



**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, $a2, $a5 # Atribuição -> a1 = x - y
```

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

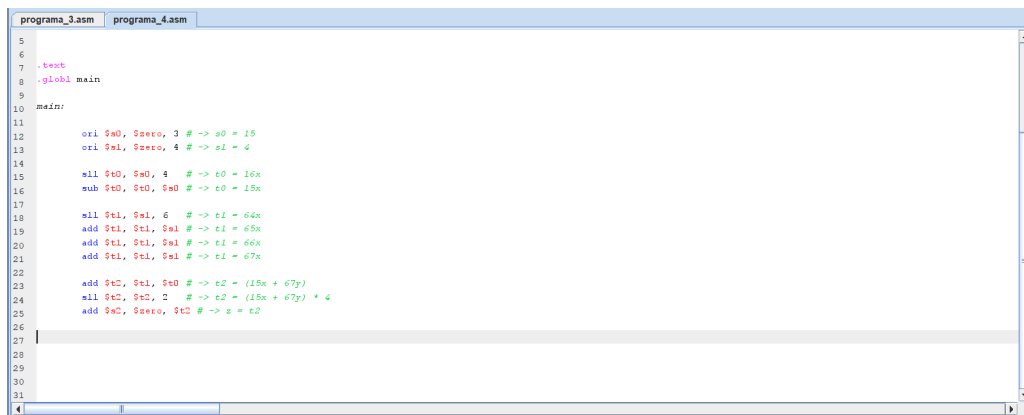
Programa - 03

```

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 = 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

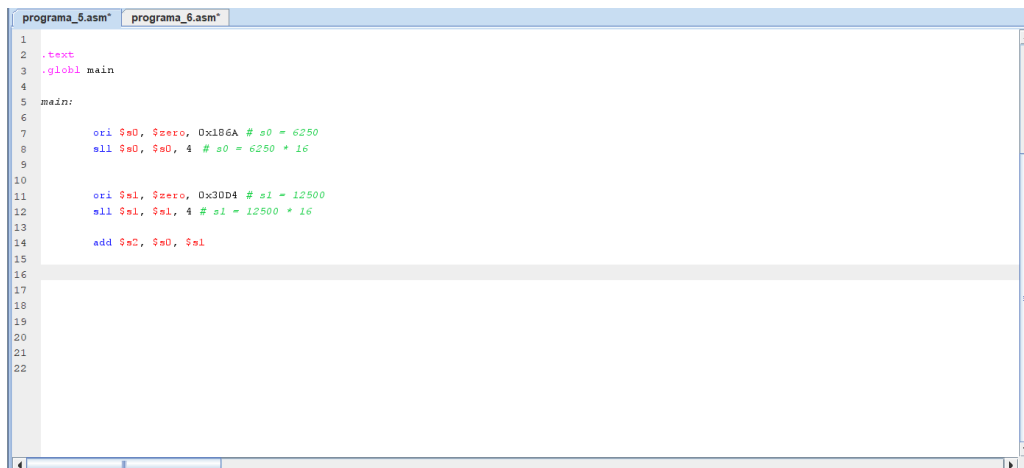
Programa - 04



```
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

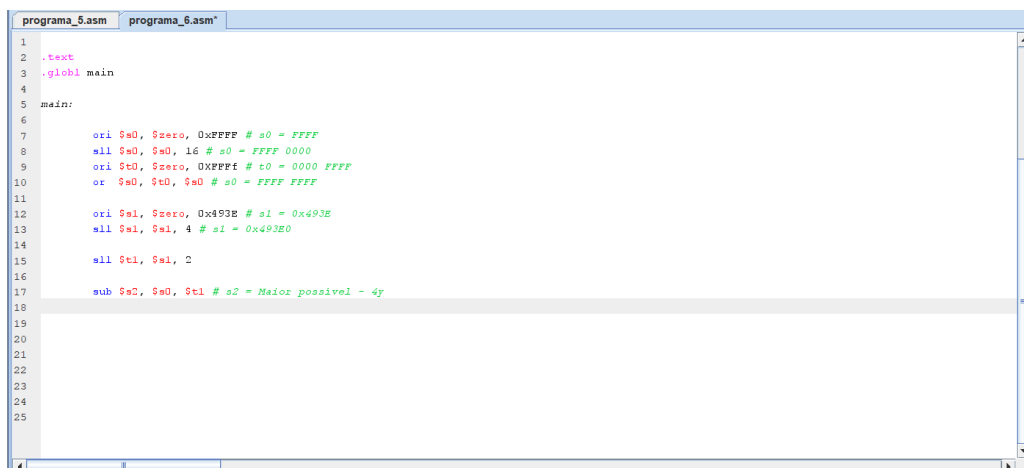
Programa - 05



```
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

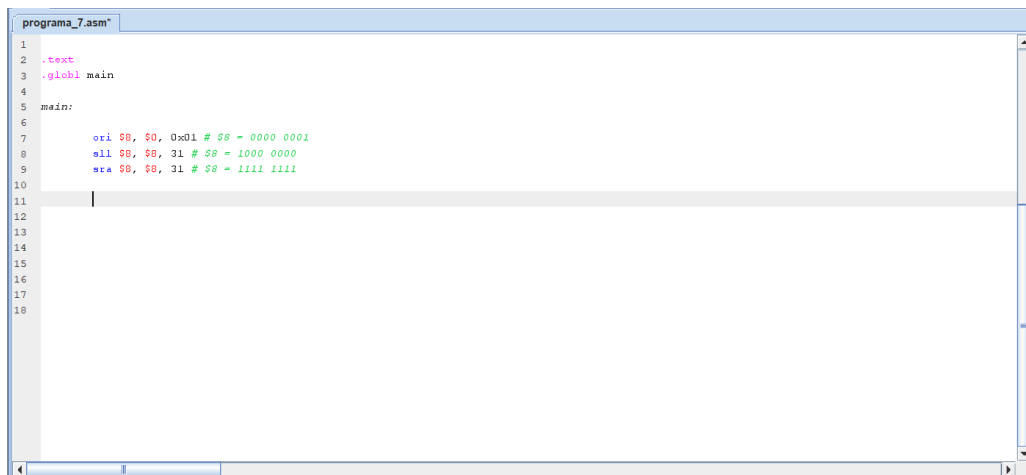
Programa - 06



```
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



