



Universidade Federal de Santa Catarina
Centro Tecnológico
Programa de Pós-Graduação em Ciência da Computação

Arquitetura de Computadores

INE 641400

Aula 1

Arquiteturas CISC e RISC: histórico e características

Prof. José Luís Güntzel
guntzel@inf.ufsc.br

www.inf.ufsc.br/~guntzel/ine641400/ine641400.html

Arquiteturas CISC e RISC

► 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

Arquiteturas CISC e RISC

► 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

Arquiteturas CISC e RISC

► 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

Arquiteturas CISC e RISC

► CISC: Complex Instruction Set Computer

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

Arquiteturas CISC e RISC

► 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

Arquiteturas CISC e RISC

► CISC: Complex Instruction Set Computer

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)

Arquiteturas CISC e RISC

► 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	índireto	12	9	15	12
	direto	36	18	34	30
	Displacement (relativo)	52	73	51	58
Movimenta dados	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

► **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

Arquiteturas CISC e RISC

► **RISC: Reduced Instruction Set Computer**

1. O efeito das instruções complexas

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

Arquiteturas CISC e RISC

► RISC: Reduced Instruction Set Computer

2 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 X 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

Arquiteturas CISC e RISC

► RISC: Reduced Instruction Set Computer

3. 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

Arquiteturas CISC e RISC

► **RISC: Reduced Instruction Set Computer**

4. 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**

Arquiteturas CISC e RISC

► 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”.

Arquiteturas CISC e RISC

► RISC: Características (cont.)

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

Arquiteturas CISC e RISC

► **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)

Arquiteturas CISC e RISC

► **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)

Arquiteturas CISC e RISC

► **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: atualmente, John Hennessy é reitor da Stanford University.

Arquiteturas CISC e RISC

► **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

Arquiteturas CISC e RISC

► 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

Arquiteturas CISC e RISC

► Bibliografia

PATTERSON, D., HENNESSY, J., "Computer Organization and Design: The Hardware Software Interface", 3rd edition, Morgan Kaufmann-Elsevier, 2005.