

# Relatório 7

Rafael Amauri Diniz Augusto - 651047

# Programa 9

**Edit   Execute**

**Text Segment**

Bkpt	Address	Code	Basic	Source
	0x00400000	0x34081001	ori \$8,\$0,4097	16: ori \$t0, \$zero, 0x1001 # T0 = 0x1001
	0x00400004	0x00084400	sll \$8,\$8,16	17: sll \$t0, \$t0, 16 # T0 = 0x10010000
	0x00400008	0xd8d10000	lw \$16,0(\$8)	19: lw \$s0, 0(\$t0) # x1 = MEM[0+T0]
	0x0040000c	0xd8d110004	lw \$17,4(\$8)	21: lw \$s1, 4(\$t0) # x2 = MEM[4+T0]
	0x00400010	0xd8d120008	lw \$18,8(\$8)	23: lw \$s2, 8(\$t0) # x3 = MEM[8+T0]
	0x00400014	0xd8d13000c	lw \$19,12(\$8)	25: lw \$s3, 12(\$t0) # x4 = MEM[12+T0]
	0x00400018	0x02114820	add \$9,\$16,\$17	27: add \$t1, \$s0, \$s1 # T1 = x1 + x2
	0x0040001c	0x01324820	add \$9,\$9,\$18	28: add \$t1, \$t1, \$s2 # T1 = x1 + x2 + x3
	0x00400020	0x0133a020	add \$20,\$9,\$19	29: add \$s4, \$t1, \$s3 # soma = x1 + x2 + x3 + x4
	0x00400024	0xad140010	sw \$20,16(\$8)	31: sw \$s4, 16(\$t0) # MEM[16+T0] = soma

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)
0x10010000	15	25	13	17	70	
0x10010020	0	0	0	0	0	
0x10010040	0	0	0	0	0	
0x10010060	0	0	0	0	0	
0x10010080	0	0	0	0	0	
0x100100a0	0	0	0	0	0	
0x100100c0	0	0	0	0	0	
0x100100e0	0	0	0	0	0	
0x10010100	0	0	0	0	0	
0x10010120	0	0	0	0	0	
0x10010140	0	0	0	0	0	
0x10010160	0	0	0	0	0	
0x10010180	0	0	0	0	0	
0x100101a0	0	0	0	0	0	
0x100101c0	0	0	0	0	0	
0x100101e0	0	0	0	0	0	

**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	53
\$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	70
\$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		4194344
hi		0
lo		0

# Programa 10

Text Segment		Data Segment		Registers	
Bkpt	Address	Code	Basic	Source	
	0x00400000	0x34081001	ori \$t0, \$zero, 0x1001 # T0 = 0x1001		
	0x00400004	0x00084400	sll \$t0, \$t0, 16 # T0 = 0x10010000		
	0x00400008	0xd1000000	lw \$t0, 0(\$t0) # x = MEM[0+T0], x = 5		
	0x0040000c	0x8d120004	lw \$t1, 4(\$t0) # z = MEM[4+T0], z = 7		
	0x00400010	0x001049c0	sll \$t1, \$t0, 7 # T1 = 128x		
	0x00400014	0x01304822	sub \$t1, \$t1, \$t0 # T1 = 127x		
	0x00400018	0x00125180	sll \$t2, \$t2, 6 # T2 = 64z		
	0x0040001c	0x01525020	add \$t2, \$t2, \$t2 # T2 = 65z		
	0x00400020	0x012a4822	sub \$t1, \$t1, \$t2 # T1 = 127x - 65y		
	0x00400024	0x21310001	addi \$t1, \$t1, 1 # y = 127x - 65y + 1		
	0x00400028	0xdad10008	sw \$t1, 8(\$t0) # MEM[8+T0] = y		

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)
0x10010000	5	7	181	0	0
0x10010020	0	0	0	0	0
0x10010040	0	0	0	0	0
0x10010060	0	0	0	0	0
0x10010080	0	0	0	0	0
0x100100a0	0	0	0	0	0
0x100100c0	0	0	0	0	0
0x100100e0	0	0	0	0	0
0x10010100	0	0	0	0	0
0x10010120	0	0	0	0	0
0x10010140	0	0	0	0	0
0x10010160	0	0	0	0	0
0x10010180	0	0	0	0	0
0x100101a0	0	0	0	0	0
0x100101c0	0	0	0	0	0
0x100101e0	0	0	0	0	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	180
\$t2	10	455
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	5
\$s1	17	181
\$s2	18	7
\$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
hi		0
lo		0

