA SUL CAMPUS DE CHAPECÓ
CURSO DE CIÊNCIA DA COMPUTAÇÃO

EMILY ZANIVAN DALLAZEN
Matrícula: 2221101066
E-mail: emilydwllazen@gmail.com

LUIZ GUILHERME ZANELLA LOPES
Matrícula: 2211100006
E-mail: zanelallopes9977@gmail.com

RELATÓRIO DE TRABALHO DE SISTEMAS DIGITAIS

VENDING MACHINE DE REFRIGERANTE

CHAPECÓ
2023

INTRODUÇÃO

Esse trabalho foi desenvolvido como parte do componente curricular de Sistemas Digitais, no curso de Ciência da Computação. Foi proposto o desenvolvimento de uma máquina de estados que representa uma máquina de refrigerantes. Ao receber o valor total de R$1,50 ou mais, o cliente poderá escolher um dos refrigerantes disponíveis e então a máquina estará pronta para sinalizar que o refrigerante será liberado. A máquina não realiza cálculo de troco, e qualquer moeda diferente de um real e de cinquenta centavos será desconsiderada pelo circuito. De mesmo modo, qualquer valor inserido a mais pelo cliente não será devolvido.

1.0 DESENVOLVIMENTO

A lógica da máquina de estados implementa um controle sequencial que gerencia as transições de estado com base nas moedas inseridas e na escolha do refrigerante. A máquina de Moore gera a saída de acordo com o estado atual, indicando qual refrigerante foi escolhido ou se nenhum foi selecionado. O refrigerante pode ser escolhido a qualquer momento, tanto antes de inserir as moedas, quanto depois.

1.1 DIAGRAMA DE ESTADOS

Para cada etapa da máquina, foi criado um estado de transição que auxiliará no controle sequencial de estados. Os estados utilizados foram:

- q_00: Estado inicial. Aguarda a inserção de moedas.
- q_05: Moeda de 50 centavos inserida. Aguarda mais moedas ou a escolha do refrigerante.
- q_10: Moeda de 1 real inserida. Aguarda mais moedas ou a escolha do refrigerante.
- q_15: Estado intermediário após a inserção de moedas. Aguarda a escolha do refrigerante.
- q_coca: Estado final para a seleção de Coca-Cola.
- q_pepsi: Estado final para a seleção de Pepsi.
- q_sprite: Estado final para a seleção de Sprite.

Com base nesses estados, foi desenvolvido o diagrama de estados e a tabela com todas as possibilidades de transições entre estes.

Para a leitura do diagrama, tomamos como base as seguintes informações:

- M é a variável de entrada de 2 bits referente à moeda inserida pelo cliente.

M	**Moeda**
00	Nenhuma
01	50 centavos
10	1 real
11	Qualquer outra

- S é a variável do seletor de refrigerante.

