PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Instituto de Ciências Exatas e Informática – ICEI

Departamento de Ciência da Computação

TRABALHO PRÁTICO 05

Belo Horizonte

2022

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Ciência da Computação

Hernane Velozo Rosa
hernane.rosa@sga.pucminas.br

Este trabalho apresenta respostas às perguntas teóricas envolvendo a linguagem Assembly no MIPS, além de conter os códigos dos exercícios práticos propostos.

Professor: Dr. Romanelli Lodron Zuim
Disciplina: Arquitetura de Computadores II

Palavras chaves: Instruções, compilador, programação de baixo-nível e MIPS.

RELATÓRIO: PARTE 01 - TEÓRICO

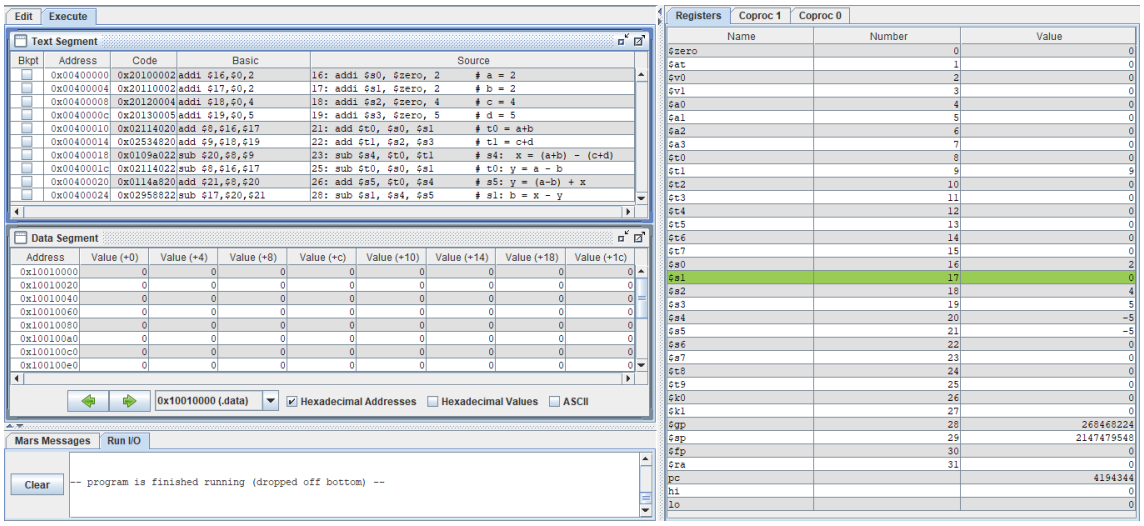
1. A) Um arquivo de texto que contém instruções de linguagem de programação.
2. B) Uma parte do processador que possui um padrão de bits.
3. A) #
4. C) 32 bits
5. D) parte do processador que contém o endereço da próxima instrução de máquina para ser obtida.
6. D) 8
7. A) Uma instrução em linguagem assembly que resulta em uma instrução em linguagem de máquina.
8. B) Um byte na memória que contém o endereço de dados.
9. A) 0x00000000
10. A) Operando imediato
11. A) Operação lógica
12. C) Ambos os operandos devem que vir de registros.
13. B) Os dados são estendidos em zero à esquerda por 16 bits.
14. C) ori $5,$0,48
15. A) Não.
16. D) andi $8,$8,0xFF
17. A) Todos os bits em zero.
18. D) Sim. Todas as instruções de máquina têm os mesmos campos, mas eles podem estar em ordens diferentes.

