

Lista de Exercícios Mentoria Embarcatech - Trilha de FPGA

Mentores: Amanda Costa Martinez e Pedro Henrique Trindade

Esta lista tem como objetivo aprofundar o conhecimento prático no desenvolvimento de circuitos digitais usando **VHDL** e **simulação comportamental com testbench**. Os exercícios abordam conceitos essenciais como: controle de fluxo, sinais bidirecionais, tipos de dados (integer, signed, unsigned), estruturas condicionais e paralelismo em hardware.

Ao final da lista, o aluno deverá ser capaz de:

- Modelar circuitos combinacionais e sequenciais em VHDL.
- Aplicar corretamente diferentes tipos de dados e realizar typecasting.
- Utilizar estruturas de controle (if, case, when, with select).
- Criar testbenches completos para validação comportamental.
- Compreender e aplicar paralelismo em circuitos digitais.

Entregáveis

- Código-fonte completo em VHDL (uma pasta por exercício).
- Testbenches funcionais para **todos os exercícios**.
- Screenshots e/ou relatórios de simulação (opcional, mas recomendado).
- Comentários no código explicando o funcionamento.

Exercício 1 – Circuito com Buffer 3-State

Descrição:

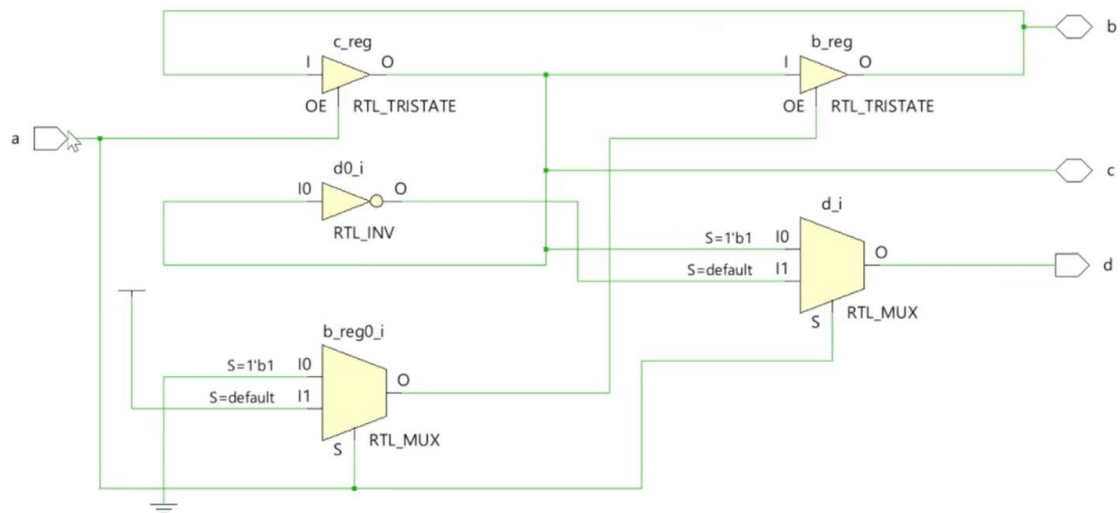
Desenvolva um circuito com 4 portas:

- a (entrada lógica): controla a direção dos dados.
- b, c: sinais **bidirecionais**.
- d: saída lógica.

Funcionamento esperado:

- Se a = '0': o valor de c deve ser passado para b.
- Se a = '1': o valor de b deve ser passado para c.
- A saída d deve ser o **inverso** do sinal presente em c.
- Use buffers 3-state para controlar a direção dos sinais em b e c.

A Figura abaixo representa o RTL do circuito a ser implementado.

**Requisitos:**

- Escreva a entidade e arquitetura do circuito.
- Desenvolva um testbench que valide os dois modos de operação e a saída d.

Exercício 2 – Contador com Diferentes Tipos de Dados

Descrição:

Implemente um contador que **incrementa a cada ciclo de clock**. O contador deve ser feito **três vezes**, utilizando diferentes tipos de dados:

- integer com intervalo de -128 a 127.
- signed(7 downto 0)
- unsigned(7 downto 0)

Requisitos:

- Para cada tipo, crie uma entidade separada.
- O contador deve possuir entrada clk, reset, e uma saída com o valor atual.
- Escreva um testbench para verificar o comportamento com reset e contagem contínua.

