ARQUITETURA DE COMPUTADORES II

EXERCÍCIO PRÁTICO 6

Aluna: Thais Andreatta da Silva Carmo

P9.

```
1  .text
2  .globl teste
3  teste:
4  ori $t1, $zero, 0x1001 # t1 = 0x00001001
5  sll $t1, $t1, 16 # t1 = 0x10010000
6  lw $s0, 0 ($t1) # s0 = 0x10010000 = 15
7  lw $s1, 4 ($t1) # s1 = 0x10010004 = 25
8  lw $s2, 8 ($t1) # s2 = 0x10010008 = 13
9  lw $s3, 12 ($t1) # s3 = 0x1001000C = 17
10 add $s4, $s0, $s1 # s4 = s0 + s1 = 15 + 25
11 add $s5, $s2, $s3 # s5 = s2 + s3 = 13 + 17
12 add $s6, $s4, $s5 # s6 = s4 + s5
13 sw $s6, 16 ($t1) # t1 = 0x10010000 = x1 + x2 + x3 +x4
14
15 .data
16 x1: .word 15
17 x2: .word 25
18 x3: .word 13
19 x4: .word 17
20 soma: .word -1
```

Edit | Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x34091001	ori $9,$0,4097	4: ori $t1, $zero, 0x1001 # t1 = 0x00001001
	0x00400004	0x00094c00	sll $9,$9,16	5: sll $t1, $t1, 16 # t1 = 0x10010000
	0x00400008	0x8d300000	lw $16,0($9)	6: lw $s0, 0 ($t1) # s0 = 0x10010000 = 15
	0x0040000c	0x8d310004	lw $17,4($9)	7: lw $s1, 4 ($t1) # s1 = 0x10010004 = 25
	0x00400010	0x8d320008	lw $18,8($9)	8: lw $s2, 8 ($t1) # s2 = 0x10010008 = 13
	0x00400014	0x8d33000c	lw $19,12($9)	9: lw $s3, 12 ($t1) # s3 = 0x1001000C = 17
	0x00400018	0x0211a020	add $20,$16,$17	10: add $s4, $s0, $s1 # s4 = s0 + s1 = 15 + 25
	0x0040001c	0x0253a820	add $21,$18,$19	11: add $s5, $s2, $s3 # s5 = s2 + s3 = 13 + 17
	0x00400020	0x0295b020	add $22,$20,$21	12: add $s6, $s4, $s5 # s6 = s4 + s5
	0x00400024	0xad360010	sw $22,16($9)	13: sw $s6, 16 ($t1) # t1 = 0x10010000 = x1 + x2 + x3 +x4

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)
0x10010000	15	25	13	17	70	0
0x10010020	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0
0x10010100	0	0	0	0	0	0
0x10010120	0	0	0	0	0	0
0x10010140	0	0	0	0	0	0
0x10010160	0	0	0	0	0	0

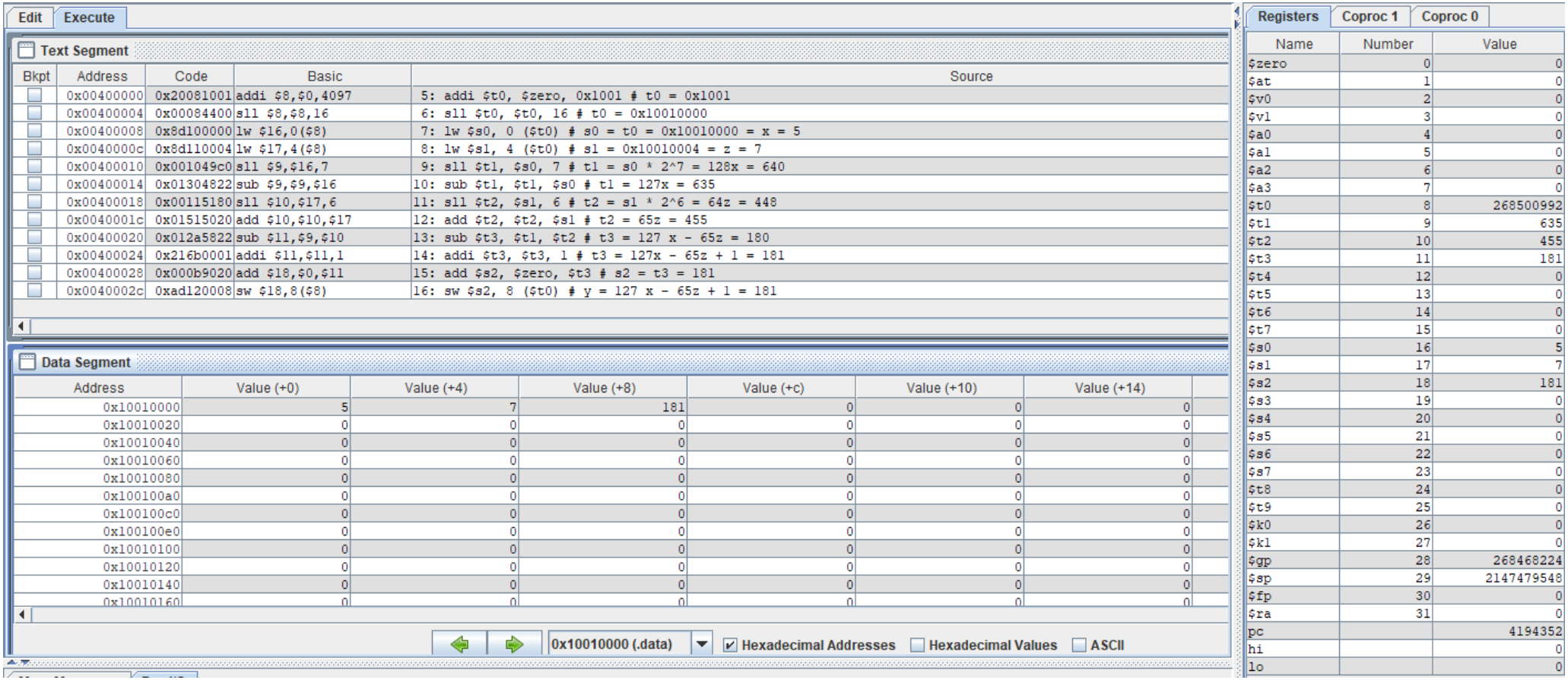
0x10010000 (.data) ☑ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

