



**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

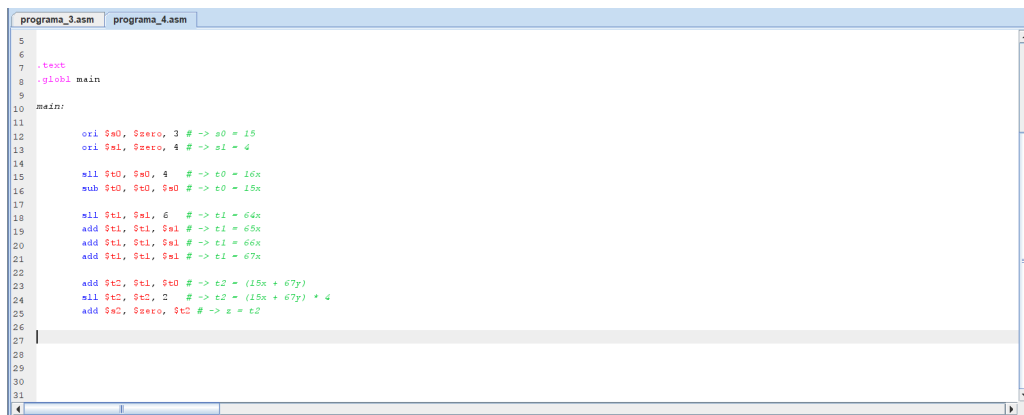
Programa - 03

```

21 main:
22
23     ori $a0, $zero, 3 # Atribuição s0 = 15
24     ori $a1, $zero, 4 # Atribuição s1 = 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 = 8y
35     add $t1, $t1, $a1 # Atribuição t1 = 16y
36     add $t1, $t1, $a1 # Atribuição t1 = 32y
37     add $t1, $t1, $a1 # Atribuição t1 = 64y
38     add $t1, $t1, $a1 # Atribuição t1 = 128y
39     add $t1, $t1, $a1 # Atribuição t1 = 256y
40     add $t1, $t1, $a1 # Atribuição t1 = 512y
41
42     add $t2, $t0, $t1 # 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

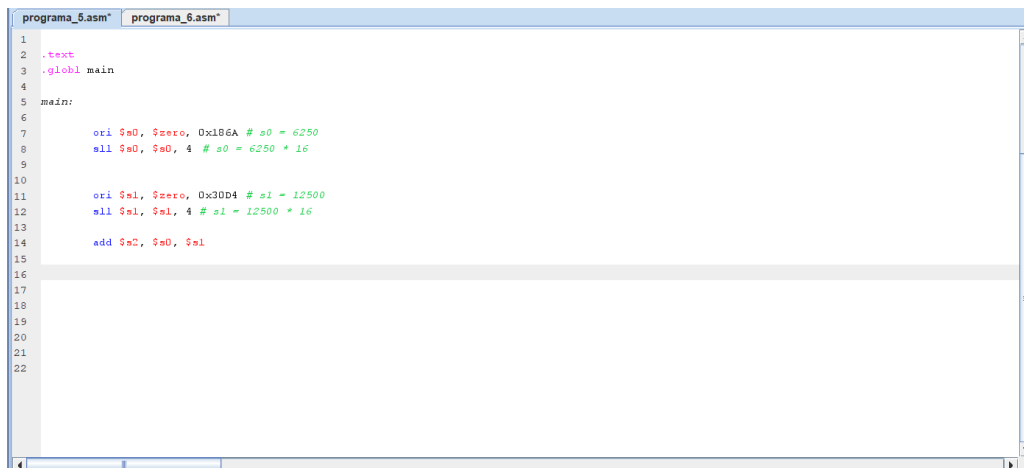
Programa - 04



```
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

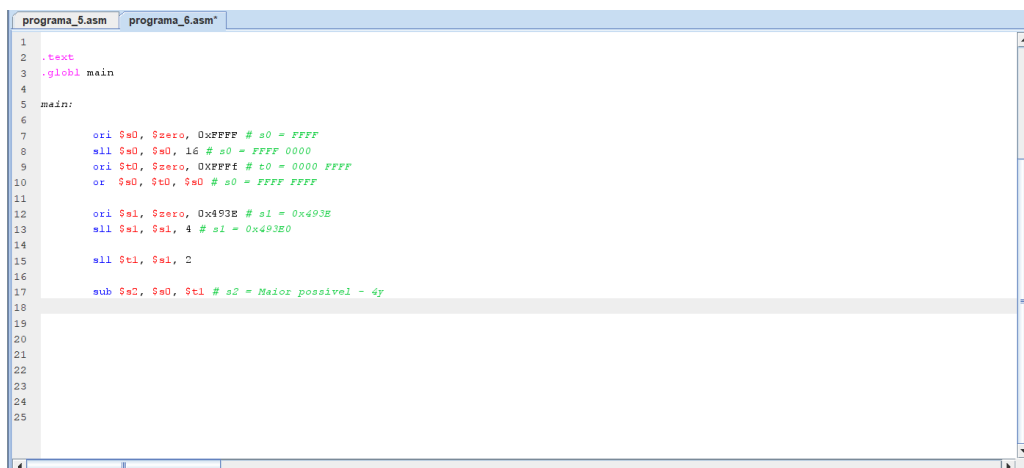
Programa - 05



```
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

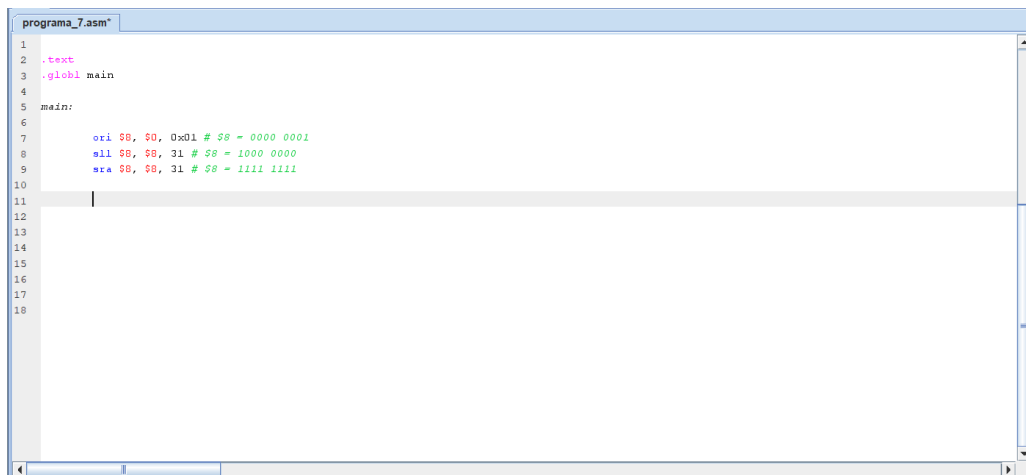
Programa - 06



