Exercício Prático 05

Carolina Morais Nigri

Parte 1

1) A
2) D
3) A
4) C
5) D
6) C
7) A
8) D
9) B
10) A
11) B
12) D
13) B
14) C
15) A
16) D
17) A
18) A

Parte 2

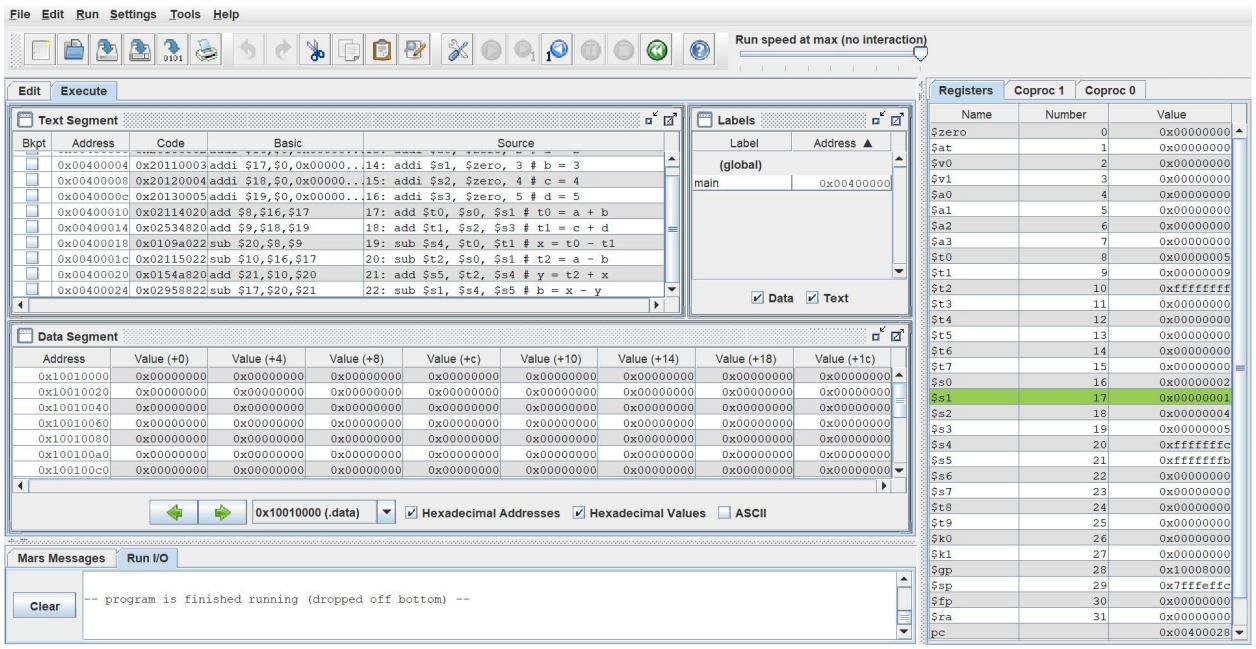
Programa 1:

Edit Execute

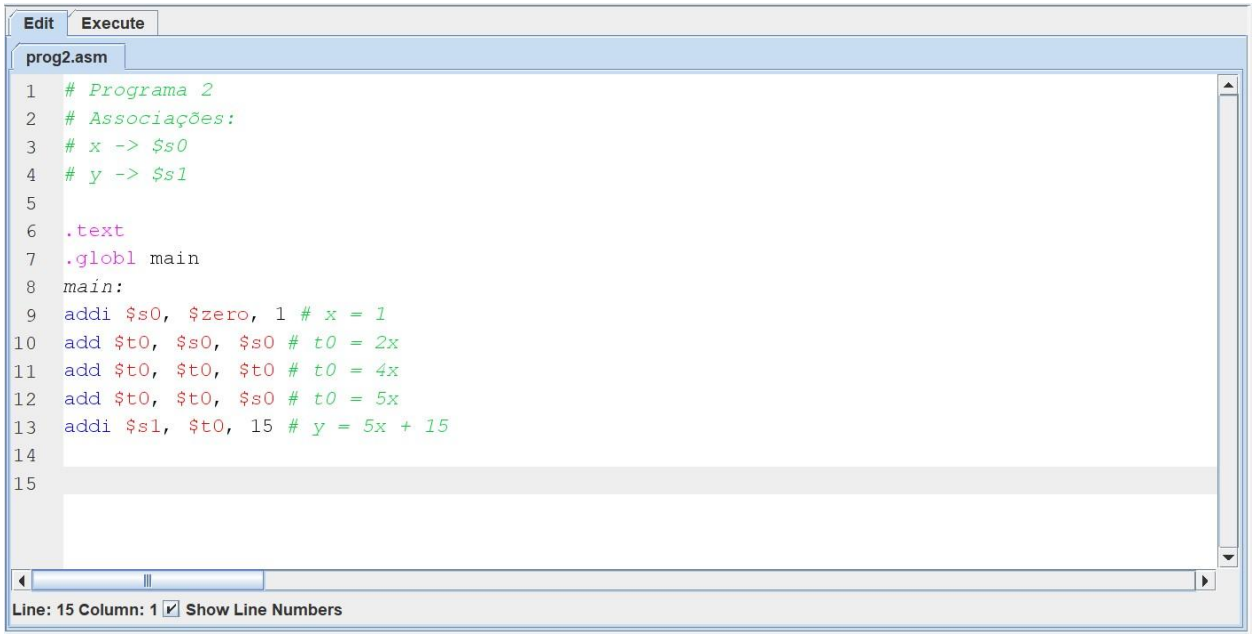
prog1.asm

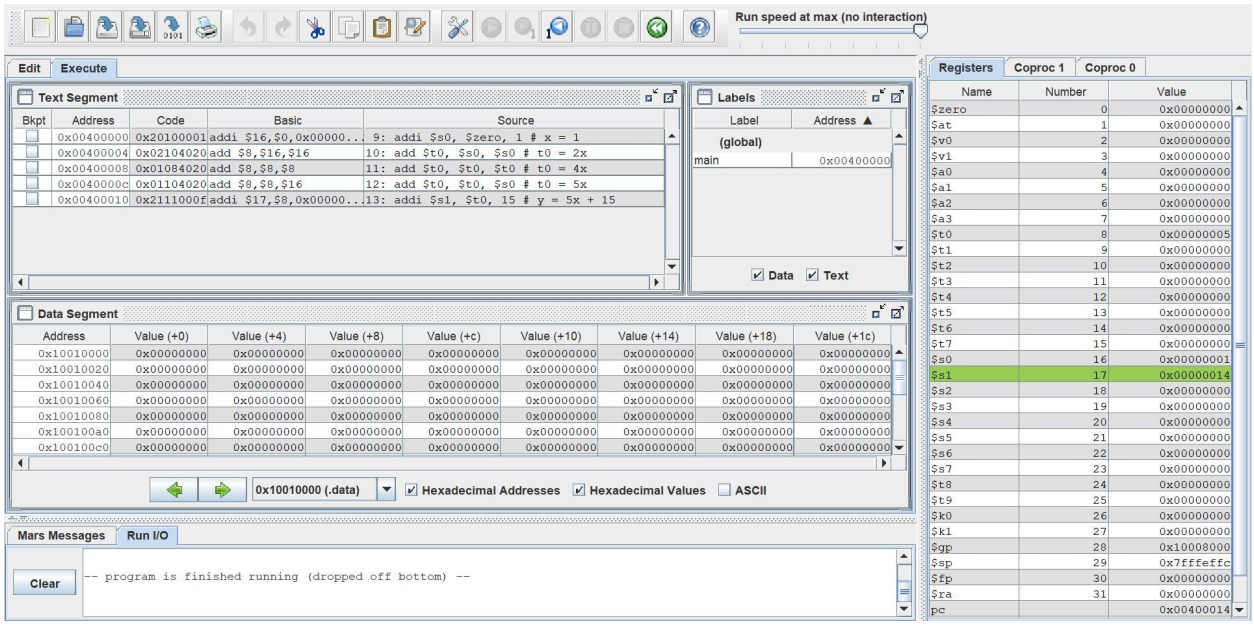
```
1  # Programa 1
2  # Associações:
3  # a -> $s0
4  # b -> $s1
5  # c -> $s2
6  # d -> $s3
7  # x -> $s4
8  # y -> $s5
9
10 .text
11 .globl main
12 main:
13 addi $s0, $zero, 2 # a = 2
14 addi $s1, $zero, 3 # b = 3
15 addi $s2, $zero, 4 # c = 4
16 addi $s3, $zero, 5 # d = 5
17 add $t0, $s0, $s1 # t0 = a + b
18 add $t1, $s2, $s3 # t1 = c + d
19 sub $s4, $t0, $t1 # x = t0 - t1
20 sub $t2, $s0, $s1 # t2 = a - b
21 add $s5, $t2, $s4 # y = t2 + x
22 sub $s1, $s4, $s5 # b = x - y
```

Line: 24 Column: 1 ☑ Show Line Numbers

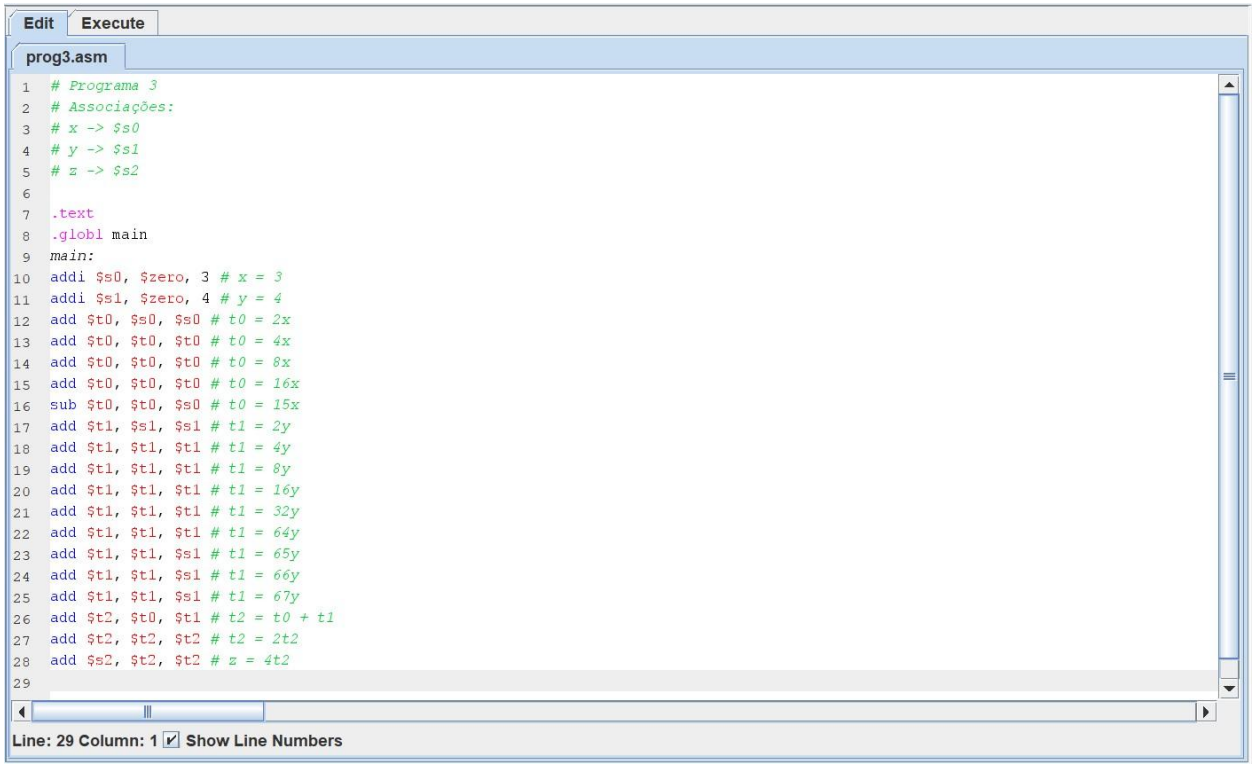


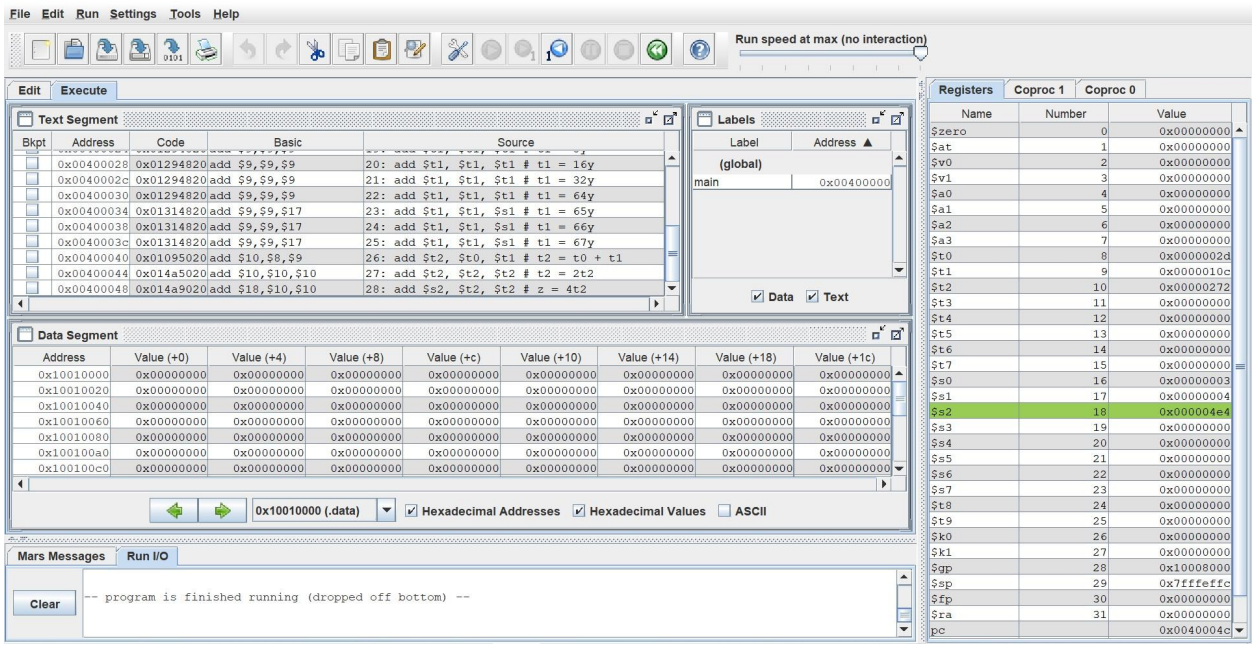
Programa 2:


