Aluno: Davi Ferreira Puddo

Questões Fechadas

Parte 1 -

1. A
2. B
3. A
4. C
5. D
6. C
7. D
8. D
9. B
10. A
11. B
12. A
13. B
14. C
15. A
16. D
17. A
18. A

Parte 2 -

1. C
2. B
3. A
4. C
5. B
6. A
7. D
8. B
9. B
10. A

Programas MIPS

1 -

```
# a = s1
# b = s2
# c = s3
# d = s4
# x = s5
# y = s6

.text

.globl main

main:

# Atribuicoes
ori $s1 $0 2    # a = 2
ori $s2 $0 3    # b = 3
ori $s3 $0 4    # c = 4
ori $s4 $0 5    # d = 5
```

```
# Contas

# x = a + b - c - d
add $s5 $s1 $s2 # x = a + b
sub $s5 $s5 $s3 # x = x - c
sub $s5 $s5 $s4 # x = x - d

# y = a - b + x
sub $s6 $s1 $s2 # y = a - b
add $s6 $s6 $s5 # y = y + x

# b = x - y
sub $s2 $s5 $s6
```

2 -

```
# x = s1
# y = s2

.text
.globl main

main:

# x = 1
ori $s1 $0 1

# y = 5x+15
add $s2 $s1 $s1 # y = 2x = 2
add $s2 $s2 $s2 # y = 2y = 4
add $s2 $s2 $s1 # y = y + x = 5
addi $s2 $s2 15 # y = y + 15
```

3 -

```
# x = s1
# y = s2
# z = s3

.text
.globl main

main:

# Atribuicoes
ori $s1 $0 3 # x = 3
ori $s2 $0 4 # y = 4

# Contas

# z = (15x + 67y)*4

  # t1 = 15x
add $t1 $s1 $s1 # t1 = 2x
add $t1 $t1 $t1 # t1 = 4x
add $t1 $t1 $t1 # t1 = 8x
add $t1 $t1 $t1 # t1 = 16x
sub $t1 $t1 $s1 # t1 = 15x

  # t2 = 67y
add $t2 $s2 $s2 # t2 = 2y
add $t2 $t2 $t2 # t2 = 4y
add $t2 $t2 $t2 # t2 = 8y
add $t2 $t2 $t2 # t2 = 16y
add $t2 $t2 $t2 # t2 = 32y
add $t2 $t2 $t2 # t2 = 64y
add $t2 $t2 $s2 # t2 = 65y
add $t2 $t2 $s2 # t2 = 66y
add $t2 $t2 $s2 # t2 = 67y
```

```
  # t1 = t1 + t2
add $t1 $t1 $t2 # t1 = (15x + 67y)

  # s3 = 4t1
add $s3 $t1 $t1 # s1 = 2t1
add $s3 $s3 $s3 # s1 = 4t1
```

4 -

```
# x = s1
# y = s2
# z = s3

.text
.globl main

main:

# Atribuicoes
ori $s1 $0 3 # x = 3
ori $s2 $0 4 # y = 4

# Contas

# z = (15x + 67y)*4

  # z = 15x
sll $s3 $s1 4   # z = 16x
sub $s3 $s3 $s1 # z = 15x

  # t1 = 67y
sll $t1 $s2 6   # t1 = 64y
add $t1 $t1 $s2 # t1 = 65y
add $t1 $t1 $s2 # t1 = 66y
add $t1 $t1 $s2 # t1 = 67y

  # z = 4(z + t1)
add $s3 $s3 $t1 # z = z + t1
sll $s3 $s3 2   # z = 4z
```

5 -

```
# x = s1
# y = s2
# z = s3

.text
.globl main

main:

# Atribuicoes

  # x = 100000
ori $s1 $0 0xC350 # x = 50000
sll $s1 $s1 1     # x = 2x = 100000

  # y = 200000
ori $s2 $0 0xC350 # y = 50000
sll $s2 $s2 2     # y = 2x = 200000

  # z = x + y
add $s3 $s1 $s2
```

