

LUCAS DOS SANTOS PIMENTEL

**MICROPROCESSADOR MIPS**

CUIABÁ

2024

LUCAS DOS SANTOS PIMENTEL

**MICROPROCESSADOR MIPS**

Trabalho apresentado ao Instituto Federal de Mato Grosso como requisito para a conclusão do curso de Engenharia da Computação.

Professor: Ruy de Oliveira

CUIABÁ

2024

**RESUMO**

sumário

[1 INTRODUÇÃO 5](#_Toc175481219)

[2 Arquitetura e componentes 6](#_Toc175481220)

[2.1 Program Counter 6](#_Toc175481221)

[2.2 Memória de instruções 7](#_Toc175481222)

[2.3 Banco de Registradores 7](#_Toc175481223)

[2.4 ALU 7](#_Toc175481224)

[2.5 Memória de Dados 7](#_Toc175481225)

[2.6 Controlador 7](#_Toc175481226)

[3 Instruções 8](#_Toc175481227)

[3.1 Registradores 8](#_Toc175481228)

[3.2 Barramento 8](#_Toc175481229)

[4 Modos de endereçamento 9](#_Toc175481230)

[5 cojunto de instruções 9](#_Toc175481231)

[6 aplicações e uso do z80 9](#_Toc175481232)

[7 conclusão 9](#_Toc175481233)

[8 referencia bibliografica 9](#_Toc175481234)

# INTRODUÇÃO

**1.1 Contextualização**

A arquitetura MIPS (Microprocessor without Interlocked Pipeline Stages) é uma das arquiteturas de conjunto de instruções mais influentes e amplamente estudadas no campo da engenharia de computadores. Desenvolvida nos anos 1980 como parte do movimento RISC (Reduced Instruction Set Computer), a MIPS foi projetada com o objetivo de simplificar as operações de hardware, melhorando a eficiência e o desempenho dos processadores.

Embora a MIPS tenha evoluído ao longo do tempo para incluir funcionalidades mais avançadas, como o pipeline e a execução fora de ordem, seu núcleo básico continua sendo uma ferramenta educacional essencial para estudantes e profissionais que desejam compreender os princípios fundamentais da arquitetura de computadores. Este trabalho explora a implementação básica da arquitetura MIPS utilizando o Logisim Evolution, uma ferramenta de simulação digital que permite a construção e análise de circuitos digitais.

**1.2 Objetivos do Artigo**

O principal objetivo deste artigo é documentar a implementação da arquitetura MIPS em um ambiente simulado, utilizando o Logisim Evolution. O foco está na criação de um processador MIPS básico, capaz de executar um conjunto de instruções fundamentais sem a complexidade adicional do pipeline. Especificamente, o artigo busca:

1. **Apresentar os conceitos básicos da arquitetura MIPS**, abordando sua estrutura e o conjunto de instruções utilizado.
2. **Detalhar o processo de implementação no Logisim Evolution**, incluindo a configuração da memória, banco de registradores, unidade de controle e ALU (Unidade Lógica e Aritmética).
3. **Demonstrar a execução de instruções básicas** (aritméticas, lógicas, de memória e de controle de fluxo) através de simulações no Logisim.
4. **Discutir as limitações e desafios encontrados durante a implementação**, bem como possíveis melhorias e expansões futuras do projeto.
5. **Contribuir para o aprendizado de estudantes e entusiastas** de arquitetura de computadores, fornecendo um exemplo prático de como construir um processador MIPS funcional em um ambiente de simulação.

# Arquitetura e componentes

Para entender sobre a arquitetura MIPS é necessário saber sobre os seus componentes e de como eles estão organizados. Resumidamente ele pode ser separado por 6 grandes blocos que juntos formar a arquitetura como um todo.

## 2.1 Program Counter

Quando se fala de processadores no geral, o program counter sempre aparece sendo essencial e indispensável. Ele tem o papel de apontar o endereço da memória a ser usada que contém uma instrução. No final do ciclo de clock normalmente é somado 1 ao endereço que estava guardando para apontar para a próxima instrução.

## 2.2 Memória de instruções

Aqui é armazenado todas as instruções que o programa irá rodar, podendo ser previamente carregado em um arquivo com as instruções com código de máquina. Cada endereço dessa memória se refere a um instruções que no caso desse projeto são de 16 bits.

## 2.3 Banco de Registradores

No banco de registradores temos um conjunto de registradores que serve para auxiliar em operações e armazenar dados temporários dependendo de cada instrução. Pode-se colocar valores previamente em algum dos registradores, porém diferentemente da memória de instruções aqui durante a execução do programa valores podem ser escritos dentro dos registradores do mesmo jeito que pode ser lidos.

## 2.4 ALU

A unidade lógica aritmética ou em inglês arithmetic logic unit (ALU), é onde é feito todas as operações lógicas e aritméticas das instruções, presente em quase todas as instruções.

## 2.5 Memória de Dados

Na memória de Dados é armazenado os valores em definitivo das operações a qual o usuário deseja guardar, sendo ele volátil, ou seja, caso o programa seja encerrado de maneira indevida, os dados que estavam na memória se perderão.

## 2.6 Controlador

A unidade de controle, é fundamental para qualquer processador. É a parte responsável por controlar entradas e saída de dados de cada bloco dependendo de cada instrução. Mesmo sendo essencial é possível o programa rodar sem ele, porém os dados teriam que ser direcionados manualmente, o que demanda mas tempo porém é melhor para o aprendizado.

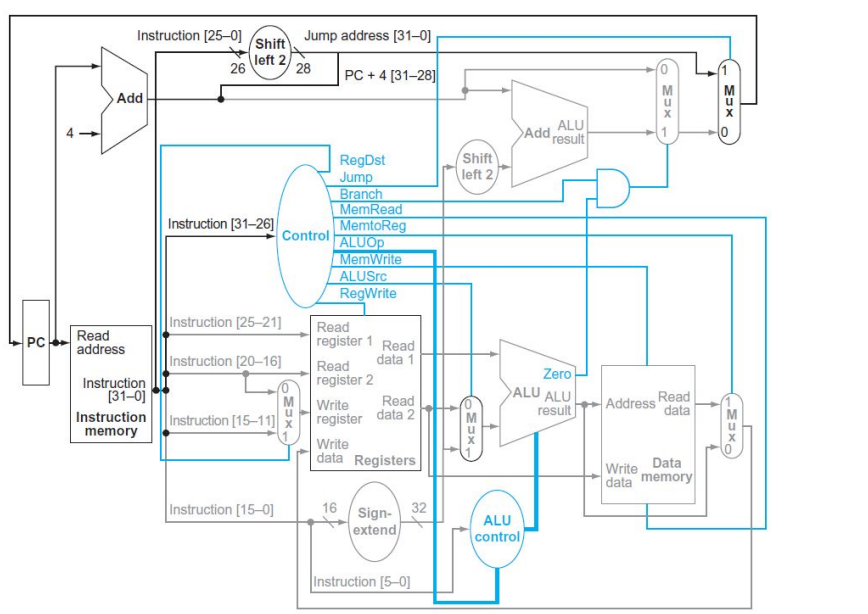


Figura 1: Arquitetura MIPS resumida

# Instruções

As instruções do MIPS se divide mem três tipos (R, I e J) e cada uma tem funcionalidades diferentes e usam números de registradores diferentes.

## Registradores

## Barramento

# Modos de endereçamento

# cojunto de instruções

# aplicações e uso do z80

# conclusão

# referencia bibliografica