RELATÓRIO: PARTE 02 - PRÁTICA

Programa 1 – Código

```
# Programa 1 - By Hernane V | Data 18/11/2022
# Arquitetura de Computadores II - Romanelli

# a = 2
# b = 2
# c = 4
# d = 5
# x = (a+b) - (c + d)
# y = a - b + x
# b = x - y

.text
.globl main
main:

addi $s0, $zero, 2      # a = 2
addi $s1, $zero, 2      # b = 2
addi $s2, $zero, 4      # c = 4
addi $s3, $zero, 5      # d = 5

add $t0, $s0, $s1       # t0 = a+b
add $t1, $s2, $s3       # t1 = c+d
sub $s4, $t0, $t1       # s4:  x = (a+b) - (c+d)

sub $t0, $s0, $s1       # t0: y = a - b
add $s5, $t0, $s4       # s5: y = (a-b) + x

sub $s1, $s4, $s5       # s1: b = x - y
.data
#fim
```

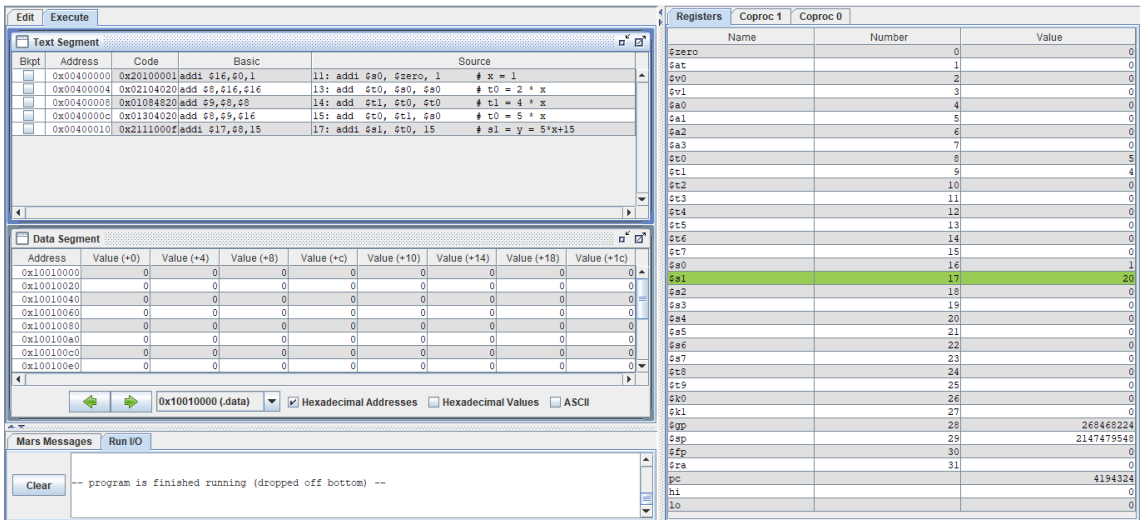
Programa 1 – Execução via MARS



Programa 2 – Código

```
# Programa 2 - By Hernane V | Data 18/11/2022
# Arquitetura de Computadores II - Romanelli

# x = 1
# y = 5 * x + 15

.text
.globl main
main:

addi $s0, $zero, 1      # x = 1

add  $t0, $s0, $s0      # t0 = 2 * x
add  $t1, $t0, $t0      # t1 = 4 * x
add  $t0, $t1, $s0      # t0 = 5 * x

addi $s1, $t0, 15       # s1 = y = 5*x+15

.data

#fim
```

