

# BÁO CÁO THỰC HÀNH TUẦN 3

## Lương Thị Tâm

### 20194663

#### Bài 1:

#### -TH1: Cộng hai số trái dấu

#### \*Code:

```
mips1.asm
1 .data
2 X: .word 0x7fffffff
3 Y: .word 0x80000000
4 .text
5 lw $s1,X
6 lw $s2,Y
7 start:
8 li $t0, 0 # trạng thái ban đầu không có tràn
9 addu $s3, $s1, $s2 # s3=s1+s2
10 xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0
11
12 bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
13 slt $t2, $s3, $s1 # t2 = s3 < s1 ? 1 : 0
14 bltz $s1, NEGATIVE #nếu s1 < 0 , nhảy đến NEGATIVE
15 beq $t2, $zero, EXIT # nếu t2 = 0 -> s3>s1 -> không tràn -> kết thúc kiểm tra
16 j OVERFLOW
17 NEGATIVE:
18 bne $t2, $zero, EXIT # nếu t2 = 1 -> s3<s1 -> không tràn -> kết thúc kiểm tra
19 OVERFLOW:
20 li $t0, 1 # có tràn số -> t0 = 1
21 EXIT:
22
```

#### \*Kết quả chạy:

Text Segment					Labels				
Offset	Address	Code	Basic	Source	Label	Address	Name	Number	Value
0	0x00400000	0x3c011001	lui \$1,4097		mips1.asm	0x00400010	\$zero	0	0
4	0x00400004	0x3c310000	lw \$17,0(\$1)		start	0x00400030	\$at	1	268500992
8	0x00400008	0x3c011001	lui \$1,4097		NEGATIVE	0x00400034	\$v0	2	0
12	0x0040000c	0x3c320004	lw \$18,4(\$1)		OVERFLOW	0x00400038	\$a0	4	0
16	0x00400010	0x24080000	addiu \$8,\$0,0	8: li \$t0, 0 # trạng thái ban đầu không có tràn	EXIT	0x10010000	\$a1	5	0
20	0x00400014	0x02329820	addu \$19,\$17,\$18	9: addu \$s3, \$s1, \$s2 # s3=s1+s2	X	0x10010004	\$a2	6	0
24	0x00400018	0x02329820	xor \$19,\$17,\$18	10: xor \$t1,\$s1,\$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác d...	Y	0x10010008	\$a3	7	0
28	0x0040001c	0x05200000	bltz \$9,6	12: bltz \$t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra		0x1001000c	\$t0	8	0
32	0x00400020	0x02715020	bltz \$10,\$17	13: slt \$t2, \$s3, \$s1 # t2 = s3 < s1 ? 1 : 0		0x10010010	\$t1	9	-1
36	0x00400024	0x02200003	bltz \$17,2	14: bltz \$s1, NEGATIVE #nếu s1 < 0 , nhảy đến NEGATIVE		0x10010014	\$t2	10	0
40	0x00400028	0x11400003	beq \$10,\$0,3	15: beq \$t2, \$zero, EXIT # nếu t2 = 0 -> s3>s1 -> không tràn -> kết thúc kiểm tra		0x10010018	\$t3	11	0
44	0x0040002c	0x06100000	j 0x00400034	16: j OVERFLOW		0x1001001c	\$t4	12	0
48	0x00400030	0x15400001	li \$10,\$0,1	18: bne \$t2, \$zero, EXIT # nếu t2 = 1 -> s3<s1 -> không tràn -> kết thúc kiểm tra		0x10010020	\$t5	13	0
52	0x00400034	0x24080001	addiu \$8,\$0,1	20: li \$t0, 1 # có tràn số -> t0 = 1		0x10010024	\$t6	14	0
<input checked="" type="checkbox"/> Data <input checked="" type="checkbox"/> Text									
Data Segment									
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)	
0x10010000	2147483647	-2147483648	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	
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	

