

**REGINALDO GREGÓRIO DE SOUZA NETO**  
**2252813**

**2 - Instruções: A linguagem de Máquina**

**2.21 Exercícios<sup>1</sup>**

<sup>1</sup> Contribuição de John Oliver, da Cal Poly, San Luis Obispo, com colaborações de Nicole Kaiyan (Universidade de Adelaide) e Milos Prvulovic (Georgia Tech)

O Apêndice B descreve o simulador do MIPS, que é útil para estes exercícios. Embora o simulador aceite pseudoinstruções, tente não usá-las em qualquer exercício que pedir para produzir código do MIPS. Seu objetivo deverá ser aprender o conjunto de instruções MIPS real, e se você tiver de contar instruções, sua contagem deverá refletir as instruções reais executadas, e não as pseudoinstruções.

Existem alguns casos em que as pseudoinstruções precisam ser usadas (por exemplo, a instrução `la` quando um valor real não é conhecido durante a codificação em assembly).

Em muitos casos, elas são muito convenientes e resultam em código mais legível (por exemplo, as instruções `li` e `move`). Se você decidir usar pseudoinstruções por esses motivos, por favor, acrescente uma sentença ou duas à sua solução, indicando quais pseudoinstruções usou e por quê.

**Exercício 2.10**

Nos problemas a seguir, a tabela de dados contém bits que representam o opcode de uma instrução. Você deverá traduzir as entradas para o código assembly e determinar que formato da instrução MIPS os bits representam.

a. 0000 0010 0001 0000 1000 0000 0010 0000<sub>dois</sub>

b. 0000 0001 0100 1011 0100 1000 0010 0010<sub>dois</sub>

Utilizar MIPS Reference Data Card.pdf																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
R Op (6 bits)						Rs (5 bits)					Rt (5 bits)					Rd (5 bits)					Shamt (5 bits)					Funct (6 bits)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
I Op						Rs					Rt					Endereço (16 bits)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
J Op						Endereço (26 bits)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
a. 0000 0010 0001 0000 1000 0000 0010 0000 <sub>dois</sub>																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
add \$s0, \$s0, \$s0 (R[rd] = R[rs] + R[rt])													OPCODE = 0X0					FUNCTION = 0X20																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
0X0						16					16					16					0					0X20																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2.10.1 [5] <2.5> Para essas entradas binárias, que instrução elas representam?

**add \$s0, \$s0, \$s0**  
**sub \$t1, \$t2, \$t3**

2.10.2 [5] <2.5> Que tipo de instrução (tipo I, tipo R) as mesmas entradas binárias representam?

**Ambas são do tipo R.**

2.10.3 [5] <2.4, 2.5> Se as entradas binárias anteriores fossem bits de dados, que número elas representariam em hexadecimal?

a. 0000 0010 0001 0000 1000 0000 0010 0000<sub>dois</sub>  
**0 2 1 0 8 0 2 0 hex**  
b. 0000 0001 0100 1011 0100 1000 0010 0010<sub>dois</sub>  
**0 1 4 b 5 8 2 2 hex**

Nos problemas a seguir, a tabela de dados contém instruções MIPS. Você deverá traduzir as entradas para os bits do opcode e determinar qual é o formato da instrução MIPS.

a. addi \$t0, \$t0, 0  
b. sw \$t1, 32(\$t2)

R Op (6 bits)	Rs (5 bits)	Rt (5 bits)	Rd (5 bits)	Shamt (5 bits)	Funct (6 bits)
I Op	Rs	Rt	Endereço (16 bits)		
J Op	Endereço (26 bits)				
I Op	Rs	Rt	Endereço (16 bits)		
a. addi \$t0, \$t0, 0 (R[rt] = R[rs] + SignExtImm)				OPCODE = 0X8	
0X8		8	8		0x0000
0 0 1 0 0 0	0 1 0 0 0 0	1 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		
2	1	0	8	0	0 0 0 0
I Op	Rs	Rt	Endereço (16 bits)		
b. sw \$t1, 32(\$t2) M[R[rs]+SignExtImm] = R[rt]				OPCODE = 0X2b	
0X2b		10	9	32	
1 0 1 0 1 1	0 1 0 1 0 0	1 0 0 1 0 0	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0		
10	13	4	9	0	0 2 0

2.10.4 [5] <2.4, 2.5> Mostre a representação hexadecimal dessas instruções.

**a. 21080000<sub>hex</sub>**  
**b. ad490020<sub>hex</sub>**

2.10.5 [5] <2.5> Que tipo (tipo I, tipo R) essas instruções representam?

**Ambas são do tipo I.**

2.10.6 [5] <2.5> Qual é a representação binária e hexadecimal dos campos opcode, Rs e Rt nessa instrução? Para as instruções de tipo R, qual é a representação hexadecimal dos campos Rd e funct? Para as instruções de tipo I, qual é a representação hexadecimal do campo imediato?

**É possível observar as notações hexadecimais e binárias nos exercícios anteriores.**

### **Exercício 2.16**

Para estes problemas, a tabela mantém diversos valores binários para o registrador \$t0.

Dado o valor de \$t0, você deverá avaliar o resultado de diferentes desvios.

a. \$t0 = 0010 0100 1001 0010 0100 1001 0010 0100<sub>dois</sub>

b. \$t0 = 0101 1111 1011 1110 0100 0000 0000 0000<sub>dois</sub>

2.16.1 [5] <2.7> Suponha que o registrador \$t0 contenha um desses valor e \$t1 tenha o valor

\$t1 = 0011 1111 1111 1000 0000 0000 0000 0000<sub>dois</sub>

Note o resultado da execução de tais instruções em certos registradores. Qual é o valor de \$t2 depois das seguintes instruções?

```
    slt $t2, $t0, $t1
    beq $t2, $ZERO, ELSE
    j DONE
ELSE: addi $t2, $0, 2
DONE:
```

**No caso A, o valor final de \$t2 será 1**

**No caso B, o valor final de \$t2 será 0**

2.16.4 [5] <2.7> Suponha que o registrador \$t0 contenha um valor da tabela anterior.

Qual é o valor de \$t2 após as instruções a seguir?

```
    slt $t2, $0, $t0
    bne $t2, $ZERO, ELSE
    j DONE
ELSE: addi $t2, $t2, 2
DONE:
```

**Em ambos os casos o valor de \$t2 será 1**