

**UIT - UNIVERSIDADE DE ITAÚNA**  
**CIÊNCIA DA COMPUTAÇÃO**

**DAVI VENTURA CARDOSO PERDIGÃO**

# **Comparativo**

## **RISC x CISC**

**Arquitetura e Organização de Computadores II**

# Introdução

A presente pesquisa consiste em um estudo comparativo sobre as duas arquiteturas de processadores mais conhecida atualmente: RISC e CISC. O processador é um circuito eletrônico responsável por executar uma série de instruções dadas pela máquina. Essas instruções são pré-definidas e armazenadas na memória principal do computador e chegam ao processador em linguagem *assembly*, que é o padrão que ele reconhece. O processador pode receber diferentes tipos de instrução, como por exemplo operações lógicas, aritméticas e controle, inclusive, justamente aqui que as diferenças entre RISC e CISC começam a surgir. CISC é a abreviação de *Complex Instruction Set Computer*, ou seja, executa instruções complexas. Já o termo RISC é a abreviação de *Reduced Instruction Set Computer* e aqui o nome mais uma vez diz muito sobre a arquitetura do processador. Ao contrário do CISC, o RISC executa instruções reduzidas.

## Projeto

Máquinas CISC têm um conjunto de instruções grande, de tamanhos variáveis, com formatos complexos. Muitas dessas instruções são bastante complicadas, executando múltiplas operações quando uma única instrução é dada (por exemplo, é possível realizar um loop complexo usando apenas uma *assembly*). O problema básico com máquinas CISC é que um conjunto pequeno de instruções complexas torna o sistema consideravelmente mais lento.

Assim, os projetistas decidiram retornar a uma arquitetura mais simples, delegando ao compilador a responsabilidade de produzir código eficiente com esse novo conjunto de instruções. Máquinas que utilizam esta filosofia são chamadas de máquinas RISC. A arquitetura RISC tem como principal objetivo simplificar as instruções de modo que elas possam ser executadas mais rapidamente. Cada instrução executa apenas uma operação, que são todas do mesmo tamanho, tem poucos formatos, e todas as operações aritméticas devem ser executadas entre registradores (dados da memória não podem ser utilizados como operandos).

## Programação

Máquinas CISC e RISC apresentam um desempenho semelhante no quesito programação: ambas as duas levam cerca de três minutos para serem executadas.

## Compilador

Os compiladores RISC tentam manter os operandos em registros de forma que podem usar simples instruções registro-registo. Os compiladores tradicionais, por outro lado, tentam descobrir o modo de endereçamento ideal e o menor formato de instrução para fazerem os acessos à memória. Em geral, os programadores de compiladores RISC preferem o modelo de execução registro-registo de forma que os compiladores possam manter os operandos que vão ser reutilizados em registros, em vez de repetirem os acessos à memória de cada vez que é necessário um operando. Usam, por isso, LOAD's e STORE's para aceder à memória para que os operandos não sejam, implicitamente, rejeitados após terminada a execução de uma determinada instrução, como acontece nas arquiteturas que utilizam um modelo de execução memória-memória.

## Comunicação com o hardware

Os processadores baseados na computação de conjunto de instruções complexas, como no caso do CISC, contêm uma micro programação, ou seja, um conjunto de códigos de instruções que são gravados no processador, permitindo-lhe receber as instruções dos programas e executá-las, utilizando as instruções contidas na sua micro programação. Seria como quebrar estas instruções, já em baixo nível, em diversas instruções mais próximas do hardware (as instruções contidas no microcódigo do processador).

Já a arquitetura RISC é constituída por um pequeno conjunto de instruções simples que são executadas diretamente pelo hardware, onde não há a intervenção de um interpretador (microcódigo), o que significa que as instruções são executadas em apenas uma microinstrução (de uma única forma e seguindo um mesmo padrão). As máquinas RISC só se tornaram viáveis devido aos avanços de software otimizado para essa arquitetura, através da utilização de compiladores otimizados e que compensem a simplicidade dessa arquitetura.

## Desempenho

As análises efetuadas de programas compilados por máquinas de arquitetura CISC, com o tempo mostraram que os compiladores não eram mais tão espertos quanto os programadores assembly na busca de instruções de máquina complexas. O programa compilador utiliza pouco da grande quantidade de instruções e dos modos de endereçamento que estão disponíveis, pois parece ser difícil analisar o programa de alto nível. Por exemplo, para efetuar a soma entre dois operandos em uma máquina com arquitetura CISC, sendo que um dos valores está em um registrador e o outro na memória, levará um certo tempo gasto para calcular o endereço de acesso a memória. Já em uma máquina com arquitetura RISC, são usadas duas instruções (diferente da arquitetura CISC que utiliza uma instrução), mas como são instruções mais simples, sua execução é bem mais rápida fazendo com que seu tempo total seja menor.

É indiscutível, porém, que em instruções complexas os processadores CISC saem-se melhor. Por isso, ao invés 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).