6 -

```
# x = s1
# y = s2
# z = s3

.text
.globl main

main:

# Atribuicoes

  # x = 0x7FFF FFFF
ori $s1 $0 0x7FFF   # x = 0x0000 7FFF
sll $s1 $s1 16      # x = 0x7FFF 0000
ori $s1 $s1 0xFFFF  # x = 0x7FFF FFFF

  # y = 300000
ori $s2 $0 0x927C
sll $s2 $s2 3

  # z = x - 4y
sll $t1 $s2 2       # t1 = 4y
sub $s3 $s1 $t1     # z = x - t1
```

7 -

```
# x = s1
# y = s2
# z = s3

.text
.globl main

main:

ori $8 $0 0x01  # t0 = 1
srl $8 $8 1     # t0 = 0
nor $8 $8 $8    # t0 = 0xFFFF FFFF
```

8 -

```
# x = s1
# y = s2
# z = s3

.text
.globl main

main:

# t0 = 0x1234 5678
ori $8 $0 0x1234    # t0 = 0x0000 1234
sll $8 $8 16        # t0 = 0x1234 0000
ori $8 $8 0x5678    # t0 = 0x1234 5678

# t1 = 0x12
srl $9 $8 24

# t2 = 0x34
srl $10 $8 16       # t2 = 0x1234
andi $10 $10 0xFF   # t2 = 0x0034

# t3 = 0x56
srl $11 $8 8        # t3 = 0x0012 3456
andi $11 $11 0xFF   # t3 = 0x0000 0056

# t4 = 0x78
andi $12 $8 0xFF
```

9 -

```
.data
x1: .word 15
x2: .word 25
x3: .word 13
x4: .word 17
soma: .word -1

.globl main

.text

main:

# Carregar posicao
lui $t0 0x1001

# Ler valores
lw $t1 0($t0)       # t1 = x1
lw $t2 4($t0)       # t2 = x2

# Somar valores
add $t3 $t1 $t2 # t3 = t1 + t2

# Ler valores
lw $t1 8($t0)       # t1 = x3
lw $t2 12($t0)      # t2 = x4

# Somar valores
add $t3 $t3 $t1 # t3 = t3 + t1
add $t3 $t3 $t2 # t3 = t3 + t2

# Escrever resultado
sw $t3 16($t0)
```

10 -

```
.data
x: .word 5
z: .word 7
y: .word 0

.globl main

.text

# y = 127x - 65z + 1
main:

# Carregar posicao
lui $t0 0x1001

# Ler valores
lw $t1 0($t0)   # t1 = x
lw $t2 4($t0)   # t2 = z

# t1 = 127x
sll $t3 $t1, 7      # t3 = 128x
sub $t1 $t3 $t1     # t1 = 127x

# t2 = 65z
sll $t3 $t2 6       # t3 = 64z
add $t2 $t3 $t2     # t2 = 65z

# t1 = t1 - t2 + 1
sub $t1 $t1 $t2
addi $t1 $t1 1
```

```
# Escrever resultado
sw $t1, 8($t0)
```

11 -

```
.data
x: .word 100000
z: .word 200000
y: .word 0

.globl main

.text

# y = x - z + 300000
main:

# Carregar posicao
lui $t0, 0x1001

# Ler valores
lw $t1, 0($t0)  # t1 = x
lw $t2, 4($t0)  # t2 = z

# Contas

  # t3 = 300000
lui $t3, 0x4
ori $t3, $t3, 0x93E0

  # t1 = t1 - t2
sub $t1, $t1, $t2

  # t1 = t1 + t3
add $t1, $t1, $t3

# Escrever resultado
sw $t1, 8($t0)
```