# Programa 11

**Text Segment**

Bkpt	Address	Code	Basic	Source
	0x00400000	0x34081001	ori \$t0, \$zero, 0x1001 # T0 = 0x1001	
	0x00400004	0x00084400	sll \$t0, \$t0, 16 # T0 = 0x10010000	
	0x00400008	0x8d100000	lw \$t0, 0(\$t0) # x = MEM[0+T0]	
	0x0040000c	0x8d120004	lw \$t1, 4(\$t0) # z = MEM[4+T0]	
	0x00400010	0x02124822	sub \$t1, \$t0, \$t2 # T1 = x - z	
	0x00400014	0x340a493e	ori \$t2, \$zero, 0x493E # T2 = 0X493E	
	0x00400018	0x000a5100	sll \$t2, \$t2, 4 # T2 = 0X49R30	
	0x0040001c	0x012a8820	add \$t1, \$t1, \$t2 # y = x - z + 300.000	
	0x00400020	0xad110008	sw \$t1, 8(\$t0) # MEM[0+T0] = y	

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)
0x10010000	100000	200000	200000	0	0
0x10010020	0	0	0	0	0
0x10010040	0	0	0	0	0
0x10010060	0	0	0	0	0
0x10010080	0	0	0	0	0
0x100100a0	0	0	0	0	0
0x100100c0	0	0	0	0	0
0x100100e0	0	0	0	0	0
0x10010100	0	0	0	0	0
0x10010120	0	0	0	0	0
0x10010140	0	0	0	0	0
0x10010160	0	0	0	0	0
0x10010180	0	0	0	0	0
0x100101a0	0	0	0	0	0
0x100101c0	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	-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
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194340
hi		0
lo		0

# Programa 12

**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	0x10010000
\$t1	9	0x10010004
\$t2	10	0x10010008
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x10010008
\$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		0x00400030
hi		0x00000000
lo		0x00000000

**Text Segment**

Bkpt	Address	Code	Basic	Source
	0x00400000	0x34081001	ori \$8,\$0,0x00001001	16: ori \$t0, \$zero, 0x1001 # T0 = 0x1001
	0x00400004	0x00084400	sll \$8,\$8,0x00000010	17: sll \$t0, \$t0, 16 # T0 = 0x10010000
	0x00400008	0xad080000	sw \$8,0x00000000(\$8)	19: sw \$t0, 0(\$t0) # T1 = 0x10010000
	0x0040000c	0x34091001	ori \$9,\$0,0x00001001	21: ori \$t1, \$zero, 0x1001 # T1 = 0x1001
	0x00400010	0x00094c00	sll \$9,\$9,0x00000010	22: sll \$t1, \$t1, 16 # T1 = 0x10010000
	0x00400014	0x35290004	ori \$9,\$9,0x00000004	23: ori \$t1, \$t1, 0x0004 # T1 = 0x10010004
	0x00400018	0xad290000	sw \$9,0x00000000(\$9)	25: sw \$t1, 0(\$t1) # T2 = 0x10010004
	0x0040001c	0x340a1001	ori \$10,\$0,0x00001001	27: ori \$t2, \$zero, 0x1001 # T2 = 0x1001
	0x00400020	0x000a5400	sll \$10,\$10,0x00000010	28: sll \$t2, \$t2, 16 # T2 = 0x10010000
	0x00400024	0x354a0008	ori \$10,\$10,0x00000008	29: ori \$t2, \$t2, 0x0008 # T2 = 0x10010008
	0x00400028	0xad4a0000	sw \$10,0x00000000(\$10)	31: sw \$t2, 0(\$t2) # T2 = 0x10010008
	0x0040002c	0x000aa8025	or \$16,\$0,\$10	33: or \$s0, \$zero, \$t2 # K = T2

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)
0x10010000	0x10010000	0x10010004	0x10010008	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data)

Hexadecimal Addresses    Hexadecimal

# Programa 13

The screenshot shows a debugger interface with three main windows:

- Text Segment:** Displays assembly code with comments. The code includes instructions like ori, sll, lw, sr1, beg, sub, and sw. A specific instruction at address 0x00400018 is highlighted.
- Data Segment:** Displays memory dump with columns for Address, Value (+0), Value (+4), Value (+8), Value (+c), Value (+10), Value (+14), and Value (+18). The first row shows the value 64 at address 0x10010000.
- Registers:** Shows the state of various registers. The \$s0 register is highlighted in green and has a value of 64. Other registers show values 0 or 1.

