

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:

Editor window showing the assembly code for Programa 1:

```

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

Mars Debugger interface showing the assembly code, registers, memory segments, and messages.

Registers:

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	0x00000009
\$t2	10	0xffffffff
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000002
\$s1	17	0x00000001
\$s2	18	0x00000004
\$s3	19	0x00000005
\$s4	20	0xfffffffffc
\$s5	21	0xfffffffffb
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$fp	28	0x10008000
\$sp	29	0x7ffffefc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x04000028

Text Segment:

Bkpt	Address	Code	Basic	Source
	0x00400004	0x20110003	addi \$17,\$0,0x0000..	14: addi \$s1, \$zero, 3 # b = 3
	0x00400004	0x20120004	addi \$18,\$0,0x0000..	15: addi \$s2, \$zero, 4 # c = 4
	0x00400005	0x20130005	addi \$19,\$0,0x0000..	16: addi \$s3, \$zero, 5 # d = 5
	0x00400014	0x20114020	add \$8,\$16,\$17	17: add \$t0, \$s0, \$s1 # t0 = a + b
	0x00400014	0x202534820	add \$9,\$18,\$19	18: add \$t1, \$s2, \$s3 # t1 = c + d
	0x00400018	0x20109a022	sub \$20,\$8,\$9	19: sub \$s4, \$t0, \$t1 # x = t0 - t1
	0x0040001c	0x202115022	sub \$10,\$16,\$17	20: sub \$t2, \$s0, \$s1 # t2 = a - b
	0x00400020	0x20154a820	add \$21,\$10,\$20	21: add \$s5, \$t2, \$s4 # y = t2 + x
	0x00400024	0x202958822	sub \$17,\$20,\$21	22: sub \$s1, \$s4, \$s5 # b = x - y

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

Mars Messages:

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

Programa 2:

Editor window (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)

Registers window:

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	0x7ffffeffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400014

Text Segment window:

Bkpt	Address	Code	Basic	Source
	0x00400000	0x20100001	addi \$16,\$0,0x0000...	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	0x21111000f	addi \$17,\$8,0x0000...	13: addi \$s1, \$t0, 15 # y = 5x + 15

Data Segment window:

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

Mars Messages window:

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

Run I/O window:

Clear

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 \$s1, \$t1 # t1 = 16y	20: add \$t1, \$t1 # t1 = 16y
	0x0040002c	0x01294820	add \$s1, \$t1 # t1 = 32y	21: add \$t1, \$t1 # t1 = 32y
	0x00400030	0x01294820	add \$s1, \$t1 # t1 = 64y	22: add \$t1, \$t1 # t1 = 64y
	0x00400034	0x01314820	add \$s1, \$s1 # t1 = 65y	23: add \$t1, \$t1, \$s1 # t1 = 65y
	0x00400038	0x01314820	add \$s1, \$s1 # t1 = 66y	24: add \$t1, \$t1, \$s1 # t1 = 66y
	0x0040003c	0x01314820	add \$s1, \$s1 # t1 = 67y	25: add \$t1, \$t1, \$s1 # t1 = 67y
	0x00400040	0x001095020	add \$t2, \$t0, \$t1 # t2 = t0 + t1	26: add \$t2, \$t0, \$t1 # t2 = t0 + t1
	0x00400044	0x01445020	add \$t2, \$t2 # t2 = 2t2	27: add \$t2, \$t2, \$t2 # t2 = 2t2
	0x00400048	0x01449020	add \$s2, \$t2, \$t2 # z = 4t2	28: add \$s2, \$t2, \$t2 # z = 4t2

Registers

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
\$gp	28	0x10008000
\$sp	29	0x7fffffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x0040004c

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 4:

Screenshot of the MARS assembly debugger interface showing the assembly code for Program 4 and its execution results.

Assembly Code (prog4.asm):