## Arquitetura Híbrida

Nos chips atuais, que são na verdade misturas das duas arquiteturas, juntamos as duas coisas. Internamente, o processador processa apenas instruções simples. Estas instruções internas, variam de processador para processador, são como uma luva, que se adapta ao projeto do chip. As instruções comuns são executadas diretamente por hardware (em um único ciclo do caminho de dados), já as instruções complexas são interpretadas por meio de microcódigo. Essa arquitetura é capaz de armazenar instruções x86, sendo compatíveis com todos os programas, mas ao mesmo tempo comportando-se internamente como chips RISC.

## Equipamentos

Atualmente não se pode afirmar com 100% de certeza que um processador utiliza apenas a arquitetura CISC ou RISC, pois os modelos atuais de processadores abrigam as características de ambas as arquiteturas. Processadores ARM usados em celulares são um exemplo de uso da arquitetura RISC, outro exemplo de uso dessa arquitetura é em consoles como o Nintendo 64 e o Playstation. Com o passar dos anos, tanto a Intel quanto a

AMD perceberam que usar alguns conceitos da arquitetura RISC em seus processadores poderia ajudá-las a criar processadores mais rápidos. Porém, ao mesmo tempo, existia a necessidade de continuar criando processadores compatíveis com os antigos. A ideia então, como citado no tópico anterior, passou a ser construir chips híbridos, que fossem capazes de executar as instruções x86, sendo compatíveis com todos os programas, mas ao mesmo tempo comportando-se internamente como chips RISC, quebrando estas instruções complexas em instruções simples, que podem ser processadas por seu núcleo RISC.

## **Vantagens**

### ***RISC***

- Os processadores RISC alcançam de duas a quatro vezes a performance dos processadores CISC, usando tecnologia de semicondutor.
- As instruções construídas para um processador RISC são simples e pequenas, o que aumenta sua performance.

### ***CISC***

- Os processadores CISC têm a vantagem de reduzir o tamanho do código executável por já possuírem muitos códigos simples em vários programas, em forma de uma única instrução.
- Os processadores baseados na computação de conjunto de instruções complexas têm uma micro programação, um conjunto de códigos de instruções que são gravados no processador, permitindo receber as instruções dos programas e assim executar, utilizando as instruções contidas na sua micro programação.

## **Desvantagens**

### ***RISC***

- O desempenho de um processador RISC está relacionado diretamente ao código gerado pelo programador. Na ocasião de um código mal desenvolvido o processador pode gastar um tempo a mais na execução das instruções, isto faz com que a performance de uma máquina RISC dependa em grande parte da qualidade do código, gerado pelo programador.

## CISC

- Leva muito tempo para carregar os comandos pela sua complexidade e o modo de funcionamento que é feita uma de cada vez.
- O processador usa mais espaço no chip, e como os chips menores armazenam mais dispositivos, isso faz com que o CISC armazene menos dispositivos.

## Conclusão

Comparando as duas arquiteturas o RISC é caracterizado por ser objetivo, simples e tem um grande uso da abordagem de pipeline, e a arquitetura CISC é voltada para execução de instruções complexas com a técnica da micro programação, resultando na diminuição do tamanho dos códigos. No decorrer do tempo, o confronto dessas arquiteturas e o aperfeiçoamento de cada uma delas permitiu que houvessem melhorias significativas, mas foram encontrados e definidos limites a que cada arquitetura pode desempenhar.

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 pela sua performance 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), a vai abandonar 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 Bibliográficas

- CALDAS, Luís. **Arquitetura RISC e CISC**. Disponível em: <[http://www.luiscaldas.com.br/unip/arquitetura/Notas%20de%20aulas/SD\\_EC\\_aula\\_05\\_org.pdf](http://www.luiscaldas.com.br/unip/arquitetura/Notas%20de%20aulas/SD_EC_aula_05_org.pdf)>. Acesso em: 01 de março de 2021.
- CARRERA, Alisson *et al.* **RISC e CISC: comparação de suas técnicas**. Disponível em: <[http://ww2.deinfo.ufrpe.br/sites/ww2.deinfo.ufrpe.br/files/artigos\\_aoc/RISCxCSICv2.pdf](http://ww2.deinfo.ufrpe.br/sites/ww2.deinfo.ufrpe.br/files/artigos_aoc/RISCxCSICv2.pdf)>. Acesso em: 01 de março de 2021.
- SÉRGIO, Luiz. **Arquitetura e Organização de Computadores**. Disponível em: <<https://educapes.capes.gov.br/bitstream/capes/206151/2/apostila%20de>

[%20AOC\\_Luiz%20S%C3%A9rgio.pdf](#)>. Acesso em: 01 de março de 2021.

- STALLINGS, William. **Arquitetura e Organização de Computadores**. Tradução da 8a edição. Editora Prentice Hall Brasil, 2002.
- TANENBAUM, Andrew S. **Organização Estruturada de Computadores**. Tradução da 5a edição. Editora Prentice Hall Brasil, 2007.
- VASCONCELOS, Laércio. **Hardware Total**. 1ª Edição. Editora Makron Books, 2002