Oldowan: Ambiente Web Interativo para Simulação do Processador MIPS32

João V. P. Soares[1], João P. R. Linares[1], Gerson Geraldo H. Cavalheiro[1]

[1]Bacharelado em Engenharia de Computação
Universidade Federal de Pelotas (UFPel)
R. Gomes Carneiro, 01 – Balsa, Pelotas – RS, 96010-610

{jvpsoares,jprlinares,gerson.cavalheiro}@inf.ufpel.edu.br

Resumo. *O ensino de Arquitetura de Computadores desafia a ilustração de conceitos complexos como pipeline e hierarquia de memória. Este artigo apresenta o Oldowan, um simulador MIPS32 baseado na Web. Diferente de soluções que exigem instalação, o Oldowan oferece acessibilidade imediata via navegador. A ferramenta permite execução passo a passo e visualização detalhada dos estágios do pipeline, Caches de Instrução e Dados, e registradores. O simulador serve como apoio pedagógico para a compreensão prática de processadores RISC, eliminando barreiras de configuração de ambiente.*

1. Introdução

O estudo de arquitetura e programação de computadores é fundamental para a formação de profissionais de computação, pois estabelece a conexão entre o hardware físico e o software, permitindo compreender como os recursos das linguagens de programação são efetivamente implementados em baixo nível. Enquanto arquiteturas didáticas simplificadas, como o Neander [Weber 2009], são excelentes para introduzir conceitos básicos de lógica digital e fluxo de dados, o avanço no currículo exige o estudo de processadores mais robustos e próximos da realidade de mercado, como o MIPS (Microprocessor without Interlocked Pipeline Stages), que implementa o modelo de arquitetura RISC (Reduced Instruction Set Computer).

A arquitetura MIPS é amplamente adotada na literatura clássica, como em [Hennessy and Patterson 2014] e [Stallings 2017], devido à sua organização RISC (*Reduced Instruction Set Computer*) limpa e regular. No entanto, a complexidade introduzida por conceitos arquiteturais avançados, como a execução de instruções em *pipeline*, o tratamento de riscos de dados (*hazards*) e a hierarquia de memória (*caches*), torna o aprendizado puramente teórico um desafio significativo para os estudantes.

Embora existam simuladores consolidados para a arquitetura MIPS, como o SPIM ou o MARS [Vollmar and Sanderson 2006], muitos exigem a instalação de software específico ou dependências de sistema (como o ambiente de execução Java). Isso pode criar barreiras de acesso em laboratórios institucionais com restrições ou em dispositivos pessoais dos alunos. Além disso, a visualização do estado interno do processador, especificamente o comportamento dinâmico das *caches* e a propagação de dados ao longo dos estágios do *pipeline*, nem sempre é apresentada de forma visualmente clara e intuitiva para o aluno em fase inicial de aprendizado.

Nesse contexto, este trabalho propõe o **Oldowan**, um simulador MIPS32 interativo desenvolvido integralmente com tecnologias Web. O objetivo primário é fornecer uma plataforma acessível onde os alunos possam escrever, montar e executar códigos na linguagem Assembly, visualizando em tempo real o impacto de cada instrução nos registradores, na memória cache e nos registradores de inter-estágio do *pipeline*. A abordagem centrada na Web elimina a necessidade de configuração de ambiente local, democratizando o acesso prático aos conceitos de computação de alto desempenho apresentados em sala de aula.

2. A Arquitetura Didática do Oldowan

O principal desafio no ensino da arquitetura MIPS é ilustrar modelos conceituais abstratos. O simulador Oldowan ataca essa dificuldade dividindo a experiência do usuário em módulos funcionais que espelham o *datapath* de um processador real, conforme ilustrado na Figura 1. Sua interface é projetada para ser interativa e expositiva, centrada nos fenômenos cruciais para o entendimento da máquina.

