

Simulador da arquitetura Neander em uma abordagem Web

João P. R. Linares¹, Gerson Geraldo H. Cavalheiro¹

¹Programa de Pós-Graduação em Computação
Universidade Federal de Pelotas (UFPel)
R. Gomes Carneiro, 01 - Balsa, Pelotas - RS, 96010-610

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

Resumo. *Esse artigo descreve o desenvolvimento de um simulador interativo baseado na arquitetura Neander, aplicado com tecnologias web. O projeto tem como objetivo proporcionar uma introdução acessível e intuitiva aos fundamentos de arquitetura de computadores e programação de baixo nível, especialmente voltado para estudantes no início da formação, baseando-se em ferramentas preexistentes e aprimorando-as. Ao oferecer uma experiência interativa no ambiente web, o simulador busca facilitar o aprendizado prático desses conceitos fundamentais.*

1. Introdução

O ensino da arquitetura de computadores representa um desafio significativo para estudantes que estão nos estágios iniciais da computação, parcialmente devido à abstração das linguagens de alto nível, que distancia os alunos dos mecanismos fundamentais do funcionamento de um computador. Nesse cenário, a arquitetura Neander se apresenta como uma alternativa didática valiosa, ao simplificar os conceitos básicos de um processador, tais como registradores, conjunto de instruções e memória, sem perder de vista elementos essenciais para a compreensão dos sistemas computacionais. Como exposto pelo professor Raul Fernando Weber em *Fundamentos de Arquitetura de Computadores* [Weber 2009], a utilização de uma arquitetura simplificada permite transmitir de forma mais clara os fundamentos que controlam o funcionamento dos computadores.

A demanda por ferramentas de apoio que facilitem o aprendizado prático desses conceitos tem impulsionado o desenvolvimento de simuladores interativos. Exemplos como o NeanderWin [Borges and Silva 2006], um simulador interativo didático já utilizado no ensino, demonstram a eficácia desse recurso. Tais ferramentas eliminam barreiras técnicas, permitindo que os alunos experimentem a execução de programas sem a necessidade de instalação de softwares complexos.

Esse trabalho propõe o desenvolvimento de um simulador interativo de arquitetura Neander, utilizando tecnologias de desenvolvimento web. Essa abordagem visa proporcionar um ambiente prático e intuitivo, no qual os usuários podem escrever e executar programas diretamente via navegador. Aprimorando os simuladores existentes e integrando os fundamentos teóricos destacados por Weber com uma interface moderna e acessível, espera-se não somente modernizar os recursos didáticos disponíveis, mas também fortalecer o aprendizado de programação de baixo nível e de conceitos essenciais da arquitetura de computadores de maneira dinâmica e envolvente.

Essa iniciativa, ancorada tanto na literatura quanto em experiências prévias com outros simuladores, contribui para aproximar o ensino teórico da prática experimental,

facilitando o acesso aos conhecimentos fundamentais e promovendo uma aprendizagem mais efetiva.

2. Fundamentação Teórica

A arquitetura Neander se mostra uma ferramenta didática poderosa e muito importante para a compreensão dos conceitos fundamentais de arquiteturas de computadores e programação em linguagem de baixo nível. Ao simplificar os mecanismos internos dos processadores, esta arquitetura permite que os estudantes se familiarizem de forma direta e intuitiva com os conceitos fundamentais que regem o funcionamento de um computador. A simplificação de conceitos complexos e a utilização de modelos práticos, como o Neander, contribuem significativamente para a compreensão do funcionamento básico dos computadores e de seu suporte à execução de programas, estabelecendo uma base sólida para estudos mais avançados.