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	64
\$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		4194332
hi		0
lo		0

## Programa 14

# Programa 15

**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	199
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	268501392
\$s1	17	100
\$s2	18	10000
\$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		4194356
hi		0
lo		0

**Text Segment**

Bkpt	Address	Code	Basic	Source
	0x00400000	0x34101001	ori \$16,\$0,4097	24: ori \$s0,\$zero,0x1001 # vetor[0] = 0x1001
	0x00400004	0x01084000	sll \$16,\$16,16	25: sll \$s0,\$s0,16 # vetor[0] = 0x10010000
	0x00400008	0x20110000	addi \$17,\$0,0	27: addi \$s1,\$zero,0 # i = 0
	0x0040000c	0x20080064	addi \$0,\$0,100	28: addi \$t0,\$zero,100 # t0 = 100
	0x00400010	0x20120000	addi \$18,\$0,0	29: addi \$s2,\$zero,0 # soma = 0
	0x00400014	0x00114840	sll \$11,\$17,1	33: sll \$t1,\$s1,1 # t1 = i*2
	0x00400018	0x21290001	addi \$9,\$9,1	34: addi \$t1,\$t1,1 # t1 = i*2 + 1
	0x0040001c	0xae090000	sw \$9,(\$16)	36: sw \$t1,0(\$s0) # vetor [i] = i*2 + 1
	0x00400020	0x02499020	add \$18,\$18,\$9	37: add \$s2,\$s2,\$t1 # soma = soma + vetor[i]
	0x00400024	0x22100004	addi \$16,\$16,4	39: addi \$s0,\$s0,4 # proxima posição de memoria
	0x00400028	0x22310001	addi \$17,\$17,1	40: addi \$s1,\$s1,1 # i++
	0x0040002c	0x1511ffff	bne \$t0,\$s1,-7	42: bne \$t0,\$s1,0 # while (i != 100)
	0x00400030	0xae120000	sw \$18,0(\$16)	44: sw \$s2,0(\$s0) # vetor[100] = soma

