



**UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
FACULDADE DE COMPUTAÇÃO**

Sistemas Digitais

Ricardo Ribeiro do Santos

**MAYKON KAZUHIRO FALCÃO TAMANAHA
HEMANUELA GABRIELA GUTIERREZ FERNANDES
LUÍS MIGUEL TABORDA FIALHO**

Implementando Simulador RISC-V

**CAMPO GRANDE - MS
30/04/2025**



1 INTRODUÇÃO

O RISC-V foi criado em 2010 pelo Professor Krste Asanović e os estudantes de graduação Yunsup Lee e Andrew Waterman, como parte do Laboratório de Computação Paralela na Universidade da Califórnia em Berkeley, que estava sob direção do Professor David Patterson (RISC-V International, 2025). RISC-V é uma arquitetura de conjunto de instruções - ISA *Open Source*, ou seja, está disponível publicamente para uso e modificação.

O presente trabalho consiste no desenvolvimento de um simulador de instruções RISC-V, uma ISA amplamente utilizada em sistemas embarcados e aplicações acadêmicas. Nesse sentido, o simulador foi projetado para interpretar e executar instruções RISC-V a partir de arquivos de entrada, simulando o comportamento de registradores e memória. Por isso, o objetivo deste trabalho é proporcionar uma ferramenta educacional para o estudo e compreensão do funcionamento interno de processadores baseados na arquitetura RISC-V.

2 OBJETIVOS

Este trabalho foi desenvolvido visando desenvolver um simulador da arquitetura RISC-V, capaz de interpretar e executar instruções nos formatos R, I, U e S, conforme especificado na documentação oficial da ISA RISC-V.

O programa deve receber como entrada um arquivo de texto contendo instruções RISC-V válidas, realizar o processamento sequencial, simular as operações nos registradores assim como na memória, e por fim, gerar um arquivo de saída com o estado final dos registradores utilizados e das posições de memória modificadas.

O simulador resultante deve interpretar as instruções de `addi`, `add`, `lui` e `sw`, permitindo a realização de operações aritméticas e armazenamento de dados. Outras instruções existentes em um simulador completo não estão incluídas.

Como exemplo, o seguinte conjunto de instruções deve ser corretamente executado:

```
addi t1 , zero , 10      # t1 = 10
addi t2 , zero , 20      # t2 = 20
add  t3 , t1 , t2         # t3 = t1 + t2 = 30
lui  t4 , 0x10000         # t4 = 0x10000000
sw   t3 , 0(t4)           # memória[t4] = t3
```

A saída esperada após a execução desse código é:

```
t1 = 10
t2 = 20
```

t3 = 30
t4 = 268435456
mem(268435456) = 30

Esse simulador visa apoiar o aprendizado e a compreensão prática da arquitetura RISC-V, possibilitando a experimentação com instruções ao nível de máquina de forma acessível e didática.

3 METODOLOGIA

A implementação do simulador foi realizada em C++, sendo dividida em cinco partes:

- **Registradores (Registers):** Representam os 32 registradores da arquitetura RISC-V, que permitem a leitura e escrita de valores.
- **Memória (Memory):** Simula a memória principal, utilizando um mapa para armazenar valores associados a endereços.
- **Instruções (Instruction):** Implementa a lógica para interpretar e executar diferentes formatos de instruções RISC-V, como R, I, S e U.
- **Simulador (Simulator):** Coordena a execução das instruções, gerencia o contador de programa (PC_counter) e exibe o estado final dos registradores e da memória.
- **Parser (Parser):** Responsável por ler e formatar os arquivos de entrada, removendo comentários e espaços desnecessários.

O fluxo de execução do simulador pode ser descrito em três passos:

1. O arquivo de entrada contendo as instruções é lido e processado pelo parser.
2. As instruções são armazenadas em um vetor e executadas sequencialmente pelo simulador.
3. Após a execução, o estado dos registradores e da memória é exibido.

