

# ARQUITETURA DE COMPUTADORES

---

Computadores com Conjunto Reduzido de Instruções

## Introdução

---

- Década de 60
  - introdução de famílias de computadores (ex: IBM 360)
  - distinção entre arquitetura e implementação do processador
  - uso de microprogramação
- Microprogramação (*Microprogrammed Control Unit*)
  - memória principal: núcleos, grande tempo de acesso
  - memória de controle: semicondutora, "barata"
  - grandes microprogramas não acrescentavam custo
  - maior desempenho => mover software para microprogramas
- Arquitetura x Organização
  - possível compromisso entre custo e desempenho não é definido unicamente pela arquitetura
  - impacto de uma instrução não é decisivo na implementação

## Processadores CISC (*Complex Instruction Set Computer*)

---

- ❑ Instruções mais complexas
    - facilitar tarefa dos compiladores
    - aumentar desempenho
  - ❑ Digital VAX 11/780
    - 303 instruções
    - 16 modos de endereçamento
  - ❑ Intel 386
    - 111 instruções
    - 8 modos de endereçamento
  - ❑ Motorola 68020
    - 109 instruções
    - 18 modos de endereçamento
  - ❑ instruções com comprimento e formato variáveis
- 

3

## Mudanças no contexto [1]

---

A partir da metade da década de 70 ...

- ❑ Memória principal: semicondutores no lugar de núcleos
    - memória principal não era mais tão lenta em relação à memória de controle
  - ❑ Introdução de memórias cache baratas e rápidas
    - acessos à memória cache tão rápidos quanto à memória de controle
- 

4

## Mudanças no contexto [2]

- Efeitos colaterais de conjuntos complexos de instruções
  - tempo de projeto mais longo, mais erros de projeto
  - dificuldade no desenvolvimento de microcódigos com até 400 Kbytes
- Compiladores utilizando sub-conjuntos da arquitetura
  - difícil utilização de instruções complexas
  - otimização de código tornava possível utilização de instruções mais simples
- Maioria das instruções do conjunto eram pouco utilizadas nas aplicações mais comuns
- Advento de circuitos VLSI e problemas no projeto de processadores em um chip único

5

## Uso de instruções

**Exemplo:** uso médio de instruções do 8086 em 3 aplicações – assembler MASM, compilador Turbo C, Lotus 1-2-3

Transferência de dados		Aritméticas / lógicas	
MOV	29	CMP	7
PUSH / POP	12	SAL / SHR / RCR	5
LEA	3	INC / DEC	5
		ADD	3
		OR / XOR	3
Controle / desvio			
JMP	2		
LOOP	4		
CALL / RET	4		
desvios condicionais	10		

6

## Fundamentos RISC [1]

---

- ❑ Idéia lançada no final da década de 70 por pesquisadores de Berkeley
- ❑ Máquina bem simples que tenha implementação bem rápida
  - poucas instruções
  - instruções bem simples
- ❑ Processador num chip único
  - ganho na lógica de controle aproveitado no bloco operacional
- ❑ Desenvolver conjunto de instruções bem simples
  - melhor aproveitamento de recursos escassos em um processador num chip único
  - perspectiva global de aumento de desempenho
- ❑ Grande número de registradores de propósito geral
  - Uso de tecnologia de compiladores para otimizar o uso de registradores

7

## Fundamentos RISC [2]

---

- ❑ Ênfase na otimização do pipeline de instruções
- ❑ Maior número de instruções nos programas compensado por
  - instruções mais rápidas
  - instruções mais curtas
- ❑ Transferir para o software (compilador) o esforço de otimização do tempo de execução
- ❑ Dar suporte a linguagens de alto nível
  - escolher instruções que otimizem desempenho tendo em vista construções mais comuns em linguagens de alto nível

8

## Ganho na lógica de controle

- ❑ Uso de lógica *hardwired*, e não microprogramada – maior velocidade
- ❑ Microprocessadores convencionais usam até 50% do espaço com bloco de controle
- ❑ Processadores RISC: controle ocupa apenas 10% do espaço
- ❑ Aumento do número de registradores
  - diminuição do gargalo de von Neumann
- ❑ Maior possibilidade de uso de pipelines homogêneos

