

# Experimento 6

## Circuitos Combinacionais: Multiplexadores

### Grupo D1

Alexandre Augusto, 15/0056940

Anderson Vieira, 19/0102322

Gabriel de Castro, 21/1055432

<sup>1</sup>Dep. Ciência da Computação – Universidade de Brasília (UnB)

CIC0231 - Laboratório de Circuitos Lógicos

12 de março de 2023

aluno1@gmail.com, aluno2@hotmail.com, aluno3@aluno.unb.br

**Abstract.** *This report will present an implementation of combinational circuits that make use of multiplexers, the project will be carried out in Quartus-II software version 13.0 SP1.*

**Resumo.** *Este relatório apresentará a implementação de circuitos combinacionais que fazem uso de multiplexadores, o projeto será realizado no software Quartus-II versão 13.0 SP1.*

## 1. Introdução

Através do software Quartus-II, os alunos implementarão somadores apenas com multiplexadores e decodificadores. Cada passo será detalhado em Procedimentos e Resultados.

### 1.1. Objetivos

Tivemos por finalidade a implementação de somadores e outras funções utilizando multiplexadores e decodificadores. Foi feito o uso do software Quartus - II para montagem do sistema virtualmente e FPGA DE2 para o sistema real.

### 1.2. Materiais

Neste experimento foram utilizados os seguintes materiais e equipamentos:

- Software Quartus-II versão 13.0 SP1;
- Kit de desenvolvimento em FPGA DE2 ou DE2-70 Intel.

## 2. Procedimentos e Resultados

Os procedimentos consistem em criar um somador de diversas formas, para tanto, foram obtidos certos resultados em comum. A tabela verdade é a mesma para todas as questões.

**Tabela 1. Tabela Verdade do somador**

Entradas			Saídas	
A	B	Cin	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Além disso, a função que representa Cout e S seguem abaixo.

$$Cout = f(A, B, Cin) = \bar{A}BCin + A\bar{B}Cin + ABC\bar{in} + ABCin \quad (1)$$

$$S = g(A, B, Cin) = \bar{A}\bar{B}Cin + \bar{A}B\bar{C}in + A\bar{B}\bar{C}in + ABCin \quad (2)$$

Serão descritos os procedimentos e resultados obtidos em laboratório logo a seguir.

## 2.1. Somador Completo com mux8

Para que seja criado um somador, necessita-se primeiro criar um multiplexador. Para tanto, foi implementado o seguinte código.

```
module full_adder(  
    input A,  
    input B,  
    input Cin,  
    output Cout,  
    output S  
);  
  
assign saida = dado[escolha];  
endmodule
```

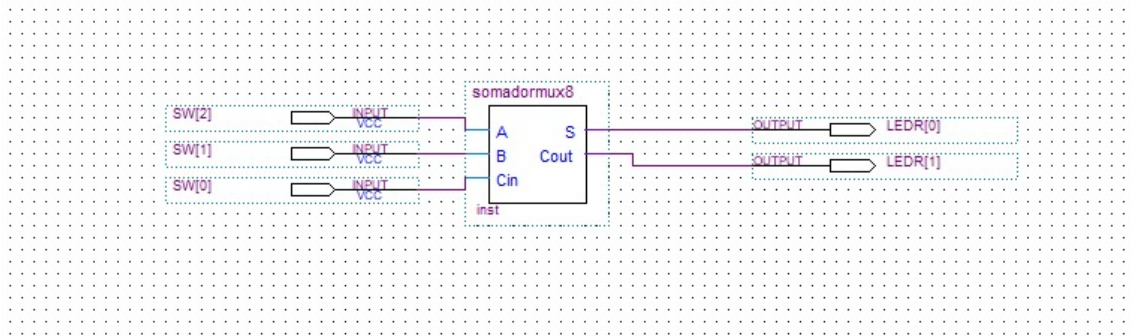


Figura 1.

## 2.2. Multiplexador e decodificador de 16 saídas

Neste exercício, foi necessário implementar um circuito que representasse a seguinte função:

$$f(A, B, C, D, E, F, G) = F.G + A.B.C.D.\bar{E}.\bar{F}.G\bar{A}.\bar{B}.\bar{C}.\bar{D}.\bar{E}.\bar{F}.G + A.\bar{B}.C.E.F.\bar{G} + \bar{A}.B.C.D.\bar{E}.F.\bar{G} + A$$

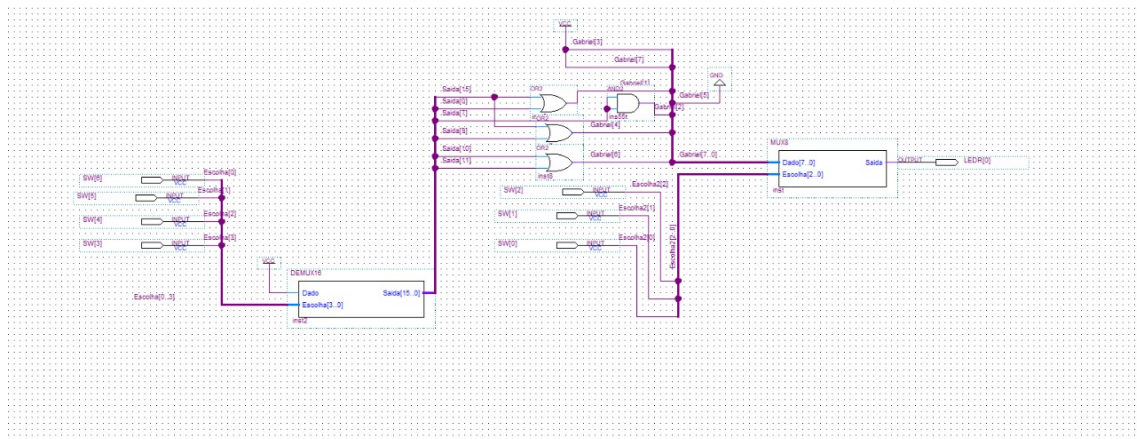
(3)

Para tanto, elaboramos a seguinte tabela verdade

Tabela 2. Tabela Verdade para  $f(A, B, C, D, E, F, G)$

Entradas							Saídas
A	B	C	D	E	F	G	f
0	0	0	0	0	0	1	1
0	1	1	1	0	1	0	1
1	0	0	1	1	0	0	1
1	0	1	*	1	1	0	1
1	1	1	1	0	0	1	1
1	1	1	1	1	0	0	1
*	*	*	*	*	1	1	1

Desta forma, obtivemos a seguinte implementação no Quartus II.



**Figura 2.**

### 3. Conclusões

Com o experimento foi possível compreender melhor o uso de codificadores, decodificadores, multiplexadores, demultiplexadores para construir diversos tipos de circuitos, incluindo somadores. Além disso, percebeu-se o tempo de mudança adequado nos circuitos para que não sejam propagados erros por motivos de atrasos.

Desta forma, tornou-se mais claro o uso de somadores completos e incompletos, circuitos importantes na computação e parte essencial em processadores modernos e diversos dispositivos eletrônicos.

### Referências

### **Auto-Avaliação**

1. V
2. V
3. F
4. V
5. F
6. V
7. V
8. F
9. V
10. F
11. V