

**ENGENHARIA DE COMPUTAÇÃO
ESAMC**

**ARQUITETURA DE COMPUTADORES
JACKSON RODRIGUES**

CAMPINAS 2025

SUMÁRIO

INTRODUÇÃO À COMPUTADORES.....	3
PIPELINE E PARARELISMO.....	3
ARQUITETURA RISC E CISC	5
MULTIPROCESSADORES E MULTICOMPUTADORES.....	8
FUTURO DOS COMPUTADORES.....	8

INTRODUÇÃO À COMPUTADORES

A evolução dos computadores acompanhou a evolução da sociedade durante os séculos XX e XXI. Entretanto, a história do computador não teve início apenas na modernidade. Os computadores de primeira geração funcionavam por meio de circuitos e válvulas eletrônicas. Possuíam o uso restrito, além de serem imensos e consumirem muita energia.

Um exemplo é o ENIAC (Eletronic Numerical Integrator and Computer) que consumia cerca de 200 quilowatts e possuía 19.000 válvulas. Ainda com dimensões muito grandes, os computadores da segunda geração funcionavam por meio de transistores, os quais substituíram as válvulas que eram maiores e mais lentas. Nesse período já começam a se espalhar o uso comercial. Os computadores da terceira geração funcionavam por circuitos integrados. Esses substituíram os transistores e já apresentavam uma dimensão menor e maior capacidade de processamento.

Com o desenvolvimento da tecnologia da informação, os computadores diminuem de tamanho, aumentam a velocidade e capacidade de processamento de dados. São incluídos os microprocessadores com gasto cada vez menor de energia. Nesse período, mais precisamente a partir da década de 90, há uma grande expansão dos computadores pessoais. Além disso, surgem os softwares integrados e a partir da virada do milênio, começam a surgir os computadores de mão. Ou seja, os smartphones, iPod, iPad e tablets, que incluem conexão móvel com navegação na web.

A evolução da arquitetura dos computadores foi fortemente impulsionada pela busca constante por maior desempenho. À medida que surgiam novas demandas tecnológicas e limitações físicas nos processadores, tornou-se necessário inovar design dos sistemas computacionais. Essa trajetória de avanços permitiu o desenvolvimento de máquinas mais rápidas, eficientes e capazes de atender aos desafios cada vez mais complexos da sociedade.

PIPELINE E PARARELISMO

O pipeline é uma técnica de arquitetura de processadores que permite a execução de várias instruções de forma sobreposta, dividindo o processo de execução em etapas sequenciais. Cada etapa realiza uma parte do trabalho, como buscar a instrução, decodificá-la, executá-la, acessar a memória e escrever o resultado.

Funciona de maneira semelhante a uma linha de montagem em uma fábrica: enquanto uma instrução está sendo executada, outra pode estar sendo decodificada e uma terceira sendo buscada. Isso permite que múltiplas instruções sejam processadas simultaneamente, cada uma em uma fase diferente do pipeline.

Os benefícios do pipeline estão no poder de eficiência, ao aproveitar melhor o tempo do processador, o pipeline permite que várias instruções sejam executadas ao mesmo tempo, reduzindo o tempo total de execução. Além disso, o throughput é reduzido, que é basicamente o tempo entre a execução de uma instrução e outra, sendo assim maior velocidade de processamento. Essa técnica aumentou significativamente o número de instruções processadas por segundo, tornando os processadores mais rápidos sem precisar aumentar a frequência. O pipeline apresenta algumas limitações entre elas: dependência de dados, dependência de controle, conflitos de recursos e custo de implementação.

Dependência de dados: É quando uma instrução depende do resultado da anterior, pode haver atrasos (chamados data hazards), pois o pipeline precisa esperar que o dado esteja disponível.

Dependência de controle: É ocasionada por desvios e instruções de decisão (como if e loop), que dificultam prever qual a instrução será executada em seguida, gerando o Branch Hazard.

Conflitos de Recursos: Quando duas ou mais instruções tentam acessar o mesmo recurso ao mesmo tempo, o pipeline pode precisar ser pausado (stall).

Custo de implementação: O controle do pipeline exige lógica adicional e torna o projeto do processador mais complexo.

Um dos grandes avanços importantes foi o paralelismo, que permite que várias tarefas sejam executadas ao mesmo tempo. Isso foi essencial para lidar com aplicações que exigem grande poder de processamento, como simulações científicas, jogos e inteligência artificial. O paralelismo levou à criação da arquitetura multicore, em que múltiplos núcleos de processamento são integrados em um único chip. Com

isso, tornou-se possível executar diversos processos simultaneamente, otimizando o desempenho geral do sistema.

