

# LABORATÓRIO 2

## ASSEMBLY MIPS

Prof<sup>a</sup> Monica Magalhães Pereira

# Plano de aula

- Entender como construir laços e saltos condicionais e incondicionais em Assembly-MIPS

# Saltos Incondicionais

- Faça o download do arquivo lab4.asm no SIGAA
- Abra o simulador MARS e os arquivos lab3.asm (aula anterior) e lab4.asm (**File...Open**)
- Você poderá visualizar e editar o programa em assembly na aba **Edit**
- Você poderá visualizar a simulação do programa, acompanhando o comportamento das memórias, do banco de registradores e a sequência de execução, na aba **Execute**
- Para iniciar a simulação, clique no botão



# Saltos Incondicionais

1. Execute o código lab3.asm e descreva o que ele faz.
2. Agora, execute o código lab4.asm e descreva o que ele faz.
3. Qual a diferença entre o lab3 e o lab4?

# SALTOS CONDICIONAIS

IGUAL E DIFERENTE

# Saltos Condicionais

- Faça o download do arquivo lab5.asm no SIGAA
- Abra o simulador MARS e o arquivo lab5.asm (**File...Open**)
- Você poderá visualizar e editar o programa em assembly na aba **Edit**
- Você poderá visualizar a simulação do programa, acompanhando o comportamento das memórias, do banco de registradores e a sequência de execução, na aba **Execute**
- Para iniciar a simulação, clique no botão



# Saltos Condicionais

4. Execute o lab5 e descreva o que ele faz.
5. Existe algum erro nesse código? Se sim, qual?
6. Quais alterações são necessárias para corrigir esse erro?

# SALTOS CONDICIONAIS

MAIOR OU MENOR



# Saltos Condicionais

- *Set on less than*

*slt \$t0, \$s3, \$s4*

*\$t0 = "1" se "\$s3 < \$s4"*

*\$t0 = "0", caso contrário*

```
slt $s1, $s2, $s3
bne $s1, $zero, Else
addi $s2, $s2, 1
j EXIT
Else:
    addi $s2, $s2, 2
EXIT: ...
```

7. Baseado no trecho de código assembly deste slide, implemente o lab6.asm que lê dois números e armazena em \$t0 e \$t1 e compara.

if (\$t0 < \$t1)

    \$t0 += 4;

else

    \$t0+=5;

LAÇOS

# Laços

7. Usando as instruções trabalhadas até o momento, implemente um código Assembly-MIPS equivalente ao seguinte código C:

Nomei o arquivo como lab7.asm:

```
int i = 0;  
while ( i != 10 ) {  
    i=i++;  
}
```

# Bibliografia

- PATTERSON, D. A. & HENNESSY, J. L.

**Organização e Projeto de Computadores** – A Interface  
Hardware/Software. 3ª ed. Campus,

## CAPÍTULO 2

- MIPS Assembly Language

<http://www.inf.uni-konstanz.de/dbis/teaching/ws0304/computing-systems/download/rs-05.pdf>

Introdução Curta ao MIPS

<http://www.di.ubi.pt/~desousa/2011-2012/LFC/mips.pdf>