2.1. Componentes e Funcionamento

A estrutura da arquitetura Neander é constituída por unidades elementais que colaboram para uma compreensão fácil do processamento interno. O acumulador (AC) funciona como o único registrador de uso geral, armazenando os resultados das operações realizadas pela Unidade Lógica e Aritmética (ULA). A ULA é responsável por executar operações aritméticas e lógicas, atuando diretamente sobre os dados contidos no acumulador. O apontador de programa (PC) organiza a sequência de execução das instruções, enquanto o registrador de estado sinaliza as condições resultantes das operações, indicando a ocorrência de valores nulos ou negativos. A memória, geralmente limitada a 256 bytes, é utilizada tanto para o armazenamento das instruções quanto para os dados, configurando um ambiente previsível para a simulação dos processos computacionais. Essa configuração, demonstrada também no NeanderWin [Borges and Silva 2006], elimina camadas de abstração e permite que os alunos acompanhem cada etapa do processamento com clareza.

2.2. Conjunto de Instruções e Processamento

Um dos grandes diferenciais da arquitetura Neander reside em seu conjunto de instruções, caracterizado pela sua simplicidade. O conjunto de instruções inclui operações de carregamento e armazenamento de dados, operações lógicas e aritméticas, desvios condicionais e instruções de parada, cobrindo assim o ciclo completo de busca e execução de programas. Essa execução dos processos possibilita uma visualização detalhada do fluxo de dados e da manipulação dos registradores, como já ressaltado em outros simuladores Neander. Além disso, a utilização de simuladores interativos reforça a eficácia desse conjunto reduzido de instruções, permitindo que os alunos experimentem de forma prática a execução de algoritmos.

2.3. Introdução à Programação de Baixo Nível

Outro aspecto de grande importância da arquitetura Neander é sua contribuição para a programação de baixo nível. Por operar com instruções diretamente representadas em código de máquina, a arquitetura proporciona um contato com a forma mais elementar de escrita e execução de programas. Essa imersão nos fundamentos do assembly favorece a compreensão dos processos de montagem e execução, além

do entendimento dos mecanismos essenciais aos compiladores e sistemas operacionais. Tal abordagem é corroborada em livros como *Arquitetura e Organização de Computadores* [Stallings 2017] e *Organização e Projeto de Computadores: a Interface Hardware/software* [Hennessy and Patterson 2014], que destacam o uso de linguagens de baixo nível, como assembly, a fim de explorar as arquiteturas de sistemas computacionais, alinhando-se aos fundamentos apresentados pela arquitetura Neander.

2.4. Relevância Acadêmica e Aplicações Práticas

A aplicação do modelo Neander em ambientes acadêmicos ressalta a importância da experimentação prática na construção do conhecimento. Ao evidenciar cada etapa da execução computacional, a arquitetura oferece um cenário didático que esclarece os conceitos básicos e prepara os estudantes para desafios mais avançados na área. O ambiente Neander incentiva a elaboração de algoritmos eficientes em um contexto controlado, como demonstrado em outros simuladores já implementados, como o NeanderSIM [Ullmann et al. 2014]. Tais ferramentas interativas mostram-se cruciais para o ensino da arquitetura de computadores, pois consolidam o entendimento dos conceitos fundamentais.

A arquitetura Neander é um instrumento essencial para a compreensão dos princípios computacionais e um meio eficaz para introduzir os alunos à programação de baixo nível. A integração de referências clássicas [Weber 2009] e [Borges and Silva 2006], com iniciativas contemporâneas, como o NeanderSIM [Ullmann et al. 2014], destaca a relevância dessa abordagem didática, ajustando as demandas atuais do ensino e preparando os estudantes para desafios futuros.

3. Implementação

O simulador desse projeto, o NeanderWeb, foi implementado utilizando tecnologias web, com o objetivo de oferecer uma ferramenta intuitiva e amplamente acessível. Sua estrutura principal é baseada em HTML, CSS e JavaScript. O HTML define a estrutura da interface, incluindo a área de edição do código, botões e exibição do programa e da memória. O CSS é responsável por tornar a interface mais intuitiva e visualmente agradável, melhorando a experiência do usuário; já o JavaScript implementa toda a lógica do simulador, lidando com a interpretação das instruções, manipulação da memória e interação com a interface. Essa escolha permite que o simulador seja executado diretamente em navegadores modernos, tornando-o acessível em qualquer dispositivo relativamente recente, sem necessidade de configurações complexas ou instalação de software.