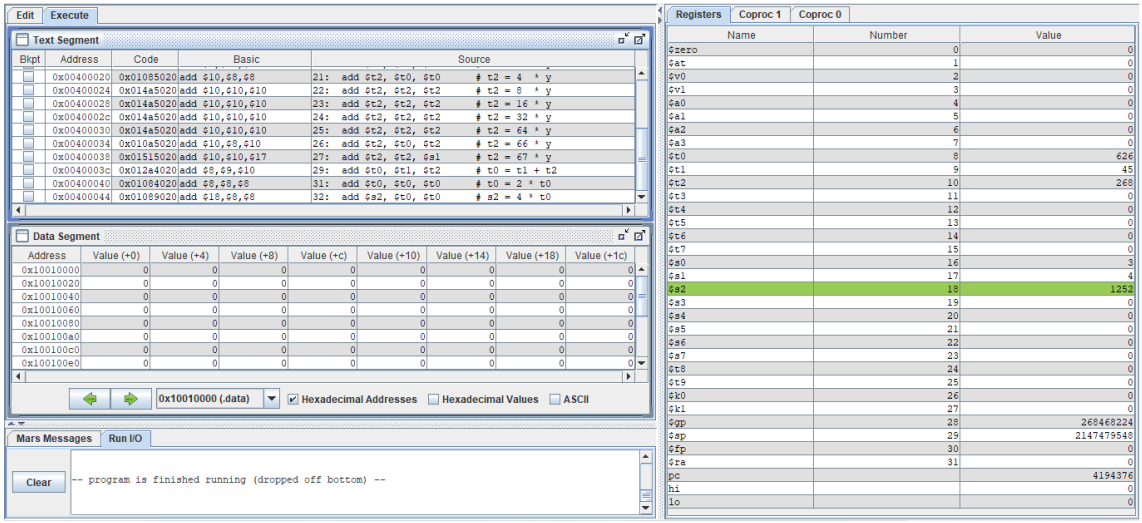
Programa 2 – Execução via MARS



Programa 3 – Código

```
# Programa 3 - By Hernane V | Data 18/11/2022
# Arquitetura de Computadores II - Romanelli

# x = 3
# y = 4
# z = (15*x + 67*y) * 4

.text
.globl main
main:
addi $s0, $zero, 3      # x = 3
addi $s1, $zero, 4      # y = 4

add $t0, $s0, $s0       # t0 = 2  * x
add $t0, $t0, $t0       # t0 = 4  * x
add $t0, $t0, $t0       # t0 = 8  * x
add $t0, $t0, $t0       # t0 = 16 * x
sub $t1, $t0, $s0       # t1 = 15 * x

add $t0, $s1, $s1       # t0 = 2  * y
add $t2, $t0, $t0       # t2 = 4  * y
add $t2, $t2, $t2       # t2 = 8  * y
add $t2, $t2, $t2       # t2 = 16 * y
add $t2, $t2, $t2       # t2 = 32 * y
add $t2, $t2, $t2       # t2 = 64 * y
add $t2, $t0, $t2       # t2 = 66 * y
add $t2, $t2, $s1       # t2 = 67 * y

add $t0, $t1, $t2       # t0 = t1 + t2

add $t0, $t0, $t0       # t0 = 2 * t0
add $s2, $t0, $t0       # s2 = 4 * t0

.data
#fim
```