```
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
```

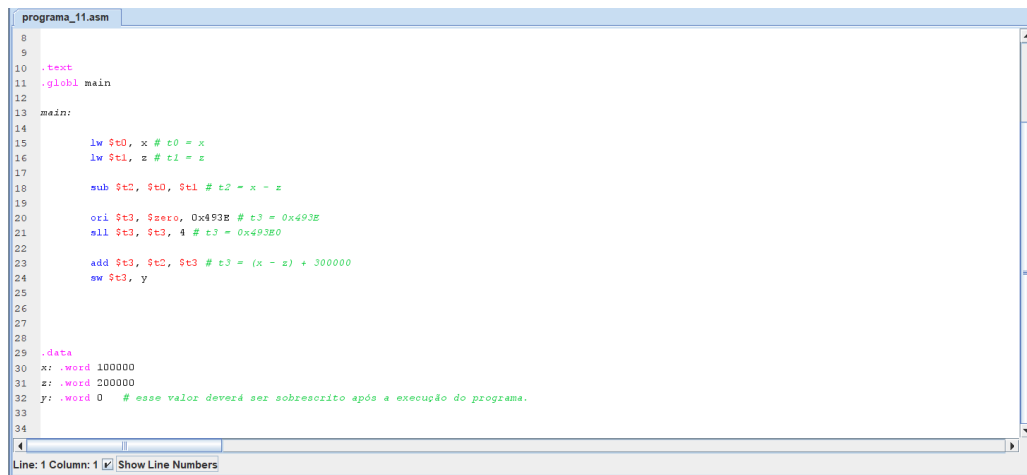
Figura 9: Imagem do programa 9

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

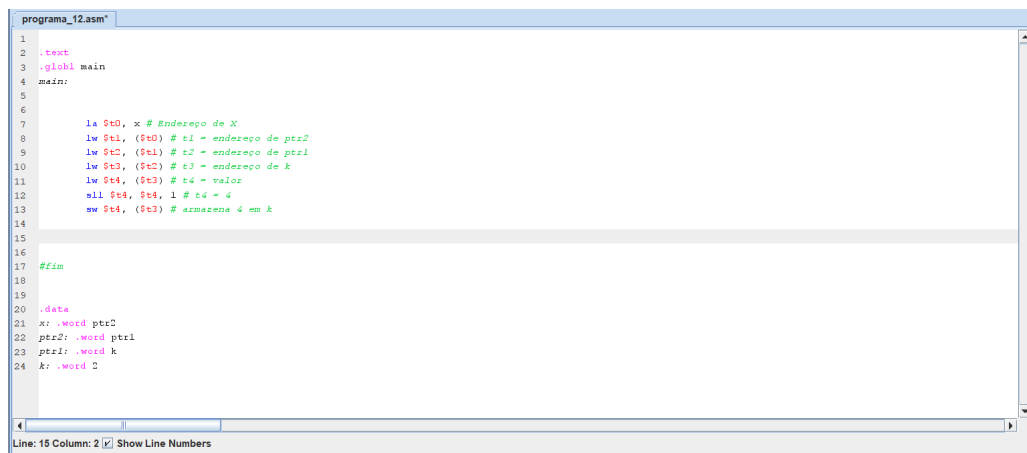
Programa - 11



```
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
1
2 .text
3 .globl main
4 main:
5
6
7     la $t0, x # Endereço de X
8     lw $t1, ($t0) # t1 = endereço de ptr2
9     lw $t2, ($t1) # t2 = endereço de ptr1
10    lw $t3, ($t2) # t3 = endereço de k
11    lw $t4, ($t3) # t4 = valor
12    sll $t4, $t4, 1 # t4 = 4
13    sw $t4, ($t3) # armazena 4 em k
14
15
16
17 #fim
18
19
20 .data
21 x: .word ptr2
22 ptr2: .word ptr1
23 ptr1: .word k
24 k: .word 2
Line: 15 Column: 2 Show Line Numbers
```

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.asm
10
11 .text
12 .globl main
13 main:
14     ori $s0, $zero, 0    # i = 0
15     ori $s1, $zero, 0    # soma = 0
16     ori $s2, $zero, 0    # resultado temporário
17     ori $t1, $zero, 100  # n = 100
18     la $t0, vetor        # t0 = endereço base do vetor
19
20 loop:
21     sll $t2, $s0, 1      # t2 = 2*i
22     addi $s2, $t2, 1     # result = 2*i + 1
23     sw $s2, 0($t0)       # vetor[i] = result
24
25     add $s1, $s1, $s2    # soma += result
26
27     addi $t0, $t0, 4     # próximo endereço
28     addi $s0, $s0, 1     # i++
29     bne $s0, $t1, loop   # repetir enquanto i != 100
30
31     sw $s1, 0($t0)
32
33 .data
34 vetor: .space 400
35 soma: .word 0
36
```

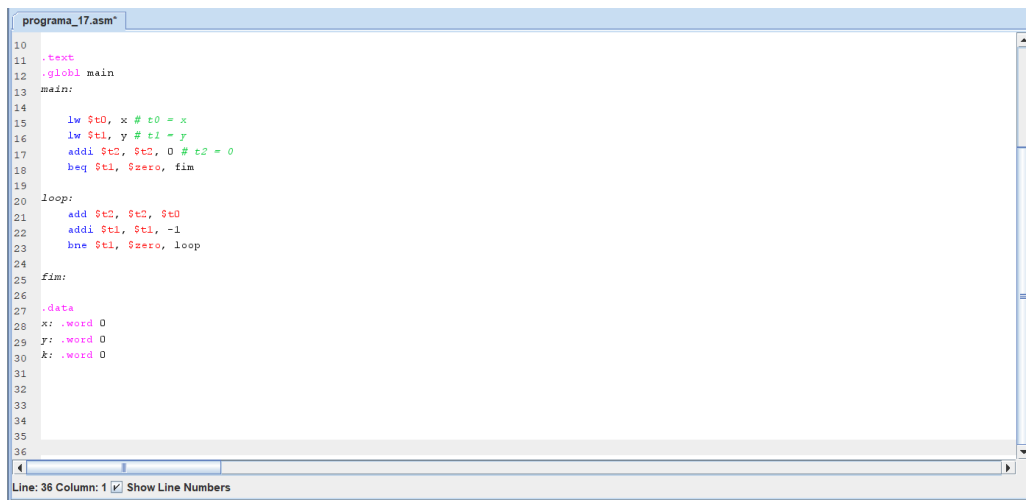
Figura 15: Imagem do programa 15

Programa - 16

```
programa_16.asm
10
11 .text
12 .globl main
13 main:
14
15     lw $t0, x # x
16     lw $t1, y # y
17     lw $t2, z # z
18
19     mult $t0, $t1
20     mflo $t3
21
22     div $t3, $t2
23     mflo $t4
24
25     sw $t4, resultado
26
27 .data
28 x: .word 0x186A00
29 y: .word 0x13980
30 z: .word 0x61A80
31 resultado: .word 0
32
33
34
35
36
```

Figura 16: Imagem do programa 16

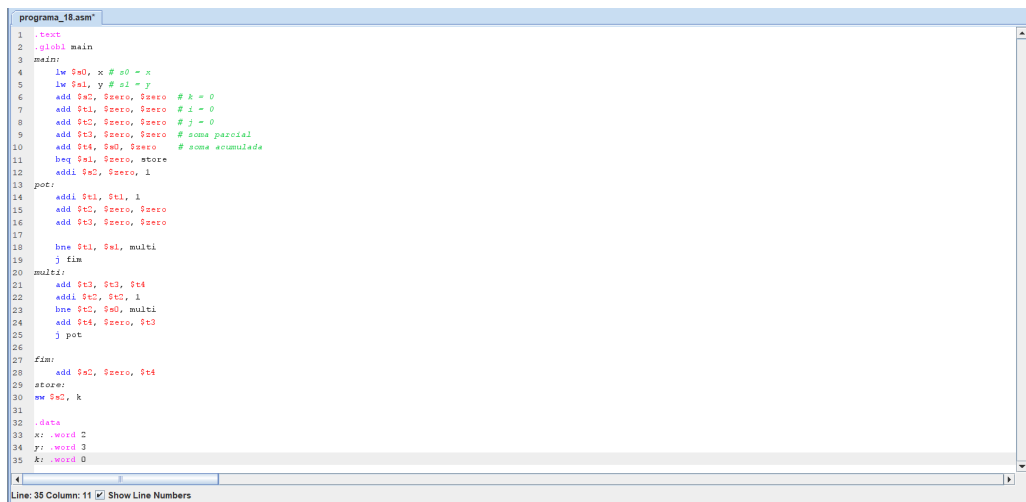
Programa - 17



```
programa_17.asm
10
11 .text
12 .globl main
13 main:
14
15     lw $t0, x # t0 = x
16     lw $t1, y # t1 = y
17     addi $t2, $t2, 0 # t2 = 0
18     beq $t1, $zero, fim
19
20 loop:
21     add $t2, $t2, $t0
22     addi $t1, $t1, -1
23     bne $t1, $zero, loop
24
25 fim:
26
27 .data
28 x: .word 0
29 y: .word 0
30 k: .word 0
31
32
33
34
35
36
Line: 36 Column: 1 ☒ Show Line Numbers
```

