

# Trabalho Prático 1 - 2024/1

## Implementação da Arquitetura "Lobo-Guará"

### Objetivo:

O objetivo deste trabalho é projetar e implementar uma arquitetura de computador de 8 bits, denominada "Lobo-Guará", utilizando a ferramenta de simulação **Logisim Evolution**. A arquitetura deve ser capaz de executar um conjunto específico de instruções e deve ser organizada de acordo com o modelo monociclo.

### Instruções da Arquitetura "Lobo-Guará":

A arquitetura "Lobo-Guará" possui um conjunto de instruções que incluem operações de branch, jump, load, store, movimentação de dados, aritmética e lógica. A lista completa de instruções está disponível na próxima página.

### Componentes a serem Implementados:

Os alunos devem implementar os seguintes componentes da arquitetura "Lobo-Guará" no **Logisim Evolution**:

- Unidade Lógico-Aritmética (ULA) conforme especificações da arquitetura.
- Unidade de Controle para interpretação e execução das instruções.
- Registradores necessários para o funcionamento da arquitetura.

### Entrega:

O trabalho é individual e deverá ser entregue em duas partes, conforme detalhado a seguir:

#### Parte 1 - Diagrama da Arquitetura (Peso: 40%):

- Diagrama em PDF contendo o diagrama da arquitetura e o projeto da ULA com eventuais detalhes do projeto.

#### Parte 2 - Projeto no Logisim Evolution (Peso: 60%):

- Projeto completo da arquitetura no formato Logisim Evolution, incluindo todos os componentes implementados e a interconexão entre eles.

As datas limite de **entrega serão sempre às 23:59 do dia**, impreterivelmente.

Casos não tratados no enunciado deverão ser discutidos com o professor.

Os trabalhos devem ser feitos individualmente. **A cópia do trabalho (plágio), acarretará em nota igual a Zero para todos os envolvidos.**

**Os trabalhos deverão ser apresentados de forma oral pelo aluno. A nota irá considerar domínio do tema, robustez da solução e rigorosidade da metodologia.**

Opcode	Tipo	Mnemonic	Nome	Operação
Controle				
0000	R	brzr	Branch On Zero Register	if (R[ra] == 0) PC = R[rb]
0001	I	brzi	Branch On Zero Immediate	if (R[0] == 0) PC = PC + Imm.
0010	R	jr	Jump Register	PC = R[rb]
0011	I	ji	Jump Immediate	PC = PC + Imm.
Dados				
0100	R	ld	Load	R[ra] = M[ R[rb] ]
0101	R	st	Store	M[ R[rb] ] = R[ra]
0110	R	movr	Move Register	R[ra] = R[rb]
0111	I	movh	Move High	R[0] = {Imm. + R[0](3:0)}
1000	I	movl	Move Low	R[0] = {R[0](7:4) + Imm.}
Aritmética				
1001	R	add	Add	R[ra] = R[ra] + R[rb]
1010	R	sub	Sub	R[ra] = R[ra] - R[rb]
Lógica				
1011	R	and	And	R[ra] = R[ra] & R[rb]
1100	R	or	Or	R[ra] = R[ra]   R[rb]
1101	R	not	Not	R[ra] = ! R[rb]
1110	R	slr	Shift Left Register	R[ra] = R[ra] << R[rb]
1111	R	srr	Shift Right Register	R[ra] = R[ra] >> R[rb]

Tipo R							
7	6	5	4	3	2	1	0
opcode				Ra		Rb	

Tipo I							
7	6	5	4	3	2	1	0
opcode				Imm			