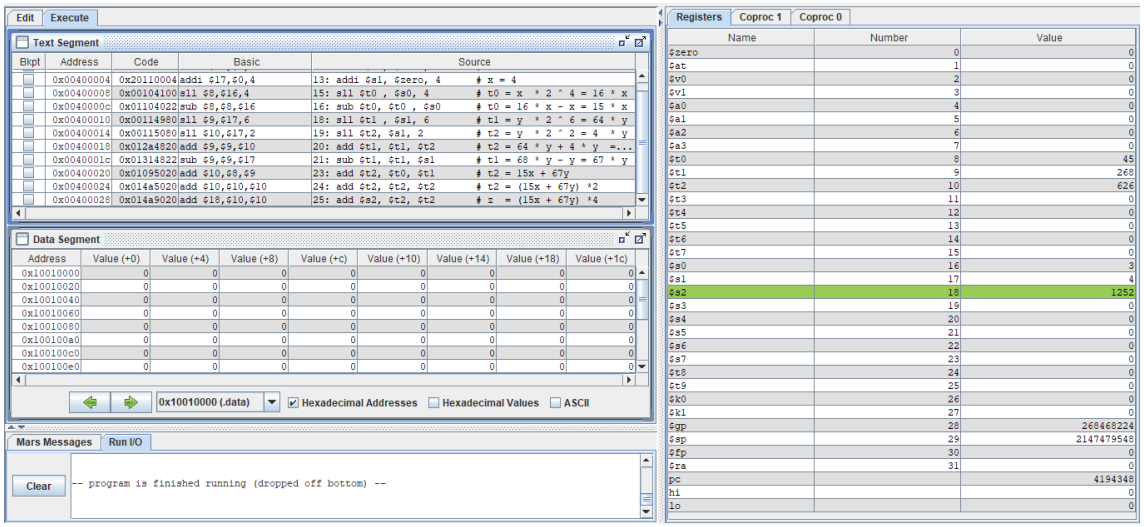
Programa 3 – Execução via MARS



Programa 4 – Código

```
# Programa 4 - By Hernane V | Data 18/11/2022
# Arquitetura de Computadores II - Romanelli

# x = 3
# y = 4
# z = (15*x + 67*y) * 4

.text
.globl main
main:

addi $s0, $zero, 3      # x = 3
addi $s1, $zero, 4      # x = 4

sll $t0 , $s0, 4        # t0 = x  * 2 ^ 4 = 16 * x
sub $t0, $t0 , $s0      # t0 = 16 * x - x = 15 * x

sll $t1 , $s1, 6        # t1 = y  * 2 ^ 6 = 64 * y
sll $t2, $s1, 2         # t2 = y  * 2 ^ 2 = 4  * y
add $t1, $t1, $t2       # t2 = 64 * y + 4 * y  = 68 * y
sub $t1, $t1, $s1       # t1 = 68 * y - y = 67 * y

add $t2, $t0, $t1       # t2 = 15x + 67y
add $t2, $t2, $t2       # t2 = (15x + 67y) *2
add $s2, $t2, $t2       # z  = (15x + 67y) *4

.data

#fim
```

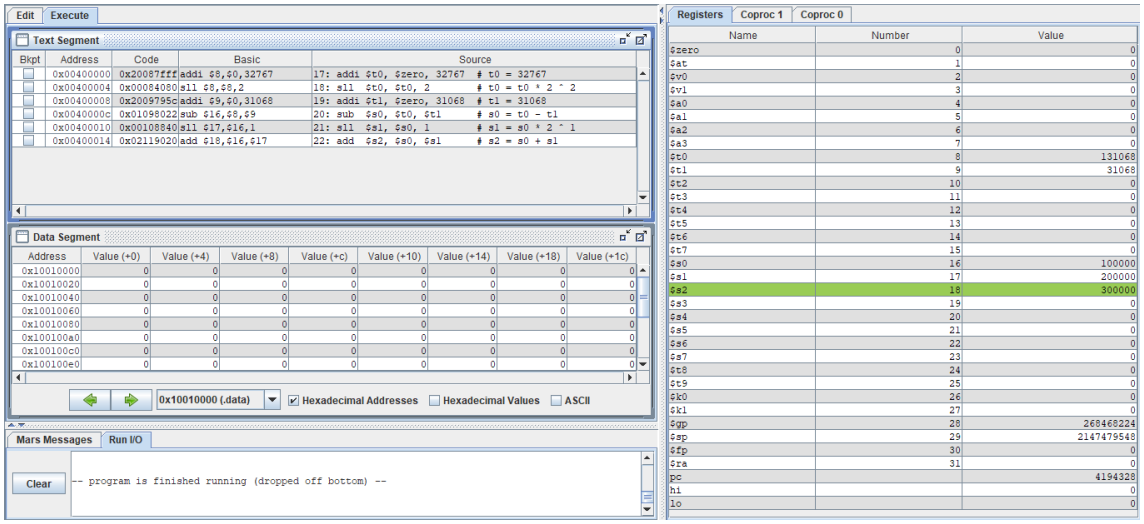
Programa 4 – Execução via MARS



Programa 5 – Código

```
# Programa 5 - By Hernane V | Data 18/11/2022
# Arquitetura de Computadores II - Romanelli

# x = 3
# y = 4
# z = (15*x + 67*y) * 4

.text
.globl main
main:


# = x
# = y
# = z

addi $t0, $zero, 32767          # t0 = 32767
sll  $t0, $t0, 2        # t0 = t0 * 2 ^ 2
addi $t1, $zero, 31068  # t1 = 31068
sub  $s0, $t0, $t1      # s0 = t0 - t1
sll  $s1, $s0, 1        # s1 = s0 * 2 ^ 1
add  $s2, $s0, $s1      # s2 = s0 + s1
.data

#fim
```

