

Resenha do Livro “Arquitetura e Organização de Computadores: uma Introdução”

Gabriel P. Silva
Instituto de Computação
Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brasil
gabriel@ic.ufrj.br

José Antonio S. Borges
Núcleo de Computação Eletrônica
Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brasil
antonio2@nce.ufrj.br

Resumo—Este artigo apresenta uma resenha do livro “Arquitetura e Organização de Computadores: uma Introdução”, lançado pela editora LTC – Livros Técnicos e Científicos em julho de 2024, que se apresenta como um texto completo e de custo acessível para explorar os fundamentos e as inovações em Arquitetura de Computadores, voltado para o ensino de graduação nos cursos de Bacharelado e Licenciatura em Computação e também Engenharia de Computação, Ciência de Dados e demais cursos tecnológicos de informática.

Palavras-chave—computadores, ensino, computação, arquitetura de computadores, organização de computadores

I. INTRODUÇÃO

No mundo atual, onde *smartphones*, armazenamento em nuvem e inteligência em dispositivos comuns se tornaram parte do cotidiano, a Arquitetura de Computadores não representa apenas o alicerce sobre o qual toda a tecnologia moderna é construída, mas também molda o futuro da inovação. Compreender os conceitos fundamentais dessa área permite que profissionais e entusiastas façam escolhas embasadas sobre *hardware* e *software*, além de contribuir para o desenvolvimento de soluções mais rápidas, seguras e eficientes.

A Computação, enquanto ciência básica ou aplicada, tem evoluído com velocidade crescente, apresentando inovações constantes desde os seus primórdios. O computador, como ferramenta básica da computação, não ficou imune a essas mudanças, apresentando uma sofisticação cada vez mais acentuada de todos os seus componentes e formas de organização. A renovação e atualização dos materiais didáticos e de apoio utilizados no ensino da organização e arquitetura de computadores é fundamental para a compreensão adequada do seu funcionamento, de modo que os alunos possam assimilar satisfatoriamente os conteúdos das demais disciplinas que compõem os cursos de graduação e tecnológicos da área da computação.

Como colocado nas diretrizes curriculares de 1998 do Ministério da Educação para os cursos da área:

“O conhecimento desta área é fundamental não apenas para aqueles que vão projetar novos computadores, mas também para aqueles que os utilizarão. O conhecimento dos princípios básicos de funcionamento dos computadores e da tecnologia embutida nestes permite um uso mais eficiente dos recursos e a determinação das classes de problemas que podem ser solucionadas com a tecnologia presente.”

Neste contexto, o livro “Arquitetura e Organização de Computadores: uma Introdução” [1], lançado pela LTC – Livros Técnicos e Científicos em julho de 2024, se apresenta como uma alternativa pragmática para entender e explorar os fundamentos e as inovações na arquitetura de computadores, oferecendo conceitos que permitem compreender melhor a tecnologia ao nosso redor, seu funcionamento e evolução.

Disponível tanto na versão impressa como *e-book*, este livro foi elaborado pelos professores da Universidade Federal do Rio de Janeiro, com expertises complementares, o que torna este livro bem balanceado nos temas relacionados ao *hardware* e ao *software* básico, com o intuito de dar ao leitor uma visão integrada do funcionamento dos computadores. Gabriel P. Silva é doutor em Engenharia de Sistemas e Computação com grande experiência em arquitetura de computadores, computação de alto desempenho, programação paralela e distribuída, e Internet das Coisas. Participou como pesquisador dos projetos dos primeiros computadores brasileiros, nas equipes dos pesquisadores pioneiros Julio Salek Aude [2], Newton Faller [3] e Cláudio Luís de Amorim [4], e trabalhou também como projetista na Cobra Computadores [5]. José Antonio Borges é doutor em Engenharia de Sistemas e Computação, tendo trabalhado em projetos de sistemas operacionais [6], *software* básico, computação gráfica, síntese de voz e *softwares* assistivos¹ para pessoas com deficiência.

Desta forma, o livro explora de forma balanceada tanto os aspectos de funcionamento do *hardware* como também as várias opções e configurações do *software* básico, com ênfase naquelas que se tornaram os padrões “de facto”, com o intuito de dar ao leitor uma visão integrada do funcionamento dos computadores.

Como os processadores da atualidade têm uma alta complexidade, que vai muito além das possibilidades de entendimento de um leitor iniciante na área, optamos por apresentar uma arquitetura de um processador didático muito simples, associado a um ambiente integrado de simulação [7], que permite experimentar a arquitetura de maneira interativa, o que inclui a edição de programas em linguagem de montagem (*assembly language*) e sua execução, com visualização em tempo real dos vários registradores da máquina, sua memória e outros

¹<https://intervox.nce.ufrj.br/dosvox/>

componentes internos, além de alguns dispositivos virtuais externos. O simulador, que é gratuito e de código aberto, é acompanhado por uma grande quantidade de exemplos, todos disponibilizados em <https://github.com/Simulador-Simus>.