```
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



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

Figura 7: Imagem do programa 7

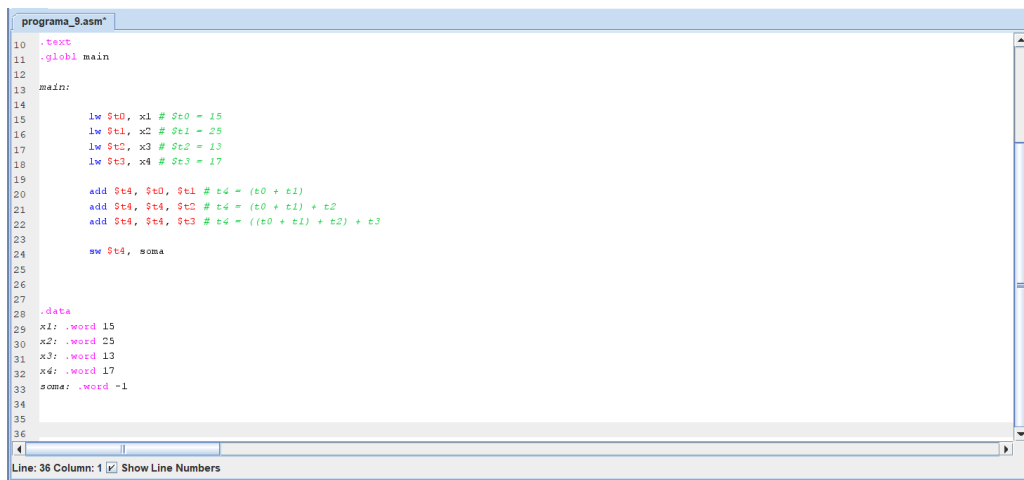
Programa - 08



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

Figura 8: Imagem do programa 8

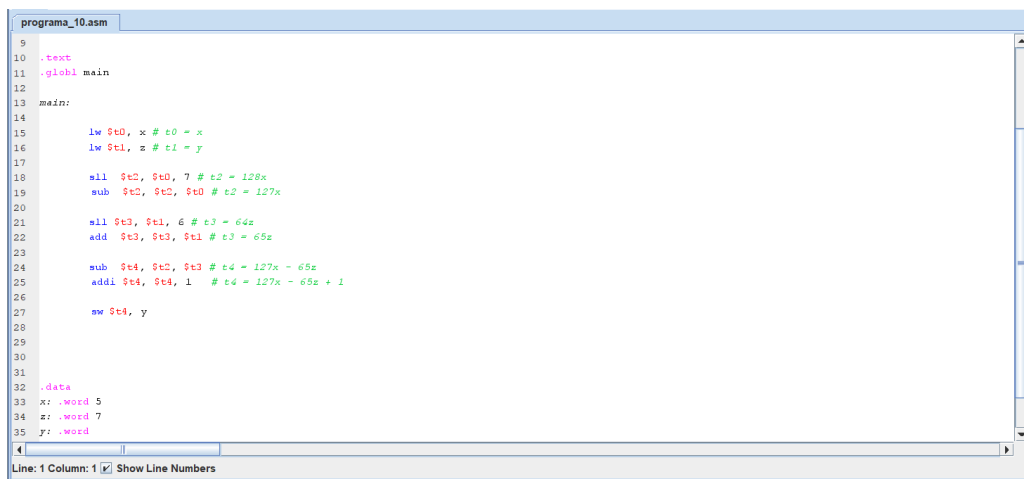
Programa - 09



```
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

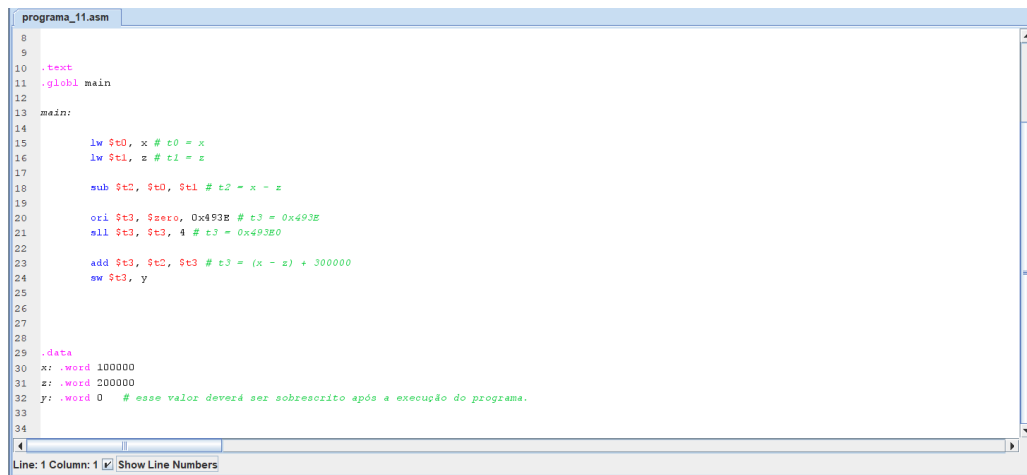
Programa - 10



```
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

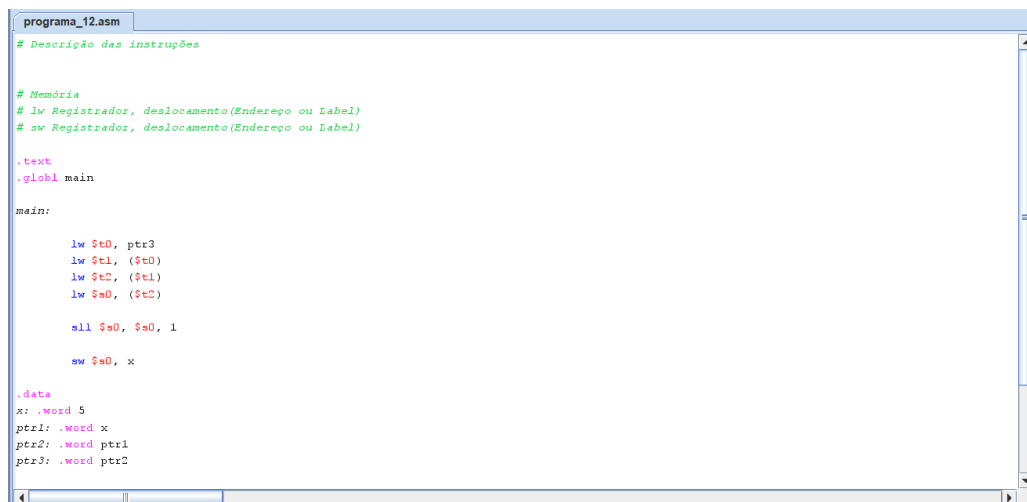
Programa - 11



```
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
.text
.globl main

main:

    lw $t0, A # t0 = A

    sra $t1, $t0, 31
    beq $t1, $zero, positivo
    sub $t0, $zero, $t0
    sw $t0, A

positivo:
    # FIM

.data
A: .word -7
```