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	0
$t1	9	268500992
$t2	10	0
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	15
$s1	17	25
$s2	18	13
$s3	19	17
$s4	20	40
$s5	21	30
$s6	22	70
$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
hi		0
lo		0

P10.

```
1  .text
2
3  .globl main
4  main:
5  addi $t0, $zero, 0x1001 # t0 = 0x1001
6  sll $t0, $t0, 16 # t0 = 0x10010000
7  lw $s0, 0 ($t0) # s0 = t0 = 0x10010000 = x = 5
8  lw $s1, 4 ($t0) # s1 = 0x10010004 = z = 7
9  sll $t1, $s0, 7 # t1 = s0 * 2^7 = 128x = 640
10 sub $t1, $t1, $s0 # t1 = 127x = 635
11 sll $t2, $s1, 6 # t2 = s1 * 2^6 = 64z = 448
12 add $t2, $t2, $s1 # t2 = 65z = 455
13 sub $t3, $t1, $t2 # t3 = 127 x - 65z = 180
14 addi $t3, $t3, 1 # t3 = 127x - 65z + 1 = 181
15 add $s2, $zero, $t3 # s2 = t3 = 181
16 sw $s2, 8 ($t0) # y = 127 x - 65z + 1 = 181
17
18 .data
19 x: .word 5
20 z: .word 7
21 y: .word 0
```



Edit | Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x20081001	addi $8,$0,4097	5: addi $t0, $zero, 0x1001 # t0 = 0x1001
	0x00400004	0x00084400	sll $8,$8,16	6: sll $t0, $t0, 16 # t0 = 0x10010000
	0x00400008	0x8d100000	lw $16,0($8)	7: lw $s0, 0 ($t0) # s0 = t0 = 0x10010000 = x = 5
	0x0040000c	0x8d110004	lw $17,4($8)	8: lw $s1, 4 ($t0) # s1 = 0x10010004 = z = 7
	0x00400010	0x001049c0	sll $9,$16,7	9: sll $t1, $s0, 7 # t1 = s0 * 2^7 = 128x = 640
	0x00400014	0x01304822	sub $9,$9,$16	10: sub $t1, $t1, $s0 # t1 = 127x = 635
	0x00400018	0x00115180	sll $10,$17,6	11: sll $t2, $s1, 6 # t2 = s1 * 2^6 = 64z = 448
	0x0040001c	0x01515020	add $10,$10,$17	12: add $t2, $t2, $s1 # t2 = 65z = 455
	0x00400020	0x012a5822	sub $11,$9,$10	13: sub $t3, $t1, $t2 # t3 = 127 x - 65z = 180
	0x00400024	0x216b0001	addi $11,$11,1	14: addi $t3, $t3, 1 # t3 = 127x - 65z + 1 = 181
	0x00400028	0x000b9020	add $18,$0,$11	15: add $s2, $zero, $t3 # s2 = t3 = 181
	0x0040002c	0xad120008	sw $18,8($8)	16: sw $s2, 8 ($t0) # y = 127 x - 65z + 1 = 181

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)
0x10010000	5	7	181	0	0	0
0x10010020	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0
0x10010100	0	0	0	0	0	0
0x10010120	0	0	0	0	0	0
0x10010140	0	0	0	0	0	0
0x10010160	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	635
$t2	10	455
$t3	11	181
$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		4194352
hi		0
lo		0

P11.

```
1  .text
2
3  .globl main
4  main:
5  addi $t0, $zero, 0x1001 # t0 = 0x1001
6  sll $t0, $t0, 16 # t0 = 0x10010000
7  lw $s0, 0 ($t0) # s0* = &t0 = &0x10010000 = x = 100000
8  lw $s1, 4 ($t0) # s1 = &0x10010004 = z = 200000
9  add $s2, $s0, $s1 # s2 = 300000
10 sub $t1, $s0, $s1 # t1 = x - z
11 add $t1, $t1, $s2 # t1 = x - z + 300000
12 add $s3, $zero, $t1 # s3 = t1
13 sw $s3, 8($t0) # y = x - z + 300000
14
15 .data
16 x: .word 100000
17 z: .word 200000
18 y: .word 0
```