Os maiores beneficiários de sua leitura são os alunos dos cursos da área de Licenciatura em Computação; Bacharelado em Ciência da Computação, Engenharia de Software, Sistemas de Informação, Engenharia de Computação, abrangendo também os Cursos Superiores de Tecnologia e Ciência de Dados. Há ainda partes do livro cujo interesse vai além destas áreas, atingindo a Matemática, a História da Tecnologia, Desenvolvimento da Computação no Brasil e diversos outros temas.

Os tópicos neste livro são abordados de uma maneira abrangente e geral, com uma profundidade adequada aos cursos mencionados acima, mas sem a perda de rigor ou precisão, procurando seguir as orientações contidas na proposta de diretrizes curriculares do Ministério da Educação², e no documento “Referenciais de Formação para os Cursos de Graduação em Computação”, de 2017, da Sociedade Brasileira de Computação [8].

II. TRABALHOS CORRELATOS

Os livros de arquitetura de computadores disponíveis no mercado brasileiro são, em sua maioria, importados, o que reflete uma dependência de fontes estrangeiras para ensino e estudo de arquitetura de computadores no Brasil. Em sua maioria apresentam um conteúdo abrangente e bastante detalhado e, por conta disso, têm um alto custo de aquisição, seja para as versões impressas ou mesmo *e-books*.

Por exemplo, “Arquitetura e Organização de Computadores” de William Stallings [9] é um dos livros mais respeitados e amplamente utilizados no campo da arquitetura e organização de computadores. A obra é conhecida por sua abordagem clara, sistemática e abrangente, o que a torna uma excelente referência tanto para estudantes quanto para profissionais que desejam entender os fundamentos da arquitetura de computadores. O livro cobre um vasto leque de tópicos, desde os princípios básicos até as tecnologias mais avançadas, proporcionando uma compreensão completa da arquitetura de computadores. A obra é frequentemente atualizada para incluir as tecnologias mais recentes e as tendências emergentes na arquitetura de computadores, mantendo-se relevante no campo. Contudo, para leitores iniciantes, a quantidade de detalhes pode ser um pouco excessiva. Enquanto a teoria é bem abordada, alguns leitores podem sentir falta de exemplos mais práticos ou de aplicações reais.

O livro de Patterson e Hennessy – “Organização e Projeto de Computadores” [10] – é indicado para cursos iniciais de Organização e Arquitetura, apresenta uma arquitetura consolidada de um processador comercial (MIPS, posteriormente RISC V). A obra combina teoria e aplicação prática, explorando tópicos como desempenho, hierarquia de memória,

paralelismo e técnicas de otimização. Dividido em capítulos bem organizados, o livro cobre desde a arquitetura básica de processadores até aspectos mais modernos, como arquiteturas RISC-V, uma contribuição significativa dos próprios autores. É uma obra bem atualizada e didática.

O livro “Arquitetura de Computadores - Uma Abordagem Quantitativa”, dos autores John L. Hennessy e David A. Patterson [11], é conhecido por sua abordagem quantitativa e analítica, utilizando métricas de desempenho para avaliar diferentes arquiteturas, sendo excelente para leitores que desejam um entendimento profundo e detalhado de arquitetura de computadores. Oferece uma análise detalhada e comparativa de diferentes arquiteturas, o que é valioso para profissionais de engenharia de computação e inclui estudos de caso, que ajudam a contextualizar a teoria em situações do mundo real, sendo amplamente utilizado em cursos avançados de arquitetura de computadores. Contudo, pode ser excessivamente técnico e difícil para iniciantes, exigindo um conhecimento prévio significativo e apresentando algumas discussões focadas em arquiteturas específicas, o que pode limitar a aplicabilidade geral do conteúdo. Apresenta o maior o custo entre todas as edições.

Outro livro que podemos apontar é “Organização Estruturada de Computadores”, de Andrew S. Tanenbaum [12], que oferece uma visão sistemática e organizada da arquitetura de computadores, ideal para iniciantes que precisam de uma base sólida. É extremamente didático, tornando-o adequado para estudantes que estão começando a estudar a organização de computadores. Cobre uma ampla gama de tópicos, oferecendo uma visão geral do campo, utilizando exemplos claros e simplificados que ajudam no entendimento dos conceitos complexos. Contudo, pode não estar tão atualizado quanto outros livros no que diz respeito às tecnologias mais recentes e apresenta também custos elevados de aquisição. A edição atual tem mais de 8 anos de lançada, e está defasada em relação ao estado da arte.

Quanto aos autores nacionais, certamente é importante destacar a obra pioneira “Introdução à Organização de Computadores” de Mário Monteiro [13], que apresenta um conteúdo abrangente, com profundidade e didáticas adequadas. Apresenta um breve histórico da computação, mas ignora os projetos nacionais desenvolvidos principalmente na década de 80, e carece de um simulador de processador para as arquiteturas apresentadas no livro. É uma obra interessante, mas que não é atualizada desde 2007 e, embora seja uma obra nacional, apresenta também os mesmos custos das obras importadas.