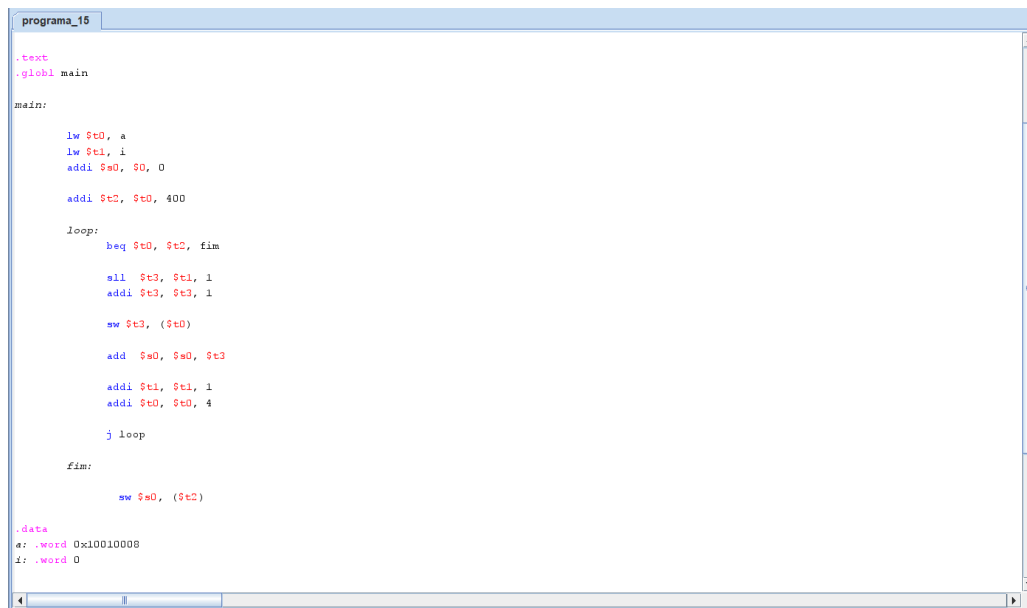
Figura 13: Imagem do programa 13

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

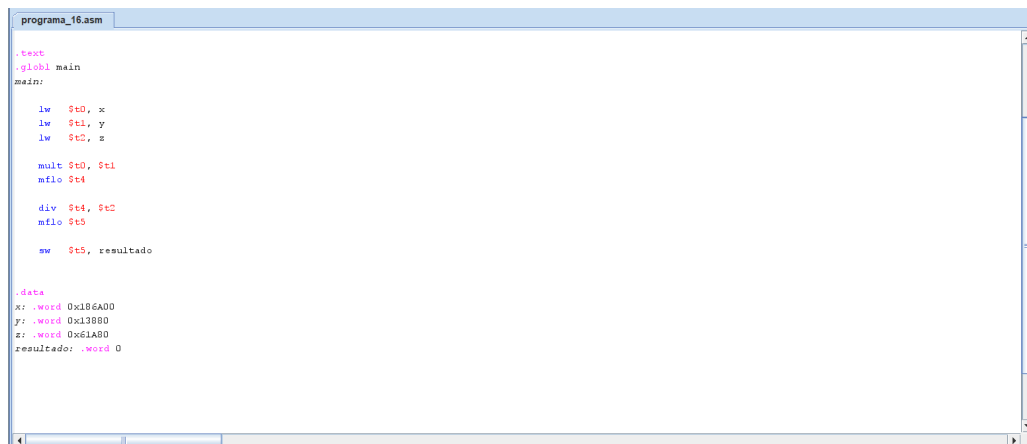
Programa - 15



```
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

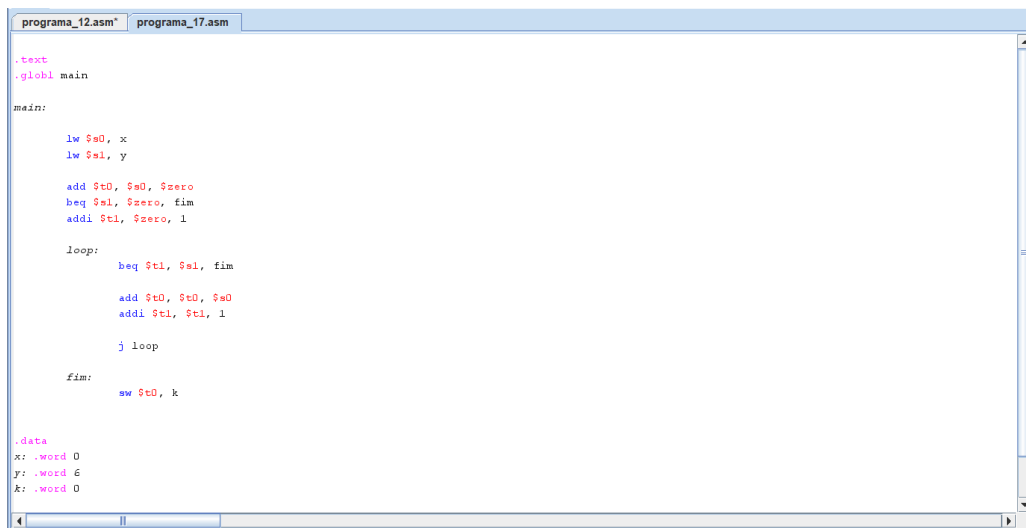
Programa - 16



```
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

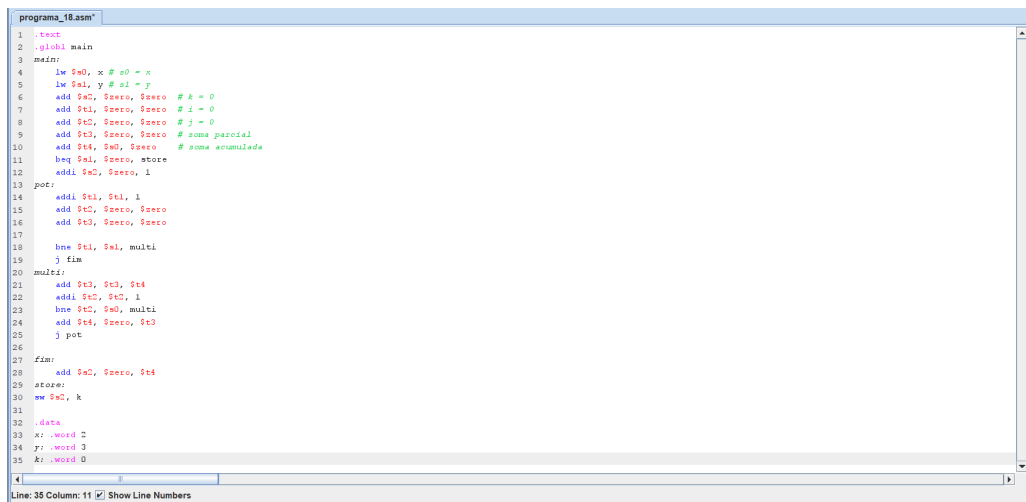
    j loop

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 # x0 = 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    add $t3, $t3, $t4
21    addi $t0, $t0, 1
22    bne $t0, $s0, multi
23    add $t4, $zero, $t3
24    j pot
25 fim:
26    add $s2, $zero, $t4
27    store:
28    sw $s2, k
29
30 .data
31 x: .word 2
32 y: .word 3
33 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         mthi $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
```

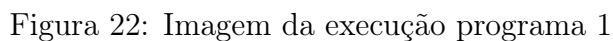
Figura 20: Imagem do programa 20

Programa - 21

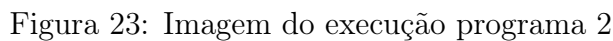
```
programa_21.asm
9
10 .text
11 .globl main
12 main:
13     lw $s0, x
14     sra $t0, $s0, 31
15     beq $t0, $zero, positive
16
17     non_positive:
18         mult $s0, $s0
19         mflo $t1
20         mult $t1, $s0
21         mflo $t2
22         mult $t2, $s0
23         mflo $t3
24         addi $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

