Arthur de Sá Braz de Matos

Ana Fernanda Sousa Cancado

1	\cap	que é um	arauivo	fonte?
Ι.	\circ	que e un	ı aiyuivo	101116:

A. um arquivo de texto que contém instruções de linguagem de programação.

- 2. O que é um registrador?
 - B. uma parte do processador que possui um padrão de bits.
- 3. Qual o caracter que, na linguagem assembly do SPIM, inicia um comentário?

A. #

4. Quantos bits há em cada instrução de máquina MIPS?

C. 32

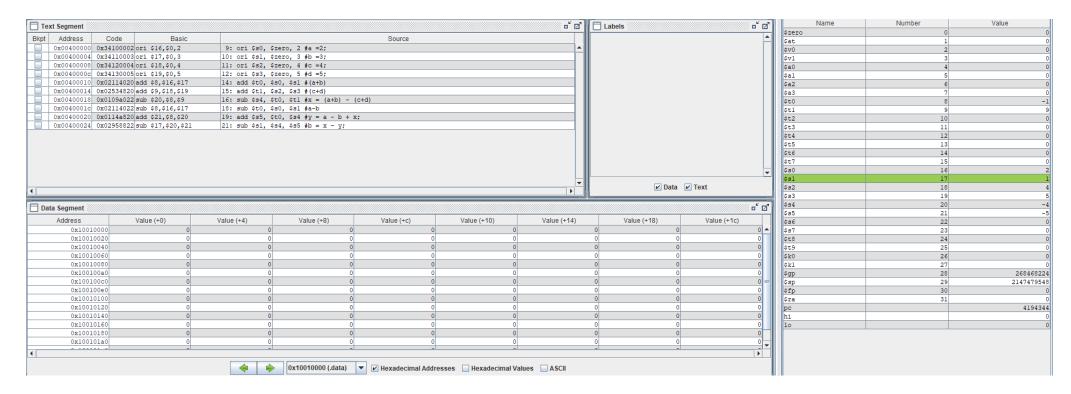
- 5. O que é o contador de programa?
 - D. parte do processador que contém o endereço da próxima instrução de máquina para ser obtida.
- 6. Ao executarmos uma instrução, quanto será adicionado ao contador de programa?

A. 1

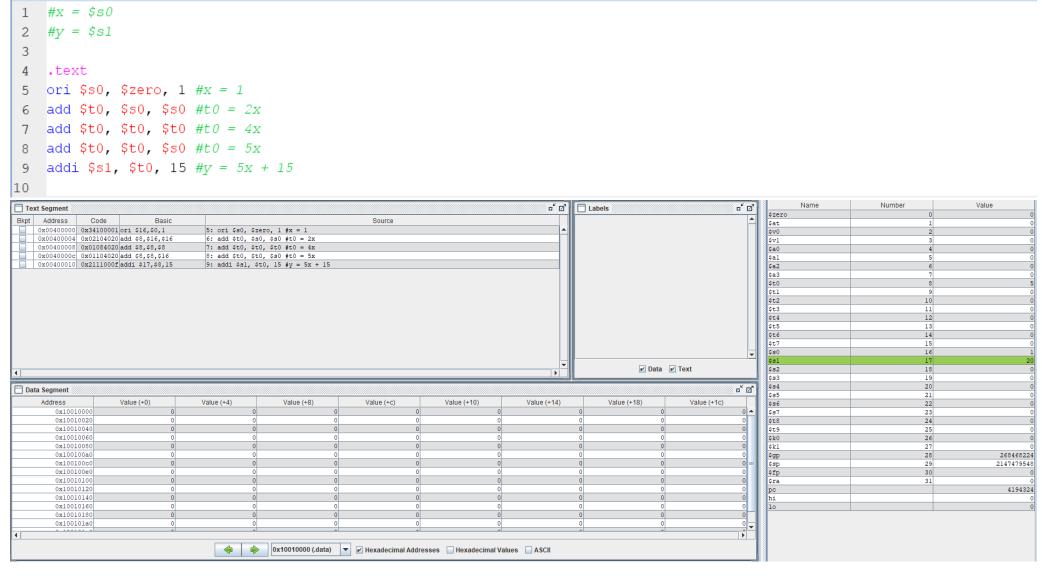
- 7. O que é uma diretiva, tal como a diretiva .text?
 - D. uma declaração que diz o montador algo sobre o que o programador quer, mas não corresponde diretamente a uma instrução de máquina.
- 8. O que é um endereço simbólico?
 - D. um nome usado no código-fonte em linguagem assembly para um local na memória.
- 9. Em qual endereço o simulador SPIM coloca a primeira instrução de máquina quando ele está sendo executado?
 - B. 0x00400000

- 10. Algumas instruções de máquina possuem uma constante como um dos operandos. Como é chamado tal operando?
 - A. operando imediato
- 11. 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 lógica
- 12. Quando uma operação é de fato executada, como estão os operandos na ALU?
 - D. Cada um dos registradores deve possuir 32 bit.
- 13. Dezesseis bits de dados de uma instrução de ori são usados como um operando imediato. Durante execução, o que deve ser feito primeiro?
 - B. Os dados são estendidos em zero à esquerda por 16 bits.
- 14. Qual das instruções seguintes armazenam no registrador \$5 um padrão de bits que representa positivo 48?
 - C. ori \$5,\$0,48
- 15. A instrução de ori pode armazenar o complemento de dois de um número em um registrador?
 - A. Não.
- 16. Qual das instruções seguintes limpa todos os bits no registrador \$8 com exceção do byte de baixa ordem que fica inalterado?
 - D. andi \$8,\$8,0xFF
- 17. Qual é o resultado de um ou exclusivo de padrão sobre ele mesmo?
 - A. Todos os bits em zero.
- 18. Todas as instruções de máquina têm os mesmos campos?
 - A. Não. Diferentes de instruções de máquina possuem campos diferentes.

```
ep6_1.asm
 1 \# a = \$ s 0
 2 	 #b = $s1
 3 \#c = $s2
 4 \# d = \$s3
 5 \# x = \$s4
 6 	 #y = $s5
8 .text
9 ori $s0, $zero, 2 #a =2;
10 ori $s1, $zero, 3 #b =3;
11 ori $s2, $zero, 4 #c =4;
12 ori $s3, $zero, 5 #d =5;
13
14 add $t0, $s0, $s1 #(a+b)
15 add $t1, $s2, $s3 #(c+d)
16 sub $s4, $t0, $t1 \#x = (a+b) - (c+d)
17
18 sub $t0, $s0, $s1 #a-b
19 add $s5, $t0, $s4 #y = a - b + x;
21 sub $s1, $s4, $s5 #b = x - y;
```



