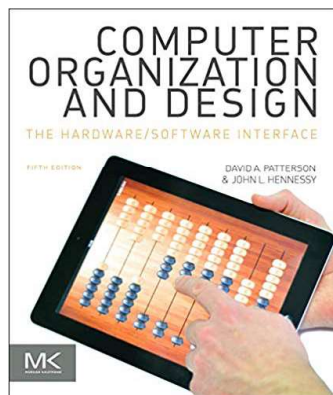


Aula 10 - Risc × Cisc

Prof. Dr. Clodoaldo A. de Moraes Lima

Material baseado no livro “Patterson, David A., Hennessy, J. L. - Computer Organization And Design: The Hardware/Software Interface”



Introdução

- Evolucionariamente, discussões foram criadas sobre o projeto do conjunto de instruções.
- Aumentar ou reduzir a complexidade do conjunto de instruções?
- Tornar compiladores mais complexos ou mais simples?
- Embate CISC x RISC.

Arquiteturas CISC e RISC

CISC (Complex Instruction Set Computing):

- Conjunto de instruções mais complexo.
- Traduz mais fielmente as instruções de alto-nível.

RISC (Reduced Instruction Set Computing):

- Possui um número de instruções reduzido.
- Instruções mais simples.

CISC: Complex Instruction Set Computer

- Até a década de 1960, os softwares de sistema eram escritos em linguagem de montagem (incluem-se os sistemas operacionais)
- Era necessário dar muito suporte de hardware à programação
 - Para substituir seqüências de operações aritméticas primitivas
 - Para métodos alternativos indiretos de acesso à memória
 - Para operações aritméticas repetitivas
 - Para dar suporte à chamada de procedimentos e passagem de parâmetros
 - Para dar suporte ao sistema operacional
 - Para dar suporte a sistemas multiprocessados

CISC: Complex Instruction Set Computer

- Conjuntos de instrução cada vez mais sofisticados, com
 - Muitas instruções, de tipos diferentes (100 a 300)
 - Muitos modos de endereçamento (8 a 20)
- Crença de que aumento de desempenho poderia ser obtido por meio de instruções mais complexas

CISC: Complex Instruction Set Computer

Exemplos de arquiteturas CISC

Nome (ano)	# instruções	# modos de end.
VAX-11/780 (1978)	303	16
Motorola MC68020	109	18
Intel 386	111	8
Intel 486	> 117	11

Efeitos Colaterais(1):

- Necessidade de especificar diferentes parâmetros em instruções distintas, levando a instruções de vários comprimentos (usualmente, em bytes ou 16 bits)
 - Comprimento totalmente variável (codificação de Huffman)
 - Codificação binária
 - Codificação múltipla do byte ou da palavra (bom aproveitamento da memória)
- Vários formatos de instruções

CISC: Complex Instruction Set Computer

Efeitos Colaterais(1):

- Instruções de vários comprimentos (no mesmo conjunto)

Nome	Tamanho das instruções
VAX-11/780	2 a 57 bytes
Motorola MC68000	1 a 5 palavras de 16 bits
Intel 8086	1 a 6 bytes
Intel 432	6 a 321 bits

Efeitos Colaterais (2):

- Grande complexidade do projeto
 - tempo de projeto mais longo
 - maior custo de desenvolvimento
 - maior possibilidade de erros de projeto
 - dificuldade no projeto do bloco de controle (microprogramado, com até 400 Kbytes de microcódigo)

CISC: Complex Instruction Set Computer

Pergunta:

“Será que as instruções extras realmente resultam em aumento da velocidade de execução?”

Arquiteturas CISC e RISC

		MASM (%)	TurboC (%)	Lotus 1-2-3 (%)	Média (%)
Acesso a operando	Memória	37	43	43	41
	Imediato	7	11	5	8
	Registrador	55	46	52	51
Acesso à memória	indireto	12	9	15	12
	direto	36	18	34	30
	Displacement (relativo)	52	73	51	58
Movimentados	MOV	30	30	21	29
	PUSH/POP	12	18	8	12
	LEA	3	6	0	3
Aritméticas e Lógicas	CMP	9	3	3	7
	SAL/SHR/RCR	0	3	12	5
	INC/DEC	3	3	3	3
	ADD	3	3	3	3
	OR/XOR	1.5	4.5	3	3
Controle e Chamada	JMP	3	1.5	1.5	2
	LOOP	0	0	12	4
	CALL/RET	3	6	3	4
	Desvio Condicional	12	12	6	10

Arquiteturas CISC e RISC

- Com o passar do tempo, conjunto de instruções ficou cada vez maior e complexo (CISC - complex instruction set computing):
 - Simplificação dos compiladores; e
 - Melhora no desempenho.
- Com instruções mais próximas de linguagem de alto-nível, tarefa dos compiladores fica mais simples.