Edit | Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x20081001	addi $8,$0,4097	5: addi $t0, $zero, 0x1001 # t0 = 0x1001
	0x00400004	0x00084400	sll $8,$8,16	6: sll $t0, $t0, 16 # t0 = 0x10010000
	0x00400008	0x8d100000	lw $16,0($8)	7: lw $s0, 0 ($t0) # s0* = &t0 = &0x10010000 = x = 100000
	0x0040000c	0x8d110004	lw $17,4($8)	8: lw $s1, 4 ($t0) # s1 = &0x10010004 = z = 200000
	0x00400010	0x02119020	add $18,$16,$17	9: add $s2, $s0, $s1 # s2 = 300000
	0x00400014	0x02114822	sub $9,$16,$17	10: sub $t1, $s0, $s1 # t1 = x - z
	0x00400018	0x01324820	add $9,$9,$18	11: add $t1, $t1, $s2 # t1 = x - z + 300000
	0x0040001c	0x00099820	add $19,$0,$9	12: add $s3, $zero, $t1 # s3 = t1
	0x00400020	0xad130008	sw $19,8($8)	13: sw $s3, 8($t0) # y = x - z + 300000

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)
0x10010000	100000	200000	200000	0	0	0
0x10010020	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0
0x10010100	0	0	0	0	0	0
0x10010120	0	0	0	0	0	0
0x10010140	0	0	0	0	0	0
0x10010160	0	0	0	0	0	0

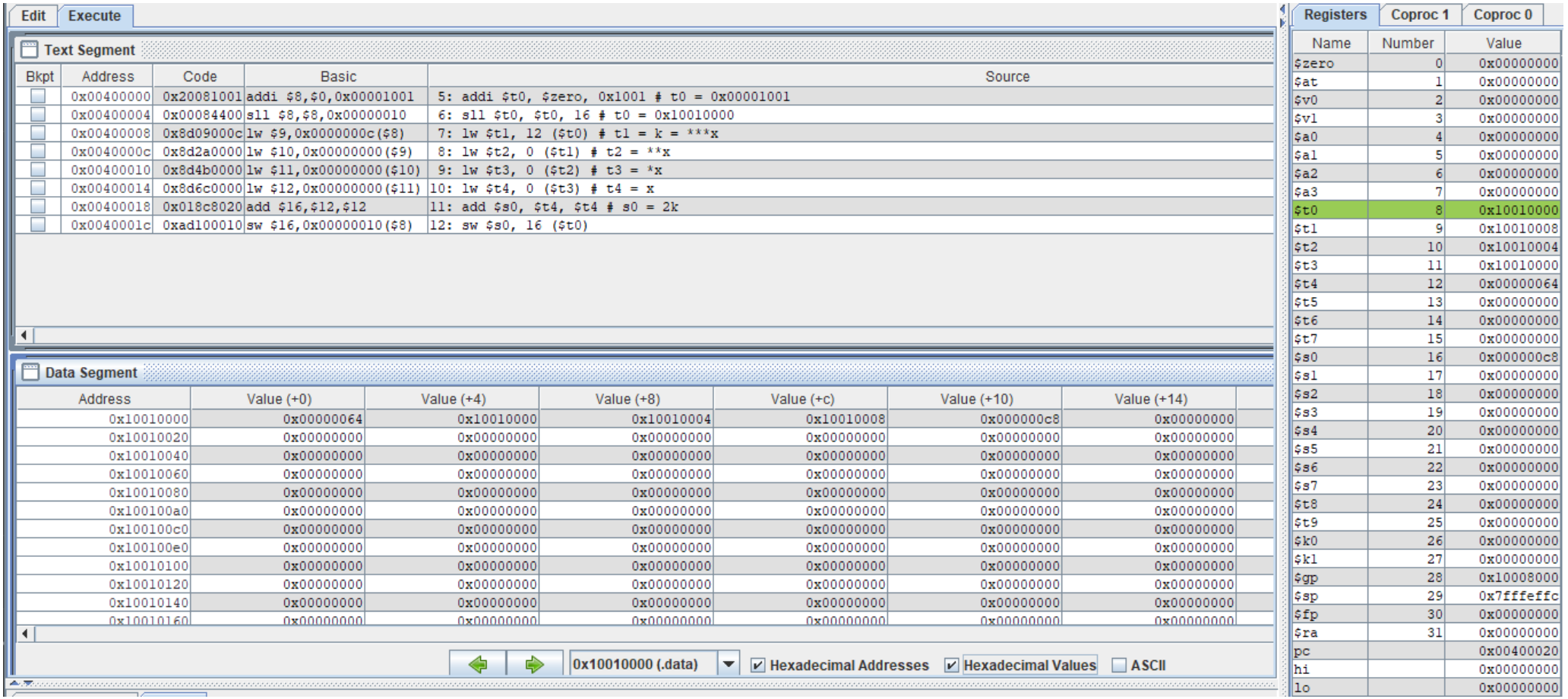
0x10010000 (.data) ☑ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

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	200000
$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	200000
$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		4194340
hi		0
lo		0

P12.

```
1  .text
2
3  .globl main
4  main:
5  addi $t0, $zero, 0x1001 # t0 = 0x00001001
6  sll $t0, $t0, 16 # t0 = 0x10010000
7  lw $t1, 12 ($t0) # t1 = k = ***x
8  lw $t2, 0 ($t1) # t2 = **x
9  lw $t3, 0 ($t2) # t3 = *x
10 lw $t4, 0 ($t3) # t4 = x
11 add $s0, $t4, $t4 # s0 = 2k
12 sw $s0, 16 ($t0) # pppx = 2k
13
14 .data
15 x: .word 100
16 px: .word 0x10010000
17 ppx: .word 0x10010004
18 pppx: .word 0x10010008
```



