

UNIFRAN
Universidade
de Franca

Arquitetura CISC vs RISC

Atualmente, há duas tecnologias de projeto de processadores:

Sistemas com conjunto de instruções complexo (Complex Instruction Set Computers – CISC)

.Sistemas com conjunto de instruções reduzido (Reduced Instruction Set Computers – RISC)

Arquitetura CISC vs RISC

A evolução acelerada da tecnologia de semicondutores, levou a indústria a criar processadores cada vez mais velozes e propiciou o surgimento de estudos para aperfeiçoar a arquitetura de computadores.

(Monteiro pag.428)

Arquitetura CISC vs RISC

O aperfeiçoamento dessas tecnologias deram origem a um novo tipo de arquitetura, denominada RISC.

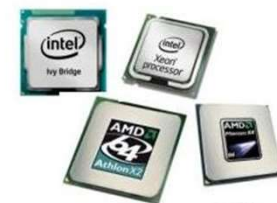
(Monteiro pag.428)

ARM®



RISC

x86



CISC

ARM significa Advanced RISC Machine

Arquitetura CISC vs RISC

A arquitetura RISC se contrapõe a arquitetura até então predominante, denominada CISC.

(Monteiro pag.428)

Arquitetura CISC vs RISC

Praticamente todos os processadores, desde os primeiros 8 bits (Intel 8080), Motorola 6800, Z-80, Intel 80486 (32-bits), minicomputadores e até os mainframes (computador de grande porte) tiveram sua arquitetura baseada nos processadores CISC.

(Monteiro pag.428)

Arquitetura CISC vs RISC

CISC... grande quantidade de instruções, com variedade de **modos de endereçamento** e poucos registradores de dados na CPU.

(Monteiro pag.428)

CISC Processors	RISC Processors
IBM 370/168	MIPS R2000
VAX 11/780	SUN SPARC
Microvax II	INTEL i860
INTEL 80386	MOTOROLA 8800
INTEL 80286	POWERPC 601
Sun-3/75	IBM RS/6000
PDP-11	MIPS R4000

Arquitetura CISC vs RISC

CISC	
VANTAGENS	DESVANTAGENS
Instruções mais complexas podem redundar em código-objeto menor, menos instruções e reflexos nos custos.	Nem sempre menos instruções acarretam em menos bits.
São microprogramados, trazem mais flexibilidade ao projeto de máquinas.	Por ser microprogramado, acarreta uma sobrecarga adicional de interpretação de cada instrução.
Muitas das instruções estão 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.	Instruções diferentes levam quantidades diferentes de período de clock para executar, o que pode tornar a máquina excessivamente lenta.

Arquitetura CISC vs RISC

A característica mais marcante dos sistemas RISC é possuir um conjunto reduzido de instruções.

*.A família SPARC da Sun possui 50 instruções. O Intel 80486 foi lançado com 147 instruções e o Pentium mais de 200.
.(Monteiro pag.431)*

Arquitetura CISC vs RISC

Com menor quantidade de instruções e todas elas com execução otimizada o sistema tem resultados com melhor desempenho, mesmo que menos instruções vão produzir programas mais longos.

(Monteiro pag.431)

Arquitetura CISC vs RISC

Em sistemas RISC todas as instruções possuem o mesmo tamanho em bits, o que facilita a sua busca que é realizada em uma única operação

(Monteiro pag.431)

Arquitetura CISC vs RISC

*Enquanto em máquinas CISC a chamada de funções para **movimentação de dados** envolve leitura e escrita com a memória RAM, em máquinas com arquitetura RISC essas operações ocorrem no processador utilizando de um **número maior de registradores**.*

(Monteiro pag.431)

Arquitetura CISC vs RISC

A arquitetura RISC possibilita a colocação de um maior número de registradores na CPU, devido a redução dos circuitos necessários para decodificação e execução de instruções.

(Monteiro pag.431)

Arquitetura CISC vs RISC

A característica mais relevante da arquitetura RISC é o uso altamente produtivo de pipeline.

Em função do formato simples e tamanho único (instruções do mesmo tamanho).

(Monteiro pag.431)

Arquitetura CISC vs RISC

RISC	
VANTAGENS	DESVANTAGENS
Executam instruções mais rápido porque seu método de codificação usa menos bits, reduzindo o tempo.	Menos instruções requerem que mais instruções sejam executadas.
As instruções são executadas diretamente pelo Hardware e não por um programa.	Falta de compatibilidade com versões anteriores.
Máquinas RISC são mais baratas e mais rápidas do que as CISC.	Requer sistema de memória rápida para alimentar suas instruções.
Simplicidade do Hardware.	A performance depende diretamente do código gerado pelo programador.

* *microarquitetura*
* *microinstruções*

Arquitetura CISC vs RISC

*Quando temos estágios que executam instruções do mesmo tamanho e ao mesmo tempo, temos uma “**linha de produção**” equilibrada e produtiva, porque nenhum estágio tem que esperar o término do outro, todos trabalham na mesma “**passada**”.*

(Monteiro pag.431)

Arquitetura CISC vs RISC

*A conclusão final sobre as duas arquiteturas é que **processadores híbridos** são a melhor opção.*

- projetos RISC se beneficiam com características CISC

- projetos CISC se beneficiam com características RISC

(Stallings 5ª-pag.498)

Arquitetura CISC vs RISC

Item	RISC	CISC
Acesso à memória	Load e Store	Qualquer Instrução
Registradores	Centenas	Dezenas
Operandos	Até 3	1 ou 2
Complexidade	No compilador	No código
Uso do pipeline	Intensamente	Moderadamente
Instruções	Poucas / simples / mesmo tamanho	Muitas / complexas / tamanhos variados
Frequências	Mais altas	Mais baixas
Programas	Grandes e complexos	Pequenos e simples
hardware	Barato e simples	Caro e complexo
Controle	Hardwired	Micro programação
CPI	Um	Vários

- Processadores híbridos são essencialmente processadores CISC (para cuidar das instruções mais complexas) com núcleo RISC (para cuidar das instruções mais simples)
- Conversão de códigos de programas de uma arquitetura CISC para a arquitetura RISC leva a um aumento de código

CPI:Ciclos por Instrução

Arquitetura CISC vs RISC

Tabela Comparativa das Arquiteturas CISC E RISC		
CARACTERISTICAS	CISC	RISC
INSTRUÇÕES POR CICLOS	<i>Instruções complexas executadas em vários ciclos</i>	<i>Instruções simples executadas em um ciclo</i>
ACESSO À MEMORIA	<i>Qualquer instrução pode referenciar a memória</i>	<i>Apenas operações LOAD/STORE em memória</i>
PIPELINE	<i>Pouco ou nenhum pipeline</i>	<i>Uso Intenso de pipeline</i>
EXECUÇÃO DAS INSTRUÇÕES	<i>Instruções executadas pelo hardware</i>	<i>Instruções interpretadas pelo próprio programa</i>
FORMATO DAS INSTRUÇÕES	<i>Instruções com formato variável</i>	<i>Instruções com formato fixo</i>
QUANTIDADE DE INSTRUÇÕES E MODOS DE ENDEREÇAMENTO	<i>Varias instruções e modos de endereçamento</i>	<i>Poucas instruções e modos de endereçamento</i>
COMPLEXIDADES DO SISTEMA	<i>Complexidade está no microprograma</i>	<i>Complexidade está no compilador</i>
REGISTRADORES	<i>Conjunto de registradores único</i>	<i>Múltiplos conjuntos de registradores</i>