Edit
Execute
prog2.asm
1 # Programa 2
2 # Associações:
3 # x -> $s0
4 # y -> $s1
5
6 .text
7 .globl main
8 main:
9 addi $s0, $zero, 1 # x = 1
10 add $t0, $s0, $s0 # t0 = 2x
11 add $t0, $t0, $t0 # t0 = 4x
12 add $t0, $t0, $s0 # t0 = 5x
13 addi $s1, $t0, 15 # y = 5x + 15
14
15
Line: 15 Column: 1
Show Line Numbers

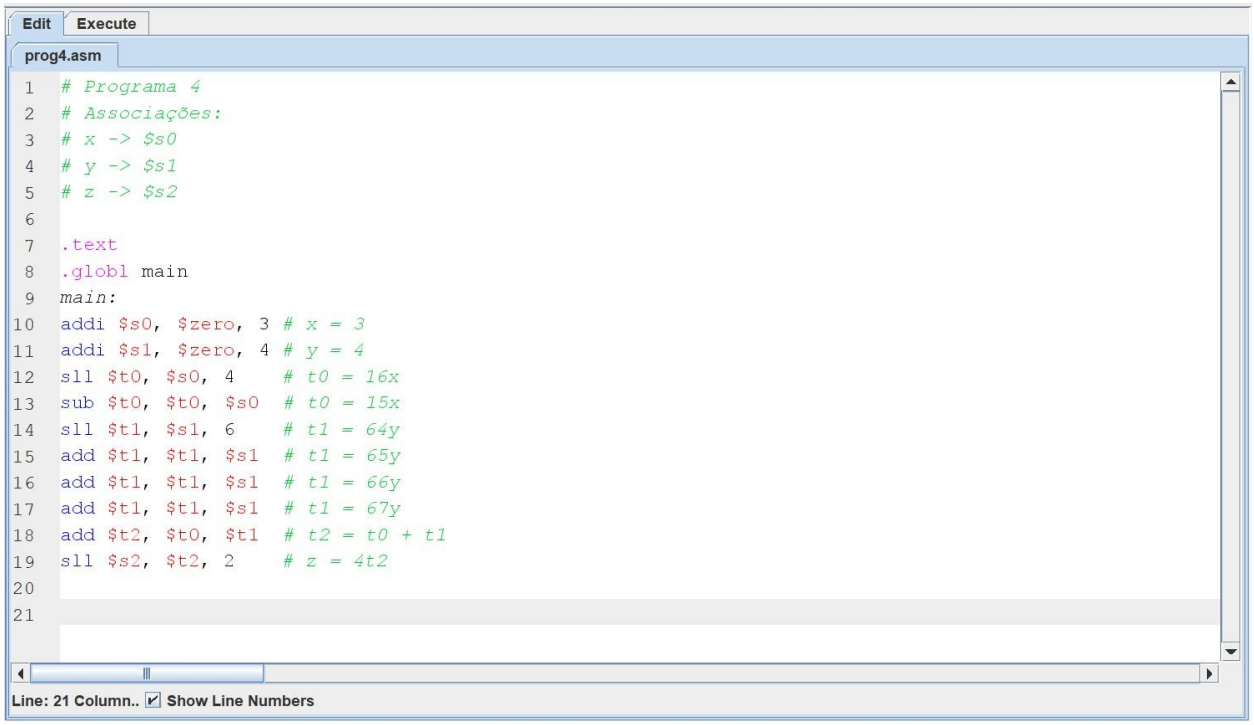

Run speed at max (no interaction)
Edit
Execute
Text Segment
Bkpt Address Code Basic Source
0x00400000 0x20100001 addi $16,$0,0x00000.. 9: addi $s0, $zero, 1 # x = 1
0x00400004 0x02104020 add $8,$16,$16 10: add $t0, $s0, $s0 # t0 = 2x
0x00400008 0x01084020 add $8,$8,$8 11: add $t0, $t0, $t0 # t0 = 4x
0x0040000c 0x01104020 add $8,$8,$16 12: add $t0, $t0, $s0 # t0 = 5x
0x00400010 0x2111000f addi $17,$8,0x00000..13: addi $s1, $t0, 15 # y = 5x + 15
Labels
Label Address
(global)
main 0x00400000
Data
Text
Data Segment
Address Value (+0) Value (+4) Value (+8) Value (+c) Value (+10) Value (+14) Value (+18) Value (+1c)
0x10010000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010020 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010040 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010060 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010080 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x100100a0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x100100c0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010000 (.data)
Hexadecimal Addresses
Hexadecimal Values
ASCII
Mars Messages
Run I/O
Clear
-- program is finished running (dropped off bottom) --
Registers
Coproc 1
Coproc 0
Name Number Value
$zero 0 0x00000000
$at 1 0x00000000
$v0 2 0x00000000
$v1 3 0x00000000
$a0 4 0x00000000
$a1 5 0x00000000
$a2 6 0x00000000
$a3 7 0x00000000
$t0 8 0x00000005
$t1 9 0x00000000
$t2 10 0x00000000
$t3 11 0x00000000
$t4 12 0x00000000
$t5 13 0x00000000
$t6 14 0x00000000
$t7 15 0x00000000
$s0 16 0x00000001
$s1 17 0x00000014
$s2 18 0x00000000
$s3 19 0x00000000
$s4 20 0x00000000
$s5 21 0x00000000
$s6 22 0x00000000
$s7 23 0x00000000
$t8 24 0x00000000
$t9 25 0x00000000
$k0 26 0x00000000
$k1 27 0x00000000
$gp 28 0x10008000
$sp 29 0x7fffeffc
$fp 30 0x00000000
$ra 31 0x00000000
pc 0x00400014

Programa 3:

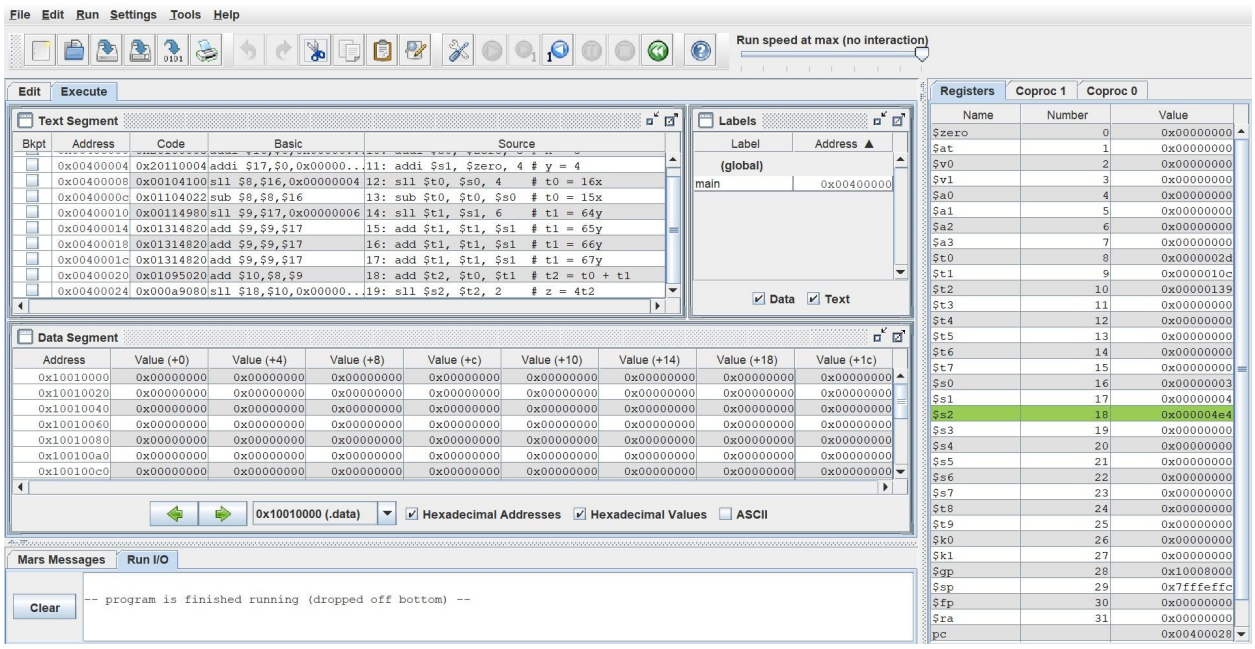

Edit
Execute
prog3.asm
1 # Programa 3
2 # Associações:
3 # x -> $s0
4 # y -> $s1
5 # z -> $s2
6
7 .text
8 .globl main
9 main:
10 addi $s0, $zero, 3 # x = 3
11 addi $s1, $zero, 4 # y = 4
12 add $t0, $s0, $s0 # t0 = 2x
13 add $t0, $t0, $t0 # t0 = 4x
14 add $t0, $t0, $t0 # t0 = 8x
15 add $t0, $t0, $t0 # t0 = 16x
16 sub $t0, $t0, $s0 # t0 = 15x
17 add $t1, $s1, $s1 # t1 = 2y
18 add $t1, $t1, $t1 # t1 = 4y
19 add $t1, $t1, $t1 # t1 = 8y
20 add $t1, $t1, $t1 # t1 = 16y
21 add $t1, $t1, $t1 # t1 = 32y
22 add $t1, $t1, $t1 # t1 = 64y
23 add $t1, $t1, $s1 # t1 = 65y
24 add $t1, $t1, $s1 # t1 = 66y
25 add $t1, $t1, $s1 # t1 = 67y
26 add $t2, $t0, $t1 # t2 = t0 + t1
27 add $t2, $t2, $t2 # t2 = 2t2
28 add $s2, $t2, $t2 # z = 4t2
29
Line: 29 Column: 1 Show Line Numbers


File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Bkpt Address Code Basic Source
0x00400028 0x01294820 add $9,$9,$9 20: add $t1, $t1, $t1 # t1 = 16y
0x0040002c 0x01294820 add $9,$9,$9 21: add $t1, $t1, $t1 # t1 = 32y
0x00400030 0x01294820 add $9,$9,$9 22: add $t1, $t1, $t1 # t1 = 64y
0x00400034 0x01314820 add $9,$9,$17 23: add $t1, $t1, $s1 # t1 = 65y
0x00400038 0x01314820 add $9,$9,$17 24: add $t1, $t1, $s1 # t1 = 66y
0x0040003c 0x01314820 add $9,$9,$17 25: add $t1, $t1, $s1 # t1 = 67y
0x00400040 0x01095020 add $10,$8,$9 26: add $t2, $t0, $t1 # t2 = t0 + t1
0x00400044 0x014a5020 add $10,$10,$10 27: add $t2, $t2, $t2 # t2 = 2t2
0x00400048 0x014a9020 add $18,$10,$10 28: add $s2, $t2, $t2 # z = 4t2
Labels
Label Address
(global)
main 0x00400000
Data Text
Data Segment
Address Value (+0) Value (+4) Value (+8) Value (+c) Value (+10) Value (+14) Value (+18) Value (+1c)
0x10010000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010020 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010040 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010060 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010080 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x100100a0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x100100c0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII
Mars Messages Run I/O
Clear
-- program is finished running (dropped off bottom) --
Registers Coproc 1 Coproc 0
Name Number Value
$zero 0 0x00000000
$at 1 0x00000000
$v0 2 0x00000000
$v1 3 0x00000000
$a0 4 0x00000000
$a1 5 0x00000000
$a2 6 0x00000000
$a3 7 0x00000000
$t0 8 0x0000002d
$t1 9 0x0000010c
$t2 10 0x00000272
$t3 11 0x00000000
$t4 12 0x00000000
$t5 13 0x00000000
$t6 14 0x00000000
$t7 15 0x00000000
$s0 16 0x00000003
$s1 17 0x00000004
$s2 18 0x000004e4
$s3 19 0x00000000
$s4 20 0x00000000
$s5 21 0x00000000
$s6 22 0x00000000
$s7 23 0x00000000
$t8 24 0x00000000
$t9 25 0x00000000
$k0 26 0x00000000
$k1 27 0x00000000
$gp 28 0x10008000
$sp 29 0x7fffeffc
$fp 30 0x00000000
$ra 31 0x00000000
pc 0x0040004c

Programa 4:



```
# Programa 4
# Associações:
# x -> $s0
# y -> $s1
# z -> $s2

.text
.globl main
main:
addi $s0, $zero, 3 # x = 3
addi $s1, $zero, 4 # y = 4
sll $t0, $s0, 4    # t0 = 16x
sub $t0, $t0, $s0  # t0 = 15x
sll $t1, $s1, 6    # t1 = 64y
add $t1, $t1, $s1  # t1 = 65y
add $t1, $t1, $s1  # t1 = 66y
add $t1, $t1, $s1  # t1 = 67y
add $t2, $t0, $t1  # t2 = t0 + t1
sll $s2, $t2, 2    # z = 4t2
```