Edit | Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x20081001	addi $8,$0,0x00001001	5: addi $t0, $zero, 0x1001 # t0 = 0x00001001
	0x00400004	0x00084400	sll $8,$8,0x00000010	6: sll $t0, $t0, 16 # t0 = 0x10010000
	0x00400008	0x8d09000c	lw $9,0x0000000c($8)	7: lw $t1, 12 ($t0) # t1 = k = ***x
	0x0040000c	0x8d2a0000	lw $10,0x00000000($9)	8: lw $t2, 0 ($t1) # t2 = **x
	0x00400010	0x8d4b0000	lw $11,0x00000000($10)	9: lw $t3, 0 ($t2) # t3 = *x
	0x00400014	0x8d6c0000	lw $12,0x00000000($11)	10: lw $t4, 0 ($t3) # t4 = x
	0x00400018	0x018c8020	add $16,$12,$12	11: add $s0, $t4, $t4 # s0 = 2k
	0x0040001c	0xad100010	sw $16,0x00000010($8)	12: sw $s0, 16 ($t0)

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)
0x10010000	0x00000064	0x10010000	0x10010004	0x10010008	0x000000c8	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

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

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	0x10010008
$t2	10	0x10010004
$t3	11	0x10010000
$t4	12	0x00000064
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x000000c8
$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		0x00400020
hi		0x00000000
lo		0x00000000

P13.

```
1  .text
2  .globl main
3  main:
4  ori $t0, $zero, 0x1001 # t0 = 0x00001001
5  sll $t0, $t0, 16 # t0 = 0x10010000
6  lw $s0, 0($t0) # s0 = 0x10010000
7  slt $s1, $s0, $zero # s1 = s0 < 0 ?
8  beq $s1, $zero, positivo # se s1 = 0 : positivo
9  sub $s0, $zero, $s0 # s0 = |s0|
10 sw $s0, 0($t0) # s0 = 0x10010000
11 j fim
12
13 positivo:
14 sw $s0, 0($t0) # s0 = 0x10010000
15 j fim
16
17 fim:
18
19 .data
20 A: .word -10
```

Edit | Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x34081001	ori $8,$0,4097	4: ori $t0, $zero, 0x1001 # t0 = 0x00001001
	0x00400004	0x00084400	sll $8,$8,16	5: sll $t0, $t0, 16 # t0 = 0x10010000
	0x00400008	0x8d100000	lw $16,0($8)	6: lw $s0, 0($t0) # s0 = 0x10010000
	0x0040000c	0x0200882a	slt $17,$16,$0	7: slt $s1, $s0, $zero # s1 = s0 < 0 ?
	0x00400010	0x12200003	beq $17,$0,3	8: beq $s1, $zero, positivo # se s1 = 0 : positivo
	0x00400014	0x00108022	sub $16,$0,$16	9: sub $s0, $zero, $s0
	0x00400018	0xad100000	sw $16,0($8)	10: sw $s0, 0($t0) # s0 = 0x10010000
	0x0040001c	0x0810000a	j 0x00400028	11: j fim
	0x00400020	0xad100000	sw $16,0($8)	14: sw $s0, 0($t0) # s0 = 0x10010000
	0x00400024	0x0810000a	j 0x00400028	15: j fim

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)
0x10010000	10	0	0	0	0	0
0x10010020	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0
0x10010100	0	0	0	0	0	0
0x10010120	0	0	0	0	0	0
0x10010140	0	0	0	0	0	0
0x10010160	0	0	0	0	0	0

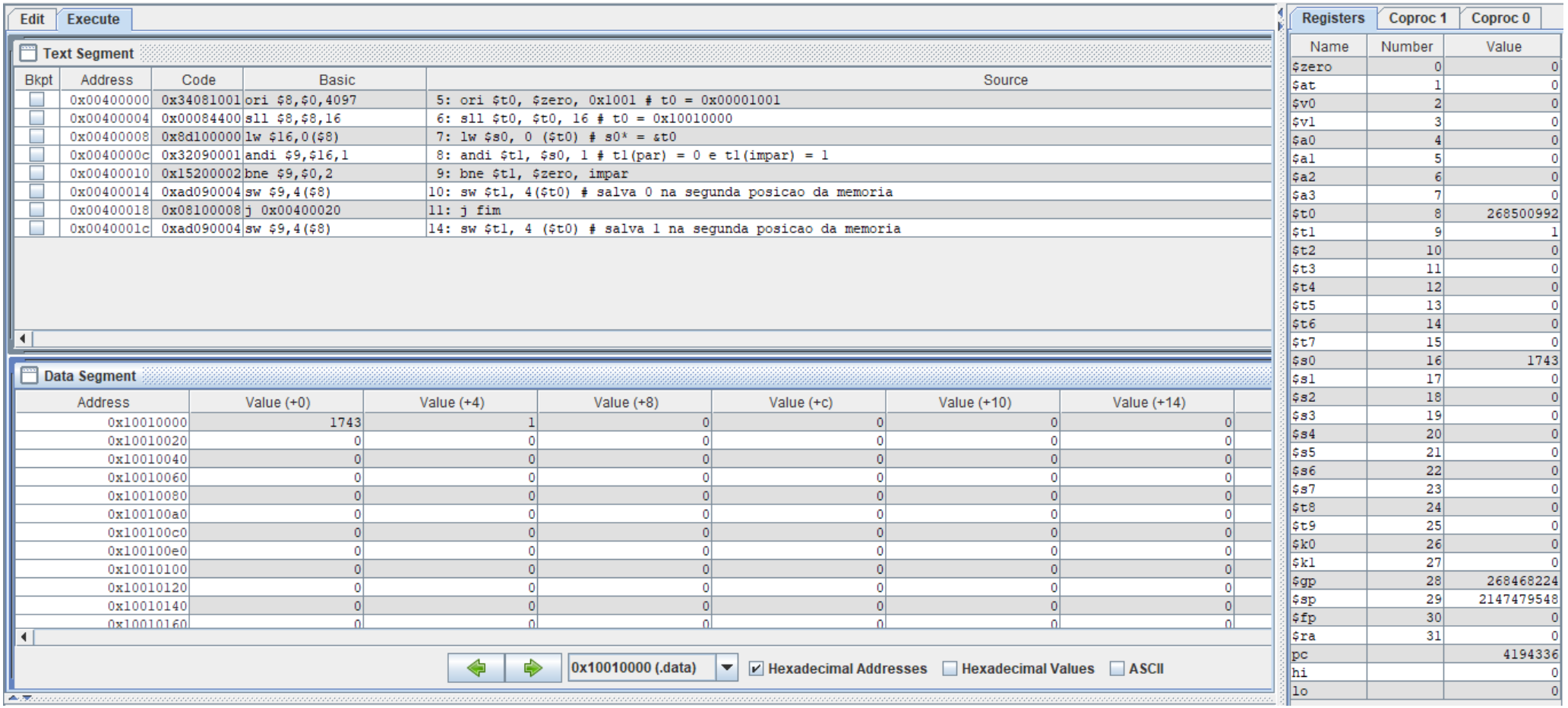
0x10010000 (.data) ☑ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