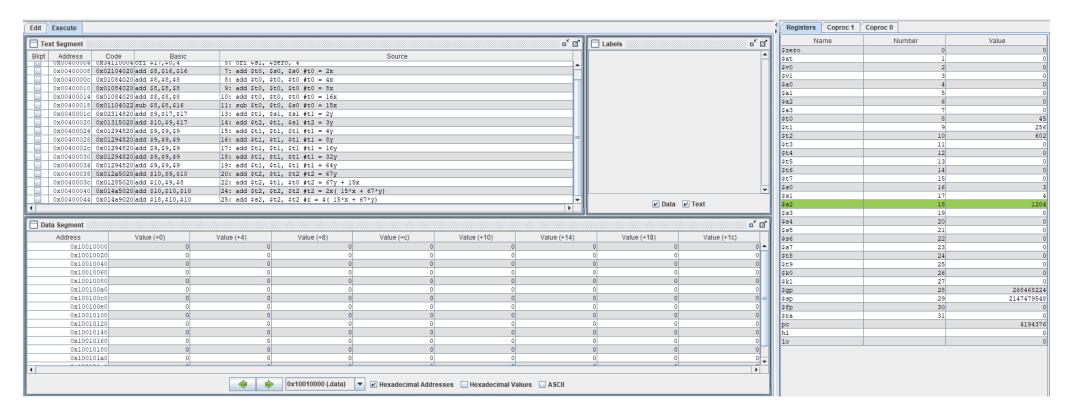
#programa 2



ep6_2.asm*

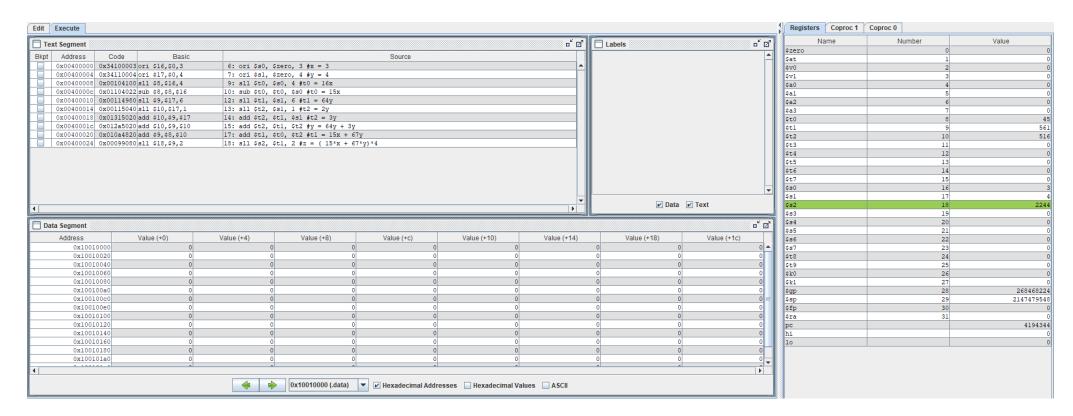
#programa 3

```
ep6_3.asm
 1 \#x = \$s0; y = \$s1; z = \$s2
 2
    .text
 4 ori $s0, $zero, 3
 5 ori $s1, $zero, 4
 7 add $t0, $s0, $s0 #t0 = 2x
 8 add $t0, $t0, $t0 #t0 = 4x
 9 add $t0, $t0, $t0 #t0 = 8x
10 add $t0, $t0, $t0 #t0 = 16x
11 sub $t0, $t0, $s0 #t0 = 15x
12
13 add $t1, $s1, $s1 #t1 = 2y
14 add $t2, $t1, $s1 #t2 = 3y
15 add $t1, $t1, $t1 #t1 = 4y
16 add $t1, $t1, $t1 #t1 = 8y
17 add $t1, $t1, $t1 #t1 = 16y
18 add $t1, $t1, $t1 #t1 = 32y
19 add $t1, $t1, $t1 #t1 = 64y
20 add $t2, $t1, $t2 #t2 = 67y
21
22 add $t2, $t1, $t0 #t2 = 67y + 15x
```



#programa 4

```
ep6_4.asm
 1 \#x = $s0
 2 \# y = \$s1
 3 \#z = $s2
 5 .text
 6 ori $s0, $zero, 3 \#x = 3
 7 ori $s1, $zero, 4 \# y = 4
 9 sll $t0, $s0, 4 #t0 = 16x
10 sub $t0, $t0, $s0 #t0 = 15x
11
12 sll $t1, $s1, 6 #t1 = 64y
13 sll $t2, $s1, 1 #t2 = 2y
14 add $t2, $t1, $s1 #t2 = 3y
15 add $t2, $t1, $t2 #y = 64y + 3y
16
17 add $t1, $t0, $t2 #t1 = 15x + 67y
18 sll $s2, $t1, 2 #z = (15*x + 67*y)*4
```



#programa 5

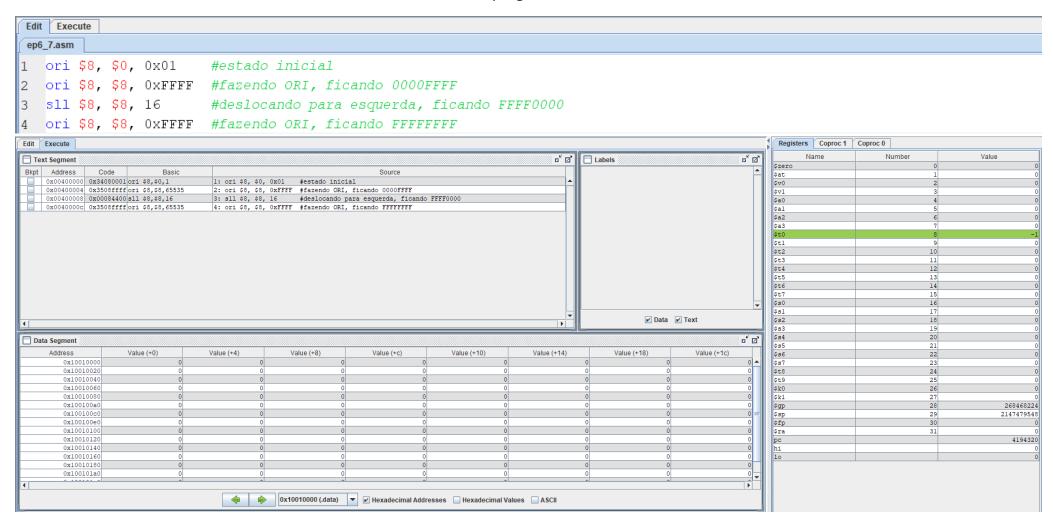
```
ep6_5.asm
 1 \# x = $s0
     #y = $s1
      .text
 5 ori $t0, $zero, 0x186A
      sll $s0, $t0, 4 #x = 100.000
 8 ori $t0, $zero, 0x30D4
 9 sll $s1, $t0, 4 \#y = 200.000
10
11 add $s2, $s1, $s0 \#z = x + y
Edit Execute
                                                                                                                                                    Registers Coproc 1 Coproc 0
                                                                                                                                                         Name
                                                                                                                                                                         Number
                                                                                                                                                                                           Value
Text Segment
                                                                                                        □ Labels
                                                                                                                                            ு் ⊿ி
 Bkpt Address
               Code
                                                                      Source
                                                                                                                                                   $at
    0x00400000 0x3408186a ori $8,$0,6250
                                      5: ori $t0, $zero, 0x186A
                                                                                                                                                   $v0
 0x00400004 0x00088100 sll $16,$8,4
                                      6: sll $s0, $t0, 4 #x = 100.000
                                                                                                                                                   $v1
    0x00400008 0x340830d4 ori $8,$0,12500
                                      8: ori $t0, $zero, 0x30D4
 0x00400008 0x340830d4 ori $8,$0,1250
                                      9: sll $sl, $t0, 4 #y = 200.000
                                                                                                                                                   $al
 0x00400010 0x02309020 add $18,$17,$16
                                     11: add $s2, $s1, $s0 #z = x + y
                                                                                                                                                   $a2
                                                                                                                                                   $a3
                                                                                                                                                   $t0
                                                                                                                                                   $t2
                                                                                                                                                   St.3
                                                                                                                                                                                  11
                                                                                                                                                   $t4
                                                                                                                                                  $t6
                                                                                                                                                   $t7
                                                                                                                                                                                  15
                                                                                                                                                   $80

✓ Data 
✓ Text

                                                                                                                                                   $82
                                                                                                                                                   $83
                                                                                                                                                   $84
Data Segment
                                                                                                                                             Address
                     Value (+0)
                                                                                    Value (+10)
                                                                                                    Value (+14)
                                                                                                                    Value (+18)
                                     Value (+4)
                                                     Value (+8)
                                                                     Value (+c)
                                                                                                                                    Value (+1c)
                                                                                                                                                   $86
        0x10010000
                                                                                                                                                   $87
                                                                                                                                                                                  23
        0x10010020
                                                                                                                                                   $t8
        0x10010040
                                                                                                                                                                                  25
        0x10010060
                                                                                                                                                   $k0
        0x10010080
                                                                                                                                                                                  27
                                                                                                                                                   $k1
        0x100100a0
                                                                                                                                                                                                268468224
        0x100100c0
                                                                                                                                                   $sp
                                                                                                                                                                                               2147479548
                                                                                                                                                   $fp
                                                                                                                                                                                  30
        0x10010100
                                                                                                                                                   $ra
                                                                                                                                                                                                  4194324
        0x10010120
        0x10010140
        0x10010160
        0x10010180
        0x100101a0
```

