

Universidade Tecnológica Federal do
Paraná
Engenharia da Computação
Lógica Reconfigurável

**Relatório 6 - Detector de máximo e
mínimo**

Aluno: Eduardo Yuji Yoshida Yamada
Professor orientador: Marcelo de Oliveira

Dezembro
2024

Conteúdo

| | | |
|---|--------------|---|
| 1 | Introdução | 1 |
| 2 | Códigos | 2 |
| 3 | Waveform | 5 |
| 4 | Diagrama RTL | 6 |

1 Introdução

O objetivo deste projeto é implementar um detector de valor mínimo e valor máximo para uma quantidade genérica de valores do tipo unsigned, onde tanto a quantidade de valores quanto a largura de cada valor podem ser definidos de forma genérica. Para isso, foram utilizadas as estruturas de package, procedure, generic e type da linguagem VHDL, permitindo a criação de um código reutilizável e flexível. O projeto visa simular um sistema capaz de identificar o valor mínimo e máximo de uma lista de valores, onde a quantidade de entradas e o número de bits de cada valor podem ser ajustados conforme a necessidade, possibilitando a aplicação do mesmo código em diferentes cenários e sistemas.

Na implementação, foi criado um tipo de dado personalizado no package, que representa um vetor de valores unsigned, e uma procedure responsável por calcular o valor mínimo e máximo dentro deste vetor. A arquitetura da entidade principal foi projetada para utilizar esse tipo de dado e permitir a simulação e verificação do funcionamento do sistema. A flexibilidade do código foi alcançada através do uso de parâmetros genéricos, possibilitando a definição do número de entradas e a largura dos valores no momento da instância da entidade, tornando o design mais modular e adaptável a diferentes requisitos.

2 Códigos

Dessa forma, dois códigos foram desenvolvidos, um deles é o package e o outro o código principal.

Código principal do projeto:

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

entity projeto6 is
    generic (
        num_bits : integer := 4;  -- Quantidade de bits
        num_inputs : integer := 4
    );
    port (
        entrada1, entrada2, entrada3, entrada4 : in
            unsigned(num_bits-1 downto 0);
        min_val, max_val : out unsigned(num_bits-1
            downto 0)
    );
end projeto6;

architecture projeto6 of projeto6 is
    --signal x : unsignedArray (0 to num_inputs -1) (
        num_bits -1 downto 0) ;
    signal x : unsignedArray := (others => (others =>
        '0'));
begin
    process(entrada1, entrada2, entrada3, entrada4)
    begin
        -- Chama a procedure para calcular os valores
        minimo e maximo
        x <= (entrada1, entrada2, entrada3, entrada4);
        findMinMax(x, min_val, max_val);
    end process;
end projeto6;
```

Código do Package:

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

```

package MinMaxPackage is

    -- Tipo de dados para um array de 4 elementos do
    -- tipo unsigned
    type unsignedArray is array (0 to 3) of unsigned(3
        downto 0); -- Define o array com 4 elementos do
        -- tipo 'unsigned' de 4 bits

    -- Procedure para encontrar o valor minimo e maximo
    procedure findMinMax(
        signal x : in unsignedArray;
        signal min_val : out unsigned;
        signal max_val : out unsigned
    );
end MinMaxPackage;

package body MinMaxPackage is

    -- Definicao da procedure para encontrar o valor
    -- minimo e maximo
    procedure findMinMax(
        signal x : in unsignedArray;
        signal min_val : out unsigned;
        signal max_val : out unsigned
    ) is
        -- Agora o array tem tamanho fixo de 4 e usa
        -- unsigned de 4 bits
        variable temp_min : unsigned(3 downto 0) := (
            others => '1');
        variable temp_max : unsigned(3 downto 0) := (
            others => '0');
    begin

        -- Encontrar os valores minimo e maximo
        for i in x'range loop
            if x(i) < temp_min then
                temp_min := x(i);
            end if;
            if x(i) > temp_max then
                temp_max := x(i);
            end if;
        end loop;
    end;
end;

```

```
        -- Atribuir os valores minimos e maximos
        min_val <= temp_min;
        max_val <= temp_max;
    end procedure findMinMax;

end MinMaxPackage;
```

3 Waveform

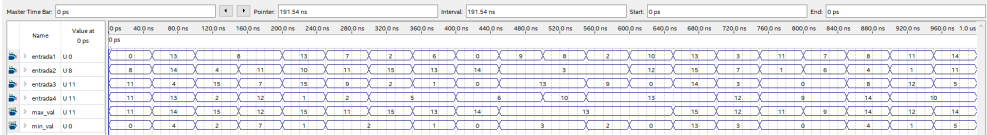
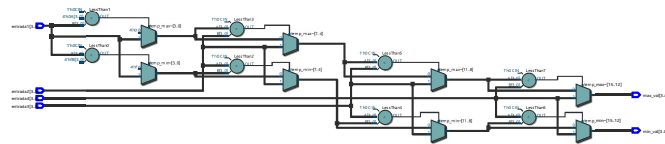


Figura 1: Waveform

4 Diagrama RTL

Date: December 13, 2024

Project: projeto6



Page 1 of 1

Revision: projeto6

Figura 2: RTL Viewer