A estrutura do simulador é organizada em três componentes principais: o editor de código, o mecanismo de execução e a exibição gráfica da memória. No editor, os usuários escrevem programas na linguagem assembly, que são processados pelo mecanismo de execução. Este interpreta as instruções, atualiza o acumulador, o contador de programa e os registradores, e executa as operações necessárias. Durante a execução, a interface exibe em tempo real as alterações na memória, permitindo que os usuários acompanhem passo a passo as mudanças e compreendam melhor o funcionamento interno. Além disso, o simulador possibilita salvar e carregar o estado da memória, possibilitando a retomada da execução em momentos distintos e aprimorando a experiência do usuário.

Uma funcionalidade importante do simulador é o modo de execução passo a passo, que permite avançar uma instrução por vez, destacando a instrução atual na interface e

exibindo as mudanças nos registradores e na memória, oferecendo uma experiência mais didática e detalhada, essencial para o aprendizado de conceitos fundamentais.

Outros diferenciais incluem um editor de código simplificado, que centraliza a escrita e edição dos programas em uma única área, e a opção de exibir os valores da memória em decimal ou hexadecimal, atendendo às diferentes necessidades dos usuários. Essas funcionalidades garantem praticidade no uso do simulador, tornando-o atrativo e funcional.

Ainda planeja-se estender o simulador, incluindo suporte a instruções vetoriais, permitindo explorar conceitos associados à exploração da concorrência, aproximando o simulador de arquiteturas modernas. Com essas melhorias, espera-se ampliar ainda mais a utilidade da ferramenta no ensino de arquitetura de computadores, trazendo um ambiente moderno e interativo para o aprendizado de programação de baixo nível. O projeto ainda se encontra em desenvolvimento e, futuramente, planeja-se disponibilizar um site e um repositório online a fim de facilitar o acesso e o uso da ferramenta.

4. Conclusão

O simulador desenvolvido neste projeto mostrou-se uma ferramenta eficaz para o ensino da arquitetura de computadores, oferecendo uma abordagem interativa ao aprendizado de programação de baixo nível. Ao permitir que estudantes executem programas diretamente no navegador, o NeanderWeb elimina barreiras técnicas, incentivando a experimentação prática e a compreensão dos princípios fundamentais da computação. A implementação baseada em tecnologias web garante ampla acessibilidade, facilitando a adoção em ambientes acadêmicos.

Entre as possibilidades de melhorias e expansões futuras estão o suporte a instruções vetoriais, permitindo explorar conceitos de processamento paralelo e otimização computacional, além de melhorias na interface, como um depurador mais avançado e visualizações mais detalhadas do estado da execução. O projeto permanece aberto a aprimoramentos, contribuindo para o ensino de computação de maneira moderna, interativa e acessível.

Referências

- Borges, J. A. S. and Silva, G. P. (2006). Neanderwin-um simulador didático para uma arquitetura do tipo acumulador. In *workshop sobre educação em arquitetura de computadores*.
- Hennessy, J. and Patterson, D. (2014). *Organização e Projeto de Computadores: A Interface Hardware/Software*. Elsevier Brasil.
- Stallings, W. (2017). *Arquitetura e Organização de Computadores*. Pearson Universidades.
- Ullmann, M., Inocêncio, A. C., Neto, E. M., Freitas, M., and Júnior, P. A. P. (2014). Neandersim: Simulador gráfico de apoio ao ensino de arquitetura de computadores. In *Anais do XXII Workshop sobre Educação em Computação*, pages 361–370, Porto Alegre, RS, Brasil. SBC.
- Weber, R. (2009). *Fundamentos de Arquitetura de Computadores - Vol.8: Série Livros Didáticos Informática UFRGS*. Bookman.