

# NeanderWeb-V: Uma Abordagem Web Simplificada para o Ensino de Paralelismo e Cache

João Pedro R. Linares<sup>1</sup>, João V. P. Soares<sup>1</sup>, Gerson Geraldo H. Cavalheiro<sup>1</sup>

<sup>1</sup>Bacharelado em Engenharia de Computação  
Universidade Federal de Pelotas  
R. Gomes Carneiro, 01 – Pelotas – RS/Brasil – 96010-610

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

**Resumo.** *Este trabalho apresenta o NeanderWeb-V, um simulador da arquitetura Neander-V, que amplia o Neander clássico, com suporte a vetorização e cache. Desenvolvido para execução direta em navegadores, o sistema oferece um ambiente interativo para explorar conceitos fundamentais de Arquitetura de Computadores e Programação. O diferencial está na possibilidade de visualizar mecanismos de paralelismo em hardware, permitindo compreender na prática o impacto desses recursos no desempenho do software. Um estudo de caso com estudantes de Computação avaliou a usabilidade do sistema, indicando sua viabilidade de uso e potencial pedagógico, embora estudos adicionais sejam necessários para medir sua eficácia educacional de forma quantitativa.*

## 1. Introdução

O ensino de arquitetura de computadores mostra-se um desafio significativo nos cursos de computação, devido à abstração das linguagens de alto nível, que distanciam os estudantes dos processos fundamentais de uma CPU. Uma estratégia para superar essa barreira no ensino é o uso de simuladores baseados em arquiteturas simplificadas.

Por conta da simplicidade, simuladores didáticos simples nem sempre buscam abordar conceitos de paralelismo e hierarquia de memória, conceitos importantes que são encontrados em praticamente todos os processadores modernos. Entretanto, ferramentas muito elaboradas dificultam a introdução em disciplinas iniciais, criando barreiras de entrada elevadas para novos estudantes.

Esse trabalho apresenta a arquitetura Neander-V e seu simulador web, o NeanderWeb-V<sup>1</sup>. Desenvolvido utilizando tecnologias web e já consolidado anteriormente [Linares and Cavalheiro 2025, Linares et al. 2025], o simulador é uma alternativa didática para introduzir conceitos de paralelismo e hierarquia de memória em cursos introdutórios, unindo a didática simples da arquitetura Neander [Weber 2000] à introdução de conceitos modernos. Sendo acessível em qualquer navegador moderno, a arquitetura elimina barreiras de instalação, permitindo que os alunos explorem esses conceitos de forma interativa.

## 2. Trabalhos Relacionados

Os simuladores educacionais abrangem um espectro que vai de modelos minimalistas, como o LMC (Little Man Computer) [Yurcik et al. 2001], a plataformas complexas baseadas na arquitetura MIPS, como MARS [Vollmar et al. 2006] e DrMIPS

---

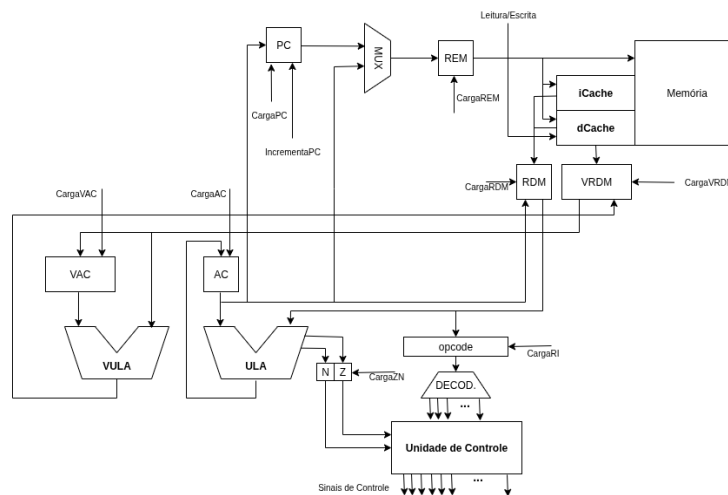
<sup>1</sup>Disponível em: <https://neanderweb.github.io>.

[Nova et al. 2013]. Enquanto simuladores mais simples raramente oferecem um caminho direto para conceitos modernos, outros possuem uma complexidade que representa uma barreira de entrada elevada para iniciantes. Como alternativa para facilitar o acesso e simplificar o uso, ferramentas executadas diretamente no navegador, como o Wepsim [García et al. 2019], tem se tornado cada vez mais comuns.

