

Organização e Arquitetura de Computadores I

Linguagem de Montagem

Sumário

- CISC
- RISC
- MIPS
- Representação de instruções

CISC

- O CISC (*Complex Instruction Set Computer*) resulta em:
 - Programas menores
 - Simplicidade nos projetos de compiladores
 - Grande número de instruções
- CISC é caracterizado pelo grande número de conjunto de instruções complexas, objetivando o uso mínimo de memória.

CISC

- CISC incorpora complexos modos de endereçamento para os operandos e usa um pequeno número de registradores.
- Máquinas CISC foram, na maioria, construídas como **microprogramadas** e só agora começou a implementar micro/macro instruções RISC.
- Exemplos de máquinas CISC:
 - PDP-11, Intel i86, Motorola 68K
- Máquinas CISC com micro/macro-instruções RISC:
 - Pentium, AMD Athlon

RISC

- O RISC (*Reduced Instruction Set Computer*), tem as seguintes características:
 - **Simplicidade**, poucas instruções com poucos formatos
 - Poucos modos de endereçamento
 - Operações entre registradores
 - **Uma instrução por ciclo**
- RISC é caracterizado por usar um conjunto pequeno e simples de instruções.

RISC

- Máquinas RISC geralmente possuem número elevado de registradores.
- As instruções das máquinas RISC são diretamente executadas pelo hardware (não há microcódigo).
- Exemplos de Máquinas RISC:
 - MIPS,
 - PowerPC;
 - I860;
 - I960;
 - SPARC;
 - ALPAH
 - HP-PA

MIPS

- MIPS fornece um bom modelo arquitetônico para estudo, não apenas devido à popularidade desse tipo de processador, mas também porque é uma arquitetura fácil de entender.
- Nos 15 anos desde que surgiu o primeiro processador MIPS, houve muitas versões de MIPS.
- **Na programação em assembler, será adotada a notação do MIPS64 e para testes o programa SPIM.**

MIPS

- O MIPS tem 32 registradores de uso geral (GPRs – *General-Purpose Registers*) de 64 bits, denominados R0, ..., R31.
- Além disso, existe um conjunto de 32 registradores de ponto flutuante (FPRs – *Floating-Point Registers*), denominados F0, ..., F31.
 - Esses registradores podem conter 32 valores de precisão simples (de 32 bits) ou 32 valores de precisão dupla (64 bits)
 - Quando o FPR contém um número de precisão simples, a outra metade do FPR não é usada;

Representação de Instruções

- Os registradores serão manipulados através de instruções. Internamente, cada instrução é representada como uma seqüência de bits.
- Para distinguir do **assembler**, a versão numérica é chamada de **linguagem de máquina**, e a sequência de dessas instruções é o **código de máquina**.
- O MIPS admite quatro classes gerais de instruções:
 - Operações da ULA
 - Carga e armazenamento
 - Desvio e Saltos
 - Operações de ponto flutuante

Representação de Instruções

- Formato dos três tipos de instruções mais comuns:

R	op	rs	rt	rd	shamt	Funct
	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits
I	op	rs	rt	Imm		
	6 bits	5 bits	5 bits	16 bits		
J	op	Target				
	6 bits	26 bits				

Representação de Instruções

● Instruções do Tipo – R (*Register*):

- Simboliza o uso de registradores pela instrução, normalmente, instruções Lógicas e Aritméticas:

● **op** – código que identifica a instrução (*opcode*). Para R é sempre 0 (exceto rfe=0x10);

● **rs, rt** – números dos dois registros que contêm os operandos;

● **rd** – número do registro que virá a conter o resultado;

● **shamt** – quantidade de bits a serem deslocados (*shift amount*);

Número de deslocamentos – apenas para as instruções sll, srl e sra (0 nas outras instruções);

● **funct** – seleciona a variante específica da operação no campo **op**, também chamado **código de função**.