Programa 5:

Edit Execute

prog5.asm

```
1  # Programa 5
2  # Associações:
3  # x -> $s0
4  # y -> $s1
5  # z -> $s2
6
7  .text
8  .globl main
9  main:
10 ori $t0, $zero, 0x1  # t0 = 0x00000001
11 sll $t0, $t0, 16     # t0 = 0x00010000
12 ori $s0, $t0, 0x86A0 # x = 0x000186A0
13 ori $t0, $zero, 0x3  # t0 = 0x00000003
14 sll $t0, $t0, 16     # t0 = 0x00030000
15 ori $s1, $t0, 0x0D40 # y = 0x00030D40
16 add $s2, $s0, $s1 # z = x + y
17
```

Line: 17 Column: 1 ☑ Show Line Numbers

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x34080001	ori $8,$0,1	10: ori $t0, $zero, 0x1 # t0 = 0x000000..
	0x00400004	0x00084400	sll $8,$8,16	11: sll $t0, $t0, 16 # t0 = 0x000100..
	0x00400008	0x351086a0	ori $16,$8,34464	12: ori $s0, $t0, 0x86A0 # x = 0x000186A0
	0x0040000c	0x34080003	ori $8,$0,3	13: ori $t0, $zero, 0x3 # t0 = 0x000000..
	0x00400010	0x00084400	sll $8,$8,16	14: sll $t0, $t0, 16 # t0 = 0x000300..
	0x00400014	0x35110d40	ori $17,$8,3392	15: ori $s1, $t0, 0x0D40 # y = 0x00030D40
	0x00400018	0x02119020	add $18,$16,$17	16: add $s2, $s0, $s1 # z = x + y

Labels

Label	Address ▲
(global)	
main	0x00400000

☑ Data ☑ Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0	0	0	0	0	0	0	0
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0

0x10010000 (.data) ☑ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

Mars Messages Run I/O

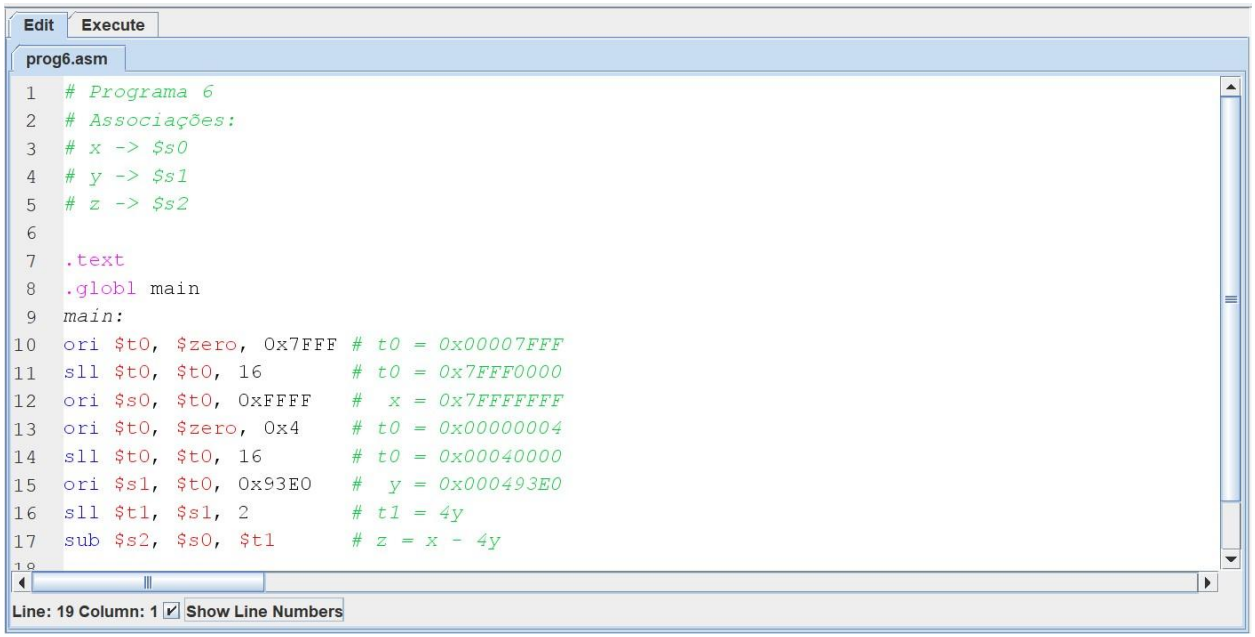
Clear

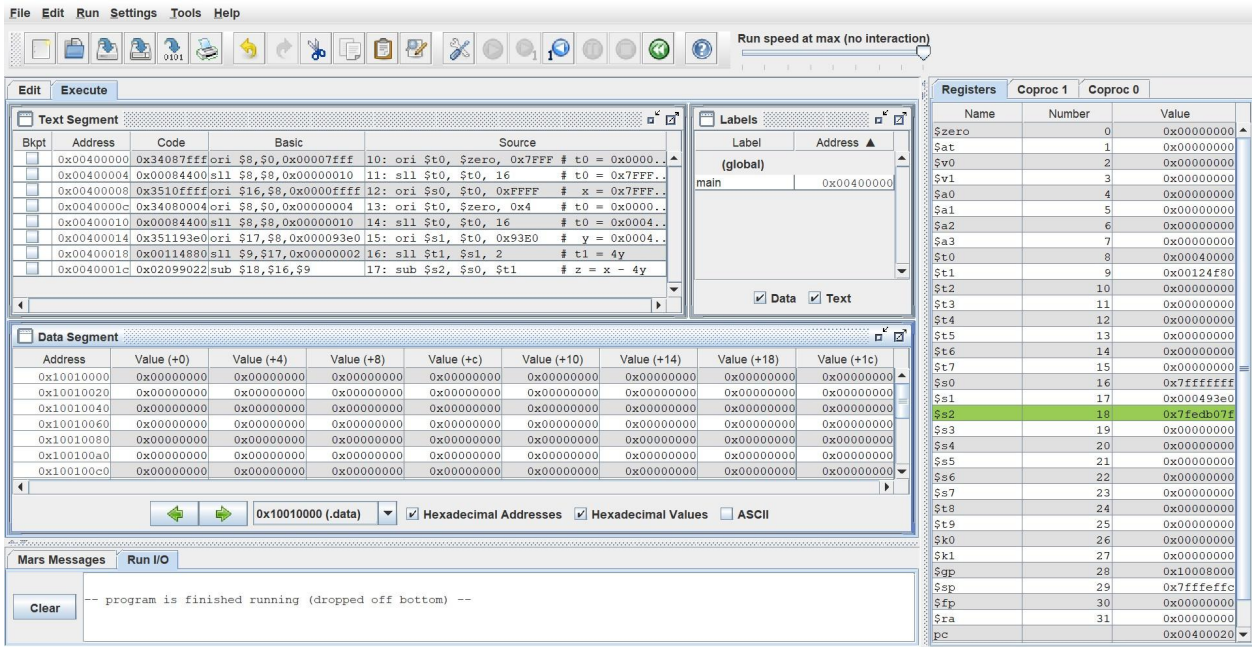
-- program is finished running (dropped off bottom) --

Registers Coproc 1 Coproc 0

Name	Number	Value
$zero	0	0
$at	1	0
$v0	2	0
$v1	3	0
$a0	4	0
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	196608
$t1	9	0
$t2	10	0
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	100000
$s1	17	200000
$s2	18	300000
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194332

Programa 6:


Edit
Execute
prog6.asm
1 # Programa 6
2 # Associações:
3 # x -> $s0
4 # y -> $s1
5 # z -> $s2
6
7 .text
8 .globl main
9 main:
10 ori $t0, $zero, 0x7FFF # t0 = 0x00007FFF
11 sll $t0, $t0, 16 # t0 = 0x7FFF0000
12 ori $s0, $t0, 0xFFFF # x = 0x7FFFFFFF
13 ori $t0, $zero, 0x4 # t0 = 0x00000004
14 sll $t0, $t0, 16 # t0 = 0x00040000
15 ori $s1, $t0, 0x93E0 # y = 0x000493E0
16 sll $t1, $s1, 2 # t1 = 4y
17 sub $s2, $s0, $t1 # z = x - 4y
Line: 19 Column: 1
Show Line Numbers


File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Bkpt Address Code Basic Source
0x00400000 0x34087fff ori $8,$0,0x00007fff 10: ori $t0, $zero, 0x7FFF # t0 = 0x0000..
0x00400004 0x00084400 sll $8,$8,0x00000010 11: sll $t0, $t0, 16 # t0 = 0x7FFF..
0x00400008 0x3510ffff ori $16,$8,0x0000ffff 12: ori $s0, $t0, 0xFFFF # x = 0x7FFF..
0x0040000c 0x34080004 ori $8,$0,0x00000004 13: ori $t0, $zero, 0x4 # t0 = 0x0000..
0x00400010 0x00084400 sll $8,$8,0x00000010 14: sll $t0, $t0, 16 # t0 = 0x0004..
0x00400014 0x351193e0 ori $17,$8,0x000093e0 15: ori $s1, $t0, 0x93E0 # y = 0x0004..
0x00400018 0x00114880 sll $9,$17,0x00000002 16: sll $t1, $s1, 2 # t1 = 4y
0x0040001c 0x02099022 sub $18,$16,$9 17: sub $s2, $s0, $t1 # z = x - 4y
Labels
Label Address
(global)
main 0x00400000
Data Text
Registers Coproc 1 Coproc 0
Name Number Value
$zero 0 0x00000000
$at 1 0x00000000
$v0 2 0x00000000
$v1 3 0x00000000
$a0 4 0x00000000
$a1 5 0x00000000
$a2 6 0x00000000
$a3 7 0x00000000
$t0 8 0x00040000
$t1 9 0x00124f80
$t2 10 0x00000000
$t3 11 0x00000000
$t4 12 0x00000000
$t5 13 0x00000000
$t6 14 0x00000000
$t7 15 0x00000000
$s0 16 0x7fffffff
$s1 17 0x000493e0
$s2 18 0x7fedb07f
$s3 19 0x00000000
$s4 20 0x00000000
$s5 21 0x00000000
$s6 22 0x00000000
$s7 23 0x00000000
$t8 24 0x00000000
$t9 25 0x00000000
$k0 26 0x00000000
$k1 27 0x00000000
$gp 28 0x10008000
$sp 29 0x7fffeffc
$fp 30 0x00000000
$ra 31 0x00000000
pc 0x00400020
Data Segment
Address Value (+0) Value (+4) Value (+8) Value (+c) Value (+10) Value (+14) Value (+18) Value (+1c)
0x10010000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010020 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010040 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010060 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010080 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x100100a0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x100100c0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010000 (.data)
Hexadecimal Addresses
Hexadecimal Values
ASCII
Mars Messages
Run I/O
Clear
-- program is finished running (dropped off bottom) --

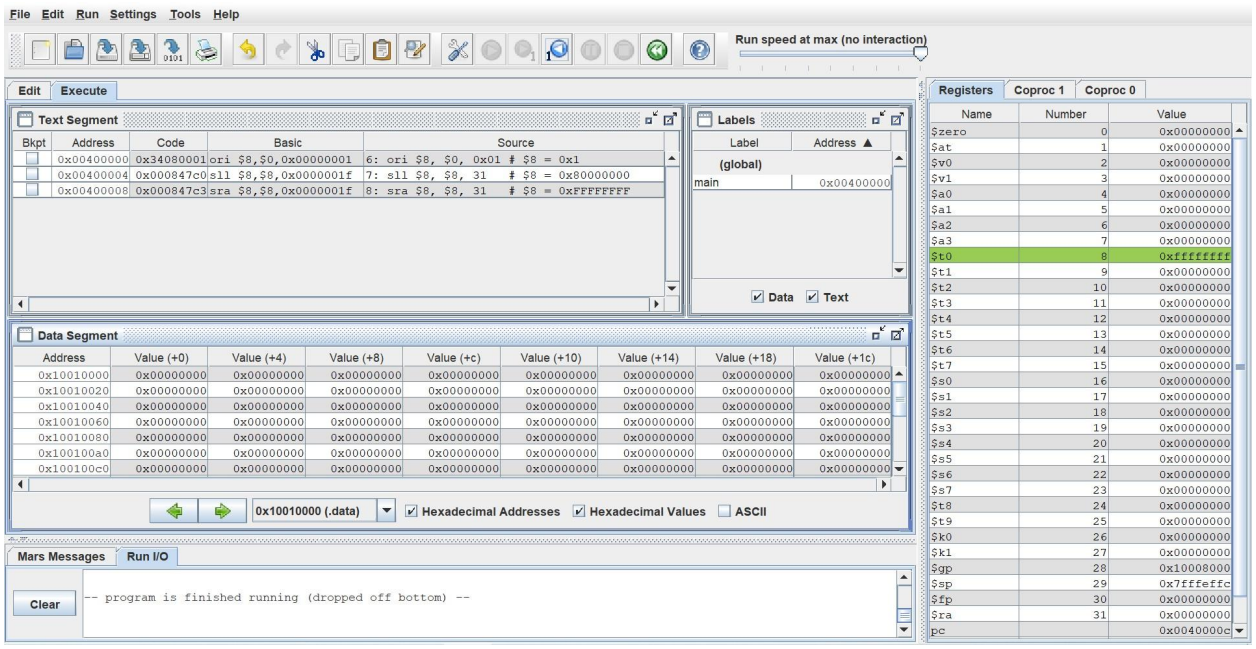
Programa 7:

