



Arquitetura e Organização de computadores

ENGENHARIA DA COMPUTAÇÃO – UFC/SOBRAL

Prof. Danilo Alves
danilo.alves@alu.ufc.br

PROJETO DE ARQUITETURA

- Problema 1: Como construir computadores de baixo custo capazes de executar todas as instruções complexas de máquinas de alto desempenho, muito mais caras?
- Uma implementação em hardware puro (sem interpretação) é usada somente nos computadores mais caros.
- Uma implementação com interpretador de instrução (por software) é usada em computadores mais baratos.



PROJETO DE ARQUITETURA

- Vantagens do interpretador em relação ao hardware puro
 - Capacidade de corrigir no campo de software eventuais erros na implementação de instruções
 - Oportunidade de incorporar novas instruções nas máquinas já existentes
 - Projeto estruturado que permite o desenvolvimento, teste e documentação de instruções complexas de maneira eficiente. Pode substituir implementações antigas de instruções.
 - Armazenamento das microinstruções do interpretador em memórias read-only (ROM), chamadas de memória de controle, muito mais rápidas do que as memórias convencionais.



PROJETO DE ARQUITETURA

- Problema 2: O uso da interpretação permitiu a **criação de um conjunto grande de instruções** de importância discutível e que eram difíceis e caras para serem implementadas diretamente por hardware (circuitos muito complexos).



CISC

- CISC - Complex Instruction Set Computer
 - Tecnologia antiga e usada para famílias de computadores compatíveis a nível de software.
 - Número maior de instruções (~200 a 300 instruções).
 - Uso extensivo de interpretação (principalmente para modelos mais baratos).
 - São fáceis de programar e permitem um uso eficiente de memória.



CISC

- Os primeiros computadores tinham poucas instruções, todas implementadas em hardware.
- Maurice Wilkes (1951) introduziu a microprogramação, que permite um conjunto grande de instruções de máquina (no nível convencional), usando um hardware simples capaz de executar apenas as chamadas microinstruções.



CISC

- Cada instrução de máquina no nível convencional é interpretada e pode dar origem à execução de muitas instruções no nível mais baixo.
- Os computadores que usam microprogramação são ditos CISC.
- Exemplos: IBM 360, DEC VAX, Motorola 68030, família Intel como 8088, 80386, Pentium etc.
- Mudanças recentes na tecnologia de software e hardware forçou uma reavaliação em termos de arquitetura.
- CISC mais modernos têm implementado alguns princípios RISC (Reduced Instruction Set Computer).
- Instruções de tamanho variável de acordo com o modo de endereçamento.



CISC

- Instruções que requerem múltiplos ciclos de clock para executar.
- Número pequeno de registradores de propósito geral.
- Possuem muitos registradores de propósito específico tais como, apontadores de pilha, tratadores de interrupção, etc.



CRÍTICAS À ARQUITETURA CISC

- Os computadores CISC estão ficando demasiadamente complexos, com várias centenas de instruções e dezenas de modos de endereçamento.
- Um grande questionamento é se muitas das instruções complexas são de fato necessárias nos programas “típicos”.
- Levantamentos feitos mostram os comandos mais usados pelos programas. Eles se concentram em alguns poucos comandos, em geral bastante simples.
- Um microprograma complexo significa maior tempo para decodificar e executar uma instrução complexa, muitas das quais raramente são usadas.



RISC

- Processador com pequeno número de instruções muito simples.
- Instruções capazes de serem executadas em um único ciclo do caminho de dados.
- Os primeiros projetos RISC foram desenvolvidos nos anos 70 e 80 pelas universidades de Stanford e Berkeley, respectivamente.
- Mesmo que uma máquina RISC precisasse de 4 ou 5 instruções para fazer o que uma máquina CISC faria com apenas 1 instrução, se a instrução RISC fosse 10 vezes mais rápida (só hardware) a máquina RISC venceria.