```

1 # Programa 4
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 sll $t0, $s0, 4 # t0 = 16x
13 sub $t0, $t0, $s0 # t0 = 15x
14 sll $t1, $s1, 6 # t1 = 64y
15 add $t1, $t1, $s1 # t1 = 65y
16 add $t1, $t1, $s1 # t1 = 66y
17 add $t1, $t1, $s1 # t1 = 67y
18 add $t2, $t0, $t1 # t2 = t0 + t1
19 sll $s2, $t2, 2 # z = 4t2
20
21

```

Registers:

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	0x00000139
\$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	0x7ffffeffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400028

Labels:

Label	Address
(global)	0x00400000
main	0x00400000

Text Segment:

Bkpt	Address	Code	Basic	Source
	0x00400004	0x20110004	addi \$t7,\$0,0x00000..	.11: addi \$s1, \$zero, 4 # y = 4
	0x00400008	0x01041000	sll \$s0,\$16,0x00000004	.12: sll \$t0, \$s0, 4 # t0 = 16x
	0x0040000c	0x01104022	sub \$s0,\$s16	.13: sub \$t0, \$t0, \$s0 # t0 = 15x
	0x00400010	0x01149800	sll \$s0,\$17,0x00000006	.14: sll \$t1, \$s1, 6 # t1 = 64y
	0x00400014	0x01314820	add \$s0,\$s17	.15: add \$t1, \$t1, \$s1 # t1 = 65y
	0x00400018	0x01314820	add \$s0,\$s17	.16: add \$t1, \$t1, \$s1 # t1 = 66y
	0x00400020	0x01314820	add \$s0,\$s17	.17: add \$t1, \$t1, \$s1 # t1 = 67y
	0x00400020	0x10950200	add \$t0,\$t1	.18: add \$t2, \$t0, \$t1 # t2 = t0 + t1
	0x00400024	0x00aa9080	sll \$s2,\$t2,2	.19: sll \$s2, \$t2, 2 # z = 4t2

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

Mars Messages:

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

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, 0xD40 # 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)

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

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x34080001	ori \$s0,\$s0,1	10: ori \$t0, \$zero, 0x1 # t0 = 0x00000001
	0x00400004	0x00084400	sll \$s0,\$s0,16	11: sll \$t0, \$t0, 16 # t0 = 0x00010000
	0x00400008	0x351086a0	ori \$s0,\$t0,0x86a0	12: ori \$s0, \$t0, 0x86A0 # x = 0x000186A0
	0x0040000c	0x34080003	ori \$s1,\$s0,3	13: ori \$t0, \$zero, 0x3 # t0 = 0x00000003
	0x00400010	0x00084400	sll \$s1,\$s1,16	14: sll \$t0, \$t0, 16 # t0 = 0x00030000
	0x00400014	0x35110d40	ori \$s2,\$s0,0xd40	15: ori \$s2, \$s0, 0xD40 # y = 0x00030D40
	0x00400018	0x02119020	add \$s2,\$s0,\$s1	16: add \$s2, \$s0, \$s1 # z = x + y

Labels

Label	Address
(global)	
main	0x00400000

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

Mars Messages Run I/O

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

Programa 6:

File 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 = 0xFFFFFFFF
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
18

```

Line: 19 Column: 1 Show Line Numbers

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$s0	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	0x7feeb07f
\$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	0x7ffffeffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x04000020

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x34087fff	ori \$8,\$0,0x00007fff	10: ori \$t0, \$zero, 0x7FFF # t0 = 0x0000..
	0x00400004	0x00084400	sll \$t0, \$t0, 16	11: sll \$t0, \$t0, 16 # t0 = 0x7FFF0000
	0x00400008	0x3510ffff	ori \$16,\$8,0x0000ffff	12: ori \$s0, \$t0, 0xFFFF # x = 0xFFFFFFFF
	0x0040000c	0x34080004	ori \$8,\$0,0x00000004	13: ori \$t0, \$zero, 0x4 # t0 = 0x00000004
	0x00400010	0x00084400	sll \$t0, \$t0, 16	14: sll \$t0, \$t0, 16 # t0 = 0x00040000
	0x00400014	0x351193e0	ori \$17,\$8,0x000093e0	15: ori \$s1, \$t0, 0x93E0 # y = 0x000493E0
	0x00400018	0x000114880	sll \$s1, \$s1, 2	16: sll \$t1, \$s1, 2 # t1 = 4y
	0x0040001c	0x02099022	sub \$s2, \$s0, \$t1	17: sub \$s2, \$s0, \$t1 # z = x - 4y

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:

The screenshot shows the Mars 32-bit simulator interface with the following components:

- Editor Tab:** Displays the assembly code for "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
```
- Registers Tab:** Shows the register values for the program.

Name	Number	Value
Szero	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	0xffffffff
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10000800
\$sp	29	0x7ffffefc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x0040000e
- Memory Tab:** Displays the memory dump for the Text Segment and Data Segment.
- Mars Messages Tab:** Displays the message: "-- program is finished running (dropped off bottom) --".

Programa 8:

Editor window showing assembly code:

```

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

Mars Debugger interface showing assembly and memory dump:

File Edit Run Settings Tools Help

Run speed at max (no interaction)

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	0x12345678
\$t1	9	0x00000012
\$t2	10	0x00000034
\$t3	11	0x00000056
\$t4	12	0x00000078
\$t5	13	0x00001234
\$t6	14	0x00005678
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffffeffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400028

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400004	0x00084400	sll \$8, \$8, 16	# \$8 = 0x12340000
	0x00400008	0x35085678	ori \$8,\$8,0x00005678	# \$8 = 0x12345678
	0x0040000c	0x350dffff	ori \$13,\$8,0x0000ffff	# \$13 = 0x1234FFFF
	0x00400010	0x000d6c02	srl \$13,\$13,16	# \$13 = 0x1234
	0x00400014	0x000d4a02	srl \$9,\$13,0x00000000	# \$9 = 0x12
	0x00400018	0x310a0fff	andi \$10,\$13,0x0000..	.10: andi \$10,\$13,0x0000..
	0x0040001c	0x310a0fff	andi \$14,\$8,0x0000f..	.11: andi \$14,\$8,0x0000f..
	0x00400020	0x00065a02	srl \$11,\$14,0x00000..	.12: srl \$11,\$14,0x00000..
	0x00400024	0x310c0fff	andi \$12,\$14,0x0000..	.13: andi \$12,\$14,0x0000..
	0x00400028	0x310c0fff	andi \$12,\$14,0x0000..	.14: andi \$12,\$14,0x0000..
	0x0040002c	0x310c0fff	andi \$12,\$14,0x0000..	.15: andi \$12,\$14,0x0000..

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

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

Clear

Programa 9:

prog9.asm

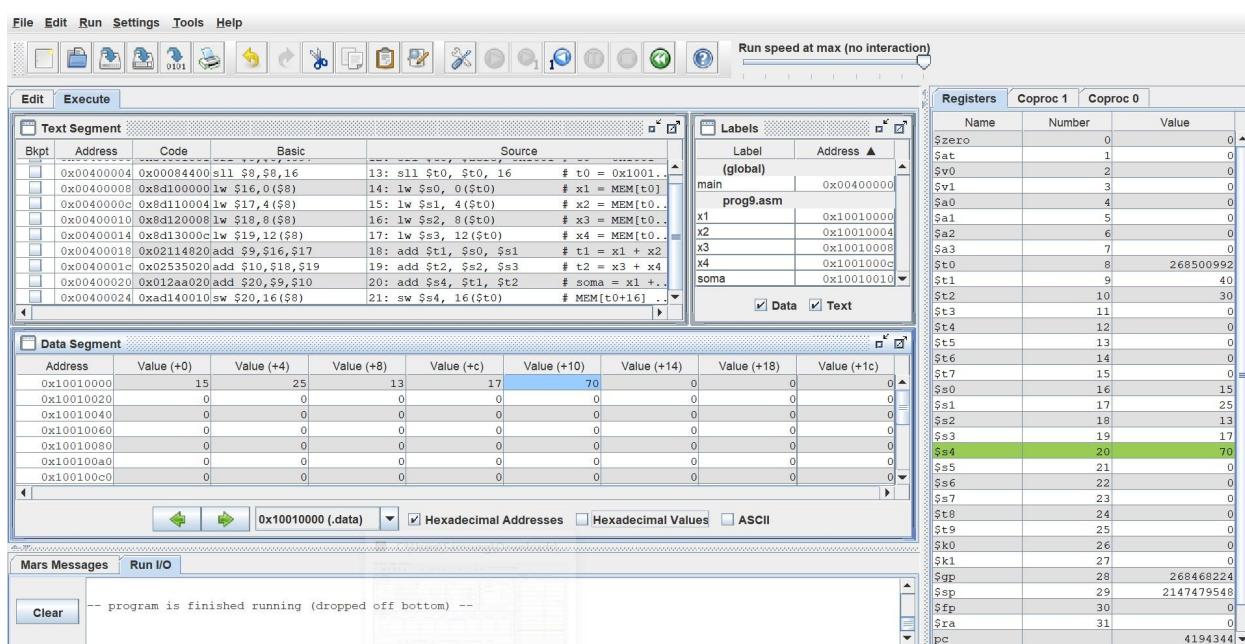
```

1  # Programa 9
2  # Associações:
3  # x1 -> s0
4  # x2 -> s1
5  # x3 -> s2
6  # x4 -> s3
7  # soma -> s4
8
9 .text
10 .globl main
11 main:
12 ori $t0, $zero, 0x1001 # t0 = 0x1001
13 sll $t0, $t0, 16        # t0 = 0x10010000
14 lw $s0, 0($t0)          # x1 = MEM[t0]
15 lw $s1, 4($t0)          # x2 = MEM[t0+4]
16 lw $s2, 8($t0)          # x3 = MEM[t0+8]
17 lw $s3, 12($t0)         # x4 = MEM[t0+12]
18 add $t1, $s0, $s1       # t1 = x1 + x2
19 add $t2, $s2, $s3       # t2 = x3 + x4
20 add $s4, $t1, $t2       # soma = x1 + x2 + x3 + x4
21 sw $s4, 16($t0)         # MEM[t0+16] = soma

22
23 .data
24 x1: .word 15
25 x2: .word 25
26 x3: .word 13
27 x4: .word 17
28 soma: .word -1
29