Edit Execute

prog7.asm

```
1  # Programa 7
2
3  .text
4  .globl main
5  main:
6  ori $8, $0, 0x01 # $8 = 0x1
7  sll $8, $8, 31   # $8 = 0x80000000
8  sra $8, $8, 31   # $8 = 0xFFFFFFFF
9
```

Line: 9 Column: 1 Show Line Numbers



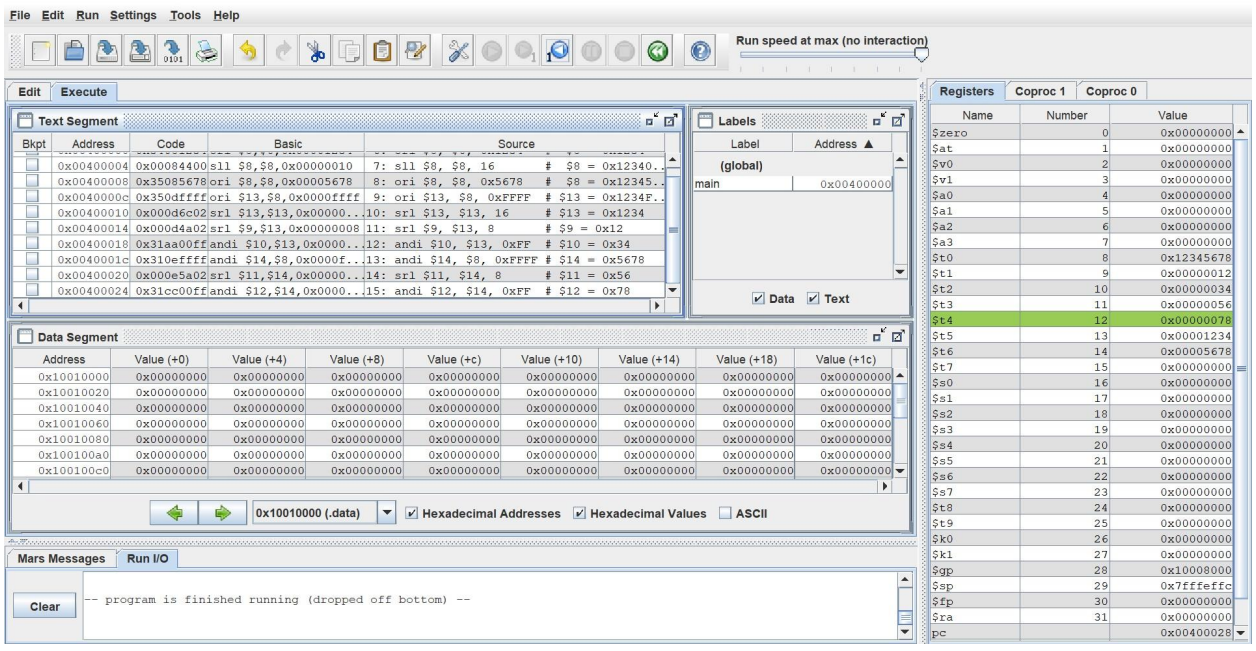
Programa 8:

Edit Execute

prog8.asm

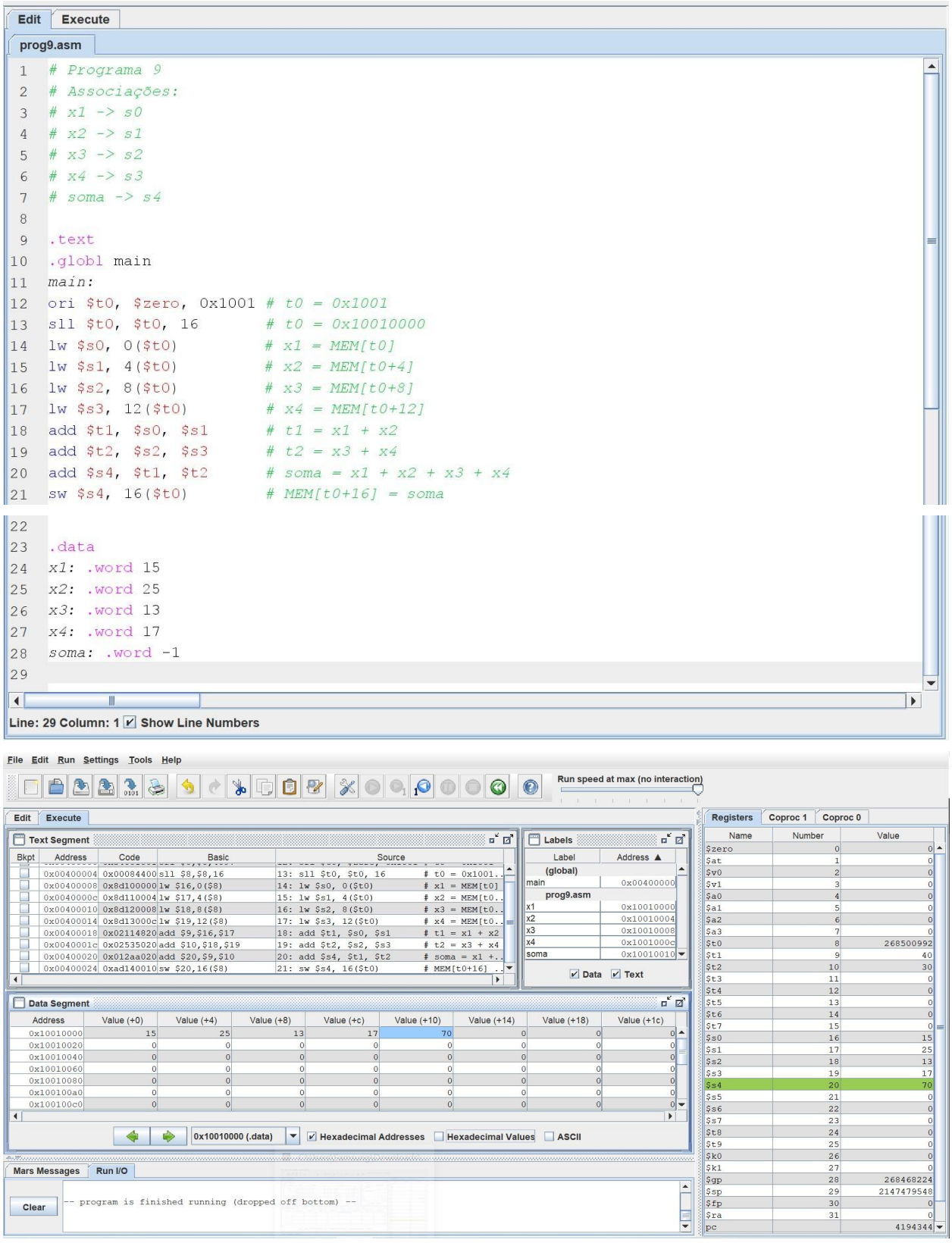
```
1  # Programa 8
2
3  .text
4  .globl main
5  main:
6  ori $8, $0, 0x1234    #  $8 = 0x1234
7  sll $8, $8, 16        #  $8 = 0x12340000
8  ori $8, $8, 0x5678    #  $8 = 0x12345678
9  ori $13, $8, 0xFFFF   # $13 = 0x1234FFFF
10 srl $13, $13, 16      # $13 = 0x1234
11 srl $9, $13, 8        # $9 = 0x12
12 andi $10, $13, 0xFF   # $10 = 0x34
13 andi $14, $8, 0xFFFF  # $14 = 0x5678
14 srl $11, $14, 8       # $11 = 0x56
15 andi $12, $14, 0xFF   # $12 = 0x78
16
```

Line: 16 Column: 1 Show Line Numbers



Programa 9:

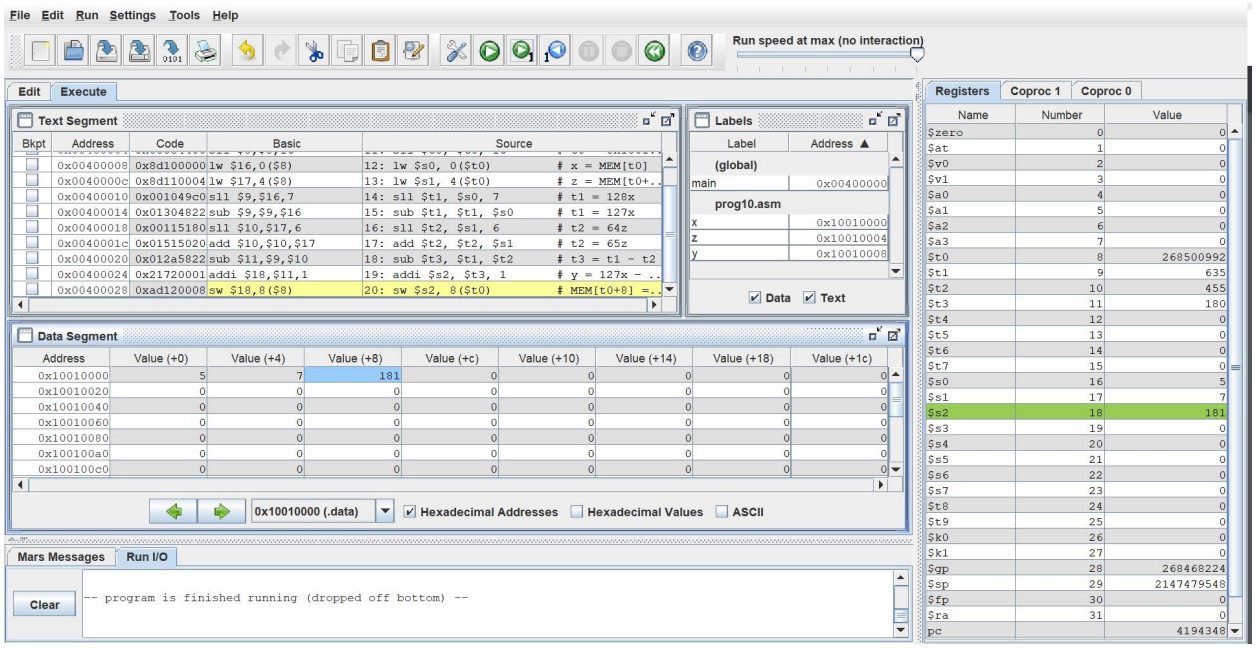
```
# Programa 9
# Associações:
# x1 -> s0
# x2 -> s1
# x3 -> s2
# x4 -> s3
# soma -> s4

.text
.globl main
main:
ori $t0, $zero, 0x1001 # t0 = 0x1001
sll $t0, $t0, 16       # t0 = 0x10010000
lw $s0, 0($t0)         # x1 = MEM[t0]
lw $s1, 4($t0)         # x2 = MEM[t0+4]
lw $s2, 8($t0)         # x3 = MEM[t0+8]
lw $s3, 12($t0)        # x4 = MEM[t0+12]
add $t1, $s0, $s1      # t1 = x1 + x2
add $t2, $s2, $s3      # t2 = x3 + x4
add $s4, $t1, $t2      # soma = x1 + x2 + x3 + x4
sw $s4, 16($t0)        # MEM[t0+16] = soma

.data
x1: .word 15
x2: .word 25
x3: .word 13
x4: .word 17
soma: .word -1
```



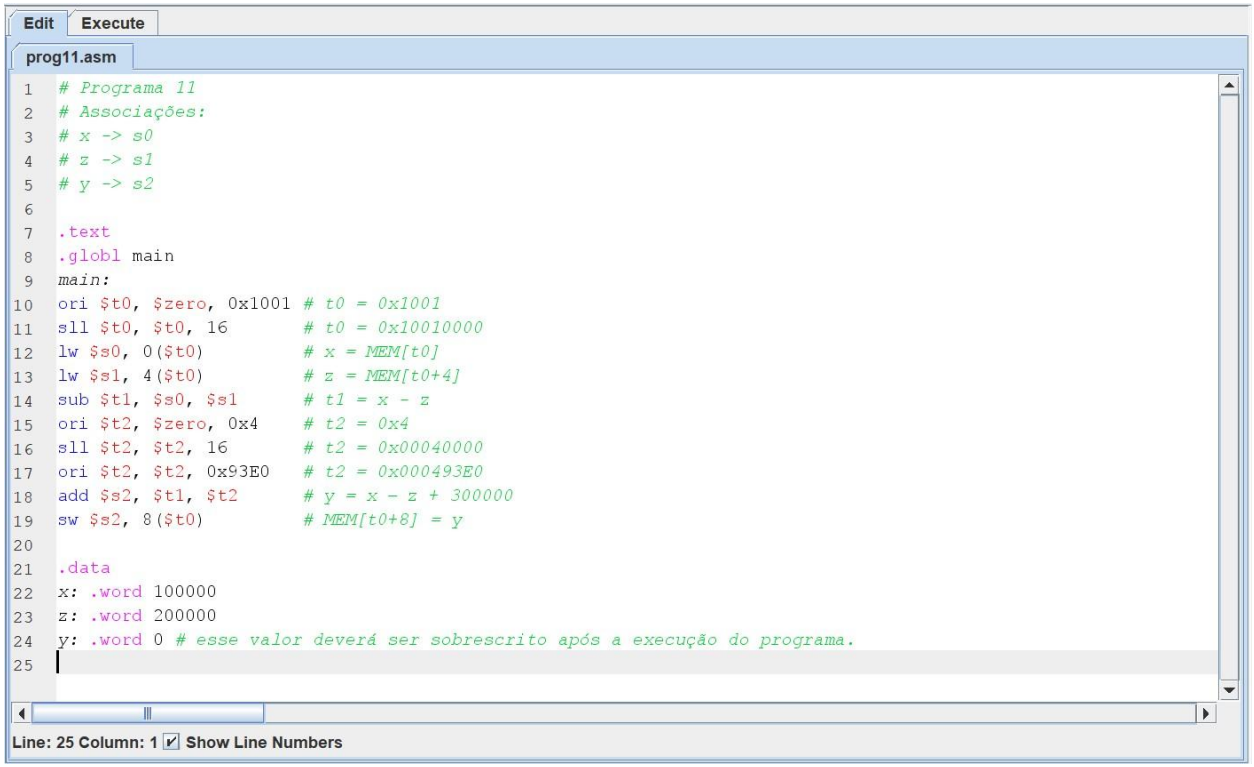
Programa 10:

```
# Programa 10
# Associações:
# x -> s0
# z -> s1
# y -> s2

.text
.globl main
main:
ori $t0, $zero, 0x1001 # t0 = 0x1001
sll $t0, $t0, 16       # t0 = 0x10010000
lw $s0, 0($t0)         # x = MEM[t0]
lw $s1, 4($t0)         # z = MEM[t0+4]
sll $t1, $s0, 7        # t1 = 128x
sub $t1, $t1, $s0      # t1 = 127x
sll $t2, $s1, 6        # t2 = 64z
add $t2, $t2, $s1      # t2 = 65z
sub $t3, $t1, $t2      # t3 = t1 - t2
addi $s2, $t3, 1       # y = 127x - 65z + 1
sw $s2, 8($t0)         # MEM[t0+8] = y

.data
x: .word 5
z: .word 7
y: .word 0 # esse valor deverá ser sobrescrito após a execução do programa
```

