



**UNIVERSIDADE FEDERAL DA FRONTEIRA
SUL**

Marco Aurélio Lunardi

Matrícula: 2221101054

Email: marcolunardi10@gmail.com

Disciplina: Sistemas Digitais

Controle de máquina de refrigerante

Introdução

Este trabalho tem como objetivo projetar uma máquina de estados finitos capaz de controlar uma máquina de refrigerante, a qual aceita moedas de 50 centavos e 1 real, e libera a lata de refrigerante ao atingir o valor de R\$1,50. Foram utilizados diagramas e tabelas para mostrar os estados e as transições possíveis, além do sistema descrito em VHDL e simulado utilizando o digital.

Solução

Primeiramente, foi pensado em quantos estados seriam necessários para representar a máquina de estados, a quantidade de entradas e a quantidade de saídas. Nesse caso, 4 estados, uma entrada e uma saída são suficientes para a resolução do problema:

Entrada: w . Se $w = 0$, então foi inserido 50 centavos na máquina. Se $w = 1$, então foi inserido 1 real na máquina.

Saída: z . Quando for 1, significa que atingiu o valor necessário e libera a lata de refrigerante.

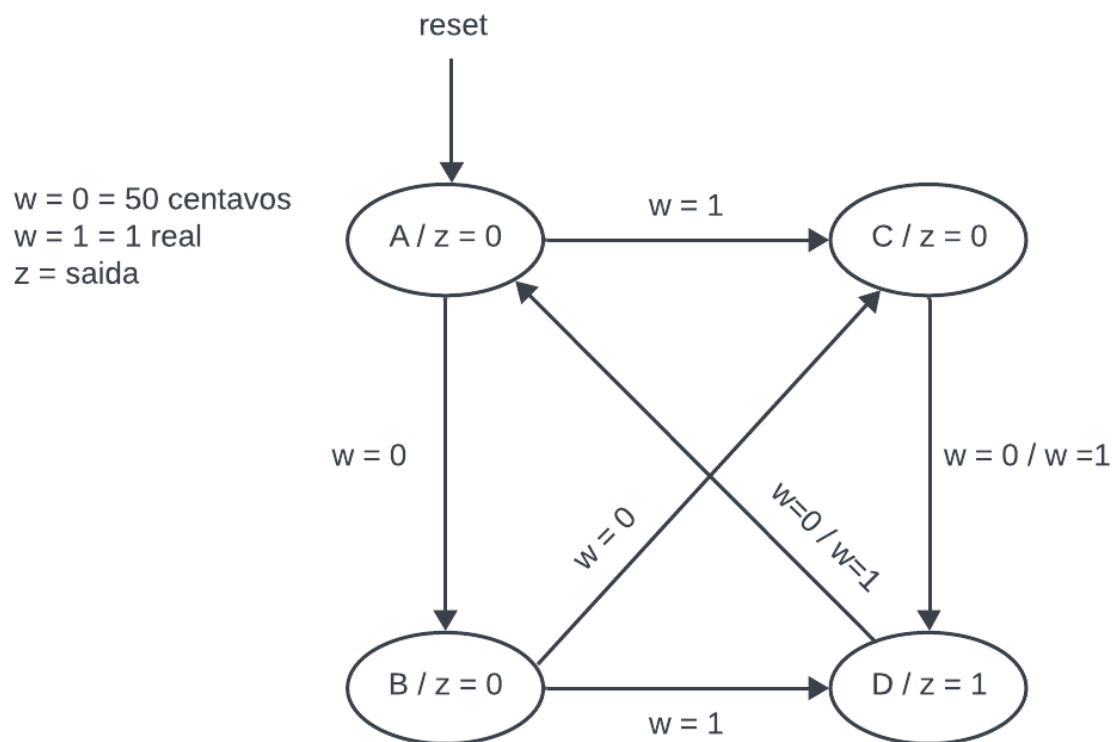
Estado A(00): Estado inicial, nenhuma moeda foi inserida, ou seja, o total é igual a 0. Pode ser acessado por um reset assíncrono.