```

Line: 29 Column: 1 Show Line Numbers



Programa 10:

Edit Execute

prog10.asm

```

1 # Programa 10
2 # Associações:
3 # x -> $0
4 # z -> $1
5 # y -> $2
6
7 .text
8 .globl main
9 main:
10 ori $t0, $zero, 0x1001 # t0 = 0x1001
11 sll $t0, $t0, 16 # t0 = 0x10010000
12 lw $s0, 0($t0) # x = MEM[t0]
13 lw $s1, 4($t0) # z = MEM[t0+4]
14 sll $t1, $s0, 7 # t1 = 128x
15 sub $t1, $t1, $s0 # t1 = 127x
16 sll $t2, $s1, 6 # t2 = 64z
17 add $t2, $t2, $s1 # t2 = 65z
18 sub $t3, $t1, $t2 # t3 = t1 - t2
19 addi $s2, $t3, 1 # y = 127x - 65z + 1
20 sw $s2, 8($t0) # MEM[t0+8] = y

21
22 .data
23 x: .word 5
24 z: .word 7
25 y: .word 0 # esse valor deverá ser sobreescrito após a execução do programa
26

```

Line: 26 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
	0x00400008	0x8d100000	lw \$16, 0(\$8)	12: lw \$s0, 0(\$t0) # x = MEM[t0]
	0x0040000c	0x8d110004	lw \$17, 4(\$8)	13: lw \$s1, 4(\$t0) # z = MEM[t0+4]
	0x00400010	0x001049c0	sll \$s1, \$s0, 7	14: sll \$t1, \$s0, 7 # t1 = 128x
	0x00400014	0x001304822	sub \$s1, \$t1, \$s0	15: sub \$t1, \$t1, \$s0 # t1 = 127x
	0x00400018	0x00115180	sll \$t1, \$s0, 6	16: sll \$t2, \$s1, 6 # t2 = 64z
	0x0040001c	0x001515020	add \$t2, \$s1, \$t1	17: add \$t2, \$t2, \$s1 # t2 = 65z
	0x00400020	0x0012a5822	sub \$s2, \$t1, \$t2	18: sub \$t3, \$t1, \$t2 # t3 = t1 - t2
	0x00400024	0x21720001	addi \$s2, \$t3, 1	19: addi \$s2, \$t3, 1 # y = 127x - ..
	0x00400028	0xadx120008	sw \$s2, 8(\$t0)	20: sw \$s2, 8(\$t0) # MEM[t0+8] = ..

Labels

Label	Address
(global)	0x00400000
main	0x00400000
prog10.asm	
x	0x10010000
z	0x10010004
y	0x10010008

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+c)
0x10010000	5	7	181	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

Registers

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	635
\$t2	10	455
\$t3	11	180
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	5
\$s1	17	7
\$s2	18	181
\$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		4194348

Mars Messages Run I/O

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

Programa 11:

Editor window showing the assembly code for Program 11:

```

Edit Execute
prog11.asm
1 # Programa 11
2 # Associações:
3 # x -> $t0
4 # z -> $t1
5 # y -> $t2
6
7 .text
8 .globl main
9 main:
10 ori $t0, $zero, 0x1001 # t0 = 0x1001
11 sll $t0, $t0, 16 # t0 = 0x10010000
12 lw $s0, 0($t0) # x = MEM[t0]
13 lw $s1, 4($t0) # z = MEM[t0+4]
14 sub $t1, $s0, $s1 # t1 = x - z
15 ori $t2, $zero, 0x4 # t2 = 0x4
16 sll $t2, $t2, 16 # t2 = 0x00040000
17 ori $t2, $t2, 0x93E0 # t2 = 0x000493E0
18 add $s2, $t1, $t2 # y = x - z + 300000
19 sw $s2, 8($t0) # MEM[t0+8] = y
20
21 .data
22 x: .word 100000
23 z: .word 200000
24 y: .word 0 # esse valor deverá ser sobreescrito após a execução do programa.
25

```

Line: 25 Column: 1 Show Line Numbers

Mars Debugger interface showing the assembly code, memory dump, registers, and messages.