Em resumo, enquanto algumas obras apresentam um conteúdo muito teórico, outras carecem de um equilíbrio entre teoria e aplicações práticas. A falta de simuladores ou ferramentas práticas pode prejudicar a compreensão dos conceitos sobre o funcionamento dos processadores.

III. MOTIVAÇÕES PARA O DESENVOLVIMENTO TEMÁTICO

A obra aqui apresentada pretende ser uma contribuição nacional relevante, oferecendo um equilíbrio entre custo, acessibilidade e aplicação prática. Além de atender às necessidades

²https://normativasconselhos.mec.gov.br/normativa/view/CNE_RES_CNECESN52016.pdf

Começa com uma análise das primeiras ferramentas de cálculo, começando pelo ábaco, seguido pela Pascalina de Blaise Pascal e as contribuições de Anton Braun e Charles Babbage. Destaca também a importância da Álgebra de Boole para a lógica computacional.

Segue com a evolução para os computadores eletromecânicos, discutindo o Z3 de Konrad Zuse, a calculadora de números complexos de George Stibitz e o Mark I de Howard Aiken.

- A primeira geração de computadores é abordada com ênfase no ENIAC, o EDVAC e o UNIVAC, pioneiros da computação eletrônica.
- Na segunda geração, o foco é nos computadores com transistores, como o PDP-1 e a série IBM 7000.
- A terceira geração é marcada pelo uso de circuitos integrados, destacando o IBM 1130, a série IBM/360, o PDP-11 e o CDC 6600.
- A quarta geração, caracterizada pelos microprocessadores, inclui o Sinclair ZX80, o Apple I e II, o IBM 370 e o VAX 11/780.
- A quinta geração é representada pela popularização dos computadores pessoais, com destaque para o Commodore 64, o IBM/PC e o Macintosh.
- A sexta geração, ainda em desenvolvimento, explora inovações como o iPhone, Arduino e Raspberry Pi. Também é discutido o avanço dos supercomputadores e a arquitetura de alto desempenho.

Não menos importante, a última seção do capítulo oferece uma visão histórica do desenvolvimento de computadores no Brasil, começando com iniciativas pioneiras como o Computador Zezinho e o Lourinha. Menciona o Patinho Feio, um computador desenvolvido no Brasil na década de 1970. Discute-se o Minicomputador G-10 e a Cobra Computadores, uma empresa nacional de destaque, com diversos desenvolvimentos de computadores e *softwares* comerciais de sucesso na década de 1980. Mencionam-se ainda outras iniciativas e destacam-se projetos acadêmicos que contribuíram para o avanço da computação no país, como Pegasus/Plurix, Multiplus e NCP-II. Finaliza com exercícios para revisar e consolidar os conceitos discutidos.

C. Capítulo 3: Organização do Computador e do Processador

Este capítulo oferece uma visão completa da organização de computadores e processadores, através de uma abordagem estruturada, cobre desde a arquitetura básica de Von Neumann até os detalhes específicos de diferentes tipos de arquiteturas de processadores e suas instruções.

Inicialmente aborda a arquitetura de Von Neumann, fundamental para os computadores modernos, pela introdução do conceito do computador controlado por programa armazenado, onde uma memória única é compartilhada para dados e instruções, permitindo a execução de diversos tipos de aplicação. Introduce o modelo de barramento de sistema para conectar processador, memória e dispositivos de E/S, descrevendo o processador como o núcleo da execução de instruções e controle de dados.

A seção sobre o funcionamento do processador explica como ele executa instruções. A execução das instruções envolve várias etapas, incluindo a busca, decodificação e execução das mesmas. A unidade de controle é discutida como a parte do processador responsável por dirigir essas etapas, incluindo as suas formas de organização: microprogramada e diretamente por *hardware*.

Em seguida detalha modos de endereçamento, que descrevem as diferentes maneiras pelas quais um processador pode acessar dados na memória, e as duas formas de organização de bytes na memória: *little-endian* e *big-endian*.

Analisa diferentes arquiteturas de processador: a arquitetura de pilha utiliza uma pilha para armazenar operandos para operações aritméticas e lógicas. A arquitetura de acumulador tem um único registrador que armazena os operandos e resultados das operações. A arquitetura memória-memória permite que todas as operações sejam realizadas diretamente na memória. A arquitetura registrador-memória permite operações entre registradores e memória. A arquitetura registrador-registrador, também conhecida como load-store, realiza operações apenas entre registradores, necessitando de instruções separadas para carregar e armazenar dados da memória.

