



**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS
GERAIS**

Instituto de Ciências Exatas e de Informática

Exercício Prático — Arquitetura de Computadores II

Vitor Costa Oliveira Rolla

Pontifícia Universidade Católica de Minas Gerais

19 de outubro de 2025

Parte 1 - Exercícios

O que é um arquivo fonte?

Um arquivo de texto que contém instruções de linguagem de programação.

O que é registrador?

É uma parte do processador que possui um padrão de bits.

Qual o caracter que, na linguagem assembly do SPIM, inicia um comentário?

O caractere que inicia um comentário em assembly é o caractere "#".

Quantos bits há em cada instrução de máquina MIPS?

Em cada instrução de máquina MIPS há um total de 32 bits.

O que é o contador de programa?

O contador de programa (program counter) é parte do processador que contém o endereço da próxima instrução de máquina para ser obtida.

Ao executarmos uma instrução, quanto será adicionado ao contador de programa?

Ao executarmos uma instrução, será adicionado 4 para o contador de programa.

O que é uma diretiva, tal como a diretiva .text?

Diretiva é uma declaração que diz o montador algo sobre o que o programador quer, mas não corresponde diretamente a uma instrução de máquina.

O que é um endereço simbólico?

O endereço simbólico é um nome usado no código-fonte em linguagem assembly para um local na memória.

Em qual endereço o simulador SPIM coloca a primeira instrução de máquina quando ele está sendo executado?

Ao executar, a primeira instrução se localiza no endereço 0x00400000.

Algumas instruções de máquina possuem uma constante como um dos operandos. Como é chamado tal operando?

Tal operando é nomeado como operando imediato.

Como é chamada uma operação lógica executada entre bits de cada coluna dos operandos para produzir um bit de resultado para cada coluna?

A operação utilizada nesse contexto é a operação bitwise.

Quando uma operação é de fato executada, como estão os operandos na ALU?

Ambos operandos devem vir de registros.

Dezesseis bits de dados de uma instrução de ori são usados como um operando imediato. Durante execução, o que deve ser feito primeiro?

Os dados são estendidos em zero à esquerda por 16 bits.

Qual das instruções seguintes armazenam no registrador \$5 um padrão de bits que representa positivo 48?

ori \$5,\$0, 48

A instrução de ori pode armazenar o complemento de dois de um número em um registrador

A resposta é não, pois se trata de operações lógicas e não aritméticas.

Qual das instruções seguintes limpa todos os bits no registrador \$8 com exceção do byte de baixa ordem que fica inalterado?

A instrução que realiza tal operação é andi \$8,\$8,0xFF, pois ao realizar o and entre os dados, ele mantém apenas o que for 1 entre os operandos.

Qual é o resultado de um ou exclusivo de padrão sobre ele mesmo?

O resultado é o contrário do original.

Todas as instruções de máquina têm os mesmos campos?

Não. Diferentes de instruções de máquina possuem campos diferentes.

Parte 2 - Exercícios

Programas

Programa - 01

```
programa_1.asm
12 # ori: Registrador Destino, Registrador A, Imediato -> Dest = A | I
13 # and: Registrador Destino, Registrador A, Registrador B -> Dest = A & B
14 # xor: Registrador Destino, Registrador A, Registrador B -> Dest = A XOR B
15 # nor: Registrador Destino, Registrador A, Registrador B -> Dest = A NOR B
16
17 .text
18 .globl main
19
20
21 main:
22
23     # Atribuição de valores
24     ori $a0, $zero, 2 # a
25     ori $a1, $zero, 3 # b
26     ori $a2, $zero, 4 # c
27     ori $a3, $zero, 5 # d
28
29     # Somando valores
30     add $t0, $a0, $a1 # Atribuição -> t0: a0 + a1 (a + b)
31     add $t1, $a2, $a3 # Atribuição -> t1: a2 + a3 (c + d)
32     sub $s4, $t0, $t1 # Atribuição -> (x)s4: t0 - t1
33
34     sub $t3, $a0, $a1 # Atribuição -> t3: a0 - a1 (a - b)
35     add $s5, $t3, $a4 # Atribuição -> (y)s5: t3 + a4 (t3 + x)
36     sub $a1, $a1, $a5 # Atribuição -> a1 = x - y
37
38
```

Figura 1: Imagem do programa 1

Programa - 02

```

2
3
4 # Aritméticas
5 # add: Registrador Destino, Registrador A, Registrador B -> Dest = A + B
6 # addi: Registrador Destino, Registrador A, Imediato -> Dest = A + Imediato
7 # sub: Registrador Destino, Registrador A, Registrador B -> Dest = A - B
8
9
10 # Lógicas
11 # andi: Registrador Destino, Registrador A, Imediato -> Dest = A & I
12 # ori: Registrador Destino, Registrador A, Imediato -> Dest = A | I
13 # and: Registrador Destino, Registrador A, Registrador B -> Dest = A & B
14 # xor: Registrador Destino, Registrador A, Registrador B -> Dest = A XOR B
15 # nor: Registrador Destino, Registrador A, Registrador B -> Dest = A NOR B
16
17 .text
18 .globl main
19
20
21 main:
22
23     ori $a0, $zero, 1 # Atribuição -> x = 1
24     add $t0, $a0, $a0 # Atribuição -> t0 = 2x
25     add $t0, $t0, $t0 # Atribuição -> t0 = 4x
26     add $t0, $t0, $a0 # Atribuição -> t0 = 5x
27     addi $a1, $t0, 15 # Atribuição -> y = 5x + 15
28
29
30
31
```

Figura 2: Imagem do programa 2

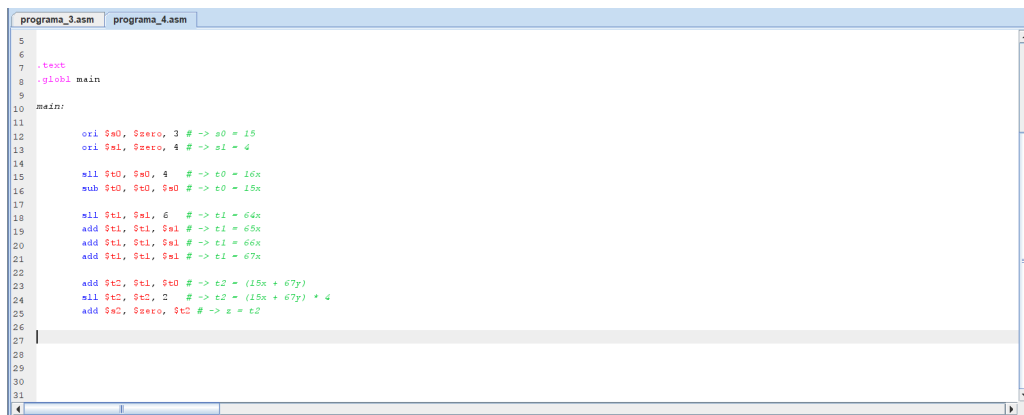
Programa - 03

```

21 main:
22
23     ori $a0, $zero, 3 # Atribuição a0 = 3
24     ori $a1, $zero, 4 # Atribuição a1 = 4
25
26     add $t0, $a0, $a0 # Atribuição t0 = 2x
27     add $t0, $t0, $t0 # Atribuição t0 = 4x
28     add $t0, $t0, $t0 # Atribuição t0 = 8x
29     add $t0, $t0, $t0 # Atribuição t0 = 16x
30     sub $t0, $t0, $a0 # Atribuição t0 = 15x
31
32     add $t1, $a1, $a1 # Atribuição t1 = 2y
33     add $t1, $t1, $a1 # Atribuição t1 = 4y
34     add $t1, $t1, $a1 # Atribuição t1 = 6y
35     add $t1, $t1, $a1 # Atribuição t1 = 8y
36     add $t1, $t1, $a1 # Atribuição t1 = 10y
37     add $t1, $t1, $a1 # Atribuição t1 = 12y
38     add $t1, $t1, $a1 # Atribuição t1 = 14y
39     add $t1, $t1, $a1 # Atribuição t1 = 16y
40     add $t1, $t1, $a1 # Atribuição t1 = 18y
41
42     add $t2, $t0, $a1 # Atribuição t2 = t0 + t1
43     add $t2, $t2, $t2 # Atribuição t2 = 2 * t2
44     add $t2, $t2, $t2 # Atribuição t2 = 4 * t2
45
46     add $a2, $zero, $t2 # Atribuição z = (15x + 67y) * 4
47
```

Figura 3: Imagem do programa 3

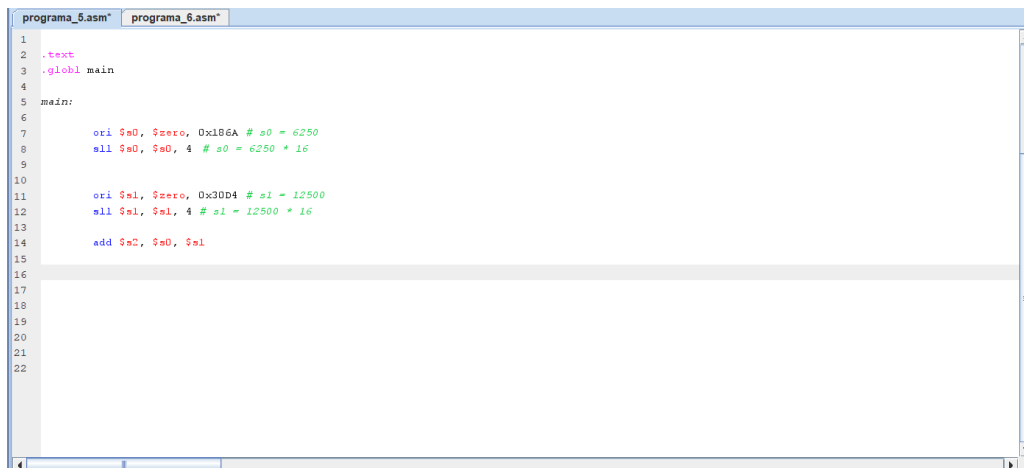
Programa - 04



```
programa_3.asm  programa_4.asm
5
6
7 .text
8 .globl main
9
10 main:
11
12     ori $a0, $zero, 3 # -> a0 = 15
13     ori $a1, $zero, 4 # -> a1 = 4
14
15     sll $t0, $a0, 4 # -> t0 = 16x
16     sub $t0, $t0, $a0 # -> t0 = 15x
17
18     sll $t1, $a1, 6 # -> t1 = 64x
19     add $t1, $t1, $a1 # -> t1 = 65x
20     add $t1, $t1, $a1 # -> t1 = 66x
21     add $t1, $t1, $a1 # -> t1 = 67x
22
23     add $t2, $t1, $t0 # -> t2 = (15x + 67y)
24     sll $t2, $t2, 2 # -> t2 = (15x + 67y) * 4
25     add $a2, $zero, $t2 # -> a2 = t2
26
27
28
29
30
31
```

Figura 4: Imagem do programa 4

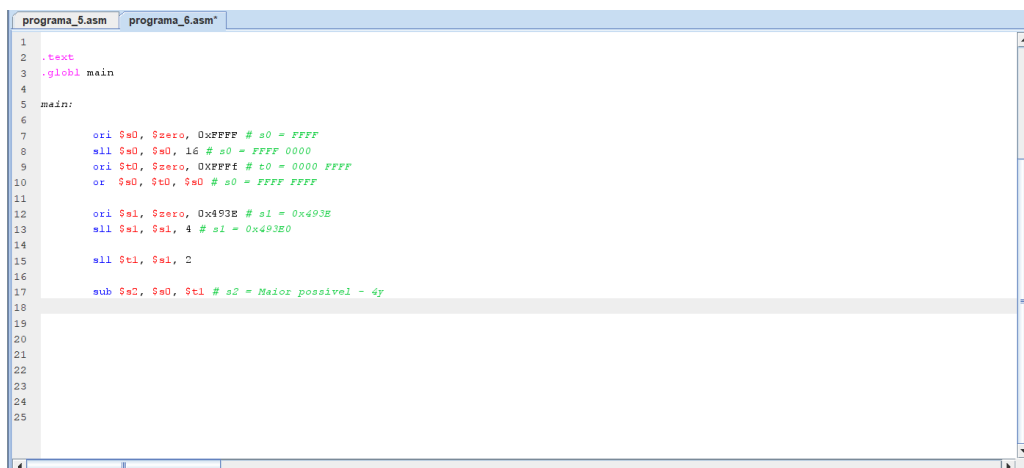
Programa - 05



```
programa_5.asm*  programa_6.asm*
1
2 .text
3 .globl main
4
5 main:
6
7     ori $a0, $zero, 0x186A # a0 = 6250
8     sll $a0, $a0, 4 # a0 = 6250 * 16
9
10
11
12     ori $a1, $zero, 0x30D4 # a1 = 12500
13     sll $a1, $a1, 4 # a1 = 12500 * 16
14
15     add $a2, $a0, $a1
16
17
18
19
20
21
22
```