#programa 6

```
ep6_6.asm
  1 \# x = $s0
      \#z = \$s2
  4
      .text
  6 ori $t0, $zero, 0xFFFF
 7 sll $t0, $t0, 16 #deslocando para a esquerda, ficando 0xFFFF0000
     ori $s0, $t0, 0xFFFF #x = FFFFFFFF
 9
10 ori $t0, $zero, 0x493E
11 sll $s1, $t0, 4 \#y = 300.000
12
     sll $t0, $s1, 2 #t0 = 4y
14
15 sub $s2, $s0, $t0 \#z = x - 4y
Edit Execute
                                                                                                                                                  Registers Coproc 1 Coproc 0
Text Segment
                                                                                                       □ Labels
                                                                                                                                           □ □
  0x00400000 0x3408ffff ori $8,$0,65535
                                      6: ori $t0, $zero, 0xFFFF
                                                                                                                                                 $v0
                                     7: sll $t0, $t0, 16 #deslocando para a esquerda, ficando 0xFFFF0000
    0x00400004 0x00084400 sll $8,$8,16
                                                                                                                                                 $v1
     0x00400008 0x3510ffff ori $16,$8,65535
                                     8: ori $s0, $t0, 0xFFFF #x = FFFFFFFF
                                                                                                                                                 $a0
 0x0040000c 0x3408493e ori $8,$0,18750
                                     10: ori $t0, $zero, 0x493E
                                                                                                                                                 $al
    0x00400010 0x00088900 s11 $17,$8,4
                                     11: s11 $s1, $t0, 4 #y = 300.000
                                                                                                                                                 $a2
    0x00400014 0x00114080 s11 $8,$17,2
                                     13: sl1 $t0, $s1, 2 #t0 = 4y
                                                                                                                                                 $a3
 0x00400018 0x02089022 sub $18,$16,$8
                                     15: sub $s2, $s0, $t0 #z = x - 4y
                                                                                                                                                 $t0
                                                                                                                                                                                               1200000
                                                                                                                                                 $t1
                                                                                                                                                $t2
$t3
                                                                                                                                                 $t4
                                                                                                                                                 $t5
                                                                                                                                                 $t6
                                                                                                                                                 $t7
                                                                                                                                                 $80
                                                                                                                                                                                                300000
                                                                                                                        ✓ Data
✓ Text
                                                                                                                                                 $83
                                                                                                                                                                               19
                                                                                                                                                 $84
Data Segment
                                                                                                                                           o" 🗹
                                                                                                                                                 $85
      Address
                     Value (+0)
                                    Value (+4)
                                                    Value (+8)
                                                                    Value (+c)
                                                                                   Value (+10)
                                                                                                   Value (+14)
                                                                                                                  Value (+18)
                                                                                                                                  Value (+1c)
                                                                                                                                                 $86
         0x1001000
                                                                                                                                                 $87
        0x10010020
                                                                                                                                                 $t8
        0x10010040
                                                                                                                                                 $t9
                                                                                                                                                                               25
         0x10010060
                                                                                                                                                 $k0
         0x10010080
                                                                                                                                                 $k1
         0x100100a0
                                                                                                                                                                                             268468224
                                                                                                                                                 $gp
        0x100100c0
                                                                                                                                                                                             2147479548
         0x100100e0
                                                                                                                                                 $fp
         0x10010100
                                                                                                                                                 $ra
                                                                                                                                                                                               4194332
        0x10010120
                                                                                                                                                 рc
        0x10010140
        0x10010160
         0x10010180
        0x100101a0
```



