

# LÓGICA DIGITAL (1001351)

## EXPERIMENTO NR.5

### Implementação de funções lógicas usando multiplexadores. <sup>1 2</sup>

---

## 1 Instruções Gerais

- Grupos definidos no AVA, só incluir os nomes de participantes efetivos;
- Ler atentamente todo o procedimento desta experiência antes de realizá-la;

## 2 Objetivos da Prática

- Implementação e teste de uma função lógica usando multiplexadores, segundo o método descrito na Seção 4.1.1 (ou 6.1.2 dependendo da edição) da referência [1]: (**Multiplexer Synthesis Using Shannon's Expansion**).
- Comparação de implementações equivalentes em Verilog estrutural e funcional;

## 3 Pré-laboratório

- a) Obtenha uma função lógica de 4 entradas definida por  $f(x_1, x_2, x_3, x_4) = \Sigma m(a, b, c, d, e, f, g)$ , sendo as letras de  $a$  a  $g$  (ou menos, de acordo com o número de participantes) os **dígitos finais** dos RAs dos participantes do grupo. Exemplo:  
 $RAs = \{10001, 10000, 10007, 10004, 10002\} \rightarrow f(x_1, x_2, x_3, x_4) = \Sigma m(1, 0, 7, 4, 2)$

## 4 Procedimentos Experimentais

Deseja-se implementar e testar a função lógica de quatro entradas, obtida no pré-laboratório, usando multiplexadores, segundo o método descrito na seção referenciada [1, 2].

Para isso, use o template abaixo e siga os seguintes passos:

1. Crie um projeto novo no EDAplayground;
2. Coloque o template fornecido na implementação e complete seu código – instanciando e ligando **os multiplexadores que precisar** – para obter a função lógica segundo o método proposto em **Verilog estrutural**;

---

<sup>1</sup>Revisão 30/09/2020: Prof. Mauricio Figueiredo, Prof. Ricardo Menotti.

<sup>2</sup>Revisão 15/09/2021: Prof. Mauricio Figueiredo, Prof. Ricardo Menotti.

```
1 module func4(  
2     input x1, x2, x3, x4,  
3     output f);  
4     // Digite o seu código abaixo  
5  
6 endmodule
```

Código 1: Template para desenvolvimento de uma função lógica de quatro entradas.

3. Desenvolva um *test bench* que ateste o correto funcionamento do seu projeto, a partir de uma implementação em **Verilog funcional** da mesma função (não usar vetores de teste); Para isso:
  - Escreva no início do código Verilog, como comentário, qual a função desejada na forma canônica com os termos em ordem (e.g.  $\Sigma m(0, 1, 2, 4, 7)$  )
  - Construa o Mapa de Karnaugh com os mintermos dos respectivos dígitos dos RAs, conforme explicado acima;
  - Extraia a função mínima correspondente e implemente na forma funcional (e.g. `assign f_esperado = (x1 && x2) || (x3 && x4);` )
  - Faça com que o teste imprima alinhadamente a tabela verdade na console do simulador;
4. Envie o link da implementação no AVA, incluindo os nomes e RAs dos participantes como comentário no início do arquivo de implementação. **Não há necessidade de relatório.**

# Referências Bibliográficas

- [1] S. Brown and Z. Vranesic, *Fundamentals of Digital Logic with Verilog Design*. McGraw Hill, 2000.
- [2] Vanderlei Bonato, “Blocos de construção de circuitos combinacionais.” [Online]. Available: [http://wiki.icmc.usp.br/images/8/88/Aula\\_13\\_-\\_logic2\\_chapter6.pdf](http://wiki.icmc.usp.br/images/8/88/Aula_13_-_logic2_chapter6.pdf)