Ministério da Educação
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Coordenação de Engenharia Elétrica (COELT)
Campus Apucarana

Relatório do laboratório 06

Maria Eduarda Pedroso (Líder) **Matrícula: 2150336**
Ruan Mateus Trizotti **Matrícula: 2152177**

Sistemas Digitais (SICO5A)

6º Experimento: VHDL DATA 28/09/2022

Objetivos

Nessa atividade todas as simulações e análises foram feitas no simulador Quartus e também com auxílio dos conteúdos disponibilizados pelo professor e sua ajuda em aula. O intuito deste relatório é fixar e analisar códigos desenvolvidos na linguagem VHDL, sendo o código desenvolvido pela aluna e a explicação também.

Materiais e equipamentos

Os materiais utilizados foram todos códigos e funções no simulador EDA, sendo estes explicados sua utilidade no relatório.

Código e análise

O código escolhido foi um feito por mim mesma que implementa uma máquina de estados, nesse caso uma de venda de refrigerante e água em determinados estados. O código está extenso mas será anexado abaixo.

```
--importando as bibliotecas
LIBRARY IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_SIGNED.ALL;
USE IEEE.NUMERIC_STD.ALL;
```

Começamos declarando quais serão as bibliotecas importadas e onde serão utilizadas.

```
--Declarando entradas e saidas
entity Maquina is
        port(
        Switch_um1                  : IN  BIT;
        Switch_um2                  : IN  BIT;
        Switch_um3                  : IN  BIT;
        Switch_50_1                 : IN  BIT;
        Switch_50_2                 : IN  BIT;
        Switch_50_3                 : IN  BIT;
        Switch_50_4                 : IN  BIT;
        Switch_Refri                    : IN  BIT;
        Switch_Agua                     : IN  BIT;
        Reset                                   : IN  BIT;
        Letra_1                 :       OUT STD_LOGIC_VECTOR (7 DOWNTO 0);
        Letra_2                 :       OUT STD_LOGIC_VECTOR (7 DOWNTO 0);
        Letra_3                 :       OUT STD_LOGIC_VECTOR (7 DOWNTO 0);
```

```
        Num1                    :     OUT STD_LOGIC_VECTOR (7 DOWNTO 0);
        Num2                    :     OUT STD_LOGIC_VECTOR (7 DOWNTO 0);
        Num3                    :     OUT STD_LOGIC_VECTOR (7 DOWNTO 0);
        LedAgua          :  OUT STD_LOGIC;
        LedRefri         :  OUT STD_LOGIC
        );

        end Maquina;
```

Para essa segunda parte do código foi declarado uma entidade máquina com as entradas e saídas e qual seu “tipo” podemos ver que temos os switchs da placa FPGA sendo as entradas, todas em bit e letras e números como saídas sendo eles um vetor lógico, LedAgua e LedRefri são os valores no qual conseguiremos ver os estados.

```
        ARCHITECTURE Main of Maquina is
        BEGIN

                PROCESS (Switch_um1, Switch_um2, Switch_um3, Switch_50_1, Switch_50_2,
Switch_50_3, Switch_50_4, Switch_Agua, Switch_Refri)
                        VARIABLE moedas      : Integer;
                        VARIABLE aux         : integer;
                        VARIABLE compra      : integer;
                        VARIABLE Vswitch_um1  : integer;
                        VARIABLE Vswitch_um2  : integer;
                        VARIABLE Vswitch_um3  : integer;
                        VARIABLE Vswitch_50_1 : integer;
                        VARIABLE Vswitch_50_2 : integer;
                        VARIABLE Vswitch_50_3 : integer;
                        VARIABLE Vswitch_50_4 : integer;
                BEGIN

                        if (Reset = '1') then
                                moedas     :=  0;
                                aux        :=  0;
                                Vswitch_um1 :=  0;
                                Vswitch_um2 :=  0;
                                Vswitch_um3 :=  0;
                                Vswitch_50_1:=  0;
                                Vswitch_50_2:=  0;
                                Vswitch_50_3:=  0;
                                Vswitch_50_4:=  0;
                                Letra_1 <= "11001110"; --R
                                Letra_2 <= "10000110"; --E
                                Letra_3 <= "10010010"; --S
                                num1    <= "10000110"; --E
                                num2    <= "11111000"; -- -|
                                num3    <= "11001110"; -- |-
                        else
```