S	**Refrigerante**
00	Nenhum selecionado
01	Coca
10	Pepsi
11	Sprite

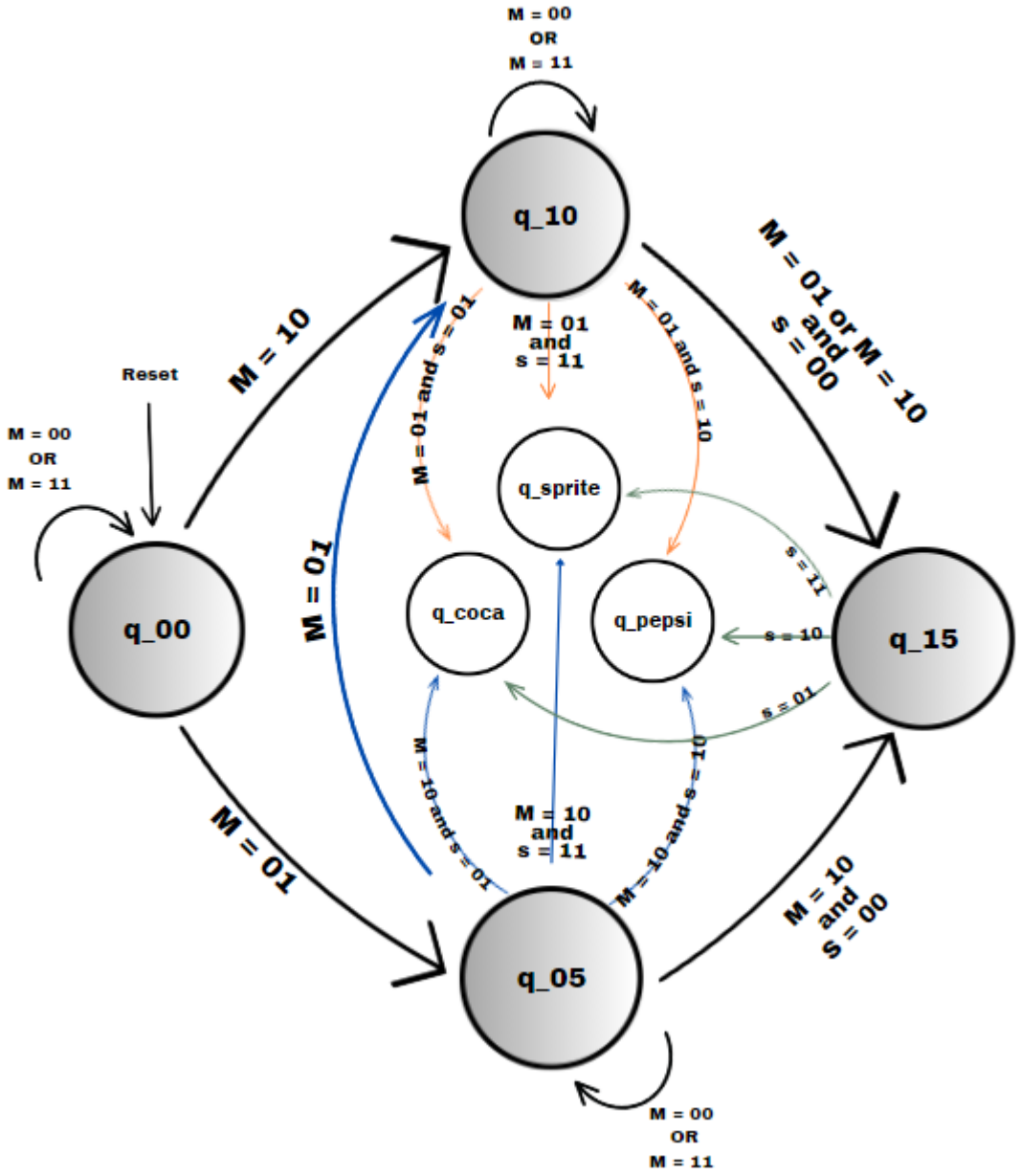

Diagrama de estados.

Abaixo, a tabela com todas as possibilidades de transições em cada estado:

Estado atual	M	S	Próximo estado
q_00	00	-	q_00
q_00	11	-	q_00
q_00	01	-	q_05
q_00	10	-	q_10
q_05	00	-	q_05
q_05	11	-	q_05
q_05	10	11	q_sprite

q_05	10	01	q_coca
q_05	10	10	q_pepsi
q_05	01	-	q_10
q_10	00	-	q_10
q_10	11	-	q_10
q_10	01	00	q_15
q_10	10	00	q_15
q_10	01	01	q_coca
q_10	01	11	q_sprite
q_10	01	10	q_pepsi
q_15	-	11	q_sprite
q_15	-	10	q_pepsi
q_15	-	01	q_coca

1.2 CÓDIGO EM VHDL

Com base no diagrama e na tabela de mudança de estados, foi feito o código em VHDL:

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

entity controle_refrigerante is
  port (
    Clock, Reset : in std_logic;
    moeda : in std_logic_vector(1 downto 0); -- '00' sem moedas, '01' moeda 50 cent, '10' moeda 1 real, '11' qualquer outra moeda
    seletorRefrigerante : in std_logic_vector(1 downto 0); -- '00' sem refrigerante, '01' coca, '10' pepsi, '11' sprite
    refrigerante : out std_logic_vector(1 downto 0) -- fazer exibir na fpga o refrigerante selecionado
  );
end controle_refrigerante;

```

```
13  architecture controle_refrigerante_architecture of controle_refrigerante is
14    -- tipagem da maquina de estados
15    type estado is (q_00, q_05, q_10, q_15, q_coca, q_pepsi, q_sprite);
16    -- estado atual da maquina de estados
17    signal estadoAtual : estado := q_00;
18    begin
19      -- detecção do clock e reset
20      process(Reset, Clock)
21      begin
22        if(Reset = '1') then
23          -- se reset entao estado atual inicial
24          estadoAtual <= q_00;
25        elsif(Clock'event and Clock = '1') then
26          -- senao verifica o estado atual
27          case estadoAtual is
28            -- se for q_00
29            when q_00 =>
30              if(moeda = "10") then
31                -- 1 real inserido
32                estadoAtual <= q_10;
33              elsif(moeda = "01") then
34                -- 50 centavos inserido
35                estadoAtual <= q_05;
36              end if;
```

