

Arquiteturas RISC e CISC

Profa. Débora Christina Muchaluat Saade

debora@midiacom.uff.br

<http://www.ic.uff.br/~debora/fac>

Maiores Avanços nos Computadores

Fundamentos de Arquiteturas de Computadores

- ✓ **Conceito de família**
 - *IBM System/360 1964*
 - *DEC PDP-8*
 - *Separa arquitetura de implementação*
- ✓ **Unidade de controle microprogramada**
 - *Idéia de Wilkes 1951*
 - *Produzida pela IBM S/360 1964*
- ✓ **Memória cache**
 - *IBM S/360 modelo 85 1969*

Maiores Avanços nos Computadores

Fundamentos de Arquiteturas de Computadores

- ✓ **Microprocessadores**
 - *Intel 4004 1971*
- ✓ **Pipelining**
 - *Introduz paralelismo no ciclo de busca e execução*
- ✓ **Múltiplos processadores**

Desenvolvimento de Máquinas CISC

Fundamentos de Arquiteturas de Computadores

- ✓ *Complex Instruction Set Computer*
- ✓ **Custos de software excedem os custos de hardware**
- ✓ **Linguagens de alto nível cada vez mais complexas**
- ✓ **Gap semântico**
- ✓ **Leva a:**
 - *Grande conjunto de instruções*
 - *Maior número de modos de endereçamento*
 - *Implementações em hardware de instruções de linguagem de alto nível*
 - Exemplo, CASE no VAX

Objetivos da Arquitetura CISC

Fundamentos de Arquiteturas de Computadores

- ✓ **Facilitar o desenvolvimento de compiladores**
- ✓ **Melhorar a eficiência de execução**
 - *Operações complexas em microcódigo*
- ✓ **Suportar linguagens de alto nível mais complexas**

Características de Execução

Fundamentos de Arquiteturas de Computadores

- ✓ **Operações executadas**
- ✓ **Operandos utilizados**
- ✓ **Seqüência de execução**
- ✓ **Estudos foram realizados baseados em programas escritos em linguagens de alto nível**
- ✓ **Estudos dinâmicos realizados durante a execução de programas**
 - *Mesmo conjunto de programas foi compilado e executado no VAX, PDP-11 e Motorola 68000*

Operações

- ✓ **Atribuições**
 - *Movimento de dados*
- ✓ **Instruções condicionais (IF, LOOP)**
 - *Controle da sequência*
- ✓ **Chamada e retorno de procedimento consome muito tempo**
- ✓ **Algumas instruções da linguagem de alto nível levam a muitas operações de linguagem de máquina**

Frequência Dinâmica Relativa Ponderada de Operações de Linguagens de Alto Nível

Fundamentos de Arquiteturas de Computadores

	Ocorrência Dinâmica		Ponderada por instrução de máquina		Ponderada por referência à memória	
	Pascal	C	Pascal	C	Pascal	C
ASSIGN	45%	38%	13%	13%	14%	15%
LOOP	5%	3%	42%	32%	33%	26%
CALL	15%	12%	31%	33%	44%	45%
IF	29%	43%	11%	21%	7%	13%
GOTO	—	3%	—	—	—	—
OTHER	6%	1%	3%	1%	2%	1%

Operandos

- ✓ **Principalmente variáveis locais escalares**
- ✓ **Otimização deve se concentrar no acesso às variáveis locais**

	Pascal	C	Média
Constante Inteira	16%	23%	20%
Variável Escalar	58%	53%	55%
Array/Structure	26%	24%	25%

Chamadas de Procedimentos

- ✓ **Consome muito tempo**
- ✓ **Depende do número de parâmetros**
- ✓ **Depende do nível de aninhamento**
- ✓ **Maioria dos programas não executa várias chamadas seguidas de vários retornos**
- ✓ **Maioria das variáveis são locais**

Implicações

- ✓ Suporte mais eficaz para linguagens de alto nível pode ser obtido pela otimização do desempenho das características mais utilizadas e responsáveis por maior consumo de tempo
- ✓ Número grande de registradores
 - *Referência aos operandos*
- ✓ Projeto cuidadoso dos pipelines
 - *Predição de desvio etc.*
- ✓ Conjunto simplificado (reduzido) de instruções

Uso de um Grande Banco de Registradores

✓ Solução baseada em software

- *Compilador responsável por otimizar uso dos registradores*
- *Tenta alocar, nos registradores, variáveis que serão mais usadas durante um determinado período de tempo*
- *Necessita de análise sofisticada dos programas*

✓ Solução baseada em hardware

- *Utilizar um número maior de registradores*
- *Permite que mais variáveis possam ser armazenadas em registradores*