Figura 5: Imagem do programa 5

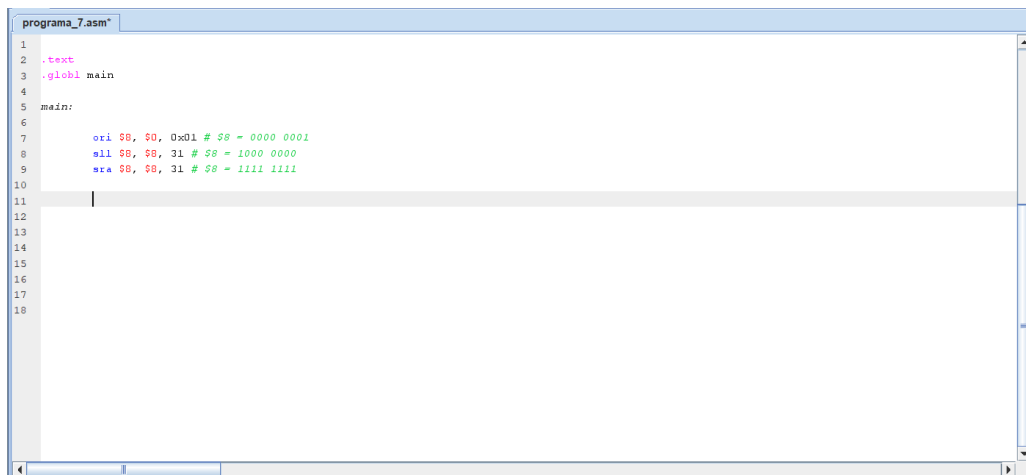
Programa - 06



```
programa_5.asm  programa_6.asm*
1
2 .text
3 .globl main
4
5 main:
6
7     ori $a0, $zero, 0xFFFF # a0 = FFFF
8     sll $a0, $a0, 16 # a0 = FFFF 0000
9     ori $t0, $zero, 0xFFFF # t0 = 0000 FFFF
10    or $a0, $t0, $a0 # a0 = FFFF FFFF
11
12    ori $a1, $zero, 0x493E # a1 = 0x493E
13    sll $a1, $a1, 4 # a1 = 0x493E0
14
15    sll $t1, $a1, 2
16
17    sub $a2, $a0, $t1 # a2 = Maior possivel - 4y
18
19
20
21
22
23
24
25
```

Figura 6: Imagem do programa 6

Programa - 07



```
programa_7.asm
1
2 .text
3 .globl main
4
5 main:
6
7     ori $0, $0, 0x01 # $0 = 0000 0001
8     sll $0, $0, 31 # $0 = 1000 0000
9     sra $0, $0, 31 # $0 = 1111 1111
10
11
12
13
14
15
16
17
18
```

Figura 7: Imagem do programa 7

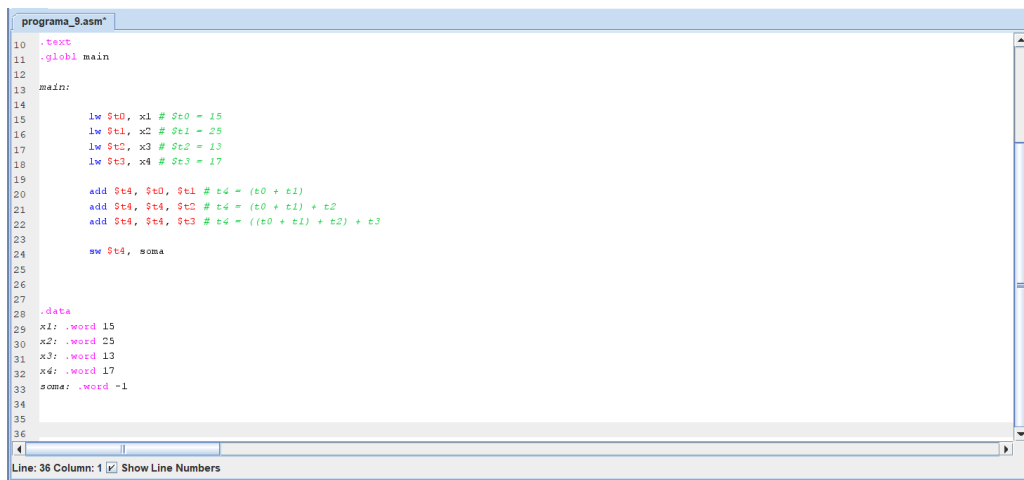
Programa - 08



```
programa_8.asm
6
7 .text
8 .globl main
9
10 main:
11
12     ori $0, $0, 0x1234 # t0 = 0x1234
13     sll $0, $0, 16 # t0 = 0x12340000
14     or $0, 0x5678 # t0 = 0x12345678
15
16     srl $9, $0, 24 # t1 = 0x00000012
17
18     srl $10, $0, 16 # t2 = 0x00001234
19     andi $10, $10, 0xFF # t2 = 0x00000034
20
21     srl $11, $0, 8 # t3 = 0x00123456
22     andi $11, $11, 0xFF # t3 = 0x00000056
23
24     andi $12, $0, 0xFF # t3 = 0x00000056
25
26
27
28
29
30
31
```

Figura 8: Imagem do programa 8

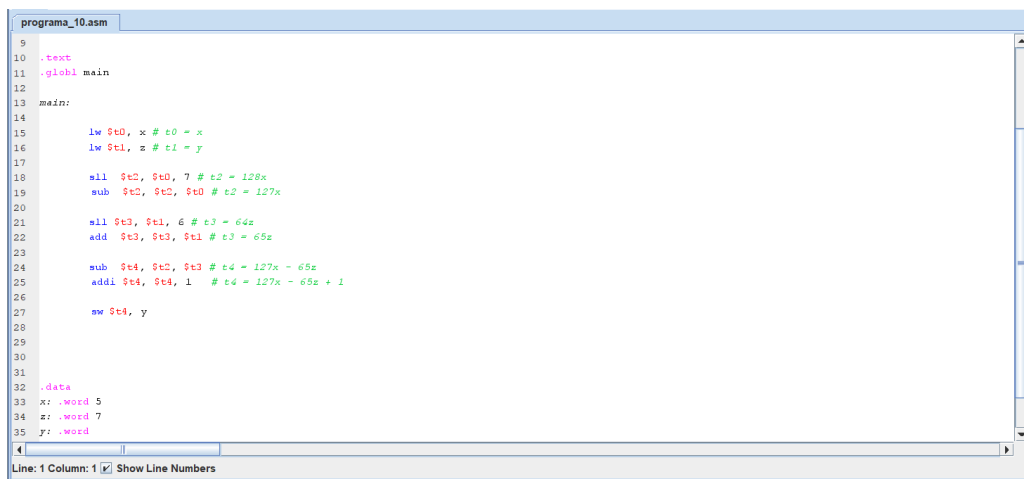
Programa - 09



```
programa_9.asm
10 .text
11 .globl main
12
13 main:
14
15     lw $t0, x1 # $t0 = 15
16     lw $t1, x2 # $t1 = 25
17     lw $t2, x3 # $t2 = 13
18     lw $t3, x4 # $t3 = 17
19
20     add $t4, $t0, $t1 # t4 = (t0 + t1)
21     add $t4, $t4, $t2 # t4 = (t0 + t1) + t2
22     add $t4, $t4, $t3 # t4 = ((t0 + t1) + t2) + t3
23
24     sw $t4, soma
25
26
27 .data
28 x1: .word 15
29 x2: .word 25
30 x3: .word 13
31 x4: .word 17
32 soma: .word -1
33
34
35
36
Line: 36 Column: 1 Show Line Numbers
```

Figura 9: Imagem do programa 9

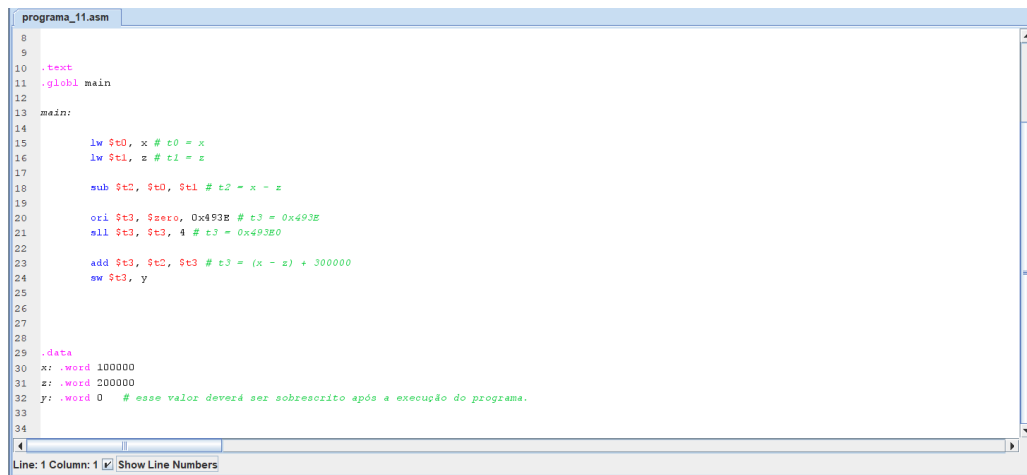
Programa - 10



```
programa_10.asm
9
10 .text
11 .globl main
12
13 main:
14
15     lw $t0, x # t0 = x
16     lw $t1, z # t1 = y
17
18     sll $t2, $t0, 7 # t2 = 128x
19     sub $t2, $t2, $t0 # t2 = 127x
20
21     sll $t3, $t1, 6 # t3 = 64z
22     add $t3, $t3, $t1 # t3 = 65z
23
24     sub $t4, $t2, $t3 # t4 = 127x - 65z
25     addi $t4, $t4, 1 # t4 = 127x - 65z + 1
26
27     sw $t4, y
28
29
30
31
32 .data
33 x: .word 5
34 z: .word 7
35 y: .word
36
Line: 1 Column: 1 Show Line Numbers
```