Discute as arquiteturas RISC (com instruções simples e eficientes) e CISC (com instruções complexas e extensas). A arquitetura RISC utiliza um conjunto reduzido de instruções simples, visando eficiência e velocidade de execução. A arquitetura CISC, por outro lado, possui um conjunto mais complexo e extenso de instruções, permitindo que operações mais complicadas sejam executadas com menos linhas de código. As vantagens e desvantagens de cada tipo de arquitetura são comparadas.

Em seguida, apresenta a arquitetura do processador Sapiens, uma evolução do processador Neander [14] [15], que é utilizado como estudo de caso, detalhando o seu conjunto de instruções, mostrando como ele lida com operações aritméticas, lógicas e de controle de fluxo. As operações de entrada e saída são detalhadas, destacando-se como o processador processa a comunicação com dispositivos externos. A microarquitetura do Sapiens é discutida, fornecendo uma visão detalhada de sua organização interna e funcionamento. O detalhamento das instruções do Sapiens oferece uma compreensão profunda de como as instruções são codificadas e executadas. Isso permite ao estudante uma compreensão melhor dos conceitos de tipo de arquitetura, código de instrução, modos de endereçamento, entre outros. O capítulo finaliza com exercícios para reforçar os conceitos discutidos.

D. Capítulo 4: Memória Principal e Hierarquia de Memória

Este capítulo proporciona uma compreensão abrangente da memória principal, hierarquia de memória, cache e memória virtual.

Apresenta, primeiramente, uma visão geral da memória principal, sua função crucial no armazenamento e recuperação de dados para a execução de programas. A seção seguinte classifica as memórias em várias categorias. As memórias de

acesso aleatório e sequencial são discutidas, destacando suas diferenças em termos de acesso e velocidade. As memórias voláteis, que perdem os dados quando a alimentação é desligada, e não-voláteis, que retêm os dados sem energia, são apresentadas. A distinção entre memórias dinâmicas, que precisam ser constantemente atualizadas, e estáticas, que não requerem atualização, é explicada. A subseção sobre memórias dinâmicas assíncronas e síncronas discute como esses tipos de memória operam em relação ao sincronismo com o relógio do sistema. A importância da paridade e dos códigos de correção de erro na detecção e correção de erros de memória é destacada. Por fim, as memórias Flash, usadas em muitos dispositivos de armazenamento modernos, são discutidas em termos de suas características e usos.

A hierarquia de memória é introduzida como uma maneira de organizar diferentes tipos de memória de acordo com a velocidade e o custo. O conceito de localidade, que descreve como os dados acessados recentemente são mais prováveis de serem acessados novamente em breve, é explicado. Métricas importantes, como taxa de acerto e tempo médio de acesso, são discutidas para avaliar o desempenho da hierarquia de memória.

A memória cache é detalhada como uma camada fundamental na hierarquia de memória. Estratégias de mapeamento da memória cache são discutidas, incluindo mapeamento direto, mapeamento associativo e mapeamento associativo por conjunto. O impacto do tamanho do bloco de cache no desempenho é analisado, bem como as operações de escrita na cache. A política de substituição de blocos, que determina qual bloco deve ser removido da cache quando um novo bloco precisa ser carregado, é discutida. Os três Cs da cache – Compulsório, Capacidade e Conflito – são apresentados como maneiras de analisar as falhas de cache. A distinção entre caches virtuais e físicas, bem como caches separadas e multinível, é explicada.

A memória virtual é discutida como uma técnica que permite que um computador use mais memória do que a fisicamente disponível, através do uso de armazenamento secundário para simular memória adicional. Esse conceito é fundamental para permitir a execução de programas grandes e complexos em sistemas com recursos limitados, além de permitir o compartilhamento adequado da memória entre os diversos programas em execução. O uso de técnicas modernas de memória virtual, como a ZRAM, que permite a criação de dispositivos de bloco de memória compactada na RAM, também são discutidas. O capítulo conclui com exercícios para reforçar e aplicar os conceitos discutidos.

E. Capítulo 5: Entrada e Saída, Barramentos e Redes de Comunicação

Este capítulo oferece uma visão completa dos sistemas de E/S, barramentos e redes de comunicação.

Começa com uma introdução aos conceitos fundamentais de entrada e saída (E/S), destacando sua importância para a interação entre o computador e dispositivos periféricos. Segue com a programação das interfaces de E/S, explicando

como o sistema operacional e os programas gerenciam esses dispositivos.

A importância das exceções e interrupções no gerenciamento eficiente dos dispositivos de E/S é destacada. Essas técnicas permitem que o processador responda rapidamente a eventos inesperados, garantindo a operação contínua e eficiente do sistema. O acesso direto à memória (DMA) é explicado como uma técnica que permite a transferência de dados entre a memória e os dispositivos de E/S sem a intervenção direta do processador, melhorando o desempenho do sistema.

A subseção sobre *chipset* discute como esses componentes integrados facilitam a comunicação entre o processador, a memória e os dispositivos de E/S, desempenhando um papel importante na arquitetura do computador.

