

# Digital Design and Computer Architecture



David Money Harris & Sarah L. Harris



**Universidade Federal de Campina Grande**  
**Unidade Acadêmica de Sistemas e Computação**  
**Curso de Bacharelado em Ciência da Computação**

## **Organização e Arquitetura de Computadores I**

---

---

### **Organização e Arquitetura Básicas de Computadores (Parte III - Exemplos)**

**Prof<sup>a</sup> Joseana Macêdo Fachine Régis de Araújo**  
**[joseana@computacao.ufcg.edu.br](mailto:joseana@computacao.ufcg.edu.br)**

Carga Horária: 60 horas

# Tópicos

- **Arquitetura do Conjunto de Instruções**
  - Exemplos (Simulador RISC-V)



David Money Harris & Sarah L. Harris





# Nível ISA (RISC-V)

## Como criar o arquivo em Assembly

- ❑ Deve ser criado um arquivo contendo as instruções em Assembly, para serem geradas as instruções em binário.
- ❑ Deve ser inserida uma instrução por linha.
- ❑ O arquivo deve ter a extensão `.s`. Ex.: `inst.s`
- ❑ Um par de `/*` e `*/` é usado para iniciar um comentário na linha, como em C ou Java.

Fonte: <http://labarc.ufcg.edu.br/oac/index.php?n=OAC.Masm>



# Nível ISA (RISC-V)

## Diretivas de Assembly

```
.section .text /* indica que agora vem instruções RISC-V */  
.global _start /* declara o label "_start" como label global */  
.skip <n> /* deixa um espaço de n bytes */  
/* uma instrução RISC-V ocupa 4 bytes */
```

## Exemplo:

```
.section .text  
.global _start  
_start: addi t3, zero, 0x67 /* uma instrução qualquer */  
sw t3, 32(zero) /* outra instrução qualquer */  
/* pode ter mais outras tantas instruções quaisquer */  
fim:  
.skip 0x80*4+(_start-fim) /* cria espaço ate o endereço 0x80 */
```

# Nível ISA (RISC-V)

## Como Simular num PC

1. **nano inst.s** (entrar no editor para manipular o arquivo inst.s)
2. **riscv64-unknown-elf-gcc -nostdlib -nostartfiles -Tlink.ld -m32 -o inst inst.s** (executar)
3. **spike --isa=RV32I -d inst** (entrar no assembler)
4. No prompt ":"
  - a) pressionar **Enter** para executar uma instrução por vez.
  - b) **reg 0** no prompt visualiza os registradores.
  - c) **q** para sair

Fonte: <http://labarc.ufcg.edu.br/oac/index.php?n=OAC.Masm>

# Nível ISA (RISC-V)

Exemplo: inst1.s

```
.section .text  
.global _start  
  
_start:  
addi s0, zero, 2  
addi s1, zero, 4  
addi s4, zero, 0  
addi s5, zero, 1
```

```
:  
core 0: 0x00000000000000200 (0x00200413) li s0, 2  
:  
core 0: 0x00000000000000204 (0x00400493) li s1, 4  
:  
core 0: 0x00000000000000208 (0x00000a13) li s4, 0  
:  
core 0: 0x0000000000000020c (0x00100a93) li s5, 1
```



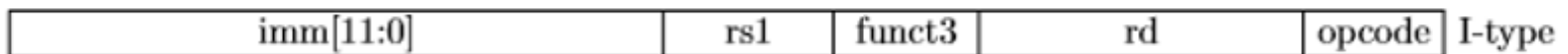
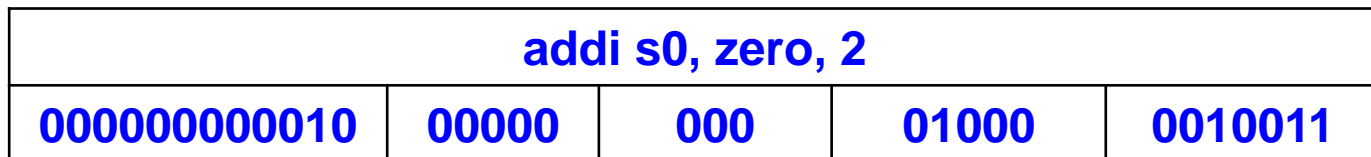
# Nível ISA (RISC-V)

## Exemplo: inst1.s

```

:
core 0: 0x00000000000000200 (0x00200413) li      s0, 2
:
core 0: 0x00000000000000204 (0x00400493) li      s1, 4
:
core 0: 0x00000000000000208 (0x00000a13) li      s4, 0
:
core 0: 0x0000000000000020c (0x00100a93) li      s5, 1

```



# Nível ISA (RISC-V)

## Exemplo: inst2.s

```
.section .text
.global _start

_start:

addi s1, zero, 0x4
sll s1, s1, 2
jal s0, 0x020c
back:
addi s0, zero, 0x0
addi t0, zero, 0x2
for:
    beq s0, t0, done
    add s1, s1, s0
    addi s0, s0, 0x1
j for
done:
add s1, s0, s0
```



# Nível ISA (RISC-V)

## Exemplo: inst2.s

```
core 0: 0x00000000000000200 (0x00400493) li      s1, 4
:
core 0: 0x00000000000000204 (0x00249493) slli   s1, s1, 2
:
core 0: 0x00000000000000208 (0x0040046f) jal    s0, pc + 0x4
:
core 0: 0x0000000000000020c (0x00000413) li      s0, 0
:
core 0: 0x00000000000000210 (0x00200293) li      t0, 2
:
core 0: 0x00000000000000214 (0x00540863) beq    s0, t0, pc + 16
:
core 0: 0x00000000000000218 (0x008484b3) add    s1, s1, s0
:
core 0: 0x0000000000000021c (0x00140413) addi   s0, s0, 1
:
core 0: 0x00000000000000220 (0xff5ff06f) j      pc - 0xc
```

# Nível ISA (RISC-V)

## Exemplo: inst2.s

```
core 0: 0x0000000000000214 (0x00540863) beq    s0, t0, pc + 16
:
core 0: 0x0000000000000218 (0x008484b3) add    s1, s1, s0
:
core 0: 0x000000000000021c (0x00140413) addi   s0, s0, 1
:
core 0: 0x0000000000000220 (0xff5ff06f) j      pc - 0xc
:
core 0: 0x0000000000000214 (0x00540863) beq    s0, t0, pc + 16
:
core 0: 0x0000000000000224 (0x008404b3) add    s1, s0, s0
```