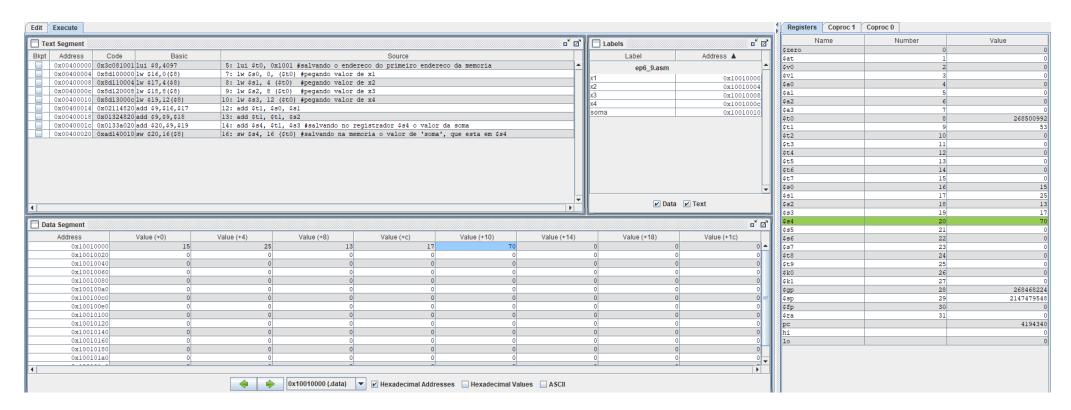
#programa 8

```
Edit Execute
 ep6_8.asm
                                         #colocando 0x12345678 no registrador $8
  2 ori $8, $zero, 0x1234
  3 sll $8, $8, 16
  4 ori $8, $8, 0x5678
  5
     srl $9, $8, 24 #colocando 0x12 no registrador $9
 8 sll $10, $8, 8 #colocando 0x34 no registrador $10
 9 srl $10, $10, 24
10
     andi $11, $8, 0xFF00 #colocando 0x56 no registrador $10
      srl $11, $11, 8
13
     andi $12, $8, 0x00FF #colocando 0x78 no registrador $10
                                                                                                                                                       Registers Coproc 1 Coproc 0
Edit Execute
                                                                                                                                                                            Number
                                                                                                                                                                                               Value
                                                                                                           □ Labels
Text Segment
                                                                                                                                                ு் ⊿ி
                                                                                                                                                      $zero
 Bkpt Address
               Code
                                                                        Source
                                                                                                                                                      $at
      0x00400000 0x34081234 ori $8,$0,4660
                                        2: ori $8, $zero, 0x1234
                                       3: sll $8, $8, 16
     0x00400004 0x00084400 s11 $8.$8.16
      0x00400008 0x35085678 ori $8,$8,22136
                                       4: ori $8, $8, 0x5678
                                                                                                                                                      $a0
      0x0040000c 0x00084e02 srl $9,$8,24
                                        6: srl $9, $8, 24
                                                          #colocando 0x12 no registrador $9
                                                                                                                                                      $al
  0x00400010 0x00085200 s11 $10,$8,8
                                       8: sll $10, $8, 8
                                                         #colocando 0x34 no registrador $10
                                                                                                                                                      $a2
     0x00400014 0x000a5602srl $10,$10,24
                                       9: srl $10, $10, 24
                                       11: andi $11, $8, 0xFF00 #colocando 0x56 no registrador $10
    0x00400018 0x310bff00 andi $11,$8,65280
                                                                                                                                                      $t0
  0x0040001c 0x000b5a02 srl $11,$11,8
                                       12: srl $11, $11, 8
                                                                                                                                                      $t1
  0x00400020 0x310c00ff andi $12,$8,255
                                       14: andi $12, $8, 0x00FF #colocando 0x78 no registrador $10
                                                                                                                                                      $t2
                                                                                                                                                      $t4
                                                                                                                                                      $t5
                                                                                                                                                      $t6
                                                                                                                                                      $t7
                                                                                                                                                                                     15
                                                                                                                                                      $80
                                                                                                                                                      $81

✓ Data 
✓ Text

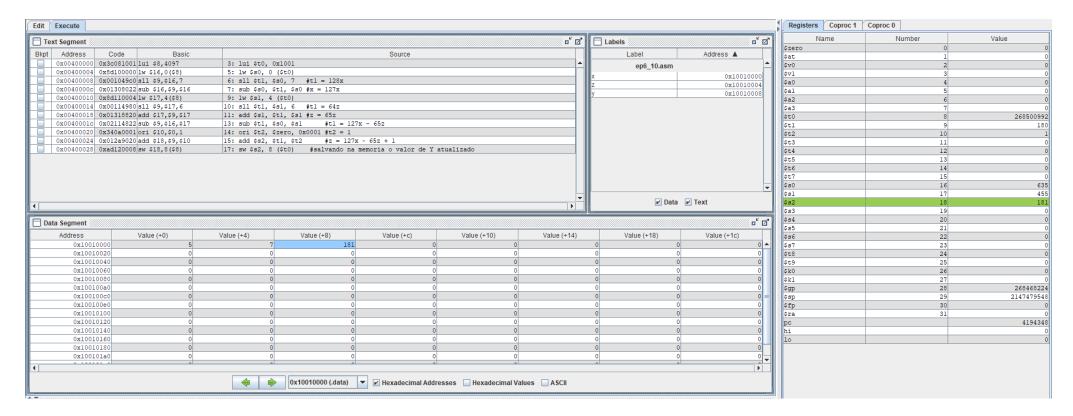
                                                                                                                                                      $82
                                                                                                                                                      $83
                                                                                                                                                      $84
                                                                                                                                                o" ⊘"
 Data Segment
                                                                                                                                                      $85
      Address
                      Value (+0)
                                      Value (+4)
                                                      Value (+8)
                                                                                      Value (+10)
                                                                                                      Value (+14)
                                                                                                                       Value (+18)
                                                                                                                                                      $86
         0x1001000
                                                                                                                                                      $87
                                                                                                                                                                                     23
         0x10010020
                                                                                                                                                      $t8
         0x10010040
                                                                                                                                                      $t9
                                                                                                                                                                                     25
         0x10010060
                                                                                                                                                      $k0
         0x10010080
                                                                                                                                                      $k1
         0x100100a0
                                                                                                                                                      $gp
                                                                                                                                                                                                    268468224
         0x100100c0
                                                                                                                                                                                                   2147479548
                                                                                                                                                      $sp
         0x100100e0
                                                                                                                                                      $fp
                                                                                                                                                                                     30
         0x10010100
                                                                                                                                                      $ra
                                                                                                                                                                                     31
         0x10010120
                                                                                                                                                                                                      4194340
                                                                                                                                                      pc
         0x10010140
         0x10010180
         0x100101a0
```

```
Edit Execute
 ep6_9.asm
 1 #x1 = $s0; x2 = $s1; x3 = $s2; x4 = $s3; soma = $s4
    #endereco inicial = A[0] = $t0 = 0x10010000
 3
 4 .text
 5 lui $t0, 0x1001 #salvando o endereco do primeiro endereco da memoria
 6
 7 lw $s0, 0, ($t0) #pegando valor de x1
 8 lw $s1, 4 ($t0) #pegando valor de x2
 9 lw $s2, 8 ($t0) #pegando valor de x3
10 lw $s3, 12 ($t0) #pegando valor de x4
11
12 add $t1, $s0, $s1
13 add $t1, $t1, $s2
14 add $s4, $t1, $s3 #salvando no registrador $s4 o valor da soma
15
16 sw $s4, 16 ($t0) #salvando na memoria o valor de 'soma', que esta em $s4
17
18 .data
19 x1: .word 15
20 x2: .word 25
21 x3: .word 13
22 x4: .word 17
23 soma: .word -1
```



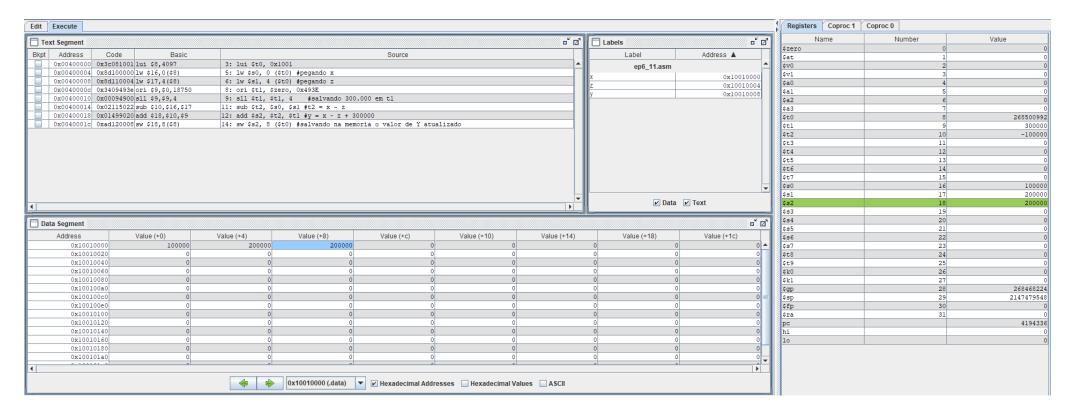
#programa 10

```
ep6_10.asm
1 #$s0 = x; $s1 = z; $s2 = y; $t0 = endereco da primeira posicao de memoria
2 .text
3 lui $t0, 0x1001
5 lw $s0, 0 ($t0)
 6 sll $t1, $s0, 7 #t1 = 128x
7 sub $s0, $t1, $s0 \#x = 127x
8
9 lw $s1, 4 ($t0)
10 sll $t1, $s1, 6 #t1 = 64z
11 add $s1, $t1, $s1 \#z = 65z
12
13 sub $t1, $s0, $s1
                     \#t1 = 127x - 65z
14 ori $t2, $zero, 0x0001 #t2 = 1
15 add $s2, $t1, $t2  #z = 127x - 65z + 1
16
17 sw $s2, 8 ($t0) #salvando na memoria o valor de Y atualizado
18
19 .data
20 x: .word 5
21 z: .word 7
22 y: .word 0
```