Figura 17: Imagem do programa 17

Programa - 18



```
programa_18.asm
1 .text
2 .globl main
3 main:
4
5     lw $a0, x # a0 = x
6     lw $a1, y # a1 = y
7     add $t0, $zero, $zero # k = 0
8     add $t1, $zero, $zero # i = 0
9     add $t2, $zero, $zero # j = 0
10    add $t3, $zero, $zero # some parcial
11    add $t4, $a0, $zero # some acumulada
12    beq $a1, $zero, store
13    addi $a1, $zero, 1
14
15 pot:
16    addi $t1, $t1, 1
17    add $t2, $zero, $zero
18    add $t3, $zero, $zero
19    bne $t1, $a1, multi
20    j fim
21
22 multi:
23    add $t3, $t3, $t4
24    addi $t2, $t2, 1
25    bne $t2, $a0, multi
26    add $t4, $zero, $t3
27    j pot
28
29 fim:
30    add $a2, $zero, $t4
31    store:
32    sw $a2, k
33
34 .data
35 x: .word 2
36 y: .word 3
37 k: .word 0
38
39
40
41
42
43
Line: 35 Column: 11 ☒ Show Line Numbers
```

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

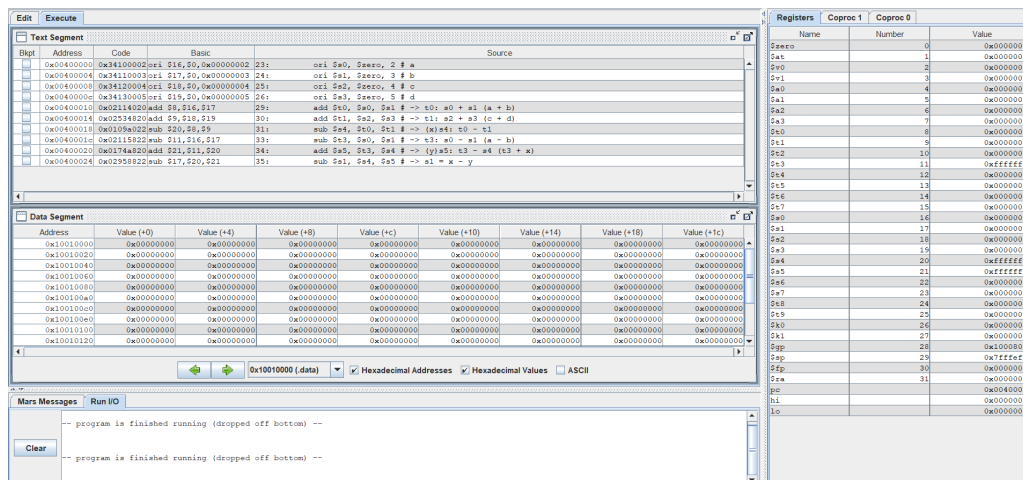


Figura 22: Imagem da execução programa 1

Execução - 02

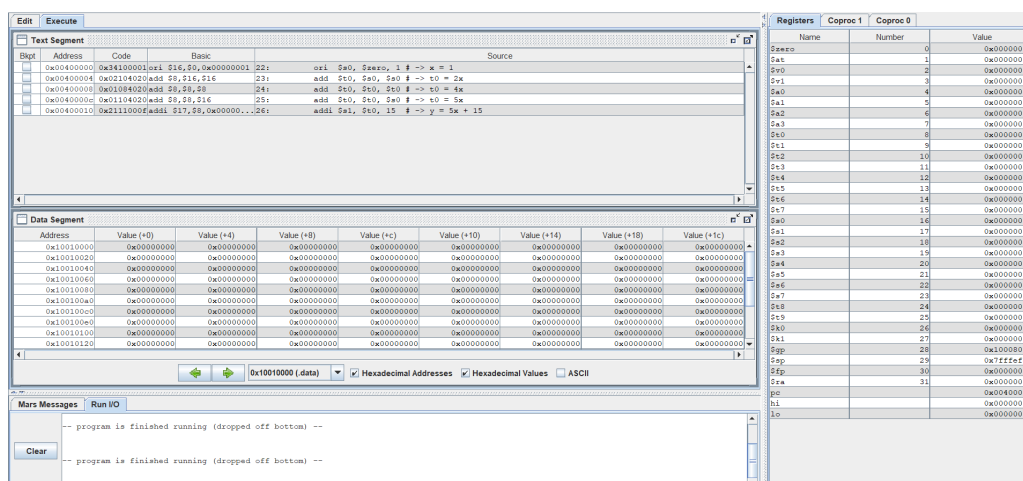


Figura 23: Imagem do execução programa 2

Execução - 03

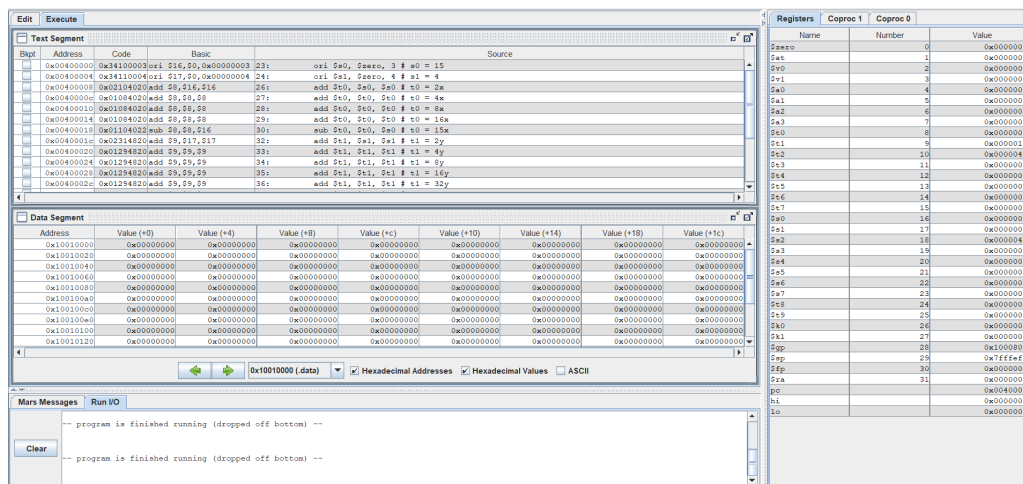


Figura 24: Imagem do execução programa 3

Execução - 04

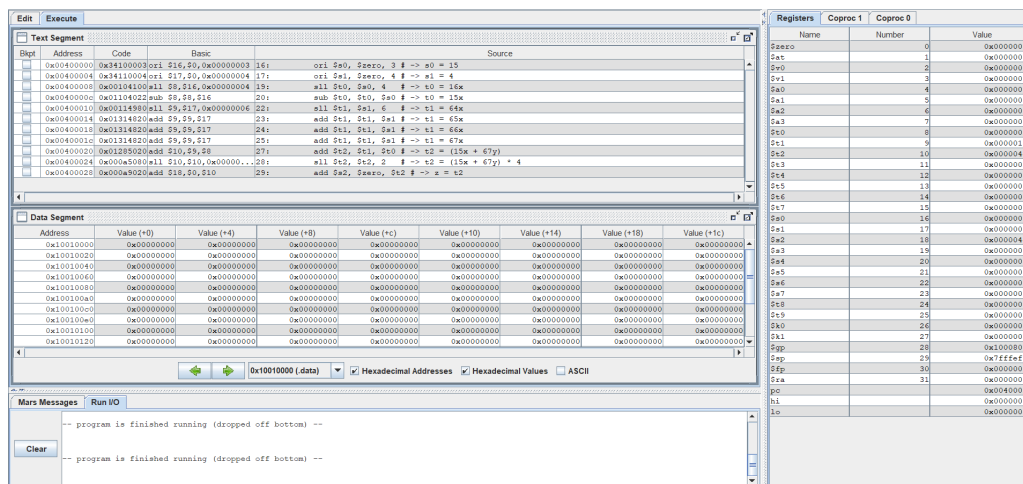


Figura 25: Imagem do execução programa 4

Execução - 05

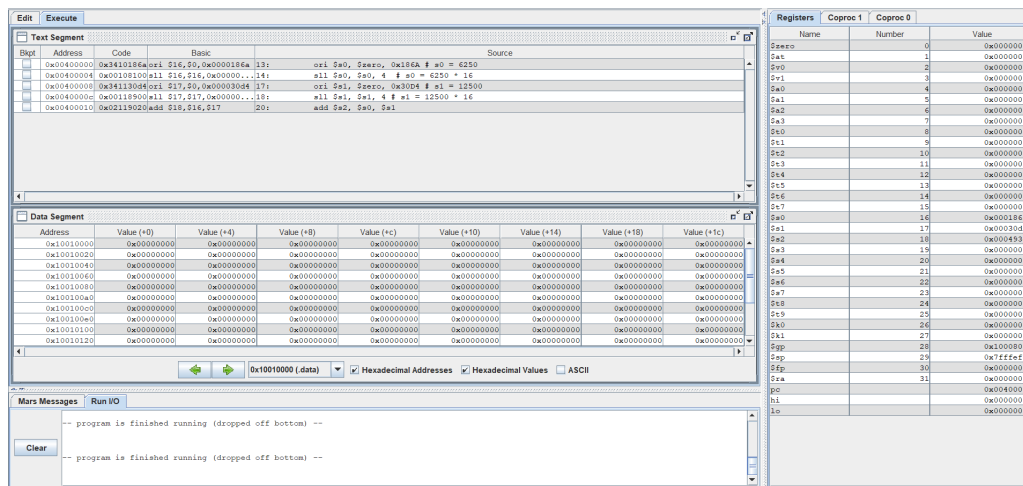


Figura 26: Imagem do execução programa 5

Execução - 06

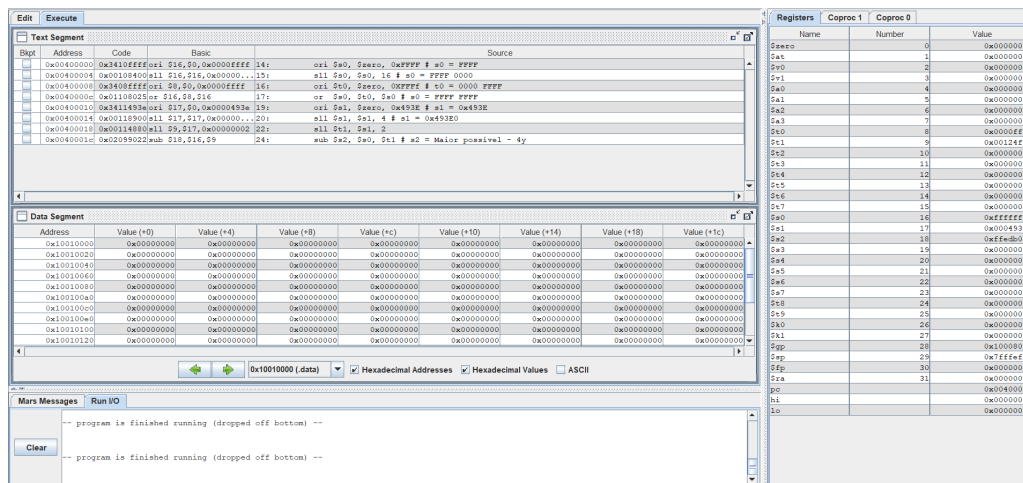


Figura 27: Imagem do execução programa 6

Execução - 07

Text Segment						Name			
Blkt	Address	Code	Basic	Source	Name	Number	Value		
0x00400000	0x34080001	ori	\$9,\$0,0x00000001	14: ori \$9, \$0, 0x01 # \$9 = 0000 0001	\$zero	0	0x00000000		
0x00400004	0x00947c01	addi	\$2,\$9,0x0000001f	15: addi \$2, \$9, 31 # \$2 = 1000 0000	\$a1	1	0x00000000		
0x00400008	0x00947c02	add	\$2,\$9,0x0000001f	16: add \$2, \$9, 31 # \$2 = 1111 1111	\$v0	2	0x00000000		
					\$v1	3	0x00000000		
					\$a0	4	0x00000000		
					\$a1	5	0x00000000		
					\$a2	6	0x00000000		
					\$a3	7	0x00000000		
					\$t0	8	0xffffffff		
					\$t1	9	0x00000000		
					\$t2	10	0x00000000		
					\$t3	11	0x00000000		
					\$t4	12	0x00000000		
					\$t5	13	0x00000000		
					\$t6	14	0x00000000		
					\$t7	15	0x00000000		
					\$t8	16	0x00000000		
					\$t9	17	0x00000000		
					\$a0	18	0x00000000		
					\$a1	19	0x00000000		
					\$a2	20	0x00000000		
					\$a3	21	0x00000000		
					\$a4	22	0x00000000		
					\$a5	23	0x00000000		
					\$a6	24	0x00000000		
					\$a7	25	0x00000000		
					\$a8	26	0x00000000		
					\$a9	27	0x00000000		
					\$a10	28	0x00000000		
					\$a11	29	0x00000000		
					\$a12	30	0x00000000		
					\$a13	31	0x00000000		
					\$a14	32	0x00000000		
					\$a15	33	0x00000000		
					\$a16	34	0x00000000		
					\$a17	35	0x00000000		
					\$a18	36	0x00000000		
					\$a19	37	0x00000000		
					\$a20	38	0x00000000		
					\$a21	39	0x00000000		
					\$a22	40	0x00000000		
					\$a23	41	0x00000000		
					\$a24	42	0x00000000		
					\$a25	43	0x00000000		
					\$a26	44	0x00000000		
					\$a27	45	0x00000000		
					\$a28	46	0x00000000		
					\$a29	47	0x00000000		
					\$a30	48	0x00000000		
					\$a31	49	0x00000000		
					\$a32	50	0x00000000		
					\$a33	51	0x00000000		
					\$a34	52	0x00000000		
					\$a35	53	0x00000000		
					\$a36	54	0x00000000		
					\$a37	55	0x00000000		
					\$a38	56	0x00000000		
					\$a39	57	0x00000000		
					\$a40	58	0x00000000		
					\$a41	59	0x00000000		
					\$a42	60	0x00000000		
					\$a43	61	0x00000000		
					\$a44	62	0x00000000		
					\$a45	63	0x00000000		
					\$a46	64	0x00000000		
					\$a47	65	0x00000000		
					\$a48	66	0x00000000		
					\$a49	67	0x00000000		
					\$a50	68	0x00000000		
					\$a51	69	0x00000000		
					\$a52	70	0x00000000		
					\$a53	71	0x00000000		
					\$a54	72	0x00000000		
					\$a55	73	0x00000000		
					\$a56	74	0x00000000		
					\$a57	75	0x00000000		
					\$a58	76	0x00000000		
					\$a59	77	0x00000000		
					\$a60	78	0x00000000		
					\$a61	79	0x00000000		
					\$a62	80	0x00000000		
					\$a63	81	0x00000000		
					\$a64	82	0x00000000		
					\$a65	83	0x00000000		
					\$a66	84	0x00000000		
					\$a67	85	0x00000000		
					\$a68	86	0x00000000		
					\$a69	87	0x00000000		
					\$a70	88	0x00000000		
					\$a71	89	0x00000000		
					\$a72	90	0x00000000		
					\$a73	91	0x00000000		
					\$a74	92	0x00000000		
					\$a75	93	0x00000000		
					\$a76	94	0x00000000		
					\$a77	95	0x00000000		
					\$a78	96	0x00000000		
					\$a79	97	0x00000000		
					\$a80	98	0x00000000		
					\$a81	99	0x00000000		
					\$a82	100	0x00000000		
					\$a83	101	0x00000000		
					\$a84	102	0x00000000		
					\$a85	103	0x00000000		
					\$a86	104	0x00000000		
					\$a87	105	0x00000000		
					\$a88	106	0x00000000		
					\$a89	107	0x00000000		
					\$a90	108	0x00000000		
					\$a91	109	0x00000000		
					\$a92	110	0x00000000		
					\$a93	111	0x00000000		
					\$a94	112	0x00000000		
					\$a95	113	0x00000000		
					\$a96	114	0x00000000		
					\$a97	115	0x00000000		
					\$a98	116	0x00000000		
					\$a99	117	0x00000000		
					\$a100	118	0x00000000		
					\$a101	119	0x00000000		
					\$a102	120	0x00000000		
					\$a103	121	0x00000000		
					\$a104	122	0x00000000		
					\$a105	123	0x00000000		
					\$a106	124	0x00000000		
					\$a107	125	0x00000000		
					\$a108	126	0x00000000		
					\$a109	127	0x00000000		
					\$a110	128	0x00000000		
					\$a111	129	0x00000000		
					\$a112	130	0x00000000		
					\$a113	131	0x00000000		
					\$a114	132	0x00000000		
					\$a115	133	0x00000000		
					\$a116	134	0x00000000		