Figura 10: Imagem do programa 10

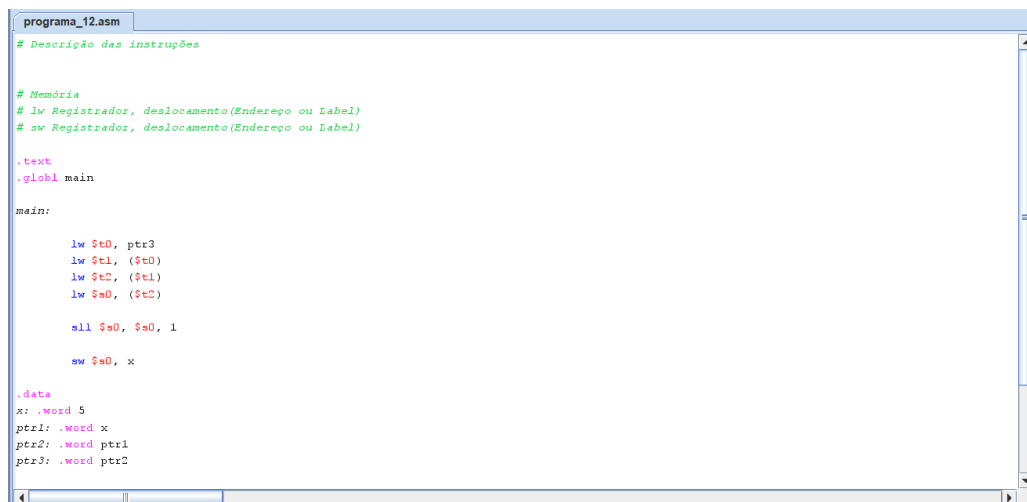
Programa - 11



```
programa_11.asm
8
9
10 .text
11 .globl main
12
13 main:
14
15     lw $t0, x # t0 = x
16     lw $t1, z # t1 = z
17
18     sub $t2, $t0, $t1 # t2 = x - z
19
20     ori $t3, $zero, 0x493E # t3 = 0x493E
21     sll $t3, $t3, 4 # t3 = 0x493E0
22
23     add $t3, $t2, $t3 # t3 = (x - z) + 300000
24     sw $t3, y
25
26
27
28
29 .data
30 x: .word 100000
31 z: .word 200000
32 y: .word 0 # esse valor deverá ser sobrescrito após a execução do programa.
33
34
Line: 1 Column: 1 Show Line Numbers
```

Figura 11: Imagem do programa 11

Programa - 12



```
programa_12.asm
# Descrição das instruções

# Memória
# lw Registrador, deslocamento(Endereço ou Label)
# sw Registrador, deslocamento(Endereço ou Label)

.text
.globl main

main:

    lw $t0, ptr3
    lw $t1, ($t0)
    lw $t2, ($t1)
    lw $s0, ($t2)

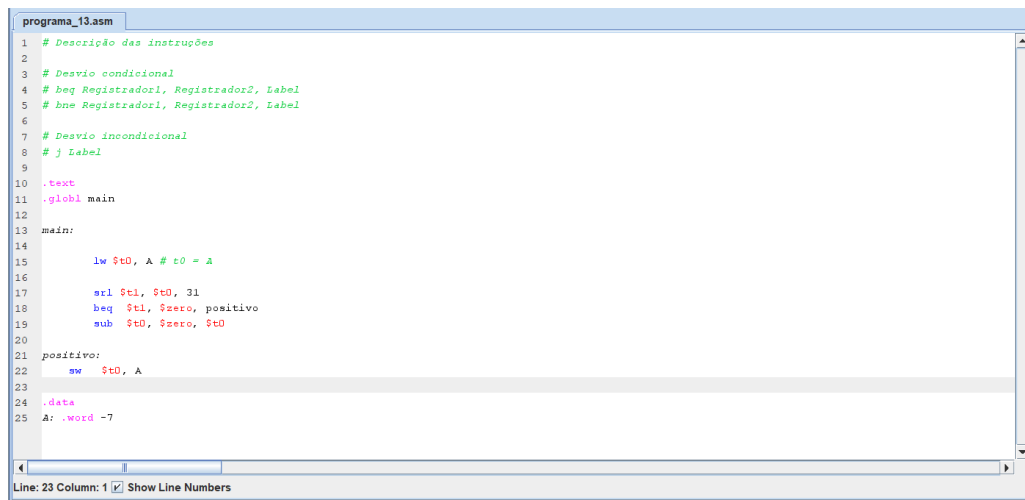
    sll $s0, $s0, 1

    sw $s0, x

.data
x: .word 5
ptr1: .word x
ptr2: .word ptr1
ptr3: .word ptr2
```

Figura 12: Imagem do programa 12

Programa - 13

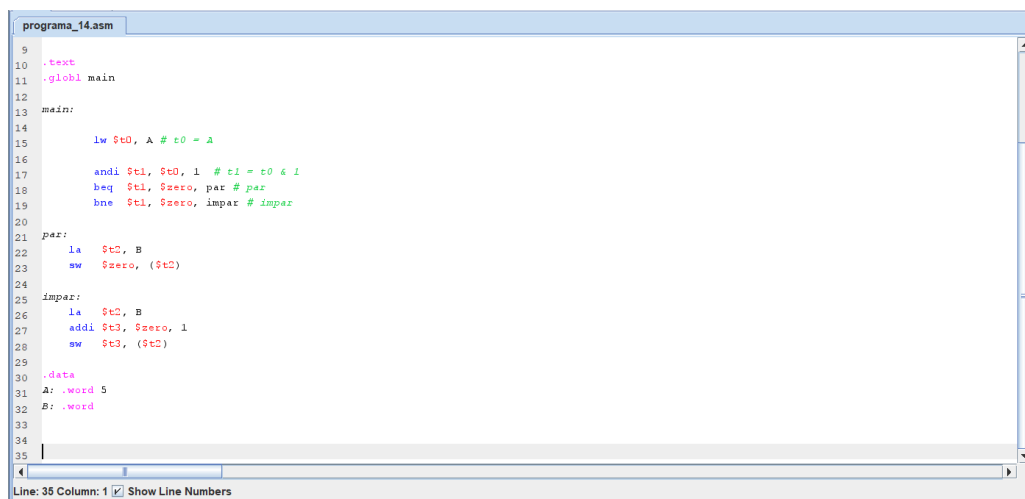


```
programa_13.asm
1  # Descrição das instruções
2
3  # Desvio condicional
4  # beq Registrador1, Registrador2, Label
5  # bne Registrador1, Registrador2, Label
6
7  # Desvio incondicional
8  # j Label
9
10 .text
11 .globl main
12
13 main:
14
15     lw $t0, A # t0 = A
16
17     srl $t1, $t0, 31
18     beq $t1, $zero, positivo
19     sub $t0, $zero, $t0
20
21 positivo:
22     sw $t0, A
23
24 .data
25 A: .word -7
```

Line: 23 Column: 1 ☒ Show Line Numbers

Figura 13: Imagem do programa 13

Programa - 14

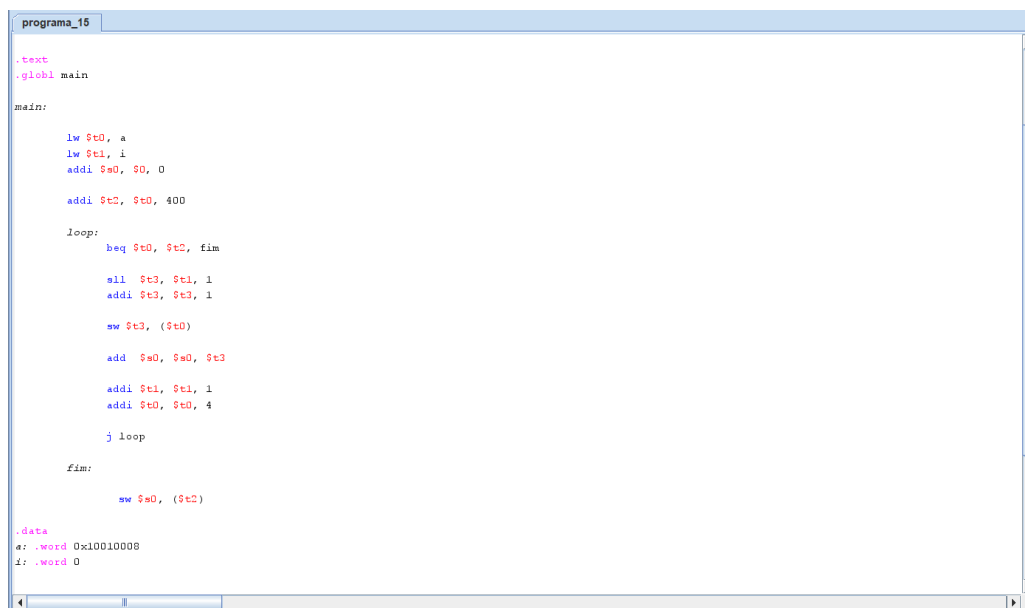


```
programa_14.asm
9
10 .text
11 .globl main
12
13 main:
14
15     lw $t0, A # t0 = A
16
17     andi $t1, $t0, 1 # t1 = t0 & 1
18     beq $t1, $zero, par # par
19     bne $t1, $zero, impar # impar
20
21 par:
22     la $t2, B
23     sw $zero, ($t2)
24
25 impar:
26     la $t2, B
27     addi $t3, $zero, 1
28     sw $t3, ($t2)
29
30 .data
31 A: .word 5
32 B: .word
33
34
35
```

Line: 35 Column: 1 ☒ Show Line Numbers

Figura 14: Imagem do programa 14

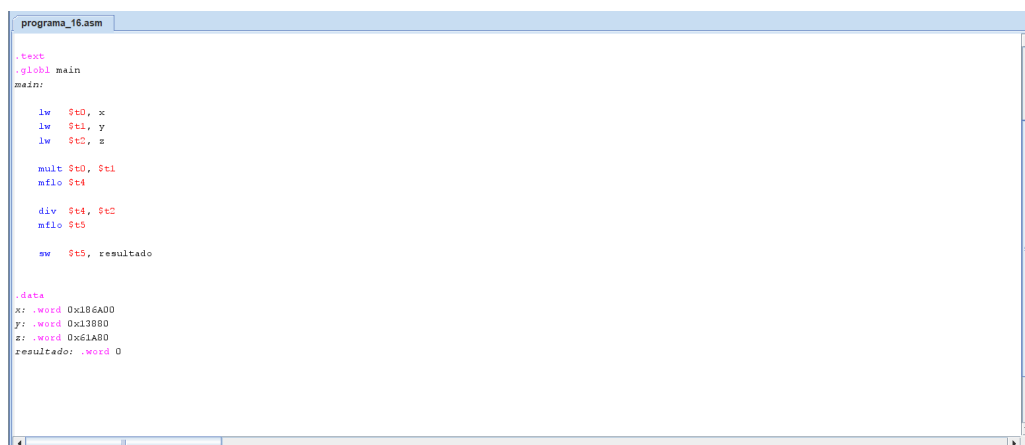
Programa - 15



```
programa_15
.text
.globl main
main:
    lw $t0, a
    lw $t1, i
    addi $s0, $0, 0
    addi $t2, $t0, 400
loop:
    beq $t0, $t2, fim
    sll $t3, $t1, 1
    addi $t3, $t3, 1
    sw $t3, ($t0)
    add $s0, $s0, $t3
    addi $t1, $t1, 1
    addi $t0, $t0, 4
    j loop
fim:
    sw $s0, ($t2)
.data
a: .word 0x10010008
i: .word 0
```

Figura 15: Imagem do programa 15

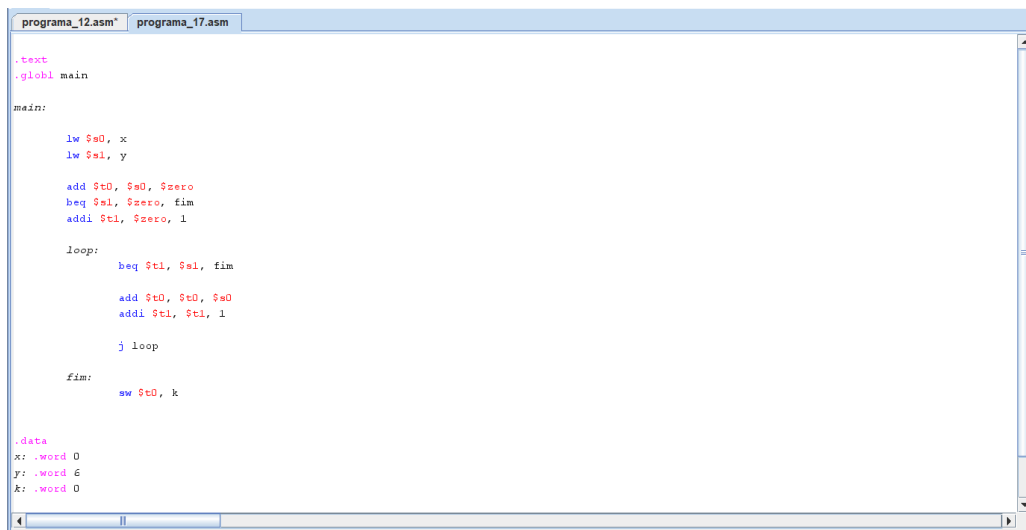
Programa - 16



```
programa_16.asm
.text
.globl main
main:
    lw $t0, x
    lw $t1, y
    lw $t2, z
    mult $t0, $t1
    mflo $t4
    div $t4, $t2
    mflo $t5
    sw $t5, resultado
.data
x: .word 0x186A00
y: .word 0x13880
z: .word 0x61A80
resultado: .word 0
```