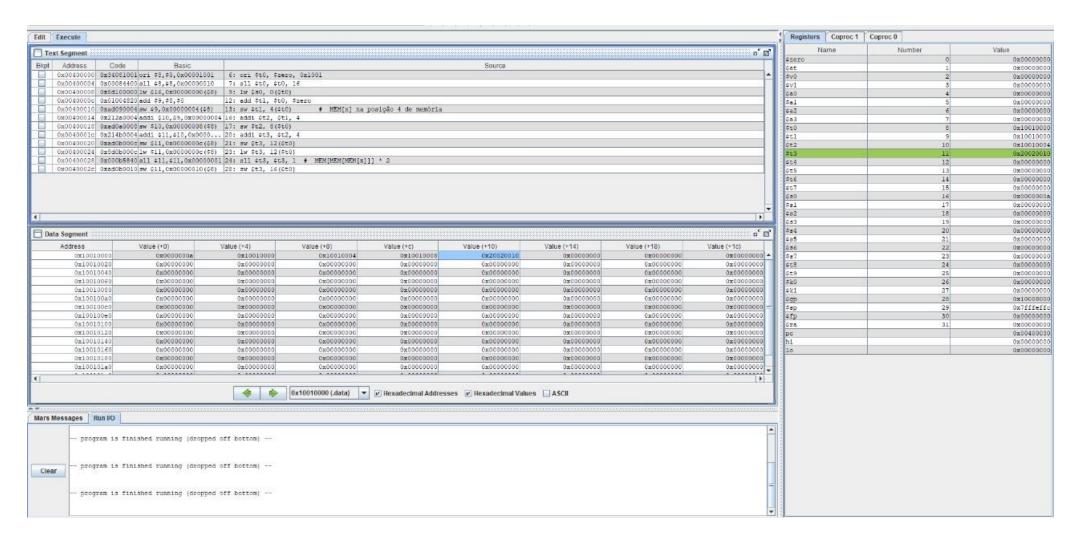
#programa 11

```
ep6_11.asm
 1 #$s0 = x; $s1 = z; $s2 = y; $t0 = endereco da primeira posicao de memoria
 2 .text
 3 lui $t0, 0x1001
 5 lw $s0, 0 ($t0) #pegando x
 6 lw $s1, 4 ($t0) #pegando z
 8 ori $t1, $zero, 0x493E
 9 sll $t1, $t1, 4 #salvando 300.000 em t1
10
11 sub $t2, $s0, $s1 #t2 = x - z
12 add $s2, $t2, $t1 \#y = x - z + 300000
13
14 sw $s2, 8 ($t0) #salvando na memoria o valor de Y atualizado
15
16 .data
17 x: .word 100000
18 z: .word 200000
19 y: .word 0
```



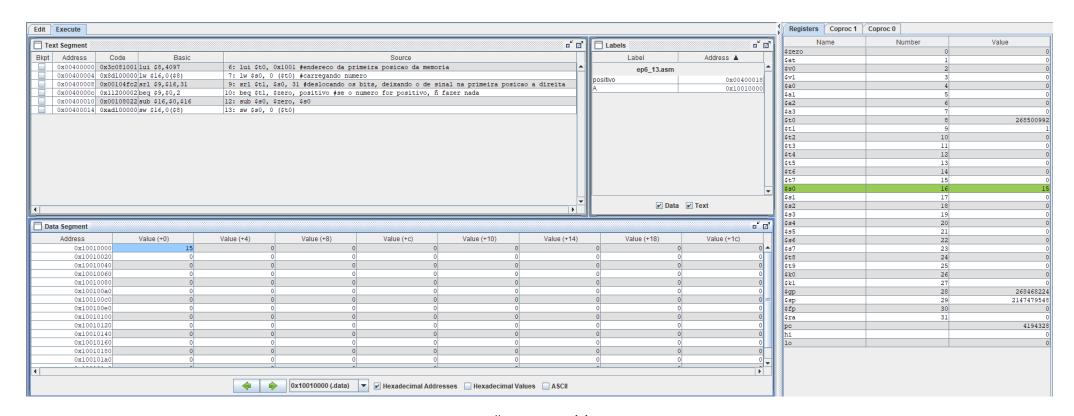
#programa 12

```
1 .data
 2 x: .word 10
 3
 4
 5 .text
 6 ori $t0, $zero, 0x1001
 7 sll $t0, $t0, 16
 8
9 lw $s0, 0($t0)
10
11
12 add $t1, $t0, $zero
13 sw $t1, 4($t0) # MEM[x] na posição 4 de memória
14
15 # endereço de MEN[MEN[x]] na terceira posição de memória
16 addi $t2, $t1, 4
17 sw $t2, 8($t0)
18
19 # MEM[MEM[MEM[x]]] na quarta posição de memória
20 addi $t3, $t2, 4
21 sw $t3, 12($t0)
22
23 lw $t3, 12($t0)
24 sl1 $t3, $t3, 1 # MEM[MEM[MEM[x]]] * 2
25
26
27 # resultado na posição 16)
28 sw $t3, 16($t0)
29
30
```



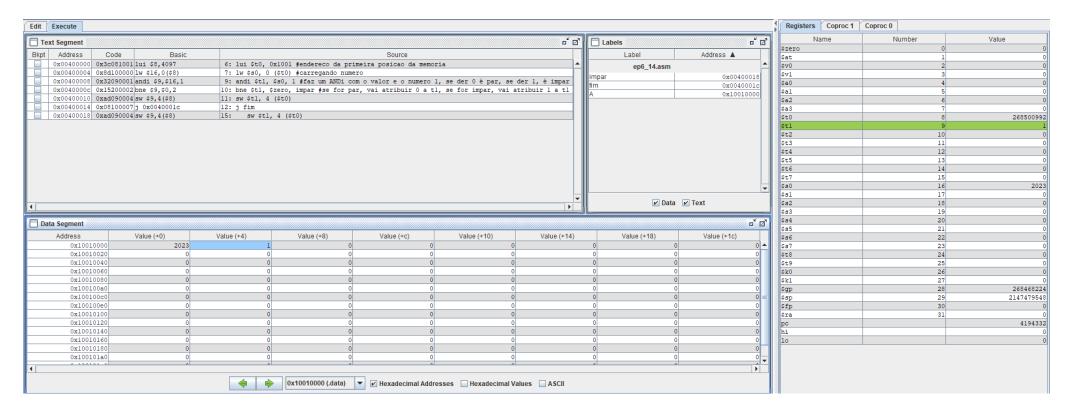
#programa 13

```
ep6_13.asm
    #t0 = primeira posicao da memoria
    #s0 = numero
    #t1 = bit de sinal (1 = negativo, 0 = positivo)
 4
    .text
 5
    lui $t0, 0x1001 #endereco da primeira posicao da memoria
    lw $s0, 0 ($t0) #carregando numero
 8
    srl $t1, $s0, 31 #deslocando os bits, deixando o de sinal na primeira posicao a direita
 9
    beq $t1, $zero, positivo #se o numero for positivo, ñ fazer nada
10
                             #caso negativo, pegar seu modulo e substituir na memória
11
    sub $s0, $zero, $s0
12
    sw $s0, 0 ($t0)
13
14
15
    positivo:
16
            #se ja for positivo ñ faz nada
17
18
19
    .data
20 A: .word -15
```



