

Simulador RISC-V

COM PIPELINING

Trabalho de Arquitetura de Computadores

Professor Leandro Santiago

Integrantes:

- Eduardo Rottschaefter
- João Victor Machado Sperle
- Allan Vignoli dos Santos
- João Vitor Pereira Rodrigues
- José Enrique Viana de Oliveira

UM SIMULADOR EDUCACIONAL que busca simular um processador RISC-V que utiliza o pipeline clássico de 5 estágios. A linguagem selecionada para a implementação foi o “Java”, devido a sua versatilidade de programação orientada a objetos. O simulador permite as seguintes tarefas de programas em Assembly RISC-V:

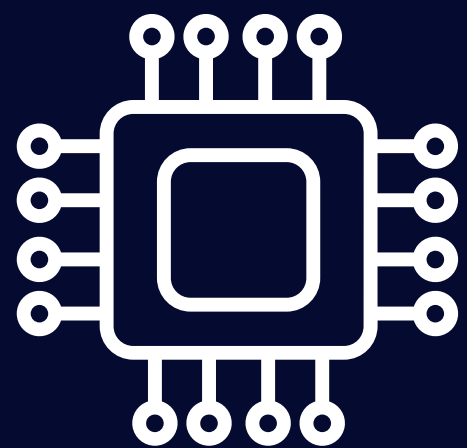
- Execução ;
- Visualização ;
- Depuração.





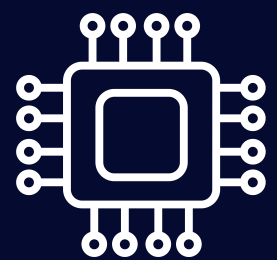
OBSERVAÇÕES:

- O simulador é voltado para fins didáticos e não cobre TODAS as instruções do RISC-V.
- O projeto pode ser expandido para incluir interface gráfica, forwarding, detecção de hazards e suporte a mais instruções.



ESTRUTURA: DO PROJETO





ESTRUTURA:

AUXILIARES (classes utilitárias)	ESTÁGIOS	UNIDADE FUNCIONAIS	EXTRA
<ul style="list-style-type: none">• AssemblyParser.java• Encoder.java• Decoder.java• InstructionInfo.java	<ul style="list-style-type: none">• InstructionFetch.java• ReadMemory.java• Execution.java• ReadMemory.java• WriteBack.java	<ul style="list-style-type: none">• BancoRegistadores.java• Memoria.java• ALU.java	<ul style="list-style-type: none">• PIPELINE: Registadores de pipeline entre os estágios.• exemple.asm: Exemplo de programa Assembly RISC-V para teste do simulador.



COMO

FUNZIONA ?





COMO FUNCIONA ?

- 1) O AssemblyParser lê o arquivo .asm, resolve labels e carrega instruções/dados na memória.
- 2) O pipeline executa instruções em 5 estágios, propagando sinais e dados entre os registradores de pipeline.
- 3) O simulador suporta instruções aritméticas, lógicas, loads, stores, branches e jumps do RISC-V básico.



COMO EXECUTAR:

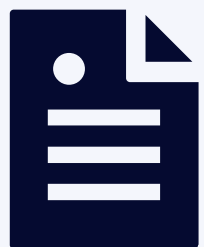
1. Compile todos os arquivos Java:

```
javac */*.java *.java
```

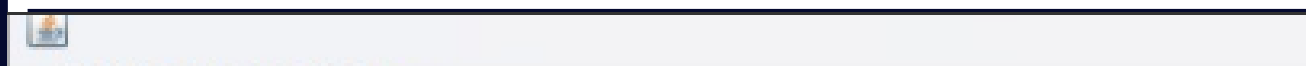
2. Execute a classe principal (exemplo):

```
java Processador
```

3. Escreva um arquivo *".asm"* para testar diferentes programas .



INTERFACE



==== Pipeline Stages ciclo 0 ====
IF/ID: add
ID/EX: addi
EX/MEM: addi
MEM/WB: addi
=====

==== Pipeline Stages ciclo 1 ====
IF/ID: addi
ID/EX: add
EX/MEM: addi
MEM/WB: addi
=====

==== Pipeline Stages ciclo 2 ====
IF/ID: add
ID/EX: addi
EX/MEM: add
MEM/WB: addi
=====

==== Pipeline Stages ciclo 3 ====
IF/ID: sub
ID/EX: add
EX/MEM: addi
MEM/WB: add
=====


==== Pipeline Stages ciclo 4 ====
IF/ID: mul
ID/EX: sub
EX/MEM: add
MEM/WB: addi
=====

==== Pipeline Stages ciclo 5 ====
IF/ID: and
ID/EX: mul
EX/MEM: sub
MEM/WB: add
=====

==== Pipeline Stages ciclo 6 ====
IF/ID: or
ID/EX: and
EX/MEM: mul
MEM/WB: sub
=====

==== Registradores ====
Registradores[0] = 0; Registradores[1] = 10; Registradores[2] = 0; Registradores[3] = 0; Registradores[4] = 0; Registradores[5] = 0; Registradores[6] = 0; Registradores[7] = 0;
Registradores[8] = 0; Registradores[9] = 0; Registradores[10] = 0; Registradores[11] = 0; Registradores[12] = 0; Registradores[13] = 0; Registradores[14] = 0; Registradores[15] = 0;
Registradores[16] = 0; Registradores[17] = 0; Registradores[18] = 0; Registradores[19] = 0; Registradores[20] = 0; Registradores[21] = 0; Registradores[22] = 0; Registradores[23] = 0;
Registradores[24] = 0; Registradores[25] = 0; Registradores[26] = 0; Registradores[27] = 0; Registradores[28] = 0; Registradores[29] = 0; Registradores[30] = 0; Registradores[31] = 0;

Selecionar Arquivo

 Digite o nome do arquivo (incluindo o .asm):

OKCancel

Esse é o projeto de simulador RISC - V

Arquitetura de Computadores

FIM

• Contribuidores:

Eduardo Rottschaefter

João Victor Machado Sperle

Allan Vignoli dos Santos

João Vitor Pereira Rodrigues

José Enrique Viana de Oliveira

• Linguagem:



Java

