

Org(omp) 15/8 (3)  
Exemplo: (página 4 é antes da 3)

A é um vetor de 100 palavras

$$g = h + A[8];$$

\$S1 \$2 \$3 - base de A

lw \$t0, 32(\$S3)

add \$S1, \$S2, \$t0 offset

alinhamento

de forma inversa ao load word (lw)  
temos o store word (sw)

Exemplo:  $A[12] = h + A[8];$

\$S2 base de A - \$S3

lw \$t0, 32(\$S3)

add \$t0, \$S2, \$t0

sw \$t0, 48(\$S3)

Ⓐ ← melhor

Ⓑ

1 lw \$t0, 0(\$S3)

1

2 lw \$t1, 4(\$S3)

2

3 lw \$t2, 8(\$S3)

5

4 lw \$t3, 12(\$S3)

3

5 add \$t4, \$t0, \$t1

6

6 add \$t4, \$t2, \$t4

4

7 add \$t4, \$t3, \$t4

7

(antes da 3)

14

## Operações em MIPS

add  $t_5, g, h$

32 registradores

ao invés de numerar os registradores

tipos  $\$r_0, \$r_1, \dots$  para temporários  
 $\$t_0, \dots$  para temporários variáveis

exemplo

$\$s_0$      $\$s_1$   $\$s_2$      $\$s_3$   $\$s_4$

$f = (g + h) - (i + j)$

add  $\$t_0, \$r_1, \$s_2$

add  $\$t_1, \$r_3, \$s_4$

sub  $\$s_0, \$t_0, \$t_1$

Como carregar e guardar os registradores com dados na memória

· Instruções de transferência de dados  
memória = vetor unidimensional

(ir para a 3)

Geralmente os programadores têm  $15/8$  15 mais variáveis do que existem registradores.

As variáveis menos usadas "voltam" para a memória.

Para um bom desempenho os compiladores devem usar os registradores de forma eficiente.

Constantes são bastante usadas:  
opção 1) guardar as CTES na memória  
2) criar outra instrução

addi \$S3, \$S3, 4

Princípio: faça com que o caso comum seja rápido.

## REPRESENTAÇÃO BINÁRIA

registros \$S<sub>0</sub> a \$S<sub>7</sub> nos registradores 16-23  
\$t<sub>0</sub> a \$t<sub>7</sub> " " 8-15

add \$t<sub>0</sub>, \$S<sub>1</sub>, \$S<sub>2</sub>

0 17(\$1) 18(\$2) 8(\$t<sub>0</sub>) 0 32

000000 10001 0010 01000 00000 100000

6bits 5bits 5bits 5bits 5bits 6bits

# Campos MIPS

6

op	rs	rt	rd	shamt	funct
6	5	5	5	5	6

op: basic operation of the instruction  
(opcode)

rs: first register source operand

rt: second " " "

rd: register destination operand

shamt: shift amount

funct: function-specific variation of the operand

Problema: e as operações com memória?  
5 bits é muito pouco para o offset

Princípio 4: um bom projeto pede bons compromissos

acima R-type (registradores)

I-type (transparência de dados)

op	rs	rt	constante ou endereço
----	----	----	-----------------------

6	5	5	16
---	---	---	----

exemplo: lw \$t0, 32(\$s3)

35 19 8 32

\$s3 \$t0

17

Os formatos iniciais dos dois tipos não são iguais. O hardware verifica o primeiro campo para saber como decodificar a instrução.

Exemplo:  $A[300] = h + A[300];$

\$s2

\$t1 base de A

lw \$t0, 1200(\$t1)

add \$t0, \$s2, \$t0

sw \$t0, 1200(\$t1)

em linguagem de máquinas:

op	rr	rt	rd	shamt	funct
----	----	----	----	-------	-------

35	9	8		1200	
----	---	---	--	------	--

0	18	8	8	0	32
---	----	---	---	---	----

43	9	8	<del>8</del>	1200	
----	---	---	--------------	------	--