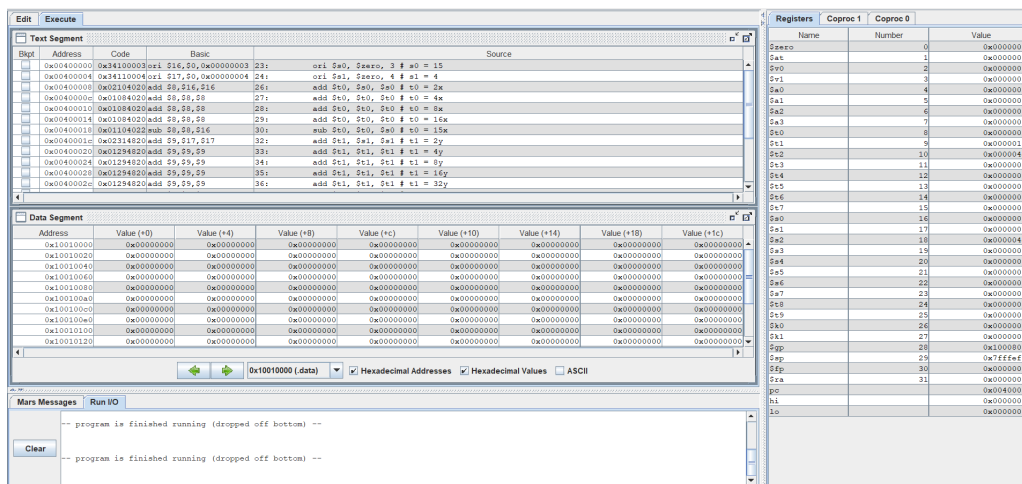


Figura 24: Imagem do execução programa 3

Execução - 04

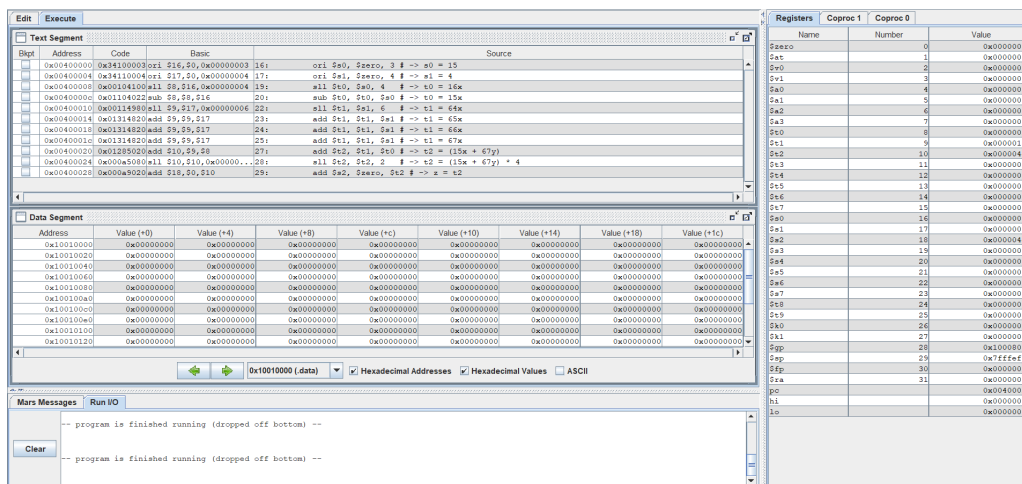


Figura 25: Imagem do execução programa 4

Execução - 05

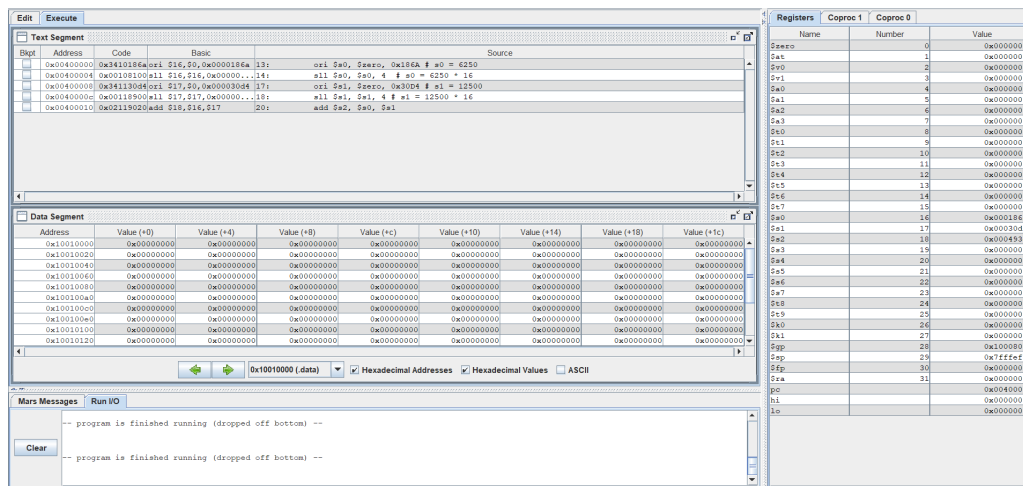


Figura 26: Imagem do execução programa 5

Execução - 06

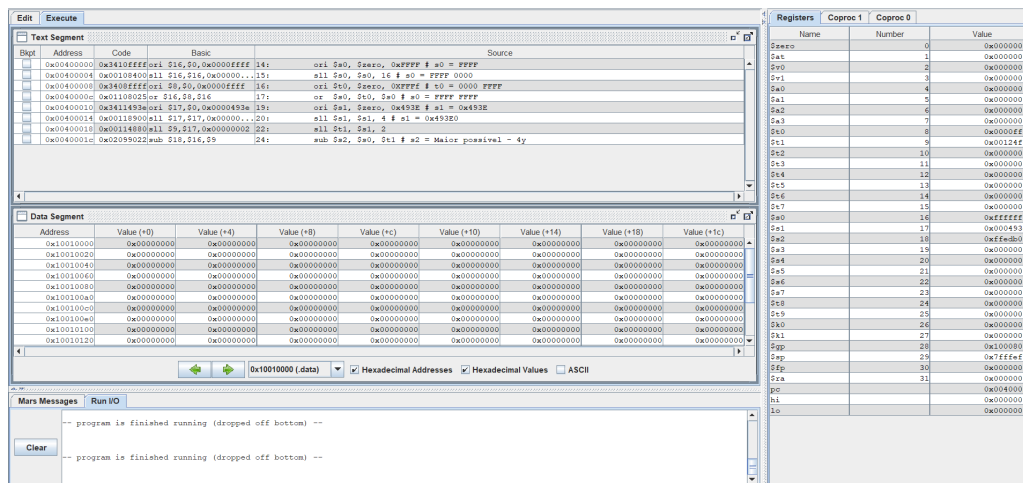


Figura 27: Imagem do execução programa 6

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			Sr0	2	0x00000000
							Sr1	3	0x00000000
							Sr2	4	0x00000000
							Sr3	5	0x00000000
							Sr4	6	0x00000000
							Sr5	7	0x00000000
							Sr6	8	0xffffffff
							Sr7	9	0x00000000
							Sr8	10	0x00000000
							Sr9	11	0x00000000
							Sr10	12	0x00000000
							Sr11	13	0x00000000
							Sr12	14	0x00000000
							Sr13	15	0x00000000
							Sr14	16	0x00000000
							Sr15	17	0x00000000
							Sr16	18	0x00000000
							Sr17	19	0x00000000
							Sr18	20	0x00000000
							Sr19	21	0x00000000
							Sr20	22	0x00000000
							Sr21	23	0x00000000
							Sr22	24	0x00000000
							Sr23	25	0x00000000
							Sr24	26	0x00000000
							Sr25	27	0x00000000
							Sr26	28	0x00000000
							Sr27	29	0x7fffffff
							Sr28	30	0x00000000
							Sr29	31	0x00000000
							Sr30		0x00000000
							Sr31		0x00000000
							Sr32		0x00000000
							Sr33		0x00000000
							Sr34		0x00000000
							Sr35		0x00000000
							Sr36		0x00000000
							Sr37		0x00000000
							Sr38		0x00000000
							Sr39		0x00000000
							Sr40		0x00000000
							Sr41		0x00000000
							Sr42		0x00000000
							Sr43		0x00000000
							Sr44		0x00000000
							Sr45		0x00000000
							Sr46		0x00000000
							Sr47		0x00000000
							Sr48		0x00000000
							Sr49		0x00000000
							Sr50		0x00000000
							Sr51		0x00000000
							Sr52		0x00000000
							Sr53		0x00000000
							Sr54		0x00000000
							Sr55		0x00000000
							Sr56		0x00000000
							Sr57		0x00000000
							Sr58		0x00000000
							Sr59		0x00000000
							Sr60		0x00000000
							Sr61		0x00000000
							Sr62		0x00000000
							Sr63		0x00000000
							Sr64		0x00000000
							Sr65		0x00000000
							Sr66		0x00000000
							Sr67		0x00000000
							Sr68		0x00000000
							Sr69		0x00000000
							Sr70		0x00000000
							Sr71		0x00000000
							Sr72		0x00000000
							Sr73		0x00000000
							Sr74		0x00000000
							Sr75		0x00000000
							Sr76		0x00000000
							Sr77		0x00000000
							Sr78		0x00000000
							Sr79		0x00000000
							Sr80		0x00000000
							Sr81		0x00000000
							Sr82		0x00000000
							Sr83		0x00000000
							Sr84		0x00000000
							Sr85		0x00000000
							Sr86		0x00000000
							Sr87		0x00000000
							Sr88		0x00000000
							Sr89		0x00000000
							Sr90		0x00000000
							Sr91		0x00000000
							Sr92		0x00000000
							Sr93		0x00000000
							Sr94		0x00000000
							Sr95		0x00000000
							Sr96		0x00000000
							Sr97		0x00000000
							Sr98		0x00000000
							Sr99		0x00000000
							Sr100		0x00000000
							Sr101		0x00000000
							Sr102		0x00000000
							Sr103		0x00000000
							Sr104		0x00000000
							Sr105		0x00000000
							Sr106		0x00000000
							Sr107		0x00000000
							Sr108		0x00000000
							Sr109		0x00000000
							Sr110		0x00000000
							Sr111		0x00000000
							Sr112		0x00000000
							Sr113		0x00000000
							Sr114		0x00000000
							Sr115		0x00000000
							Sr116		0x00000000
							Sr117		0x00000000
							Sr118		0x00000000
							Sr119		0x00000000
							Sr120		0x00000000
							Sr121		0x00000000
							Sr122		0x00000000
							Sr123		0x00000000
							Sr124		0x00000000
							Sr125		0x00000000
							Sr126		0x00000000
							Sr127		0x00000000
							Sr128		0x00000000
							Sr129		0x00000000
							Sr130		0x00000000
							Sr131		0x00000000
							Sr132		0x00000000
							Sr133		0x00000000
							Sr134		0x00000000
							Sr135		0x00000000
							Sr136		0x00000000
							Sr137		0x00000000
							Sr138		0x00000000
							Sr139		0x00000000
							Sr140		0x00000000
							Sr141		0x00000000
							Sr142		0x00000000
							Sr143		0x00000000
							Sr144		0x00000000
							Sr145		0x00000000
							Sr146		0x00000000
							Sr147		0x00000000
							Sr148		0x00000000
							Sr149		0x00000000
							Sr150		0x00000000
							Sr151		0x00000000
							Sr152		0x00000000
							Sr153		0x00000000
							Sr154		0x00000000
							Sr155		0x00000000
							Sr156		0x00000000
							Sr157		0x00000000
							Sr158		0x00000000
							Sr159		0x00000000
							Sr160		0x00000000
							Sr161		0x00000000
							Sr162		0x00000000
							Sr163		0x00000000
							Sr164		0x00000000
							Sr165		0x00000000
							Sr166		0x00000000
							Sr167		0x00000000
							Sr168		0x00000000
							Sr169		0x00000000
							Sr170		0x00000000
							Sr171		0x00000000
							Sr172		0x00000000
							Sr173		0x00000000
							Sr174		0x00000000
							Sr175		0x00000000
							Sr176		0x00000000
							Sr177		0x00000000
							Sr178		0x00000000
							Sr179		0x00000000
							Sr180		0x00000000
							Sr181		0x00000000
							Sr182		0x00000000
							Sr183		0x00000000
							Sr184		0x00000000
							Sr185		0x00000000
							Sr186		0x00000000
							Sr187		0x00000000
							Sr188		0x00000000
							Sr189		0x00000000
							Sr190		0x00000000
							Sr191		0x00000000
							Sr192		0x00000000
							Sr193		0x00000000
							Sr194		0x00000000
							Sr195		0x00000000
							Sr196		0x00000000
							Sr197		0x00000000
							Sr198		0x00000000
							Sr199		0x00000000
							Sr200		0x00000000
							Sr201		0x00000000
							Sr202		0x00000000
							Sr203		0x00000000
							Sr204		0x00000000
							Sr205		0x00000000
							Sr206		0x00000000
							Sr207		0x00000000
							Sr208		0x00000000
							Sr209		0x00000000
							Sr210		0x00000000
							Sr211		0x00000000
							Sr212		0x00000000
							Sr213		0x00000000
							Sr214		0x00000000
							Sr215		0x00000000
							Sr216		0x00000000
							Sr217		0x00000000
							Sr218		0x00000000
							Sr219		0x00000000
							Sr220		0x00000000
							Sr221		0x00000000
							Sr222		0x00000000
							Sr223		0x00000000
							Sr224		0x00000000
							Sr225		0x00000000
							Sr226		0x00000000
							Sr227		0x00000000
							Sr228		0x00000000
							Sr229		0x00000000
							Sr230		0x00000000
							Sr231		0x00000000
							Sr232		0x00000000
							Sr233		0x00000000
							Sr234		0x00000000
							Sr235		0x00000000
							Sr236		0x00000000
							Sr237		0x00000000
							Sr238		0x00000000
							Sr239		0x00000000
							Sr240		0x00000000
							Sr241		0x00000000
							Sr242		0x00000000
							Sr243		0x00000000
							Sr244		0x00000000
							Sr245		0x00000000
							Sr246		0x00000000
							Sr247		0x00000000
							Sr248		0x00000000
							Sr249		0x00000000
							Sr250		0x00000000
							Sr251		0x00000000
							Sr252		0x00000000
							Sr253		0x00000000
							Sr254		0x00000000
							Sr255		0x00000000
							Sr256		0x00000000
							Sr257		0x00000000
							Sr258		0x00000000
							Sr259		0x00000000
							Sr260		0x00000000
							Sr261		0x00000000
							Sr262		0x00000000
							Sr263		0x00000000
							Sr264		0x00000000
							Sr265		0x00000000
							Sr266		0x00000000
							Sr267		0x00000000
							Sr268		0x00000000
							Sr269		0x00000000
							Sr270		0x00000000
							Sr271		0x00000000
							Sr272		0x00000000
							Sr273		0x00000000
							Sr274		0x00000000
							Sr275		0x00000000
							Sr276		0x00000000

