

## DESENVOLVIMENTO DE UM SIMULADOR WEB PARA O ENSINO DE ARQUITETURA DE COMPUTADORES

JOÃO PEDRO RODRIGUES LINARES<sup>1</sup>; GERSON GERALDO H. CAVALHEIRO<sup>1</sup>

<sup>1</sup>*Universidade Federal de Pelotas – {jprlinares, gerson.cavalheiro}@inf.ufpel.edu.br*

### 1. INTRODUÇÃO

O ensino de arquitetura de computadores é um dos principais desafios nos cursos de computação. À medida que os sistemas se tornam mais poderosos e complexos, as camadas de abstração, como as linguagens de alto nível, distanciam os estudantes dos mecanismos fundamentais de execução que ocorrem no processador (CPU). Essa barreira é crucial para formar profissionais que compreendam as implicações de suas decisões de software no hardware subjacente. Uma estratégia pedagógica para essa finalidade é o uso de simuladores didáticos interativos, que permitem a experimentação direta em ambientes visuais sem exigir a instalação de softwares complexos. Nesse contexto, a arquitetura Neander (WEBER, 2000) se estabeleceu como um padrão por seu modelo simples, que contém os componentes essenciais de um processador e facilita a compreensão inicial dos conceitos de ciclo de instrução e memória.

Ainda que cumpram seu papel, os simuladores didáticos tradicionais raramente abordam conceitos de paralelismo, cada vez mais presentes nas arquiteturas modernas. Ferramentas clássicas, como as baseadas na arquitetura LMC (YURCIK, 2001), focam na simplicidade, mas não são suficientes para introduzir ao aluno as fontes de desempenho das máquinas atuais. Por outro lado, ferramentas mais complexas, baseadas em MIPS ou outras arquiteturas customizáveis (SKRIEN, 2001), podem sobrecarregar alunos iniciantes com uma complexidade que ofusca os conceitos essenciais. O desafio é conectar o aprendizado fundamental com os conceitos de paralelismo em nível de hardware, como caches e instruções vetoriais, que permanecem pouco explorados em disciplinas introdutórias, em parte pela ausência de ferramentas adequadas. Iniciativas nacionais como o SimuS (SILVA, 2016), apesar de terem evoluído da arquitetura Neander, ainda deixam a introdução ao paralelismo de dados e à hierarquia de memória de forma simples e integrada como um campo a ser explorado.

Este trabalho tem como objetivo apresentar o NeanderWeb-V, um simulador web (LINARES, 2025) para a arquitetura hipotética Neander-V, que busca preencher a lacuna entre a simplicidade e a contemporaneidade. A proposta estende o modelo original do Neander, adicionando suporte a operações vetoriais, para introduzir o conceito de paralelismo de dados (SIMD), e a simulação de cache, a fim de materializar a ideia de hierarquia de memória. O NeanderWeb-V foi projetado para apoiar o ensino introdutório de arquitetura de computadores, sendo acessível diretamente em navegadores. Seu principal destaque é a exposição de mecanismos de paralelismo em nível de hardware, o que permite que estudantes escrevam e executem programas enquanto visualizam, em tempo real, o impacto de recursos paralelos no desempenho. Desta forma, a ferramenta ensina os conceitos fundamentais da arquitetura de

computadores, e também contribui para o entendimento das arquiteturas de alto desempenho do mundo real.

## 2. METODOLOGIA

O trabalho trouxe a concepção teórica da arquitetura estendida Neander-V, e também a implementação prática de seu simulador, o NeanderWeb-V. Essa abordagem busca garantir que cada decisão de projeto seja alinhada com os objetivos pedagógicos de facilitar o aprendizado de conceitos computacionais.

A arquitetura Neander-V foi projetada como uma evolução do Neander original, porém buscando preservar a sua simplicidade, e a compatibilidade com o conjunto de instruções base, mantendo a arquitetura com um acumulador (AC), um contador de programa (PC), uma memória de 256 bytes e flags de estado (Negativo e Zero), a primeira extensão foi a introdução do paralelismo de dados. Feito através de um registrador vetorial, o VAC (Vector Accumulator), capaz de armazenar um vetor de quatro elementos de 8 bits. Um novo conjunto de instruções vetoriais foi então adicionado para operar sobre o VAC, permitindo a execução de uma única instrução sobre múltiplos dados (SIMD) e servindo como uma introdução simplificada a este pilar de ensino da arquitetura de computadores.

