2 - Instruções: A linguagem de Máquina
2.21 Exercícios[1]

[1] 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_{dois}
b. 0000 0001 0100 1011 0100 1000 0010 0010_{dois}

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_{dois}

add $s0, $s0, $s0				R[rd] = R[rs] + R[rt] ($0/20_{hex}$)			
0x00	16	16	16	0	(0x20)		
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	1 0 0 0 0 0				
0	2	1	0	8	0	2	0

b. 0000 0001 0100 1011 0100 1000 0010 0010_{dois}

sub $t2, $t2, $t2				R[rd] = R[rs] - R[rt] ($0/22_{hex}$)			
0x00	10	10	10	0	(0x22)		
0 0 0 0 0 0 0 1 0 1 0 0 1 0 1 1	0 1 0 0 1	0 0 0 0 0	1 0 0 0 1 0				
0	1	6	B	6	8	2	2

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

a) Esta entrada binária representa uma operação *add*.

b) Esta entrada binária representa uma operação *sub*.

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

a) É classificada como uma instrução do tipo R.

b) Também é classificada como uma instrução 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) 0x02108020

b) 0x016B6822

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)				

a. 0010 0001 0000 1000 0000 0000 0000 0000

a. addi $t0 ,$t0 ,0			R[rt] = R[rs] + SignExtImm (8_{hex})
0x08	8	8	0x0
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
2	1	0	8	0	0	0	0

b. 1010 1101 0010 1000 0000 0000 0010 0000

b. sw $t1, 32($t2)			M[R[rs]+SignExtImm] = R[rt] ($2b_{hex}$)
0x2B	9	8	0x100000
1 0 1 0 1 1	0 1 0 0 1	0 1 0 0 0	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
A	D	2	8	0	0	2	0

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

a) 0x21080000

b) 0xAD280020

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

a) Esta instrução é classificada sendo do tipo I.

b) Esta instrução também é classificada sendo 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?

Tabela 1 – <u>TIPO R</u>

a)

opcode 000000_{bin} → $0x00_{hex}$
rs 10000_{bin} → $0x10_{hex}$
rt 10000_{bin} → $0x10_{hex}$
rd $0x10_{hex}$
funct $0x20_{hex}$

b)

opcode 000000_{bin} → $0x00_{hex}$
rs 01010_{bin} → $0x0A_{hex}$
rt 01010_{bin} → $0x0A_{hex}$
rd $0x0A_{hex}$
funct $0x22_{hex}$

Tabela 2 – <u>TIPO I</u>

a)

opcode 001000_{bin} → $0x08_{hex}$
rs 01000_{bin} → $0x08_{hex}$
rt 01000_{bin} → $0x08_{hex}$
im $0x0_{hex}$

b)

opcode 101011_{bin} → $0x2B_{hex}$
rs 01001_{bin} → $0x09_{hex}$
rt 01000_{bin} → $0x08_{hex}$
im 100000_{bin} → $0x20_{hex}$

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_{dois}
b. $t0 = 0101 1111 1011 1110 0100 0000 0000 0000_{dois}

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_{dois}

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:
```

a) $t2 = 1
b) $t2 = 2

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:
```

a) $t2 = 3
b) $t2 = 3