

[< VOLTAR](#)

Pipeline, RISC e CISC

Abordaremos nesta aula assuntos referentes às arquiteturas RISC e CISC. Esses dois tipos de arquiteturas diferem muito entre si. Porém tanto uma quanto a outra traz consigo vantagens e desvantagens.

NESTE TÓPICO

Marcar
tópico

Reduced instruction set computer – RISC (computador com um conjunto reduzido de instruções)

É uma linha de arquitetura de computadores que favorece um conjunto simples e pequeno de instruções que levam aproximadamente a mesma quantidade de tempo para serem executadas. A maioria dos microprocessadores modernos são RISCs, por exemplo DEC Alpha, SPARC, MIPS, e PowerPC.

Os processadores baseados na computação de conjunto de instruções reduzido não tem microprogramação, as instruções são executadas diretamente pelo hardware. Como característica, essa arquitetura, além de não ter microcódigo, tem o conjunto de instruções reduzido, bem como baixo nível de complexidade.

A ideia foi inspirada pela descoberta de que muitas das características incluídas na arquitetura tradicional de processadores para ganho de desempenho foram ignoradas pelos programas que foram executados neles. Mas o desempenho do processador em relação à memória que ele acessava era crescente. Isso resultou num número de técnicas para otimização do processo dentro do processador, enquanto ao mesmo tempo tentava reduzir o número total de acessos à memória.

Caracterização das arquiteturas RISC:

- conjunto reduzido e simples de instruções
- formatos simples e regulares de instruções
- operandos sempre em registros



- modos simples de endereçamento à memória
- uma operação elementar por ciclo de máquina
- uso de *pipeline*

RISC é também a arquitetura adotada para os processadores dos videogames modernos, que proporcionam um hardware extremamente dedicado somente à execução do jogo, tornando-o muito mais rápido em relação a microcomputadores com mais recursos.



Complex instruction set computer – CISC (computador com um conjunto complexo de instruções)

É uma linha de arquitetura de processadores capaz de executar centenas de instruções complexas diferentes, sendo, assim, extremamente versátil. Exemplos de processadores CISC são os 386 e os 486 da Intel.

Os processadores baseados na computação de conjunto de instruções complexas contêm uma microprogramação, ou seja, um conjunto de códigos de instruções que são gravados neles, permitindo-lhes receber as instruções dos programas e executá-las. Seria como quebrar essas instruções, já em baixo nível, em diversas instruções mais próximas do hardware (as instruções contidas no microcódigo do processador). Como característica marcante, essa arquitetura contém um conjunto grande de instruções, a maioria delas em um elevado grau de complexidade.

Examinando do ponto de vista um pouco mais prático, a vantagem da arquitetura CISC é que já temos muitas das instruções guardadas no próprio processador, o que facilita o trabalho dos programadores de linguagem de máquina; disponibilizando, assim, praticamente todas as instruções que serão usadas em seus programas. Os processadores CISC têm a vantagem de reduzir o tamanho do código executável por já possuírem muito do código comum em vários programas, em forma de uma única instrução.

Porém, do ponto de vista da performance, os CISCs têm algumas desvantagens em relação aos RISCs, entre elas a impossibilidade de se alterar alguma instrução composta para se melhorar a performance. O código equivalente às instruções compostas do CISC pode ser escrito nos RISCs da forma desejada, usando um conjunto de instruções simples, da maneira que mais se adequar. Sendo assim, existe uma disputa entre tamanho do código x desempenho.

RISC X CISC

Sempre houve uma grande polêmica a respeito de qual dessas plataformas é melhor. No começo da década de 1980, a tendência era construir chips com conjuntos de instruções cada vez mais complexos. Alguns fabricantes, porém, resolveram seguir o caminho oposto, criando o padrão RISC. Ao contrário dos complexos CISC, os processadores RISC são capazes de executar apenas algumas poucas instruções simples. A família SPARC, da SUN, possui cerca de 50 instruções, enquanto os VAX-11/780 têm até 303 instruções, e o Intel 80486 foi lançado com 147 instruções de máquina. Com menor quantidade de instruções e com cada uma delas tendo sua execução



otimizada, o sistema deve produzir seus resultados com melhor desempenho, mesmo considerando-se que uma menor quantidade de instruções vai conduzir a programas um pouco mais longos.

Justamente por isso, os chips baseados nessa arquitetura são mais simples e muito mais baratos. Outra vantagem dos processadores RISC é que, por terem um menor número de circuitos internos, podem trabalhar a frequências mais altas.

A ideia principal é que, apesar de um processador CISC ser capaz de executar centenas de instruções diferentes, apenas algumas são usadas frequentemente. Poderíamos então criar um processador otimizado para executar apenas essas instruções simples que são mais usadas.