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	1
$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		4194344
hi		0
lo		0

P14.

```
1  .text
2
3  .globl main
4  main:
5  ori $t0, $zero, 0x1001 # t0 = 0x00001001
6  sll $t0, $t0, 16 # t0 = 0x10010000
7  lw $s0, 0 ($t0) # s0* = &t0
8  andi $t1, $s0, 1 # t1(par) = 0 e t1(impar) = 1
9  bne $t1, $zero, impar # se t1 != 0 : impar
10 sw $t1, 4($t0) # salva 0 na segunda posicao da memoria
11 j fim
12
13 impar:
14 sw $t1, 4 ($t0) # salva 1 na segunda posicao da memoria
15 fim:
16
17 .data
18 A: .word 1743
```



Edit | Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x34081001	ori $8,$0,4097	5: ori $t0, $zero, 0x1001 # t0 = 0x00001001
	0x00400004	0x00084400	sll $8,$8,16	6: sll $t0, $t0, 16 # t0 = 0x10010000
	0x00400008	0x8d100000	lw $16,0($8)	7: lw $s0, 0 ($t0) # s0* = &t0
	0x0040000c	0x32090001	andi $9,$16,1	8: andi $t1, $s0, 1 # t1(par) = 0 e t1(impar) = 1
	0x00400010	0x15200002	bne $9,$0,2	9: bne $t1, $zero, impar
	0x00400014	0xad090004	sw $9,4($8)	10: sw $t1, 4($t0) # salva 0 na segunda posicao da memoria
	0x00400018	0x08100008	j 0x00400020	11: j fim
	0x0040001c	0xad090004	sw $9,4($8)	14: sw $t1, 4 ($t0) # salva 1 na segunda posicao da memoria

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)
0x10010000	1743	1	0	0	0	0
0x10010020	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0
0x10010100	0	0	0	0	0	0
0x10010120	0	0	0	0	0	0
0x10010140	0	0	0	0	0	0
0x10010160	0	0	0	0	0	0

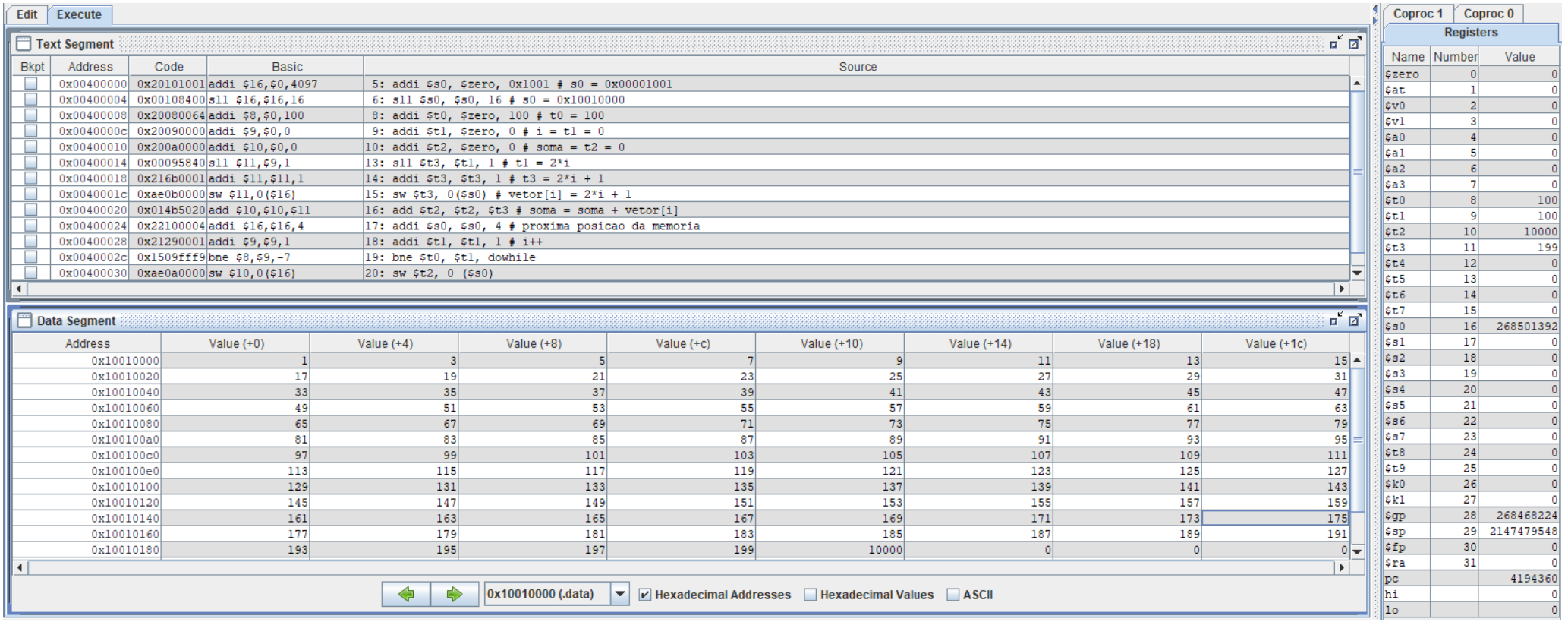
0x10010000 (.data) ☑ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