Execução - 10

Text Segment						Source	
Offset	Address	Code	Basic				
0	0x04000000	0x3e011001 lui	\$1,0x00001001	15:	lw \$t0, x # t0 = x		
1	0x04000004	0xc2200000 lw	\$9,0x00000000(\$t1)				
2	0x04000008	0xc3011001 lui	\$1,0x00001001	16:	lw \$t1, x # t1 = y		
3	0x0400000c	0xc2200004 lw	\$9,0x00000004(\$t1)				
4	0x04000010	0x00059502 sub	\$10,\$9,0x00000007	18:	add \$t2, \$t0, 7 # t2 = t0+7		
5	0x04000014	0xc0149502 sub	\$10,\$t0,\$9	19:	sub \$t2, \$t2, \$t0 # t2 = t2 - t0		
6	0x04000018	0x00059500 addi	\$11,\$10,0x00000006	21:	add \$t3, \$t2, 6 # t3 = t2 + 6		
7	0x0400001c	0xc0149502 addi	\$11,\$t1,\$9	22:	add \$t3, \$t3, \$t1 # t3 = t3 + t1		
8	0x04000020	0xc0149502 sub	\$12,\$t0,\$t1	24:	sub \$t4, \$t2, \$t3 # t4 = t2 - t3		
9	0x04000024	0x2018c001 addi	\$12,\$t2,0x0000...05:		addi \$t4, \$t4, 1 # t4 = t2 - t3 + 1		
10	0x04000028	0xc3011001 lui	\$1,0x00001001	27:	sw \$t4, y		
11	0x0400002c	0xc2200000 sw	\$12,0x00000000(\$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	0x00000005
\$t1	9	0x00000007
\$t2	10	0x00000006
\$t3	11	0x00000007
\$t4	12	0x00000005
\$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	0x10009900
\$fp	29	0x7ffffefc
\$ra	30	0x00000000
\$pc	31	0x00000000
\$hi		0x00000000
\$lo		0x00000000

Mars Messages Run I/O

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

Clear

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

Figura 31: Imagem do execução programa 10

Execução - 11

Text Segment						Source	
Offset	Address	Code	Basic				
0	0x04000000	0x3e011001 lui	\$1,0x00001001	15:	lw \$t0, x # t0 = x		
1	0x04000004	0xc2200000 lw	\$9,0x00000000(\$t1)				
2	0x04000008	0xc3011001 lui	\$1,0x00001001	16:	lw \$t1, x # t1 = x		
3	0x0400000c	0xc2200004 lw	\$9,0x00000004(\$t1)				
4	0x04000010	0x00059502 sub	\$10,\$9,\$9	18:	sub \$t2, \$t0, \$t1 # t2 = x - x		
5	0x04000014	0xc0149502 sub	\$10,\$t0,\$9	20:	add \$t3, \$zero, 0x4932 # t3 = 0x4932		
6	0x04000018	0x00059500 addi	\$11,\$t1,0x0000...21:		add \$t3, \$t2, 4 # t3 = 0x4932		
7	0x0400001c	0xc0149502 addi	\$11,\$t0,\$t1	23:	add \$t3, \$t2, \$t3 # t3 = (x - x) + 300000		
8	0x04000020	0xc3011001 lui	\$1,0x00001001	24:	sw \$t3, y		
9	0x04000024	0xc2200000 sw	\$12,0x00000000(\$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	0x00000000
\$t1	9	0x00000000
\$t2	10	0xffff7960
\$t3	11	0x20000019
\$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	0x10009900
\$fp	29	0x7ffffefc
\$ra	30	0x00000000
\$pc	31	0x00000000
\$hi		0x00000000
\$lo		0x00000000

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						Source	
Offset	Address	Code	Basic				
0	0x04000000	0x3e011001 lui	\$1,0x00001001	13:	lw \$t0, ptr3		
1	0x04000004	0xc2200000 lw	\$9,0x00000000(\$t1)				
2	0x04000008	0xc2200000 lw	\$9,0x00000000(\$t1)	14:	lw \$t1, (\$t0)		
3	0x0400000c	0xc2200000 lw	\$10,0x00000000(\$t1)	15:	lw \$t2, (\$t1)		
4	0x04000010	0xc2200000 lw	\$16,0x00000000(\$t1)	16:	lw \$a0, (\$t2)		
5	0x04000014	0xc0100040 addi	\$16,\$t6,0x0000...18:		add \$a0, \$a0, 1		
6	0x04000018	0xc3011001 lui	\$1,0x00001001	20:	sw \$a0, x		
7	0x0400001c	0xc2200000 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	0x0000000a
\$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	0x10009900
\$fp	29	0x7ffffefc
\$ra	30	0x00000000
\$pc	31	0x00000000
\$hi		0x00000000
\$lo		0x00000000