Registers:

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	-100000
\$t2	10	300000
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	100000
\$s1	17	200000
\$s2	18	200000
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194344

Messages:

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

Programa 12:

Edit Execute

prog12.asm

```

1 # Programa 12
2 # Associações:
3 # x -> s0
4 # k -> sl
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)

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x34081001	ori \$t0, \$zero, 0x10000001	9: ori \$t0, \$zero, 0x1001 # t0 = 0x1001
	0x00400004	0x00084400	sll \$t0, \$t0, 16	10: sll \$t0, \$t0, 16 # t0 = 0x10010000
	0x00400008	0x35080000	ori \$t0, \$t0, 0xC	11: ori \$t0, \$t0, 0xC # t0 = 0x1001000C
	0x0040000c	0x8d100000	lw \$s0, 0(\$t0)	12: lw \$s0, 0(\$t0) # x = MEM[t0]
	0x00400010	0x8e090000	lw \$s0, 0(\$s0)	13: lw \$s0, 0(\$s0) # t1 = MEM[x]
	0x00400014	0x8d290000	lw \$s1, 0(\$t1)	14: lw \$s1, 0(\$t1) # t1 = MEM[MEM[x]]
	0x00400018	0x8d310000	sll \$s1, \$s1, 1	15: sll \$s1, \$s1, 1 # k = 2k
	0x0040001c	0x20118840	sw \$s1, 0(\$t1)	17: sw \$s1, 0(\$t1) # MEM[MEM[MEM[x]]] = 2k

Labels

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

Registers

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	0x7ffffefc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400024

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

Mars Messages Run I/O

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

Programa 13:

Edit Execute

prog13.asm

