

Inversor de Palavras

Laboratório 2

2020.1



PUC
RIO

Fernando Homem da Costa - 1211971

Departamento de Engenharia Elétrica
Pontifícia Universidade Católica do Rio de Janeiro
ENG1448 - Computação Digital
Professor: Wouter Caarls
14 de Setembro de 2020

Lista de Figuras

1	Criando a Entidade Inversor	4
2	Waveform Graph	7
3	Esquemático Porta Lógica	7
4	Esquemático RTL	8



CONTEÚDO

Conteúdo

1	Introdução	3
2	Desenvolvimento	4
2.1	Exercício 1:	4
2.2	Exercício 2:	5
2.3	Exercício 3:	5
2.4	Exercício 4:	6
2.5	Exercício 5:	7



1 Introdução

Este projeto é a segunda tarefa da disciplina de Computação Digital (ENG1448) do primeiro semestre de 2020, ministrada pelo professor Wouter Caarls.

Nesta tarefa, deve-se desenhar e testar uma entidade em VHDL para inverter (ou não) a ordem de um vetor de 4 bits.



2 Desenvolvimento

2.1 Exercício 1:

Cria um novo projeto e um novo módulo de VHDL em ISE. A entidade dentro do módulo tem uma entrada e uma saída tipo *std_logic_vector* de 4 bits, e mais uma entrada tipo *std_logic* (o seletor).

```
-----  
-- Name: Fernando Homem da Costa  
--  
-- Create Date:      17:24:45 09/13/2020  
-- Module Name:      module_lab02 - rtl  
-- Revision:  
-- Revision 0.01 - File Created  
--  
-----  
  
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
  
entity inversor is  
    Port ( A : in std_logic_vector(3 downto 0);  
          B : in std_logic;  
          C : out std_logic_vector(3 downto 0));  
end inversor;
```

Figura 1: Criando a Entidade Inversor



2.2 Exercício 2:

Escreve o código para realizar a tarefa: inverter a ordem da primeira entrada quando o seletor está em '1', e passar sem modificações quando o seletor está em '0'.

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

entity inversor is
    Port(
        A : in std_logic_vector(3 downto 0);
        B : in std_logic;
        C : out std_logic_vector(3 downto 0));
end inversor;

architecture rtl of inversor is

begin
    C <= A(0)&A(1)&A(2)&A(3) when B = '1' else A;
end rtl;
```

2.3 Exercício 3:

Cria um novo testbench para testar o seu código. Tira todo o código relacionado ao clock do código gerado pelo ISE, e reescreve o código do stimulus process da forma que está no slide 34. O teste deve provar que o seu código está funcionando corretamente.

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY lab2_tb IS
END lab2_tb;

ARCHITECTURE behavior OF lab2_tb IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT inversor
    PORT(
        A : IN  std_logic_vector(3 downto 0);
        B : IN  std_logic;
```



2.4 Exercício 4:

```
        C : OUT  std_logic_vector(3 downto 0)
    );
END COMPONENT;

--Inputs
signal A : std_logic_vector(3 downto 0) := (others => '0');
signal B : std_logic := '0';

--Outputs
signal C : std_logic_vector(3 downto 0);

BEGIN

    -- Instantiate the Unit Under Test (UUT)
    uut: invensor PORT MAP (
        A => A,
        B => B,
        C => C
    );

    A(0) <= '0' after 10ns;
    A(1) <= '0' after 10ns;
    A(2) <= '0' after 10ns;
    A(3) <= '1' after 10ns;
    B <= '1' after 50ns;

END;
```

2.4 Exercício 4:

Roda o simulador para verificar o funcionamento do código. Cola uma captura do waveform graph no relatório.



2.5 Exercício 5:

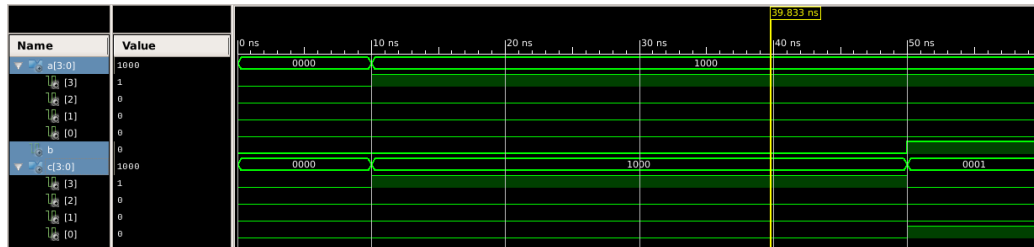


Figura 2: Waveform Graph

2.5 Exercício 5:

Sintetize o seu código e analise o esquemático RTL. Cola uma captura do esquemático no nível de portas lógicas no relatório, e explique o seu funcionamento.

Resposta: O esquemático do nível de porta lógica (Figura 3) não está de acordo com a minha solução proposta. Ainda sobre o RLT Schematic, ele é um esquema gerado após a síntese do HDL. Ele é uma representação da parte interna do circuito em função de seus componentes genéricos, como portas lógicas. É independente ao dispositivo Xilinx utilizado. Entretanto, adotarei como base para minha resposta a Figura 4. Quando o seletor (B) for igual a 1, saída deverá está na ordem inversa em relação à entrada A. Para isso, utilizaremos o operador (concatenação) para unir bit a bit na ordem inversa. Caso o seletor (B) seja igual a 0, a saída é igual a entrada A.

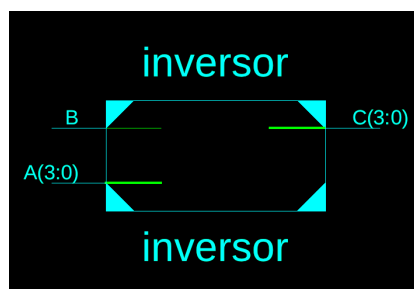


Figura 3: Esquemático Porta Lógica



2.5 Exercício 5:

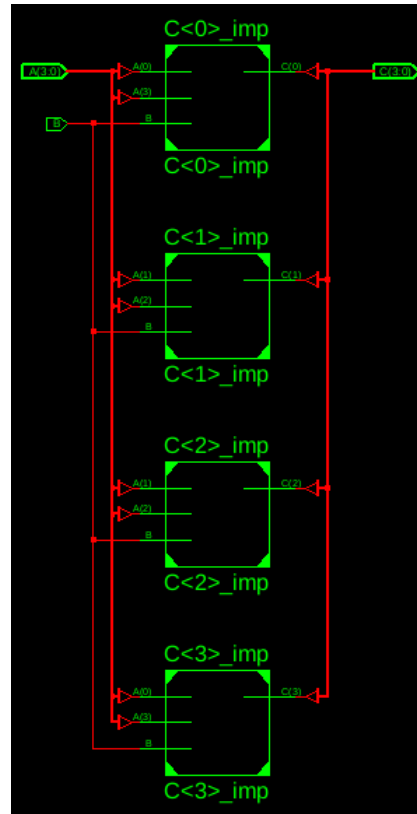


Figura 4: Esquemático RTL