Registradores para Variáveis Locais

- ✓ **Maioria das referências a variáveis locais, então, devem ser armazenadas em registradores**
- ✓ **Reduz acesso à memória**
- ✓ **A definição de *local* muda com cada chamada a procedimento**
- ✓ **Variáveis locais devem ser armazenadas na memória para que registradores sejam utilizados pelo procedimento**
- ✓ **Parâmetros devem ser passados para procedimento chamado**
- ✓ **Resultados devem ser retornados pelos procedimentos**
- ✓ **Valores das variáveis salvos na memória devem ser retornados aos registradores**

Variáveis Globais

- ✓ **Alocadas pelo compilador na memória**
 - *Ineficiente para variáveis usadas frequentemente*
- ✓ **Conjunto de registradores para variáveis globais**

Otimização do Uso de Registradores pelos Compiladores

Fundamentos de Arquiteturas de Computadores

- ✓ **Suponha um pequeno número de registradores (16-32)**
- ✓ **Otimização do uso deve ser função do compilador**
- ✓ **Programas em linguagem de alto nível não fazem referências explícitas a registradores**
- ✓ **Atribui registrador simbólico ou virtual a cada variável candidata**
- ✓ **Mapeia número ilimitado de registradores simbólicos em número fixo de registradores reais**

Otimização do Uso de Registradores pelos Compiladores

Fundamentos de Arquiteturas de Computadores

- ✓ **Registradores simbólicos que não se sobrepõem podem compartilhar registradores reais**
- ✓ **Caso se esgotem os registradores reais disponíveis, algumas variáveis são alocadas na memória**

RISC

- ✓ *Reduced Instruction Set Computer*

- ✓ **Características importantes**
 - *Conjunto limitado e simples de instruções*
 - *Operações de registrador para registrador*
 - *Modos simples de endereçamento*
 - *Otimização de pipeline*

Por que CISC (1)?

✓ Simplificação do compilador?

- *Controvérsias ...*
- *Instruções de linguagem de máquina são mais complexas de se descobrir onde utilizar*
- *Otimização mais difícil*

✓ Programas menores?

- *Programas usam menos memória mas ...*
- *Memória se tornou mais barata*
- *Podem parecer menores em forma simbólica, mas podem ocupar mais bits*
 - Maior número de instruções requer mais bits para código de operação

Por que CISC (2)?

- ✓ **Programas mais rápidos?**
 - *Tendência em usar instruções mais simples*
 - *Unidade de controle mais complexa e mais memória para microprograma fazem com que aumente o tempo de execução das instruções mais simples*

- ✓ **Não está claro que a arquitetura CISC é a solução apropriada**

Características RISC

- ✓ **Uma instrução por ciclo de máquina (tempo para buscar dois operandos em registradores, executar uma operação de ULA e armazenar resultado em registrador)**
- ✓ **Operações de registrador para registrador**
- ✓ **Poucos e simples modos de endereçamento**
- ✓ **Poucos e simples formatos de endereçamento**
- ✓ **Hardwired ao invés de microprograma**
- ✓ **Formato fixo das instruções**
- ✓ **Compiladores devem despende mais tempo e esforço**

RISC x CISC

- ✓ **Muitos projetos atualmente incorporam características CISC e RISC**
- ✓ **PowerPC – RISC que incorpora características CISC**
- ✓ **Pentium II – CISC que incorpora características RISC**

Controvérsia

✓ **Quantitativa**

- *Comparar tamanho dos programas e velocidades de execução*

✓ **Qualitativa**

- *Examinar suporte a linguagem de alto nível e uso ideal da tecnologia VLSI*

✓ **Problemas**

- *Não existem duas máquinas RISC e CISC que sejam diretamente comparáveis*
- *Não existe um conjunto definitivo de programas para teste*
- *Difícil separar efeitos do hardware dos efeitos do compilador*
- *Maioria das comparações realizadas em máquinas experimentais e não comerciais*
- *Maioria das máquinas comerciais possuem uma mistura de características*
 - Convergência das tecnologias₂₂

Comparação de processadores

Fundamentos de Arquiteturas de Computadores

Characteristic	Complex Instruction Set (CISC) Computer			Reduced Instruction Set (RISC) Computer		Superscalar		
	IBM 370/168	VAX 11/780	Intel 80486	SPARC	MIPS R4000	PowerPC	Ultra SPARC	MIPS R10000
Year developed	1973	1978	1989	1987	1991	1993	1996	1996
Number of instructions	208	303	235	69	94	225		
Instruction size (bytes)	2-6	2-57	1-11	4	4	4	4	4
Addressing modes	4	22	11	1	1	2	1	1
Number of general-purpose registers	16	16	8	40 - 520	32	32	40 - 520	32
Control memory size (Kbits)	420	480	246	—	—	—	—	—
Cache size (KBytes)	64	64	8	32	128	16-32	32	64