Problema

Conjunto de instruções grande torna complexo compilador encontrar o local onde utilizar cada uma.

Expectativa com o uso da arquitetura CISC:

- Obtenção de programas menores e mais rápidos;
- Programas menores → Menor consumo de memória.
- Menor número de instruções = Menor número de bytes de instruções que precisam ser buscados.
- Programas menores ocupam menos páginas na memória, o que reduz a falta de páginas.

Arquiteturas CISC e RISC

Na prática:

- Programas compilados para arquitetura CISC não possuem garantia que serão menores que os compilados para RISC.

	(Patterson, 1982a) 11 programas C	(Katevenis, 1983) 12 programas C	(Heath, 1984) 5 programas C
RISC I	1,0	1,0	1,0
VAX-11/780	0,8	0,67	
M68000	0,9		0,9
Z8002	1,2		1,12
PDP-11/70	0,9	0,71	

RISC: Reduced Instruction Set Computer

- Questões que levaram à arquitetura RISC
 - 1 O efeito das instruções complexas
 - 2 Uso eficaz dos transistores na implementação em circuito integrado VLSI
 - 3 O overhead do controle microprogramado
 - 4 O uso de compiladores

RISC: Reduced Instruction Set Computer

1 O efeito das instruções complexas

- CISCs possuem um formato curto para as instruções mais freqüentes
- RISCs simplesmente evitam instruções pouco usadas (somente são implementadas instruções que serão usadas freqüentemente)

RISC: Reduced Instruction Set Computer

- ① Uso eficaz dos transistores na implementação em circuito integrado VLSI
 - “It was found that hardware support for complex instructions is not the most effective way of utilizing the transistors in a VLSI processor” (Katavenis, 1985)
 - tamanho do chip/complexidade \times velocidade de funcionamento (compromisso)
 - CISCs: bloco de controle ocupa mais de 50% da área do chip
 - RISCs: bloco de controle ocupa menos de 20% da área do chip

RISC: Reduced Instruction Set Computer

- ① O overhead do controle microprogramado
 - Primeiros CISCs
 - usam controle microprogramado
 - Memória principal era lenta, feita de núcleos magnéticos
 - ROM para armazenar o controle seria bem mais rápida que a memória principal
 - RISCs
 - Era VLSI, memória semicondutora barata e rápida
 - Controle hardwired, mais rápido

RISC: Reduced Instruction Set Computer

- 1 O uso de compiladores
 - Compiladores não conseguem tirar proveito adequado das instruções mais complexas (instruções mais "exóticas" são raramente usadas)
 - Compiladores capazes de usar as características RISC têm sido desenvolvidos e aperfeiçoados

RISC: Características

- 1 Poucas instruções, instruções simples
- 2 Instruções rápidas, com um throughput próximo de 1 (uma instrução executada por ciclo de relógio)
- 3 Todas as instruções do mesmo tamanho (com pouca variação de formato)
- 4 Poucos modos de endereçamento
- 5 Número razoável de registradores de propósito geral (tip. 32)

Obs: throughput pode ser traduzido por "vazão".

RISC: Características (cont)

- 1 Acesso à memória somente com LOAD e STORE
- 2 Uso de instruções compare-and-branch
- 3 Operações lógicas e aritméticas entre registradores (instruções com três endereços de registrador)
- 4 Poucos modos de endereçamento
- 5 Poucos tipos de dados
- 6 Dados e instruções em memórias (caches) separadas (arquitetura Harvard)
- 7 Uso de pipeline (simplificação do bloco de controle)
- 8 Bloco de controle hardwired (ao invés de microprogramado)

RISC: Características (descrição)

- Uma instrução por ciclo de clock.
- Operações de registrador para registrador.
- Modos de endereçamento simples.
- Formatos das instruções simples.

RISC: Características (descrição)

- 1 - Uma instrução por ciclo de clock.
 - 1 ciclo de clock : Tempo para buscar dois operandos em registradores, executar uma operação na ULA e armazenar o resultado em um registrador.
- 2 - Operações de registrador para registrador.
 - Somente operações de carga e armazenamento para acesso à memória.
 - Simplificação do projeto, consequentemente, a unidade de controle.
 - Operações registrador-registrador são particulares aos projetos de arquitetura RISC.