Na arquitetura da Maquina declaramos quais são os processos de estado do switch e quais serão as variáveis que serão utilizadas durante a execução, após isso temos um if que seta os valores padrões das variáveis toda vez que é inicializado a maquina.

```
------------------------------COLOCAR_MOEDAS---------------------------------------

                        if switch_um1 = '1' then
                                Vswitch_um1 := 10;
                        else
                                Vswitch_um1 := 0;
                        end if;

                        if switch_um2 = '1' then
                                Vswitch_um2 := 10;
                        else
                                Vswitch_um2 := 0;
                        end if;

                        if switch_um3 = '1' then
                                Vswitch_um3 := 10;
                        else
                                Vswitch_um3 := 0;
                        end if;

                        if Switch_50_1 = '1' then
                                Vswitch_50_1 := 5;
                        else
                                Vswitch_50_1 := 0;
                        end if;

                        if Switch_50_2 = '1' then
                                Vswitch_50_2 := 5;
                        else
                                Vswitch_50_2 := 0;
                        end if;

                        if Switch_50_3 = '1' then
                                Vswitch_50_3 := 5;
                        else
                                Vswitch_50_3 := 0;
                        end if;

                        if Switch_50_4 = '1' then
                                Vswitch_50_4 := 5;
                        else
                                Vswitch_50_4 := 0;
                        end if;

moedas := Vswitch_um1 + Vswitch_um2 + Vswitch_um3 + Vswitch_50_1 + Vswitch_50_2
+Vswitch_50_3 +Vswitch_50_4;
```

Basicamente aqui podemos ter a entrada dos dados para definir qual será o estado atual, cada switch específico tem um valor associado ao seu estado em alta, switch_um1 tem valor de 10 para representar um real, switch_50 tem valor de 5 representando 50 centavos, temos vários para quando estiverem em alta ir acrescentando no valor de moedas e assim sabermos qual o real valor que a maquina está recebendo.

```
-----------------------------------ESTADOS----------------------------------------

                    if (moedas < 15 and Switch_Agua = '0' and Switch_Refri= '0' ) then --1
estado
                            Letra_1  <= "10001110"; -- f
                            Letra_2  <= "10001000"; -- a
                            Letra_3  <= "11000111"; -- l
                            LedAgua  <= '0';
                            LedRefri <= '0';

                    elsif (moedas = 15 and Switch_Agua = '0' and Switch_Refri= '0') then --2
estado
                            Letra_1  <= "00001000"; -- a.
                            Letra_2  <= "11111111"; --
                            Letra_3  <= "11111111"; --
                            LedAgua  <= '1';
                            LedRefri <= '0';

                    elsif (moedas >=20 and Switch_Agua = '0' and Switch_Refri= '0') then --3
estado
                            Letra_1  <= "00001000"; -- a.
                            Letra_2  <= "10000110"; -- e
                            Letra_3  <= "01001110"; -- r.
                            LedAgua  <= '1';
                            LedRefri <= '1';
                    else
                            moedas := moedas;
                    end if;
```

Nessa parte do código está toda a prática de uma máquina de estados, como podemos ver temos 3 estados que são disparados com valores específicos:

- moedas<15 estado um (não compra nada)
- moedas = 15 estado dois (compra apenas água)
- moedas>20 estado tres (compra agua e refri)

Nas estruturas de comparação temos também a parte dos switchs, essa sendo apenas para que não apareça no display qual estado está quando os mesmo estão em alta simulando uma compra. Letra_1, Letra_2, Letra_3, LedAgua e LedRefri são displays de saída no qual passamos qual parte dele estará em alta qual em baixa formando a letra comentada na direita.

```
----------------------------------COMPRAR-----------------------------------------

                    if (moedas >= 15 and Switch_Agua = '1' and Switch_Refri = '0') then
                            Letra_1  <= "01000110"; -- c.
                            Letra_2  <= "10001000"; -- a
                            Letra_3  <= "11111111"; --
                            LedAgua  <= '0';
                            LedRefri <= '0';
                    elsif (moedas >= 20 and Switch_Refri = '1' and Switch_Agua = '0') then
                            Letra_1  <= "01000110"; -- c.
                            Letra_2  <= "11001110"; -- r
                            Letra_3  <= "11111111"; --
                            LedAgua  <= '0';
                            LedRefri <= '0';
                    end if;
```