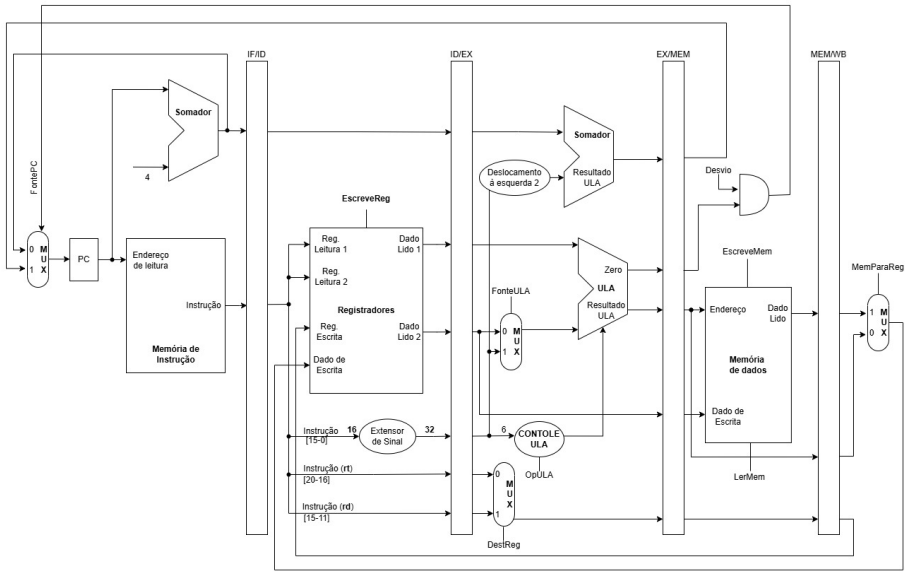

Figura 1. Diagrama do *datapath* da arquitetura MIPS

2.1. Execução em Pipeline e Interação

Diferente de simuladores monociclo, que focam apenas no estado da memória e dos registradores pré e pós-execução, o Oldowan simula a sobreposição temporal da execução das instruções. Um dos grandes diferenciais do simulador é a visualização explícita dos cinco estágios clássicos do *pipeline* MIPS: *Instruction Fetch* (IF), *Instruction Decode* (ID), *Execution* (EX), *Memory Access* (MEM) e *Write Back* (WB).

O simulador fornece uma tabela dinâmica que ilustra conceitualmente o conteúdo dos *latches* (registradores intermediários) a cada ciclo de *clock*. Essa visualização permite que o aluno acompanhe não apenas a evolução do Contador de Programa (PC), mas também a propagação dos sinais de controle e dados da Unidade Lógico-Aritmética (ULA) estágio por estágio. Ao utilizar o modo de execução passo a passo (“Step”), fenômenos complexos, como o surgimento de bolhas (*stalls*) devido a dependências de dados ou a

ação do *branch delay slot*, são evidenciados em tempo real, transformando a teoria em uma observação tangível.

2.2. Organização de Memória e Caches

O entendimento da hierarquia de memória é outro gargalo no estudo de arquitetura, devido à natureza transparente das *caches* L1 e L2 para o montador Assembly. No Oldowan, a memória principal e o sistema de cache foram modelados para fornecer *feedback* visual imediato.

A ferramenta conta com painéis distintos para a Cache de Instruções (I-Cache) e a Cache de Dados (D-Cache), modeladas como memórias associativas por conjunto temporárias. Ao executar operações de Load ou Store, ou mesmo durante a busca de uma instrução (IF), o aluno pode observar a modificação das *tags*, bits de validade e a atualização do estado baseada na política de substituição LRU (*Least Recently Used*). Essa abordagem elimina camadas de abstração e permite que os alunos observem a mecânica dos acertos (*hits*) e faltas (*misses*) de cache de forma clara e isolada.

2.3. Introdução à Programação de Baixo Nível

Um aspecto de grande importância da ferramenta é sua contribuição para o aprendizado da linguagem Assembly. Para operar o *pipeline* simulado, o aluno deve escrever programas em MIPS32. O Oldowan conta com um editor de texto e um montador integrado no próprio navegador, que atua em duas passadas, gerando código de máquina compatível com o simulador.

A capacidade de escrever, em uma única interface, o conjunto reduzido de instruções, cobrindo operações lógicas, de manipulação aritmética, desvios e acesso à memória, facilita o ciclo de experimentação. Essa imersão prática nos fundamentos do Assembly favorece o entendimento sobre como lógicas de alto nível (estruturas de repetição, chamadas de sub-rotina) são mapeadas para o hardware, reforçando conceitos abordados em livros de referência [Hennessy and Patterson 2014, Stallings 2017]. Adicionalmente, o sistema lida de forma autônoma com pseudoinstruções, abstraindo limitações do hardware, suavizando a curva de aprendizado para estudantes habituados a linguagens imperativas.

