

MIPS – Lab. Arq. Computadores

Novas instruções bne, beq, j, slt, slti, mult, div mfhi e mflo.

Consulte as transparências para as instruções:

Jump (j)

Branch if equal (beq)

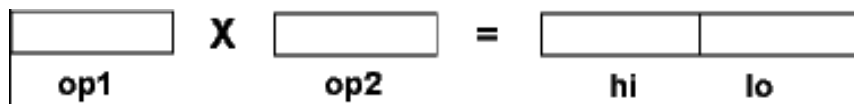
Branch if not equal (bne)

Set on less then (slt)

Set on less then com imediato (slti)

Unidade de Multiplicação no MIPS

A unidade multiplicadora do Mips possui dois registradores de 32-bit denominados **hi** e **lo**. Não são registradores de propósito geral. Quando dois registradores com operandos de 32-bit são multiplicados, **hi** e **lo** possuem o resultado de 64 bits.



Bits 32 até 63 estão em **hi** e bits 0 até 31 em **lo**.

mult e multu

mult s,t # hilo <-- \$s * \$t (operandos em com. de dois)

multu s,t # hilo <-- \$s * \$t (operandos unsigned)

A multiplicação inteira nunca causa um trap.

Perg.: Dois inteiros pequenos são multiplicados, onde está o resultado?

Resp: Se o resultado for inferior a 32 bits, em lo, hi possuirá zeros.

Bits Significantes

Os bits significantes em um num. Positivo ou unsigned são todos os bits à direita do bit com valor 1 mais à esquerda do número:

0000 0000 0100 0011 0101 0110 1101 1110

O número possui 23 bits significantes.

Os bits significantes em um num. Negativo, são todos os bits à direita do bit com valor 0 mais à esquerda do número:

1111 1111 1011 1100 1010 1001 0010 0010

O número possui 23 bits significantes.

Para garantir que o produto não possuirá mais do que 32 bits, a soma dos bits

significantes dos números deverá ser menor ou igual a 32.

Perg.: Aproximadamente quantos bits significantes você espera do produto:

01001010×00010101

Resp: Aproximadamente 12 bits.

mfhi e mflo

São as instruções utilizadas para mover os resultados da multiplicação para registradores de uso geral.

mfhi d # d <-- hi. Mover de Hi

mflo d # d <-- lo. Mover de Lo

Perg: Deve-se mover os resultados de lo e hi antes de uma multiplicação seguinte?

Resp: Sim.

Perg.: Como computar $5 \times X - 74$?

Resp.:

```
## novoMult.asm
##
## Programa para calcular  $5 \times x - 74$ 
##
## $8 x
## $9 resultado
.text
.globl main
main:
ori $8, $0, 12            # x em $8
ori $9, $0, 5            # 5 em $9
mult $9, $8              # lo <-- 5x
mflo $9                  # $9 = 5x
addi $9, $9, -74         # $9 = 5x - 74
## End of file
```

Perg.: O que significa o "u" nas seguintes instruções:

addu _____

multu _____

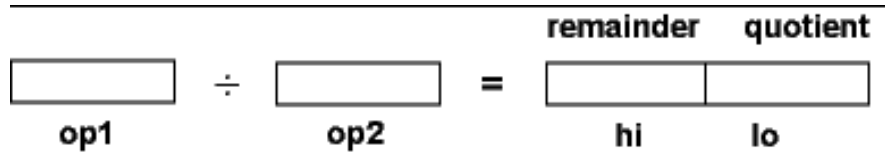
Resp:

addu, não acontece um trap no overflow.

multu, os operandos são unsigned.

div e divu

De maneira análoga, temos em hi e lo o resultado e o quociente, respectivamente.



```
div s,t      # lo <-- s div t
              # hi <-- s mod t
              # operandos em comp. de dois
divu s,t     # lo <-- s div t
              # hi <-- s mod t
              # operandos em comp. De dois
```

Perg.: Calcular $(y + x) / (y - x)$

Resp.:

```
## exemplodiv.asm
##
## Programa para calcular  $(y + x) / (y - x)$ 
##
## $8 x
## $9 y
## $10 x/y
## $11 x%y
.text
.globl main
main:
ori $8, $0, 8           # x em $8
ori $9, $0, 36          # y em $9
add $10, $9, $8          # $10 <-- (y+x)
sub $11, $9, $8          # $11 <-- (y-x)
div $10, $11             # hilo <-- (y+x)/(y-x)
mflo $10                 # $10 <-- quociente
mfhi $11                 # $11 <-- resultado
## End of file
```

Perg.: $(36+8) / (36-8) =$

Resp: $(36+8) / (36-8) = 1 \text{ R } 16$, or $0x1 \text{ R } 0x10$

Exercícios

Responda

1. Se tivermos 2 inteiros, cada um com 32 bits, quantos bits podemos esperar para o produto?

- A. 16
- B. 32
- C. 64
- D. 128

2. Quais os registradores que armazenam os resultados na multiplicação?

- A. high e low
- B. hi e lo
- C. R0 e R1
- D. \$0 e \$1

3. Qual a operação usada para multiplicar inteiros em comp. de dois?

- A. mult
- B. multu
- C. multi
- D. mutt

4. Qual instrução move os bits menos significativos da multiplicação para o reg. 8?

- A. move \$8,lo
- B. mvlo \$8,lo
- C. mflo \$8
- D. addu \$8,\$0,lo

5. Se tivermos dois inteiros, cada um com 32 bits, quantos bits deveremos estar preparados para receber no **quociente**?