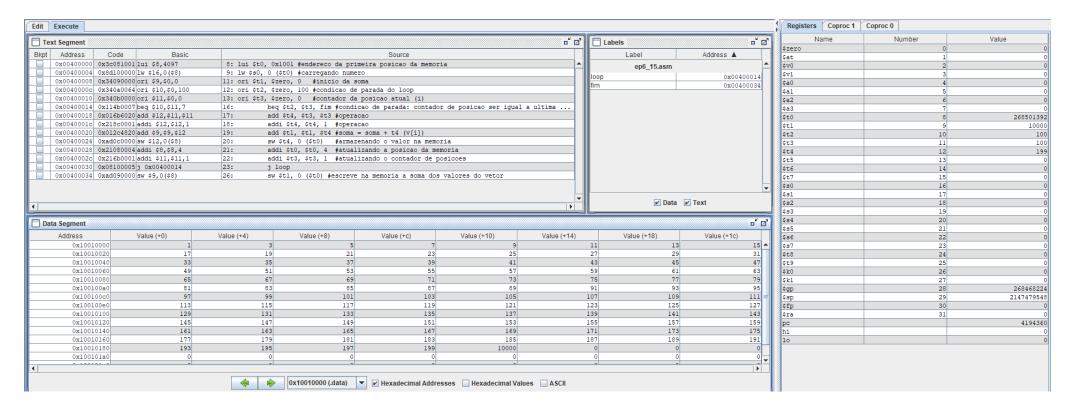
#programa 14

```
ep6_14.asm
    #t0 = primeira posicao da memoria
    #s0 = numero
    #t1 = bit de verificacao de par ou impar (0 = par, 1 = impar)
 3
 4
 5
    .text
    lui $t0, 0x1001 #endereco da primeira posicao da memoria
    lw $s0, 0 ($t0) #carregando numero
 7
 8
    andi $t1, $s0, 1 #faz um ANDi com o valor e o numero 1, se der 0 é par, se der 1, é impar
    bne $t1, $zero, impar #se for par, vai atribuir 0 a t1, se for impar, vai atribuir 1 a t1
10
11
    sw $t1, 4 ($t0)
12
    j fim
13
14
    impar:
15
       sw $t1, 4 ($t0)
16
17
    fim:
18
19
    .data
20 A: .word 2023
```



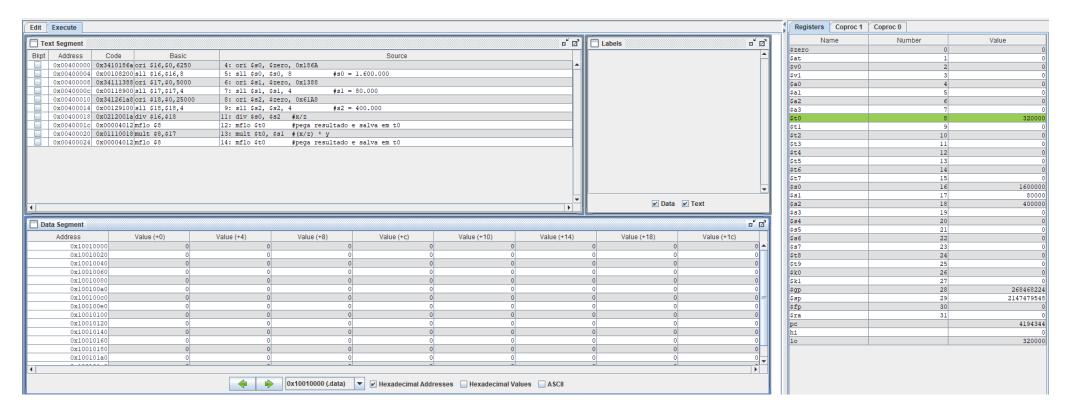
#programa 15

```
ep6_15.asm
 3 #t3 = contador da posicao atual (i)
 4 #t4 = valor a ser inserido na memoria
 5
 6 .text
 7 lui $t0, 0x1001 #endereco da primeira posicao da memoria
 8 lw $s0, 0 ($t0) #carregando numero
10 ori $t1, $zero, 0 #inicio da soma
ori $t2, $zero, 100 #condicao de parada do loop
ori $t3, $zero, 0 #contador da posicao atual (i)
13
14 loop:
15
            beq $t2, $t3, fim #condicao de parada: contador de posicao ser igual a ultima posicao do array + 1
16
            add $t4, $t3, $t3 #operacao
17
            addi $t4, $t4, 1 #operacao
18
            add $t1, $t1, $t4 \#soma = soma + t4 (v[i])
19
            sw $t4, 0 ($t0) #armazenando o valor na memoria
20
           addi $t0, $t0, 4 #atualizando a posicao da memoria
21
            addi $t3, $t3, 1 #atualizando o contador de posicoes
22
            j loop
23
24 fim:
25
            sw $t1, 0 ($t0) #escreve na memoria a soma dos valores do vetor
26
   .data
28
```



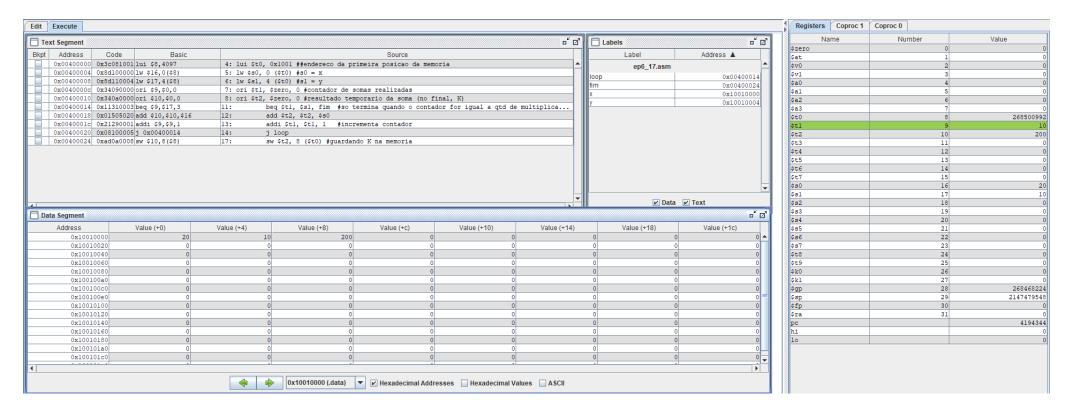
#programa 16

```
ep6_16.asm
1 \#$s0 = x; $s1 = y; $s2 = z
 2
 3 .text
 4 ori $s0, $zero, 0x186A
 5 sll $s0, $s0, 8 #s0 = 1.600.000
 6 ori $s1, $zero, 0x1388
7 sll $s1, $s1, 4 \#s1 = 80.000
8 ori $s2, $zero, 0x61A8
9 s11 $s2, $s2, 4 #s2 = 400.000
10
11 div $s0, $s2 #x/z
12 mflo $t0 #pega resultado e salva em t0
13 mult $t0, $s1 \#(x/z) * y
14 mflo $t0 #pega resultado e salva em t0
15
16 .data
17
```