A memória cache também teve papel crucial nesse avanço. Por estar localizada próxima ao processador e possuir alta velocidade, a cache armazena dados frequentemente utilizados, reduzindo o tempo de acesso à memória principal. Isso acelerou consideravelmente.

ARQUITETURA RISC E CISC

Além disso, arquiteturas como RISC (Reduced Instruction Set Computing) trouxeram um conjunto de instruções mais simples e rápidas de executar, facilitando a implementação de pipelines e reduzindo o consumo de energia. Junto a isso, técnicas como execução fora de ordem e hiperthreading aumentaram a eficiência interna do processador, permitindo que mais instruções fossem concluídas em menos tempo.

A principal razão para buscar sempre evoluir as arquiteturas de computadores é atender as crescentes demandas da sociedade e da tecnologia. À medida que surgem novas aplicações ao longo da história como por exemplo automação industrial, sistemas embarcados, jogos, redes sociais, simulações científicas e desenvolvimento de inteligência artificial os computadores precisam ser mais rápidos, eficientes e capazes para processar grandes volumes de dados em menos tempo. Logo o desenvolvimento de CPU's era necessário o que levou ao avanço de múltiplos núcleos, paralelismo, memória cache, e novas instruções otimizadas.

RISC (Reduced Instruction Set Computer) e CISC (Complex Instruction Set) são os tipos principais de arquitetura de processadores. Eles determinam a quantidade e a complexidade das instruções suportadas por um chip.

O risc usa um conjunto reduzido de instruções e é mais comum em arquiteturas de processadores para dispositivos móveis, como ARM. Já as arquiteturas do tipo CISC, como a x86, são mais comuns em CPUs para computadores e servidores.

Como as instruções do Risc são a mais simples, elas podem ser carregadas de maneira mais eficiente na memória e executadas de forma mais previsível. O objetivo do RISC é executar cada instrução em um único ciclo de clock.

Segue algumas arquiteturas:

Arm: criada pela Acorn Computers em 1983 e hoje de propriedade da Arm Ltd., é a mais popular em dispositivos móveis, wearables e sistemas embarcados. É conhecida pela eficiência energética e versatilidade para funcionar em equipamentos de todos os tamanhos, de fones de ouvido a grandes servidores.

RISC-V: é uma arquitetura de padrão aberto que surgiu em 2010 na Universidade da Califórnia, Berkeley. Usada em uma variedade de aplicações, desde microcontroladores até supercomputadores, é livre de royalties e altamente flexível.

PowerPC: desenvolvido pela aliança AIM (Apple, IBM, Motorola) em 1991. Originalmente usado em Macs da Apple, foi depois adotada em sistemas embarcados e consoles de videogame como Xbox 360 e Nintendo Wii.

MIPS: foi criada pela MIPS Technologies em 1985 e é utilizada em uma variedade de sistemas, incluindo videogames, roteadores e sistemas embarcados. É conhecida por sua eficiência, design modular e uso em educação e pesquisa.

SPARC: desenvolvida pela Sun Microsystems em 1987, foi usada em servidores e workstations devido à alta escalabilidade.

Arquiteturas do tipo CISC, como a x86, são projetadas para executar mais operações com menos linhas de código de baixo nível. Ou seja, enquanto o RISC pode exigir várias instruções para determinada tarefa, o CISC pode fazer o mesmo trabalho com uma única instrução.

Segue algumas arquiteturas:

x86: criada pela Intel em 1978, no processador Intel 8086 de 16 bits, é usada em desktops, notebooks e servidores. É a arquitetura mais popular em PCs domésticos e evoluiu ao longo das décadas para suportar instruções de 32 e 64 bits.

x86-64: também conhecida como amd64, foi criada pela AMD em 1999. É uma extensão da arquitetura x86 que suporta computação de 64 bits, permitindo maior desempenho e capacidade de memória, mantendo a compatibilidade com softwares legados. A implementação da Intel também é conhecida como Intel 64.

IA-64 (Itanium): foi criada por HP e Intel em 2001 para ser usada em servidores de alto desempenho. Tem como base o conceito de execução explícita em paralelo (EPIC), que aumenta a eficiência da execução de instruções. Por não ser retrocompatível com o x86, não se popularizou em PCs.

IBM Z: criada pela IBM, é usada principalmente em mainframes. Tem como principal característica a alta confiabilidade e disponibilidade, sendo que o Z se refere a “zero downtime” (tempo fora do ar).

