

## Laboratório 1: Comparador de Bits:

Professor: Felipe Calliari | Monitor: Cristiano Nascimento

Aluno: Pedro Gabriel Serodio Sales e Thiago Levis

Matrícula: 2211911 e 1812899

## 1 Introdução

O objetivo deste relatório é apresentar as soluções para o laboratório 1 da disciplina ENG1448 assim como apresentações do laboratório em vídeo.

## 2 Resolução

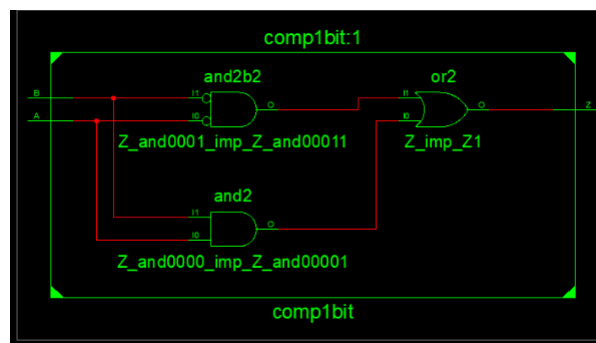
### 1 : Apresentação do código destacado e suas representações geradas (RTL e Technology)

```

1      entity complbit is
2          Port ( A : in STD_LOGIC;
3                B : in STD_LOGIC;
4                Z : out STD_LOGIC);
5      end complbit;
6
7      architecture Behavioral of complbit is
8
9      begin
10         Z<= (A AND B) OR (NOT A AND NOT B);
11      end Behavioral;
12

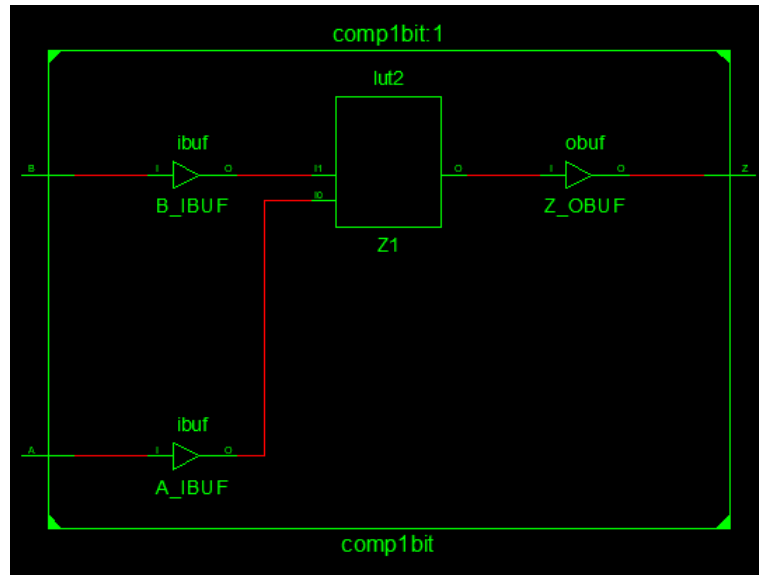
```

Código Base



(1)

Figura 1: RTL Schematic



(2)

Figura 2: Technology Schematic

#### 1.a : Verificação no Summary do número de LUTs e IOBs utilizadas e disponíveis

Device Utilization Summary (estimated values)				
Logic Utilization	Used	Available	Utilization	
Number of Slices	1	4656	0%	
Number of 4 input LUTs	1	9312	0%	
Number of bonded IOBs	3	232	1%	

(3)

Figura 3: LUTs e IOBs utilizadas e disponíveis

#### 1.b Delay do circuito pelo Detailed Report

Delay: 6.209ns (Levels of Logic = 3)

#### 1.c Análise do RTL Schematic e Technology Schematic. Investigação das diferenças

A visualização do RTL abre um arquivo NGR que pode ser visualizado como um esquema em nível de porta.

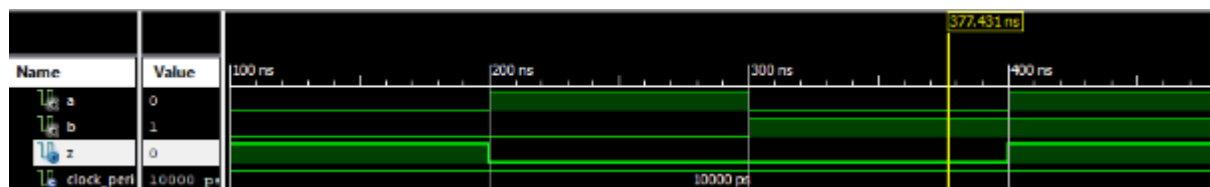
Este esquema é gerado após a fase de síntese de HDL do processo de síntese. Ele mostra uma representação do design pré-otimizado em termos de símbolos genéricos, como somadores, multiplicadores, contadores, portas AND e portas OR, que são independentes do dispositivo Xilinx alvo.