Figura 16: Imagem do programa 16

Programa - 17



```
programa_12.asm  programa_17.asm

.text
.globl main

main:

    lw $s0, x
    lw $s1, y

    add $t0, $s0, $zero
    beq $s1, $zero, fim
    addi $t1, $zero, 1

loop:
    beq $t1, $s1, fim

    add $t0, $t0, $s0
    addi $t1, $t1, 1

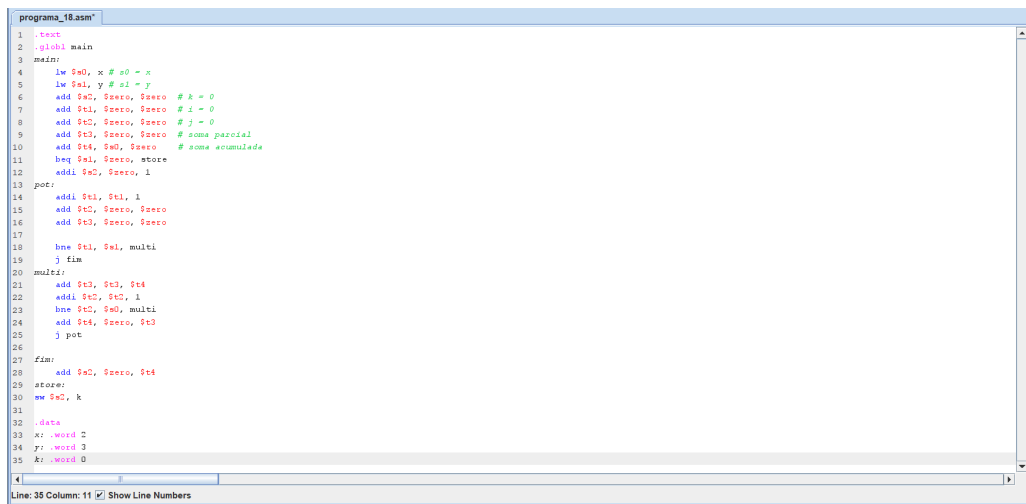
    j loop

fim:
    sw $t0, k

.data
x: .word 0
y: .word 6
k: .word 0
```

Figura 17: Imagem do programa 17

Programa - 18



```
programa_18.asm

1 .text
2 .globl main
3 main:
4     lw $s0, x # s0 = x
5     lw $s1, y # s1 = y
6     add $t0, $zero, $zero # k = 0
7     add $t1, $zero, $zero # i = 0
8     add $t2, $zero, $zero # j = 0
9     add $t3, $zero, $zero # some parcial
10    add $t4, $s0, $zero # some acumulada
11    beq $s1, $zero, store
12    addi $s0, $zero, 1
13 pot:
14    addi $t1, $t1, 1
15    add $t0, $zero, $zero
16    add $t3, $zero, $zero
17    bne $t1, $s1, multi
18    j fim
19 multi:
20 multi:
21    add $t3, $t3, $t4
22    addi $t0, $t0, 1
23    bne $t0, $s0, multi
24    add $t4, $zero, $t3
25    j pot
26
27 fim:
28    add $s2, $zero, $t4
29 store:
30    sw $s2, k
31
32 .data
33 x: .word 2
34 y: .word 3
35 k: .word 0
```

Figura 18: Imagem do programa 18

Programa - 19

```
programa_19.asm
11 .text
12 .globl main      # O valor presente em A é 2 em B é 4 para a execução deste programa -> Não foi possível exibir no print devido ao tamanho.
13 main:
14
15     lw $a0, A
16     lw $t1, B
17     add $t0, $zero, $zero
18     add $t2, $a0, $zero
19     count0:
20         beq $t2, $zero, done0
21         srl $t5, $t2, 1
22         addi $t0, $t0, 1
23         j count0
24     done0:
25         add $t1, $zero, $zero
26         add $t2, $t1, $zero
27     count1:
28         beq $t2, $zero, done1
29         srl $t2, $t2, 1
30         addi $t1, $t1, 1
31         j count1
32     done1:
33         sub $t4, $t0, 32
34         beq $t4, $zero, mul164
35         lwr $t4, $zero, check1
36     check1:
37         sub $t4, $t1, 32
38         beq $t4, $zero, mul164
39         lwr $t4, $zero, mul32
40     mul32:
41         mul $a2, $a0, $t1
42     mul164:
43         mult $a0, $a1
44         mflo $a2
45         mghi $a3
```

Figura 19: Imagem do programa 19

Programa - 20

```
programa_20.asm
6
7 .text
8 .globl main
9 main:
10     lw $a0, x
11     andi $t0, $a0, 1
12     beq $t0, $zero, even
13
14     odd:
15         mul $t1, $a0, $a0
16         mul $t2, $t1, $a0
17         mul $t3, $t2, $t1
18         sub $t3, $t3, $t2
19         addi $t3, $t3, 1
20         sw $t3, y
21
22     even:
23         mul $t1, $a0, $a0
24         mul $t2, $t1, $a0
25         mul $t3, $t2, $a0
26         add $t4, $t3, $t2
27         mul $t1, $t1, 1
28         sub $t4, $t4, $t1
29         sw $t4, y
30
31 .data
32 x: .word 4
33 y: .word 0
34
35
36
37
38
39
Line: 39 Column: 1 Show Line Numbers
```

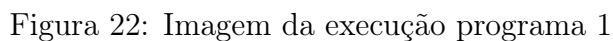
Figura 20: Imagem do programa 20

Programa - 21

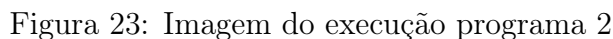
```
programa_21.asm
9
10 .text
11 .globl main
12 main:
13     lw $s0, x
14     sra $t0, $s0, 31
15     beq $t0, $zero, positive
16
17     non_positive:
18         mult $s0, $s0
19         mflo $t1
20         mult $t1, $s0
21         mflo $t2
22         mult $t2, $s0
23         mflo $t3
24         addi $t3, $t3, -1
25         sw $t3, y
26
27
28     positive:
29         mult $s0, $s0
30         mflo $t1
31         mult $t1, $s0
32         mflo $t2
33         addi $t2, $t2, 1
34         sw $t2, y
35
36 .data
37 x: .word 3
38 y: .word 0
39
40
41
42
43
```

