

Atividade 6 - Detector de Máximo e Mínimo

Aluno: Deivid da Silva Galvão RA: 2408740
Aluno: João Vitor Nakahodo Yoshida RA: 2419904
Professor orientador: Marcelo de Oliveira

UTFPR
Engenharia de Computação
LRCO7A - Lógica Reconfigurável

Relatório

Relatório do Trabalho Prático Disciplinar apresentado como requisito parcial à obtenção de nota na disciplina de Lógica Reconfigurável do Curso Superior de Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Aluno: Deivid da Silva Galvão RA: 2408740

Aluno: João Vitor Nakahodo Yoshida RA: 2419904

Professor orientador: Marcelo de Oliveira

Dezembro
2024

Conteúdo

1	Introdução	1
2	Conceitos novos dessa atividade	1
2.1	Package:	1
2.2	Procedure:	1
2.3	Type:	1
2.4	Generic:	1
3	Implementação	1
4	Resultados Obtidos	3

1 Introdução

A atividade 6 visa desenvolver um circuito digital capaz de identificar os valores mínimos e máximos em um conjunto de dados representados como números unsigned. Utilizando a linguagem de descrição de hardware VHDL, o projeto faz uso de estruturas como package, procedure, generic e type. Com o pacote definido, foi possível criar um vetor de números unsigned com flexibilidade tanto no número de bits quanto na quantidade de valores. Uma procedure foi implementada para percorrer o vetor de dados, identificando e atualizando os valores mínimos e máximos conforme necessário.

2 Conceitos novos dessa atividade

2.1 Package:

Um package em VHDL é uma coleção de definições e declarações que podem ser reutilizadas em diferentes partes do código.

2.2 Procedure:

Uma procedure em VHDL é um bloco de código reutilizável projetado para executar uma tarefa específica. Ela permite a modularização do código, facilitando a manutenção e a reutilização em diferentes partes do projeto.

2.3 Type:

A palavra-chave type é utilizada para declarar um novo tipo de dado em VHDL, permitindo a criação de tipos personalizados que podem ser usados para melhorar a clareza e a modularidade do código.

2.4 Generic:

A palavra-chave generic é utilizada para definir parâmetros que podem ser ajustados ou personalizados durante a instanciação de uma entidade. Isso permite maior flexibilidade e adaptabilidade no design do código, facilitando a modificação de características específicas sem a necessidade de alterar o código principal.

3 Implementação

Codigo VHDL

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
use work.maxmin_pkg.ALL;

entity prog6 is
  generic (
    Bits : positive := 4;
```

```

        Inputs : positive := 4
    );
    port (
        entrada1, entrada2, entrada3, entrada4 : in ...
        ...unsigned(Bits - 1 downto 0);
        min_value, max_value : out unsigned(Bits - 1 downto 0)
    );
end entity prog6;

architecture prog6 of prog6 is
    signal values : unsigned_vector(0 to Inputs - 1);
begin
    values <= (entrada1, entrada2, entrada3, entrada4);
    process (values)
    begin
        encontrar_min_max(values, Inputs, min_value, max_value);
    end process;
end architecture prog6;

```

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERICSTD.ALL;

package maxmin_pkg is
    subtype const_unsigned_vector is unsigned(3 downto 0);

    type unsigned_vector is array (natural range <>) of ...
    ...const_unsigned_vector;

    procedure encontrar_min_max(
        values : in unsigned_vector;
        Inputs : in positive;
        signal min_value, max_value : out const_unsigned_vector
    );
end package maxmin_pkg;

package body maxmin_pkg is
    procedure encontrar_min_max(
        values : in unsigned_vector;
        Inputs : in positive;
        signal min_value, max_value : out const_unsigned_vector
    ) is
        variable temp_min, temp_max : const_unsigned_vector;
    begin
        temp_min := values(0);
        temp_max := values(0);
        for i in 0 to Inputs - 1 loop
            if values(i) < temp_min then

```

```

        temp_min := values(i);
    end if;
    if values(i) > temp_max then
        temp_max := values(i);
    end if;
end loop;
min_value <= temp_min;
max_value <= temp_max;
end procedure;
end package body maxmin_pkg;

```

Entity prog6: Define um componente com entradas (entrada1, entrada2, entrada3, entrada4) e saídas (min value, max value). As entradas e saídas são do tipo unsigned, que representa números binários sem sinal.