No contexto brasileiro, a arquitetura Neander destaca-se como uma iniciativa pioneira, auxiliando o ensino dos conceitos fundamentais dos processadores por sua simplicidade. Evoluções como o NeanderWin [Borges and Silva 2006] e o SimuS [Silva and Borges 2016] buscaram superar as limitações do modelo original, expandindo modos de endereçamento e conjuntos de instruções. O NeanderWeb-V ocupa um espaço pouco explorado, buscando preservar a clareza didática do Neander ao mesmo tempo que introduz vetorização e cache de forma simples.

### 3. Arquitetura Neander-V

A arquitetura Neander-V, ilustrada na Figura 1, foi desenvolvida como uma evolução do processador Neander clássico. Ela busca preservar a compatibilidade e a simplicidade da versão original, ao mesmo tempo em que introduz conceitos ligados ao paralelismo em nível de hardware. A base da arquitetura foi projetada com propósitos educacionais, fornecendo uma configuração mínima ideal para facilitar o entendimento de como um processador funciona. Essa base se mantém no modelo Neander-V, o qual dispõe de um único registrador de dados principal: o acumulador (AC), responsável por guardar os resultados gerados pela Unidade Lógica e Aritmética (ULA). O controle da sequência de execução fica a cargo do apontador de programa (PC). Além disso, a memória total de 256 bytes é dividida de forma compartilhada entre dados e instruções. As *flags* N (negativo) e Z (zero) são encarregadas de sinalizar o estado atual das operações.



**Figura 1. Arquitetura Neander-V.**

A alteração mais importante em relação à arquitetura Neander original é o suporte a operações com vetores, o que expõe o paralelismo de dados. Para isso, o sistema incorpora um registrador específico chamado VAC (*Vector Accumulator*), que consegue armazenar simultaneamente um vetor composto por 4 elementos de 8 bits. Foi adicionado

também um novo conjunto de instruções com suporte ao processamento vetorial focado em manipular os dados presentes nesse VAC. Tais instruções (como VLD para carregamento, VST para armazenamento e operações como VADD e VEADD) viabilizam a execução simultânea de uma única operação sobre múltiplos dados (SIMD), ilustrando um conceito-chave das arquiteturas contemporâneas.

A outra adição a ferramenta é a simulação de memória cache, permitindo aos alunos explorarem na prática a hierarquia de memória e a localidade de referência. O simulador implementa uma cache voltada para instruções (I-Cache) com 8 bytes de capacidade e uma cache para dados (D-Cache) com 16 bytes. Com o intuito de manter a simplicidade didática, cada uma dessas caches possui apenas uma linha, utilizando o mapeamento direto. No caso da D-Cache, adota-se a política de escrita *write-through* (garantindo a consistência imediata com a memória principal), e a alocação de blocos acontece somente durante falhas de leitura (*no-write-allocate*). Por fim, a interface web exibe de modo dinâmico os contadores de *hits* e *misses*, entregando um feedback visual instantâneo a respeito da eficiência e do desempenho do código executado.

#### 4. Implementação e Avaliação

O NeanderWeb-V foi implementado utilizando HTML5, CSS3 e JavaScript (ES6), garantindo sua execução direta em navegadores modernos sem necessidade de instalação. A interface exibe editores de texto com realce de sintaxe, grade de memória 16x16, status dos registradores (PC, AC, VAC) e contadores de cache, além de permitir controle de execução passo a passo.

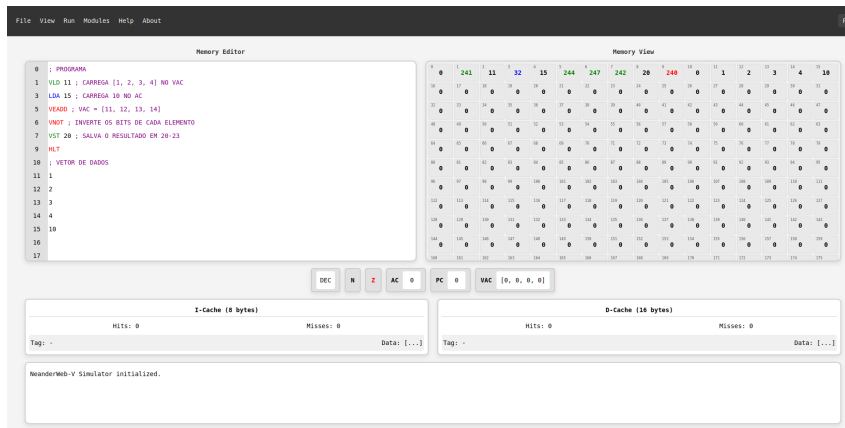


Figura 2. Interface do NeanderWeb-V.