Mars Messages Run 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

Text Segment						Name		Number	Value
Bit	Address	Code	Basic	Source					
	0x0400000	0x0e11001	lui \$1,0x00001001	25: lw \$t0, A # t0 = A		\$zero	0	0x00000000	
	0x0400004	0xc280000	lw \$9,0x00000000(\$t1)			\$t1	1	0x10110000	
	0x0400008	0x008f6f3	and \$9,\$9,0x00000014	27: andi \$t1, \$t0, 31		\$t0	2	0x00000000	
	0x040000c	0x1120003	bneq \$9,\$9,0x00000003	28: bneq \$t1, \$zero, positivo		\$t1	3	0x00000000	
	0x0400010	0x008402d	sub \$9,\$9,28	29: sub \$t0, \$zero, \$t0		\$a1	4	0x00000000	
	0x0400014	0xc211001	lui \$1,0x00001001	30: sw \$t0, A		\$a2	5	0x00000000	
	0x0400018	0xc280000	sw \$9,0x00000000(\$t1)			\$a3	6	0x00000000	
						\$t0	7	0x00000000	
						\$t1	8	0x00000000	
						\$t2	9	0xffffffff	
						\$t3	10	0x00000000	
						\$t4	11	0x00000000	
						\$t5	12	0x00000000	
						\$t6	13	0x00000000	
						\$t7	14	0x00000000	
						\$t8	15	0x00000000	
						\$t9	16	0x00000000	
						\$a1	17	0x00000000	
						\$a2	18	0x00000000	
						\$a3	19	0x00000000	
						\$a4	20	0x00000000	
						\$a5	21	0x00000000	
						\$a6	22	0x00000000	
						\$a7	23	0x00000000	
						\$a8	24	0x00000000	
						\$a9	25	0x00000000	
						\$k0	26	0x00000000	
						\$k1	27	0x00000000	
						\$gp	28	0x00000000	
						\$fp	29	0x7fffffc0	
						\$tp	30	0x00000000	
						\$ra	31	0x00000000	
						\$s0		0x00000000	
						\$s1		0x00000000	
						\$s2		0x00000000	
						\$s3		0x00000000	
						\$s4		0x00000000	
						\$s5		0x00000000	
						\$s6		0x00000000	
						\$s7		0x00000000	
						\$s8		0x00000000	
						\$s9		0x00000000	
						\$k0		0x00000000	
						\$k1		0x00000000	
						\$gp		0x00000000	
						\$fp		0x7fffffc0	
						\$tp		0x00000000	
						\$ra		0x00000000	
						\$s0		0x00000000	
						\$s1		0x00000000	
						\$s2		0x00000000	
						\$s3		0x00000000	
						\$s4		0x00000000	
						\$s5		0x00000000	
						\$s6		0x00000000	
						\$s7		0x00000000	
						\$s8		0x00000000	
						\$s9		0x00000000	
						\$k0		0x00000000	
						\$k1		0x00000000	
						\$gp		0x00000000	
						\$fp		0x7fffffc0	
						\$tp		0x00000000	
						\$ra		0x00000000	
						\$s0		0x00000000	
						\$s1		0x00000000	
						\$s2		0x00000000	
						\$s3		0x00000000	
						\$s4		0x00000000	
						\$s5		0x00000000	
						\$s6		0x00000000	
						\$s7		0x00000000	
						\$s8		0x00000000	
						\$s9		0x00000000	
						\$k0		0x00000000	
						\$k1		0x00000000	
						\$gp		0x00000000	
						\$fp		0x7fffffc0	
						\$tp		0x00000000	
						\$ra		0x00000000	
						\$s0		0x00000000	
						\$s1		0x00000000	
						\$s2		0x00000000	
						\$s3		0x00000000	
						\$s4		0x00000000	
						\$s5		0x00000000	
						\$s6		0x00000000	
						\$s7		0x00000000	
						\$s8		0x00000000	
						\$s9		0x00000000	
						\$k0		0x00000000	
						\$k1		0x00000000	
						\$gp		0x00000000	
						\$fp		0x7fffffc0	
						\$tp		0x00000000	
						\$ra		0x00000000	
						\$s0		0x00000000	
						\$s1		0x00000000	
						\$s2		0x00000000	
						\$s3		0x00000000	
						\$s4		0x00000000	
						\$s5		0x00000000	
						\$s6		0x00000000	
						\$s7		0x00000000	
						\$s8		0x00000000	
						\$s9		0x00000000	
						\$k0		0x00000000	
						\$k1		0x00000000	
						\$gp		0x00000000	
						\$fp		0x7fffffc0	
						\$tp		0x00000000	
						\$ra		0x00000000	
						\$s0		0x00000000	
						\$s1		0x00000000	
						\$s2		0x00000000	
						\$s3		0x00000000	
						\$s4		0x00000000	
						\$s5		0x00000000	
						\$s6		0x00000000	
						\$s7		0x00000000	
						\$s8		0x00000000	
						\$s9		0x00000000	
						\$k0		0x00000000	
						\$k1		0x00000000	
						\$gp		0x00000000	
						\$fp		0x7fffffc0	
						\$tp		0x00000000	
						\$ra		0x00000000	
						\$s0		0x00000000	
						\$s1		0x00000000	
						\$s2		0x00000000	
						\$s3		0x00000000	
						\$s4		0x00000000	
						\$s5		0x00000000	
						\$s6		0x00000000	
						\$s7		0x00000000	
						\$s8		0x00000000	
						\$s9		0x00000000	
						\$k0		0x00000000	
						\$k1		0x00000000	
						\$gp		0x00000000	
						\$fp		0x7fffffc0	
						\$tp		0x00000000	
						\$ra		0x00000000	
						\$s0		0x00000000	
						\$s1		0x00000000	
						\$s2		0x00000000	
						\$s3		0x00000000	
						\$s4		0x00000000	
						\$s5		0x00000000	
						\$s6		0x00000000	
						\$s7		0x00000000	
						\$s8		0x00000000	
						\$s9		0x00000000	
						\$k0		0x00000000	
						\$k1		0x00000000	
						\$gp		0x00000000	
						\$fp		0x7fffffc0	
						\$tp		0x00000000	
						\$ra		0x00000000	
						\$s0		0x00000000	
						\$s1		0x00000000	
						\$s2		0x00000000	
						\$s3		0x00000000	
						\$s4		0x00000000	
						\$s5		0x00000000	
						\$s6		0x00000000	
						\$s7		0x00000000	
						\$s8		0x00000000	
						\$s9		0x00000000	
						\$k0		0x00000000	
						\$k1		0x00000000	
						\$gp		0x00000000	
						\$fp		0x7fffffc0	
						\$tp		0x00000000	
						\$ra		0x00000000	
						\$s0		0x00000000	
						\$s1		0x00000000	
						\$s2		0x00000000	
						\$s3		0x00000000	
						\$s4		0x00000000	
						\$s5		0x00000000	
						\$s6		0x00000000	
						\$s7		0x00000000	
						\$s8		0x00000000	
						\$s9		0x00000000	
						\$k0		0x00000000	
						\$k1		0x00000000	
						\$gp		0x00000000	
						\$fp		0x7fffffc0	
						\$tp		0x00000000	
						\$ra		0x00000000	
						\$s0		0x00000000	
						\$s1		0x00000000	
						\$s2		0x00000000	
						\$s3		0x00000000	
						\$s4		0x00000000	
						\$s5		0x00000000	
						\$s6		0x00000000	
						\$s7		0x00000000	
						\$s8		0x00000000	
						\$s9		0x00000000	
						\$k0		0x00000000	
						\$k1		0x00000000	
						\$gp		0x00000000	
						\$fp		0x7fffffc0	
						\$tp		0x00000000	
						\$ra		0x00000000	
						\$s0		0x00000000	
						\$s1		0x00000000	
						\$s2		0x00000000	
						\$s3		0x00000000	
						\$s4		0x00000000	
						\$s5		0x00000000	
						\$s6		0x00000000	
						\$s7		0x00000000	
						\$s8		0x00000000	
						\$s9		0x00000000	
						\$k0		0x00000000	
						\$k1		0x00000000	
						\$gp		0x0000000	

