

# Arquitetura e Organização de Computadores

## 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      : 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 XREGS;
```

onde:

- WSIZE é o tamanho da palavra do banco de registradores
- *wren*: habilitação de escrita. Ao ser acionado ('1'), o registrador endereçado por *rd* é escrito com o valor presente na entrada *data* na transição de subida do relógio.
- *clk*: relógio do circuito.
- *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.

### 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.

A verificação consiste nas seguintes atividades:

- Preencher o banco de registradores com uma sequência de valores (laço *for* pode ser usado)
- Ler os valores escritos e verificar com ASSERT se estão corretos. Fazer a leitura para ambas as saídas
- Neste processo, escrever um valor diferente de zero no registrador zero do banco, e verificar se ele não é alterado

### **Entrega:**

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