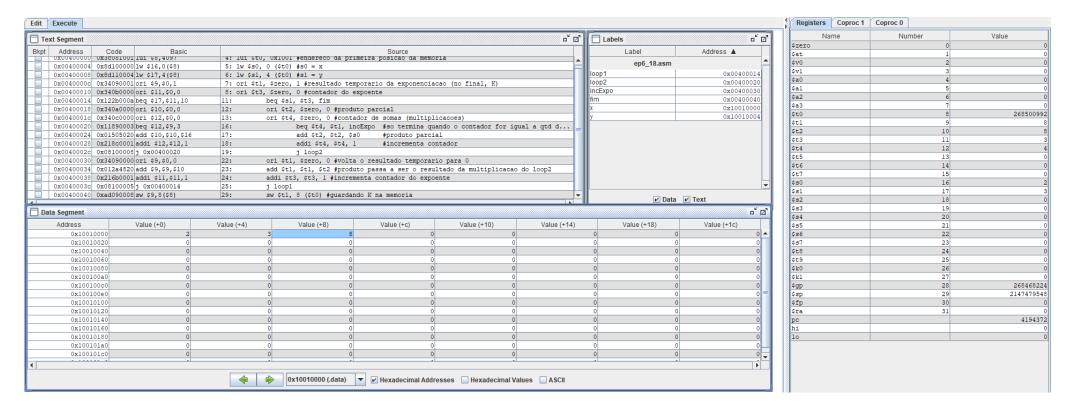
#programa 17

```
ep6_17.asm
 1 \#\$s0 = x; \$s1 = y; \$t2 = k
 2
   .text
 3
 4 lui $t0, 0x1001 ##endereco da primeira posicao da memoria
 5 lw $s0, 0 ($t0) #s0 = x
 6 lw $s1, 4 ($t0) \#s1 = y
 7 ori $t1, $zero, 0 #contador de somas realizadas
   ori $t2, $zero, 0 #resultado temporario da soma (no final, K)
 9
10 loop:
            beq $t1, $s1, fim #so termina quando o contador for igual a qtd de multiplicacoes necessarias
11
12
            add $t2, $t2, $s0
13
            addi $t1, $t1, 1 #incrementa contador
            j loop
14
15
16 fim:
17
            sw $t2, 8 ($t0) #guardando K na memoria
18
19 .data
20 x: .word 20
  y: .word 10
```



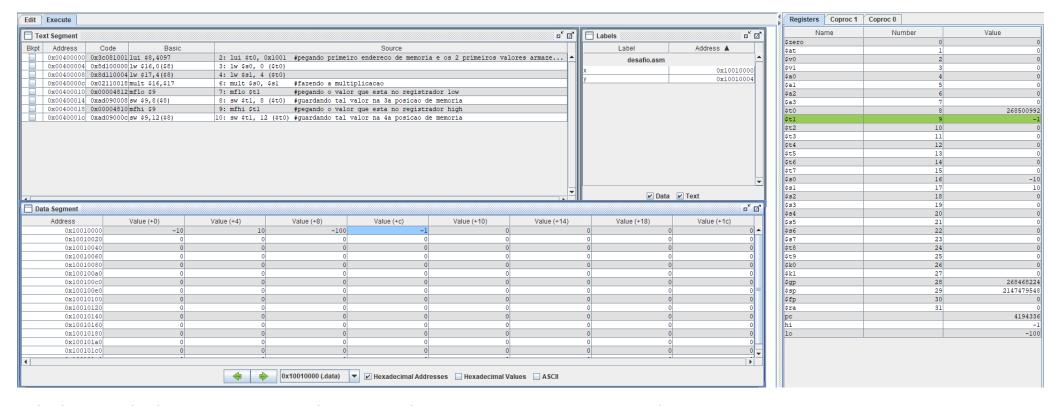
#programa 18

```
ep6_18.asm
 1 \#$s0 = x; $s1 = y
   .text
 4 lui $t0, 0x1001 #endereco da primeira posicao da memoria
 5 lw $s0, 0 ($t0) \#s0 = x
 6 lw $s1, 4 ($t0) \#s1 = y
 7 ori $t1, $zero, 1 #resultado temporario da exponenciacao (no final, K)
 8 ori $t3, $zero, O #contador do expoente
 9
10 loop1:
11
            beq $s1, $t3, fim
            ori $t2, $zero, 0 #produto parcial
12
            ori $t4, $zero, 0 #contador de somas (multiplicacoes)
13
14
15
            100p2:
                   beq $t4, $t1, incExpo #so termina quando o contador for igual a qtd de somas necessarias
16
17
                    add $t2, $t2, $s0
                                          #produto parcial
                    addi $t4, $t4, 1
                                           #incrementa contador
18
19
                    j loop2
20
21 incExpo:
22
            ori $t1, $zero, 0 #volta o resultado temporario para 0
23
            add $t1, $t1, $t2 #produto passa a ser o resultado da multiplicacao do loop2
24
            addi $t3, $t3, 1 #incrementa contador do expoente
25
            j loop1
26
27
28 fim:
29
            sw $t1, 8 ($t0) #guardando K na memoria
30
31 .data
32 x: .word 2
33 v: .word 3
```



#desafio

```
desafio.asm
 1 .text
    lui $t0, 0x1001 #pegando primeiro endereco de memoria e os 2 primeiros valores armazenados
   lw $s0, 0 ($t0)
   lw $s1, 4 ($t0)
 5
   mult $s0, $s1 #fazendo a multiplicacao
             #pegando o valor que esta no registrador low
    mflo $t1
    sw $t1, 8 ($t0) #guardando tal valor na 3a posicao de memoria
                 #pegando o valor que esta no registrador high
    mfhi $t1
    sw $t1, 12 ($t0) #quardando tal valor na 4a posicao de memoria
10
11
12
    .data
   x: .word -10
13
   y: .word 10
14
```



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

C. 64

2. Quais os registradores que armazenam os resultados na multiplicação?

B. hi e lo

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

A. mult

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

C. mflo \$8

5. <u>Se tivermos dois inteiros, cada um com 32 bits, quantos bits deveremos estar preparados para receber no quociente?</u>