9

## Exemplo *simplista e tendencioso*

$V1 = V2 + V3 + V4 + V5$   
V1 a V5 são variáveis em memória

Processador CISC

supondo instruções com 3 endereços,  
ocupando 3 palavras

$V1 = V2 + V3$

$V1 = V1 + V4$

$V1 = V1 + V5$

$3 \times 3 = 9$  palavras

$3 \times 6 = 18$  acessos à memória

Processador RISC

instruções ocupam 1 palavra

LOAD R2, V2

LOAD R3, V3

ADD R1, R2, R3

LOAD R4, V4

ADD R1, R1, R4

LOAD R5, V5

ADD R1, R1, R5

STORE V1, R1

$8 \times 1 = 8$  palavras

$8 \times 1 + 5 \times 1 = 13$  acessos à memória

10

## Comparação entre processadores

	<b>CISC</b>			<b>RISC</b>	
	IBM	DEC VAX	Intel	Motorola	MIPS
	370/168	11/780	486	88000	R4000
	1973	1978	1989	1988	1991
No. of instruction					
	208	303	235	51	94
Instruction size (octets)					
	2-6	2-57	1-11	4	4
Addressing modes					
	4	22	11	3	1
GP Registers					
	16	16	8	32	32
Control memory (k bytes) (microprogramming)					
	420	480	246	0	0

11

## Medidas concretas – MIPS [1]

- Fase de execução da instrução num único ciclo de relógio
  - instruções tão rápidas quanto micro-instruções
- Todas as instruções do mesmo tamanho e com o mesmo formato (ou com poucas variações de formato)
  - simplificar implementação do controle
- Dados imediatos pequenos e deslocamentos pequenos
- Acesso à memória principal apenas através de instruções LOAD e STORE
  - demais instruções fazem operações apenas entre registradores
  - simplificar implementação do controle (pipeline)
  - tornar operações aritméticas e lógicas mais rápidas

12

## Medidas concretas – MIPS [2]

---

- ❑ usar modos de endereçamento bem simples
  - simplificar implementação do controle
- ❑ poucos tipos de dados
- ❑ usar arquitetura Harvard
  - memórias de dados e instruções separadas permitem aumentar *bandwidth* de memória
- ❑ uso de instruções *compare-and-branch*
  - estatísticas mostram que até 98% dos branches são precedidos por uma comparação
  - comparação e branch reunidos numa única instrução evitam necessidade de flags (N, Z) e permitem execução num único ciclo

---

13

## Literatura

---

- ❑ *Stallings – cap 12*
- ❑ *Patterson – apêndice D*

---

14

## RISC x CISC

---

- ❑ Não é uma questão fechada
    - Falta um par de máquinas RISC e CISC para comparação
    - Falta um conjunto de *benchmark* definitivo
    - Difícil separar efeitos do hardware e do compilador
    - Análises feitas em máquinas 'experimentais' e não-comerciais
  - ❑ Muitos projetistas usam conceitos das duas filosofias
    - Ex. PowerPC e Pentium II
- 

15

---

Você está vivendo os anos em que um homem se transforma naquilo que ele será para o resto da vida. Por isso, tome cuidado. Muito poder significa grande responsabilidade.

*Ben Parker - Homem Aranha, o filme*

---

16



## Questões adicionais...

---

- ☐ Para um dos processadores abaixo, pesquise:
    - Numero de instruções
    - Tamanho das instruções (em bytes)
    - Modos de endereçamento
    - Numero de registradores de propósito geral
      - ☐ Z-80 (Zilog)
      - ☐ 8086 (Intel)
      - ☐ PowerPC (IBM)
      - ☐ ARM7
      - ☐ PIC 18F (Microchip)
      - ☐ ColdFire (Motorola - Freescale)
      - ☐ MicroBlaze (Xilinx) - DLX
      - ☐ XScale (Intel)
      - ☐ OMAP (Texas Instruments)
      - ☐ SPARC (SUN)
      - ☐ TMS320C6000 (Texas Instruments)
      - ☐ 8051 (Intel)
      - ☐ MC56F800 (Motorola - Freescale)
      - ☐ Pentium II (Intel)
      - ☐ Geode (AMD)
- 

17