12 -

```
.data
x: .word 25

.globl main

.text
main:

# Criar estrutura

  # Preparacao inicial
lui $t0 0x1001      # Endereco de x
ori $t1 $t0 0x0020

  # Gravar endereco de x em 0x1001 0020
sw $t0 0($t1)

  # Gravar endereco de x* em 0x1001 0024
sw $t1 4($t1)
addi $t2 $t1 4  # Guardar endereco de x** em t2

  # Gravar endereco de x** em 0x1001 0004
sw $t2 4($t0)

# Ler dados e fazer mulplicacao

  # Ler x***
lw $t1 4($t0)
```

```
  # Ler x**
lw $t1 0($t1)

  # Ler x*
lw $t1 0($t1)

  # Ler x
lw $t2 0($t1)

  # t2 = 2x
sll $t2 $t2 1

  # Gravar resultado
sw $t2 0($t1)
```

13 -

```
.data
#x: .word 25
x: .word -25

.globl main

.text

main:

lui $t0 0x1001  # t0 = endereco de x
lw $s0 0($t0)   # Ler x

lui $t2 0x8000  # t2 = 0x8000 0000
and $t1 $s0 $t2 # Comparar primeiro bit

bne $t2 $t1 pos # Pular se positivo

# Inverter x
sub $s0 $0 $s0

# Escrever resultado
sw $s0 0($t0)

pos:
```

14 -

```
.data
x: .word 23

.globl main

.text

main:

lui $t0 0x1001  # Endereco de x
lw $s0 0($t0)   # Ler x

andi $t1 $s0 0x1 # "Ler" ultimo bit

sw $t1 4($t0) # Escrever resultado
```

15 -

```
.data
arr: .word 0

.globl main
```

```
# t0 = Endereco
# t1 = Index
# t2 = Valor atual
# t3 = Somatorio
# t4 = Ultima posicao

.text

main:

lui $t0 0x1001
addi $t4 $t0 400

loop:   # do

add $t2 $t1 $t1 # t2 = 2i
addi $t2 $t2 1  # t2 = 2i + 1

# Somar valores
add $t3 $t3 $t2

# Escrever valor
sw $t2 0($t0)

addi $t0 $t0 4  # Proximo endereco
addi $t1 $t1 1  # i++

bne $t4 $t0 loop # while (i < 100)

# Escrever somatorio
sw $t3 0($t0)
```

16 -

```
.globl main

.data
x: .word 0x186A0
y: .word 0x13880
z: .word 0x61A80

.text

# Operacao = (x * y) / z

main:

# Endereco dos valores
lui $t1 0x1001

# Ler valores
lw $s1 0($t1)
lw $s2 4($t1)
lw $s3 8($t1)

# Copiar valores
or $s6 $0 $s1   # s6 = s1
or $s7 $0 $s3   # s6 = s3

# Chamar funcao de divisao
jal div_s5_s6_s7

# Copiar valores
or $s6 $0 $s5   # s6 = s5
or $s7 $0 $s2   # s7 = s2

# Chamar funcao de multiplicacao
jal mult_s5_s6_s7
```

```
or $s4 $0 $s5

j fim

# s5 = s6 * s7
mult_s5_s6_s7:
  or $t1 $0 $s6  # Iterador
  or $s5 $0 $0   # Resultado

  loop1:
  add $s5 $s5 $s7
  addi $t1 $t1 -1

  bne $t1 $0 loop1

  jr $ra

# s5 = s6 / s7
div_s5_s6_s7:
  or $t1 $0 $s6 # Iterador
  or $s5 $0 $0  # Resultado

  loop2:
  addi $s5 $s5 1
  sub $t1 $t1 $s7
  srl $t2 $t1 31

  beq $t2 $0 loop2

  addi $s5 $s5 -1
  jr $ra

fim:
```

17 -

```
.data
x: .word 8
y: .word 5

.globl main

.text

# Operacao = (x * y)

main:

# Endereco dos valores
lui $s0 0x1001

# Ler valores
lw $s1 0($s0)
lw $s2 4($s0)

jal mult_s3_s1_s2 # Chamar funcao

sw $s3 8($s0) # Escrever resultado

j fim

# s3 = s1 * s2
mult_s3_s1_s2:

  or $t1 $0 $s1 # Iterador
  or $s3 $0 $0  # Resultado

  loop1:
  add $s3 $s3 $s2
  addi $t1 $t1 -1
  bne $t1 $0 loop1
  jr $ra

fim:
```