System/360: foi a primeira família de arquiteturas da IBM a cobrir uma ampla gama de aplicações, criada em 1964. A arquitetura IBM Z é retrocompatível com programas de System/360 até hoje.

Motorola 68000: arquitetura criada pela Motorola em 1979 e usada em uma série de computadores, incluindo o primeiro Apple Macintosh (Macintosh 128K) e o videogame Sega Mega Drive.

Arquiteturas RISC costumam ser mais eficientes em consumo de energia do que as CISC, porque suas operações consomem menos energia por instrução.

Como uma arquitetura do tipo RISC parte de um conjunto de instruções mais simples, é possível projetar chips que consumam menos energia para dispositivos menores com bateria, como celulares e notebooks. Já as CPUs para servidores podem ter extensões na arquitetura para melhorar o desempenho em tarefas específicas.

Também é possível projetar CPUs do tipo CISC com menor TDP, ou seja, que gerem menos calor e consumam menos energia. A Intel, por exemplo, chegou a fabricar processadores Atom com arquitetura x86 para smartphones com Android. No entanto, esse tipo de aplicação costuma ser menos comum no CISC.

MULTIPROCESSADORES E MULTICOMPUTADORES

Em multiprocessadores e multicomputadores são arquiteturas paralelas, mas com diferenças cruciais. Multiprocessadores são sistemas com múltiplos processadores compartilhando uma mesma memória, já multicomputadores têm processadores com memória individual, comunicando através de mensagens. Na memória compartilhada, todos os processadores acessam a mesma memória principal, facilitando o compartilhamento de dados e a sincronização entre os processos. Na Escalabilidade (capacidade de adicionar processadores) é limitada em comparação com os multicomputadores, principalmente por questões de latência e compartilhamento de recursos.

Em multiprocessadores, existem diferentes tipos de implementação, como sistemas de memória compartilhada (SMP), memória distribuída com memória compartilhada (NUMA) e memória distribuída com memória compartilhada em níveis (COMA). Os sistemas operacionais para multiprocessadores precisam gerenciar a memória compartilhada e a sincronização entre os processadores.

Em multicomputadores, a memória distribuída em cada processador tem sua própria memória local, o que significa o design de hardware e software, mas exige comunicação explícita entre os processadores. Na escalabilidade os multicomputadores são mais flexíveis que os multiprocessadores, podendo ter grandes números de processadores conectados em uma rede, e a comunicação entre os processadores é feita através da troca de mensagens (message passing), o que pode introduzir uma latência maior do que a memória compartilhada. Os sistemas operacionais para multicomputadores são, em geral, mais flexíveis, pois podem ser configurados para diferentes tipos de interconexão e arquitetura.

A escolha entre os multiprocessadores e multicomputadores depende do tipo de aplicação, do orçamento e da necessidade de escalabilidade. Multiprocessadores são mais adequados para aplicações que precisam de acesso rápido à mesma memória, enquanto multicomputadores são mais apropriados para aplicações que exigem muitos processadores e toleram uma maior latência de comunicação.

FUTURO DOS COMPUTADORES

O futuro nos computadores é promissor, ao longo das décadas os computadores passaram por grandes desenvolvimentos avançados que hoje a tendência é que fique

melhor ainda, empresas como Nvidia, AMD, Intel, IBM, Google, Microsoft estão fortemente investindo no avanço do desenvolvimento computacional, a Nvidia por exemplo investe fortemente em IA generativa para chips dedicados para suas GPUs, Microsoft e IBM estão fortemente no desenvolvimento em computação quântica, que é uma espécie de super computador que lida com processamentos de dados ultra vezes mais rápidos lidando com cálculos complexos que levava anos para ser resolvido graças união da física quântica que lida com estudo e comportamento atômicos, a IBM segue com Quantum, seu super computador quântico, enquanto a Microsoft segue com projeto StationQ e Majorana, uma espécie de chip quântico. E Tanto a Intel quanto a AMD segue no desenvolvimento avançado de processadores avançados com integrações de inteligência artificial.

Entender sobre os computadores é muito importante porque é algo que pode ajudar no dia a dia à fazer o uso eficiente e criar soluções alternativas, isso também impacta na sociedade e no mercado de trabalho, como por exemplo possuir habilidades computacionais seja para desenvolver uma pesquisa, desenvolver um projeto, e também ajuda a pessoa a buscar um equipamento ideal seguindo especificações, como por exemplo um arquiteto e um design que precisa de computador com poder de renderização, ou para um usuário comum que não precisa de uma máquina superpotente para realizar tarefas básicas do dia a dia.