

# Conjunto de Instruções

## Parte II

# Histórico de revisões

2

Revisão	Data	Responsável	Descrição
0.1	-x-	Prof. Cesar Zeferino	Primeira versão
0.2	08/2017	Prof. Cesar Zeferino	Revisão do modelo

**Observação:** Este material foi produzido por pesquisadores do Laboratório de Sistemas Embarcados e Distribuídos (LEDS – Laboratory of Embedded and Distributed Systems) da Universidade do Vale do Itajaí e é destinado para uso em aulas ministradas por seus pesquisadores.

# Introdução

3

## ❑ Objetivo

- ❑ Conhecer as instruções de lógica do processador MIPS e suas aplicações

## ❑ Conteúdo

- ❑ Operações e instruções de lógica bit a bit no MIPS

# Introdução

4

## ❑ Bibliografia

- ❑ PATTERSON, David A.; HENNESSY, John L. Abstrações e tecnologias computacionais. *In*: \_\_\_\_\_. **Organização e projeto de computadores: a interface hardware/software**. 4. ed. Rio de Janeiro: Campus, 2014. cap. 2. Disponível em:  
<<http://www.sciencedirect.com/science/book/9788535235852>>. Acesso em: 13 mar. 2017.
  
- ❑ Edições anteriores
  - ❑ Patterson e Hennessy (2005, cap. 2)
  - ❑ Patterson e Hennessy (2000, cap. 4)

# 1 Operações de deslocamento

## ❑ Shift left logical – **sll**

- ❑ Desloca o número para a esquerda

**sll** rd, rt, shamt

## ❑ Shift right logical – **srl**

- ❑ Desloca o número para a direita

**srl** rd, rt, shamt

- ❑ O campo *shamt* (*shift amount*) indica a quantidade de deslocamentos a ser aplicada ao número

## ❑ Exemplo

**sll** \$t0, \$s0, 8 # \$t0=\$s0<<8 (desloca p/ esquerda)

opcode	rs	rt	rd	shamt	funct
000000	00000	10000	01000	01000	000000
Formato R	Não usado	\$s0	\$t0	8	sll

# 1 Operações de deslocamento

- ❑ Uma operação de deslocamento de um número para esquerda em  $n$  posições tem o efeito correspondente a multiplicar o número por  $2^n$ 
  - ❑  $4 \ll 2 = 4 \times 2^2 = 4 \times 4 = 16$
- ❑ Por outro lado uma operação de deslocamento de um número para direita em  $n$  posições tem o efeito correspondente a dividir o número por  $2^n$ 
  - ❑  $8 \gg 1 = 8 \div 2^1 = 8 \div 2 = 4$

## 2 Operação *and*

### ❑ Usada para “desligar” bits de um dado (torná-los iguais a 0)

- ❑ Aplicando-se a operação lógica *and* com uma máscara 0 sobre qualquer valor de bit, produz um bit igual a 0. Já uma máscara 1 produz um bit igual ao bit processado.

### ❑ Instruções

**and** rd, rs, rt

**andi** rt, rs, imediato

### ❑ Exemplo

	01100110111001101110001101111001	<sub>dois</sub>	
<b>and</b>	<u>11111111111111111111111111111111</u>	<b>00</b> <sub>dois</sub>	(máscara de bits)
	011001101110011011100011011110	<b>00</b> <sub>dois</sub>	





# 4 Operação *xor*

- ❑ Usada para inverter os bits de um número (complementá-los o bit)
  - ❑ Aplicando-se a operação lógica *xor* com uma máscara 1 sobre qualquer valor de bit, produz um bit igual que é o complemento do bit processado 0. Já uma máscara 0 produz um bit igual ao bit processado.
  - ❑ Instruções
 

```
xor  rd, rs, rt
xori rt, rs, imediato
```
  - ❑ Exemplo

```

01100110111001101110001101111001dois
xor 0000000000000000000000000000000011dois (máscara de bits)
01100110111001101110001101111010dois

```

# 5 Outras instruções

10

## ❑ **nor**

```
nor rd, rs, rt
```

## ❑ **not**

```
not rdest, rsrc
```

❑ not é uma pseudo-instrução implementada usando um nor

❑ Exemplo

```
not $s1, $s2
```

É implementada como

```
nor $s1, $s2, $zero
```

## 6 Registrador \$zero

11

- ❑ Registrador somente de leitura com endereço 0 cujo conteúdo é sempre igual a

00

- ❑ Largamente utilizado em instruções de desvio condicional e de comparação

# Resumo

12

## ❑ Arquitetura estudada do MIPS

- ❑ Registradores  $\$s7-0$ ,  $\$t7-0$  e  **$\$zero$**
- ❑ Memória  $2^{30}$  palavras de 32 bits
- ❑ Formatos de instrução R e I
- ❑ Modos de endereçamento imediato, via registrador e via registrador-base
- ❑ Instruções `add`, `sub`, `addi`, `lw`, `sw`, **`sll`**, **`srl`**, **`and`**, **`andi`**, **`or`**, **`ori`**, **`xor`**, **`xori`**, **`nor`** e **`not`**

# Resumo: Conjunto de Instruções

13

## □ Aritmética

Instrução	Formato	Modo de endereçamento
add	R	Via registrador
sub	R	Via registrador
addi	I	Imediato

## □ Transferência

Instrução	Formato	Modo de endereçamento
lw	I	Via registrador-base
sw	I	Via registrador-base

# Resumo: Conjunto de Instruções

14

## ❑ Lógica

Instrução	Formato	Modo de endereçamento
sll	R	Via registrador
srl	R	Via registrador
and	R	Via registrador
andi	I	Imediato
or	R	Via registrador
ori	I	Imediato
xor	R	Via registrador
xori	I	Imediato
nor	R	Via registrador
not	Pseudo-instrução	