					\$a117	135	0x00000000		
					\$a118	136	0x00000000		
					\$a119	137	0x00000000		
					\$a120	138	0x00000000		
					\$a121	139	0x00000000		
					\$a122	140	0x00000000		
					\$a123	141	0x00000000		
					\$a124	142	0x00000000		
					\$a125	143	0x00000000		
					\$a126	144	0x00000000		
					\$a127	145	0x00000000		
					\$a128	146	0x00000000		
					\$a129	147	0x00000000		
					\$a130	148	0x00000000		
					\$a131	149	0x00000000		
					\$a132	150	0x00000000		
					\$a133	151	0x00000000		
					\$a134	152	0x00000000		
					\$a135	153	0x00000000		
					\$a136	154	0x00000000		
					\$a137	155	0x00000000		
					\$a138	156	0x00000000		
					\$a139	157	0x00000000		
					\$a140	158	0x00000000		
					\$a141	159	0x00000000		
					\$a142	160	0x00000000		
					\$a143	161	0x00000000		
					\$a144	162	0x00000000		
					\$a145	163	0x00000000		
					\$a146	164	0x00000000		
					\$a147	165	0x00000000		
					\$a148	166	0x00000000		
					\$a149	167	0x00000000		
					\$a150	168	0x00000000		
					\$a151	169	0x00000000		
					\$a152	170	0x00000000		
					\$a153	171	0x00000000		
					\$a154	172	0x00000000		
					\$a155	173	0x00000000		
					\$a156	174	0x00000000		
					\$a157	175	0x00000000		
					\$a158	176	0x00000000		
					\$a159	177	0x00000000		
					\$a160	178	0x00000000		
					\$a161	179	0x00000000		
					\$a162	180	0x00000000		
					\$a163	181	0x00000000		
					\$a164	182	0x00000000		
					\$a165	183	0x00000000		
					\$a166	184	0x00000000		
					\$a167	185	0x00000000		
					\$a168	186	0x00000000		
					\$a169	187	0x00000000		
					\$a170	188	0x00000000		
					\$a171	189	0x00000000		
					\$a172	190	0x00000000		
					\$a173	191	0x00000000		
					\$a174	192	0x00000000		
					\$a175	193	0x00000000		
					\$a176	194	0x00000000		
					\$a177	195	0x00000000		
					\$a178	196	0x00000000		
					\$a179	197	0x00000000		
					\$a180	198	0x00000000		
					\$a181	199	0x00000000		
					\$a182	200	0x00000000		
					\$a183	201	0x00000000		
					\$a184	202	0x00000000		
					\$a185	203	0x00000000		
					\$a186	204	0x00000000		
					\$a187	205	0x00000000		
					\$a188	206	0x00000000		
					\$a189	207	0x00000000		
					\$a190	208	0x00000000		
					\$a191	209	0x00000000		
					\$a192	210	0x00000000		
					\$a193	211	0x00000000		
					\$a194	212	0x00000000		
					\$a195	213	0x00000000		
					\$a196	214	0x00000000		
					\$a197	215	0x00000000		
					\$a198	216	0x00000000		
					\$a199	217	0x00000000		
					\$a200	218	0x00000000		
					\$a201	219</			

Execução - 10

Text Segment						Name		Number	Value
Offset	Address	Code	Basic	Source					
0x04000000	0x04011001	lui	\$1, 0x000001001	15: lw \$t0, x # t0 = x			Zero	0	0x00000000
0x04000004	0xc2800001	lw	\$9, 0x00000000 (\$1)				\$a0	1	0x10010000
0x04000008	0x04011001	lui	\$1, 0x000001001	16: lw \$t1, x # t1 = y			\$v0	2	0x00000000
0x0400000c	0xc2900041	lw	\$9, 0x00000004 (\$1)				\$v1	3	0x00000000
0x04000010	0x00059300	add	\$11, \$10, \$9, 0x00000007	18: sll \$t0, \$t0, 7 # t2 = 127x			\$a0	4	0x00000000
0x04000014	0xc1495022	mub	\$10, \$10, \$9	19: mub \$t0, \$t0, \$t0 # t2 = 127x			\$a1	5	0x00000000
0x04000018	0x00059300	add	\$11, \$11, \$9, 0x00000006	21: sll \$t3, \$t3, 6 # t3 = 64x			\$a2	6	0x00000000
0x0400001c	0xc1495020	add	\$11, \$11, \$9	22: add \$t3, \$t3, \$t1 # t3 = 64x			\$a3	7	0x00000000
0x04000020	0xc0480022	mub	\$12, \$12, \$11	24: mub \$t4, \$t4, \$t3 # t4 = 127x - 65x			\$t0	8	0x00000005
0x04000024	0xc218c001	addi	\$12, \$12, 0x0000...	25: addi \$t4, \$t4, 1 # t4 = 127x - 65x + 1			\$t1	9	0x00000007
0x04000028	0xc2001001	lui	\$1, 0x000001001	27: sw \$t4, y			\$t2	10	0x00000076
0x0400002c	0xc2000000	sw	\$12, 0x00000000 (\$1)				\$t4	12	0x000000b5
							\$t5	13	0x00000000
							\$t6	14	0x00000000
							\$t7	15	0x00000000
							\$t8	16	0x00000000
							\$t9	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
							\$b0	26	0x00000000
							\$b1	27	0x00000000
							\$gp	28	0x00000000
							\$gp	29	0x7ffffeff
							\$fp	30	0x00000000
							\$ra	31	0x00000000
							\$lo		0x00000000

Figura 31: Imagem do execução programa 10

Execução - 11

Text Segment						Name		Number	Value
Offset	Address	Code	Basic	Source					
0x04000000	0x04011001	lui	\$1, 0x000001001	15: lw \$t0, x # t0 = x			Zero	0	0x00000000
0x04000004	0xc2800001	lw	\$9, 0x00000000 (\$1)				\$a0	1	0x10010000
0x04000008	0x04011001	lui	\$1, 0x000001001	16: lw \$t1, x # t1 = x			\$v0	2	0x00000000
0x0400000c	0xc2900041	lw	\$9, 0x00000004 (\$1)				\$v1	3	0x00000000
0x04000010	0x00059300	add	\$11, \$10, \$9, 0x00000007	18: sll \$t0, \$t0, \$t1 # t2 = x - x			\$a0	4	0x00000000
0x04000014	0xc1495022	mub	\$10, \$10, \$9	19: sll \$t3, \$t3, 4 # t3 = 0x4380			\$a2	6	0x00000000
0x04000018	0x00059300	add	\$11, \$11, \$9, 0x00000006	21: add \$t3, \$t3, \$t1 # t3 = 0x4380 + 300000			\$t0	8	0x00018640
0x0400001c	0xc1495020	add	\$11, \$11, \$9	22: sw \$t3, y			\$t1	9	0x00033440
0x04000020	0xc0480022	mub	\$12, \$12, \$11				\$t2	10	0x7ffff640
0x04000024	0xc218c001	addi	\$12, \$12, 0x0000...				\$t3	11	0x20020018
0x04000028	0xc2001001	lui	\$1, 0x000001001				\$t4	12	0x00000000
0x0400002c	0xc2000000	sw	\$12, 0x00000000 (\$1)				\$t5	13	0x00000000
							\$t6	14	0x00000000
							\$t7	15	0x00000000
							\$t8	16	0x00000000
							\$t9	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
							\$b0	26	0x00000000
							\$b1	27	0x00000000
							\$gp	28	0x00000000
							\$gp	29	0x7ffffeff
							\$fp	30	0x00000000
							\$ra	31	0x00000000
							\$lo		0x00000028
							\$hi		0x00000000
							\$lo		0x00000000