Figura 21: Imagem do programa 21

Execução - 01



Execução - 02



Execução - 03

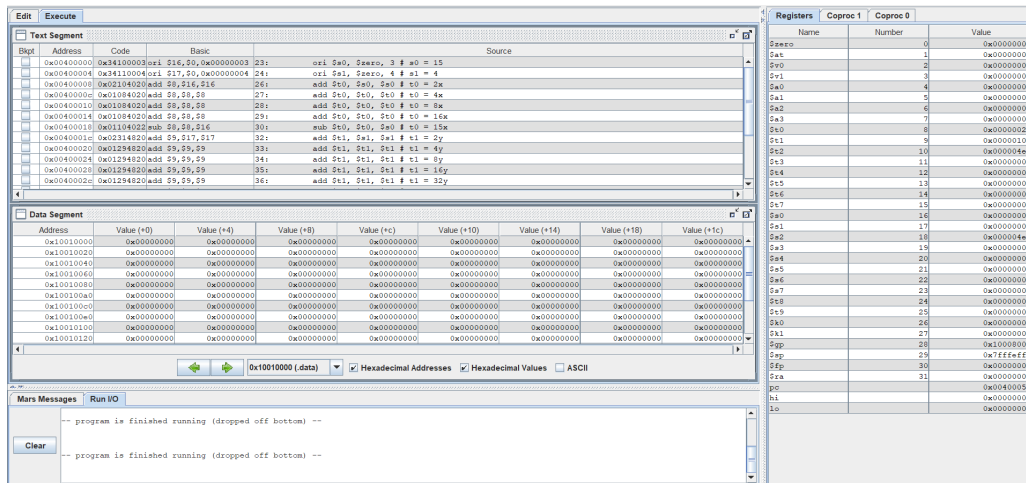


Figura 24: Imagem do execução programa 3

Execução - 04

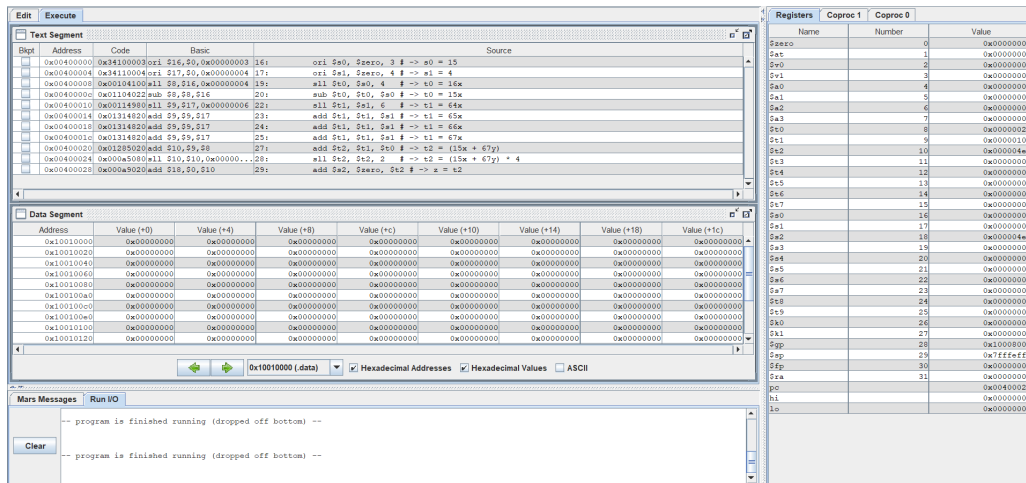


Figura 25: Imagem do execução programa 4

Execução - 05

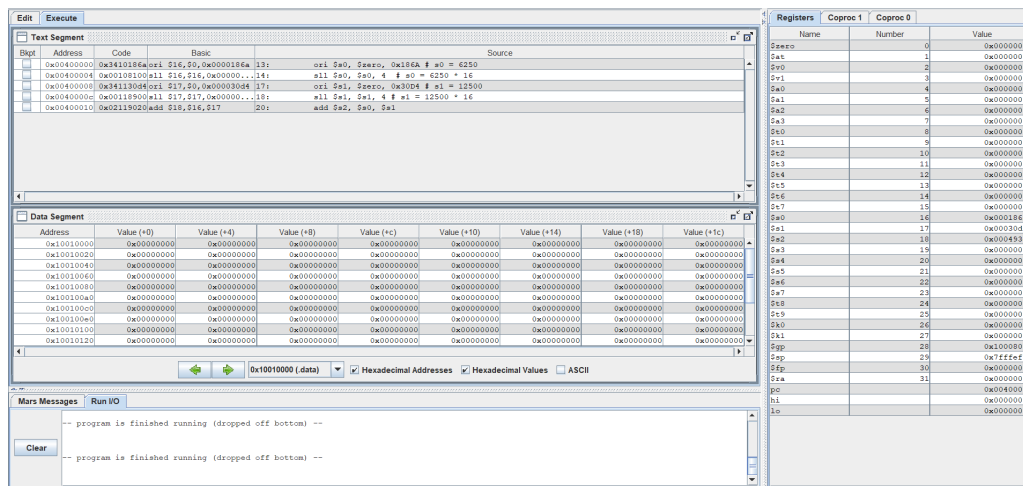


Figura 26: Imagem do execução programa 5

Execução - 06

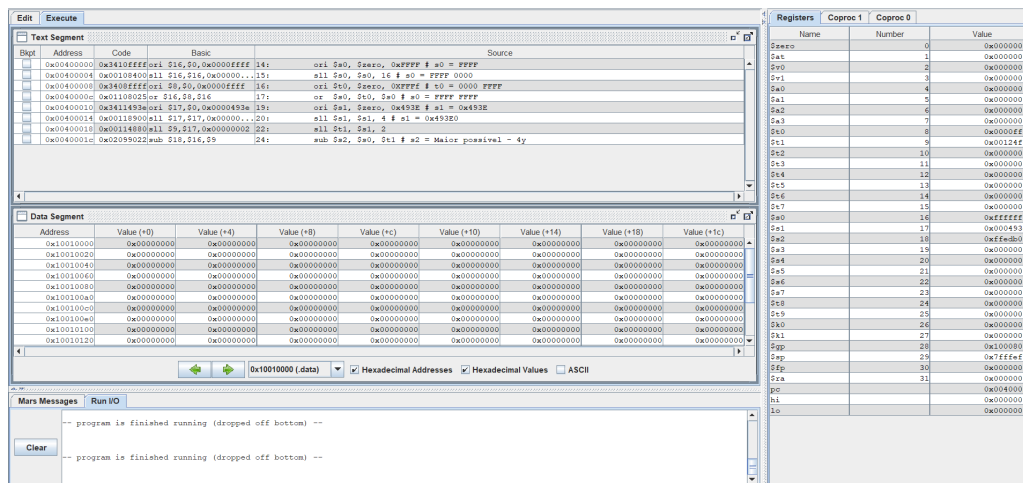


Figura 27: Imagem do execução programa 6

Execução - 07

Text Segment						Name		Number	Value
Blkt	Address	Code	Basic	Source					
0x00400000	0x34080001	ori 29,29,0x00000000	14:	ori 29, 29, 0x0 = 0000 0001			Zero	0	0x00000000
0x00400004	0x009470c1	ori 29,29,0x0000001f	15:	ori 29, 29, 31 # 29 = 1000 0000			Set	1	0x00000000
0x00400008	0x009470c2	ori 29,29,0x0000001f	16:	ori 29, 29, 31 # 29 = 1111 1111			Zero	2	0x00000000
Data Segment									
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)	
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Zero
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Set
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Zero
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Set
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Zero
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Set
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Zero
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Set
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Zero
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Set
Mars Messages						Run I/O			
-- program is finished running (dropped off bottom) --									
Clear									
-- program is finished running (dropped off bottom) --									

Figura 28: Imagem do execução programa 7

Execução - 08

Text Segment						Name		Number	Value
Blkt	Address	Code	Basic	Source					
0x00400000	0x34081234	ori 29,29,0x00001234	15:	ori 29, 29, 0x1234 # 10 = 0x1234			Zero	0	0x00000000
0x00400004	0x00944011	ori 29,29,0x00000019	16:	ori 29, 29, 16 # 10 = 0x12340000			Set	1	0x00000000
0x00400008	0x35085678	ori 29,29,0x00005678	17:	ori 29, 29, 5678 # 10 = 0x12345678			Zero	2	0x00000000
0x0040000c	0x00944012	ori 29,29,0x00000019	19:	ori 29, 29, 24 # 11 = 0x00000012			Set	3	0x00000000
0x00400010	0x00944013	ori 29,29,0x00000019	21:	ori 29, 29, 16 # 12 = 0x00001234			Zero	4	0x00000000
0x00400014	0x11400000	ori 29,29,0x00000019	22:	ori 29, 29, 0x11400000 # 13 = 0x00000034			Set	5	0x00000000
0x00400018	0x00944014	ori 29,29,0x00000019	24:	ori 29, 29, 8 # 13 = 0x00123456			Zero	6	0x00000000
0x0040001c	0x11400000	ori 29,29,0x00000019	25:	ori 29, 29, 0x11400000 # 13 = 0x00000056			Set	7	0x00000000
0x00400020	0x11400000	ori 29,29,0x00000019	27:	ori 29, 29, 0x11400000 # 13 = 0x00000056			Zero	8	0x00000000
Data Segment									
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)	
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Zero
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Set
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Zero
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Set
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Zero
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Set
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Zero
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Set
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Zero
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Set
Mars Messages						Run I/O			
-- program is finished running (dropped off bottom) --									
Clear									
-- program is finished running (dropped off bottom) --									

Figura 29: Imagem do execução programa 8

Execução - 09

Text Segment						Name		Number	Value
Blkt	Address	Code	Basic	Source					
0x00400000	0x34011001	lui 21,0x00001001	15:	lui 20, w1 # 20 = 15			Zero	0	0x00000000
0x00400004	0x34020001	lui 21,0x00000001	16:	lui 21, w2 # 21 = 25			Set	1	0x00000000
0x00400008	0x34020004	lui 21,0x00000004	17:	lui 21, w3 # 21 = 13			Zero	2	0x00000000
0x0040000c	0x34011001	lui 21,0x00001001	18:	lui 21, w4 # 21 = 17			Set	3	0x00000000
0x00400010	0x34011001	lui 21,0x00001001	20:	add 24, 24, 24 # 24 = (20 + 1)			Zero	4	0x00000000
0x00400014	0x34020001	lui 21,0x00000001	21:	add 24, 24, 24 # 24 = (20 + 1) + 2			Set	5	0x00000000
0x00400018	0x34020004	lui 21,0x00000004	22:	add 24, 24, 24 # 24 = (20 + 1) + 2 + 3			Zero	6	0x00000000
0x0040001c	0x34011001	lui 21,0x00001001	24:	sw 24, w24			Set	7	0x00000000
Data Segment									
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)	
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Zero
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Set
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Zero
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Set
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Zero
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Set
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Zero
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Set
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Zero
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	Set
Mars Messages						Run I/O			
-- program is finished running (dropped off bottom) --									
Clear									
-- program is finished running (dropped off bottom) --									

