

# Arquitetura de Computadores I

Flávio Márcio

Assembly MIPS

If's e Loop's

# Instruções de Desvio

- beq registrador1, registrador2, L1
  - se o valor do registrador1 for igual ao do registrador2 o programa será desviado para o labelL1. ( beq = branch if equal).
- bne registrador1, registrador2, L1
  - se o valor do registrador1 não for igual ao do registrador2 o programa será desviado para o labelL1. ( beq = branch if not equal).
- j L1
  - Salto incondicional para o label L1

# Exemplo IF

- Seja o comando abaixo:
- if (  $i == j$  ) go to L1;
- $f = g + h$ ;
- L1:  $f = f - i$ ;
- Supondo que as 5 variáveis correspondam aos registradores  $\$s0..\$s4$ , respectivamente, como fica o código MIPS para o comando?
- Solução
- beq  $\$s3, \$s4, L1$  # vá para L1 se  $i = j$
- add  $\$s0, \$s1, \$s2$  #  $f = g + h$ , executado se  $i \neq j$
- L1: sub  $\$s0, \$s0, \$s3$  #  $f = f - i$ , executado se  $i = j$

# Exemplo If - else

- Seja o comando abaixo:
- `if ( i == j) f = g + h; else f = g – h;`
- Solução
- `bne $s3,$s4,Else # vá para Else se i != j`
- `add $s0,$s1,$s2 # f = g + h, se i == j`
- `j Exit # vá para Exit`
- `Else: sub $s0,$s1,$s2 # f = g – h, se i != j`
- `Exit:`

# Exemplo loop

- Loop:  $g = g + A[i];$
- $i = i + j;$
- if (  $i \neq h$  ) go to Loop
  
- Solução
- Loop: add \$t1,\$s3,\$s3 #  $t1 = 2 * i$
- add \$t1,\$t1,\$t1 #  $t1 = 4 * i$
- add \$t1,\$t1,\$s5 # \$t1 recebe endereço de A[i]
- lw \$t0,0(\$t1) # \$t0 recebe A[i]
- add \$s1,\$s1,\$t0 #  $g = g + A[i]$
- add \$s3,\$s3,\$s4 #  $i = i + j$
- bne \$s3,\$s2,Loop # se  $i \neq h$  vá para Loo

# Exemplo While

- while (save[i] == k)
- i = i + j;
- Solução: para i, j e k correspondendo a \$s3, \$s4 e \$s5, respectivamente, e o endereço base do array em \$s6, temos:
- Loop: add \$t1,\$s3,\$s3 # \$t1 = 2 \* i
- add \$t1,\$t1,\$t1 # \$t1 = 4 \* i
- add \$t1,\$t1,\$s6 # \$t1 = endereço de save[i]
- lw \$t0,0(\$t1) # \$t0 recebe save[i]
- bne \$t0,\$s5,Exit # va para Exit se save[i] != k
- add \$s3,\$s3,\$s4 # i = i + j
- j Loop
- Exit:

# Instruções maior ou menor

- `slt reg_temp, reg1, reg2`
  - se `reg1` é menor que `reg2`, `reg_temp` é setado para 1, caso contrário é resetado para 0.
  - Nos processadores MIPS o registrador `$0` possui o valor zero (`$zero`).
- Exemplo: Compilando o teste less than.
- `slt $t0,$s0,$s1` # `$t0` é setado se `$s0 < $s1`
- `bne $t0,$zero,Less`
- # vá para Less, se `$t0 != 0` , ou seja `a < b`