Figura 32: Imagem do execução programa 11

Execução - 12

Text Segment						Name		Number	Value
Offset	Address	Code	Basic	Source					
0x04000000	0x04011001	lui	\$1, 0x000001001	7: la \$t0, x # Endereco de x			Zero	0	0x00000000
0x04000004	0xc1428000	ori	\$9, \$1, 0x00000000				\$a0	1	0x10010000
0x04000008	0x00059300	add	\$11, \$10, \$9, 0x00000007	8: lw \$t1, (\$t0) # t1 = endereco de ptr2			\$v0	2	0x00000000
0x0400000c	0xc2d2a000	lw	\$10, 0x00000000 (\$9)	9: lw \$t2, (\$t1) # t2 = endereco de ptr1			\$v1	3	0x00000000
0x04000010	0xc1484000	lw	\$11, 0x00000000 (\$...	10: lw \$t3, (\$t2) # t3 = endereco de k			\$a0	4	0x00000000
0x04000014	0xc1484000	lw	\$11, 0x00000000 (\$...	11: lw \$t4, (\$t3) # t4 = valor			\$a1	5	0x00000000
0x04000018	0x00060000	add	\$11, \$12, 0x0000...	12: sll \$t4, \$t4, 1 # t4 = 4			\$a2	6	0x00000000
0x0400001c	0xc2d2a000	sw	\$12, 0x00000000 (\$...	13: sw \$t4, (\$t3) # armazena 4 em k			\$t0	8	0x10010000
							\$t1	9	0x10010004
							\$t2	10	0x10010008
							\$t3	11	0x20020018
							\$t4	12	0x00000004
							\$t5	13	0x00000000
							\$t6	14	0x00000000
							\$t7	15	0x00000000
							\$t8	16	0x00000000
							\$t9	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
							\$b0	26	0x00000000
							\$b1	27	0x00000000
							\$gp	28	0x00000000
							\$gp	29	0x7ffffeff
							\$fp	30	0x00000000
							\$ra	31	0x00000000
							\$lo		0x00000000
							\$hi		0x00000000
							\$lo		0x00000000

