SISTEMAS DIGITAIS INTEGRADOS

PRIMOS - VHDL
Maior número primo de 8 bits

Eric do Vale de Castro - 15/0124236

Gabriel Guimarães Almeida de Castro - 15/0126425

Brasília, Novembro de 2018

ENGENHARIA
MECATRÔNICA
UNIVERSIDADE DE BRASÍLIA

Descrição

Este é um trabalho da matéria Sistemas Digitais Integrados e quem o pediu foi o professor Carlos Humberto Llanos Quintero. Segue abaixo o enunciado:

1. **Exercício**

   Implementar em VHDL o seguintes exercício: Escreva a função maior_primo que recebe um número inteiro maior ou igual a 2 como parâmetro e devolve o maior número primo menor ou igual ao número passado à função (vide apêndice A).

2. **Orientações para desenvolvimento do projeto**

   Seguir as seguintes especificações:

   (a) Todas as implementações devem ser feitas em vhd (sem usar diagrama esquemático).

   (b) O project navigator → Files só deve conter arquivos *.vhd na mesma pasta raiz.

   (c) O resultado da síntese do projeto não deve conter latch(es) ou warning(s) de implementação.

   (d) Gerar o test bench do projeto e verificar o correto funcionamento para vários números (Obrigatório).

   (e) Fazer o relatório (formato IEEE) contendo : a descrição sucinta do trabalho, resultados da simulação (imagens) e diagrama estrutural (RTL).

   (f) Para a avaliação, zipar o projeto e o relatório numa pasta nomeada com seu código de estudante.

Projeto

Código

Esta é a primeira parte do código onde só incluímos as bibliotecas e a entidade. São dois STD_LOGIC_VECTOR's.

```
library IEEE;
use IEEE.std_logic_1164.all;
USE ieee.numeric_std.ALL;

----------------------------------------------------
entity prime is
        port ( N: in STD_LOGIC_VECTOR (7 downto 0);
               Y: out STD_LOGIC_VECTOR (7 downto 0));
end prime;
----------------------------------------------------
```

Nesta próxima parte temos a arquitetura onde contém uma função que verifica se o número é primo. Se sim retorna 1, se não retorna 0.

Esta função recebe apenas números ímpares e por isso começa checar no 3. Essas otimizações foram feitas como o de verificar apenas números ímpares fez com que o número de componentes diminuísse significativamente. E o *loop* só checa até o 16 porque se o número não for primo, ele vai ter um divisor até sua raiz, e o maior número é 255, e 256 tem raiz quadrada igual a 16. Para melhor informação sobre o método olhar *trial division*. [1]

```
architecture prime_arch of prime is

-----Function to check the if the number is prime -----
function check_prime (constant N_int: in integer) return STD_LOGIC is
        variable prime: boolean;
        variable y: STD_LOGIC; -- output
        begin
                prime := true;
                -- 3 because it checks only even numbers and  --
```

```
                -- 16 because the input have 8 bits, sqrt(256)--
                -- is 16. See Trial division method           --
                for i in 3 to 16 loop
                        if N_int mod i = 0 and N_int /= i then
                                prime := false;
                        end if;
                end loop;
                if prime then y := '1';
                else y := '0';
                end if;
return y;
end check_prime;

begin
-------------------------------------------------------
        process(N)
                variable N_int: integer;
                variable bigger: integer;
                begin
                        N_int := to_integer(unsigned(N));
                        if N_int=0 or N_int=1 then N_int := 2;
                        elsif (N_int > 2) then
                                -- It forces to check only even numbers
                                if(N_int mod 2 = 0) then N_int := N_int - 1;
                                end if;
                                L1: for j in 1 to 7 loop
                                                if (check_prime(N_int) = '1') then
                                                        exit L1;
                                                else N_int := N_int - 2; --even numbers
                                                end if;
                                end loop L1;
                        end if;
                        Y <= std_logic_vector(to_unsigned(N_int, Y'length));
        end process;
-------------------------------------------------------
end prime_arch;
```

O código realizado para o TestBanch foi:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
```

```
ENTITY prime_tb IS


END prime_tb;


ARCHITECTURE prime_arch OF prime_tb IS

        -- constants
        -- signals
        SIGNAL N : STD_LOGIC_VECTOR(7 DOWNTO 0) := (others => '0');
        SIGNAL Y : STD_LOGIC_VECTOR(7 DOWNTO 0) := (others => '0');


        COMPONENT prime
                 PORT (
                N : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
                Y : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)
                );

        END COMPONENT;


BEGIN

        DUT : prime
        PORT MAP (
-- list connections between master ports and signals
        N => N,
        Y => Y
        );

        -- Entradas adicionadas para fazer a simulação.
        N <= "00000000", "01100100" after 100ns,
                                                "00111100" after 200ns,
                                                "01011100" after 300ns,
                                                "01101100" after 400ns,
                                                "01110011" after 500ns,
                                                "01110111" after 600ns,
                                                "01111111" after 700ns,
                                                "01001011" after 800ns,
                                                "01100010" after 900ns,
```

```
                                        "00101010" after 1000ns;