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	1
$t2	10	0
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	1743
$s1	17	0
$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		4194336
hi		0
lo		0

P15.

```
1  .text
2
3  .globl main
4  main:
5  addi $s0, $zero, 0x1001 # s0 = 0x00001001
6  sll $s0, $s0, 16 # s0 = 0x10010000
7
8  addi $t0, $zero, 100 # t0 = 100
9  addi $t1, $zero, 0 # i = t1 = 0
10 addi $t2, $zero, 0 # soma = t2 = 0
11
12 dowhile:
13 sll $t3, $t1, 1 # t1 = 2*i
14 addi $t3, $t3, 1 # t3 = 2*i + 1
15 sw $t3, 0($s0) # vetor[i] = 2*i + 1
16 add $t2, $t2, $t3 # soma = soma + vetor[i]
17 addi $s0, $s0, 4 # proxima posicao da memoria
18 addi $t1, $t1, 1 # i++
19 bne $t0, $t1, dowhile # se t1 != t0 : dowhile
20 sw $t2, 0 ($s0) # s0 = soma
21 j fim
22 fim:
23
24 .data
```



Edit | Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x20101001	addi $16,$0,4097	5: addi $s0, $zero, 0x1001 # s0 = 0x00001001
	0x00400004	0x00108400	sll $16,$16,16	6: sll $s0, $s0, 16 # s0 = 0x10010000
	0x00400008	0x20080064	addi $8,$0,100	8: addi $t0, $zero, 100 # t0 = 100
	0x0040000c	0x20090000	addi $9,$0,0	9: addi $t1, $zero, 0 # i = t1 = 0
	0x00400010	0x200a0000	addi $10,$0,0	10: addi $t2, $zero, 0 # soma = t2 = 0
	0x00400014	0x00095840	sll $11,$9,1	13: sll $t3, $t1, 1 # t1 = 2*i
	0x00400018	0x216b0001	addi $11,$11,1	14: addi $t3, $t3, 1 # t3 = 2*i + 1
	0x0040001c	0xae0b0000	sw $11,0($16)	15: sw $t3, 0($s0) # vetor[i] = 2*i + 1
	0x00400020	0x014b5020	add $10,$10,$11	16: add $t2, $t2, $t3 # soma = soma + vetor[i]
	0x00400024	0x22100004	addi $16,$16,4	17: addi $s0, $s0, 4 # proxima posicao da memoria
	0x00400028	0x21290001	addi $9,$9,1	18: addi $t1, $t1, 1 # i++
	0x0040002c	0x1509fff9	bne $8,$9,-7	19: bne $t0, $t1, dowhile
	0x00400030	0xae0a0000	sw $10,0($16)	20: sw $t2, 0 ($s0)

Data Segment

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
0x10010140	161	163	165	167	169	171	173	175
0x10010160	177	179	181	183	185	187	189	191
0x10010180	193	195	197	199	10000	0	0	0

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

Coproc 1 | Coproc 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	100
$t1	9	100
$t2	10	10000
$t3	11	199
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	268501392
$s1	17	0
$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		4194360
hi		0
lo		0

P16. Se optarmos por realizar (x * y) antes da divisão por Z, o lo tem um overflow e o programa não funciona corretamente.

```
.text

.globl main
main:
ori $t0, $zero, 0x1001 # t0 = 0x00001001
sll $t0, $t0, 16 # t0 = 0x10010000
lw $s0, 0($t0) # s0 = x
lw $s1, 4 ($t0) # s1 = y
lw $s2, 8 ($t0) #s2 = z
mult $s0, $s1 # x * y
mfhi $s4 # s4 = hi
mflo $s3 # s3 = lo
or $s5, $zero, $s4 # s5 = 0000hi
sll $s5, $s5, 16 # s5 = hi | 0000
or $s5, $s5, $s3 # s5 = hi | lo
div $s5, $s2 # (x * y) / z
mflo $s5 # s5 = lo = (x * y) / z
sw $s5, 12 ($t0) # t0 + 12 = (x * y) / z

.data
x: .word 0x186A00
y: .word 0x13880
z: .word 0x61A80
```