Estado B(01): É acessado quando o total de valor inserido é 50 centavos, ou seja, quando a máquina está no estado A e $w=0$.

Estado C(10): É acessado quando o total de valor inserido é 1 real. Pode ser acessado tanto do estado A(quando $w = 1$), quanto do estado B(quando $w = 0$).

Estado D(11): É acessado quando o total de valor inserido é 1,50 reais(ou 2 reais). É o único estado onde a saída z é igual a 1, ou seja, o estado onde a lata de refrigerante é liberada. Pode ser acessada tanto do estado B(quando $w = 1$), quanto do estado C, nesse último caso independentemente de w valer 0 ou 1. Como no estado C o total de moedas é 1 real, qualquer valor inserido alcançará o valor necessário para liberar a lata de refrigerante. Caso w for igual a 1, o refrigerante acabará custando 2 reais.

Diagrama de estados:



Tabelas de estados e saídas:

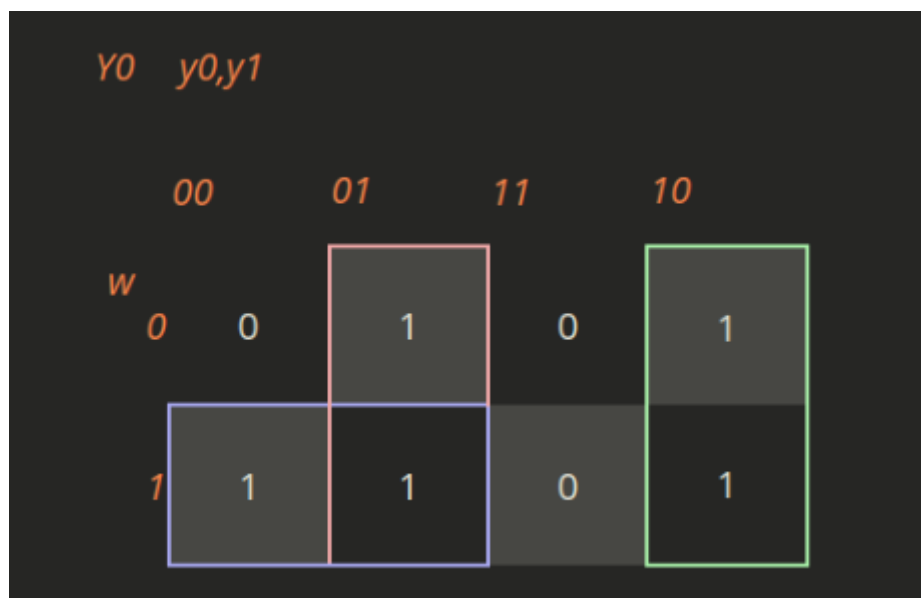
Estado atual	w	reset	z	Próximo estado
A	1	0	0	C
A	1	1	0	A
A	0	0	0	B
A	0	1	0	A
B	1	0	1	D
B	1	1	0	A
B	0	0	0	C
B	0	1	0	A
C	1	0	1	D
C	1	1	0	A
C	0	0	1	D
C	0	1	0	A
D	X	X	X	X
D	X	1	0	A
D	X	X	X	X
D	X	1	0	A

Codificando os estados:

estado A($y_0 = 0$ e $y_1 = 0$), estado B($y_0 = 0$ e $y_1 = 1$), estado C($y_0 = 1$ e $y_1 = 0$), estado D($y_0 = 1$ e $y_1 = 1$). Y_0 e Y_1 representam o próximo estado. A tabela abaixo mostra essas codificações e o próximo estado de acordo com a entrada w .

y_0y_1	w	Y_0Y_1
00	0	01
00	1	10
01	0	10
01	1	11
10	0	11
10	1	11
11	0	00
11	1	00

Para encontrar as equações que representam a lógica de próximo estado e saída, foram utilizados mapas de karnaugh com as codificações y_0y_1 e a entrada w . As imagens abaixo mostram os mapas e as equações para Y_0 , Y_1 e a saída z .