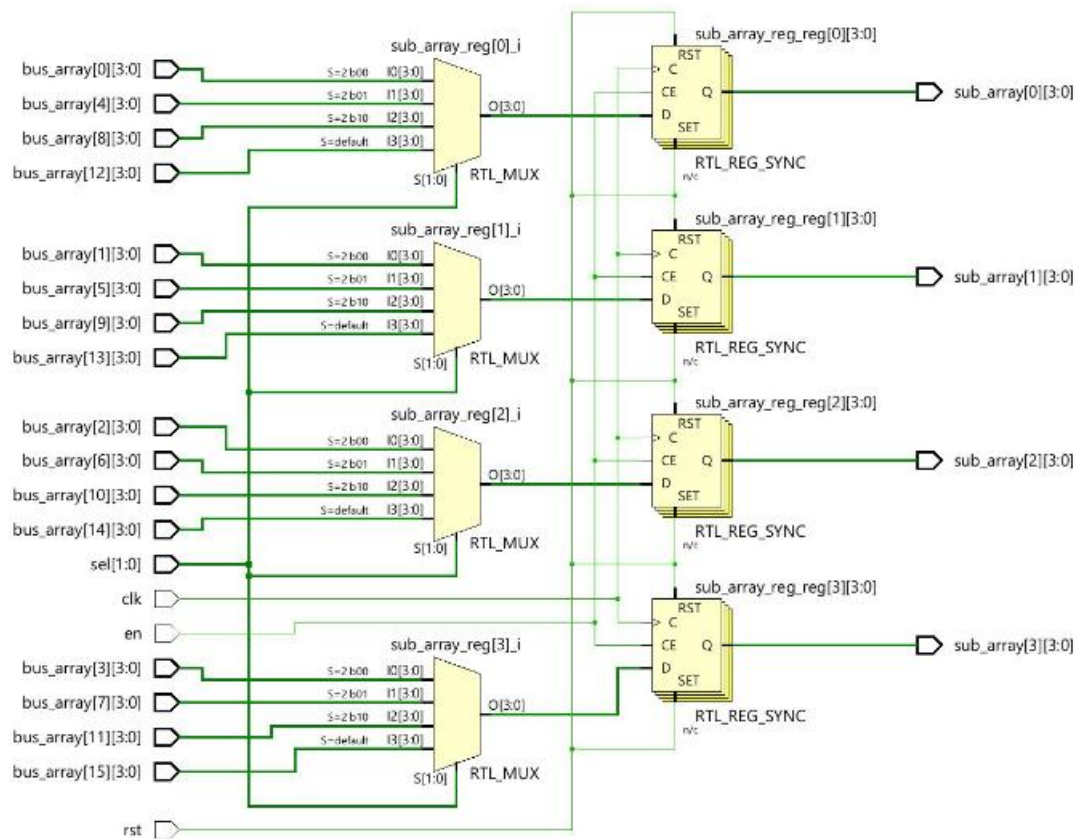
Exercício 3 – MUX com Diferentes Estruturas de Controle

Descrição:

Implemente um multiplexador 4:1 com 4 entradas de 8 bits e uma seleção de 2 bits. Faça 4 versões:

1. Usando case
2. Usando if-elsif
3. Usando when-else
4. Usando with-select

A Figura abaixo representa o RTL do circuito a ser implementado.



Requisitos:

- Cada implementação deve estar em uma arquitetura separada da mesma entidade.
- Crie um **único testbench** que verifique todas as implementações, mudando seletor e valores de entrada.

Exercício 4 – Multiplicações em Paralelo e Sequencial

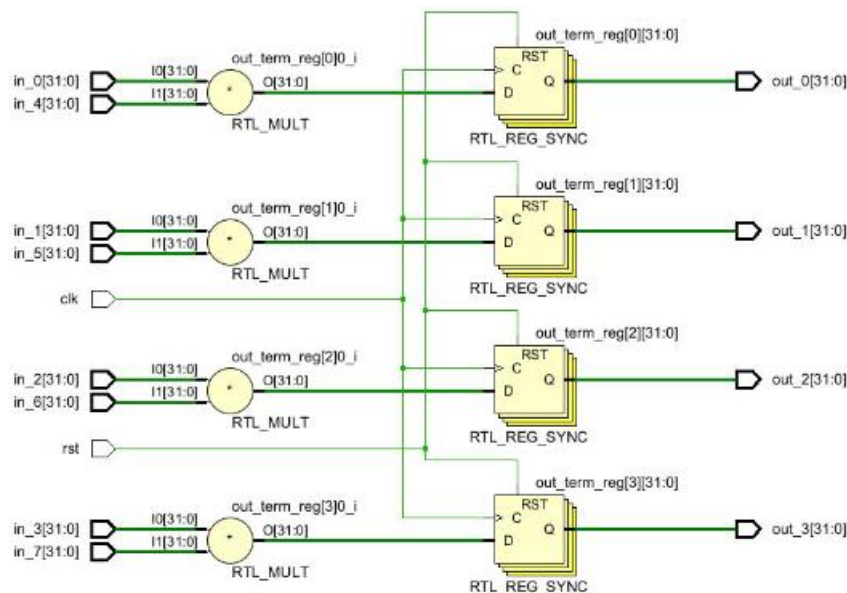
Descrição:

Implemente um circuito que recebe **8 inteiros como entrada** (por exemplo, in(0) to in(7)), e realiza multiplicações **aos pares**:

- $in(0)*in(1)$, $in(2)*in(3)$, ..., $in(6)*in(7)$

Faça **duas versões**:

- **Versão A (paralela):** usa um loop que instancia todas as multiplicações simultaneamente.



- **Versão B (sequencial):** usa uma máquina de estados ou contador interno que executa as multiplicações uma por vez, por ciclo de clock.

Requisitos:

- Use vetores (array of integer) para armazenar entradas e saídas.
- Crie um testbench para validar ambas as versões, com diferentes conjuntos de entrada.