A segunda extensão foi a introdução da simulação de uma hierarquia de memória. Para isso, foram adicionados um cache de instruções (I-Cache) de 8 bytes e um cache de dados (D-Cache) de 16 bytes. A política de escrita adotada para a D-Cache é write-through com alocação no-write-allocate, escolhida para simplificar o modelo de consistência e focar a atenção do estudante no fenômeno da localidade de referência. O impacto disso é destacado na interface do simulador, que exibe dinamicamente os contadores de hits e misses, permitindo ao aluno quantificar e observar o ganho de desempenho dos algoritmos.

O simulador foi desenvolvido utilizando tecnologias web (HTML, CSS e JavaScript), o que permite sua execução em qualquer navegador moderno, sem a necessidade de instalação, e removendo barreiras de acesso para estudantes e educadores. A ferramenta oferece um ambiente de desenvolvimento integrado e visual. A Figura 1 demonstra a interface principal do simulador, que é organizada em painéis lógicos para fornecer feedback visual de forma clara para o usuário.

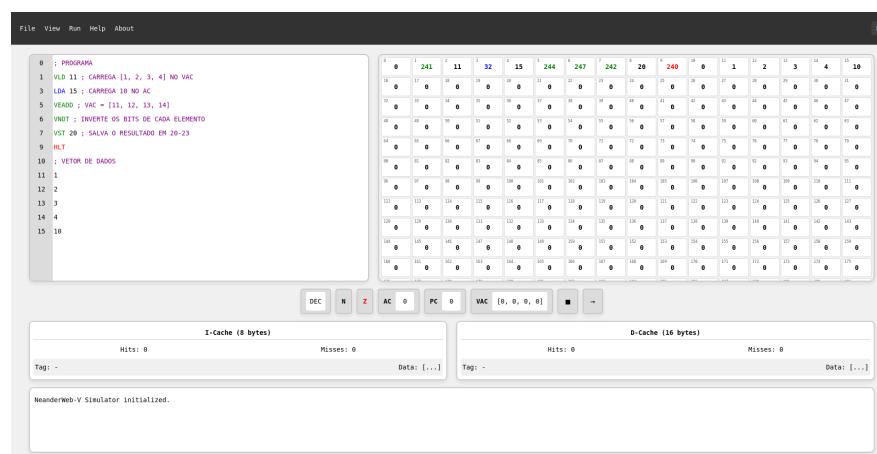


Figura 1 – Interface do NeanderWeb-V

A interface permite a escrita do código em um editor à esquerda, com realce de sintaxe, e a visualização da memória de 256 bytes em uma matriz à direita. A execução pode ser contínua ou passo a passo, que permite ao usuário visualizar em detalhes a execução do código. A visualização do estado de todos os componentes internos é dinâmica. Os painéis inferiores exibem os registradores (AC, PC, VAC), as flags, e o estado dos caches, com as células de memória e as linhas de cache sendo destacadas durante o acesso. A interface também inclui funcionalidades de conveniência, como suporte multilíngue (português e inglês), a opção de carregar e salvar arquivos, assim como permite ao usuário mudar o modo de visualização entre decimal e hexadecimal.

### 3. RESULTADOS E DISCUSSÃO

A aplicabilidade da arquitetura Neander-V foi avaliada por meio de exemplos práticos e um estudo de usabilidade com estudantes da computação. Primeiramente, demonstrou-se a expressividade da nova arquitetura com um programa que inverte o sinal de um vetor de 4 elementos usando a operação de complemento de 2. O código realiza a operação eficientemente com poucas instruções: carregamento do vetor para o VAC (VLD), inversão bit a bit de todos os elementos (VNOT), carregamento do valor escalar 1 no AC (LDA), soma deste valor a todos os elementos do VAC (VEADD) e, por fim, armazena o resultado na memória (VST).

