

# Arquitetura e Organização de Computadores

## Turma C - 2022/02

### Projeto do Banco de Registradores do RISC-V

**Objetivo:** projetar, simular e sintetizar um banco de registradores similar ao utilizado no RISC-V.

#### Características:

- 32 registradores de 32 bits
- dois barramentos de leitura
- um barramento para escrita de dado
- registrador 0 (índice zero) é constante, não pode ser alterado. Qualquer leitura deste registrador retorna o valor zero e escritas não afetam o seu valor.

#### Interface:

```
entity XREGS is
  generic (WSIZE : natural := 32);
  port (
    clk, wren, rst           : in std_logic;
    rs1, rs2, rd             : in std_logic_vector(4 downto 0);
    data                     : in std_logic_vector(WSIZE-1 downto 0);
    ro1, ro2                 : out std_logic_vector(WSIZE-1 downto 0));
end bregMIPS;
```

onde:

- WSIZE é o tamanho da palavra do banco de registradores
- *wren*: habilitação de escrita. Ao ser acionado, o registrador endereçado por *rd* é escrito com o valor presente no barramento *data* na transição de subida do relógio.
- *clk*: relógio do circuito.
- *rst*: sinal de reset, zera o conteúdo de todos os registradores do banco.
- *rs1*: endereço do registrador a ser lido em *ro1*, 5 bits.
- *rs2*: endereço do registrador a ser lido em *ro2*, 5 bits.
- *data*: valor a ser escrito no registrador endereçado por *rd*, 32 bits.
- *ro1*: porta de saída para leitura do registrador endereçado por *rs1*.
- *ro2*: porta de saída para leitura do registrador endereçado por *rs2*.

**Obs:** um banco de registradores é similar a uma memória. Deve-se criar um tipo de dados (comando *type*) representando um vetor de palavras (comando *array*) e instanciá-lo.

```
type regArray is array(natural range <>) of std_logic_vector(31 downto 0);
signal breg: regArray;
```

Escritas deve ser realizadas na transição do relógio para gerar estruturas de armazenamento. A ocorrência de uma transição de relógio pode ser verificada com a chamada da função *rising\_edge(clock)*.

**Verificação dos resultados usando ASSERT:**

- ASSERT é um comando utilizado para verificação dos resultados no *testbench*.
- Formato:

ASSERT condição

[REPORT “mensagem”]

[SEVERITY severity\_level]

- REPORT permite enviar mensagens específicas relativas ao erro. A mensagem é escrita caso a condição avalie para falso
- SEVERITY indica o nível de gravidade do erro, que pode assumir um dos seguintes valores: Note, Warning, Error (default) ou Failure.

Usando ASSERT pode-se simplesmente checar se o resultado esperado foi obtido, imprimindo mensagem de erro em caso contrário.

**Entrega:**

- relatório contendo:
  - Identificação do aluno
  - Código VHDL do XREG e do *testbench*
  - Responda: qual a opção usada para simular a constante zero em XREGS[0] ?