Figura 30: Imagem do execução programa 9

Execução - 10

Text Segment

0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

0x04000000

0x04000004

0x04000008

0x0400000c

0x04000010

0x04000014

0x04000018

0x0400001c

0x04000020

0x04000024

0x04000028

0x0400002c

0x04000030

0x04000034

0x04000038

0x0400003c

0x04000040

0x04000044

0x04000048

0x0400004c

0x04000050

0x04000054

0x04000058

0x0400005c

0x04000060

0x04000064

0x04000068

0x0400006c

0x04000070

0x04000074

0x04000078

0x0400007c

0x04000080

0x04000084

0x04000088

0x0400008c

0x04000090

0x04000094

0x04000098

0x0400009c

0x040000a0

0x040000a4

0x040000a8

0x040000ac

0x040000b0

0x040000b4

0x040000b8

0x040000bc

0x040000c0

0x040000c4

0x040000c8

0x040000cc

0x040000d0

0x040000d4

0x040000d8

0x040000dc

0x040000e0

0x040000e4

0x040000e8

0x040000ec

0x040000f0

0x040000f4

0x040000f8

0x040000fc

0x04000100

0x04000104

0x04000108

0x0400010c

0x04000110

0x04000114

0x04000118

0x0400011c

0x04000120

0x04000124

0x04000128

0x04000132

0x04000136

0x0400013a

0x0400013e

0x04000140

0x04000144

0x04000148

0x0400014c

0x04000150

0x04000154

0x04000158

0x0400015c

0x04000160

0x04000164

0x04000168

0x0400016c

0x04000170

0x04000174

0x04000178

0x0400017c

0x04000180

0x04000184

0x04000188

0x0400018c

0x04000190

0x04000194

0x04000198

0x0400019c

0x040001a0

0x040001a4

0x040001a8

0x040001ac

0x040001b0

0x040001b4

0x040001b8

0x040001bc

0x040001c0

0x040001c4

0x040001c8

0x040001cc

0x040001d0

0x040001d4

0x040001d8

0x040001dc

0x040001e0

0x040001e4

0x040001e8

0x040001ec

0x040001f0

0x040001f4

0x040001f8

0x040001fc

0x04000200

0x04000204

0x04000208

0x0400020c

0x04000210

0x04000214

0x04000218

0x0400021c

0x04000220

0x04000224

0x04000228

0x0400022c

0x04000230

0x04000234

0x04000238

0x0400023c

0x04000240

0x04000244

0x04000248

0x0400024c

0x04000250

0x04000254

0x04000258

0x0400025c

0x04000260

0x04000264

0x04000268

0x0400026c

0x04000270

0x04000274

0x04000278

0x0400027c

0x04000280

0x04000284

0x04000288

0x0400028c

0x04000290

0x04000294

0x04000298

0x0400029c

0x040002a0

0x040002a4

0x040002a8

0x040002ac

0x040002b0

0x040002b4

0x040002b8

0x040002bc

0x040002c0

0x040002c4

0x040002c8

0x040002cc

0x040002d0

0x040002d4

0x040002d8

0x040002dc

0x040002e0

0x040002e4

0x040002e8

0x040002ec

0x040002f0

0x040002f4

0x040002f8

0x040002fc

0x04000300

0x04000304

0x04000308

0x0400030c

0x04000310

0x04000314

0x04000318

0x0400031c

0x04000320

0x04000324

0x04000328

0x0400032c

0x04000330

0x04000334

0x04000338

0x0400033c

0x04000340

0x04000344

0x04000348

0x0400034c

0x04000350

0x04000354

0x04000358

0x0400035c

0x04000360

0x04000364

0x04000368

0x0400036c

0x04000370

0x04000374

0x04000378

0x0400037c

0x04000380

0x04000384

0x04000388

0x0400038c

0x04000390

0x04000394

0x04000398

0x0400039c

0x040003a0

0x040003a4

0x040003a8

0x040003ac

0x040003b0

0x040003b4

0x040003b8

0x040003bc

0x040003c0

0x040003c4

0x040003c8

0x040003cc

0x040003d0

0x040003d4

0x040003d8

0x040003dc

0x040003e0

0x040003e4

0x040003e8

0x040003ec

0x040003f0

0x040003f4

0x040003f8

0x040003fc

0x04000400

0x04000404

0x04000408

0x0400040c

0x04000410

0x04000414

0x04000418

0x0400041c

0x04000420

0x04000424

0x04000428

0x0400042c

0x04000430

0x04000434

0x04000438

0x0400043c

0x04000440

0x04000444

0x04000448

0x0400044c

0x04000450

0x04000454

0x04000458

0x0400045c

0x04000460

0x04000464

0x04000468

0x0400046c

0x04000470

0x04000474

0x04000478

0x0400047c

0x04000480

0x04000484

0x04000488

0x0400048c

0x04000490

0x04000494

0x04000498

0x0400049c

0x040004a0

0x040004a4

0x040004a8

0x040004ac

0x040004b0

0x040004b4

0x040004b8

0x040004bc

0x040004c0

0x040004c4

0x040004c8

0x040004cc

0x040004d0

0x040004d4

0x040004d8

0x040004dc

0x040004e0

0x040004e4

0x040004e8

0x040004ec

0x040004f0

0x040004f4

0x040004f8

0x040004fc

0x04000500

0x04000504

0x04000508

0x0400050c

0x04000510

0x04000514

0x04000518

0x0400051c

0x04000520

0x04000524

0x04000528

0x0400052c

0x04000530

0x04000534

0x04000538

0x0400053c

0x04000540

0x04000544

0x04000548

0x0400054c

0x04000550

0x04000554

0x04000558

0x0400055c

0x04000560

0x04000564

0x04000568

0x0400056c

0x04000570

0x04000574

0x04000578

0x0400057c

0x04000580

0x04000584

0x04000588

0x0400058c

0x04000590

0x04000594

0x04000598

0x0400059c

0x040005a0

0x040005a4

0x040005a8

0x040005ac

0x040005b0

0x040005b4

0x040005b8

0x040005bc

0x040005c0

0x040005c4

0x040005c8

0x040005cc

0x040005d0

0x040005d4

0x040005d8

0x040005dc

0x040005e0

0x040005e4

0x040005e8

0x040005ec

0x040005f0

0x040005f4

0x040005f8

0x040005fc

0x04000600

0x04000604

0x04000608

0x0400060c

0x04000610

0x04000614

0x04000618

0x0400061c

0x04000620

0x04000624

0x04000628

0x0400062c

0x04000630

0x04000634

0x04000638

0x0400063c

0x04000640

0x04000644

0x04000648

0x0400064c

0x04000650

0x04000654

0x04000658

0x0400065c

0x04000660

0x04000664

0x04000668

0x0400066c

0x04000670

0x04000674

0x04000678

0x0400067c

0x04000680

0x04000684

0x04000688

0x0400068c

0x04000690

0x04000694

0x04000698

0x0400069c

0x040006a0

0x040006a4

0x040006a8

0x040006ac

0x040006b0

0x040006b4

0x040006b8

0x040006bc

0x040006c0

0x040006c4

0x040006c8

0x040006cc

0x040006d0

0x040006d4

0x040006d8

0x040006dc

0x040006e0

0x040006e4

0x040006e8

0x040006ec

0x040006f0

0x040006f4

0x040006f8

0x040006fc

0x04000700

0x04000704

0x04000708

0x0400070c

0x04000710

0x04000714

0x04000718

0x0400071c

0x04000720

0x04000724

0x04000728

0x0400072c

0x04000730

0x04000734

0x04000738

0x0400073c

0x04000740

0x04000744

0x04000748

0x0400074c

0x04000750

0x04000754

0x04000758

0x0400075c

0x04000760

0x04000764

0x04000768

0x0400076c

0x04000770

0x04000774

0x04000778

0x0400077c

0x04000780

0x04000784

0x04000788

0x0400078c

0x04000790

0x04000794

0x04000798

0x0400079c

0x040007a0

0x040007a4

0x040007a8

0x040007ac

0x040007b0

0x040007b4

0x040007b8

0x040007bc

0x040007c0

0x040007c4

0x040007c8

0x040007cc

0x040007d0

0x040007d4

0x040007d8

0x040007dc

0x040007e0

0x040007e4

0x040007e8

0x040007ec

0x040007f0

0x040007f4

0x040007f8

0x040007fc

0x04000800

0x04000804

0x04000808

0x0400080c

0x04000810

0x04000814

0x04000818

0x0400081c

0x04000820

0x04000824

0x04000828

0x0400082c

0x04000830

0x04000834

0x04000838

0x0400083c

0x04000840

0x04000844

0x04000848

0x0400084c

0x04000850

0x04000854

0x04000858

0x0400085c

0x04000860

0x04000864

0x04000868

0x0400086c

0x04000870

0x04000874

0x04000878

0x0400087c

0x04000880

0x04000884

0x04000888

0x0400088c

0x04000890

0x04000894

0x04000898

0x0400089c

0x040008a0

0x040008a4

0x040008a8

0x040008ac

0x040008b0

0x040008b4

0x040008b8

0x040008bc

0x040008c0

0x040008c4

0x040008c8

0x040008cc

0x040008d0

0x040008d4

0x040008d8

0x040008dc

0x040008e0

0x040008e4

0x040008e8

0x040008ec

0x040008f0

0x040008f4

0x040008f8

0x040008fc

0x04000900

0x04000904

0x04000908

0x0400090c

0x04000910

0x04000914

0x04000918

0x0400091c

0x04000920

0x04000924

0x04000928

0x0400092c

0x04000930

0x04000934

0x04000938

0x0400093c

0x04000940

0x04000944

0x04000948

0x0400094c

0x04000950

0x04000954

0x04000958

0x0400095c

0x04000960

0x04000964

0x04000968

0x0400096c

0x04000970

0x04000974

0x04000978

0x0400097c

0x04000980

0x04000984

0x04000988

0x0400098c

0x04000990

0x04000994

0x04000998

0x0400099c

0x040009a0

0x040009a4

0x040009a8

0x040009ac

0x040009b0

0x040009b4

0x040009b8

0x040009bc

0x040009c0

0x040009c4

0x040009c8

0x040009cc

0x040009d0

0x040009d4

0x040009d8

0x040009dc

0x040009e0

0x040009e4

0x040009e8

0x040009ec

0x040009f0

0x040009f4

0x040009f8

0x040009fc

0x04000a00

0x04000a04

0x04000a08

0x04000a0c

0x04000a10

0x04000a14

0x04000a18

0x04000a1c

0x04000a20

0x04000a24

0x04000a28

0x04000a2c

0x04000a30

0x04000a34

0x04000a38

0x04000a3c

0x04000a40

0x04000a44

0x04000a48

0x04000a4c

0x04000a50

0x04000a54

0x04000a58

0x04000a5c

0x04000a60

0x04000a64

0x04000a68

0x04000a6c

0x04000a70

0x04000a74

0x04000a78

0x04000a7c

0x04000a80

0x04000a84

0x04000a88

0x04000a8c

0x04000a90

0x04000a94

0x04000a98

0x04000a9c

0x04000aa0

0x04000aa4

0x04000aa8

0x04000aac

0x04000ab0

0x04000ab4

0x04000ab8

0x04000abc

0x04000ac0

0x04000ac4

0x04000ac8

0x04000acc

0x04000ad0

0x04000ad4

0x04000ad8

0x04000adc

0x04000ae0

0x04000ae4

0x04000ae8

0x04000aec

0x04000af0

0x04000af4

0x04000af8

0x04000afc

0x04000b00

0x04000b04

0x04000b08

0x04000b0c

0x04000b10

0x04000b14

0x04000b18

0x04000b1c

0x04000b20

0x04000b24

0x04000b28

0x04000b2c

0x04000b30

0x04000b34

0x04000b38

0x04000b3c

0x04000b40

0x04000b44

0x04000b48

0x04000b4c

0x04000b50

0x04000b54

0x04000b58

0x04000b5c

0x04000b60

0x04000b64

0x04000b68

0x04000b6c

0x04000b70

0x04000b74

0x04000b78

0x04000b7c

0x04000b80

0x04000b84

0x04000b88

0x04000b8c

0x04000b90

0x04000b94

0x04000b98

0x04000b9c

0x04000ba0

0x04000ba4

0x04000ba8

0x04000bac

0x04000bb0

0x04000bb4

0x04000bb8

0x04000bbc

0x04000bc0

0x04000bc4

0x04000bc8

0x04000bcc

0x04000bd0

0x04000bd4

0x04000bd8

0x04000bdc

0x04000be0

0x04000be4

0x04000be8

0x04000bec

0x04000bf0

0x04000bf4

0x04000bf8

0x04000bfc

0x04000c00

0x04000c04

0x04000c08

0x04000c0c

0x04000c10

0x04000c14

0x04000c18

0x04000c1c

0x04000c20

0x04000c24

0x04000c28

0x04000c2c

0x04000c30

0x04000c34

0x04000c38

0x04000c3c

0x04000c40

0x04000c44

0x04000c48

0x04000c4c

0x04000c50

0x04000c54

0x04000c58

0x04000c5c

0x04000c60

0x04000c64

0x04000c68

0x04000c6c

0x04000c70

0x04000c74

0x04000c78

0x04000c7c

0x04000c80

0x04000c84

0x04000c88

0x04000c8c

0x04000c90

0x04000c94

0x04000c98

0x04000c9c

0x04000ca0

0x04000ca4

0x04000ca8

0x04000cac

0x04000cb0

0x04000cb4

0x04000cb8

0x04000cbc

0x04000cc0

0x04000cc4

0x04000cc8

0x04000ccc

0x04000cd0

0x040

Figura 31: Imagem do execução programa 10

Execução - 11

Text Segment

0x040000000x0e011001lui\$1,0x0000100115:lw\$20,x#t0=x

0x040000040xc2200000lw\$9,0x00000000(\$t1)

0x040000080xc2011001lui\$1,0x0000100116:lw\$21,x#t1=x

0x0400000c0xc2200004lw\$9,0x00000004(\$t1)

0x040000100x00059502sub\$10,\$9,\$918:sub\$2,\$20,\$21#t2=x-x

0x040000140xc0145021addi\$11,\$t0,0x0000493e20:ori\$23,\$zero,0x493e#t3=0x493e

0x040000180x00059500addi\$11,\$t1,0x0000...21:all\$23,\$2,\$2#t3=0x493e0

0x0400001c0xc0145020addi\$11,\$t0,\$t123:add\$23,\$2,\$2#t3=(x-x)+300000

0x040000200xc2011001lui\$1,0x0000100124:sw\$23,y

0x040000240xc0200000sw\$11,0x00000000(\$t1)

NameNumberValue

\$zero00x00000000

\$at10x10010000

\$v020x00000000

\$v130x00000000

\$a040x00000000

\$a150x00000000

\$a260x00000000

\$a370x00000000

\$t080x00000000

\$t190x00000000

\$t2100xffff7960

\$t3110x20000019

\$t4120x00000000

\$t5130x00000000

\$t6140x00000000

\$t7150x00000000

\$s0160x00000000

\$s1170x00000000

\$s2180x00000000

\$s3190x00000000

\$s4200x00000000

\$s5210x00000000

\$s6220x00000000

\$s7230x00000000

\$s8240x00000000

\$t9250x00000000

\$k0260x00000000

\$k1270x00000000

\$gp280x10009000

\$fp290x7ffffefc

\$dp300x00000000

\$ra310x00000000

\$PC0x00400000

\$hi0x00000000

\$lo0x00000000

AddressValue(+0)Value(+4)Value(+8)Value(+c)Value(+10)Value(+14)Value(+18)Value(+1c)

0x100100000x0000186a0x000030400x000030400x000000000x000000000x000000000x000000000x00000000

0x100100200x000000000x000000000x000000000x000000000x000000000x000000000x000000000x00000000

0x100100400x000000000x000000000x000000000x000000000x000000000x000000000x000000000x00000000

0x100100600x000000000x000000000x000000000x000000000x000000000x000000000x000000000x00000000

0x100100800x000000000x000000000x000000000x000000000x000000000x000000000x000000000x00000000

0x100100a00x000000000x000000000x000000000x000000000x000000000x000000000x000000000x00000000

0x100100c00x000000000x000000000x000000000x000000000x000000000x000000000x000000000x00000000

0x100100e00x000000000x000000000x000000000x000000000x000000000x000000000x000000000x00000000

0x100101000x000000000x000000000x000000000x000000000x000000000x000000000x000000000x00000000

0x100101200x000000000x000000000x000000000x000000000x000000000x000000000x000000000x00000000

0x100100000(data)

Hexadecimal Addresses

Hexadecimal Values

ASCII

Mars Messages

Run I/O

-- program is finished running (dropped off bottom) --

Clear

-- program is finished running (dropped off bottom) --

Figura 32: Imagem do execução programa 11

Execução - 12

Text Segment

Offset	Address	Code	Basic		Source
0	0x04000000	0x0e011001	lui \$1,0x00001001	13:	lw \$t0, ptr3
1	0x04000004	0xc2200000	lw \$9,0x00000000(\$t1)		
2	0x04000008	0xc2011001	lui \$1,0x00001001	14:	lw \$t1, (\$t0)
3	0x0400000c	0xc2200000	lw \$10,0x00000000(\$t0)	15:	lw \$t2, (\$t1)
4	0x04000010	0xc2011001	lui \$1,0x00001001	16:	lw \$a0, (\$t2)
5	0x04000014	0xc0101040	addi \$11,\$t0,0x0000...18:		addi \$a0, \$a0, 1
6	0x04000018	0xc2011001	lui \$1,0x00001001	20:	sw \$a0, x
7	0x0400001c	0xc0200000	sw \$16,0x00000000(\$t1)		

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x10010000
\$t1	9	0x10010004
\$t2	10	0x10010000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$s8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10009000
\$fp	29	0x7ffffefc
\$dp	30	0x00000000
\$ra	31	0x00000000
\$PC		0x00400000
\$hi		0x00000000
\$lo		0x00000000