É indiscutível, porém, que em muitas tarefas os processadores CISC saem-se melhor, principalmente pelo seu grande número de recursos. Por isso, em vez da vitória de uma das duas tecnologias, atualmente vemos processadores híbridos, que são essencialmente processadores CISC, mas incorporam muitos recursos encontrados nos processadores RISC ou vice-versa.

Apesar de, por questões de marketing, muitos fabricantes ainda venderem "processadores RISC", não existe praticamente nenhum processador nos dias atuais que siga estritamente uma das duas filosofias. Tanto processadores da família x86, como o Pentium II, Pentium III e AMD Athlon, quanto processadores supostamente RISC, como o MIPS R10000 e o HP PA-8000, misturam características das duas arquiteturas, por simples questão de desempenho.

Examinando de um ponto de vista um pouco mais prático, a vantagem de uma arquitetura CISC é que já temos muitas das instruções guardadas no próprio processador, o que facilita o trabalho dos programadores, pois ele já dispõe de praticamente todas as instruções que serão usadas em seus programas. No caso de um chip estritamente RISC, o programador já teria um pouco mais de trabalho, pois, como disporia apenas de instruções simples, teria sempre que combinar várias instruções sempre que precisasse executar alguma tarefa mais complexa.

Os chips atuais são na verdade misturas das duas arquiteturas. Internamente, o processador executa apenas instruções simples. Essas instruções internas variam de processador para processador. Sobre essas instruções internas, temos um circuito decodificador, que converte as instruções complexas utilizadas pelos programas em várias instruções simples que podem ser entendidas pelo processador. Essas instruções complexas são iguais em todos os processadores usados em micros PC.

O conjunto básico de instruções usadas em micros PC é chamado de conjunto x86.

Esse conjunto é composto por um total de 187 instruções, que são as utilizadas por todos os programas. Além desse conjunto principal, alguns processadores trazem também instruções alternativas, que permitem aos programas executar algumas tarefas mais rapidamente do que seria possível usando as instruções x86 padrão.



Outra característica importante da arquitetura RISC, que a distingue da arquitetura CISC, refere-se ao modo de realizar chamadas de rotinas e passagem de parâmetros.

Os estudos sobre comportamento dos programas revelaram que chamadas de funções requerem usualmente poucos dados, mas consomem, na transferência, demorados acessos à memória em leituras e escritas. Nas máquinas CISC, a chamada de funções conduz a operação de leitura/escrita com a memória para passagem de parâmetro e recuperação de dados; nas máquinas com a arquitetura RISC, isso ocorre basicamente no processador, utilizando-se para isso mais registradores que nas máquinas CISC. Os parâmetros e variáveis são manuseados na própria UCP (Unidade Central de Processamento). A possibilidade de colocação de mais registradores na UCP é possível por causa da redução dos circuitos necessários à decodificação e execução de instruções.

Com isso, o desempenho total do processador melhora, que executa mais otimizadamente as chamadas de funções que ocorrem em quantidade apreciável na média dos programas.

Para facilitar o trabalho dos compiladores, o conjunto de instruções de máquinas CISC tende a possuir modos de endereçamento. Uma simples instrução de soma pode ser realizada com os operandos localizados de diversos modos: podem-se somar valores que estão armazenados em registradores; outra instrução pode realizar a mesma soma, com um operando na memória e outro em um registrador; ou ainda outra instrução pode realizar a operação de soma com os dois operandos armazenados na memória.

No caso das máquinas RISC, a busca por soluções mais simples conduziu à criação, de um modo geral, de dois tipos de instruções: LOAD/STORE para acesso à memória utilizando somente o modo direto, e demais operações matemáticas do processador. Essa técnica simplifica consideravelmente o projeto e a implementação das instruções, reduzindo ainda mais os ciclos do relógio necessários à sua realização.

Projetar processadores que executam várias instruções quase totalmente em paralelo é uma técnica bastante eficaz para acelerar o desempenho dos processadores, reduzindo o tempo de execução das instruções para poucos ciclos. *Pipelining* é utilizado em larga escala em arquiteturas RISC. O objetivo do projeto do processador RISC tem sido, no que se refere a essa área, completar a execução de uma instrução a cada ciclo de relógio. Há no mercado alguns métodos de medir e divulgar o desempenho de processadores de computação, bem como diversas unidades de medidas decorrentes, os quais, em conjunto, podem confundir o observador, em vez de servir de elemento básico de comparação e auxílio à tomada de decisão em algum procedimento de escolha.

