

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA

Circuitos Digitais – INF01058

## LAB 05 – ULA

Ana Cláudia Rodrigues — 343123  
Bruno Samuel Ardenghi Gonçalves — 550452

# 1 Introdução

Neste laboratório desenvolvemos uma versão de Unidade Lógica e Aritmética (ULA), a qual possui operação aritmética de soma e operações lógicas de AND, OR e NOT. O projeto foi implementado e simulado pelo software Quartus, posteriormente sintetizado e testado em uma FPGA Cyclone III D0 (modelo EP3C16F484C6).

## 1.1 Entradas

- $A_{3-0}$ : primeiro operando
- $B_{3-0}$ : segundo operando
- $op\_sel_{1-0}$ : seletor de operação

## 1.2 Saídas

- $S_{3-0}$ : resultado da operação
- $flag\_Z$ : indicador de resultado igual a zero ( $S_n = 0$ )
- $flag\_N$ : indicador de resultado negativo ( $S_3 = 1$ )

## 1.3 Módulos e funções

- Multiplexador 4:1 de 4 bits

$$S_n = ((A_n \cdot \overline{op\_sel_n}) + (B_n \cdot op\_sel_n))$$

OBS: para a lógica de dois níveis implementada no circuito, temos:

$$S_n = (((RCA \cdot \overline{op\_sel_0}) + (AND \cdot op\_sel_0)) \cdot \overline{op\_sel_1}) \\ + (((OR \cdot \overline{op\_sel_0}) + (NOT \cdot op\_sel_0)) \cdot op\_sel_1)$$

- Ripple Carry Adder (RCA) de 4 bits

$$C_0 = A_0 \cdot B_0$$

$$S_0 = A_0 \oplus B_0$$

$$C_{n \geq 1} = \overline{\overline{((A_n \oplus B_n) \cdot C_{n-1})} \cdot \overline{(A_n \cdot B_n)}}$$

$$S_{n \geq 1} = (A_n \oplus B_n) \oplus C_{n-1}$$

- AND de 4 bits

$$S_n = A_n \cdot B_n$$

- OR de 4 bits

$$S_n = A_n + B_n$$

- NOT de 4 bits

$$S_n = \overline{A_n}$$

## 2 Resultados

Total de elementos lógicos	11
Total de pinos	16
Área total	27

Table 1: Consumo de recursos do FPGA em área

Frequencia máxima	134.77 MHz
Atraso crítico	6.920 ( $B_1 \rightarrow S_2$ )

Table 2: Frequência máxima de operação e atraso crítico (Slow, 85°C)

## 3 Capturas de tela

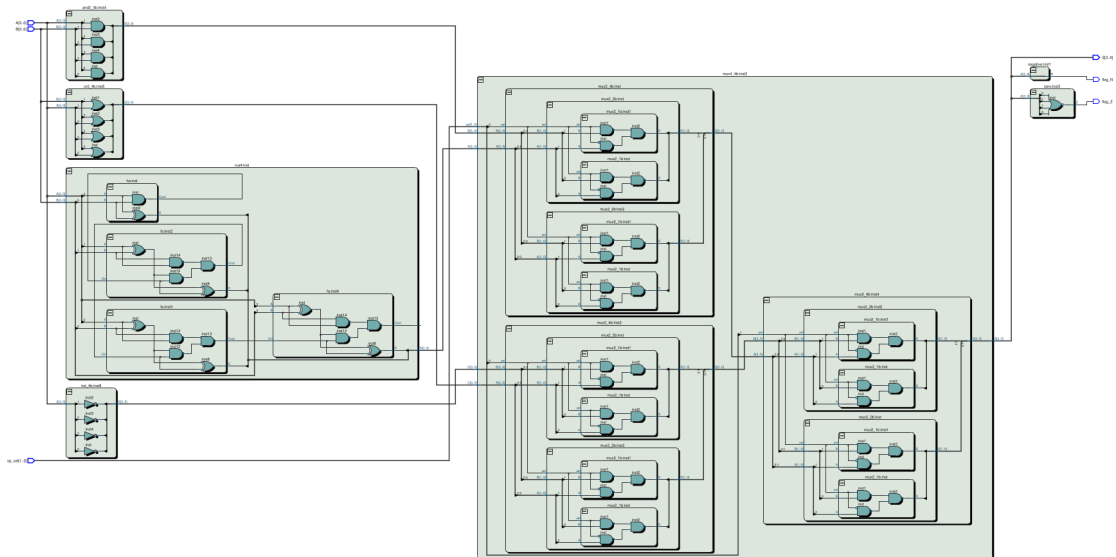


Figure 1: Captura de tela do netlist

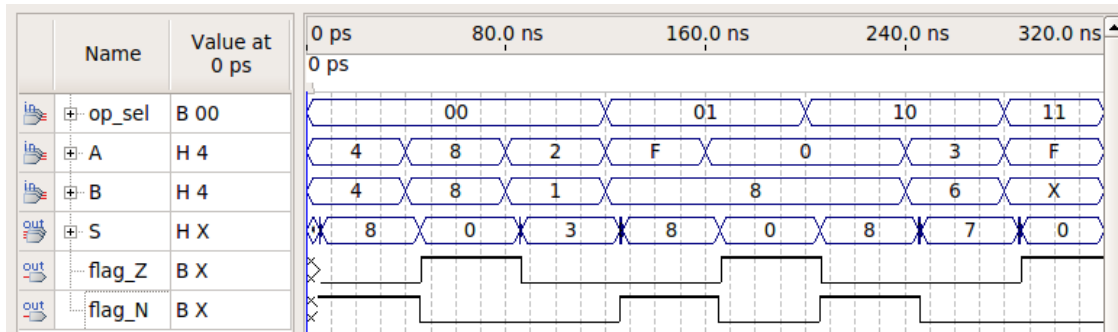


Figure 2: Captura de tela da simulação em forma de onda com atraso

## 4 Arquivo de restrições

```
create_clock -name clk -period 10
set_clock_uncertainty -from clk 0.1

set_input_delay -clock clk -max 0.2 [all_inputs]
set_input_delay -clock clk -min 0.01 [all_inputs]
set_output_delay -clock clk -max 0.2 [all_outputs]
set_output_delay -clock clk -min 0.01 [all_outputs]
```

Listing 1: constraints.sdc