6. Após a instrução div, qual registrador possui o quociente?

A. lo

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

D. div

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

A. 1110 0110

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

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

10. Qual sequencia de instruções avalia 3x+7, onde x é iniciado no reg. \$8 e o resultado armazenado em \$9?

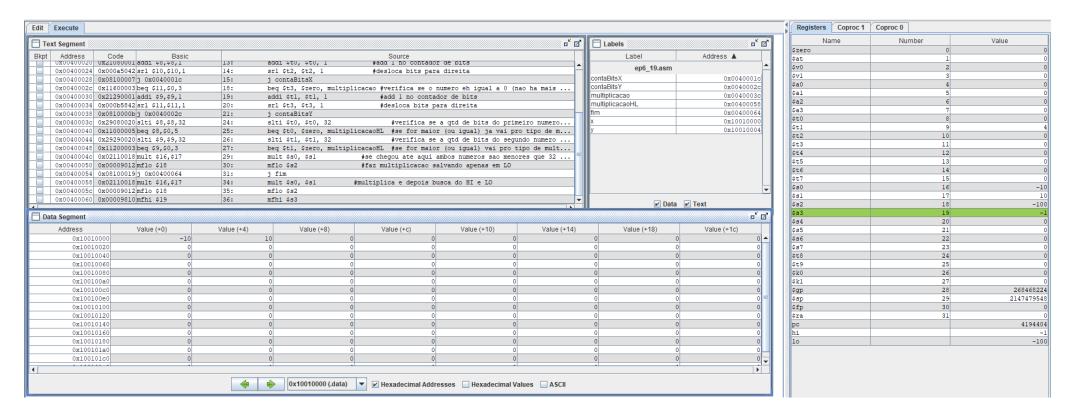
A. ori \$3,\$0,3

mult \$8,\$3

mflo \$9

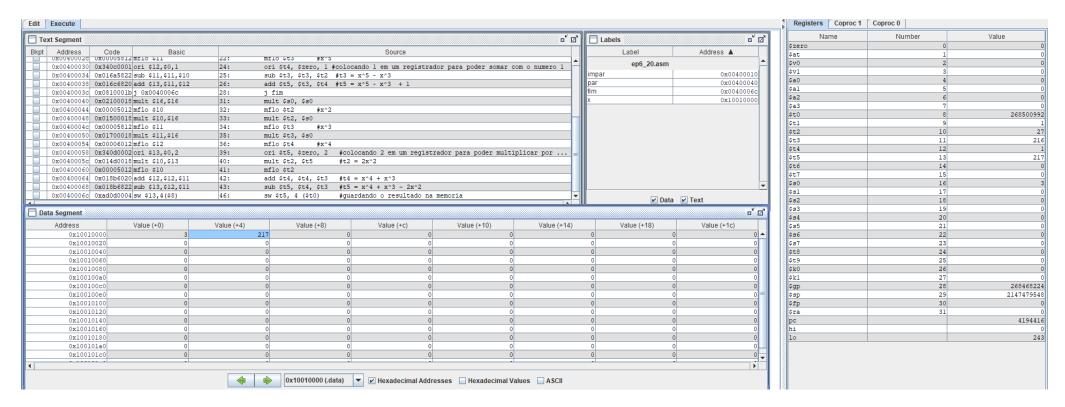
addi \$9,\$9,7

```
ep6_19.asm
 1 .text
 2 lui $t0, 0x1001
 3 lw $s0, 0 ($t0)
 4 lw $s1, 4 ($t0)
 6 ori $tO, $zero, O #contador X
 7 ori $t1, $zero, 0 #contador Y
 8 or $t2, $zero, $s0 #copia de s2
 9 or $t3, $zero, $s1 #copia de s1
10
11 contaBitsX:
12
            beq $t2, $zero, contaBitsY #verifica se o numero eh igual a 0 (nao ha mais bits significativos)
13
            addi $t0, $t0, 1
                                        #add 1 no contador de bits
            srl $t2, $t2, 1
                                        #desloca bits para direita
14
15
            j contaBitsX
16
17 contaBitsY:
18
            beq $t3, $zero, multiplicacao #verifica se o numero eh igual a 0 (nao ha mais bits significativos)
19
                                          #add 1 no contador de bits
            addi $t1, $t1, 1
20
            srl $t3, $t3, 1
                                          #desloca bits para direita
21
            j contaBitsY
22
23 multiplicacao:
24
                                             #verifica se a qtd de bits do primeiro numero eh menor que 32
            slti $t0, $t0, 32
25
            beq $t0, $zero, multiplicacaoHL #se for major (ou igual) ja vai pro tipo de multiplicacao que salva em HI e LO
26
            slti $t1, $t1, 32
                                             #verifica se a qtd de bits do segundo numero eh menor que 32
27
            beq $t1, $zero, multiplicacaoHL #se for major (ou igual) vai pro tipo de multiplicacao que salva em HI e LO
28
29
            mult $s0, $s1
                                     #se cheqou ate aqui ambos numeros sao menores que 32 bits
            mflo $s2
                                     #faz multiplicacao salvando apenas em LO
30
31
            j fim
32
33 multiplicacaoHL:
34
            mult $s0, $s1
                                   #multiplica e depois busca do HI e LO
35
            mflo $s2
36
            mfhi $s3
37
38 fim:
39
40
   .data
   x: .word -10
42
   y: .word 10
```



#programa 20

```
ep6 20.asm
 1 .text
   lui $t0, 0x1001
                        #primeira posicao da memoria
    lw $s0, 0 ($t0)
                        #pegando x
   andi $t1, $s0, 1
                        #verifica se eh par ou impar (se $t1 = 1, impar; se $t1 = 0, par)
   beq $t1, $zero, par #se for par vai para o label par, caso impar, sequir as proximas linhas
 8
    impar:
           mult $s0, $s0 \#x^2
9
           mflo $t2
10
           mult $t2, $s0
11
12
           mflo $t2
                         #x^3
13
           mult $t2, $s0 \#x^4
14
           mflo $t3
15
           mult $t3, $s0
           mflo $t3
                         #x^5
16
17
18
            ori $t4, $zero, 1 #colocando 1 em um registrador para poder somar com o numero 1
            sub $t3, $t3, $t2 \#t3 = x^5 - x^3
19
            add $t5, $t3, $t4 #t5 = x^5 - x^3 + 1
20
21
22
            j fim
23
24 par:
           mult $s0, $s0
25
26
           mflo $t2
                         #x^2
27
           mult $t2, $s0
28
           mflo $t3
                          #x^3
29
           mult $t3, $s0
           mflo $t4
                         #x^4
30
31
32
33
            ori $t5, $zero, 2 #colocando 2 em um registrador para poder multiplicar por pelo numero 2
                                #t2 = 2x^2
34
           mult $t2, $t5
           mflo $t2
35
36
            add $t4, $t4, $t3 \#t4 = x^4 + x^3
            sub $t5, $t4, $t3 #t5 = x^4 + x^3 - 2x^2
37
38
    fim:
39
            sw $t5, 4 ($t0)
                                #quardando o resultado na memoria
40
```



#programa 21

```
ep6_21.asm
 1 .text
 2 lui $t0, 0x1001
                       #primeira posicao da memoria
    lw $s0, 0 ($t0)
                       #pegando x
   slt $t1, $zero, $s0 #verificando se x > 0 (se $t1= 1, sim; se $t1= 0, nao)
    bne $t1, $zero, maior
 7
    menorIgual:
 8
            mult $s0, $s0 #x^2
 9
            mflo $t2
10
11
            mult $t2, $s0 #x^3
12
            mflo $t2
13
            mult $t2, $s0
14
            mflo $t2
                          #x^4
15
16
            ori $t3, $zero, 1 #colocando 1 em um registrador para poder subtrair com o numero 1
17
            sub $t3, $t2, $t3 \#t3 = x^4 - 1
18
19
            j fim
20
21
    maior:
22
            mult $s0, $s0 #x^2
23
            mflo $t2
24
            mult $t2, $s0
25
            mflo $t2
                        #x^3
26
27
            ori $t3, $zero, 1 #colocando 1 em um registrador para poder somar com o numero 1
28
            add $t3, $t2, $t3 \#t3 = x^3 + 1
29
30 fim:
31
            sw $t3, 4 ($t0)
32
33 .data
34 x: .word 2
```