Figura 33: Imagem do execução programa 12

Execução - 13

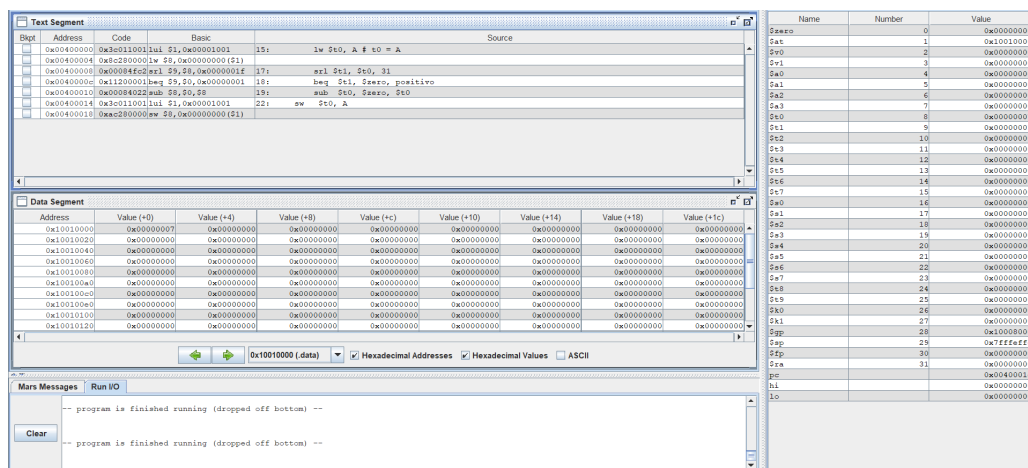


Figura 34: Imagem do execução programa 13

Execução - 14

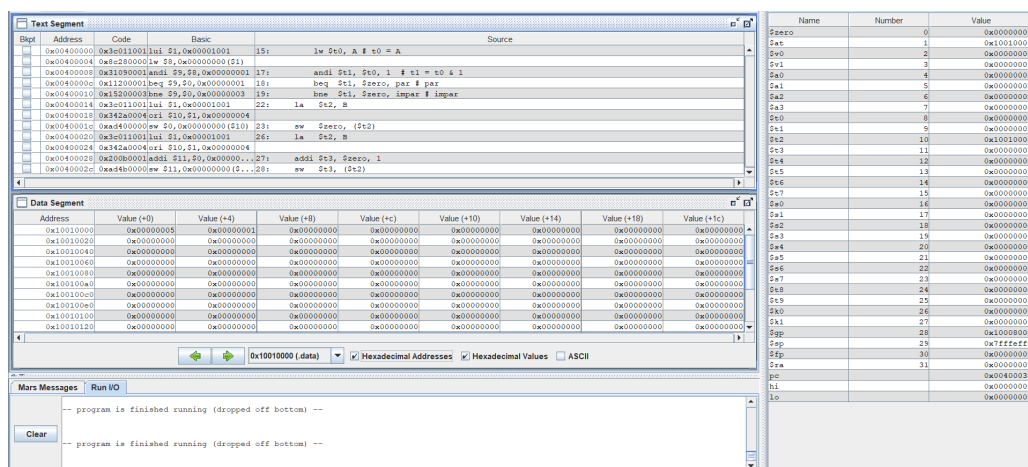


Figura 35: Imagem do execução programa 14

Execução - 15

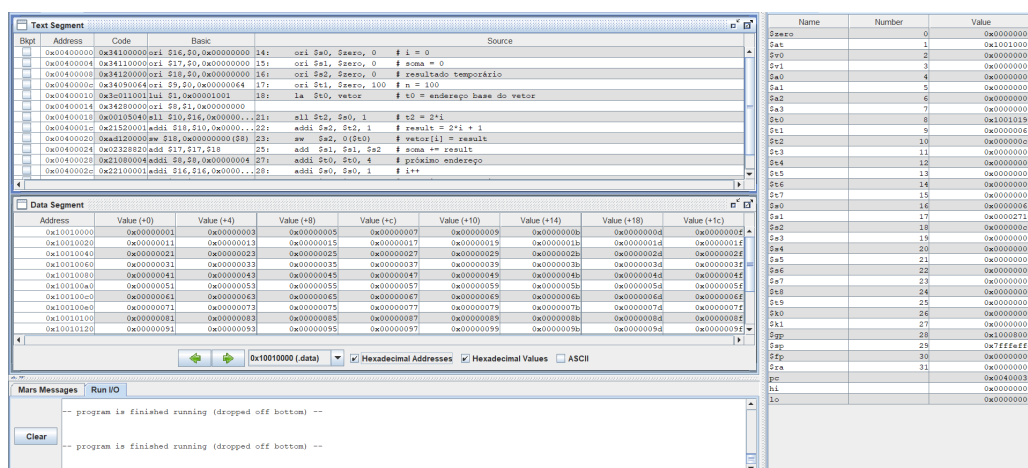


Figura 36: Imagem do execução programa 15

Execução - 16

Text Segment						Name		Number	Value
Bit	Address	Code	Basic	Source					
	0x00400000	0x00100101	lui \$1,0x00001001	15: lw \$t0, x # x			\$zero	0	0x00000000
	0x00400004	0xc2300001	lw \$t0,0x00000000(\$t1)				\$at	1	0x10101000
	0x00400008	0x00100101	lui \$1,0x00001001	16: lw \$t1, y # y			\$v0	2	0x00000000
	0x0040000c	0xc2300004	lw \$t0,0x00000004(\$t1)				\$v1	3	0x00000000
	0x00400010	0x00100101	lui \$1,0x00001001	17: lw \$t2, z # z			\$a0	4	0x00000000
	0x00400014	0xc2300001	lw \$t0,0x00000000(\$t1)				\$a1	5	0x00000000
	0x00400018	0x01090018	mult \$t0, \$t1				\$a2	6	0x00000000
	0x0040001c	0x00000012	mflo \$t1				\$a3	7	0x00000000
	0x00400020	0x01400018	div \$t1, \$t2				\$t0	8	0x00100000
	0x00400024	0x00000012	mflo \$t2				\$t1	9	0x00100000
	0x00400028	0x00000012	mflo \$t4				\$t2	10	0x00000000
	0x0040002c	0x00000012	mflo \$t4				\$t3	11	0x00000000
	0x00400030	0x00000012	mflo \$t4				\$t4	12	0xfffffff6
	0x00400034	0x00000012	mflo \$t4				\$t5	13	0x00000000
	0x00400038	0x00000012	mflo \$t4				\$t6	14	0x00000000
	0x0040003c	0x00000012	mflo \$t4				\$t7	15	0x00000000
	0x00400040	0x00000012	mflo \$t4				\$t8	16	0x00000000
	0x00400044	0x00000012	mflo \$t4				\$t9	17	0x00000000
	0x00400048	0x00000012	mflo \$t4				\$a4	18	0x00000000
	0x0040004c	0x00000012	mflo \$t4				\$a5	19	0x00000000
	0x00400050	0x00000012	mflo \$t4				\$a6	20	0x00000000
	0x00400054	0x00000012	mflo \$t4				\$a7	21	0x00000000
	0x00400058	0x00000012	mflo \$t4				\$a8	22	0x00000000
	0x0040005c	0x00000012	mflo \$t4				\$a9	23	0x00000000
	0x00400060	0x00000012	mflo \$t4				\$t0	24	0x00000000
	0x00400064	0x00000012	mflo \$t4				\$t1	25	0x00000000
	0x00400068	0x00000012	mflo \$t4				\$t2	26	0x00000000
	0x0040006c	0x00000012	mflo \$t4				\$t3	27	0x00000000
	0x00400070	0x00000012	mflo \$t4				\$t4	28	0x10000000
	0x00400074	0x00000012	mflo \$t4				\$t5	29	0xfffffff6
	0x00400078	0x00000012	mflo \$t4				\$t6	30	0x00000000
	0x0040007c	0x00000012	mflo \$t4				\$t7	31	0x00000000
	0x00400080	0x00000012	mflo \$t4				\$t8	32	0x00000000
	0x00400084	0x00000012	mflo \$t4				\$t9	33	0x00000000
	0x00400088	0x00000012	mflo \$t4				\$a0	34	0x00000000
	0x0040008c	0x00000012	mflo \$t4				\$a1	35	0x00000000
	0x00400090	0x00000012	mflo \$t4				\$a2	36	0x00000000
	0x00400094	0x00000012	mflo \$t4				\$a3	37	0x00000000
	0x00400098	0x00000012	mflo \$t4				\$a4	38	0x00000000
	0x0040009c	0x00000012	mflo \$t4				\$a5	39	0x00000000
	0x004000a0	0x00000012	mflo \$t4				\$a6	40	0x00000000
	0x004000a4	0x00000012	mflo \$t4				\$a7	41	0x00000000
	0x004000a8	0x00000012	mflo \$t4				\$a8	42	0x00000000
	0x004000ac	0x00000012	mflo \$t4				\$a9	43	0x00000000
	0x004000b0	0x00000012	mflo \$t4				\$t0	44	0x00000000
	0x004000b4	0x00000012	mflo \$t4				\$t1	45	0x00000000
	0x004000b8	0x00000012	mflo \$t4				\$t2	46	0x00000000
	0x004000bc	0x00000012	mflo \$t4				\$t3	47	0x00000000
	0x004000c0	0x00000012	mflo \$t4				\$t4	48	0x00000000
	0x004000c4	0x00000012	mflo \$t4				\$t5	49	0x00000000
	0x004000c8	0x00000012	mflo \$t4				\$t6	50	0x00000000
	0x004000cc	0x00000012	mflo \$t4				\$t7	51	0x00000000
	0x004000d0	0x00000012	mflo \$t4				\$t8	52	0x00000000
	0x004000d4	0x00000012	mflo \$t4				\$t9	53	0x00000000
	0x004000d8	0x00000012	mflo \$t4				\$a0	54	0x00000000
	0x004000dc	0x00000012	mflo \$t4				\$a1	55	0x00000000
	0x004000e0	0x00000012	mflo \$t4				\$a2	56	0x00000000
	0x004000e4	0x00000012	mflo \$t4				\$a3	57	0x00000000
	0x004000e8	0x00000012	mflo \$t4				\$a4	58	0x00000000
	0x004000ec	0x00000012	mflo \$t4				\$a5	59	0x00000000
	0x004000f0	0x00000012	mflo \$t4				\$a6	60	0x00000000
	0x004000f4	0x00000012	mflo \$t4				\$a7	61	0x00000000
	0x004000f8	0x00000012	mflo \$t4				\$a8	62	0x00000000
	0x004000fc	0x00000012	mflo \$t4				\$a9	63	0x00000000