A visualização de um esquema de tecnologia abre um arquivo NGC que pode ser visualizado como um esquema específico de arquitetura.

Este esquema é gerado após a fase de otimização e direcionamento tecnológico do processo de síntese. Ele mostra uma representação do design em termos de elementos lógicos otimizados para o dispositivo ou tecnologia Xilinx alvo; por exemplo, em termos de LUTs, lógica de transporte, buffers de E/S e outros componentes específicos da tecnologia. A visualização deste esquema permite que você veja uma representação em nível de tecnologia do seu HDL otimizado para uma arquitetura Xilinx específica, o que pode ajudá-lo a descobrir problemas de design no início do processo de design.

## 2 Observação do ISim Simulator do teste solicitado

Os resultados apontados foram satisfatórios e condizem com o processo observado.



(4)

Figura 4: ISim Simulator

## 3 Trocar as atribuições dentro da arquitetura e verificar as diferenças apresentadas

Para o primeiro caso:

```

1 primeira_comp <= a AND b;
2 segunda_comp <= not(a) AND not(b);
3 z <= primeira_comp OR segunda_comp;
4

```

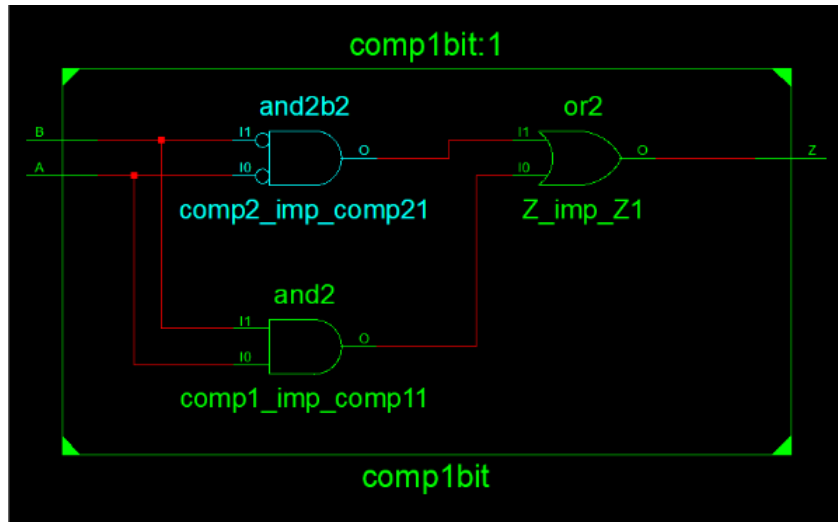
Primeiro Caso: Ordem Não Invertida

Não ocorreram modificações em todos os casos.

Device Utilization Summary (estimated values)				[...]
Logic Utilization	Used	Available	Utilization	
Number of Slices	1	4656	0%	
Number of 4 input LUTs	1	9312	0%	
Number of bonded IOBs	3	232	1%	

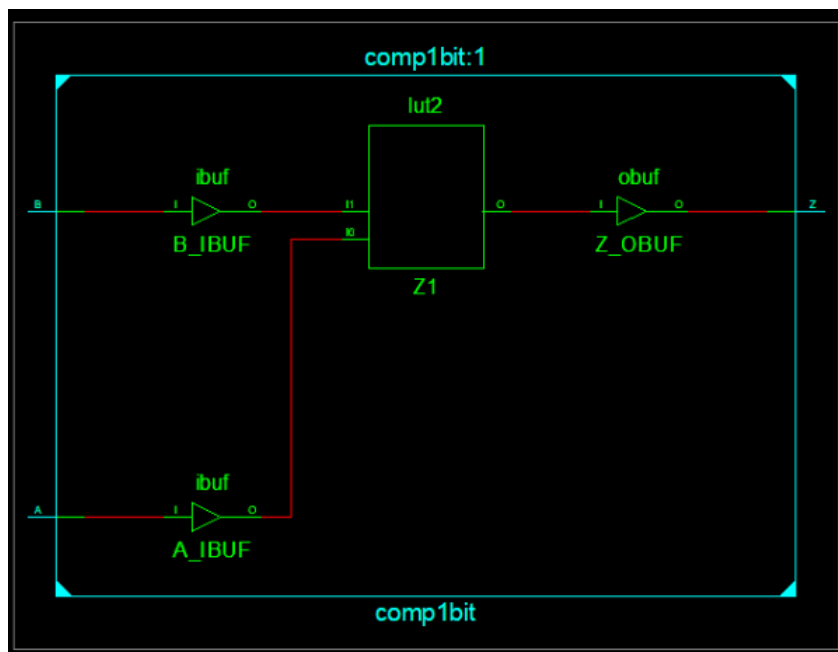
(5)