```
37
38            -- se for q_05
39            when q_05 =>
40              if(moeda = "10") then
41
42                  -- 1 real inserido vai para o estado final
43                if(seletorRefrigerante = "00") then
44                  estadoAtual <= q_15;
45                elsif(seletorRefrigerante = "01") then
46                    estadoAtual <= q_coca;
47                elsif(seletorRefrigerante = "10") then
48                    estadoAtual <= q_pepsi;
49                elsif(seletorRefrigerante = "11") then
50                    estadoAtual <= q_sprite;
51                end if;
52
53              elsif(moeda = "01") then
54                -- 50 centavos inserido
55                estadoAtual <= q_10;
56              end if;
```

```
57
58            -- se for q_10
59            when q_10 =>
60              if(moeda = "01" or moeda = "10") then
61                -- 50 centavos inserido ou 1 real inserido vai para o estado final
62                if(seletorRefrigerante = "00") then
63                  estadoAtual <= q_15;
64                elsif(seletorRefrigerante = "01") then
65                    estadoAtual <= q_coca;
66                elsif(seletorRefrigerante = "10") then
67                    estadoAtual <= q_pepsi;
68                elsif(seletorRefrigerante = "11") then
69                    estadoAtual <= q_sprite;
70                end if;
71              end if;
72
```

```
            -- se for q_15
            when q_15 =>
              -- nesse estado é necessário escolher o refrigerante
                if(seletorRefrigerante = "01") then
                    estadoAtual <= q_coca;
                elsif(seletorRefrigerante = "10") then
                    estadoAtual <= q_pepsi;
                elsif(seletorRefrigerante = "11") then
                    estadoAtual <= q_sprite;
                end if;

            -- ja selecionado o refrigerante
            when others =>
              -- volta para o estado inicial
              estadoAtual <= q_00;
          end case;
        end if;
      end process;

      -- maquina de moore (independe da entrada para definir estado final)
      process(estadoAtual)
      begin
        case estadoAtual is
          -- coca = 01
          when q_coca =>
            refrigerante <= "01";

          -- pepsi = 10
          when q_pepsi =>
            refrigerante <= "10";

          -- sprite = 11
          when q_sprite =>
            refrigerante <= "11";

          -- qualquer outro estado n exibe nada
          when others =>
            refrigerante <= "00";
        end case;
      end process;

  end controle_refrigerante_architecture;
```

Entity
A entidade (controle_refrigerante) especifica as portas de entrada (Clock, Reset, moeda, seletorRefrigerante) e a porta de saída (refrigerante). As entradas incluem um sinal de clock, um sinal de reset, sinais para moedas e seletores de refrigerante. A saída é um sinal que representa o refrigerante selecionado.

Arquitetura
A arquitetura (controle_refrigerante_architecture) implementa a lógica de controle para a vending machine. O estado atual da máquina de estados é representado por um sinal chamado estadoAtual, que é do tipo estado. Os possíveis estados da máquina de estados são q_00, q_05, q_10, q_15, q_coca, q_pepsi e q_sprite.

Processo de Clock e Reset
Há um processo sensível a mudanças no sinal de Reset e no sinal de Clock. Se o sinal de Reset estiver em alto nível (1), o estado atual é resetado para q_00. Se ocorrer uma borda de

subida no sinal de Clock e o sinal de Reset estiver em nível baixo (0), o estado atual é atualizado com base nas condições especificadas no código.

Máquina de Estados
A máquina de estados é implementada usando uma estrutura de caso (case). Cada estado tem condições específicas para transição para outros estados. Por exemplo, se o estado atual for q_00, ele verifica se uma moeda de 1 real ou 50 centavos foi inserida e muda para q_10 ou q_05 respectivamente. Os outros estados seguem padrões semelhantes com base nas entradas de moeda e no seletor de refrigerante.

Processo de Saída
Há um segundo processo que é sensível ao estado atual. Este processo implementa uma máquina de Moore, onde o sinal de saída (refrigerante) é determinado apenas pelo estado atual da máquina, independente das entradas. O refrigerante é selecionado de acordo com os estados q_coca, q_pepsi e q_sprite. Em qualquer outro estado, a saída é configurada como "00".

Link do repositório do GitHub: https://github.com/Lugui14/refrigerante_vhdl

3.0 CONCLUSÃO

Ao final de todas as etapas do trabalho, foi possível obter o diagrama de estados e o código VHDL do projeto, conforme o esperado. Algumas dificuldades foram encontradas durante a realização da atividade, porém ao longo do desenvolvimento desta, foi possível aprimorar o conhecimento sobre o conteúdo. Sendo assim, percebe-se a necessidade da realização desse projeto para a compreensão da CCR e dos tópicos estudados em sala.