2.4. Relevância Acadêmica e Acessibilidade

A aplicação do Oldowan em ambientes acadêmicos ressalta a importância da experimentação prática. Ao evidenciar o fluxo interno do hardware, a arquitetura de software oferece um cenário didático que constrói a fundação necessária para os estudantes enfrentarem desafios mais avançados, como paralelismo e multiprocessamento escalável.

O ambiente baseado em Web não requer *setup* de dependências nos computadores dos laboratórios institucionais e viabiliza a continuação dos estudos remotos de forma simplificada em qualquer plataforma. Tais ferramentas interativas mostram-se cruciais para o ensino da arquitetura de computadores contemporânea, ajustando-se à adoção natural de plataformas conectadas (*cloud-based*) no meio acadêmico.

3. Implementação

O simulador Oldowan foi implementado adotando apenas tecnologias abertas nativas da *World Wide Web* com o objetivo principal de maximizar o seu alcance didático sem sacrificar a fidelidade arquitetural.

O projeto dispensa servidor, realizando toda a avaliação de código no lado do cliente utilizando *scripts* em **JavaScript** de execução rápida, estruturados sobre marcações **HTML5** e estilizados via **CSS3**. O HTML define áreas como a visualização gráfica do *pipeline* e representações de registradores hexadecimais de rápida leitura.

A *engine* de simulação do Oldowan foi estruturada modularmente: um interpretador cuida de decodificar mnemônicos em código de máquina, enquanto as funções lógicas do *datapath* imitam rigorosamente os relógios ("*clocks*") da CPU real. A lógica da execução do *pipeline* foi escrita com chamadas matemáticas reversas (para impedir a leitura de uma informação incompleta pelo estágio a seguir em um ciclo único), o que possibilita simular, com precisão, instâncias de concorrência ou paradas de processamento em cada janela de instrução, tudo isso encapsulado em um objeto executável por qualquer navegador nativo atualizado.

O projeto segue ativo em processos de documentação e refinamento no repositório local e almeja a implantação livre em redes universitárias abertas.

4. Conclusão

O simulador Web Oldowan desenvolvido neste projeto mostrou-se uma ferramenta com enorme potencial prático e visual para o ensino da moderna arquitetura MIPS. Ao permitir que os estudantes programem e depurem lógicas computacionais puras de banco de memória sem abandonar o navegador de uso diário de internet, o simulador mitiga as barreiras sistêmicas iniciais, estimulando a experimentação acadêmica e solidificando conceitos que anteriormente repousavam isolados sob representações em livros-texto estáticos. Este protótipo, em contínuo desenvolvimento, pode ser acessado via `https://jaosoares2.github.io/Oldowan/`.

A clareza na exposição do *pipeline* MIPS e das políticas de acesso da hierarquia dos métodos de cache consolida sua utilidade em abordagens do ciclo didático. Nesse contexto, a ferramenta permite exemplificar, na prática, como a ordenação das instruções em um programa pode evitar conflitos e afetar diretamente o desempenho da execução. Entre as possibilidades de melhorias e expansões futuras, destacam-se a possível incorporação de lógicas para representação animada vetorial (*datapath* visual e interativo do hardware, ligando fios e portas lógicas na tela) e até o aprimoramento da análise algorítmica estatística de erros de cache. Com esta iniciativa de tecnologia aberta, colabora-se formalmente para a acessibilidade do entendimento fundamental computacional na nova geração acadêmica.

Referências

Hennessy, J. L. and Patterson, D. A. (2014). *Organização e projeto de computadores: a interface hardware/software*, volume 4. Elsevier Brasil.

Stallings, W. (2017). Arquitetura e organização de computadores. Technical report, Pearson.

Vollmar, K. and Sanderson, P. (2006). Mars: an education-oriented mips assembly language hardware simulator. *Journal of Computing Sciences in Colleges*, 21(4):239–246.

Weber, R. F. (2009). *Fundamentos de Arquitetura de Computadores-Vol. 8: Série Livros Didáticos Informática UFRGS*. Bookman Editora.