## Projeto 02 - Arquitetura de Computadores II

## Simulador ciclo a ciclo de um pipeline de 5 estágios com cache integrada

**Integrantes**: Arthur Cardoso, Henrique de Sousa Paixão, Lais Vitoria, Myllena Rodrigues Oliveira e Samara Solis Sanches

Este documento tem como objetivo descrever de forma clara e completa o funcionamento e as etapas de implementação do Projeto 02 da disciplina Arquitetura de Computadores II. Ele foi atualizado com base na versão final do projeto concluído, servindo como guia técnico e de apoio para a apresentação do grupo em sala.

Objetivo Geral:

O projeto consiste no desenvolvimento de um simulador interativo de pipeline RISC-V, implementado em JavaScript, HTML e CSS. O simulador permite visualizar o funcionamento ciclo a ciclo de um processador com pipeline de 5 estágios (IF, ID, EX, MEM, WB), incluindo o tratamento de hazards, forwarding, predição de desvios e o uso de caches L1 de instrução e de dados.

Objetivos Específicos:

* Implementar a simulação completa do pipeline RISC-V de 5 estágios.
* Aplicar técnicas de forwarding e stalls para correção de hazards de dados.
* Adicionar preditor de desvio (1-bit) para análise de fluxo de controle.
* Integrar caches L1 de instrução e dados com política LRU e write-back/write-allocate.
* Criar uma interface gráfica (HTML/CSS/JS) que permita interação com o simulador.
* Exibir o estado do pipeline, registradores e métricas em tempo real.
* Exportar automaticamente os resultados em formato CSV para análise posterior.

Ferramentas Utilizadas:

* + Visual Studio Code (editor principal);
  + Live Server (execução via navegador);
  + HTML5, CSS3 e JavaScript (estrutura da interface e lógica de controle);
  + Simulator.js (núcleo lógico da simulação).

Estrutura do Projeto

Lógica Interna (simulador.js):

A lógica do simulador é implementada no arquivo simulator.js. Ele gerencia os estágios do pipeline, realiza o avanço dos ciclos (função tick) e coleta métricas de desempenho. Inclui ainda o tratamento de hazards, forwarding, e a integração com as caches e o preditor de desvio.

Interface Gráfica (index.html):

A interface foi construída em HTML e CSS com JavaScript integrado. Ela contém os seguintes elementos principais:

* Caixa de texto (textarea) para entrada do código do programa;
* Botões de controle: Carregar Programa, Próximo Ciclo, Executar Tudo, Resetar e Exportar CSV;
* Tabela de visualização do pipeline (IF, ID, EX, MEM, WB);
* Tabela de registradores (x0–x31);
* Tabela de métricas (Ciclos, CPI, Stalls, Flushes, Branch Hits e Cache Hit Rates).

Etapas do Desenvolvimento:

|  |  |  |
| --- | --- | --- |
| Etapa | Descrição | Status |
| Fase 1 – Lógica Interna | Implementação do pipeline, forwarding e hazards. | Concluído |
| Fase 2 – Caches e Preditor | Implementação do preditor de desvio e caches L1I/L1D. | Concluído |
| Fase 3 – Interface (UI) | Desenvolvimento da interface interativa em HTML/CSS. | Concluído |
| Fase 4 – Integração e Testes | Conexão do simulador com a interface e testes com benchmarks. | Concluído |
| Fase 5 – Exportação e Análise | Geração de CSV e coleta de métricas de desempenho. | Concluído |

Resultados Obtidos:

O simulador apresenta uma execução visual e detalhada do pipeline. Cada ciclo de clock pode ser acompanhado passo a passo, com atualização em tempo real dos registradores, caches e métricas. A interface permite compreender facilmente o comportamento do processador, tornando o projeto uma ferramenta didática completa.

Conclusão:

O projeto foi concluído com sucesso, atendendo a todos os requisitos previstos. O simulador fornece uma representação fiel e interativa do funcionamento interno de um processador com pipeline RISC-V. Além de servir como apoio para a compreensão de conceitos complexos de arquitetura, o sistema possibilita a análise de desempenho por meio das métricas exportadas em CSV.