A Figura 2 apresenta uma visão da interface do simulador. A janela à esquerda permite a edição do programa, e a da direita, a visualização da memória. Na parte inferior, é possível visualizar os registradores e a memória cache.

Para avaliar o uso da ferramenta, realizou-se um teste com alunos da disciplina de Introdução à Ciência da Computação (UFPEL), focando no ganho de desempenho obtido com a vetorização. A avaliação da interface utilizou o System Usability Scale (SUS) [Brooke 1996] com 15 estudantes, obtendo-se um índice global de 57,5.

Esse valor é interpretado de forma crítica neste trabalho, embora a análise tenha mostrado médias elevadas em simplicidade e consistência visual, evidenciando que os

alunos consideraram a interface clara e fácil de usar. Contudo, houve pontuações mais baixas na percepção de facilidade de aprendizado e confiança no uso, sugerindo que as dificuldades decorrem do domínio inicial da arquitetura, indicando a necessidade futura de estratégias de suporte integrado na ferramenta.

## 5. Conclusão e Trabalhos Futuros

O NeanderWeb-V apresenta-se como uma ferramenta de apoio ao ensino introdutório de Arquitetura de Computadores, eliminando barreiras técnicas e facilitando a experimentação com paralelismo de dados e hierarquia de memória. A capacidade de medir ganhos de desempenho torna conceitos abstratos mais concretos para os estudantes.

Trabalhos futuros tem como objetivo buscar o aprimoramento da interface para apoiar o aprendizado inicial e expandir a arquitetura em camadas. Estão planejadas a inclusão da visualização do pipeline de instruções, aumento da memória e implementação de um sistema de sub-rotinas, mantendo a flexibilidade e a simplicidade para as fases iniciais da formação em Computação.

## 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*.
- Brooke, J. (1996). Sus: A “quick and dirty” usability scale. In Jordan, P. W., Thomas, B., McClelland, I. L., and Weerdmeester, B., editors, *Usability Evaluation in Industry*, pages 189–194. Taylor & Francis.
- García, F., Calderón-Mateos, A., Alonso-Monsalve, S., and Prieto-Cepeda, J. (2019). WepSIM: an online interactive educational simulator integrating microdesign, micro-programming, and assembly language programming. *IEEE Transactions on Learning Technologies*, 13(1):211–218.
- Linares, J., Bois, A. D., and Cavalheiro, G. (2025). Simulador web para o ensino de arquitetura de computadores com suporte a vetores e cache. In *Anais do XXVI Simpósio em Sistemas Computacionais de Alto Desempenho*, pages 121–132, Porto Alegre, RS, Brasil. SBC.
- Linares, J. P. and Cavalheiro, G. G. H. (2025). Simulador da arquitetura Neander em uma abordagem Web. In *ERAD-RS*, pages 113–116. SBC.
- Nova, B., Ferreira, J. C., and Araújo, A. (2013). Tool to support computer architecture teaching and learning. In *2013 1st Int. Conf. Port. Soc. Eng. Educ.*, pages 1–8. IEEE.
- Silva, G. P. and Borges, J. A. d. S. (2016). SimuS: Um simulador para o ensino de arquitetura de computadores. *Int. J. Comput. Archit. Educ.*, 5(1).
- Vollmar, K., Sanderson, and Pete (2006). MARS: an education-oriented MIPS assembly language simulator. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education, SIGCSE '06*, page 239–243, New York. ACM.
- Weber, R. F. (2000). *Fundamentos de arquitetura de computadores*. Sagra Luzzatto.
- Yurcik, W., Brumbaugh, and Larry (2001). A web-based little man computer simulator. In *Proc. of the 32nd Tech. Symp. on Computer Science Education*, pages 204–208.