$$Y_0 = y_0'.y_1 + y_0.y_1' + w.y_0'$$

$Y1$ $y0, y1$

	00	01	11	10
w				
0	1	0	0	1
1	0	1	0	1

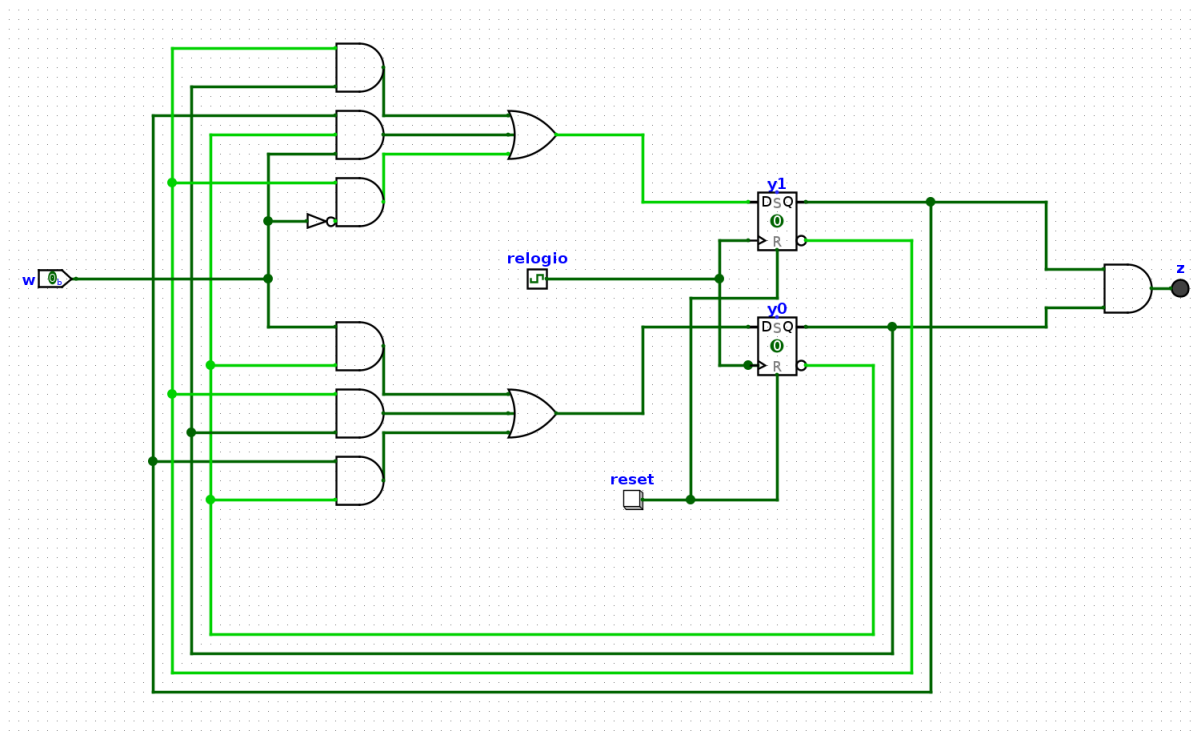
$$Y1 = w'.y1' + w.y0'.y1 + y0.y1'$$

z $y0$

	0	1
$y1$		
0	0	0
1	0	1

$$z = y1.y2$$

Para um primeiro teste de funcionamento, foi construído um circuito no logisim com as equações de lógica de próximo estado e saída, representado na imagem abaixo.



Posteriormente, o sistema digital foi descrito utilizando VHDL.