Line: 26 Column: 1 Show Line Numbers

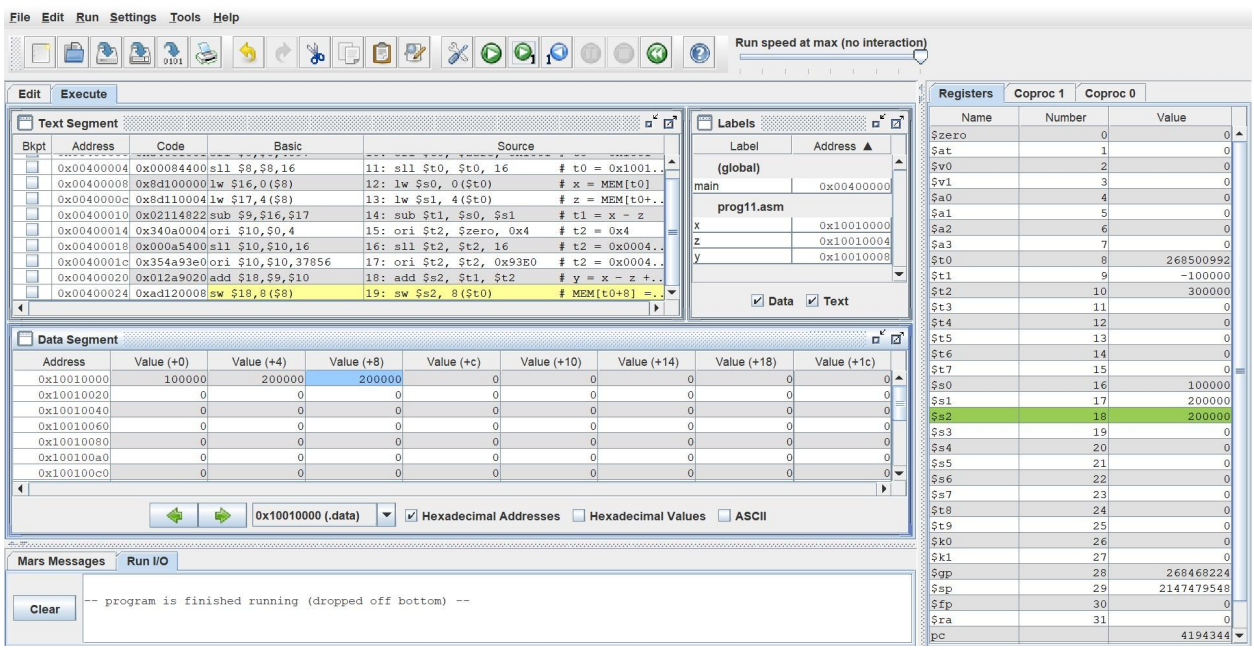


Programa 11:



```
# Programa 11
# Associações:
# x -> s0
# z -> s1
# y -> s2

.text
.globl main
main:
ori $t0, $zero, 0x1001 # t0 = 0x1001
sll $t0, $t0, 16       # t0 = 0x10010000
lw $s0, 0($t0)         # x = MEM[t0]
lw $s1, 4($t0)         # z = MEM[t0+4]
sub $t1, $s0, $s1      # t1 = x - z
ori $t2, $zero, 0x4    # t2 = 0x4
sll $t2, $t2, 16       # t2 = 0x00040000
ori $t2, $t2, 0x93E0   # t2 = 0x000493E0
add $s2, $t1, $t2      # y = x - z + 300000
sw $s2, 8($t0)         # MEM[t0+8] = y

.data
x: .word 100000
z: .word 200000
y: .word 0 # esse valor deverá ser sobrescrito após a execução do programa.
```

Line: 25 Column: 1 Show Line Numbers



Programa 12:

Edit | Execute

prog12.asm

```
1  # Programa 12
2  # Associações:
3  # x -> s0
4  # k -> s1
5
6  .text
7  .globl main
8  main:
9  ori $t0, $zero, 0x1001 # t0 = 0x1001
10 sll $t0, $t0, 16       # t0 = 0x10010000
11 ori $t0, $t0, 0xC      # t0 = 0x1001000C
12 lw $s0, 0($t0)         # x = MEM[t0]
13 lw $t1, 0($s0)         # t1 = MEM[x]
14 lw $t1, 0($t1)         # t1 = MEM[MEM[x]]
15 lw $s1, 0($t1)         # k = MEM[ MEM[MEM[x]] ]
16 sll $s1, $s1, 1        # k = 2k
17 sw $s1, 0($t1)         # MEM[ MEM[MEM[x]] ] = 2k
18
19 .data
20 k:  .word 100000
21 p2: .word 0x10010000
22 p1: .word 0x10010004
23 x:  .word 0x10010008
```

Line: 24 Column: 1 ☑ Show Line Numbers

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit | Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x34081001	ori $8,$0,0x00001001	9: ori $t0, $zero, 0x1001 # t0 = 0x1001
	0x00400004	0x00084400	sll $8,$8,0x00000010	10: sll $t0, $t0, 16 # t0 = 0x1001..
	0x00400008	0x3508000c	ori $8,$8,0x0000000c	11: ori $t0, $t0, 0xC # t0 = 0x1001..
	0x0040000c	0x8d100000	lw $16,0x00000000($8)	12: lw $s0, 0($t0) # x = MEM[t0]
	0x00400010	0x8e090000	lw $9,0x00000000($16)	13: lw $t1, 0($s0) # t1 = MEM[x]
	0x00400014	0x8d290000	lw $9,0x00000000($9)	14: lw $t1, 0($t1) # t1 = MEM[ME..
	0x00400018	0x8d310000	lw $17,0x00000000($9)	15: lw $s1, 0($t1) # k = MEM[ME..
	0x0040001c	0x00118840	sll $17,$17,0x00000..	16: sll $s1, $s1, 1 # k = 2k
	0x00400020	0xad310000	sw $17,0x00000000($9)	17: sw $s1, 0($t1) # MEM[MEM[ME..

Labels

Label	Address ▲
(global)	
main	0x00400000
prog12.asm	
k	0x10010000
p2	0x10010004
p1	0x10010008
x	0x1001000c

☑ Data ☑ Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00030d40	0x10010000	0x10010004	0x10010008	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values ☐ ASCII

Mars Messages | Run I/O

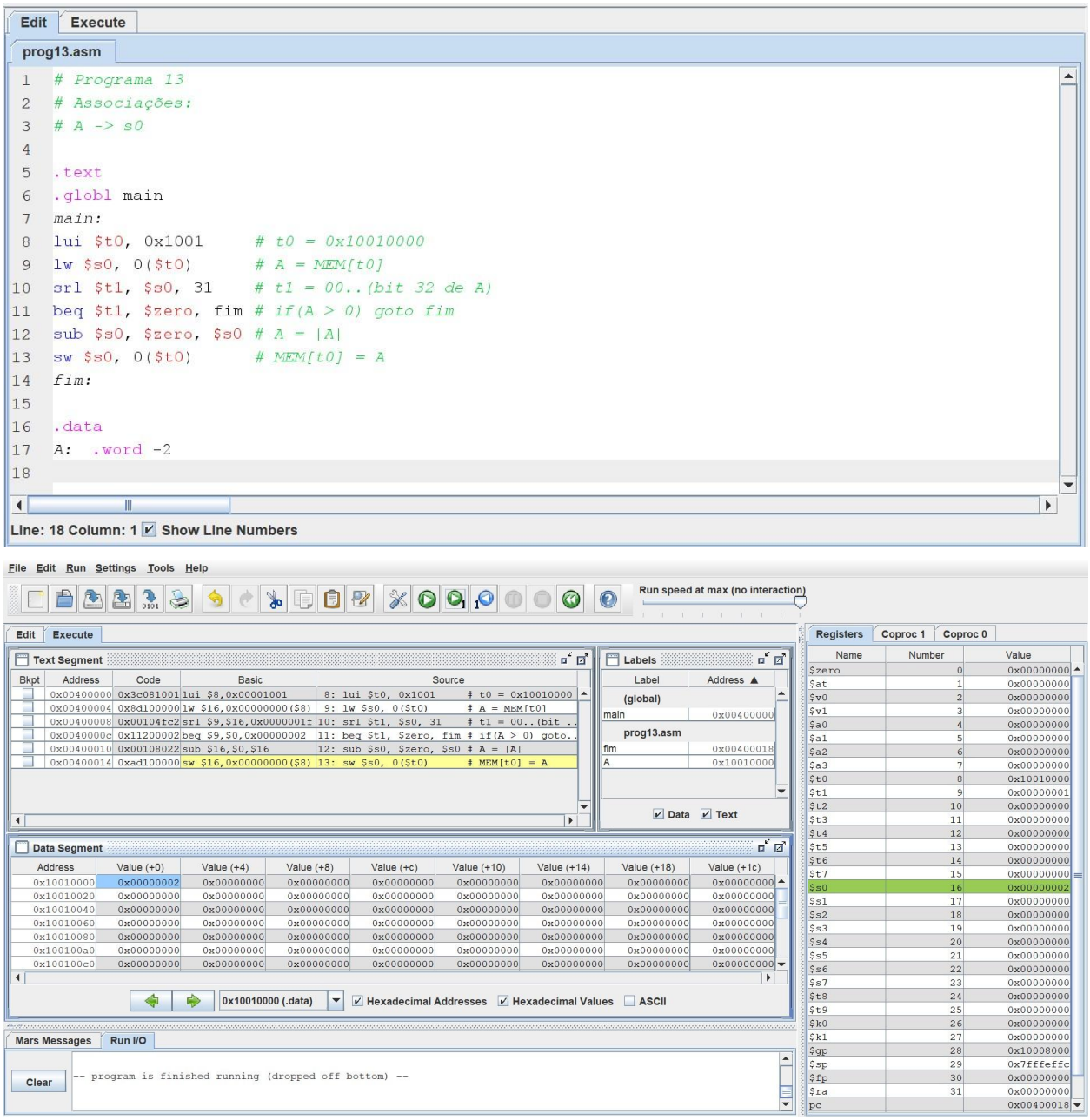
Clear

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

Registers | Coproc 1 | Coproc 0

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x1001000c
$t1	9	0x10010000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x10010008
$s1	17	0x00030d40
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400024

Programa 13:



```
# Programa 13
# Associações:
# A -> s0

.text
.globl main
main:
lui $t0, 0x1001      # t0 = 0x10010000
lw $s0, 0($t0)       # A = MEM[t0]
srl $t1, $s0, 31     # t1 = 00..(bit 32 de A)
beq $t1, $zero, fim # if(A > 0) goto fim
sub $s0, $zero, $s0 # A = |A|
sw $s0, 0($t0)       # MEM[t0] = A
fim:

.data
A:  .word -2
```

Line: 18 Column: 1 Show Line Numbers

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c081001	lui $8,0x00001001	8: lui $t0, 0x1001 # t0 = 0x10010000
	0x00400004	0x8d100000	lw $16,0x00000000($8)	9: lw $s0, 0($t0) # A = MEM[t0]
	0x00400008	0x00104fc2	srl $9,$16,0x0000001f	10: srl $t1, $s0, 31 # t1 = 00..(bit ..
	0x0040000c	0x11200002	beq $9,$0,0x00000002	11: beq $t1, $zero, fim # if(A > 0) goto..
	0x00400010	0x00108022	sub $16,$0,$16	12: sub $s0, $zero, $s0 # A = |A|
	0x00400014	0xad100000	sw $16,0x00000000($8)	13: sw $s0, 0($t0) # MEM[t0] = A

Labels

Label	Address
(global)	
main	0x00400000
prog13.asm	
fim	0x00400018
A	0x10010000

Data Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000002	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Mars Messages Run I/O