Para a análise quantitativa, foi comparada a implementação da soma de um valor escalar a um vetor, contrastando o código Neander original com a versão vetorial Neander-V. A versão vetorial demonstrou uma redução expressiva no tamanho do código, passando de 13 para apenas 5 linhas. Esse ganho se traduziu diretamente em desempenho de acessos à memória, a execução do código vetorial exigiu apenas 11 acessos, enquanto a versão escalar necessitou de 37. Adicionalmente, a simulação permitiu demonstrar o impacto do alinhamento de dados no cache. Em um cenário com dados desalinhados, a versão escalar teve seu desempenho drasticamente reduzido, com o número de misses na D-Cache saltando de 1 para 20. A versão vetorial sofreu um impacto mínimo, com os misses aumentando apenas de 1 para 2.

Foi conduzido um estudo de usabilidade com 28 estudantes do primeiro semestre de Engenharia de Computação, na disciplina de Introdução à Engenharia de Computação, durante uma atividade prática. Ao final, esses estudantes responderam a um questionário. Os resultados foram muito positivos: 86% consideraram o simulador fácil de usar e que suas funcionalidades estavam bem integradas, e apenas 5% o acharam desnecessariamente complexo. Houve um contraste interessante nas respostas: apesar da alta percepção de facilidade, 14% indicaram que precisam de alguma ajuda técnica. A hipótese levantada a partir disso é que a complexidade percebida por uma minoria não está na interface da ferramenta, mas na necessidade de compreender os conceitos da própria arquitetura Neander-V para utilizá-la plenamente, o que está alinhado com seus objetivos pedagógicos.

### 4. CONCLUSÕES

Este artigo apresentou o NeanderWeb-V, um simulador que estende a clássica arquitetura didática Neander com suporte a vetorização e simulação de

cache, conceitos fundamentais em sistemas computacionais modernos. A principal inovação do trabalho é oferecer um ambiente interativo e acessível via web que integra os conceitos básicos e avançados de arquitetura, e permitindo que estudantes experimentem a ferramenta diretamente no navegador, eliminando barreiras técnicas. A ferramenta se mostrou eficaz em facilitar o entendimento de conceitos abstratos, como o paralelismo de dados e a hierarquia de memória. Os resultados quantitativos reforçam seu valor pedagógico, ao demonstrar de forma clara e mensurável os ganhos de desempenho obtidos com as extensões vetoriais.

Como trabalhos futuros, o projeto tem um roteiro que inclui melhorias de interface baseadas no feedback dos usuários, como facilitar a navegação em posições de memória distantes e ajustar o comportamento do editor de código. Busca-se incluir a visualização do pipeline de execução, a fim de representar explicitamente o paralelismo em nível de instrução. Também é planejado a expansão da capacidade da arquitetura, com a introdução de suporte a laços e sub-rotinas, o que demandará a criação de novos componentes e instruções, como registradores adicionais (como um segundo registrador vetorial), acesso indireto à memória e instruções do tipo CALL/RET, com suporte de um registrador de pilha. Para manter a proposta didática e minimalista, considera-se que estas novas funcionalidades sejam implementadas em camadas, permitindo que o usuário habilite os recursos de forma incremental, conforme os objetivos de ensino. Por fim, projeta-se, a longo prazo, desenvolver uma versão multiprocessada (multicore) da arquitetura, permitindo o estudo de estratégias de escalonamento de threads.

## 5. REFERÊNCIAS BIBLIOGRÁFICAS

- LINARES, J. P.; CAVALHEIRO, G. G. H. Simulador da arquitetura neander em uma abordagem web. In: Escola Regional de Alto Desempenho da Região Sul, p. 113-116, 2025.
- SILVA, G. P.; BORGES, J. A. d. S. Simus: Um simulador para o ensino de arquitetura de computadores. International Journal of Computer Architecture Education, v. 5, n. 1, 2016.
- SKRIEN, D. CPU Sim 3.1: A tool for simulating computer architectures for computer organization classes. Journal on Educational Resources in Computing, v. 1, n. 4, p. 46-59, 2001.
- WEBER, R. F. Fundamentos de arquitetura de computadores. Porto Alegre: Sagra Luzzatto, 2000.
- YURCIK, W.; BRUMBAUGH, L. A web-based little man computer simulator. In: Proceedings of the 32nd Technical Symposium on Computer Science Education, p. 204-208, 2001.