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x10010000	0x10010004	0x10010008	0x00000000	0x00000000	0x00000000	0x00000000
0x10010004	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010008	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001000c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010010	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010014	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010018	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001001c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010024	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010028	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001002c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (data)

☒ Hexadecimal Addresses☒ Hexadecimal Values☐ ASCII

Mars MessagesRun I/O

-- program is finished running (dropped off bottom) --

Clear

-- program is finished running (dropped off bottom) --

Figura 33: Imagem do execução programa 12

Execução - 13

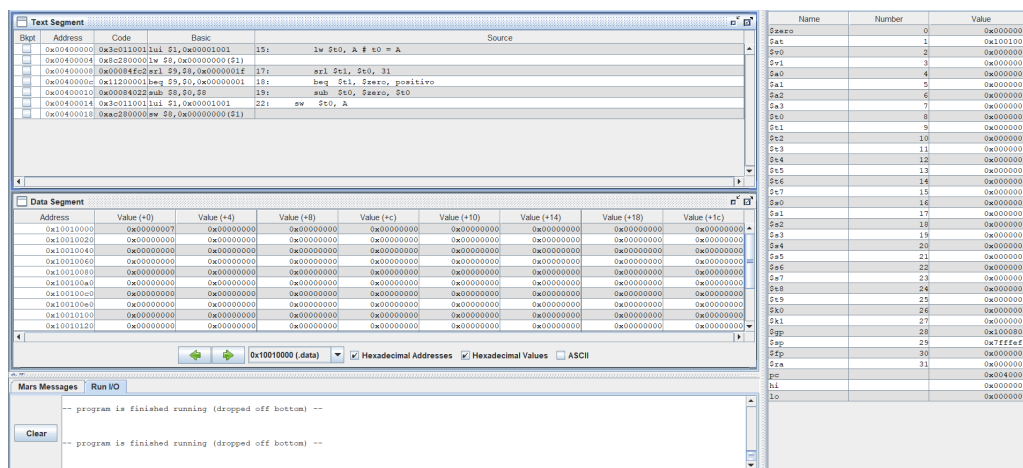


Figura 34: Imagem do execução programa 13

Execução - 14

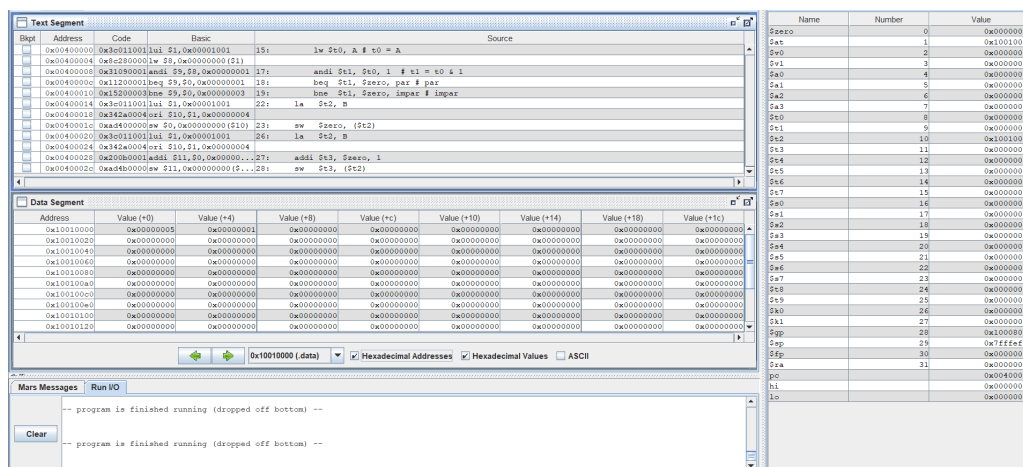


Figura 35: Imagem do execução programa 14

Execução - 15

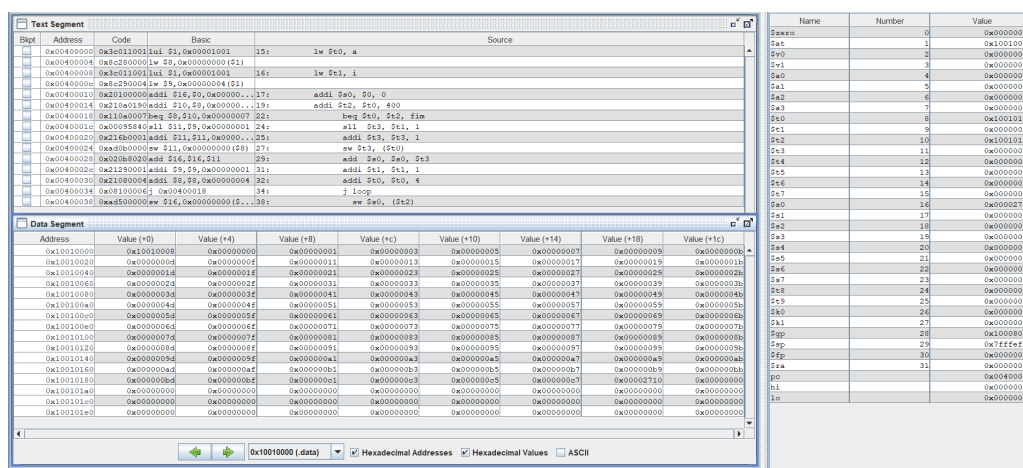


Figura 36: Imagem do execução programa 15

Execução - 16

Text Segment					Name	Number	Value
Offset	Address	Code	Basic	Source			
0x00400000	0x0011001	lui	\$1, 0x000001001	14: lw \$t0, x	\$zero	0	0x00000000
0x00400004	0xc300000	lw	\$t0, 0x00000000(\$1)	15: lw \$t1, y	\$at	1	0x001001000
0x00400008	0x0011001	lui	\$1, 0x000001001	16: lw \$t2, z	\$v0	2	0x00000000
0x0040000c	0xc300004	lw	\$t0, 0x00000004(\$1)	17: lw \$t3, x	\$v1	3	0x00000000
0x00400010	0x0011001	lui	\$1, 0x000001001	18: mult \$t0, \$t1	\$a0	4	0x00000000
0x00400014	0xc300008	lw	\$t0, 0x00000008(\$1)	19: mflo \$t4	\$a1	5	0x00000000
0x00400018	0x0010001	mult	\$t0, \$t1	20: mflo \$t5	\$a2	6	0x00000000
0x0040001c	0x0000001	mflo	\$t4	21: mflo \$t5	\$a3	7	0x00000000
0x00400020	0x0010001	mflo	\$t4	22: mflo \$t5	\$a4	8	0x00000000
0x00400024	0x0000001	mflo	\$t4	23: mflo \$t5	\$a5	9	0x00000000
0x00400028	0x0011001	lui	\$1, 0x000001001	24: sw \$t5, resultado	\$a6	10	0x00000000
0x0040002c	0xc30000c	sw	\$t5, 0x0000000c(\$1)		\$a7	11	0x00000000
					\$a8	12	0x00000000
					\$a9	13	0x00000000
					\$t0	14	0x00000000
					\$t1	15	0x00000000
					\$t2	16	0x00000000
					\$t3	17	0x00000000
					\$t4	18	0x00000000
					\$t5	19	0x00000000
					\$t6	20	0x00000000
					\$t7	21	0x00000000
					\$t8	22	0x00000000
					\$t9	23	0x00000000
					\$t0	24	0x00000000
					\$t1	25	0x00000000
					\$t2	26	0x00000000
					\$t3	27	0x00000000
					\$t4	28	0x00000000
					\$t5	29	0x00000000
					\$t6	30	0x00000000
					\$t7	31	0x00000000
					\$t8	32	0x00000000
					\$t9	33	0x00000000
					\$t0	34	0x00000000
					\$t1	35	0x00000000
					\$t2	36	0x00000000
					\$t3	37	0x00000000
					\$t4	38	0x00000000
					\$t5	39	0x00000000
					\$t6	40	0x00000000
					\$t7	41	0x00000000
					\$t8	42	0x00000000
					\$t9	43	0x00000000
					\$t0	44	0x00000000
					\$t1	45	0x00000000
					\$t2	46	0x00000000
					\$t3	47	0x00000000
					\$t4	48	0x00000000
					\$t5	49	0x00000000
					\$t6	50	0x00000000
					\$t7	51	0x00000000
					\$t8	52	0x00000000
					\$t9	53	0x00000000
					\$t0	54	0x00000000
					\$t1	55	0x00000000
					\$t2	56	0x00000000
					\$t3	57	0x00000000
					\$t4	58	0x00000000
					\$t5	59	0x00000000
					\$t6	60	0x00000000
					\$t7	61	0x00000000
					\$t8	62	0x00000000
					\$t9	63	0x00000000
					\$t0	64	0x00000000
					\$t1	65	0x00000000
					\$t2	66	0x00000000
					\$t3	67	0x00000000
					\$t4	68	0x00000000
					\$t5	69	0x00000000
					\$t6	70	0x00000000
					\$t7	71	0x00000000
					\$t8	72	0x00000000
					\$t9	73	0x00000000
					\$t0	74	0x00000000
					\$t1	75	0x00000000
					\$t2	76	0x00000000
					\$t3	77	0x00000000
					\$t4	78	0x00000000
					\$t5	79	0x00000000
					\$t6	80	0x00000000
					\$t7	81	0x00000000
					\$t8	82	0x00000000
					\$t9	83	0x00000000
					\$t0	84	0x00000000
					\$t1	85	0x00000000
					\$t2	86	0x00000000
					\$t3	87	0x00000000
					\$t4	88	0x00000000
					\$t5	89	0x00000000
					\$t6	90	0x00000000
					\$t7	91	0x00000000
					\$t8	92	0x00000000
					\$t9	93	0x00000000
					\$t0	94	0x00000000
					\$t1	95	0x00000000
					\$t2	96	0x00000000
					\$t3	97	0x00000000
					\$t4	98	0x00000000
					\$t5	99	0x00000000
					\$t6	100	0x00000000
					\$t7	101	0x00000000
					\$t8	102	0x00000000
					\$t9	103	0x00000000
					\$t0	104	0x00000000
					\$t1	105	0x00000000
					\$t2	106	0x00000000
					\$t3	107	0x00000000
					\$t4	108	0x00000000
					\$t5	109	0x00000000
					\$t6	110	0x00000000
					\$t7	111	0x00000000
					\$t8	112	0x00000000
					\$t9	113	0x00000000
					\$t0	114	0x00000000
					\$t1	115	0x00000000
					\$t2	116	0x00000000
					\$t3	117	0x00000000
					\$t4	118	0x00000000
					\$t5	119	0x00000000
					\$t6	120	0x00000000
					\$t7	121	0x00000000
					\$t8	122	0x00000000
					\$t9	123	0x00000000
					\$t0	124	0x00000000
					\$t1	125	0x00000000
					\$t2	126	0x00000000
					\$t3	127	0x00000000
					\$t4	128	0x00000000
					\$t5	129	0x00000000
					\$t6	130	0x00000000
					\$t7	131	0x00000000
					\$t8	132	0x00000000
					\$t9	133	0x00000000
					\$t0	134	0x00000000
					\$t1	135	0x00000000
					\$t2	136	0x00000000
					\$t3	137	0x00000000
					\$t4	138	0x00000000
					\$t5	139	0x00000000
					\$t6	140	0x00000000
					\$t7	141	0x00000000
					\$t8	142	0x00000000
					\$t9	143	0x00000000
					\$t0	144	0x00000000
					\$t1	145	0x00000000
					\$t2	146	0x00000000
					\$t3	147	0x00000000
					\$t4	148	0x00000000
					\$t5	149	0x00000000
					\$t6	150	0x00000000
					\$t7	151	0x00000000
					\$t8	152	0x00000000
					\$t9	153	0x00000000
					\$t0	154	0x00000000
					\$t1	155	0x00000000
					\$t2	156	0x00000000
					\$t3	157	0x00000000
					\$t4	158	0x00000000
					\$t5	159	0x00000000
					\$t6	160	0x00000000
					\$t7	161	0x00000000
					\$t8	162	0x00000000
					\$t9	163	0x00000000
					\$t0	164	0x00000000
					\$t1	165	0x00000000
					\$t2	166	0x00000000
					\$t3	167	0x00000000
					\$t4	168	0x00000000
					\$t5	169	0x00000000
					\$t6	170	0x00000000
					\$t7	171	0x00000000
					\$t8	172	0x00000000
					\$t9	173	0x00000000
					\$t0	174	0x00000000
					\$t1	175	0x00000000
					\$t2	176	0x00000000
					\$t3	177	0x00000000
					\$t4	178	0x00000000
					\$t5	179	0x00000000
					\$t6	180	0x00000000
					\$t7	181	0x00000000
					\$t8	182	0x00000000
					\$t9	183	0x00000000
					\$t0	184	0x00000000
					\$t1	185	0x00000000
					\$t2	186	0x00000000
					\$t3	187	0x00000000
					\$t4	188	0x00000000
					\$t5	189	0x00000000
					\$t6	190	0x00000000
					\$t7	191	0x00000000
					\$t8	192	0x00000000
					\$t9	193	0x00000000
					\$t0	194	0x00000000
					\$t1	195	0x00000000
					\$t2	196	0x00000000
					\$t3	197	0x00000000
					\$t4	198	0x00000000
					\$t5	199	0x00000000
					\$t6	200	0x00000000
					\$t7	201	0x00000000
					\$t8	202	0x00000000
					\$t9	203	0x00000000
					\$t0	204	0x00000000
					\$t1	205	0x00000000
					\$t2	206	0x00000000
					\$t3	207	0x00000000
					\$t4	208	0x00000000
					\$t5	209	0x00000000
					\$t6	210	0x00000000
					\$t7	211	0x