	0x00400100	0x00000012	mflo \$t4				\$t0	64	0x00000000
	0x00400104	0x00000012	mflo \$t4				\$t1	65	0x00000000
	0x00400108	0x00000012	mflo \$t4				\$t2	66	0x00000000
	0x0040010c	0x00000012	mflo \$t4				\$t3	67	0x00000000
	0x00400110	0x00000012	mflo \$t4				\$t4	68	0x00000000
	0x00400114	0x00000012	mflo \$t4				\$t5	69	0x00000000
	0x00400118	0x00000012	mflo \$t4				\$t6	70	0x00000000
	0x0040011c	0x00000012	mflo \$t4				\$t7	71	0x00000000
	0x00400120	0x00000012	mflo \$t4				\$t8	72	0x00000000
	0x00400124	0x00000012	mflo \$t4				\$t9	73	0x00000000
	0x00400128	0x00000012	mflo \$t4				\$a0	74	0x00000000
	0x0040012c	0x00000012	mflo \$t4				\$a1	75	0x00000000
	0x00400130	0x00000012	mflo \$t4				\$a2	76	0x00000000
	0x00400134	0x00000012	mflo \$t4				\$a3	77	0x00000000
	0x00400138	0x00000012	mflo \$t4				\$a4	78	0x00000000
	0x0040013c	0x00000012	mflo \$t4				\$a5	79	0x00000000
	0x00400140	0x00000012	mflo \$t4				\$a6	80	0x00000000
	0x00400144	0x00000012	mflo \$t4				\$a7	81	0x00000000
	0x00400148	0x00000012	mflo \$t4				\$a8	82	0x00000000
	0x0040014c	0x00000012	mflo \$t4				\$a9	83	0x00000000
	0x00400150	0x00000012	mflo \$t4				\$t0	84	0x00000000
	0x00400154	0x00000012	mflo \$t4				\$t1	85	0x00000000
	0x00400158	0x00000012	mflo \$t4				\$t2	86	0x00000000
	0x0040015c	0x00000012	mflo \$t4				\$t3	87	0x00000000
	0x00400160	0x00000012	mflo \$t4				\$t4	88	0x00000000
	0x00400164	0x00000012	mflo \$t4				\$t5	89	0x00000000
	0x00400168	0x00000012	mflo \$t4				\$t6	90	0x00000000
	0x0040016c	0x00000012	mflo \$t4				\$t7	91	0x00000000
	0x00400170	0x00000012	mflo \$t4				\$t8	92	0x00000000
	0x00400174	0x00000012	mflo \$t4				\$t9	93	0x00000000
	0x00400178	0x00000012	mflo \$t4				\$a0	94	0x00000000
	0x0040017c	0x00000012	mflo \$t4				\$a1	95	0x00000000
	0x00400180	0x00000012	mflo \$t4				\$a2	96	0x00000000
	0x00400184	0x00000012	mflo \$t4				\$a3	97	0x00000000
	0x00400188	0x00000012	mflo \$t4				\$a4	98	0x00000000
	0x0040018c	0x00000012	mflo \$t4				\$a5	99	0x00000000
	0x00400190	0x00000012	mflo \$t4				\$a6	100	0x00000000
	0x00400194	0x00000012	mflo \$t4				\$a7	101	0x00000000
	0x00400198	0x00000012	mflo \$t4				\$a8	102	0x00000000
	0x0040019c	0x00000012	mflo \$t4				\$a9	103	0x00000000
	0x004001a0	0x00000012	mflo \$t4				\$t0	104	0x00000000
	0x004001a4	0x00000012	mflo \$t4				\$t1	105	0x00000000
	0x004001a8	0x00000012	mflo \$t4				\$t2	106	0x00000000
	0x004001ac	0x00000012	mflo \$t4				\$t3	107	0x00000000
	0x004001b0	0x00000012	mflo \$t4				\$t4	108	0x00000000
	0x004001b4	0x00000012	mflo \$t4				\$t5	109	0x00000000
	0x004001b8	0x00000012	mflo \$t4				\$t6	110	0x00000000
	0x004001bc	0x00000012	mflo \$t4				\$t7	111	0x00000000
	0x004001c0	0x00000012	mflo \$t4				\$t8	112	0x00000000
	0x004001c4	0x00000012	mflo \$t4				\$t9	113	0x00000000
	0x004001c8	0x00000012	mflo \$t4				\$a0	114	0x00000000
	0x004001cc	0x00000012	mflo \$t4				\$a1	115	0x00000000
	0x004001d0	0x00000012	mflo \$t4				\$a2	116	0x00000000
	0x004001d4	0x00000012	mflo \$t4				\$a3	117	0x00000000
	0x004001d8	0x00000012	mflo \$t4				\$a4	118	0x00000000
	0x004001dc	0x00000012	mflo \$t4				\$a5	119	0x00000000
	0x004001e0	0x00000012	mflo \$t4				\$a6	120	0x00000000
	0x004001e4	0x00000012	mflo \$t4				\$a7	121	0x00000000
	0x004001e8	0x00000012	mflo \$t4				\$a8	122	0x00000000
	0x004001ec	0x00000012	mflo \$t4				\$a9	123	0x00000000
	0x004001f0	0x00000012	mflo \$t4				\$t0	124	0x00000000
	0x004001f4	0x00000012	mflo \$t4				\$t1	125	0x00000000
	0x004001f8	0x00000012	mflo \$t4				\$t2	126	0x00000000
	0x004001fc	0x00000012	mflo \$t4				\$t3	127	0x00000000
	0x00400200	0x00000012	mflo \$t4				\$t4	128	0x00000000
	0x00400204	0x00000012	mflo \$t4				\$t5	129	0x00000000
	0x00400208	0x00000012	mflo \$t4				\$t6	130	0x00000000
	0x0040020c	0x00000012	mflo \$t4				\$t7	131	0x00000000
	0x00400210	0x00000012	mflo \$t4				\$t8	132	0x00000000
	0x00400214	0x00000012	mflo \$t4				\$t9	133	0x00000000
	0x00400218	0x00000012	mflo \$t4				\$a0	134	0x00000000
	0x0040021c	0x00000012	mflo \$t4				\$a1	135	0x00000000
	0x00400220	0x00000012	mflo \$t4				\$a2	136	0x00000000
	0x00400224	0x00000012	mflo \$t4						