A seção sobre transmissão da informação aborda as diferenças entre transmissão em paralelo e transmissão serial. A transmissão em paralelo permite a transferência de múltiplos bits simultaneamente, enquanto a transmissão serial envia bits individualmente, o que pode ser mais eficiente para longas distâncias. A importância da detecção e correção de erros de transmissão também é discutida, destacando técnicas como bits de paridade e códigos de correção de erro que garantem a integridade dos dados durante a transferência.

Os barramentos de E/S são explorados em detalhe, começando com suas características gerais e evoluindo para uma discussão sobre a evolução dos barramentos ao longo do tempo. Barramentos de E/S legados são mencionados, seguidos por uma análise dos barramentos modernos como USB (Universal Serial Bus), PCIe (PCI Express), SATA (Serial ATA) e barramentos de vídeo.

A seção sobre redes de comunicação abrange várias tecnologias de rede, começando com a fibra óptica, que oferece alta velocidade e grande capacidade de transmissão de dados. A Ethernet é discutida como a tecnologia de rede local mais comum, enquanto Wi-Fi é destacada por sua conveniência e mobilidade. O Bluetooth é apresentado como uma tecnologia de comunicação sem fio de curto alcance. O capítulo termina com exercícios práticos para reforçar os conceitos discutidos.

F. Capítulo 6: Armazenamento e Periféricos

O capítulo proporciona uma compreensão abrangente de armazenamento e periféricos.

Inicialmente apresenta uma descrição dos processos de inicialização do computador, destacando o papel do BIOS e da UEFI. A seção seguinte detalha as funções do BIOS, que é o *firmware* responsável por iniciar o *hardware* do computador antes de iniciar o sistema operacional. Em contraste, a UEFI é uma versão mais moderna e flexível, oferecendo recursos avançados e uma interface gráfica.

Segue com a exploração dos discos rígidos (HDD), detalhando a estrutura interna desses dispositivos, suas características de desempenho e diferentes esquemas de particionamento. A estrutura interna dos HDDs é descrita, destacando componentes como discos, cabeças de leitura/gravação e motores. A subseção também discute o desempenho dos

HDDs em termos de velocidade de rotação, latência e taxa de transferência.

Os dispositivos de estado sólido (SSD) são apresentados como uma alternativa aos HDDs, com vantagens significativas em termos de velocidade, durabilidade e consumo de energia. Uma comparação entre HDDs e SSDs é feita, destacando as vantagens e desvantagens de cada tecnologia em termos de desempenho, custo e capacidade.

A próxima seção aborda outros tipos de dispositivos de armazenamento, como unidades de fita, CDs, DVDs e Blu-rays, cada um com suas próprias características e usos específicos.

A seção seguinte explora o conceito de RAID (Redundant Array of Independent Disks), explicando os principais níveis de RAID e suas respectivas vantagens e desvantagens.

- RAID 0: Oferece desempenho melhorado através da distribuição de dados, mas sem redundância.
- RAID 1: Fornece espelhamento de dados para redundância, mas sem aumento de desempenho.
- RAID 10 e 01: Combina as vantagens do RAID 0 e RAID 1.
- RAID 5 e 6: Oferecem redundância com paridade distribuída, com RAID 6 fornecendo proteção adicional.

A subseção sobre sistemas de armazenamento discute DAS (Direct Attached Storage), SAN (Storage Area Network) e NAS (Network Attached Storage), comparando suas características e usos.

- DAS: Conexão direta de dispositivos de armazenamento ao servidor.
- SAN: Rede especializada de alta velocidade que fornece acesso a armazenamento em nível de bloco.
- NAS: Dispositivo de armazenamento conectado à rede que fornece acesso a arquivos.

O armazenamento em nuvem é apresentado também como uma solução moderna e flexível para o armazenamento de dados, destacando suas vantagens em termos de escalabilidade, acessibilidade e custo-benefício.

A última seção do capítulo aborda diversos periféricos, incluindo teclados, mouses, impressoras e dispositivos de vídeo.

- Teclados: Diferentes tipos de teclados, incluindo mecânicos e de membrana, e suas características.
- Mouse: Tipos de mouses, como ópticos e a laser, e suas características de desempenho.
- Impressoras: Tipos de impressoras, incluindo jato de tinta e laser, e suas aplicações.
- Vídeo: Dispositivos de vídeo, como monitores e projetores, e suas características.

O capítulo conclui com exercícios práticos para reforçar e aplicar os conceitos discutidos.