RISC: Características (descrição)

- Utilização de modos de endereçamento simples.
 - Quase todas as instruções utilizam um modo específico de endereçamento (ex.: por registrador);
 - Outros modos de endereçamento podem ser incluídos (ex.: deslocamento, relativo ao PC, etc).
 - Simplificação do conjunto de instruções e unidade de controle.
- Utilização de formatos de instruções simples.
 - Um único formato de instrução ou formatos de instruções simplificados;
 - Tamanhos e posições das instruções são fixos e alinhados de acordo com o tamanho de uma palavra;
 - Busca de instruções é simplificada devido ao seu tamanho.

Benefícios da abordagem RISC:

- Relacionados ao desempenho:
 - Otimizações podem ser executadas pelo compilador: Instruções mais simples oferecem maior oportunidade de melhorias pelo compilador.
 - Instruções simples = Unidade de controle simples.
 - Conjunto reduzido de instruções = pipeline mais eficiente.
 - Maior suscetibilidade a interrupções. Interrupções são verificadas sempre entre instruções elementares.

Benefícios da abordagem RISC:

- Relacionados a implementação:
 - Quando VLSI (very-large-scale integration) é utilizado, o projeto e implementação do processador são alterados.
 - VLSI permite que um processador seja alocado em uma única pastilha.
 - Duas motivações principais para o uso da arquitetura RISC:

Relacionados a implementação:

Duas motivações principais para o uso da arquitetura RISC:

- 1. Desempenho
 - Atrasos de transmissão são minimizados;
 - Atividades mais frequentes, conseqüentemente mais simples, são priorizadas;
 - Processadores RISC foram projetados considerando tal cenário.
 - Tamanho do microcódigo de controle (em % da área):
 - Microprocessador comum: 50%.
 - RISC: 6%.

Relacionados a implementação:

Duas motivações principais para o uso da arquitetura RISC:

- 2. Tempo de projeto e implementação
 - Projetar circuitos VLSI é extremamente complicado;
 - Necessidade de se reprojeter tudo a cada novo microprocessador;
 - Facilidade RISC: Arquitetura com um conjunto reduzido de instruções facilita o processo.

RISC: Reduced Instruction Set Computer

IBM 801: O Primeiro Computador RISC (1975-1979)

- 120 instruções (todas com 32 bits)
- 32 registradores de propósito geral
- Acesso à memória somente com LOAD e STORE
- Operações lógicas e aritméticas entre registradores (instruções com três endereços de registrador)
- Dois modos de endereçamento
- Compilador dedicado (responsável pelas otimizações no código)
- Pipeline de 4 estágios
- Implementação discreta (chips SSI/MSI em tecnologia ECL)

RISC: Reduced Instruction Set Computer

RISC I/II: Protótipos da UC Berkeley, California (1982-1983)

- Tese de doutorado de Katevenis, orientado por David Patterson
- RISC I: protótipo fabricado em um único chip em 1982
- RISC II: protótipo fabricado em um único chip em 1983
- RISC II deu origem ao processador comercial Sparc da SUN Microsystems (www.sun.com)

RISC: Reduced Instruction Set Computer

MIPS: Protótipo da Stanford University, California (1983)

- John Hennessy, Mark Horowitz e outros
- Protótipo fabricado em um único chip em 1983
- Originou o processador comercial MIPS, da MIPS Computer System Corporation (www.mips.com)
- A MIPS foi co-fundada por John Hennessy (in sabbatical)

Obs:

Em 2000, John Hennessy foi nomeado reitor da Stanford University.

RISC: Reduced Instruction Set Computer

RISC I, RISC II e MIPS: Características

características	RISC I	RISC II	MIPS
registradores	78	138	16
instruções	31	39	55
Modos de endereçamento	2	2	2
Formatos de instrução	2	2	4
Estágios de pipeline	2	3	5

RISC: Reduced Instruction Set Computer

RISCs Comerciais

- SUN Sparc
- MIPS R2000, R3000, R4000, R5000
- Motorola MC88100 (lançado em 1988)
 - 59 instruções, todas com 32 bits
 - 32 registradores de propósito geral
 - Operações lógicas e aritméticas entre registradores (instruções com três endereços de registrador)
 - Acesso à memória somente com LOAD e STORE
 - Arquitetura tipo “Harvard” (memória de dados e memória de instruções)
 - Pipeline de 4 estágios

Resumo Características

Muitas instruções;

CISC:

- Instruções complexas;
- Microcodificação das instruções;
- Alta capacidade de endereçamento para operações de memória;
- Número reduzido de registradores.

Resumo Características

Conjunto reduzido de instruções;

RISC:

- Instruções menos complexas;
- Unidade de controle em hardware;
- Baixa capacidade de endereçamento para operações de memória, com apenas duas instruções básicas: Load e Store.
- Grande número de registradores.

Resumo Características

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