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, \$a0, \$a0
0x00400020	0x00400020	0x00400020	0x00400020	23: add \$t0, \$a0, \$a0
0x00400024	0x00400024	0x00400024	0x00400024	24: add \$t0, \$a0, \$a0
0x00400028	0x00400028	0x00400028	0x00400028	25: add \$t0, \$a0, \$a0
0x0040002c	0x0040002c	0x0040002c	0x0040002c	26: add \$t0, \$a0, \$a0

Registers

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000002
\$v0	2	0x00000000
\$t0	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000002
\$t1	9	0x00000003
\$t2	10	0x00000004
\$t3	11	0x00000005
\$t4	12	0xffffffff
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$a1	16	0x00000004
\$a1	17	0x00000004
\$a2	18	0x00000008
\$a3	19	0x00000000
\$a4	20	0x00000000
\$a5	21	0x00000000
\$a6	22	0x00000000
\$a7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$t0	26	0x00000000
\$t1	27	0x00000000
\$t2	28	0x00000000
\$t3	29	0x00000000
\$t4	30	0x00000000
\$t5	31	0x00000000
\$t6	32	0x00000000
\$t7	33	0x00000000
\$t8	34	0x00000000
\$t9	35	0x00000000
\$t0	36	0x00000000
\$t1	37	0x00000000
\$t2	38	0x00000000
\$t3	39	0x00000000
\$t4	40	0x00000000
\$t5	41	0x00000000
\$t6	42	0x00000000
\$t7	43	0x00000000
\$t8	44	0x00000000
\$t9	45	0x00000000
\$t0	46	0x00000000
\$t1	47	0x00000000
\$t2	48	0x00000000
\$t3	49	0x00000000
\$t4	50	0x00000000
\$t5	51	0x00000000
\$t6	52	0x00000000
\$t7	53	0x00000000
\$t8	54	0x00000000
\$t9	55	0x00000000
\$t0	56	0x00000000
\$t1	57	0x00000000
\$t2	58	0x00000000
\$t3	59	0x00000000
\$t4	60	0x00000000
\$t5	61	0x00000000
\$t6	62	0x00000000
\$t7	63	0x00000000
\$t8	64	0x00000000
\$t9	65	0x00000000
\$t0	66	0x00000000
\$t1	67	0x00000000
\$t2	68	0x00000000
\$t3	69	0x00000000
\$t4	70	0x00000000
\$t5	71	0x00000000
\$t6	72	0x00000000
\$t7	73	0x00000000
\$t8	74	0x00000000
\$t9	75	0x00000000
\$t0	76	0x00000000
\$t1	77	0x00000000
\$t2	78	0x00000000
\$t3	79	0x00000000
\$t4	80	0x00000000
\$t5	81	0x00000000
\$t6	82	0x00000000
\$t7	83	0x00000000
\$t8	84	0x00000000
\$t9	85	0x00000000
\$t0	86	0x00000000
\$t1	87	0x00000000
\$t2	88	0x00000000
\$t3	89	0x00000000
\$t4	90	0x00000000
\$t5	91	0x00000000
\$t6	92	0x00000000
\$t7	93	0x00000000
\$t8	94	0x00000000
\$t9	95	0x00000000
\$t0	96	0x00000000
\$t1	97	0x00000000
\$t2	98	0x00000000
\$t3	99	0x00000000
\$t4	100	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

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.