

# CI1212 Arquitetura de Computadores - 2025/1 - UFPR

## Especificação do Trabalho 2

O aluno deverá desenvolver o projeto de implementação da arquitetura REDUX-V. Trata-se de uma arquitetura de 8-bits, do tipo load-store endereçada por byte. A organização a ser implementada **poderá** ser Monociclo. A lista de instruções bem como seus formatos são dados na próxima página. **A memória RAM utilizada deverá ser o componente “dual\_port\_ram” disponibilizado. O clock do processador deverá ser configurado para durar 2 ticks, caso contrário a memória não funcionará corretamente.**

**Adicionalmente, a implementação deverá utilizar os opcodes reservados para pelo menos 3 instruções novas a sua escolha.**

O trabalho deve ser desenvolvido individualmente. Recomenda-se a seguinte ordem para o desenvolvimento do trabalho:

1. Desenvolvimento com diagrama de caixas do projeto do caminho de dados (datapath) do processador.
2. Desenvolvimento com diagrama de caixas do projeto interno da ULA, pensando no código da ULA a ser usado para cada operação.
3. Desenvolvimento no Logisim Evolution do projeto com componentes e fios.
4. **Reescrita do programa entregue no Trabalho 1 utilizando as instruções novas.**
5. **O Assembly do novo programa deverá estar presente no relatório.**
6. **O relatório deverá conter uma justificativa das instruções escolhidas, demonstrando a utilidade delas no novo programa de teste.**
7. **As memórias de controle e a memória RAM devem ser entregues preenchidas.**

## Regras Gerais de Entrega

A implementação será feita no logisim evolution, onde poderão ser utilizados todos os componentes prontos ali existentes. Cada aluno deverá entregar o diagrama do REDUX-V bem como o projeto no logisim evolution. A entrega será feita pelo Moodle dividida em duas partes

- Diagrama em PDF contendo o **diagrama (datapath) do bloco operativo do REDUX-V, o projeto da ULA** com eventuais detalhes do projeto, os códigos assembly comentados que estão sendo entregues.
- Projeto no formato **Logisim Evolution**

As datas limite de **entrega serão sempre às 6h (a.m.) do dia**, impreterivelmente. (não confundir com 18h)

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 rigurosidade da metodologia.**

# Descrição da Arquitetura REDUX-V

## Informações gerais

- 8. **Tamanho da palavra:** 8 bits
- 9. Arquitetura **Load-Store**
- 10. **Granularidade** do endereçamento: 1 Byte
- 11. Arquitetura **Von Newmann**

## Conjunto de instruções (ISA)

Opcode	Tipo	Mnemonic	Nome	Operação
0000	R	brzr	Branch On Zero Register	if (R[ra] == 0) PC = R[rb]
0001	I	ji	Jump Immediate	PC = PC + Imm.
0010	R	ld	Load	R[ra] = M[ R[rb] ]
0011	R	st	Store	M[ R[rb] ] = R[ra]
0100	I	addi	Add Immediate	R[0] = R[0] + Imm.
0101	Reservado - Instruções adicionais			
0110				
0111				
1000	R	not	Not	R[ra] = not R[rb]
1001	R	and	And	R[ra] = R[ra] and R[rb]
1010	R	or	Or	R[ra] = R[ra] or R[rb]
1011	R	xor	Xor	R[ra] = R[ra] xor R[rb]
1100	R	add	Add	R[ra] = R[ra] + R[rb]
1101	R	sub	Sub	R[ra] = 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]

## Formatos das instruções

Tipo I								
Bits	7	6	5	4	3	2	1	0
	Opcode				Imm.			

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