Edit | Execute

Text Segment

Bkpt	Address	Code	Basic	
	0x00400000	0x34081001	ori $8,$0,4097	5: ori $t0, $zero, 0x1001 # t0 = 0x00001001
	0x00400004	0x00084400	sll $8,$8,16	6: sll $t0, $t0, 16 # t0 = 0x10010000
	0x00400008	0x8d100000	lw $16,0($8)	7: lw $s0, 0($t0) # s0 = x
	0x0040000c	0x8d110004	lw $17,4($8)	8: lw $s1, 4 ($t0) # s1 = y
	0x00400010	0x8d120008	lw $18,8($8)	9: lw $s2, 8 ($t0) #s2 = z
	0x00400014	0x02110018	mult $16,$17	10: mult $s0, $s1 # x * y
	0x00400018	0x0000a010	mfhi $20	11: mfhi $s4 # s4 = hi
	0x0040001c	0x00009812	mflo $19	12: mflo $s3 # s3 = lo
	0x00400020	0x0014a825	or $21,$0,$20	13: or $s5, $zero, $s4 # s5 = 0000hi
	0x00400024	0x0015ac00	sll $21,$21,16	14: sll $s5, $s5, 16 # s5 = hi | 0000
	0x00400028	0x02b3a825	or $21,$21,$19	15: or $s5, $s5, $s3 # s5 = hi | lo
	0x0040002c	0x02b2001a	div $21,$18	16: div $s5, $s2 # (x * y) / z
	0x00400030	0x0000a812	mflo $21	17: mflo $s5 # s5 = lo = (x * y) / z
	0x00400034	0xad15000c	sw $21,12($8)	18: sw $s5, 12 ($t0) # t0 + 12 = (x * y) / z

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Val
0x10010000	1600000	80000	400000	-2118	
0x10010020	0	0	0	0	
0x10010040	0	0	0	0	
0x10010060	0	0	0	0	
0x10010080	0	0	0	0	
0x100100a0	0	0	0	0	
0x100100c0	0	0	0	0	
0x100100e0	0	0	0	0	
0x10010100	0	0	0	0	
0x10010120	0	0	0	0	
0x10010140	0	0	0	0	
0x10010160	0	0	0	0	
0x10010180	0	0	0	0	

Coproc 1 | Coproc 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	0
$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	-849018880
$s4	20	29
$s5	21	-2118
$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		4194360
hi		-246016
lo		-2118

Mas se optarmos em realizar primeiro a divisão por z e depois a multiplicação por Y, o programa funciona corretamente.

```
1  .text
2
3  .globl main
4  main:
5  ori $t0, $zero, 0x1001 # t0 = 0x00001001
6  sll $t0, $t0, 16 # t0 = 0x10010000
7  lw $s0, 0($t0) # s0 = x
8  lw $s1, 4 ($t0) # s1 = y
9  lw $s2, 8 ($t0) #s2 = z
10 div $s0, $s2 # x / z
11 mflo $s3 # s3 = lo
12 mult $s3, $s1 # (x / z) * y
13 mflo $s4 # s4 = lo
14 mfhi $s5 # s5 = hi
15 or $t1, $zero, $s5 # t1 = 0000 | hi
16 sll $t1, $t1, 16 # t1 = hi | 0000
17 and $t1, $t1, $s4 # t1 = hi | lo
18 sw $t1, 12 ($t0) # t0 + 12 = (x / z) * y
19
20 .data
21 x: .word 0x186A00
22 y: .word 0x13880
23 z: .word 0x61A80
```

Edit | Execute

Text Segment

Bkpt	Address	Code	Basic	
	0x00400000	0x34081001	ori $8,$0,4097	5: ori $t0, $zero, 0x1001 # t0 = 0x00001001
	0x00400004	0x00084400	sll $8,$8,16	6: sll $t0, $t0, 16 # t0 = 0x10010000
	0x00400008	0x8d100000	lw $16,0($8)	7: lw $s0, 0($t0) # s0 = x
	0x0040000c	0x8d110004	lw $17,4($8)	8: lw $s1, 4 ($t0) # s1 = y
	0x00400010	0x8d120008	lw $18,8($8)	9: lw $s2, 8 ($t0) #s2 = z
	0x00400014	0x0212001a	div $16,$18	10: div $s0, $s2 # x / z
	0x00400018	0x00009812	mflo $19	11: mflo $s3 # s3 = lo
	0x0040001c	0x02710018	mult $19,$17	12: mult $s3, $s1 # (x / z) * y
	0x00400020	0x0000a012	mflo $20	13: mflo $s4 # s4 = lo
	0x00400024	0x0000a810	mfhi $21	14: mfhi $s5 # s5 = hi
	0x00400028	0x00154825	or $9,$0,$21	15: or $t1, $zero, $s5 # t1 = 0000 | hi
	0x0040002c	0x00094c00	sll $9,$9,16	16: sll $t1, $t1, 16 # t1 = hi | 0000
	0x00400030	0x01344824	and $9,$9,$20	17: and $t1, $t1, $s4 # t1 = hi | lo
	0x00400034	0xad09000c	sw $9,12($8)	18: sw $t1, 12 ($t0) # t0 + 12 = (x / z) * y

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Val
0x10010000	1600000	80000	400000	0	
0x10010020	0	0	0	0	
0x10010040	0	0	0	0	
0x10010060	0	0	0	0	
0x10010080	0	0	0	0	
0x100100a0	0	0	0	0	
0x100100c0	0	0	0	0	
0x100100e0	0	0	0	0	
0x10010100	0	0	0	0	
0x10010120	0	0	0	0	
0x10010140	0	0	0	0	
0x10010160	0	0	0	0	
0x10010180	0	0	0	0	