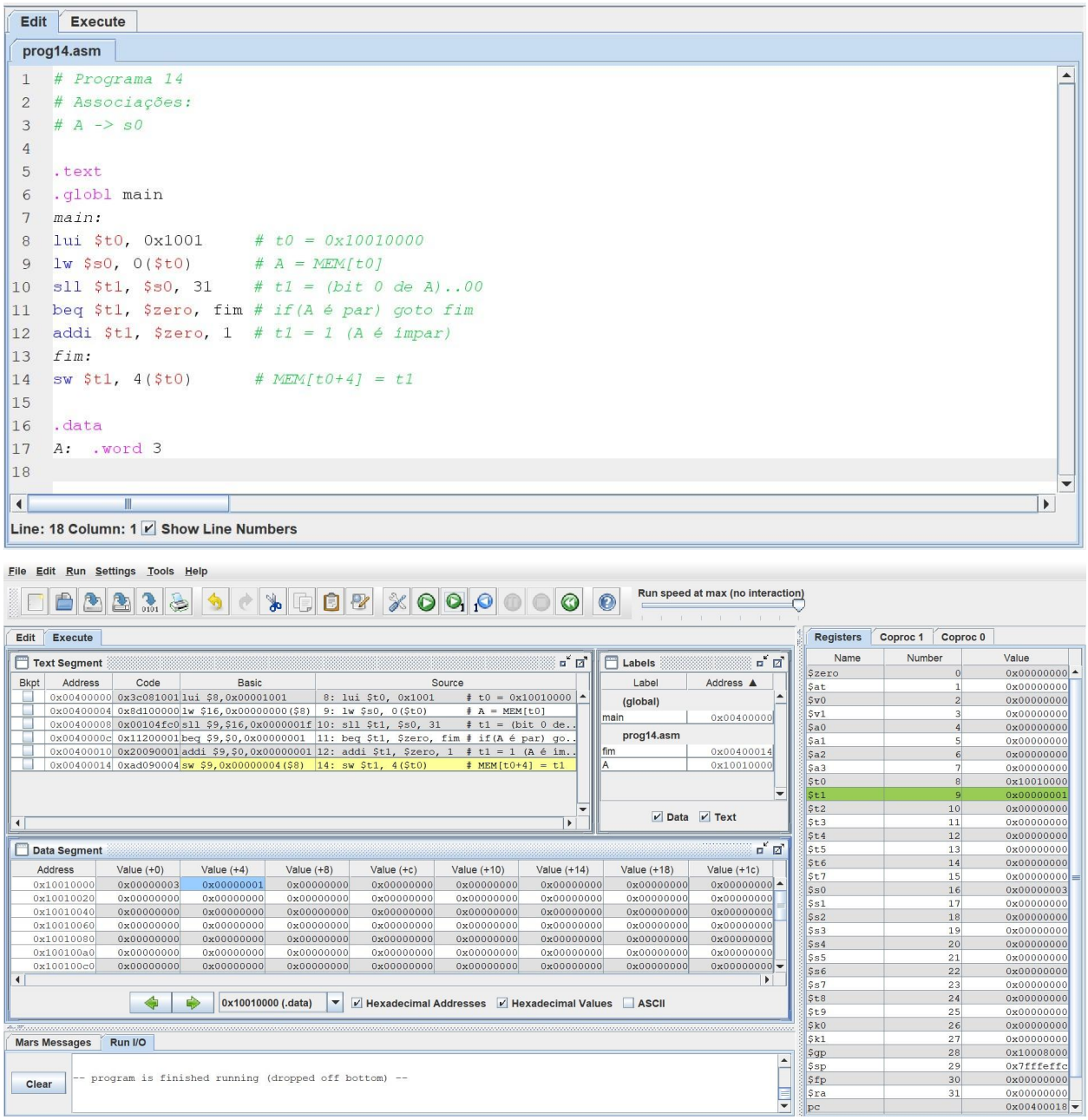
Clear

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

Registers Coproc 1 Coproc 0

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x10010000
$t1	9	0x00000001
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000002
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400018

Programa 14:



Edit Execute

prog14.asm

```
1  # Programa 14
2  # Associações:
3  # A -> s0
4
5  .text
6  .globl main
7  main:
8  lui $t0, 0x1001      # t0 = 0x10010000
9  lw $s0, 0($t0)       # A = MEM[t0]
10 sll $t1, $s0, 31     # t1 = (bit 0 de A)..00
11 beq $t1, $zero, fim  # if(A é par) goto fim
12 addi $t1, $zero, 1   # t1 = 1 (A é impar)
13 fim:
14 sw $t1, 4($t0)       # MEM[t0+4] = t1
15
16 .data
17 A:  .word 3
18
```

Line: 18 Column: 1 ☑ Show Line Numbers

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c081001	lui $8,0x00001001	8: lui $t0, 0x1001 # t0 = 0x10010000
	0x00400004	0x8d100000	lw $16,0x00000000($8)	9: lw $s0, 0($t0) # A = MEM[t0]
	0x00400008	0x00104fc0	sll $9,$16,0x0000001f	10: sll $t1, $s0, 31 # t1 = (bit 0 de..
	0x0040000c	0x11200001	beq $9,$0,0x00000001	11: beq $t1, $zero, fim # if(A é par) go..
	0x00400010	0x20090001	addi $9,$0,0x00000001	12: addi $t1, $zero, 1 # t1 = 1 (A é im..
	0x00400014	0xad090004	sw $9,0x00000004($8)	14: sw $t1, 4($t0) # MEM[t0+4] = t1

Labels

Label	Address ▲
(global)	
main	0x00400000
prog14.asm	
fim	0x00400014
A	0x10010000

☑ Data ☑ Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000003	0x00000001	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values ☐ ASCII

Mars Messages Run I/O

Clear

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

Registers Coproc 1 Coproc 0

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x10010000
$t1	9	0x00000001
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000003
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400018

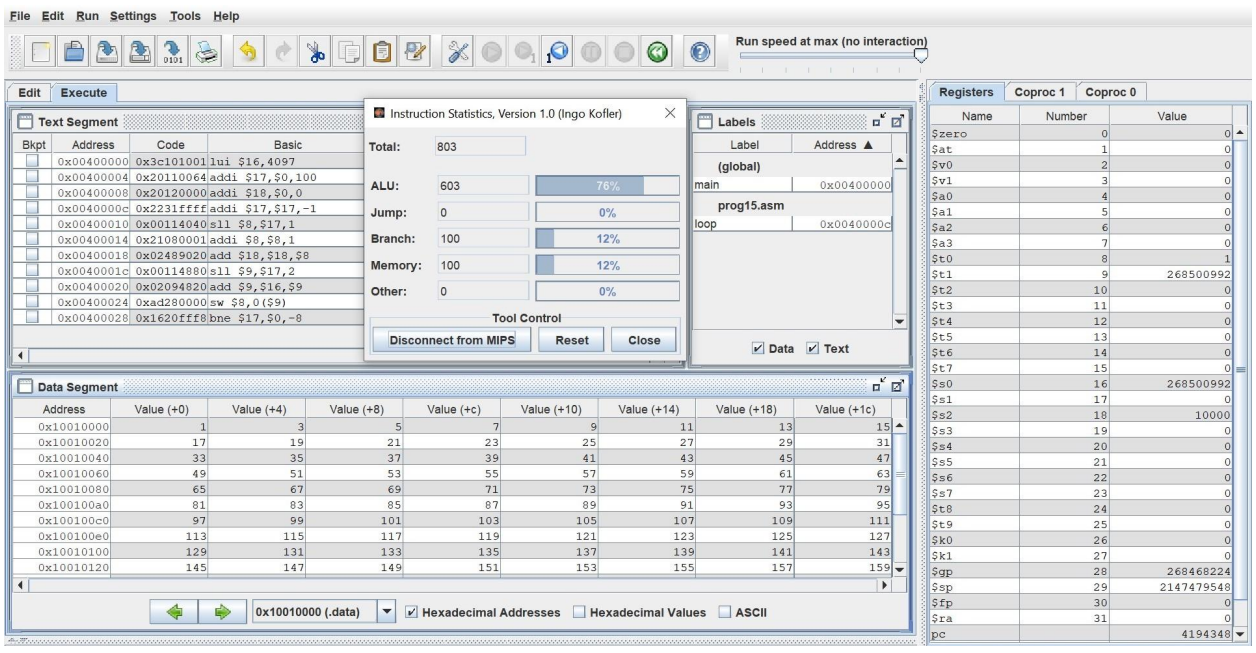
Programa 15:

Edit Execute

prog15.asm

```
1  # Programa 15
2  # Associações:
3  # vetor[0] -> $s0
4  # i -> $s1
5  # soma -> $s2
6
7  .text
8  .globl main
9  main:
10         lui $s0, 0x1001       # s0 = vetor[0]
11         addi $s1, $zero, 100  # i = 100
12         addi $s2, $zero, 0    # soma = 0
13 loop:
14         addi $s1, $s1, -1     # i = i - 1
15         sll $t0, $s1, 1       # t0 = 2i
16         addi $t0, $t0, 1      # t0 = 2i + 1
17         add $s2, $s2, $t0     # soma = soma + vetor[i]
18         sll $t1, $s1, 2       # t1 = 4i
19         add $t1, $s0, $t1     # t1 = end(vetor[i])
20         sw $t0, 0($t1)        # vetor[i] = 2i + 1
21         bne $s1, $zero, loop # if(i != 0) goto loop
22
```

Line: 22 Column: 1 ☑ Show Line Numbers



Programa 16:

Edit Execute

prog16.asm

```
1  # Programa 16
2  # Associações:
3  # x -> $s0
4  # y -> $s1
5  # z -> $s2
6
7  .text
8  .globl main
9  main:
10         lui $s0, 0x0018       # x = 0x00180000
11         ori $s0, $s0, 0x6A00  # x = 0x00186A00
12         lui $s1, 0x0001       # y = 0x00010000
13         ori $s1, $s1, 0x3880  # y = 0x00013880
14         lui $s2, 0x0006       # z = 0x00060000
15         ori $s2, $s2, 0x1A80  # z = 0x00061A80
16         mult $s1, $s2         # (hilo) = x * y
17         mflo $t0              # t0 = lo
18         div $t0, $s2          # (x*y) / z
19         mflo $t1              # t1 = lo
20
```

Line: 20 Column: 1 ☑ Show Line Numbers

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400004	0x36106a00	ori $16,$16,27136	11: ori $s0, $s0, 0x6A00 # x = 0..
	0x00400008	0x3c110001	lui $17,1	12: lui $s1, 0x0001 # y = 0..
	0x0040000c	0x36313880	ori $17,$17,14464	13: ori $s1, $s1, 0x3880 # y = 0..
	0x00400010	0x3c120006	lui $18,6	14: lui $s2, 0x0006 # z = 0..
	0x00400014	0x36521a80	ori $18,$18,6784	15: ori $s2, $s2, 0x1A80 # z = 0..
	0x00400018	0x02320018	mult $17,$18	16: mult $s1, $s2 # (hilo..
	0x0040001c	0x00004012	mflo $8	17: mflo $t0 # t0 = ..
	0x00400020	0x0112001a	div $8,$18	18: div $t0, $s2 # (x*y)..
	0x00400024	0x00004812	mflo $9	19: mflo $t1 # t1 = ..

Labels

Label	Address ▲
(global)	
main	0x00400000

☑ Data ☑ Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0	0	0	0	0	0	0	0
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0

0x10010000 (.data) ☑ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

Registers Coproc 1 Coproc 0

Name	Number	Value
$zero	0	0
$at	1	0
$v0	2	0
$v1	3	0
$a0	4	0
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	1935228928
$t1	9	4838
$t2	10	0
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	1600000
$s1	17	80000
$s2	18	400000
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194344

Mars Messages Run I/O

Clear

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

Programa 17:

Edit Execute

prog17.asm

```
1   # Programa 17
2   # Associações:
3   # x -> $s0
4   # y -> $s1
5   # k -> $s2
6
7   .text
8   .globl main
9   main:
10          lui $t0, 0x1001             # t0 = 0x10010000
11          lw $s0, 0($t0)              # x = MEM[t0]
12          lw $s1, 4($t0)              # y = MEM[t0+4]
13          addi $t1, $s1, 0            # t1 = y
14  multiplica:
15          add $s2, $s2, $s0           # k = k + x
16          addi $t1, $t1, -1           # t1 = t1 - 1
17          bne $t1, $zero, multiplica  # if(t1 != 0) goto multiplica
18  fim:
19          sw $s2, 8($t0)             # MEM[t0+8] = k
20  .data
21  x: .word 10
22  y: .word 4
23  k: .word 0
```

Line: 24 Column: 1 Show Line Numbers

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Text Segment

Bkpt	Address	Code	Basic		Source
	0x00400000	0x3c081001	lui $8,4097	10:	lui $t0, 0x1001
	0x00400004	0x8d100000	lw $16,0($8)	11:	lw $s0, 0($t0)
	0x00400008	0x8d110004	lw $17,4($8)	12:	lw $s1, 4($t0)
	0x0040000c	0x22290000	addi $9,$17,0	13:	addi $t1, $s1, 0
	0x00400010	0x02509020	add $18,$18,$16	15:	add $s2, $s2, $s0
	0x00400014	0x2129ffff	addi $9,$9,-1	16:	addi $t1, $t1, -1
	0x00400018	0x1520fffd	bne $9,$0,-3	17:	bne $t1, $zero, multiplica
	0x0040001c	0xad120008	sw $18,8($8)	19:	sw $s2, 8($t0)

Labels

Label	Address ▲
(global)	
main	0x00400000
prog17.asm	
multiplica	0x00400010
fim	0x0040001c
x	0x10010000
y	0x10010004
k	0x10010008

Data Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	10	4	40	0	0	0	0	0
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Registers Coproc 1 Coproc 0

Name	Number	Value
$zero	0	0
$at	1	0
$v0	2	0
$v1	3	0
$a0	4	0
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	268500992
$t1	9	0
$t2	10	0
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	10
$s1	17	4
$s2	18	40
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194336