RISC

- Menor quantidade de instruções e tamanho fixo
- Execução otimizada de chamada de funções
- Menor quantidade de modos de endereçamento
- Modo de execução com pipelining



RISC

- Execução em um ciclo de clock. Esta característica é resultado da otimização de cada instrução, aliada a uma técnica chamada de Pipelining - Pipelining é uma técnica que permite execução simultânea de partes, ou estágios, de instruções, tornando o processo mais eficiente;
- Grande número de registradores para evitar uma quantidade elevada de interações com a memória.
- Instruções menos complexas
- Unidade de controle hardwarizada
- Baixa capacidade de endereçamento para operações de memória, com apenas duas instruções básicas, LOAD e STORE



RISC

- Não há microprograma para interpretar as instruções.
- Existe um conjunto reduzido de instruções simples RISC parecidas com as microinstruções da arquitetura CISC.
- O código gerado pelos compiladores é constituído de instruções simples desse conjunto reduzido. Essas instruções são armazenadas na memória RAM, buscadas e executadas diretamente em hardware na CPU, sem nenhuma interpretação.
- As instruções são executadas na sua maioria em apenas um ciclo da máquina.



CISC x RISC

- Máquinas RISC conseguem maior MIPS que máquinas CISC .
- Para instruções de ponto flutuante, as máquinas RISC necessitam de hardware especial para terem desempenho equivalente às CISC.
- Múltiplos conjuntos de registradores das RISC contribuem para um maior desempenho.
- Alguns programas criados em linguagens de alto nível precisam de uma biblioteca de procedimentos para rodar eficientemente em máquinas RISC, o que pode ser realizado via microcódigo nas CISC.



CISC x RISC

- As arquiteturas RISC são mais fáceis de produzir devido à sua simplicidade e menor número de transistores.
- Compiladores para RISCs são mais complexos pois precisam usar eficientemente os recursos de pipeline e de alocação de registradores.



CISC x RISC

RISC	CISC
Instruções simples durando 1 ciclo	Instruções complexas durando vários ciclos
Apenas LOAD/STORE referencia a memória	Qualquer instrução pode referenciar a memória
Alto uso de Pipeline	Baixo uso de Pipeline
Instruções executadas pelo hardware (<i>Hardwired</i>)	Instruções interpretadas pelo microprograma (Microprogramação)
Instruções com formato fixo	Instruções de vários formatos
Poucas instruções e modos de endereçamento	Muitas instruções e modos de endereçamento
Múltiplos conjuntos de registradores	Conjunto único de registradores
A complexidade está no compilador	A complexidade está no microprograma



ATIVIDADE

1. Explique cada um dos termos seguintes com suas próprias palavras:

- a. Tradutor.
- b. interpretador.
- c. Máquina virtual.

2. É concebível um compilador gerar saída para o nível de microarquitetura em vez de para o nível ISA? Discuta prós e contras dessa proposta.



ATIVIDADE

3. Considere um computador multinível no qual todos os níveis são diferentes. Cada nível tem instruções que são m vezes mais poderosas do que as do nível abaixo dele; isto é, uma instrução de nível r pode fazer o trabalho de m instruções de nível $r - 1$. Se um programa de nível 1 requer k segundos para executar, quanto tempo levariam programas equivalentes nos níveis 2, 3 e 4 admitindo que são necessárias n instruções de nível r para interpretar uma única instrução de nível $r + 1$?
4. Em que sentido hardware e software são equivalentes? E não equivalentes?
5. Escolha três pessoas que você considera serem as mais influentes na criação do moderno hardware de computadores e redija um curto relatório que descreva suas contribuições e o motivo de escolhê-las.
6. Escolha três pessoas que você considera serem as mais influentes na criação do moderno software de sistemas de computação e redija um curto relatório que descreva suas contribuições e o motivo de escolhê-las.



ATIVIDADE

7. Explique os tipos de paralelismo.
8. Cite os três elementos principais que formam o processador.
9. Explique as diferenças entre as arquiteturas RISC e CISC, apresentando as vantagens e desvantagens de cada abordagem.

