**Implementação em VHDL**

Universidade Católica de Pelotas

Guilherme Moura Baccarin

**Introdução**

Este relatório tem como objetivo analisar e descrever o processo de implementação e execução de uma calculadora de raiz quadrada e de um processador MIPS na linguagem de programação VHDL.

Projeto proposto para a avaliação da disciplina de Sistemas Digitais II, ministrada pelo professor Eduardo Antonio Cesar da Costa, do curso de Engenharia da Computação da Universidade Católica de Pelotas, durante o segundo semestre do ano de 2020.

**Sumário**

[**Introdução** 2](#_Toc53089594)

[**Bibliotecas** 4](#_Toc53089595)

[Nesse projeto foram usados os pacotes: 5](#_Toc53089596)

[**VHDL** 6](#_Toc53089597)

[Benefícios 7](#_Toc53089598)

[Desvantagens 7](#_Toc53089599)

[Estrutura de um programa VHDL 8](#_Toc53089600)

[Entity 9](#_Toc53089601)

[Architecture 9](#_Toc53089602)

[**Implementação** 10](#_Toc53089603)

[**Resultados** 12](#_Toc53089604)

[Compilação 12](#_Toc53089605)

[Simulação 13](#_Toc53089606)

**Bibliotecas**

Para o desenvolvimento de algoritmos mais complexos em quase qualquer linguagem de programação é necessário o uso de bibliotecas.

Biblioteca é uma coleção de [subprogramas](https://pt.wikipedia.org/wiki/Subprograma) utilizados no desenvolvimento de [software](https://pt.wikipedia.org/wiki/Software). Bibliotecas contém código e dados auxiliares, que provém serviços a programas independentes, o que permite o compartilhamento e a alteração de código e dados de forma [modular](https://pt.wikipedia.org/wiki/M%C3%B3dulo_(programa%C3%A7%C3%A3o)).

Para o desenvolvimento desse projeto foi utilizada a biblioteca IEEE.

A organização **IEEE** (Instituto de Engenheiros Elétricos e Eletrônicos) foi formada em 1963 e seus sócios são engenheiros, cientistas da computação, profissionais de telecomunicações etc. O **IEEE** é responsável pela criação de normas e padrões, e o comitê 802 é responsável pela parte de redes.

O IEEE criou a biblioteca IEEE VHDL. Isso foi estendido pela Synopsys; suas extensões são redistribuíveis gratuitamente.

## Nesse projeto foram usados os pacotes:

**ieee.std\_logic\_1164 >>** O pacote Std\_Logic\_1164 contém definições de tipos, subtipos e funções, que estendem o VHDL em uma lógica de vários valores. Não faz parte do padrão VHDL, mas é um padrão separado do mesmo corpo de padronização.

**numeric\_std >>** Ele fornece funções aritméticas para vetores. Substituições de std\_logic\_vector são definidas para aritmética assinada e não assinada. Ele define tipos numéricos e funções aritméticas para uso com ferramentas de síntese.

**VHDL**

Originou como linguagem para descrever hardware, no contexto do programa americano “Very High Speed Integrated Circuits” (VHSIC), iniciado em 1980.

* VHDL -> VHSIC Hardware Description Language
* Padrão IEEE em 1987 (Institute of Electrical and Electronics Engineers), revisado em 1993.
* Linguagem utilizada mundialmente por empresas de CAD para especificação, simulação, síntese, propriedade intelectual.
* Outras linguagens de descrição de hardware – SystemC, VERILOG, Handel-C, SDL, ISP, Esterel.

## Benefícios

* Projetos independentes da tecnologia (implementação física é postergada).
* Ferramentas de CAD compatíveis entre si.
* Flexibilidade: reutilização, escolha de ferramentas e fornecedores.
* Facilidade de atualização dos projetos.
* Exploração em alto nível de abstração permite analisar diferentes alternativas de implementação.
* Verificação do comportamento do sistema digital através de simulação.
* Redução do tempo de projeto > reduz “time-to-market” > redução de custo.
* Linguagem fortemente tipada > elimina erros de baixo nível.

## Desvantagens

* Hardware gerado é menos otimizado.
* Controlabilidade/Observabilidade de projeto reduzidas.
* Falta de pessoal treinado para lidar com a linguagem.

## Estrutura de um programa VHDL

* Cada módulo tem sua própria “entity” e “architecture”
* As arquiteturas podem ser descritas tanto no nível comportamental quanto estrutural ou uma mistura
* Toda a comunicação ocorre através das portas declaradas em cada entity, observando-se o tipo, tamanho e a direção
* Várias funções e tipos básicos são armazenados em bibliotecas (library). A biblioteca “IEEE” sempre é incluída
* Praticamente todos os módulos escritos em VHDL iniciam com:
  + - **library ieee;**
    - **use ieee.std\_logic\_1164.all;**

## Entity

Todo componente VHDL tem que ser definido como uma entidade (entity), o que nada mais é do que uma representação formal de uma simples porta lógica até um sistema lógico completo. Na declaração de uma entidade, descreve-se o conjunto de entradas e saídas que constituem o projeto.

## Architecture

A declaração da arquitetura (“architecture”) descreve o comportamento da entidade, define o seu funcionamento interno, isto é, como as entradas e saídas influem no funcionamento e como se relacionam com outros sinais internos. A declaração de uma arquitetura pode conter comandos concorrentes ou sequenciais. Sua organização pode conter declaração de sinais, constante, componentes, operadores lógicos etc, assim como comandos(ex: BEGIN, END). VHDL permite ter mais de uma architecture para a mesma entidade. Uma Arquitetura consiste em duas partes: a seção de declaração da arquitetura. e o corpo da arquitetura

**Implementação**

* Declaração das bibliotecas.

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

* Declaração e estrutura da entidade (entradas e saídas).

entity raiz\_quadrada is

  port (

    clk : in std\_logic;

    x : in natural range 0 to 100;

    r : out natural range 0 to 10

    );

end raiz\_quadrada;

* Declaração e implementação da lógica da arquitetura do código.

architecture arquitetura of raiz\_quadrada is

  signal z : natural range 0 to 10;

  signal x : natural range 0 to 100;

  signal y : natural range 0 to 100;

  signal t : natural range 0 to 2;

begin

  PROCESS(clk)

  begin

  t <= 1;

  x <= 2;

  y <= 4;

  z <= 1;

  if (clk = '1' OR clk = '0') then

    -- while(t > 0) loop

    if t > 0 and y < x then

      z <= z + 1;

      y <= y + x + 1;

      x <= x + 2;

    end if;

    if y < x then

      t <= 1;

    else

      t <= 0;

      r <= z;

    end if;

  end if;

  end process;

end arquitetura;

**Resultados**

## Compilação

Resultado da compilação do arquivo vhdl, mostrando quantos componentes foram usados, a quantidade de pinos que foram usados, o total de registradores que foram utilizados e uma série de outras informações sobre a compilação do arquivo

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

## Simulação

A imagem mostra o resultado da saída após a simulação do código de cálculo da raiz quadrada.

Mostrando o ciclo do relógio oscilando entre um e zero e o resultado, no caso sendo a entrada X igual a 25 e a saída R sendo 5, mostrando que o cálculo está correto.

Gráfico

Descrição gerada automaticamente