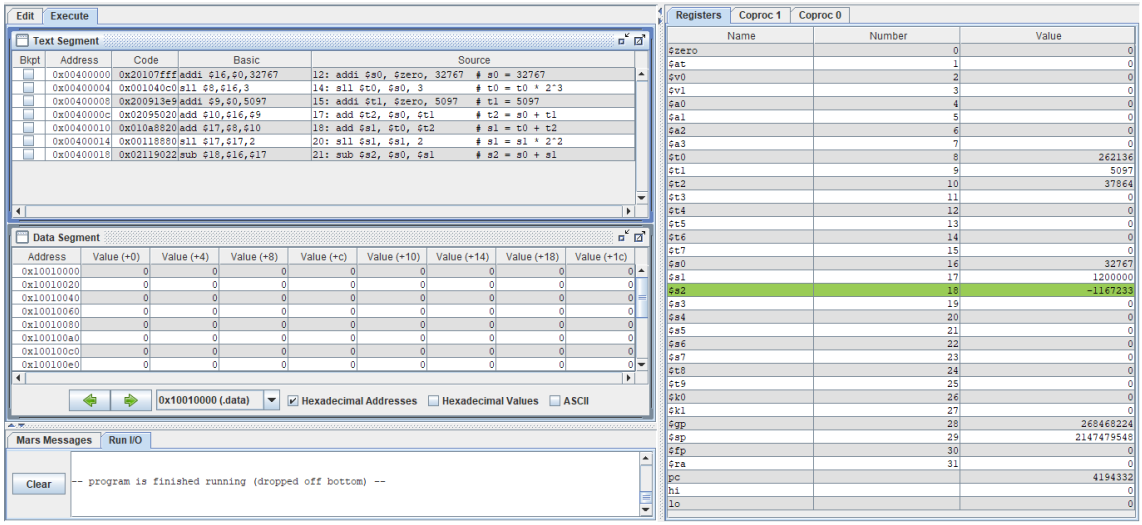
Programa 5 – Execução via MARS



Programa 6 – Código

```
# Programa 6 - By Hernane V | Data 18/11/2022
# Arquitetura de Computadores II - Romanelli

# x = 32767
# y = 300000
# z = x - 4y

.text
.globl main
main:

addi $s0, $zero, 32767 # s0 = 32767

sll $t0, $s0, 3        # t0 = t0 * 2^3
addi $t1, $zero, 5097  # t1 = 5097

add $t2, $s0, $t1      # t2 = s0 + t1
add $s1, $t0, $t2      # s1 = t0 + t2

sll $s1, $s1, 2        # s1 = s1 * 2^2
sub $s2, $s0, $s1      # s2 = s0 + s1
.data

#fim
```