Uma das unidades de medida mais conhecidas e também ambíguas é o MIPS – milhões de instruções por segundo. É ambígua porque cada processador executa uma instrução de modo diferente, e ainda porque cada um possui instruções diferentes. MIPS não é uma boa unidade de medida de comparação entre processadores RISC e CISC porque pode iludir o observador com os resultados, por causa do princípio conceitual de ambas as arquiteturas. Como as máquinas RISC possuem instruções mais simples,



tendem a consumir mais instruções de máquina em um programa do que os correspondentes processadores CISC, e isso pode mostrar um total de MIPS superior, conduzindo a uma possível conclusão errônea para os processadores RISC.

Um resumo das diferenças entre RISC e CISC pode ser visto no quadro a seguir:

Processadores RISC	Processadores CISC
Instruções simples levando 1 ciclo	Instruções complexa levando múltiplos ciclos
Apenas operações LOAD/STORE referenciam a memória	Qualquer instrução pode referenciar a memória
Altamente pipelined	Não tem pipeline, ou tem pouco
Instruções executadas pelo hardware	Instruções executadas pelo microprograma
Instruções com formato fixo	Instruções de vários formatos
Poucas instruções e modos	Muitas instruções e modos
A complexidade está no compilador	A complexidade está no microprograma
Múltiplos conjuntos de registradores	Conjunto único de registradores



Pipeline

Trata-se de uma forma de obter uma alta performance ao "partir" o processamento de uma instrução numa série de estágios, que são ligados como as estações numa linha de montagem. Essa linha de montagem para processamento de instruções tem o nome de *pipelining*. À medida que as instruções fluem ao longo do *pipeline*, o hardware em cada estágio realiza algum processamento, até que as instruções que deixam o *pipeline* são completamente processadas. A alta performance é obtida pelo paralelismo no processamento das várias instruções ao mesmo tempo, cada uma em diferentes estágios do *pipeline*.

Quando é carregada uma nova instrução, ela primeiramente passa pelo primeiro estágio, que trabalha nela durante apenas um ciclo de *clock*, passando-a adiante para o segundo estágio. A instrução continua então sendo executada sucessivamente pelo segundo, terceiro, quarto e quinto estágios do processador. A vantagem dessa técnica é que o primeiro estágio não precisa ficar esperando a instrução passar por todos os demais para carregar a próxima, e sim carregar uma nova instrução assim que se livra da primeira, ou seja, depois do primeiro pulso de *clock*. As instruções trafegam dentro do processador na ordem em que são processadas. Mesmo que a instrução já tenha sido processada ao passar pelo primeiro ou segundo estágio, terá que continuar seu caminho e passar por todos os demais. Se por acaso a instrução não for completada mesmo após passar pelos cinco,



voltará para o primeiro e será novamente processada, até que tenha sido concluída. Dessa maneira, conseguimos que o processador seja capaz de processar simultaneamente, em um único ciclo de *clock*, várias instruções que normalmente demorariam vários ciclos para serem processadas.

É assim uma técnica fundamental de processamento que, sendo inicialmente introduzida nas arquiteturas RISC, estendeu-se às CISC, estando hoje presente, por exemplo, nos processadores Intel Pentium.esses processadores possuem a vantagem de rodar código nativo DOS e Windows, mas a desvantagem de uma arquitetura já exausta.

Os diversos processadores RISC – PowerPC da IBM/Apple/Motorola, R4x00 da MIPS, SPARC da SUN, PA-RISC da HP e Alpha da DEC – estão competindo para se tornar o padrão RISC, e os desenvolvedores de software terão que escolher não só entre RISC e CISC, mas também entre os RISC. Uma questão com que já estão se defrontando muitos usuários do mundo inteiro é: sair ou não da tradicional arquitetura de processamento CISC e migrar para a arquitetura RISC?

Uma pesquisa realizada no ano passado apontou que, apesar do RISC ter melhor desempenho, pelo menos 95% dos computadores "desktop" ainda usavam CISC por dois motivos simples: CISC é mais barato e roda a maioria dos softwares que todo mundo quer usar. O primeiro motivo já não é mais verdadeiro, pois o PowerPC 601 já tira uma vantagem de preço sobre o CISC.

A decisão de mudança passa por investimentos futuros em novas máquinas, novas arquiteturas de processamento da informação, novos servidores dos sistemas corporativos de informação e também pela escolha de qual será a nova plataforma e ambiente operacional a serem utilizados por esses sistemas nos próximos anos.

Existe um número significativo de desenvolvedores de aplicações para os processadores RISC, o que pode ser um sinal seguro da tendência do mercado corporativo, principalmente com servidores de redes pesadas – ou com aplicações críticas; com isso a Intel deverá perder uma fatia do mercado no caso de servidores que vinham sendo atendidos com os processadores 486 DX2 e Pentium. Dessa forma, nos próximos anos o usuário passará a contar com uma oferta mais diversificada de plataformas de hardware poderosas com preços competitivos, além de uma enorme variedade de sistemas operacionais.