END prime_arch;
```

Resultado

Realizado todo o código em VHDL, o procedimento foi feito em TestBanch para ser realizado toda a simulação no ModelSim, que apresenta um procedimento cronológico de tempo com a execução.

Desse modo, para ter uma melhor visualização da execução do projeto, transferiu-se o código para uma FPGA, colocando os Switch como entrada e os Led como saída, apresentando estes resultados:

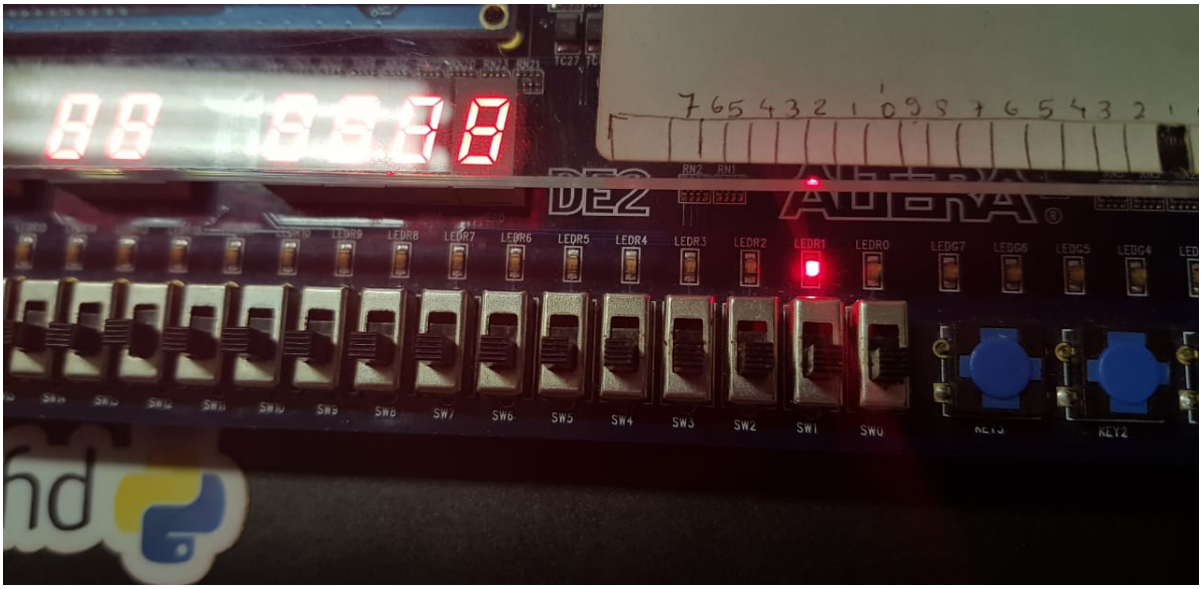

Figura 1: Primeiro primo no ponto 0

O primo defino no número 0 foi o 2, pois não existe primo nem no 1 e nem no próprio 0.

Figura 2: Primeiro primo no ponto 12

Para realizar o experimento foi colocado o valor 12 na entrada, obtendo na saída o número 11.

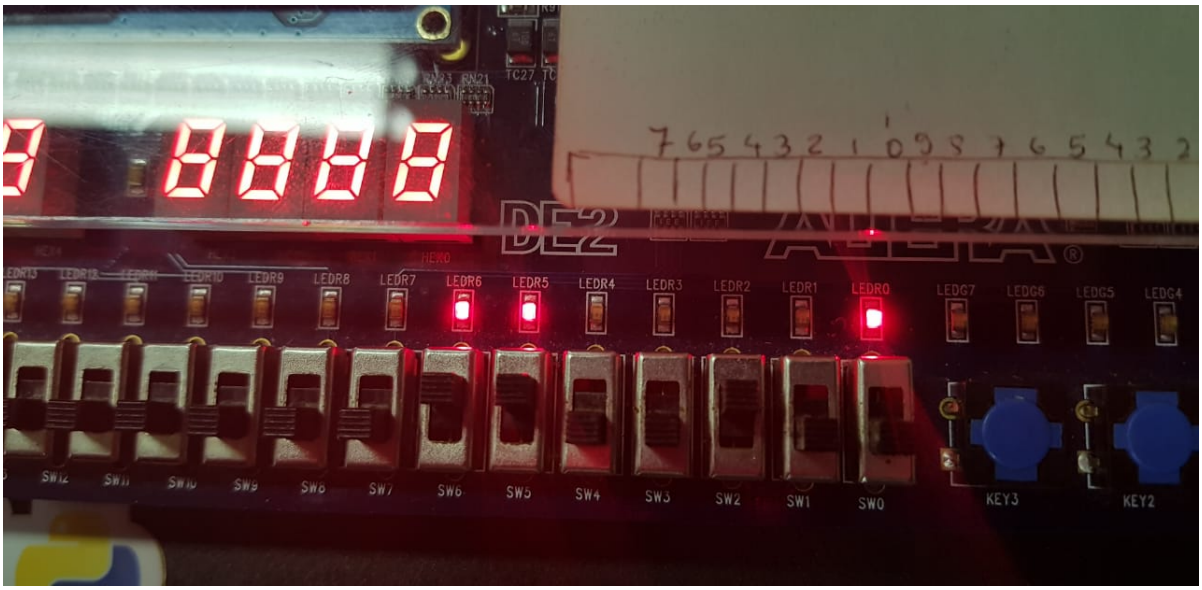

Figura 3: Primeiro primo no ponto 100

E por último a entrada foi adicionado foi o 100, recebendo por fim o número 97.

Video

Segue abaixo um link de um vídeo do projeto rodando na FPGA: `https://youtu.be/rMd0rsXgF7E`

REFERÊNCIAS BIBLIOGRÁFICAS

[1] TRIAL division. Wikimedia Foundation, Nov 2018. Disponível em: <https://en.wikipedia.org/wiki/Trial_division>.

[2] TUTORIAL - Introduction to VHDL. Disponível em: <https://www.nandland.com/vhdl/tutorials/tutorial-introductiontovhdlforbeginners.html>.