Execução - 16

Text Segment						Name	
Offset	Address	Code	Basic	Source	Number	Value	
0x00400000	0x0011001	lui	\$1,0x00001001	14: lw \$t0, x	\$zero	0	
0x00400004	0xc300000	lw	\$t0,0x00000000(\$t1)		\$at	1	
0x00400008	0x0011001	lui	\$1,0x00001001	15: lw \$t1, y	\$v0	2	
0x0040000c	0xc300004	lw	\$t0,0x00000004(\$t1)		\$v1	3	
0x00400010	0x0011001	lui	\$1,0x00001001	16: lw \$t2, z	\$a0	4	
0x00400014	0xc300008	lw	\$t0,0x00000008(\$t1)		\$a1	5	
0x00400018	0x00100018	mult	\$t0,\$t1		\$a2	6	
0x0040001c	0x0000012	mflw	\$t2		\$a3	7	
0x00400020	0x0000014	dsp	\$t4,\$t2		\$t0	8	
0x00400024	0x0000015	mflw	\$t5		\$t1	9	
0x00400028	0x0000016	mflw	\$t5,resultado		\$t2	10	
0x0040002c	0xc30000c	sw	\$t5,resultado		\$t3	11	
0x00400030	0xc300010	sw	\$t5,resultado		\$t4	12	
0x00400034	0xc300014	sw	\$t5,resultado		\$t5	13	
0x00400038	0xc300018	sw	\$t5,resultado		\$t6	14	
0x0040003c	0xc30001c	sw	\$t5,resultado		\$t7	15	
0x00400040	0xc300020	sw	\$t5,resultado		\$a0	16	
0x00400044	0xc300024	sw	\$t5,resultado		\$a1	17	
0x00400048	0xc300028	sw	\$t5,resultado		\$a2	18	
0x0040004c	0xc30002c	sw	\$t5,resultado		\$a3	19	
0x00400050	0xc300030	sw	\$t5,resultado		\$a4	20	
0x00400054	0xc300034	sw	\$t5,resultado		\$a5	21	
0x00400058	0xc300038	sw	\$t5,resultado		\$a6	22	
0x0040005c	0xc30003c	sw	\$t5,resultado		\$a7	23	
0x00400060	0xc300040	sw	\$t5,resultado		\$a8	24	
0x00400064	0xc300044	sw	\$t5,resultado		\$a9	25	
0x00400068	0xc300048	sw	\$t5,resultado		\$b0	26	
0x0040006c	0xc30004c	sw	\$t5,resultado		\$b1	27	
0x00400070	0xc300050	sw	\$t5,resultado		\$b2	28	
0x00400074	0xc300054	sw	\$t5,resultado		\$b3	29	
0x00400078	0xc300058	sw	\$t5,resultado		\$b4	30	
0x0040007c	0xc30005c	sw	\$t5,resultado		\$b5	31	
0x00400080	0xc300060	sw	\$t5,resultado		\$b6	32	
0x00400084	0xc300064	sw	\$t5,resultado		\$b7	33	
0x00400088	0xc300068	sw	\$t5,resultado		\$b8	34	
0x0040008c	0xc30006c	sw	\$t5,resultado		\$b9	35	
0x00400090	0xc300070	sw	\$t5,resultado		\$ba	36	
0x00400094	0xc300074	sw	\$t5,resultado		\$bb	37	
0x00400098	0xc300078	sw	\$t5,resultado		\$bc	38	
0x0040009c	0xc30007c	sw	\$t5,resultado		\$bd	39	
0x004000a0	0xc300080	sw	\$t5,resultado		\$be	40	
0x004000a4	0xc300084	sw	\$t5,resultado		\$bf	41	
0x004000a8	0xc300088	sw	\$t5,resultado		\$c0	42	
0x004000ac	0xc30008c	sw	\$t5,resultado		\$c1	43	
0x004000b0	0xc300090	sw	\$t5,resultado		\$c2	44	
0x004000b4	0xc300094	sw	\$t5,resultado		\$c3	45	
0x004000b8	0xc300098	sw	\$t5,resultado		\$c4	46	
0x004000bc	0xc30009c	sw	\$t5,resultado		\$c5	47	
0x004000c0	0xc3000a0	sw	\$t5,resultado		\$c6	48	
0x004000c4	0xc3000a4	sw	\$t5,resultado		\$c7	49	
0x004000c8	0xc3000a8	sw	\$t5,resultado		\$c8	50	
0x004000cc	0xc3000ac	sw	\$t5,resultado		\$c9	51	
0x004000d0	0xc3000b0	sw	\$t5,resultado		\$ca	52	
0x004000d4	0xc3000b4	sw	\$t5,resultado		\$cb	53	
0x004000d8	0xc3000b8	sw	\$t5,resultado		\$cc	54	
0x004000dc	0xc3000bc	sw	\$t5,resultado		\$cd	55	
0x004000e0	0xc3000c0	sw	\$t5,resultado		\$ce	56	
0x004000e4	0xc3000c4	sw	\$t5,resultado		\$cf	57	
0x004000e8	0xc3000c8	sw	\$t5,resultado		\$d0	58	
0x004000ec	0xc3000cc	sw	\$t5,resultado		\$d1	59	
0x004000f0	0xc3000d0	sw	\$t5,resultado		\$d2	60	
0x004000f4	0xc3000d4	sw	\$t5,resultado		\$d3	61	
0x004000f8	0xc3000d8	sw	\$t5,resultado		\$d4	62	
0x004000fc	0xc3000dc	sw	\$t5,resultado		\$d5	63	
0x00400100	0xc3000e0	sw	\$t5,resultado		\$d6	64	
0x00400104	0xc3000e4	sw	\$t5,resultado		\$d7	65	
0x00400108	0xc3000e8	sw	\$t5,resultado		\$d8	66	
0x0040010c	0xc3000ec	sw	\$t5,resultado		\$d9	67	
0x00400110	0xc3000f0	sw	\$t5,resultado		\$da	68	
0x00400114	0xc3000f4	sw	\$t5,resultado		\$db	69	
0x00400118	0xc3000f8	sw	\$t5,resultado		\$dc	70	
0x0040011c	0xc3000fc	sw	\$t5,resultado		\$dd	71	
0x00400120	0xc300100	sw	\$t5,resultado		\$de	72	
0x00400124	0xc300104	sw	\$t5,resultado		\$df	73	
0x00400128	0xc300108	sw	\$t5,resultado		\$e0	74	
0x0040012c	0xc30010c	sw	\$t5,resultado		\$e1	75	
0x00400130	0xc300110	sw	\$t5,resultado		\$e2	76	
0x00400134	0xc300114	sw	\$t5,resultado		\$e3	77	
0x00400138	0xc300118	sw	\$t5,resultado		\$e4	78	
0x0040013c	0xc30011c	sw	\$t5,resultado		\$e5	79	
0x00400140	0xc300120	sw	\$t5,resultado		\$e6	80	
0x00400144	0xc300124	sw	\$t5,resultado		\$e7	81	
0x00400148	0xc300128	sw	\$t5,resultado		\$e8	82	
0x0040014c	0xc30012c	sw	\$t5,resultado		\$e9	83	
0x00400150	0xc300130	sw	\$t5,resultado		\$ea	84	
0x00400154	0xc300134	sw	\$t5,resultado		\$eb	85	
0x00400158	0xc300138	sw	\$t5,resultado		\$ec	86	
0x0040015c	0xc30013c	sw	\$t5,resultado		\$ed	87	
0x00400160	0xc300140	sw	\$t5,resultado		\$ee	88	
0x00400164	0xc300144	sw	\$t5,resultado		\$ef	89	
0x00400168	0xc300148	sw	\$t5,resultado		\$f0	90	
0x0040016c	0xc30014c	sw	\$t5,resultado		\$f1	91	
0x00400170	0xc300150	sw	\$t5,resultado		\$f2	92	
0x00400174	0xc300154	sw	\$t5,resultado		\$f3	93	
0x00400178	0xc300158	sw	\$t5,resultado		\$f4	94	
0x0040017c	0xc30015c	sw	\$t5,resultado		\$f5	95	
0x00400180	0xc300160	sw	\$t5,resultado		\$f6	96	
0x00400184	0xc300164	sw	\$t5,resultado		\$f7	97	
0x00400188	0xc300168	sw	\$t5,resultado		\$f8	98	
0x0040018c	0xc30016c	sw	\$t5,resultado		\$f9	99	
0x00400190	0xc300170	sw	\$t5,resultado		\$fa	100	
0x00400194	0xc300174	sw	\$t5,resultado		\$fb	101	
0x00400198	0xc300178	sw	\$t5,resultado		\$fc	102	
0x0040019c	0xc30017c	sw	\$t5,resultado		\$fd	103	
0x004001a0	0xc300180	sw	\$t5,resultado		\$fe	104	
0x004001a4	0xc300184	sw	\$t5,resultado		\$ff	105	
0x004001a8	0xc300188	sw	\$t5,resultado		\$0	106	
0x004001ac	0xc30018c	sw	\$t5,resultado		\$1	107	
0x004001b0	0xc300190	sw	\$t5,resultado		\$2	108	
0x004001b4	0xc300194	sw	\$t5,resultado		\$3	109	
0x004001b8	0xc300198	sw	\$t5,resultado		\$4	110	
0x004001bc	0xc30019c	sw	\$t5,resultado		\$5	111	
0x004001c0	0xc3001a0	sw	\$t5,resultado		\$6	112	
0x004001c4	0xc3001a4	sw	\$t5,resultado		\$7	113	
0x004001c8	0xc3001a8	sw	\$t5,resultado		\$8	114	
0x004001cc	0xc3001ac	sw	\$t5,resultado		\$9	115	
0x004001d0	0xc3001b0	sw	\$t5,resultado		\$a	116	
0x004001d4	0xc3001b4	sw	\$t5,resultado		\$b	117	
0x004001d8	0xc3001b8	sw	\$t5,resultado		\$c	118	
0x004001dc	0xc3001bc	sw	\$t5,resultado		\$d	119	
0x004001e0	0xc3001c0	sw	\$t5,resultado		\$e	120	
0x004001e4	0xc3001c4	sw	\$t5,resultado		\$f	121	
0x004001e8	0xc3001c8	sw	\$t5,resultado		\$0	122	
0x004001ec	0xc3001cc	sw	\$t5,resultado		\$1	123	
0x004001f0	0xc3001d0	sw	\$t5,resultado		\$2	124	
0x004001f4	0xc3001d4	sw	\$t5,resultado		\$3	125	
0x004001f8	0xc3001d8	sw	\$t5,resultado		\$4	126	
0x004001fc	0xc3001dc	sw	\$t5,resultado		\$5	127	
0x00400200	0xc3001e0	sw	\$t5,resultado		\$6	128	
0x00400204	0xc3001e4	sw	\$t5,resultado		\$7	129	
0x00400208	0xc3001e8	sw	\$t5,resultado		\$8	130	
0x0040020c	0xc3001ec	sw	\$t5,resultado		\$9	131	
0x00400210	0xc3001f0	sw	\$t5,resultado		\$a	132	
0x00400214	0xc3001f4	sw	\$t5,resultado		\$b	133	
0x00400218	0xc3001f8	sw	\$t5,resultado		\$c	134	
0x0040021c	0xc3001fc	sw	\$t5,resultado		\$d	135	
0x00400220	0xc300200	sw	\$t5,resultado		\$e	136	
0x00400224	0xc300204	sw	\$t5,resultado		\$f	137	
0x00400228	0xc300208	sw	\$t5,resultado		\$0	138	
0x0040022c	0xc30020c	sw	\$t5,resultado		\$1	139	
0x00400230	0xc300210	sw	\$t5,resultado		\$2	140	
0x00400234	0xc300214	sw	\$t5,resultado		\$3	141	
0x00400238	0xc300218	sw	\$t5,resultado		\$4	142	
0x0040023c	0xc30021c	sw	\$t5,resultado		\$5	143	
0x00400240	0xc300220	sw	\$t5,resultado		\$6	144	
0x00400244	0xc300224	sw	\$t5,resultado		\$7	145	
0x00400248	0xc300228	sw	\$t5,resultado		\$8	146	
0x0040024c	0xc30022c	sw	\$t5,resultado		\$9	147	
0x00400250	0xc300230	sw	\$t5,resultado		\$a	148	
0x00400254	0xc300234	sw	\$t5,resultado		\$b	149	
0x00400258	0xc300238	sw	\$t5,resultado		\$c	150	
0x0040025c	0xc30023c	sw	\$t5,resultado		\$d	151	
0x00400260	0xc300240	sw	\$t5,resultado		\$e	152	
0x00400264	0xc300244	sw	\$t5,resultado		\$f	153	
0x00400268	0xc300248	sw	\$t5,resultado		\$0	154	
0x0040026c	0xc30024c	sw	\$t5,resultado		\$1	155	
0x00400270	0xc300250	sw	\$t5,resultado		\$2	156	
0x00400274	0xc300254	sw	\$t5,resultado		\$3	157	
0x00400278	0xc300258	sw	\$t5,resultado		\$4	158	
0x0040027c	0xc30025c	sw	\$t5,resultado		\$5	159	
0x00400280	0xc300260	sw	\$t5,resultado		\$6	160	
0x00400284	0xc300264	sw	\$t5,resultado		\$7	161	
0x00400288	0xc300268	sw	\$t5,resultado		\$8	162	
0x0040028c	0xc30026c	sw	\$t5,resultado		\$9	163	
0x00400290	0xc300270	sw	\$t5,resultado		\$a	164	
0x00400294	0xc300274	sw	\$t5,resultado		\$b	165	
0x00400298	0xc300278	sw	\$t5,resultado		\$c	166	
0x0040029c	0xc30027c	sw	\$t5,resultado		\$d	167	
0x004002a0	0xc300280	sw	\$t5,resultado		\$e	168	
0x004002a4	0xc300284	sw	\$t5,resultado		\$f	169	
0x004002a8	0xc300288	sw	\$t5,resultado		\$0	170	
0x004002ac	0xc30028c	sw	\$t5,resultado		\$1	171	
0x004002b0	0xc300290	sw	\$t5,resultado		\$2	172	
0x004002b4	0xc300294	sw	\$t5,resultado		\$3	173	
0x004002b8	0xc300298	sw	\$t5,resultado		\$4	174	
0x004002bc	0xc30029c	sw	\$t5,resultado		\$5	175	
0x004002c0	0xc3002a0	sw	\$t5,resultado		\$6	176	
0x004002c4	0xc3002a4	sw	\$t5,resultado		\$7	177	
0x004002c8	0xc3002a8	sw	\$t5,resultado		\$8	178	
0x004002cc	0xc3002ac	sw	\$t5,resultado		\$9	179	
0x004002d0	0xc3002b0	sw	\$t5,resultado		\$a	180	
0x004002d4	0xc3002b4	sw	\$t5,resultado		\$b	181	
0x004002d8	0xc3002b8	sw	\$t5,resultado		\$c	182	
0x004002dc	0xc3002bc	sw	\$t5,resultado		\$d	183	
0x004002e0	0xc3002c0	sw	\$t5,resultado		\$e	184	
0x004002e4	0xc3002c4	sw	\$t5,resultado		\$f	185	
0x004002e8	0xc3002c8	sw	\$t5,resultado		\$0	186	
0x004002ec	0xc3002cc	sw	\$t5,resultado		\$1	187	