O futuro

A maior ameaça para as arquiteturas RISC e CISC pode não ser nenhuma delas (por oposição à outra), mas uma nova arquitetura denominada EPIC (*Explicit Parallel Instruction Computer*). Como se pode depreender da palavra "paralelo", a arquitetura EPIC pode executar várias instruções em paralelo umas com as outras. Essa filosofia foi criada pela Intel e é, de certa forma, a combinação das arquiteturas RISC e CISC.

A Intel e a Hewlett Packard estão a desenvolver um processador usando essa filosofia sob o nome MERCED (IA-64), e a Microsoft já está a desenvolver uma plataforma (WIN64) para ele. O processador MERCED será um processador de 64 bits.



Se essa arquitetura for bem-sucedida poderá tornar-se a maior ameaça à arquitetura RISC.

Todas as grandes marcas de fabricantes de processadores, excetuando a Sun e a Motorola, estão neste momento a comercializar produtos baseados no x86, e alguns estão apenas à espera de que o MERCED venha para o mercado. Por causa do mercado dos x86, não é provável que a arquitetura CISC desapareça num futuro próximo, mas a arquitetura RISC poderá vir a ser uma arquitetura em extinção. O futuro poderá trazer-nos processadores baseados na arquitetura EPIC, bem como mais famílias de processadores CISC, enquanto os processadores baseados em arquiteturas RISC poderão tender a desaparecer do mercado.



A princípio, os computadores eram bastante simples, o microprocessador possuía poucas instruções e apenas um ou dois modos de endereçamento. A memória principal era lenta, então era preciso diminuir o número de acessos a essa memória. A saída foi colocar microinstruções no processador. Isso o tornou cada vez mais complexo à medida que aumentava-se o número de microinstruções, o que resultou na máquina CISC.

A arquitetura RISC parece ser uma boa saída para diminuir a complexidade dos computadores. Essa máquina possui um pequeno número de microinstruções verticais. O programa é compilado e executado diretamente pelo hardware.

Cada vez mais as tecnologias RISC e CISC estão se aproximando: processadores RISC estão aumentando seu conjunto de instruções e os CISC estão adotando técnicas originalmente implementadas no RISC.

A simplificação das instruções é um grande mérito, e provavelmente continuará a influenciar futuras arquiteturas. Os princípios RISC e CISC poderão viver harmoniosamente em um único projeto. As memórias cache maiores (que diminuem a dependência dos acessos à memória) e uma melhoria na tecnologia dos compiladores diminuem ainda mais as diferenças apregoadas entre as máquinas RISC e CISC.

A diferença entre processadores RISC e CISC já não reside no tamanho nem no tipo do conjunto de instruções, mas sim na arquitetura em si. As nomenclaturas RISC e CISC já não descrevem a realidade das arquiteturas atuais. O que conta atualmente é a velocidade com que o processador consegue executar as instruções que lhe são passadas e a fiabilidade com que consegue correr o software.

Hoje em dia os fabricantes de processadores, sejam eles RISC ou CISC, estão a utilizar todos os truques de modo a melhorarem o desempenho e permitir algum avanço em relação aos seus concorrentes. Ambas as arquiteturas têm sobrevivido no mercado por razões diferentes, a arquitetura RISC pelo seu desempenho e a arquitetura CISC pela compatibilidade de software.

O futuro poderá não trazer a vitória a nenhum deles, mas sim a sua provável extinção, já que a Intel, que sempre foi a empresa líder na fabricação da arquitetura x86 (arquitetura CISC), vai abandoná-la em favor da arquitetura RISC depois de ter assinado com a HP para o projeto do Merced. A arquitetura EPIC pode então fazer com que as arquiteturas RISC e CISC se tornem obsoletas.



Referências

MACHADO, Francis B.; MAIA, Luiz P. *Arquitetura de sistemas operacionais*. 4. ed. Rio de Janeiro: LTC, 2007.

STALLINGS, Willian. *Arquitetura e organização de computadores*. 5. ed. Prentice Hall. São Paulo, 2006.

TANENBAUM. Andrew S. *Organização estruturada de computadores*. 5. ed. Rio de Janeiro: LTC, 2007.

WEBER, Raul Fernando. *Arquitetura de computadores pessoais*. 2. ed. Porto Alegre: Sagra Luzzatto, 2003.

_____. *Fundamentos de arquitetura de computadores*. 3. ed. Porto Alegre: Sagra Luzzatto, 2004.



Avalie este tópico



ANTERIOR

Estrutura da CPU - Organização de registradores - Ciclo de instrução

Biblioteca

(<https://www.uninove.br/conhec-a->

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site



Índice

Introdução às arquiteturas paralelas e tádonso=)

Ajuda?

PRÓXIMO: (<https://ava.un>

ia de FV

mia de FV

® Todos os direitos reservados