```

1 # Programa 13
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 srl $t1, $s0, 31     # t1 = 00..(bit 32 de A)
11 beq $t1, $zero, fim # if(A > 0) goto fim
12 sub $s0, $zero, $s0 # A = |A|
13 sw $s0, 0($t0)       # MEM[t0] = A
14 fim:
15
16 .data
17 A: .word -2
18

```

Line: 18 Column: 1 Show Line Numbers

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Text Segment

Bkpt	Address	Code	Basic	Source
0x00400000	0x3c081001	lui \$s0, 0x00001001		8: lui \$t0, 0x1001 # t0 = 0x10010000
0x00400004	0x8d1000001	lw \$s1, 0(\$t0)		9: lw \$s0, 0(\$t0) # A = MEM[t0]
0x00400008	0x00104fc2	srl \$t1, \$s0, 31		10: srl \$t1, \$s0, 31 # t1 = 00..(bit ..)
0x0040000c	0x11200002	beg \$s0, \$0,0x00000002		11: beq \$t1, \$zero, fim # if(A > 0) goto ..
0x00400010	0x00108022	sub \$s0, \$zero, \$s0	# A = A	12: sub \$s0, \$zero, \$s0 # A = A
0x00400014	0xad100000	sw \$s0, 0(\$t0)		13: sw \$s0, 0(\$t0) # MEM[t0] = A

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	0x7ffffefc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400018

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)
0x10010000	0x00000002	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Hexadecimal Addresses Hexadecimal Values ASCII

Mars Messages Run I/O

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

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 ímpar)
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)

Registers Coproc 1 Coproc 0

Name	Number	Value
Szero	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	0x7ffffefc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400018

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c081001	lui \$t0, 0x1001	8: lui \$t0, 0x1001 # t0 = 0x10010000
	0x00400004	0x8d100000	lw \$s0, 0(\$t0)	9: lw \$s0, 0(\$t0) # A = MEM[t0]
	0x00400008	0x00104fc0	sll \$t1, \$s0, 31	10: sll \$t1, \$s0, 31 # t1 = (bit 0 de A)..00
	0x0040000c	0x11200001	beq \$t1, \$zero, fim	11: beq \$t1, \$zero, fim # if(A é par) go...
	0x00400010	0x20090001	addi \$t1, \$zero, 1	12: addi \$t1, \$zero, 1 # t1 = 1 (A é impar)
	0x00400014	0xd0ad0004	sw \$t1, 4(\$t0)	14: sw \$t1, 4(\$t0) # MEM[t0+4] = t1

Labels

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

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+c)
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

Mars Messages Run I/O

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

Programa 15:

Editor window (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

Tool window (Instruction Statistics, Version 1.0):

Total	ALU:	Jump:	Branch:	Memory:	Other:
803	603	0	100	100	0
	76%	0%	12%	12%	0%

Registers window:

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	1
\$t1	9	268500992
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	268500992
\$s1	17	0
\$s2	18	10000
\$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		4194348

Data Segment window:

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	1	3	5	7	9	11	13	15
0x10010020	17	19	21	23	25	27	29	31
0x10010040	33	35	37	39	41	43	45	47
0x10010060	49	51	53	55	57	59	61	63
0x10010080	65	67	69	71	73	75	77	79
0x100100a0	81	83	85	87	89	91	93	95
0x100100c0	97	99	101	103	105	107	109	111
0x100100e0	113	115	117	119	121	123	125	127
0x10010100	129	131	133	135	137	139	141	143
0x10010120	145	147	149	151	153	155	157	159

Tool bar:

Programa 16:

Editor window showing assembly code for 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

Mars Debugger interface showing assembly and registers:

File Edit Run Settings Tools Help

Run speed at max (no interaction)

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

Text Segment and Data Segment panes showing memory dump and breakpoints.

Mars Messages Run I/O

-- 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)

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
\$s8	24	0
\$s9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		419436

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c081001	lui \$t0, 0x1001	10: lui \$t0, 0x1001
	0x00400004	0x8d100000	lw \$s0, 0(\$t0)	11: lw \$s0, 0(\$t0)
	0x00400008	0x8d110004	lw \$s1, 4(\$t0)	12: lw \$s1, 4(\$t0)
	0x0040000c	0x22290000	addi \$s2, \$s1, 0	13: addi \$s2, \$s1, 0
	0x00400010	0x202509020	add \$s2, \$s2, \$s0	14: add \$s2, \$s2, \$s0
	0x00400014	0x2129ffff	addi \$s1, \$s1, -1	15: addi \$s1, \$s1, -1
	0x00400018	0x1520ffff	bne \$t1, \$zero, multiplica	16: bne \$t1, \$zero, multiplica
	0x0040001c	0xdad12008	sw \$s2, 8(\$t0)	17: sw \$s2, 8(\$t0)
	0x00400020	0x1520ffff	bne \$t1, \$zero, multiplica	18: bne \$t1, \$zero, multiplica
	0x00400024	0xdad12008	sw \$s2, 8(\$t0)	19: sw \$s2, 8(\$t0)

Labels

Label	Address
(global)	0x00400000
main	0x00400010
prog17.asm	0x00400010
multiplica	0x00400014
fim	0x0040001c
x	0x10010000
y	0x10010004
k	0x10010008

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

Mars Messages Run I/O

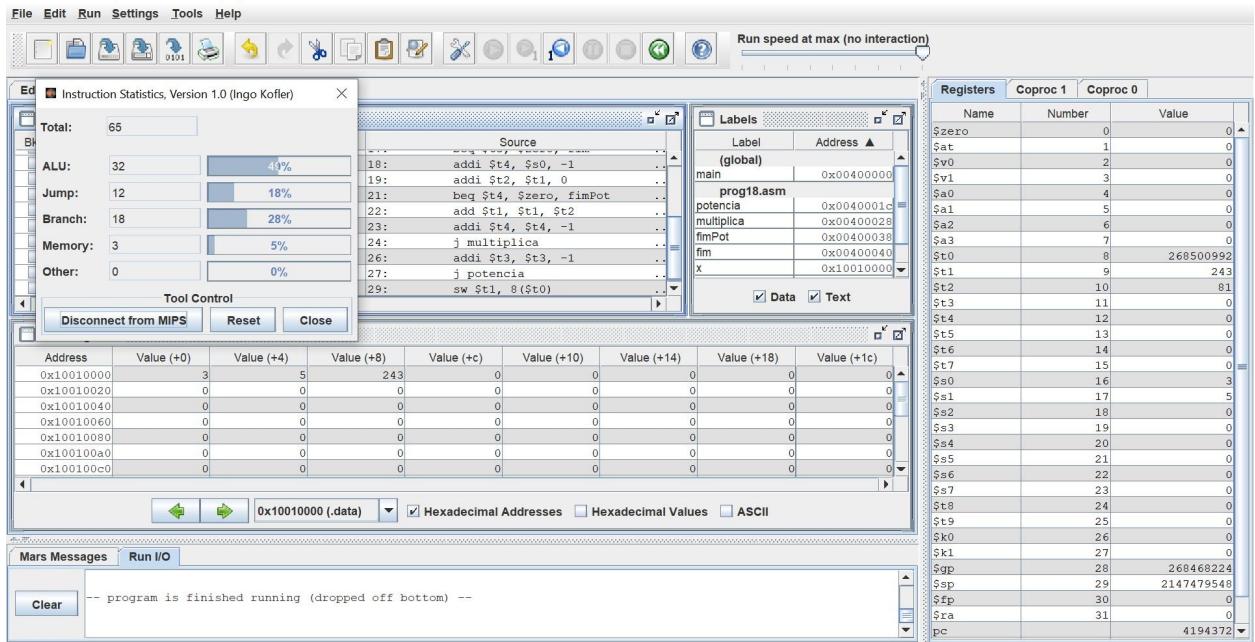
Clear -- program is finished running (dropped off bottom) --

Programa 18:

The screenshot shows a text editor window with the tab 'prog18.asm' selected. The code is written in assembly language for a processor like the MIPS. The code initializes variables \$t0, \$s0, and \$s1, sets up a loop to calculate \$t3 = y - 1, multiplies \$t4 by \$t3, and stores the result at memory location \$t0+8.