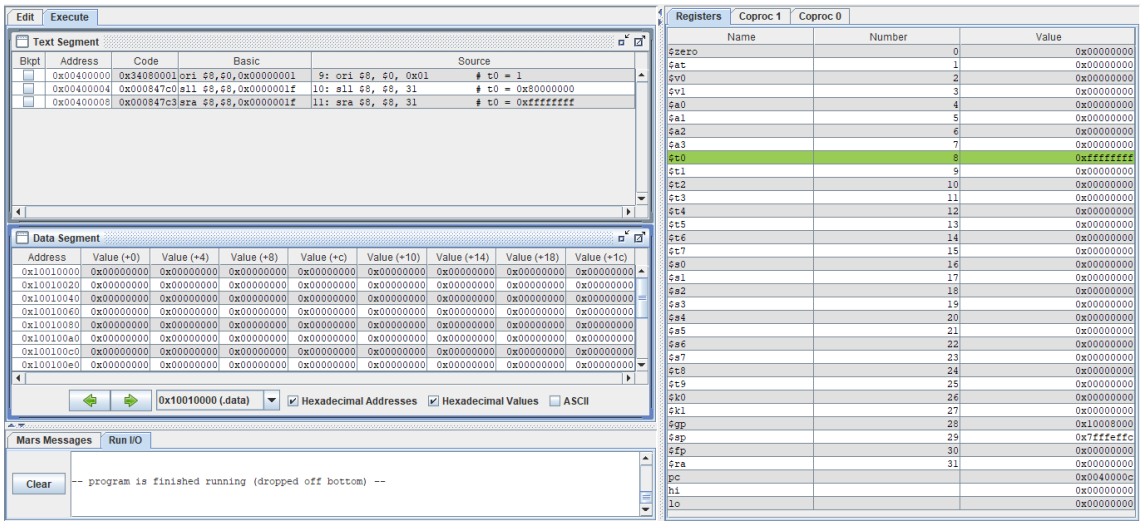
Programa 6 – Execução via MARS



Programa 7 – Código

```
# Programa 7 - By Hernane V | Data 18/11/2022
# Arquitetura de Computadores II - Romanelli

# ori $8, $0, 0x01

.text
.globl main
main:
ori $8, $0, 0x01        # t0 = 1
sll $8, $8, 31          # t0 = 0x80000000
sra $8, $8, 31          # t0 = 0xffffffff

.data

#fim
```

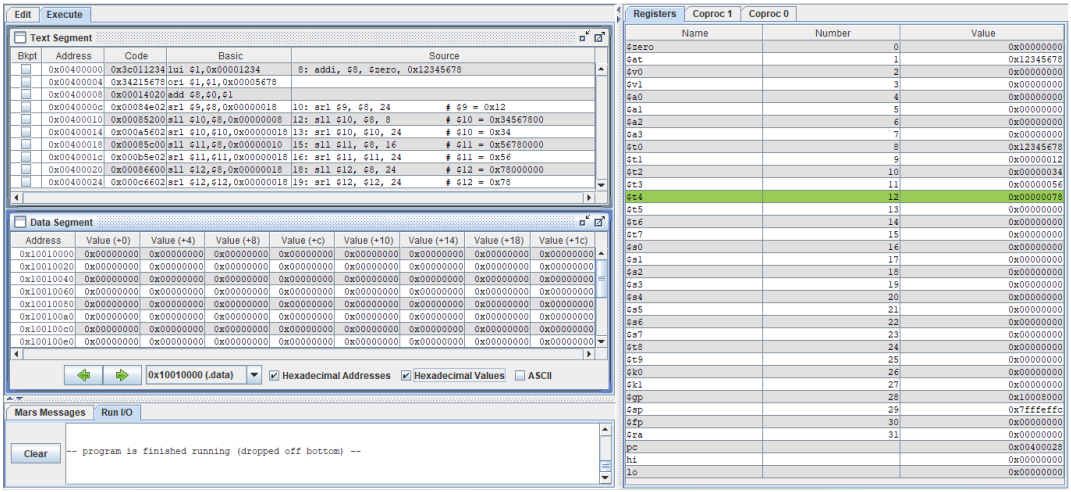
Programa 7 – Execução via MARS



Programa 8 – Código

```
# Programa 7 - By Hernane V | Data 18/11/2022
# Arquitetura de Computadores II - Romanelli

.text
.globl main
main:

addi, $8, $zero, 0x12345678

srl $9, $8, 24          # $9 = 0x12

sll $10, $8, 8          # $10 = 0x34567800
srl $10, $10, 24        # $10 = 0x34

sll $11, $8, 16         # $11 = 0x56780000
srl $11, $11, 24        # $11 = 0x56

sll $12, $8, 24         # $12 = 0x78000000
srl $12, $12, 24        # $12 = 0x78
.data
#fim
```

Programa 8 – Execução via MARS



REPOSITÓRIO DE PMG-AC2

Disponível em: https://github.com/hernanevelozo/PMG-AC2

REFERÊNCIAS:

HENNESSY, John L. **Arquitetura de computadores uma abordagem quantitativa**. Rio de Janeiro **GEN LTC 2019 1** recurso online ISBN 9788595150669.

SPRINGERLINK (ONLINE SERVICE). **The MIPS-X RISC Microprocessor**. 1st **ed. 1989. XXIV**, 232 p ISBN9781475767629.