- A. 16
- B. 32
- C. 64
- D. 128

6. Após a instrução div, qual registrador possui o quociente?

- A. lo
- B. hi
- C. high
- D. \$2

7. Qual a inst. Usada para dividir dois inteiros em comp. de dois?

- A. dv
- B. divide
- C. divu
- D. div

8. Faça um arithmetic shift right de dois no seguinte padrão de bits: 1001 1011

- A. 1110 0110
- B. 0010 0110
- C. 1100 1101
- D. 0011 0111

9. Qual o efeito de um **arithmetic shift right** de uma posição?

- A. Se o inteiro for unsigned, o shift o divide por 2. Se o inteiro for signed, o shift o divide por 2.
- B. Se o inteiro for unsigned, o shift o divide por 2. Se o inteiro for signed, o shift pode resultar em um valor errado.
- C. Se o inteiro for unsigned, o shift pode ocasionar um valor errado. Se o inteiro for signed, o shift o divide por 2.
- D. O shift multiplica o número por dois.

10. Qual sequencia de instruções avalia $3x+7$, onde x é iniciado no reg. \$8 e o resultado armazenado em \$9?

A.

ori \$3,\$0,3
mult \$8,\$3
mflo \$9
addi \$9,\$9,7

B.

ori \$3,\$0,3
mult \$8,\$3
addi \$9,\$8,7

C.

ori \$3,\$0,3
mult \$8,\$3
mfhi \$9
addi \$9,\$9,7

D.

mult \$8,3
mflo \$9
addi \$9,\$9,7

PROGRAMAS

// programa 13:

Escreva um programa que leia um valor A da memória, identifique se o número é negativo ou não e encontre o seu módulo. O valor deverá ser reescrito sobre A.

// programa 14:

Escreva um programa que leia da memória um valor de Temperatura TEMP. Se $TEMP \geq 30$ e $TEMP \leq 50$ uma variável FLAG, também na memória, deverá receber o valor 1, caso contrário, FLAG deverá ser zero.

// programa 15:

Escrever um programa que crie um vetor de 100 elementos na memória onde $vetor[i] = 2*i + 1$. Após a última posição do vetor criado, escrever a soma de todos os valores armazenados do vetor.

// programa 16:

Considere que a partir da primeira posição livre da memória temos um vetor com 100 elementos. Escrever um programa que ordene esse vetor de acordo com o algoritmo da bolha. Faça o teste colocando um vetor totalmente desordenado e verifique se o algoritmo funciona.

// programa 17

$$1) \ y = \begin{cases} x^4 + x^3 - 2x^2 & \text{se } x \text{ for par} \\ x^5 - x^3 + 1 & \text{se } x \text{ for impar} \end{cases}$$

Os valores de x devem ser lidos da primeira posição livre da memória e o valor de y deverá ser escrito na segunda posição livre.

// programa 18

$$2) \ y = \begin{cases} x^3 + 1 & \text{se } x > 0 \\ x^4 - 1 & \text{se } x \leq 0 \end{cases}$$

Os valores de x devem ser lidos da primeira posição livre da memória e o valor de y deverá ser escrito na segunda posição livre.

// programa 19

Escreva um programa que avalie a expressão: $(x*y)/z$.

Use $x = 1600000$ ($=0x186A00$), $y = 80000$ ($=0x13880$), e $z = 400000$ ($=0x61A80$).
Inicializar os registradores com os valores acima.

// programa 20

Escreva um programa que gere um vetor com os números ímpares até 100. O valor 1 deverá estar na primeira posição livre da memória.

Após gerar e armazenar o vetor, seu programa deverá varrer todo o vetor, ler cada termo, somar em uma variável auxiliar e armazenar na última posição a soma de todos os elementos.

Mostre a tabela de porcentagens das instruções utilizadas.

// programa 21

Escreva um programa que gere um vetor de inteiros até 100. Seu programa deverá armazenar na memória os números pares separados dos ímpares. Armazene primeiro os pares e logo a seguir os ímpares.

Mostre a tabela de porcentagens das instruções utilizadas.

// programa 22

Para a expressão a seguir, escreva um programa que calcule o valor de k :

$$k = x * y$$

O valor de x deve ser lido da primeira posição livre da memória e o valor de y deverá lido da segunda posição livre. O valor de k , após calculado, deverá ainda ser escrito na terceira posição livre da memória.

// programa 23

O mesmo programa 22, porém: $k = x^y$