Representação de Instruções

● Instruções do Tipo – R (*Register*):

- Exemplo de uma instrução do tipo R:

add \$t0, \$t2, \$s0

op = 0

funct = 32

rs = \$t2 = 10

rt = \$s0 = 16

rd = \$t0 = 8

shamt = 0

<i>op</i>	<i>rs</i>	<i>rt</i>	<i>rd</i>	<i>shamt</i>	<i>funct</i>
000000	01010	10000	01000	00000	100000

Representação de Instruções

● Instruções do Tipo – I (*immediate*):

- Simboliza instruções imediatas e de transferência de dados, os campos são:
 - **op** – código que identifica a instrução ($\neq 0$);
 - **rs, rt** – números dos dois registros utilizados na instrução;
 - **imm** – constante ou endereço de memória.

<i>6 bits</i>	<i>5 bits</i>	<i>5 bits</i>	<i>16 bits</i>
<i>op</i>	<i>rs</i>	<i>rt</i>	<i>imm</i>

Representação de Instruções

● Instruções do Tipo – I (*immediate*):

- Exemplo:

addi \$t0, \$t2, 256

op = 0x08

rt = \$t0 = 8

rs = \$t2 = 10

imm = 256

<i>op</i>	<i>rs</i>	<i>rt</i>	<i>endereço</i>
001000	01010	01000	0000 0001 0000 0000

Representação de Instruções

● Instruções básicas dos formatos R e I:

add \$s1,\$s2,\$s3

$\$s1 = \$s2 + \$s3$

sub \$s1,\$s2,\$s3

$\$s1 = \$s2 - \$s3$

addi \$s1,\$s2,const

$\$s1 = \$s2 + \text{constante}$

lw \$s1,ender(\$s2)

$\$s1 = \text{memória}[\$s2 + \text{ender}]$

sw \$s1, ender(\$s2)

$\text{memória}[\$s2 + \text{ender}] = \$s1$

Representação de Instruções

● Instruções do Tipo – I (*immediate*):

- Exemplo:

lw \$t0, 0(\$t1)

op = 0x23

rs = \$t1 = 9

rt = \$t0 = 8

imm = 0

<i>op</i>	<i>rs</i>	<i>rt</i>	<i>endereço</i>
100011	01001	01000	0000 0000 0000 0000

Representação de Instruções

● Instruções do Tipo – I (*immediate*):

■ Exemplo:

- Supondo que o endereço-base de um array A[300] esteja armazenado em \$t1. Qual o código de montagem do MIPS para carregar no registrador temporário \$t0 o conteúdo da última posição do array A?

3	1000 1111 ... 0010
2	1000 1001 ... 0100
1	1010 0101 ... 1000
0	1000 0000 ... 0000
Endereço	Dados (32 bits)

Representação de Instruções

● Instruções do Tipo – I (*immediate*):

Supondo que \$t1 guarde o endereço base do array A[300], o deslocamento necessário é $300 \times 4 = 1200$

lw \$t0, 1200(\$t1)

op = 0x23

rt = \$t0 = 8

rs = \$t1 = 9

imm = 1200

<i>op</i>	<i>rs</i>	<i>rt</i>	<i>endereço</i>
100011	01000	01000	0000 0100 1011 0000

Representação de Instruções

● Instruções do Tipo – I (*immediate*):

Armazena o conteúdo do registrador temporário \$t0 no endereço de memória 1200 + endereço-base

sw \$t0, 1200(\$t1)

op = 0x29

rt = \$t0 = 8

rs = \$t1 = 9

imm = 1200

<i>op</i>	<i>rs</i>	<i>rt</i>	<i>endereço</i>
101011	01001	01000	0000 0100 1011 0000

Representação de Instruções

● Instruções do Tipo – I (*immediate*):

- Suponha que A é um array de 100 elementos cujo endereço-base está no registrador \$s3 e que o compilador associa as variáveis g, h e i aos registradores \$s1, \$s2 e \$s4. Qual o código gerado para o MIPS, correspondente ao comando C abaixo?

$g = h + A[i];$

Representação de Instruções

● Instruções do Tipo – I (*immediate*):

add \$t1, \$s4, \$s4	# \$t1 recebe $2 * i$
add \$t1, \$t1, \$t1	# \$t1 recebe $4 * i$
add \$t1, \$t1, \$s3	# \$t1 recebe o endereço de $A[i]$
lw \$t0, 0(\$t1)	# \$t0 recebe $A[i]$
add \$s1, \$s2, \$t0	# g recebe $h + A[i]$