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

--w = key2
--clock = key1
--reset = key0
--z = sw0

entity maqRefri is
  port(
    key : in std_logic_vector(3 downto 0);
    ledr : out std_logic_vector(9 downto 0)
  );
end maqRefri;

architecture Behavior of maqRefri is
  type Tipo_estado is (A,B,C,D);
  signal y_atual, y_prox : Tipo_estado;
begin
  process (key(2), y_atual)
  begin
    case y_atual is
```

```

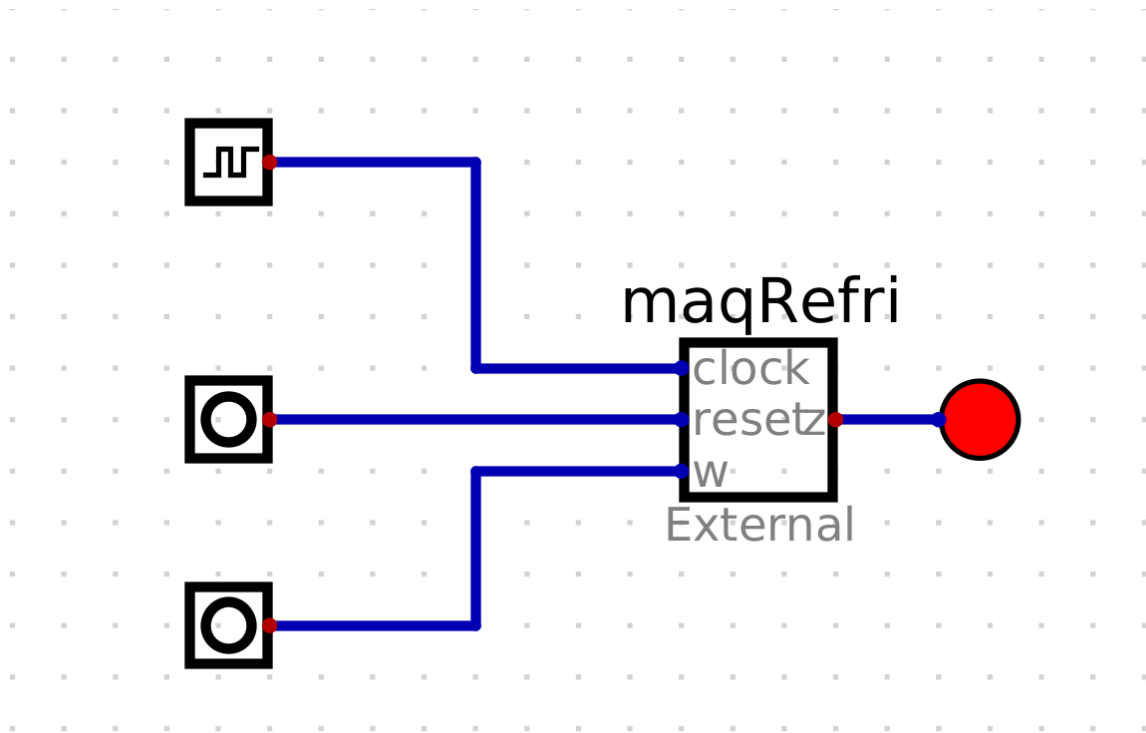
        when A =>
            if key(2) = '1' then
                y_prox <= B;
            else
                y_prox <= C;
            end if;
        when B =>
            if key(2) = '1' then
                y_prox <= C;
            else
                y_prox <= D;
            end if;
        when C =>
            if key(2) = '1' then
                y_prox <= D;
            else
                y_prox <= D;
            end if;
        when D =>
            if key(2) = '1' then
                y_prox <= A;
            else
                y_prox <= A;
            end if;
    end case;
end process;

process (key(1), key(0))
begin
    if key(0) = '0' then
        y_atual <= A;
    elsif (key(1)'event and key(1) = '0') then
        y_atual <= y_prox;
    end if;
end process;

ledr(0) <= '1' when y_atual = D else '0';
end Behavior;

```

O código foi compilado utilizando o programa Digital, gerando a máquina de estados capaz de realizar o controle de uma máquina de refrigerante, que foi posteriormente simulada também utilizando o Digital.



Conclusão

Este projeto foi de extrema importância para uma melhor fixação dos conteúdos estudados na disciplina, principalmente flip-flops e descrição do sistema em VHDL, podendo assim ter mais repertório para projetos futuros.