## -TH2: Cộng hai số dương có tràn số

### \*Code:

```
1 .data
2 X: .word 0x7fffffff
3 Y: .word 1
4 .text
5     lw $s1,X
6     lw $s2,Y
7 start:
8     li $t0, 0 # trạng thái ban đầu không có tràn
9     addu $s3, $s1, $s2 # s3=s1+s2
10    xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0
11
12    bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
13    slt $t2, $s3, $s1 # t2 = s3 < s1 ? 1 : 0
14    bltz $s1, NEGATIVE #nếu s1 < 0 , nhảy đến NEGATIVE
15    beq $t2, $zero, EXIT # nếu t2 = 0 -> s3>s1 -> không tràn -> kết thúc kiểm tra
16    j OVERFLOW
17 NEGATIVE:
18     bne $t2, $zero, EXIT # nếu t2 = 1 -> s3<s1 -> không tràn -> kết thúc kiểm tra
19 OVERFLOW:
20     li $t0, 1 # có tràn số -> t0 = 1
21 EXIT:
22
```

### \*Kết quả chạy:

Text Segment					Labels		Registers		
Bkpt	Address	Code	Basic	Source	Label	Address	Name	Number	Value
	0x00400000	0x3c011001	lui \$1,4097	5: lw \$s1,X	mips1.asm	0x00400010	\$zero	0	0
	0x00400004	0x8c310000	lw \$17,0(\$1)			0x00400030	\$at	1	268500992
	0x00400008	0x3c011001	lui \$1,4097	6: lw \$s2,Y		0x00400034	\$v0	2	0
	0x0040000c	0x8c320004	lw \$18,4(\$1)			0x00400038	\$v1	3	0
	0x00400010	0x24080000	addiu \$8,\$0,0	8: li \$t0, 0 # trạng thái ban đầu không có tràn		0x0040003c	\$a0	4	0
	0x00400014	0x02329821	addu \$19,\$17,\$18	9: addu \$s3, \$s1, \$s2 # s3=s1+s2		0x00400040	\$a1	5	0
	0x00400018	0x02329820	xor \$9,\$17,\$18	10: xor \$t1,\$s1,\$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác d...		0x00400044	\$a2	6	0
	0x0040001c	0x05200006	bltz \$9,6	12: bltz \$t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra		0x00400048	\$a3	7	0
	0x00400020	0x0272502a	blt \$10,\$19,\$17	13: slt \$t2, \$s3, \$s1 # t2 = s3 < s1 ? 1 : 0		0x0040004c	\$t0	8	0
	0x00400024	0x0c200002	bltz \$17,2	14: bltz \$s1, NEGATIVE #nếu s1 < 0 , nhảy đến NEGATIVE			\$t1	9	2147483646
	0x00400028	0x11400003	beq \$10,\$0,3	15: beq \$t2, \$zero, EXIT # nếu t2 = 0 -> s3>s1 -> không tràn -> kết thúc kiểm tra			\$t2	10	1
	0x0040002c	0x06100000	j OVERFLOW	16: j OVERFLOW			\$t3	11	0
	0x00400030	0x15400001	bne \$10,\$0,1	18: bne \$t2, \$zero, EXIT # nếu t2 = 1 -> s3<s1 -> không tràn -> kết thúc kiểm tra			\$t4	12	0
	0x00400034	0x24080001	addiu \$8,\$0,1	20: li \$t0, 1 # có tràn số -> t0 = 1			\$t5	13	0
							\$t6	14	0
							\$t7	15	0
							\$t8	16	0
							\$t9	17	2147483647
							\$t10	18	1
							\$t11	19	-2147483648
							\$t12	20	0
							\$t13	21	0
							\$t14	22	0
							\$t15	23	0
							\$t16	24	0
							\$t17	25	0
							\$t18	26	0
							\$t19	27	0
							\$t20	28	268468224
							\$t21	29	2147475546
							\$t22	30	0
							\$t23	31	0
							\$t24		4194368
							\$t25		0
							\$t26		0

Data Segment	Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
	0x10010000	2147483647	1	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
	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

## -TH3: Cộng hai số dương không tràn số

### \*Code:

```
1 .data
2 X: .word 5
3 Y: .word 1
4 .text
5     lw $s1,X
6     lw $s2,Y
7 start:
8     li $t0, 0 # trạng thái ban đầu không có tràn
9     addu $s3, $s1, $s2 # s3=s1+s2
10    xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0
11
12    bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
13    slt $t2, $s3, $s1 # t2 = s3 < s1 ? 1 : 0
14    bltz $s1, NEGATIVE #nếu s1 < 0 , nhảy đến NEGATIVE
15    beq $t2, $zero, EXIT # nếu t2 = 0 -> s3>s1 -> không tràn -> kết thúc kiểm tra
16    j OVERFLOW
17 NEGATIVE:
18    bne $t2, $zero, EXIT # nếu t2 = 1 -> s3<s1 -> không tràn -> kết thúc kiểm tra
19 OVERFLOW:
20    li $t0, 1 # có tràn số -> t0 = 1
21 EXIT:
22
```