Coproc 1 | Coproc 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	0
$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	4
$s4	20	320000
$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		4194360
hi		0
lo		320000

P17.

```
1  .text
2
3  .globl main
4  main:
5  ori $t0, $zero, 0x1001 # t0 = 0x00001001
6  sll $t0, $t0, 16 # t0 = 0x10010000
7  lw $s1, 0 ($t0) # s1 = x
8  lw $s2, 4 ($t0) # s2 = y
9  addi $s3, $zero, 0 # s3 = contador
10 addi $s4, $zero, 0 # s4 = soma
11 do:
12 add $s4, $s4, $s1  # soma = soma + x
13 addi $s3, $s3, 1 # contador++
14 bne $s3, $s2, do # se s3 != s3 : do
15 j fim
16
17 fim:
18 sw $s4, 8 ($t0) # t0 + 8 = soma
19
20 .data
21 x: .word 5
22 y: .word 7
```

Edit | Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x34081001	ori $8,$0,4097	5: ori $t0, $zero, 0x1001 # t0 = 0x00001001
	0x00400004	0x00084400	sll $8,$8,16	6: sll $t0, $t0, 16 # t0 = 0x10010000
	0x00400008	0x8d110000	lw $17,0($8)	7: lw $s1, 0 ($t0) # s1 = x
	0x0040000c	0x8d120004	lw $18,4($8)	8: lw $s2, 4 ($t0) # s2 = y
	0x00400010	0x20130000	addi $19,$0,0	9: addi $s3, $zero, 0 # s3 = contador
	0x00400014	0x20140000	addi $20,$0,0	10: addi $s4, $zero, 0 # s4 = soma
	0x00400018	0x0291a020	add $20,$20,$17	12: add $s4, $s4, $s1
	0x0040001c	0x22730001	addi $19,$19,1	13: addi $s3, $s3, 1 # contador++
	0x00400020	0x1672fffd	bne $19,$18,-3	14: bne $s3, $s2, do
	0x00400024	0x0810000a	j 0x00400028	15: j fim
	0x00400028	0xad140008	sw $20,8($8)	18: sw $s4, 8 ($t0)

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	5	7	35	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
0x100100e0	0	0	0	0	0	0	0	0
0x10010100	0	0	0	0	0	0	0	0
0x10010120	0	0	0	0	0	0	0	0
0x10010140	0	0	0	0	0	0	0	0
0x10010160	0	0	0	0	0	0	0	0
0x10010180	0	0	0	0	0	0	0	0

Mars Messages | Run I/O

Coproc 1 | Coproc 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	0
$t2	10	0
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	5
$s2	18	7
$s3	19	7
$s4	20	35
$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
hi		0
lo		0

P18.

```
1  .text
2  .globl main
3  main:
4  ori $t0, $zero, 0x1001 # t0 = 0x00001001
5  sll $t0, $t0, 16 # t0 = 0x10010000
6  lw $s1, 0 ($t0) # s1 = x
7  lw $s2, 4 ($t0) # s2 = y
8  addi $s3, $zero, 0 # s3 = contador
9  addi $s4, $zero, 1 # s4 = potencia
10
11 do:
12 mult $s4, $s1 # pot = pot * x
13 mflo $s4
14 addi $s3, $s3, 1 # contador++
15 bne $s3, $s2, do # contador != y
16 j fim
17
18 fim:
19 sw $s4, 8 ($t0) # k = x ^ y
20
21 .data
22 x: .word 5
23 y: .word 3
```

Edit | Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x34081001	ori $8,$0,4097	5: ori $t0, $zero, 0x1001 # t0 = 0x00001001
	0x00400004	0x00084400	sll $8,$8,16	6: sll $t0, $t0, 16 # t0 = 0x10010000
	0x00400008	0x8d110000	lw $17,0($8)	7: lw $s1, 0 ($t0) # s1 = x
	0x0040000c	0x8d120004	lw $18,4($8)	8: lw $s2, 4 ($t0) # s2 = y
	0x00400010	0x20130000	addi $19,$0,0	9: addi $s3, $zero, 0 # s3 = contador
	0x00400014	0x20140001	addi $20,$0,1	10: addi $s4, $zero, 1 # s4 = potencia
	0x00400018	0x02910018	mult $20,$17	13: mult $s4, $s1 # pot = pot * x
	0x0040001c	0x0000a012	mflo $20	14: mflo $s4
	0x00400020	0x22730001	addi $19,$19,1	15: addi $s3, $s3, 1 # contador++
	0x00400024	0x1672fffc	bne $19,$18,-4	16: bne $s3, $s2, do # contador != y
	0x00400028	0x0810000b	j 0x0040002c	17: j fim
	0x0040002c	0xad140008	sw $20,8($8)	20: sw $s4, 8 ($t0) # k = x ^ y

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	5	3	125	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
0x100100e0	0	0	0	0	0	0	0	0
0x10010100	0	0	0	0	0	0	0	0
0x10010120	0	0	0	0	0	0	0	0
0x10010140	0	0	0	0	0	0	0	0
0x10010160	0	0	0	0	0	0	0	0
0x10010180	0	0	0	0	0	0	0	0

Mars Messages | Run I/O

Coproc 1 | Coproc 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	0
$t2	10	0
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	5
$s2	18	3
$s3	19	3
$s4	20	125
$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		4194352
hi		0
lo		125