18 -

```
.data
x: .word 8
y: .word 5

.globl main

.text
# Operacao = x^y
main:

# Endereco dos valores
lui $s0 0x1001

# Ler valores
lw $s3 0($s0)
lw $s4 4($s0)

# Calcular potencia
jal pot_s2_s3_s4

# Escrever resultado
sw $s2 8($s0)

j fim

# s5 = s6 * s7
mult_s5_s6_s7:

  or $t1 $0 $s6 # Iterador
  or $s5 $0 $0  # Resultado

  loop1:
  add $s5 $s5 $s7
  addi $t1 $t1 -1

  bne $t1 $0 loop1

  jr $ra

# s2 = s3^s4
pot_s2_s3_s4:

  or $t9 $0 $ra     # Endereco de retorno final
  addi $t8 $s4 -1
  or $s7 $s3 $0
  or $s2 $s3 $0

  loop2:
  or $s6 $s2 $0

  jal mult_s5_s6_s7
  or $s2 $0 $s5
  addi $t8 $t8 -1
  srl $t2 $t8 31

  bne $t8 $0 loop2

  or $ra $0 $t9
  jr $ra

fim:
```

19 -

...

...

...

20 -

```
.data
x: 5

.text

.globl main

# Operacao
# {x^4+x^3-2x^2} -> x = par
# {x^5-x^3+1} -> x = impar

main:

# Endereco de x
lui $t0 0x1001

# Ler x
lw $s0 0($t0)
andi $t1 $s0 0x1  # "Ler" ultimo bit

beq $t1 $zero par # par = 0 | impar = 1

impar:

mult $s0 $s0    # x^2
mflo $t3        # t3 = x^2

mult $t3 $s0    # x^3
mflo $t3        # t3 = x^3

mult $t3 $s0    # x^4
mflo $t2        # t2 = x^4

mult $t2 $s0    # x^5
mflo $t2        # t2 = x^5

sub $t2 $t2 $t3     # t2 = x^5 - x^3
addi $s1 $t2 1      # y = x^5 - x^3 + 1

j fim

par:

mult $s0 $s0    # x^2
mflo $t4        # t4 = x^2

mult $t4 $s0    # x^3
mflo $t3        # t3 = x^3

mult $t3 $s0    # x^4
mflo $t2        # t2 = x^4

sll $t4 $t4 1   # t4 = 2x^2

add $t2 $t2 $t3     # t2 = x^4 + x^3
sub $s1 $t2 $t4     # y = x^4 + x^3 - 2x^2

fim:

# Escrever resultado
sw $s1 4($t0)
```

21 -

```
.data
#x: .word 5
x: .word -5

.text

# x = s0
# y = s1

# Operacao =
# (x^3+1) -> x > 0
# (x^4-1) -> x <= 0

.globl main
main:

# Endereco de x
lui $t0 0x1001

# Ler x
lw $s0 0($t0)

slti $t1 $s0 1 # t1 = 1 -> (x < 1) | t1 = 0 -> (x >= 1)

beq $t1 $0 mz  # se t1 = 0 | (x > 0)

# Menor igual a zero

mult $s0 $s0 # x^2
mflo $t2     # t2 = x^2

mult $t2 $t2 # x^4
mflo $t2     # t2 = x^4

# y = x^4 - 1
addi $s1 $t2 -1

j fim

mz:          # Maior que zero
mult $s0 $s0 # x^2
mflo $t2     # t2 = x^2

mult $t2 $s0 # x^3
mflo $t2     # t2 = x^3

# y = x^3 + 1
addi $s1 $t2 1

fim:

# Escrever resultado
sw $s1 4($t0)
```