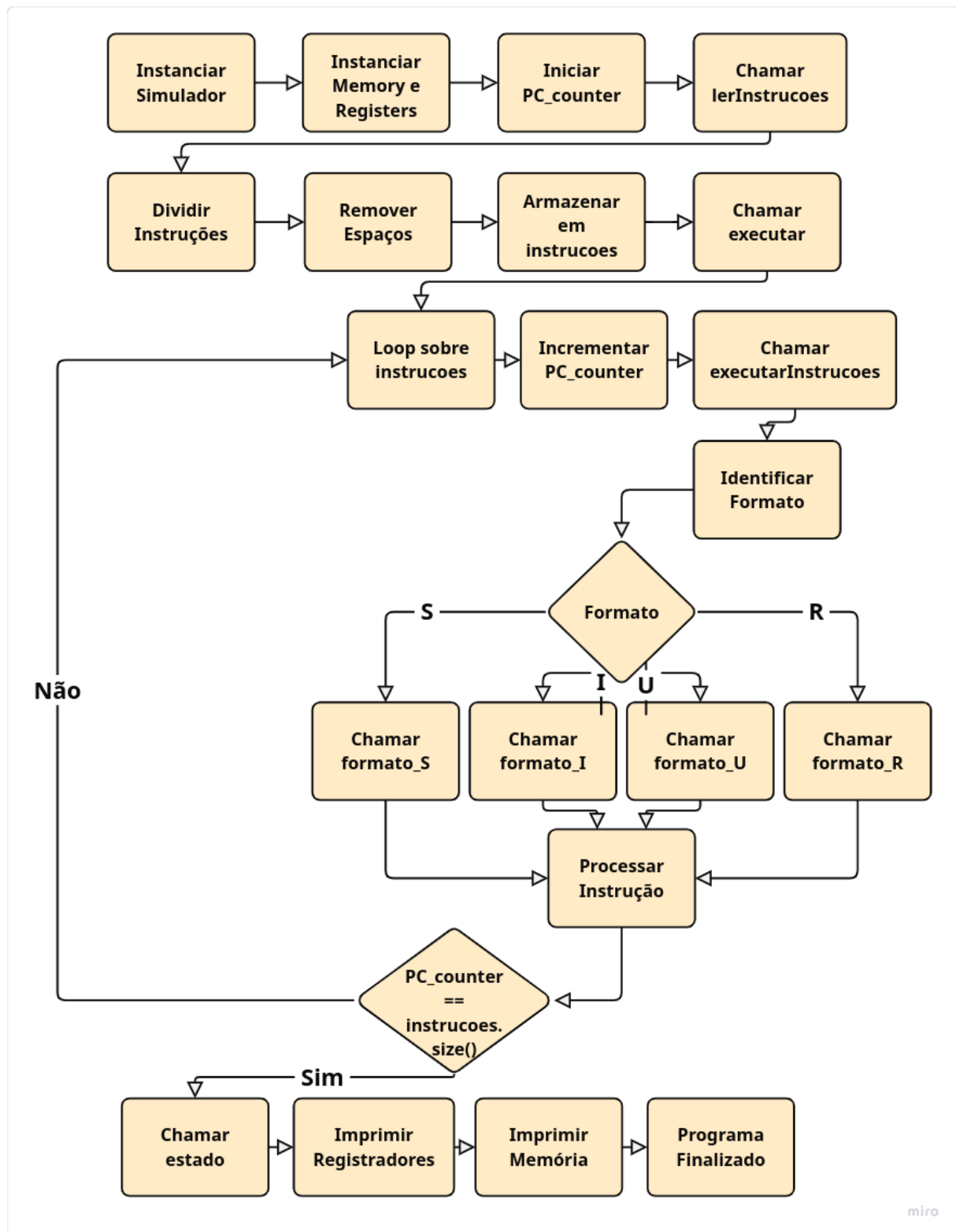


Figura 1 – Fluxograma do código

4 RESULTADOS

Para testar o simulador, foram usados dois arquivos de texto, sendo eles de título `entrada1.txt` e `entrada2.txt`, ambos contendo instruções RISC-V válidas. Os resultados obtidos comprovam que o simulador consegue interpretar e executar correta-

mente as instruções, atualizando os valores dos registradores e da memória conforme o esperado.

EXEMPLO DE EXECUÇÃO COM ENTRADA1.TXT:

Instruções:

```
addi t1, zero, 10      # t1 = 10
addi t2, zero, 20      # t2 = 20
add  t3, t1, t2        # t3 = t1 + t2 = 30
lui   t4, 0x10000      # t4 = 0x10000000 = 268435456
sw    t3, 0(t4)        # memória[268435456] = 30
```

Estado final dos registradores:

- t1 = 10
- t2 = 20
- t3 = 30
- t4 = 268435456

Estado final da memória:

- mem(268435456) = 30

EXEMPLO DE EXECUÇÃO COM ENTRADA2.TXT:

Instruções:

```
addi t1, zero, 5       # t1 = 5
addi t2, zero, 7       # t2 = 7
add  t3, t1, t2        # t3 = t1 + t2 = 12
lui   t4, 0x10000      # t4 = 0x10000000 = 268435456
sw    t3, 0(t4)        # memória[268435456] = 12
```

Estado final dos registradores:

- t1 = 5
- t2 = 7
- t3 = 12
- t4 = 268435456

Estado final da memória:

- mem(268435456) = 12

Podemos concluir que os resultados confirmam que o simulador desenvolvido é funcional e atende aos requisitos propostos.

5 CONCLUSÕES

Neste trabalho, foi desenvolvido um simulador de instruções RISC-V. Pela observação dos resultados, podemos dizer que este trabalho resultou em uma ferramenta educacional e eficaz para a execução e análise de instruções no modelo RISC-V.

Assim, permitindo a visualização do estado dos registradores e da memória após sua execução, de modo a facilitar a compreensão sobre o funcionamento interno de processadores que se baseiam nessa arquitetura.

É possível a continuidade e expansão desse trabalho para incluir mais instruções e suportes a outros tipos de entrada.

REFERÊNCIAS

cplusplus.com. **C++ Reference Manual**. 2025. <<https://cplusplus.com/>>. Acesso em: 20 abr. 2025.

PATTERSON, D. A.; HENNESSY, J. L. **Computer Organization and Design RISC-V Edition: The Hardware Software Interface**. [S.l.]: Morgan Kaufmann, 2017. Acesso em: 20 abr. 2025.

RISC-V International. **RISC-V Overview**. 2025. <<https://riscv.org/about/>>. Acesso em: 20 abr. 2025.

STALLINGS, W. **Arquitetura e Organização de Computadores**. [S.l.]: Pearson, 2018. Livro-texto sobre arquitetura de computadores.

WATERMAN, A.; ASANOVIĆ, K. **The RISC-V Instruction Set Manual, Volume I: User-Level ISA**. [S.l.]: RISC-V Foundation, 2019. Acesso em: 20 abr. 2025.