```
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 multiplicar:
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 multiplicar              # goto multiplicar
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

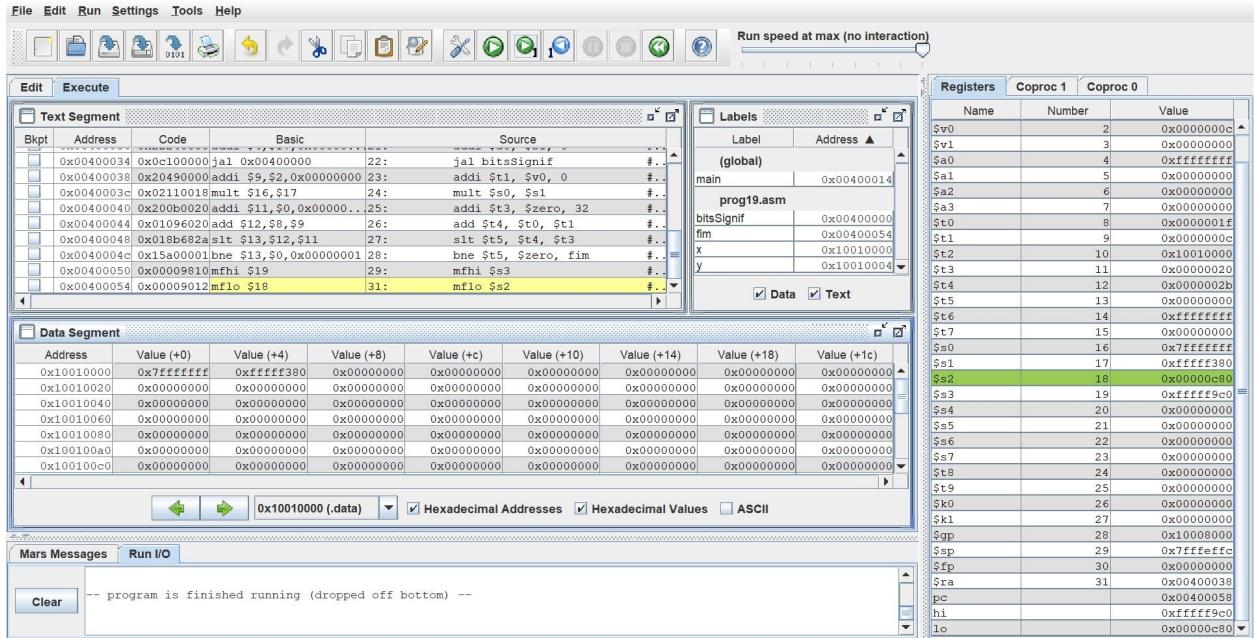
Programa 19:

The screenshot shows a debugger window with the following details:

- Menu Bar:** Edit, Execute.
- Title Bar:** prog19.asm.
- Code Area:** Displays the assembly code for Programa 19. The code includes comments explaining the purpose of each instruction. It defines a global symbol `main`, sets up registers `$t2` to `$t5` for intermediate calculations, and uses `mflo` and `mfhi` instructions to handle the multiplication result. Data section entries `x: .word 0x7FFFFFFF` and `y: .word -3200` are also present.
- Status Bar:** Line: 36 Column: 1 Show Line Numbers.

```
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, 0x10010000       # 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 -> (hi,lo)
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
```

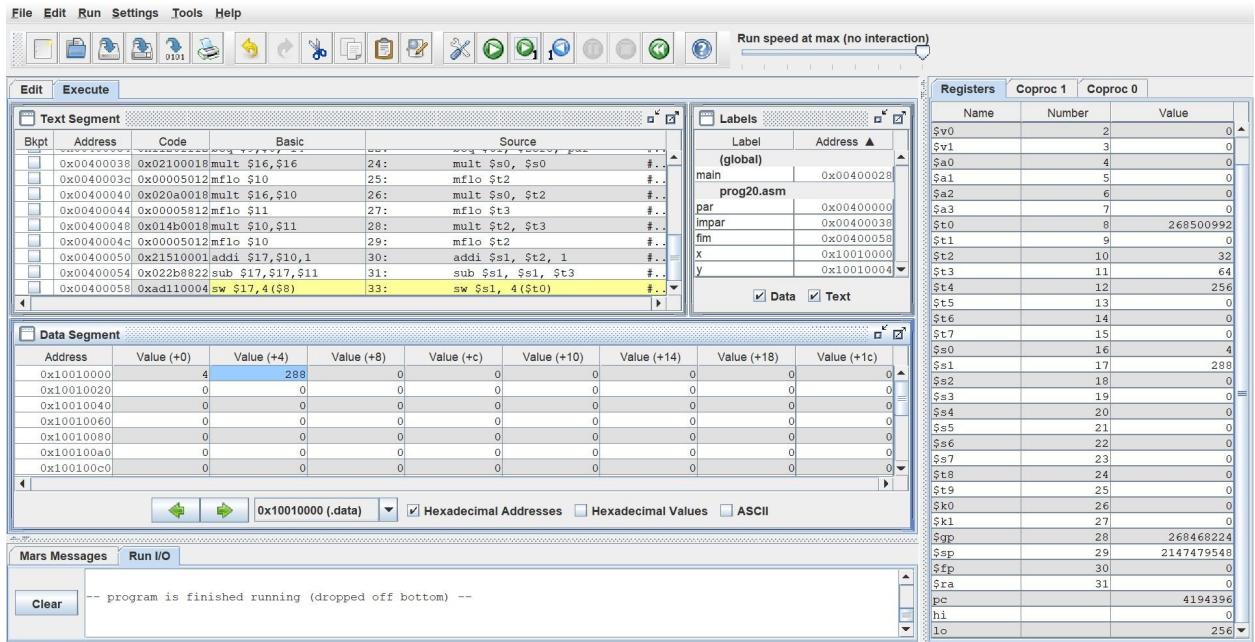


Programa 20:

The screenshot shows a text editor window with the following assembly code for Programa 20:

```
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
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
23
24     impar:
25     mult $s0, $s0           # x^2
26     mflo $t2                # t2 = x^2
27     mult $s0, $t2           # x^3
28     mflo $t3                # t3 = x^3
29     mult $t2, $t3           # x^5
30     mflo $t2                # t2 = x^5
31     addi $s1, $t2, 1         # y = x^5 + 1
32     sub $s1, $s1, $t3       # y = x^5 - x^3 + 1
33     fim:
34     sw $s1, 4($t0)          # MEM[t0+4] = y
35 .data
36 x: .word 4
37 y: .word 0
```

The code implements two parallel execution paths. The first path (labeled 'par') calculates $y = x^4 + x^3 - 2x^2$. The second path (labeled 'impar') calculates $y = x^5 + 1$. Both paths start by loading x into $s0$, then calculate powers of x using multiplication and floating-point operations. The results are then combined or stored. The program concludes with a word value at memory location $t0+4$.



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, 0x10010000      # 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	0x010492a	slt \$t1, \$zero, \$s0	17: slt \$t1, \$zero, \$s0 #..
	0x00400024	0x1520ffff	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)	0x00400018
main	0x00400018
prog21.asm	0x00400000
positivo	0x00400000
naoPositivo	0x00400028
fim	0x0040003c
X	0x10010000
y	0x10010004

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

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	269468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194368