Execução - 19

Text Segment

Offset	Address	Code	Basic	Source
0x00400000	0x00400000	0x00400000	0x00400000	15: lw \$a0, A
0x00400004	0x00400004	0x00400004	0x00400004	16: lw \$a1, B
0x00400008	0x00400008	0x00400008	0x00400008	17: add \$a0, \$zero, \$zero
0x0040000c	0x0040000c	0x0040000c	0x0040000c	18: beq \$t0, \$zero, done0
0x00400010	0x00400010	0x00400010	0x00400010	19: add \$t0, \$a0, \$a0
0x00400014	0x00400014	0x00400014	0x00400014	20: add \$t0, \$a0, \$a0
0x00400018	0x00400018	0x00400018	0x00400018	21: add \$t0, \$a0, \$a0
0x0040001c	0x0040001c	0x0040001c	0x0040001c	22: add \$t0, \$t0, \$t0
0x00400020	0x00400020	0x00400020	0x00400020	23: j count0
0x00400024	0x00400024	0x00400024	0x00400024	24: add \$t1, \$zero, \$zero
0x00400028	0x00400028	0x00400028	0x00400028	25: add \$t0, \$t0, \$zero
0x0040002c	0x0040002c	0x0040002c	0x0040002c	26: add \$t0, \$t0, \$zero

Text Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x00100000	0x00000002	0x00000004	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00100020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00100040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00100060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00100080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x001000a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x001000c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x001000e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00100100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00100120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00100140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00100160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages

Run IO

program is finished running (dropped off bottom)

program is finished running (dropped off bottom)

program is finished running (dropped off bottom)

Clear

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000002
\$v0	2	0x00000000
\$t0	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$s0	6	0x00000000
\$s1	7	0x00000000
\$s2	8	0x00000000
\$s3	9	0x00000000
\$s4	10	0x00000000
\$s5	11	0x00000000
\$s6	12	0xffffffff
\$s7	13	0x00000000
\$s8	14	0x00000000
\$s9	15	0x00000000
\$t1	17	0x00000004
\$t2	18	0x00000008
\$t3	19	0x00000000
\$t4	20	0x00000000
\$t5	21	0x00000000
\$t6	22	0x00000000
\$t7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$t0	26	0x00000000
\$t1	27	0x00000000
\$t2	28	0x00000000
\$t3	29	0xffffffff
\$t4	30	0x00000000
\$t5	31	0x00000000
\$t6	32	0x00000000
\$t7	33	0x00000000
\$t8	34	0x00000000
\$t9	35	0x00000000

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.