G. Capítulo 7: Programação em Linguagem de Montagem

Entendemos que o uso de exemplos de programas em linguagem de montagem é importante para a compreensão do conjunto de instruções e do funcionamento do processador. Para isso, lançamos mão de um processador didático muito

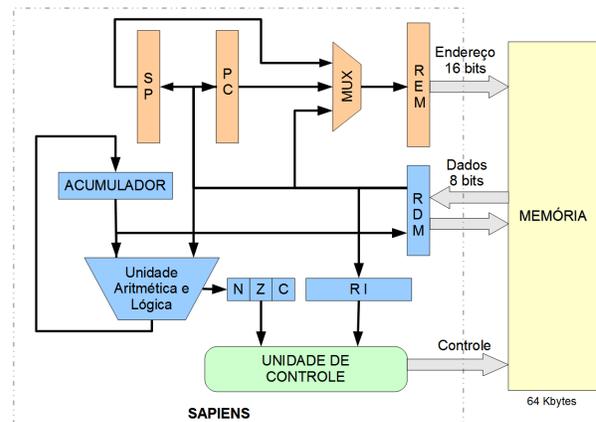


Figura 2. Arquitetura do processador didático Sapiens

simples, cuja experimentação pode ser feita com uso de um simulador didático [7] disponível gratuitamente no endereço <https://github.com/Simulador-Simus>. Neste repositório encontramos o código fonte do simulador e executáveis para diversos tipos de sistema operacional, como Linux e Windows. Além disso, estão disponíveis um manual detalhado de utilização do simulador, todos os exemplos em linguagem de montagem utilizados neste livro.

O capítulo 7 oferece uma introdução completa à programação em linguagem de montagem, cobrindo desde conceitos básicos até operações avançadas. A inclusão de exemplos detalhados e exercícios práticos proporciona uma base sólida para entender e aplicar a linguagem de montagem no contexto de arquitetura de computadores.

Este capítulo primeiro explica como o código escrito em linguagens de alto nível é convertido em instruções que o *hardware* do computador pode entender e executar. Discute como o código-fonte é transformado em código de máquina por meio dos processos de compilação e montagem, que traduzem o código para uma “linguagem de baixo nível”, mais próxima do *hardware*.

A seguir, o capítulo apresenta a linguagem de montagem usada pelo processador didático Sapiens (Fig. 2), detalhando sua estrutura e principais componentes. Explica também como a programação em linguagem de máquina funciona, ressaltando a importância de entender essa linguagem para otimizar e controlar o *hardware* com precisão.

Visando proporcionar uma experiência agradável ao leitor, foi adotada uma sequência didática em que as operações que são comuns nas linguagens de programação de alto nível são traduzidas para uma codificação padronizada, algo semelhante ao que um compilador faria no processo de geração de código. Todo o processo de execução pode ser acompanhado passo a passo, verificando-se a alteração dos conteúdos dos registros como da memória onde o programa é executado. Dispositivos virtuais de entrada e saída permitem um *feedback* adequado do resultado da execução dos programas para o estudante.

Alguns exemplos em linguagem de montagem são apresentados, tais como:

- atribuir valores constantes a variáveis e como copiar valores de uma variável para outra na linguagem de montagem;
- operações aritméticas básicas envolvendo uma variável e uma constante, como adição e subtração;
- operações entre duas e três variáveis, um passo em direção a cálculos mais complexos;
- como implementar testes e desvios condicionais na linguagem de montagem, permitindo a execução de diferentes caminhos de código com base em condições específicas;
- como implementar laços simples e com contadores, uma estrutura de controle fundamental na programação;
- operações de soma e subtração em 16 bits, importantes para manipulações de dados maiores;
- instruções de entrada e saída (IN e OUT), que permitem a comunicação direta com dispositivos periféricos;
- uso da instrução TRAP, utilizada para realizar chamadas ao sistema operacional, permitindo operações como leitura e escrita em arquivos;
- como acessar e manipular elementos em um vetor, um conceito importante para a organização e processamento de dados;
- uso de subrotinas para modularizar e reutilizar código, bem como a utilização da pilha para gerenciar chamadas de função e variáveis locais.

O capítulo termina com uma série de exercícios práticos que permitem ao leitor aplicar os conceitos aprendidos, reforçando a compreensão e habilidade na programação em linguagem de montagem.

H. Capítulo 8: Arquiteturas Avançadas

O capítulo inicia com a explicação do conceito de pipeline em processadores, detalhando como a execução de instruções é dividida em várias etapas para melhorar o desempenho. A abordagem é ilustrada com exemplos de como diferentes estágios de execução (como busca, decodificação e execução) são realizados simultaneamente. A seção fornece uma visão mais detalhada sobre a implementação de pipelines, abordando questões como *hazards* (conflitos de dados, controle e estrutural) e técnicas para minimizá-los, como *buffers* e técnicas de previsão de desvios. A arquitetura do MIPS R2000 é apresentada como um exemplo prático de um processador com pipeline, destacando como ele implementa e gerencia os diferentes estágios do pipeline para otimizar a execução de instruções.