Mars Messages Run I/O

Clear

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

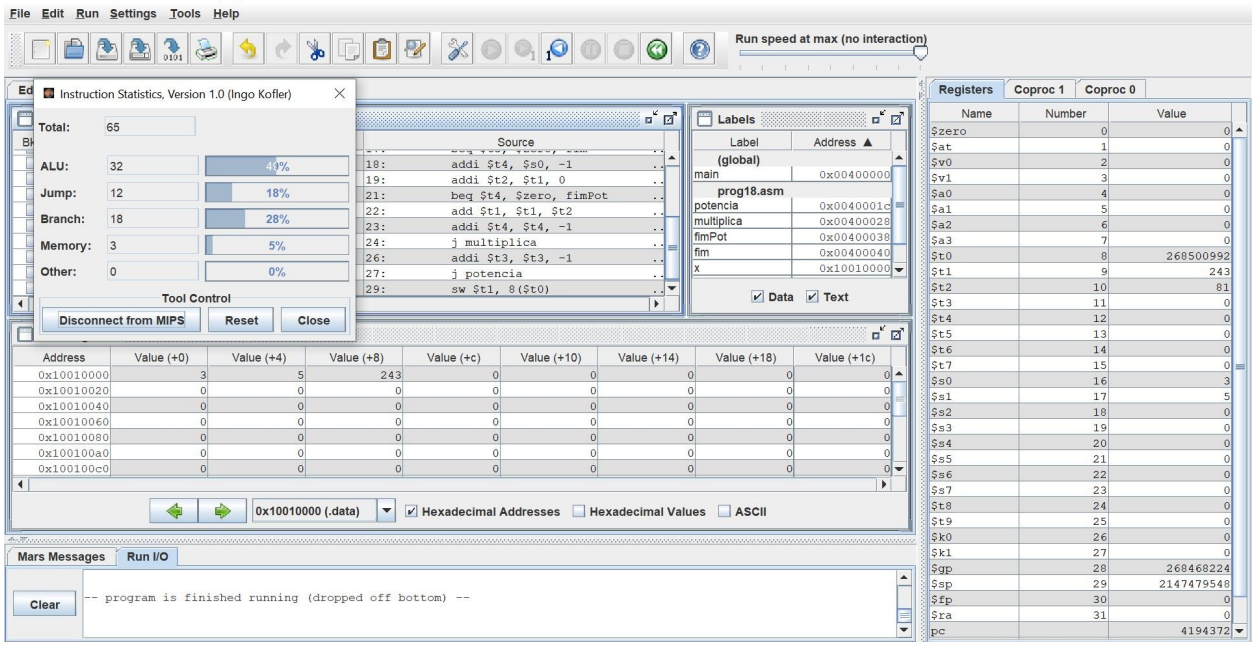
Programa 18:

Edit | Execute

prog18.asm

```
1  # Programa 18
2  # Associações:
3  # x -> $s0
4  # y -> $s1
5
6  .text
7  .globl main
8  main:
9          lui $t0, 0x1001              # t0 = 0x10010000
10         lw $s0, 0($t0)               # x = MEM[t0]
11         lw $s1, 4($t0)               # y = MEM[t0+4]
12         addi $t1, $zero, 1           # t1 = 1
13         beq $s1, $zero, fim          # if(y = 0) goto fim
14         addi $t1, $s0, 0             # t1 = x
15         addi $t3, $s1, -1            # t3 = y - 1
16 potencia:
17         beq $t3, $zero, fim          # if(t3 = 0) goto fim
18         addi $t4, $s0, -1            # t4 = x - 1
19         addi $t2, $t1, 0             # t2 = t1
20 multiplica:
21         beq $t4, $zero, fimPot       # if(t4 = 0) goto fimPot
22         add $t1, $t1, $t2            # t1 = t1 + t2
23         addi $t4, $t4, -1            # t4 = t4 - 1
24         j multiplica                 # goto multiplica
25 fimPot:
26         addi $t3, $t3, -1            # t3 = t3 - 1
27         j potencia                   # goto potencia
28 fim:
29         sw $t1, 8($t0)              # MEM[t0+8] = t1
30 .data
31 x: .word 3
32 y: .word 5
33 k: .word 0
34
```

Line: 34 Column: 1 ☑ Show Line Numbers



Parte 3

1) C
2) B
3) A
4) C
5) B
6) A
7) D
8) A
9) A
10) A

Parte 4

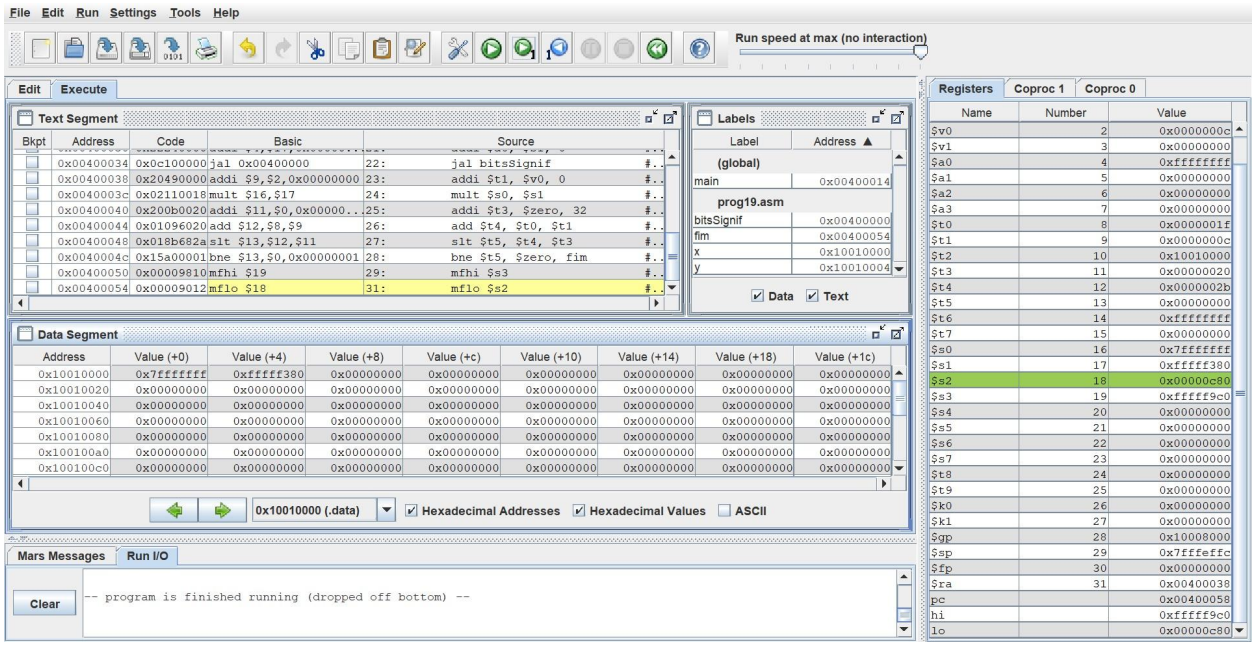
Programa 19:

Edit | Execute

prog19.asm

```
1  # Programa 19
2  # x -> $s0
3  # y -> $s1
4
5  .globl main
6  .text
7  bitsSignif:
8          sra $t6, $a0, 31          # t6 = a0 >> 31
9          sra $a0, $a0, 1           # a0 = a0 >> 1
10         addi $v0, $v0, 1          # v0 = v0 + 1
11         bne $a0, $t6, bitsSignif  # if(a0 != t6) goto bitsSignif
12         jr $ra                    # return
13 main:
14         lui $t2, 0x1001           # t2 = 0x10010000
15         lw $s0, 0($t2)            # x = MEM[t2]
16         lw $s1, 4($t2)            # y = MEM[t2+4]
17         addi $a0, $s0, 0          # a0 = x
18         jal bitsSignif            # call bitsSignif
19         addi $t0, $v0, 0          # t0 = bitsSignif(x)
20         addi $v0, $zero, 0        # v0 = 0
21         addi $a0, $s1, 0          # a0 = y
22         jal bitsSignif            # call bitsSignif
23         addi $t1, $v0, 0          # t1 = bitsSignif(y)
24         mult $s0, $s1             # x * y -> (hilo)
25         addi $t3, $zero, 32       # t3 = 32
26         add $t4, $t0, $t1         # t4 = t0 + t1
27         slt $t5, $t4, $t3         # if(t4 < 32) t5 = 1
28         bne $t5, $zero, fim       # if(t5 != 0) goto fim
29         mfhi $s3                  # s3 = hi
30 fim:
31         mflo $s2                  # s2 = lo
32
33 .data
34 x: .word 0x7FFFFFFF
35 y: .word -3200
36
```

Line: 36 Column: 1 ☑ Show Line Numbers


File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Bkpt Address Code Basic Source
0x00400034 0x0c100000 jal 0x00400000 22: jal bitsSignif #..
0x00400038 0x20490000 addi $9,$2,0x00000000 23: addi $t1, $v0, 0 #..
0x0040003c 0x02110018 mult $16,$17 24: mult $s0, $s1 #..
0x00400040 0x200b0020 addi $11,$0,0x00000... 25: addi $t3, $zero, 32 #..
0x00400044 0x01096020 add $12,$8,$9 26: add $t4, $t0, $t1 #..
0x00400048 0x018b682a slt $13,$12,$11 27: slt $t5, $t4, $t3 #..
0x0040004c 0x15a00001 bne $13,$0,0x00000001 28: bne $t5, $zero, fim #..
0x00400050 0x00009810 mfhi $19 29: mfhi $s3 #..
0x00400054 0x00009012 mflo $18 31: mflo $s2 #..
Labels
Label Address
(global)
main 0x00400014
prog19.asm
bitsSignif 0x00400000
fim 0x00400054
x 0x10010000
y 0x10010004
Data Text
Data Segment
Address Value (+0) Value (+4) Value (+8) Value (+c) Value (+10) Value (+14) Value (+18) Value (+1c)
0x10010000 0x7fffffff 0xffff380 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010020 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010040 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010060 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010080 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x100100a0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x100100c0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII
Mars Messages Run I/O
Clear
-- program is finished running (dropped off bottom) --
Registers Coproc 1 Coproc 0
Name Number Value
$v0 2 0x0000000c
$v1 3 0x00000000
$a0 4 0xffffffff
$a1 5 0x00000000
$a2 6 0x00000000
$a3 7 0x00000000
$t0 8 0x0000001f
$t1 9 0x0000000c
$t2 10 0x10010000
$t3 11 0x00000020
$t4 12 0x0000002b
$t5 13 0x00000000
$t6 14 0xffffffff
$t7 15 0x00000000
$s0 16 0x7fffffff
$s1 17 0xfffff380
$s2 18 0x00000c80
$s3 19 0xfffff9c0
$s4 20 0x00000000
$s5 21 0x00000000
$s6 22 0x00000000
$s7 23 0x00000000
$t8 24 0x00000000
$t9 25 0x00000000
$k0 26 0x00000000
$k1 27 0x00000000
$gp 28 0x10008000
$sp 29 0x7fffeffc
$fp 30 0x00000000
$ra 31 0x00400038
pc 0x00400058
hi 0xfffff9c0
lo 0x00000c80

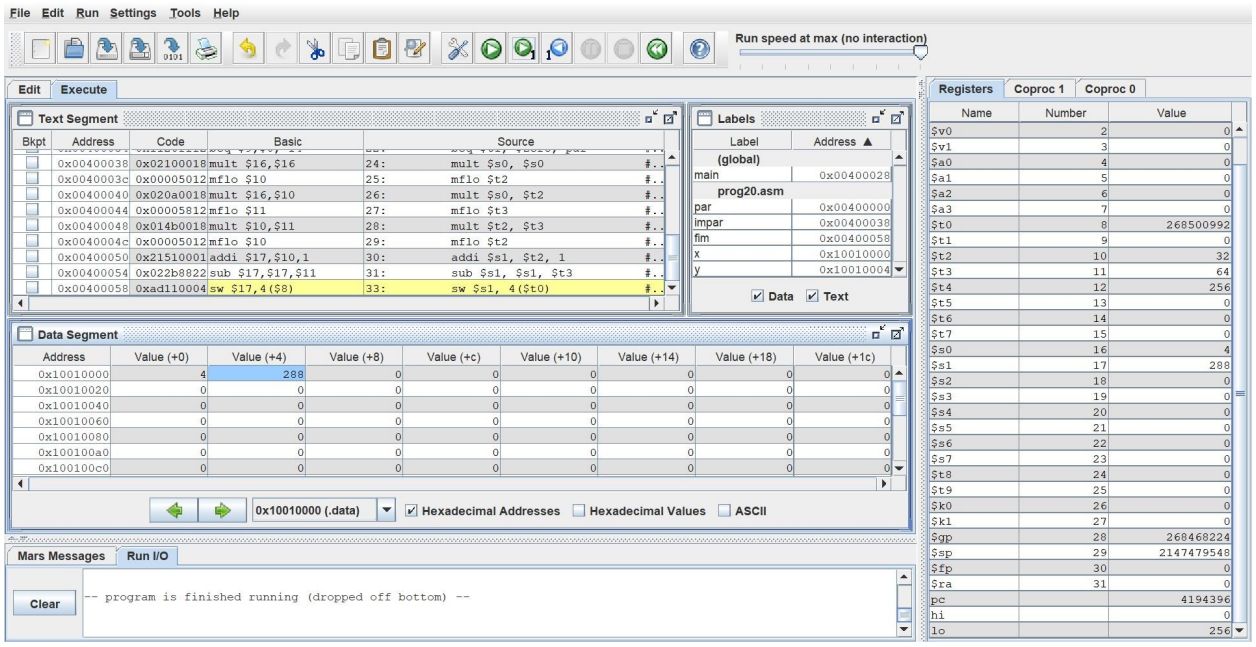
Programa 20:

Edit | Execute

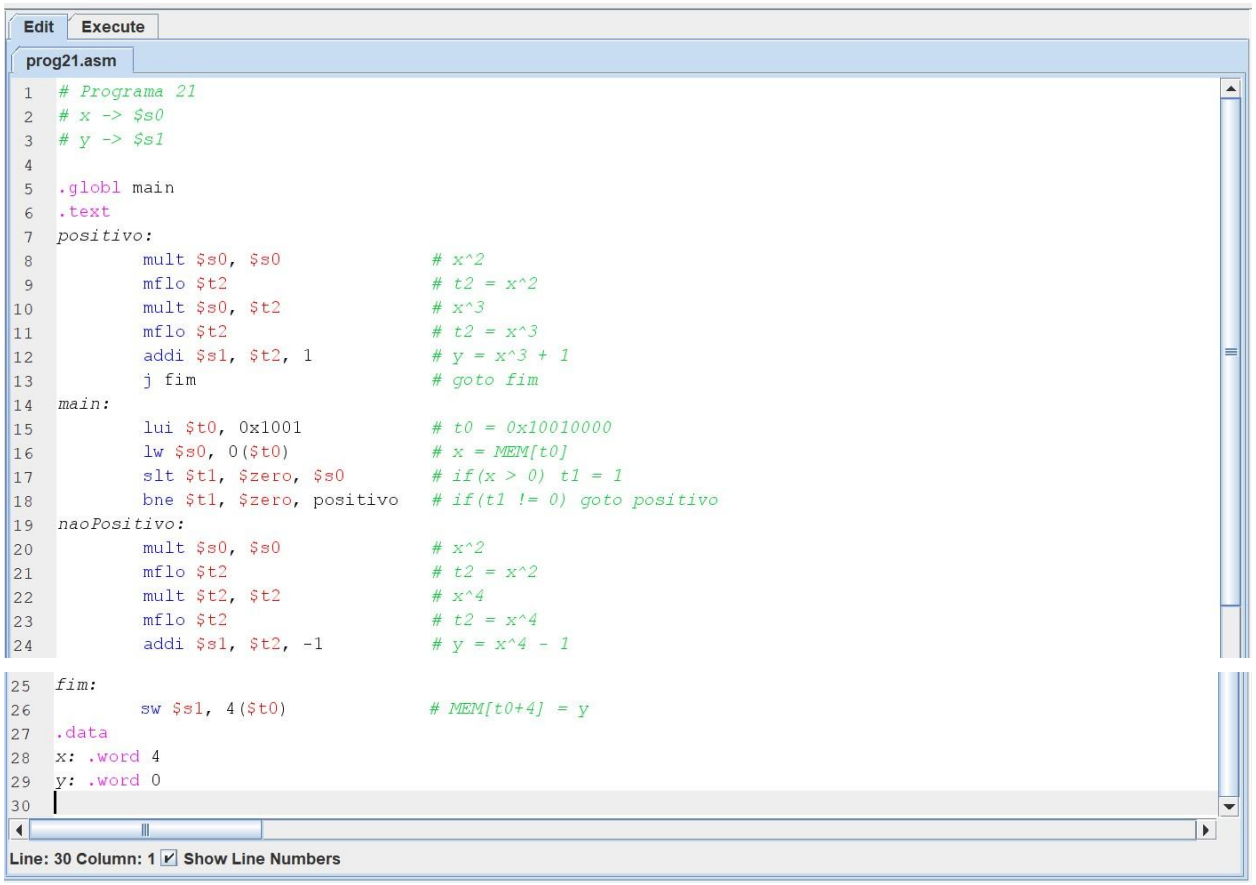
prog20.asm

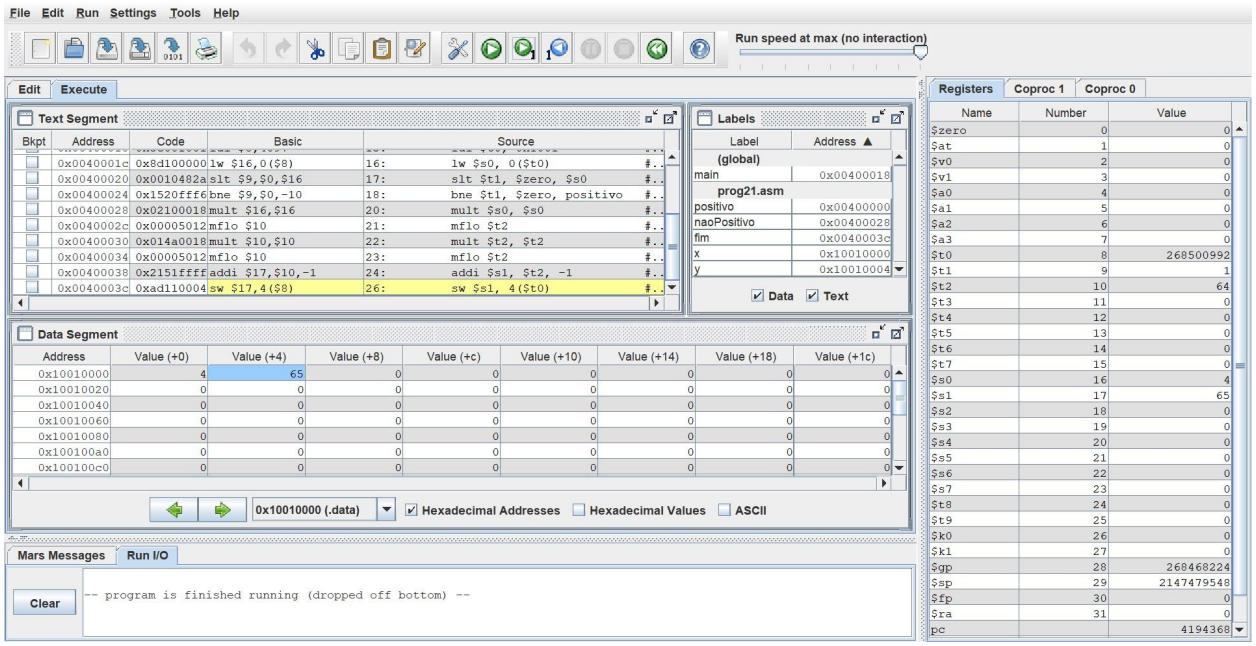
```
1  # Programa 20
2  # x -> $s0
3  # y -> $s1
4
5  .globl main
6  .text
7  par:
8          mult $s0, $s0              # x^2
9          mflo $t2                   # t2 = x^2
10         mult $s0, $t2              # x^3
11         mflo $t3                   # t3 = x^3
12         mult $t2, $t2              # x^4
13         mflo $t4                   # t4 = x^4
14         sll $t2, $t2, 1            # t2 = 2x^2
15         add $s1, $t3, $t4          # y = x^4 + x^3
16         sub $s1, $s1, $t2          # y = x^4 + x^3 - 2x^2
17         j fim                      # goto fim
18 main:
19         lui $t0, 0x1001            # t0 = 0x10010000
20         lw $s0, 0($t0)             # x = MEM[t0]
21         sll $t1, $s0, 31           # t1 = (bit 0 de x)..00
22         beq $t1, $zero, par        # if(x é par) goto par
23 impar:
24         mult $s0, $s0              # x^2
25         mflo $t2                   # t2 = x^2
26         mult $s0, $t2              # x^3
27         mflo $t3                   # t3 = x^3
28         mult $t2, $t3              # x^5
29         mflo $t2                   # t2 = x^5
30         addi $s1, $t2, 1           # y = x^5 + 1
31         sub $s1, $s1, $t3          # y = x^5 - x^3 + 1
32 fim:
33         sw $s1, 4($t0)             # MEM[t0+4] = y
34 .data
35 x: .word 4
36 y: .word 0
37
```

Line: 37 Column: 1 ☑ Show Line Numbers


File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Bkpt Address Code Basic Source
0x00400038 0x02100018 mult $16,$16 24: mult $s0, $s0
0x0040003c 0x00005012 mflo $10 25: mflo $t2
0x00400040 0x020a0018 mult $16,$10 26: mult $s0, $t2
0x00400044 0x00005812 mflo $11 27: mflo $t3
0x00400048 0x014b0018 mult $10,$11 28: mult $t2, $t3
0x0040004c 0x00005012 mflo $10 29: mflo $t2
0x00400050 0x21510001 addi $17,$10,1 30: addi $s1, $t2, 1
0x00400054 0x022b8822 sub $17,$17,$11 31: sub $s1, $s1, $t3
0x00400058 0xad110004 sw $17,4($8) 33: sw $s1, 4($t0)
Labels
Label Address
(global)
main 0x00400028
prog20.asm
par 0x00400000
impar 0x00400038
fim 0x00400058
x 0x10010000
y 0x10010004
Data Text
Data Segment
Address Value (+0) Value (+4) Value (+8) Value (+c) Value (+10) Value (+14) Value (+18) Value (+1c)
0x10010000 4 288 0 0 0 0 0 0
0x10010020 0 0 0 0 0 0 0 0
0x10010040 0 0 0 0 0 0 0 0
0x10010060 0 0 0 0 0 0 0 0
0x10010080 0 0 0 0 0 0 0 0
0x100100a0 0 0 0 0 0 0 0 0
0x100100c0 0 0 0 0 0 0 0 0
0x10010000 (.data)
Hexadecimal Addresses
Hexadecimal Values
ASCII
Mars Messages
Run I/O
Clear
-- program is finished running (dropped off bottom) --
Registers
Coproc 1
Coproc 0
Name Number Value
$v0 2 0
$v1 3 0
$a0 4 0
$a1 5 0
$a2 6 0
$a3 7 0
$t0 8 268500992
$t1 9 0
$t2 10 32
$t3 11 64
$t4 12 256
$t5 13 0
$t6 14 0
$t7 15 0
$s0 16 4
$s1 17 288
$s2 18 0
$s3 19 0
$s4 20 0
$s5 21 0
$s6 22 0
$s7 23 0
$t8 24 0
$t9 25 0
$k0 26 0
$k1 27 0
$gp 28 268468224
$sp 29 2147479548
$fp 30 0
$ra 31 0
pc 4194396
hi 0
lo 256

Programa 21:


Edit Execute
prog21.asm
1 # Programa 21
2 # x -> $s0
3 # y -> $s1
4
5 .globl main
6 .text
7 positivo:
8 mult $s0, $s0 # x^2
9 mflo $t2 # t2 = x^2
10 mult $s0, $t2 # x^3
11 mflo $t2 # t2 = x^3
12 addi $s1, $t2, 1 # y = x^3 + 1
13 j fim # goto fim
14 main:
15 lui $t0, 0x1001 # t0 = 0x10010000
16 lw $s0, 0($t0) # x = MEM[t0]
17 slt $t1, $zero, $s0 # if(x > 0) t1 = 1
18 bne $t1, $zero, positivo # if(t1 != 0) goto positivo
19 naoPositivo:
20 mult $s0, $s0 # x^2
21 mflo $t2 # t2 = x^2
22 mult $t2, $t2 # x^4
23 mflo $t2 # t2 = x^4
24 addi $s1, $t2, -1 # y = x^4 - 1
25 fim:
26 sw $s1, 4($t0) # MEM[t0+4] = y
27 .data
28 x: .word 4
29 y: .word 0
30
Line: 30 Column: 1 Show Line Numbers


File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Bkpt Address Code Basic Source
0x0040001c 0x8d100000 lw $16,0($8) 16: lw $s0, 0($t0) #..
0x00400020 0x0010482a slt $9,$0,$16 17: slt $t1, $zero, $s0 #..
0x00400024 0x1520fff6 bne $9,$0,-10 18: bne $t1, $zero, positivo #..
0x00400028 0x02100018 mult $16,$16 20: mult $s0, $s0 #..
0x0040002c 0x00005012 mflo $10 21: mflo $t2 #..
0x00400030 0x014a0018 mult $10,$10 22: mult $t2, $t2 #..
0x00400034 0x00005012 mflo $10 23: mflo $t2 #..
0x00400038 0x2151ffff addi $17,$10,-1 24: addi $s1, $t2, -1 #..
0x0040003c 0xad110004 sw $17,4($8) 26: sw $s1, 4($t0) #..
Labels
Label Address
(global)
main 0x00400018
prog21.asm
positivo 0x00400000
naoPositivo 0x00400028
fim 0x0040003c
x 0x10010000
y 0x10010004
Data Text
Data Segment
Address Value (+0) Value (+4) Value (+8) Value (+c) Value (+10) Value (+14) Value (+18) Value (+1c)
0x10010000 4 65 0 0 0 0 0 0
0x10010020 0 0 0 0 0 0 0 0
0x10010040 0 0 0 0 0 0 0 0
0x10010060 0 0 0 0 0 0 0 0
0x10010080 0 0 0 0 0 0 0 0
0x100100a0 0 0 0 0 0 0 0 0
0x100100c0 0 0 0 0 0 0 0 0
0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII
Mars Messages Run I/O
Clear
-- program is finished running (dropped off bottom) --
Registers Coproc 1 Coproc 0
Name Number Value
$zero 0 0
$at 1 0
$v0 2 0
$v1 3 0
$a0 4 0
$a1 5 0
$a2 6 0
$a3 7 0
$t0 8 268500992
$t1 9 1
$t2 10 64
$t3 11 0
$t4 12 0
$t5 13 0
$t6 14 0
$t7 15 0
$s0 16 4
$s1 17 65
$s2 18 0
$s3 19 0
$s4 20 0
$s5 21 0
$s6 22 0
$s7 23 0
$t8 24 0
$t9 25 0
$k0 26 0
$k1 27 0
$gp 28 268468224
$sp 29 2147479548
$fp 30 0
$ra 31 0
pc 4194368