**Labels**



Data  Text

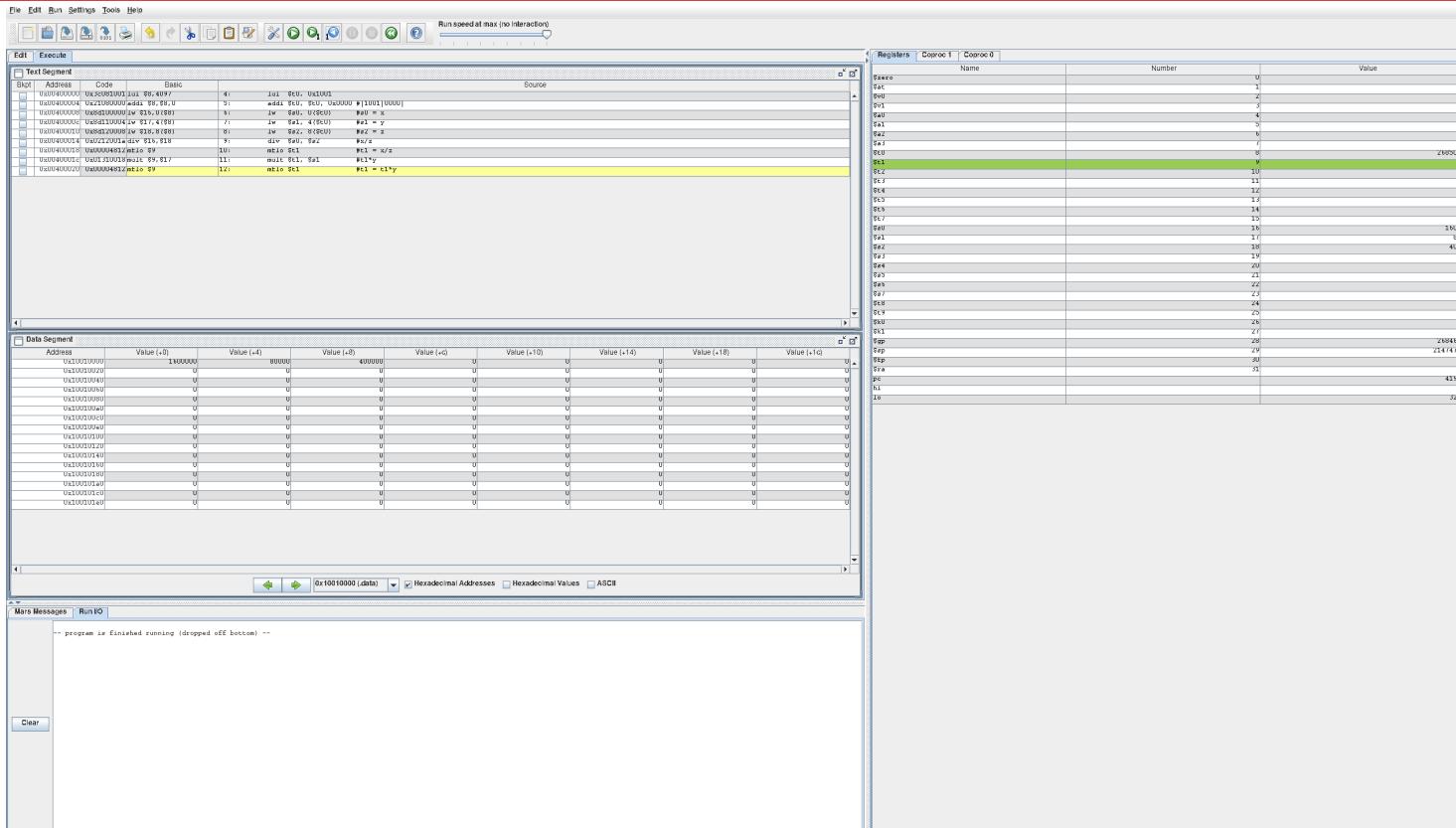
**Data Segment**

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
0x100101a0	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	100
\$t1	9	199
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	268501392
\$s1	17	100
\$s2	18	10000
\$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		4194356
hi		0
lo		0

## Programa 16



# Programa 17

The screenshot shows a debugger interface with several windows:

- Registers:** Shows the state of various registers. The \$R0 register is highlighted in green and has a value of 28800992.
- Text Segment:** Displays assembly code. One line of code is highlighted:

```
    add $R2, $R2, $R2 # R2 = 2 * R2
```
- Data Segment:** Shows memory dump starting at address U41001000. The value at U41001040 is highlighted in green and has a value of 214147943.
- Mars Messages:** Logs messages from the Mars program. It shows multiple "Reset: reset completed." messages and two "-- program is finished running (dropped off bottom) --" messages.

# Programa 18

The screenshot shows a debugger interface with several windows:

- Text Segment:** Displays assembly code in columns for Address, Code, and Source. The source code includes comments like "Finalizando na memoria a multiplicação".
- Registers:** Shows the state of various registers (R000-R019, R020-R029, R030-R039, R040-R049, R050-R059, R060-R069, R070-R079, R080-R089, R090-R099, R0A0-R0A9, R0B0-R0B9, R0C0-R0C9, R0D0-R0D9, R0E0-R0E9, R0F0-R0F9, R0G0-R0G9, R0H0-R0H9, R0I0-R0I9, R0J0-R0J9, R0K0-R0K9, R0L0-R0L9, R0M0-R0M9, R0N0-R0N9, R0O0-R0O9, R0P0-R0P9, R0Q0-R0Q9, R0R0-R0R9, R0S0-R0S9, R0T0-R0T9, R0U0-R0U9, R0V0-R0V9, R0W0-R0W9, R0X0-R0X9, R0Y0-R0Y9, R0Z0-R0Z9) and their corresponding values.
- Data Segment:** Displays memory dump from address U4100010000 to U4100010140. It shows values for bytes, words, and double words across memory locations.
- Mars Messages:** A log window showing messages related to the program's execution, including "Reset: reset completed." and "program is finished running (dropped off bottom)".