Em seguida, explicamos o conceito de arquiteturas *superpipelined*, que possuem pipelines mais profundos (mais estágios) para aumentar a taxa de execução de instruções e melhorar o *throughput*. Discutem-se os desafios e soluções associados a pipelines mais profundos, incluindo o gerenciamento de latências e a complexidade de controle. O Pentium 4 é utilizado como um estudo de caso para demonstrar uma arquitetura

superpipelined, com ênfase em como o processador lida com a profundidade do pipeline e os desafios associados.

Em seguida, descrevem-se as arquiteturas superescalares que permitem a execução simultânea de múltiplas instruções por ciclo de relógio, aumentando a capacidade de execução paralela. Os detalhes de implementação das arquiteturas superescalares são apresentados, incluindo técnicas para o despacho e a execução de múltiplas instruções, e o gerenciamento de recursos. A arquitetura ARM Cortex A72 é apresentada como um exemplo de arquitetura superescalar, com detalhes sobre como ela executa múltiplas instruções por ciclo e lida com a execução paralela.

A abordagem de usar múltiplos núcleos de processamento em um único chip (*multicores e manycores*), como forma de aumentar o desempenho e a capacidade de processamento paralelo, é apresentada. Os aspectos técnicos e desafios da implementação de processadores multicore e manycore, incluindo comunicação entre núcleos, escalabilidade e gerenciamento de cache são discutidos também. O Intel Xeon Phi é apresentado como um exemplo de processador *manycore*, destacando suas características e como ele lida com tarefas paralelas em larga escala.

A arquitetura VLIW, que utiliza palavras de instrução muito longas para codificar múltiplas operações em um único ciclo de relógio, visando o paralelismo de instruções, é descrita em seguida. Uma visão geral do desenvolvimento e evolução das arquiteturas VLIW é oferecida a seguir, incluindo a discussão de como elas surgiram e como foram aprimoradas ao longo do tempo. O Itanium IA-64 é apresentado como um exemplo de arquitetura VLIW, com detalhes sobre como ele utiliza palavras de instrução longas e como isso afeta o desempenho e a programação.

O conceito de *multithreading* é explorado a seguir, mostrando que ele permite a execução de múltiplas *threads* em paralelo dentro de um único núcleo de processador, melhorando a utilização do processador e a eficiência. As técnicas e desafios associados à implementação de *multithreading* são discutidos, incluindo o gerenciamento de contexto de *threads* e a sincronização entre elas. O IBM Power5 é utilizado como um exemplo de processador com suporte a *multithreading*, destacando suas características e como ele gerencia a execução de múltiplas *threads*.

Os aceleradores, como GPUs e FPGAs, que são projetados para realizar operações específicas com alta eficiência e desempenho são discutidos em seguida. A GPU Kepler é apresentada como um exemplo de acelerador, com detalhes sobre sua arquitetura e como ela é otimizada para operações paralelas intensivas, como computação gráfica e cálculo científico.

Neste capítulo são apresentadas e discutidas as seguintes arquiteturas paralelas:

- Arquiteturas SISD (Single Instruction, Single Data) - Explora a arquitetura SISD, onde uma única instrução é executada em um único dado por vez, representando a arquitetura de computadores mais básica.

- Arquiteturas SIMD (Single Instruction, Multiple Data) - Descreve a arquitetura SIMD, que permite a execução da mesma instrução em múltiplos dados simultaneamente, melhorando a eficiência em operações vetoriais.
- Arquiteturas MIMD (Multiple Instruction, Multiple Data) - Aborda a arquitetura MIMD, onde múltiplas instruções são executadas em múltiplos dados simultaneamente, permitindo uma maior flexibilidade e paralelismo.

O supercomputador Netuno [16], utilizado na UFRJ a partir de 2008 é mostrado como um exemplo de arquitetura paralela, voltado para aplicações de alto desempenho no processamento de prospecção e análise de petróleo. O capítulo termina com exercícios para aplicar e reforçar os conceitos discutidos.

V. CONCLUSÃO

Apresentamos neste artigo uma resenha do livro “Arquitetura e Organização de Computadores: uma Introdução”, que consideramos uma contribuição importante para o ensino de Computação e Engenharia no Brasil. O livro procura abordar de uma forma objetiva e atualizada conceitos importantes para a compreensão do funcionamento do computador e suas diversas formas de organização. Através do uso de diversos exemplos, estudos de caso, exercícios, e um simulador didático de processador, procura tornar o conteúdo apresentado mais acessível ao estudante. O livro tem forte atrativo também através de centenas de fotos, diagramas e ilustrações, algumas delas provenientes do acervo pessoal dos autores.

O livro está disponível em dois formatos: impresso e eletrônico (*e-book* para leitura *online* em bibliotecas virtuais ou em aplicativos do tipo Kindle).

Para além do texto do livro, os autores, com o apoio da editora LTC, oferecem alguns recursos *online*, de grande valia para os professores, em particular:

- um conjunto de *slides* em formato PDF que cobre, em seqüência, todos os temas do livro, o que facilita muito a sua aplicação em cursos presenciais ou remotos;
- resolução de muitos exercícios extras que podem ser usados como material de estudo para os alunos.