Já nessas estruturas de comparação temos também a parte dos switchs, essa sendo apenas para que apareça no display qual estado está no momento, sendo esses a possibilidade de ser efetuada a compra da água ou do refri.

```
--------------------------------LED NUMEROS-----------------------------------------

                    num1 <= "11111111"; --
                    aux := moedas;
                     if aux < 10 then                    --unidade
                                num2 <= "01000000"; -- 0
                     elsif aux >= 10 and aux < 20 then
                                num2 <= "01111001"; -- 1
                                aux:= moedas - 10;
                    elsif aux >= 20 and aux < 30 then
                                num2 <= "00100100"; -- 2
                                aux:= moedas - 20;
                    elsif aux >= 30 and aux < 40 then
                                num2 <= "00110000"; -- 3
                                aux:= moedas - 30;
                    elsif aux >= 40 and aux < 50 then
                                num2 <= "00011001"; -- 4
                                aux:= moedas - 40;
                    elsif aux >= 50 and aux < 60 then
                                num2 <= "00010010"; -- 5
                                aux:= moedas - 50;
                    end if;

                            case aux is                     --centavos
                                when  0 =>
                                        num3 <= "11000000"; -- 0
                                when 5 =>
                                        num3 <= "10010010"; -- 5
                                when others =>
```

```
                        end case;

    end if;

            end PROCESS;
    end main;
```

Essa parte do código é simplesmente para mostrar qual o valor que está na maquina, quando temos 1 real, 1,50 e afins, também podemos notar que o processo todo acaba nesse trecho de código, sendo as outras partes que virão apenas comentários para ajudar no entendimento das saídas e códigos para as mesmas.

```
--0 = 11000000
--1 = 11111001
--2 = 10100100
--3 = 10110000
--4 = 10011001
--5 = 10010010
--6 = 10000010
--7 = 11111000
--8 = 10000000
--9 = 10011000

--A = 10001000
--E = 10000110
--R = 11001110
--F = 10001110
--L = 11000111

-- Hex0 == Numero3
-- Hex1 == numero2
-- Hex2 == numero1
-- Hex3 == Letra 3
-- Hex4 == Letra 2
-- Hex5 == Letra 1
```

Explicando esses comentários temos o que cada vetor binário significa, quando temos 11000000 essa representação no display faz com que o número um apareça e assim sucessivamente, Hex0 em diante é para controle de qual display é qual.

Maria Eduarda Pedroso

Resultados e Conclusão

Com essa prática, aprendemos como criar código em VHDL e como a linguagem funciona ao contrário de linguagens de programação como C ou assembly onde os comandos/instruções são executados na ordem em que aparecem no código-fonte, em VHDL a avaliação de

expressões e atribuição de valores ocorrem em paralelo, com efeito semelhante a esse de sinais em um circuito.

Muitos dos tópicos desta prática estão relacionados ao aprendizado teórico ministrado pelo professor em sala de aula, pois não há cálculo e sim análise, acredito que essa atividade seja crucial para melhor compreensão do tema, pois é muito raro aprender um idioma como este.

Resultados e Conclusão

Ruan Mateus Trizotti

Entender como o código VHDL é criado nos ajuda e também nos beneficia em qualquer empreendimento futuro.

VHDL é uma tradução próxima para C, Pascal ou outra linguagem assembly. Ele usa atribuição sequencial de valores e cálculos que expressam dados. Isso dá ao VHDL uma sensação intrinsecamente elétrica semelhante à corrente que percorre um circuito. VHDL é uma linguagem imperativa - mas não como C, que também é chamada de imperativa. Emular sinais elétricos com VHDL requer entender a diferença entre linguagens imperativas e declarativas.