Figura 5: LUTs e IOBs: Caso [1]



(6)

Figura 6: RTL Schematic: Caso [1]



(7)

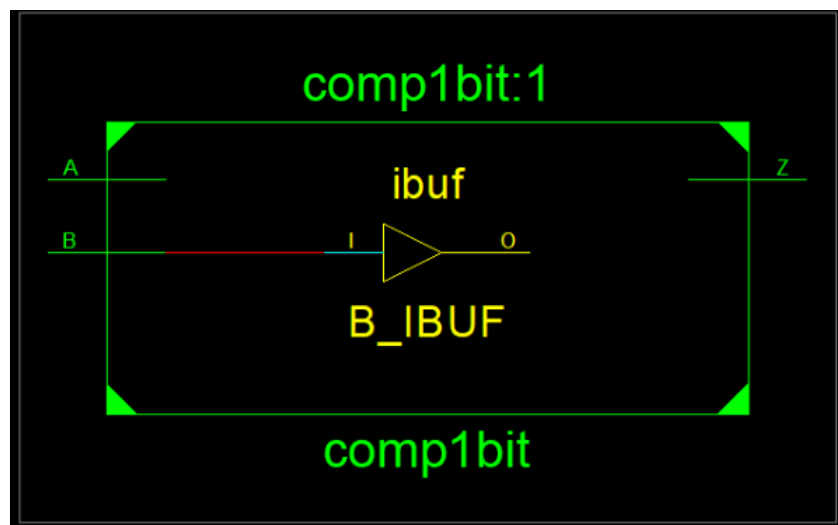
Figura 7: Technology Schematic: Caso [1]

Para o segundo caso:

```
1      z <= primeira_comp OR segunda_comp;  
2      primeira_comp <= a AND b;  
3      segunda_comp <= not(a) AND not(b);  
4
```

Segundo Caso: Ordem Invertida

No segundo caso ocorreram modificações apenas no Technology Schematic, no qual agora apenas representa um Buffer.



(8)

Figura 8: Technology Schematic: Caso [2]

#### 4 Implementação e utilização do iMPACT para configuração do kit

Implementação bem sucedida com vídeo de apresentação no final do relatório. O código final também será enviado junto ao relatório.

## 5 Iniciar um novo projeto e fazer um comparador de 2 bits

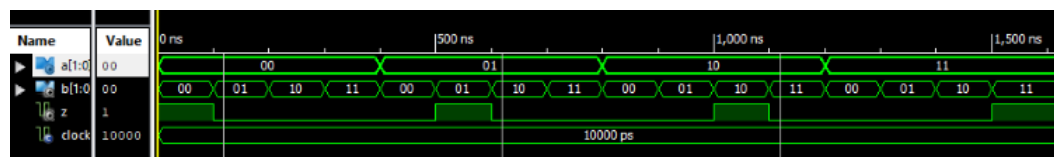
A preparação do projeto com as estruturas solicitadas e instancias solicitadas obteve resultados satisfatórios, no qual utilizamos a arquitetura abaixo:

```
1      architecture Behavioral of comp2bit is
2          signal comp : STD_LOGIC_VECTOR (1 downto 0) := (others =>
3              '0');
4          begin
5              CompBit0 : entity .work.complbit (Behavioral)
6                  port map(
7                      a => a(0) ,
8                      b => b(0) ,
9                      z => comp(0)
10                 );
11              CompBit1 : entity .work.complbit (Behavioral)
12                  port map(
13                      a => a(1) ,
14                      b => b(1) ,
15                      z => comp(1)
16                 );
17              Z <= comp(0) AND comp(1);
18          end Behavioral;
```

Projeto Two Bits: Arquitetura

## 6 Modificações e Adaptações do testbench para o comparador de 2 bits

O código do teste seguirá em anexo com o relatório e obtivemos resultados satisfatórios.



(9)

Figura 9: ISim Simulator: Projeto Two Bits

## 7 Implemente o comparador

A implementação do comparador ocorreu de forma esperada e obtivemos resultados satisfatórios. O código da implementação estará anexado junto com o envio do relatório, assim como o vídeo de apresentação.

## 3 Vídeo de Apresentação

O vídeo de apresentação segue ao lado: **Lab 1: Comparadores de Bits**

ou acessar pelo url: <https://youtu.be/dqNgnGApL44?si=LU8nBNTYVQnP01Yf>