Como trabalhos futuros pretendemos desenvolver uma versão JavaScript para execução direto no navegador ou no NodeJS, de modo a tornar o uso do simulador independente de distribuições ou tipos de sistemas operacionais. Adicionalmente, estamos preparando uma outra versão do simulador que apresente o caminho de dados, e como ocorre, visualmente, o percurso de cada instrução entre os diferentes componentes do modelo processador utilizado, tais como acumulador, registrador de instruções, apontador de pilha e apontador de instruções, entre outros.

Enfatizamos, finalmente, que o código fonte do simulador é aberto, e distribuído segundo um modelo GPL de *software* livre, o que permite que sejam agregadas contribuições ao seu desenvolvimento. Todos os recursos que foram aqui apresentados estão disponíveis no endereço <https://simulador-simus.github.io>.

REFERÊNCIAS

- [1] G. P. Silva and J. A. S. Borges, *Arquitetura e Organização de Computadores: Uma Introdução*. Rio de Janeiro: LTC, 2024.
- [2] J. S. Aude and et al., “Multiplus: a modular high-performance multiprocessor,” UFRJ, Rio de Janeiro, Tech. Rep., 1991. [Online]. Available: <https://pantheon.ufrj.br/handle/11422/1554>
- [3] N. Faller and et al., “O Projeto Pegasus-32X/Plurix,” in *Anais do XVII Congresso Nacional de Informática - SUCESU*. Rio de Janeiro: SUCESU, 7 1984, p. 10.4.
- [4] C. L. Amorim and et al., “A Segunda Geração de Computadores de Alto Desempenho da COPPE/UFRJ,” in *Anais do VIII Simpósio de Brasileiro de Arquitetura de Computadores e Processamento de Alto Desempenho*. SBC, 1996, pp. 87–98.
- [5] R. Rodrigues and G. P. Silva, “Um Supermini com Arquitetura baseada em Múltiplos Microprocessadores de 32 bits,” in *Anais do XIX Congresso Brasileiro de Informática - SUCESU*. Rio de Janeiro: SUCESU, 1986, pp. 131–135.
- [6] J. A. S. Borges and N. Faller, “O CP/M No Unix - Uma Experiência de Emulação,” in *Anais do Congresso Regional da SUCESU*. São Paulo: SUCESU, 1987.
- [7] G. P. SILVA and J. A. S. BORGES, “SimuS - Um Simulador Para o Ensino de Arquitetura de Computadores,” *International Journal of Computer Architecture Education (IJCAE)*, vol. 5, no. 1, pp. 7–12, 12 2016. [Online]. Available: http://www2.sbc.org.br/ceacpad/ijcae/v5_n1_dec_2016/IJCAE_v5_n1_dec_2016_paper_2_vf.pdf
- [8] A. F. Zorzo, D. Nunes, E. Matos, I. Steinmacher, J. Leite, R. M. Araujo, R. Correia, and S. Martins, *Referenciais de Formação para os Cursos de Graduação em Computação*. Sociedade Brasileira de Computação (SBC), 2017.
- [9] W. Stallings, *Arquitetura e Organização de Computadores*, 8th ed. São Paulo: Pearson, 2010.
- [10] D. Patterson and J. L. Hennessy, *Arquitetura de Computadores: Uma Abordagem Quantitativa*, 3rd ed. Rio de Janeiro: Campus, 2003.
- [11] J. L. Hennessy and D. A. Patterson, *Arquitetura de Computadores: Uma Abordagem Quantitativa*, 5th ed. Elsevier, 2013.
- [12] A. S. Tanenbaum, *Organização Estruturada de Computadores*, 6th ed. São Paulo: Pearson Universidades, 2013.
- [13] M. Monteiro, *Introdução à Organização de Computadores*, 5th ed. Rio de Janeiro: LTC, 2007.
- [14] R. F. Weber, *Fundamentos de Arquitetura de Computadores*, 4th ed. Porto Alegre: Bookman, 2012.
- [15] J. A. S. Borges and G. P. Silva, *O Simulador Neander-X para o Ensino de Arquitetura de Computadores*. Porto Alegre: Sociedade Brasileira de Computação, 12 2016, vol. 5. [Online]. Available: http://www2.sbc.org.br/ceacpad/ijcae/v5_n1_dec_2016/IJCAE_v5_n1_dec_2016_paper_2_vf.pdf
- [16] G. P. Silva, J. C. Correa, C. Bentes, S. Guedes, and M. Gabioux, “The Experience in Designing and Evaluating the High Performance Cluster Netuno,” *International Journal of Parallel Programming*, vol. 42, no. 2, pp. 265–286, 2014. [Online]. Available: <http://dx.doi.org/10.1007/s10766-012-0224-7>