Architecture prog6: Declara um vetor de sinais (values) que armazena as entradas. Dentro de um processo, a função encontrar min max é chamada para calcular os valores mínimo e máximo das entradas, que são atribuídos às saídas (min value, max value).

Package maxmin pkg: Define tipos e subtipos usados na arquitetura. Contém a declaração da procedure encontrar minimo e maximo, que buscar o valor mínimo e máximo de um vetor de unsigned.

Package Body maxmin pkg: Implementa a procedure encontrar os valores mínimos e máximos. A procedure inicializa os valores mínimo e máximo com o primeiro elemento do vetor, depois itera sobre todos os elementos do vetor para atualizar os valores mínimo e máximo conforme necessário. No final, atribui esses valores às saídas

4 Resultados Obtidos

Diagrama RTL gerado pelo Quartus Prime de acordo com o código VHDL.

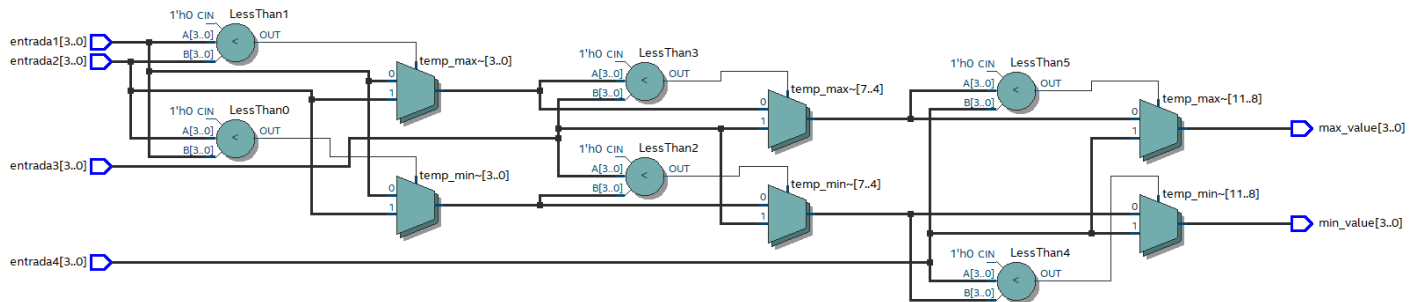


Figura 1: Diagrama RTL

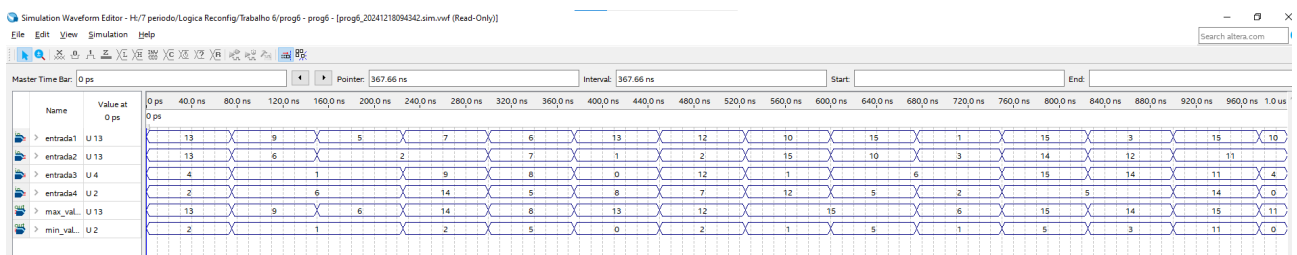


Figura 2: Simulação