Execução - 19

Text Segment										Registers		
Offset	Address	Code	Basic	Source						Name	Number	Value
0	0x00400000	0x3e011001	lui \$1,0x00000101	15: lw \$a0, A						\$zero	0	0x00000000
1	0x00400004	0x20000011	lw \$16,0x00000000(\$1)							\$a1	1	0x00000000
2	0x00400008	0x3e011001	lui \$1,0x00000101	16: lw \$a1, B						\$v0	2	0x00000000
3	0x0040000c	0x3e010004	lui \$17,0x00000004(\$1)							\$v1	3	0x00000000
4	0x00400010	0x00000020	add \$9,\$0,\$0							\$a0	4	0x00000000
5	0x00400014	0x00000020	add \$10,\$0,\$0							\$a1	5	0x00000000
6	0x00400018	0x14000003	beq \$10,\$0,0x00000003	20: beq \$a1, \$zero, done0						\$a2	6	0x00000000
7	0x0040001c	0x00000042	ori \$10,\$10,0x00000042	21: add \$a2, \$a0, \$zero						\$a3	7	0x00000000
8	0x00400020	0x11000001	addi \$9,\$9,0x00000001	22: add \$a3, \$a0, \$1						\$a4	8	0x00000002
9	0x00400024	0x00100006	0x00400018	23: j count0						\$a5	9	0x00000003
10	0x00400028	0x00000020	add \$9,\$0,\$0	25: add \$a5, \$zero, \$zero						\$a6	10	0x00000000
11	0x0040002c	0x00000020	add \$10,\$10,\$0	26: add \$a6, \$a5, \$zero						\$a7	11	0x00000000
12	0x00400030	0x00000000								\$a8	12	0xffffffff
13	0x00400034	0x00000000								\$a9	13	0x00000000
14	0x00400038	0x00000000								\$a10	14	0x00000000
15	0x0040003c	0x00000000								\$a11	15	0x00000000
16	0x00400040	0x00000000								\$a12	16	0x00000000
17	0x00400044	0x00000000								\$a13	17	0x00000004
18	0x00400048	0x00000000								\$a14	18	0x00000008
19	0x0040004c	0x00000000								\$a15	19	0x00000000
20	0x00400050	0x00000000								\$a16	20	0x00000000
21	0x00400054	0x00000000								\$a17	21	0x00000000
22	0x00400058	0x00000000								\$a18	22	0x00000000
23	0x0040005c	0x00000000								\$a19	23	0x00000000
24	0x00400060	0x00000000								\$a20	24	0x00000000
25	0x00400064	0x00000000								\$a21	25	0x00000000
26	0x00400068	0x00000000								\$a22	26	0x00000000
27	0x0040006c	0x00000000								\$a23	27	0x00000000
28	0x00400070	0x00000000								\$a24	28	0x00000000
29	0x00400074	0x00000000								\$a25	29	0x00000000
30	0x00400078	0x00000000								\$a26	30	0x00000000
31	0x0040007c	0x00000000								\$a27	31	0x00000000
32	0x00400080	0x00000000								\$a28	32	0x00000000
33	0x00400084	0x00000000								\$a29	33	0x00000000
34	0x00400088	0x00000000								\$a30	34	0x00000000
35	0x0040008c	0x00000000								\$a31	35	0x00000000
36	0x00400090	0x00000000								\$a32	36	0x00000000
37	0x00400094	0x00000000								\$a33	37	0x00000000
38	0x00400098	0x00000000								\$a34	38	0x00000000
39	0x0040009c	0x00000000								\$a35	39	0x00000000
40	0x004000a0	0x00000000								\$a36	40	0x00000000
41	0x004000a4	0x00000000								\$a37	41	0x00000000
42	0x004000a8	0x00000000								\$a38	42	0x00000000
43	0x004000ac	0x00000000								\$a39	43	0x00000000
44	0x004000b0	0x00000000								\$a40	44	0x00000000
45	0x004000b4	0x00000000								\$a41	45	0x00000000
46	0x004000b8	0x00000000								\$a42	46	0x00000000
47	0x004000bc	0x00000000								\$a43	47	0x00000000
48	0x004000c0	0x00000000								\$a44	48	0x00000000
49	0x004000c4	0x00000000								\$a45	49	0x00000000
50	0x004000c8	0x00000000								\$a46	50	0x00000000
51	0x004000cc	0x00000000								\$a47	51	0x00000000
52	0x004000d0	0x00000000								\$a48	52	0x00000000
53	0x004000d4	0x00000000								\$a49	53	0x00000000
54	0x004000d8	0x00000000								\$a50	54	0x00000000
55	0x004000dc	0x00000000								\$a51	55	0x00000000
56	0x004000e0	0x00000000								\$a52	56	0x00000000
57	0x004000e4	0x00000000								\$a53	57	0x00000000
58	0x004000e8	0x00000000								\$a54	58	0x00000000
59	0x004000ec	0x00000000								\$a55	59	0x00000000
60	0x004000f0	0x00000000								\$a56	60	0x00000000
61	0x004000f4	0x00000000								\$a57	61	0x00000000
62	0x004000f8	0x00000000								\$a58	62	0x00000000
63	0x004000fc	0x00000000								\$a59	63	0x00000000
64	0x00400100	0x00000000								\$a60	64	0x00000000
65	0x00400104	0x00000000								\$a61	65	0x00000000
66	0x00400108	0x00000000								\$a62	66	0x00000000
67	0x0040010c	0x00000000								\$a63	67	0x00000000
68	0x00400110	0x00000000								\$a64	68	0x00000000
69	0x00400114	0x00000000								\$a65	69	0x00000000
70	0x00400118	0x00000000								\$a66	70	0x00000000
71	0x0040011c	0x00000000								\$a67	71	0x00000000
72	0x00400120	0x00000000								\$a68	72	0x00000000
73	0x00400124	0x00000000								\$a69	73	0x00000000
74	0x00400128	0x00000000								\$a70	74	0x00000000
75	0x0040012c	0x00000000								\$a71	75	0x00000000
76	0x00400130	0x00000000								\$a72	76	0x00000000
77	0x00400134	0x00000000								\$a73	77	0x00000000
78	0x00400138	0x00000000								\$a74	78	0x00000000
79	0x0040013c	0x00000000								\$a75	79	0x00000000
80	0x00400140	0x00000000								\$a76	80	0x00000000
81	0x00400144	0x00000000								\$a77	81	0x00000000
82	0x00400148	0x00000000								\$a78	82	0x00000000
83	0x0040014c	0x00000000								\$a79	83	0x00000000
84	0x00400150	0x00000000								\$a80	84	0x00000000
85	0x00400154	0x00000000								\$a81	85	0x00000000
86	0x00400158	0x00000000								\$a82	86	0x00000000
87	0x0040015c	0x00000000								\$a83	87	0x00000000
88	0x00400160	0x00000000								\$a84	88	0x00000000
89	0x00400164	0x00000000								\$a85	89	0x00000000
90	0x00400168	0x00000000								\$a86	90	0x00000000
91	0x0040016c	0x00000000								\$a87	91	0x00000000
92	0x00400170	0x00000000								\$a88	92	0x00000000
93	0x00400174	0x00000000								\$a89	93	0x00000000
94	0x00400178	0x00000000								\$a90	94	0x00000000
95	0x0040017c	0x00000000								\$a91	95	0x00000000
96	0x00400180	0x00000000								\$a92	96	0x00000000
97	0x00400184	0x00000000								\$a93	97	0x00000000
98	0x00400188	0x00000000								\$a94	98	0x00000000
99	0x0040018c	0x00000000								\$a95	99	0x00000000
100	0x00400190	0x00000000								\$a96	100	0x00000000
101	0x00400194	0x00000000								\$a97	101	0x00000000
102	0x00400198	0x00000000								\$a98	102	0x00000000
103	0x0040019c	0x00000000								\$a99	103	0x00000000
104	0x004001a0	0x00000000								\$a100	104	0x00000000
105	0x004001a4	0x00000000								\$a101	105	0x00000000
106	0x004001a8	0x00000000								\$a102	106	0x00000000
107	0x004001ac	0x00000000								\$a103	107	0x00000000
108	0x004001b0	0x00000000								\$a104	108	0x00000000
109	0x004001b4	0x00000000								\$a105	109	0x00000000
110	0x004001b8	0x00000000								\$a106	110	0x00000000
111	0x004001bc	0x00000000								\$a107	111	0x00000000
112	0x004001c0	0x00000000								\$a108	112	0x00000000
113	0x004001c4	0x00000000								\$a109	113	0x00000000
114	0x004001c8	0x00000000								\$a110	114	0x00000000
115	0x004001cc	0x00000000								\$a111	115	0x00000000
116	0x004001d0	0x00000000								\$a112	116	0x00000000
117	0x004001d4	0x00000000								\$a113	117	0x00000000
118	0x004001d8											

Perguntas a serem respondidas

Se tivermos 2 inteiros, cada um com 32 bits, quantos bits podemos esperar para o produto?

A resposta correta é 64 bits

Quais os registradores que armazenam os resultados na multiplicação?

Os registradores que armazenam os resultados de uma multiplicação são o hi e o lo, que guardam os bits mais significativos e menos significativos, respectivamente.

Qual a operação usada para multiplicar inteiros em comp. de dois?

Em MIPS, a operação realizada para multiplicar inteiros em complemento de dois é a operação mult.

Qual instrução move os bits menos significativos da multiplicação para o reg. 8?

Para mover o bit menos significativo para o registrador 8, é necessário utilizar mflo.

Se tivermos dois inteiros, cada um com 32 bits, quantos bits deveremos estar preparados para receber no quociente?

Para uma divisão inteira de 32 bits, o quociente pode armazenar até os 32 bits.

Após a instrução div, qual registrador possui o quociente?

O registrador que armazenará o quociente é o registrador lo.

Qual a inst. Usada para dividir dois inteiros em comp. de dois?

Tal instrução diz a respeito sobre a instrução div.

Faça um arithmetic shift right de dois no seguinte padrão de bits: 1001 1011

O resultado de uma instrução sra aplicada ao número 1001 1011, deslocando-se 2 bits, resulta em: 1110 0110

Qual o efeito de um arithmetic shift right de uma posição?

Se o inteiro for unsigned, o shift pode ocasionar um valor errado. Se o inteiro for signed, o shift o divide por 2.