### \*Kết quả chạy:

Text Segment					Labels		Name			Number	Value
Bkpt	Address	Code	Basic	Source	Label	Address					
	0x00400000	0x3c011001	lui \$1,4097	5: lw \$s1,X	mips1.asm	0x00400010	\$zero		0	0	
	0x00400004	0x8c310000	lw \$1,0(\$1)			0x00400011	\$t1		1	268500992	
	0x00400008	0x3c011001	lui \$1,4097			0x00400012	\$t0		2	0	
	0x0040000c	0x8c320004	lw \$18,4(\$1)			0x00400013	\$t1		3	0	
	0x00400010	0x24080000	addu \$8,\$0,0	8: li \$t0, 0 # trạng thái ban đầu không có tràn		0x00400014	\$t0		4	0	
	0x00400014	0x02209821	addu \$19,\$17,\$18	9: addu \$s3, \$s1, \$s2 # s3=s1+s2		0x00400015	\$t0		5	0	
	0x00400018	0x02324826	xor \$9,\$17,\$18	10: xor \$t1,\$s1,\$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác d...		0x00400016	\$t0		6	0	
	0x0040001c	0x05200006	bltz \$9,0	12: bltz \$t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra		0x00400017	\$t0		7	0	
	0x00400020	0x02775029	slr \$10,\$19,\$17	13: slt \$t2, \$s3, \$s1 # t2 = s3 < s1 ? 1 : 0		0x00400018	\$t0		8	0	
	0x00400024	0x06200002	bltz \$17,2	14: bltz \$s1, NEGATIVE #nếu s1 < 0 , nhảy đến NEGATIVE		0x00400019	\$t0		9	4	
	0x00400028	0x11400003	beq \$10,\$0,3	15: beq \$t2, \$zero, EXIT # nếu t2 = 0 -> s3>s1 -> không tràn -> kết thúc kiểm tra		0x0040001a	\$t0		10	0	
	0x0040002c	0x08100000	j OVERFLOW	16: j OVERFLOW		0x0040001b	\$t0		11	0	
	0x00400030	0x15400001	bne \$10,\$0,1	18: bne \$t2, \$zero, EXIT # nếu t2 = 1 -> s3<s1 -> không tràn -> kết thúc kiểm tra		0x0040001c	\$t0		12	0	
	0x00400034	0x24080001	addu \$8,\$0,1	20: li \$t0, 1 # có tràn số -> t0 = 1		0x0040001d	\$t0		13	0	
						0x0040001e	\$t0		14	0	
						0x0040001f	\$t0		15	0	
						0x00400020	\$t0		16	0	
						0x00400021	\$t0		17	5	
						0x00400022	\$t0		18	1	
						0x00400023	\$t0		19	6	
						0x00400024	\$t0		20	0	
						0x00400025	\$t0		21	0	
						0x00400026	\$t0		22	0	
						0x00400027	\$t0		23	0	
						0x00400028	\$t0		24	0	
						0x00400029	\$t0		25	0	
						0x0040002a	\$t0		26	0	
						0x0040002b	\$t0		27	0	
						0x0040002c	\$t0		28	268468224	
						0x0040002d	\$t0		29	2147475548	
						0x0040002e	\$t0		30	0	
						0x0040002f	\$t0		31	0	
						0x00400030	\$t0		32	4194360	
						0x00400031	\$t0		33	0	
						0x00400032	\$t0		34	0	
						0x00400033	\$t0		35	0	
						0x00400034	\$t0		36	0	
						0x00400035	\$t0		37	0	
						0x00400036	\$t0		38	0	
						0x00400037	\$t0		39	0	
						0x00400038	\$t0		40	0	
						0x00400039	\$t0		41	0	
						0x0040003a	\$t0		42	0	
						0x0040003b	\$t0		43	0	
						0x0040003c	\$t0		44	0	
						0x0040003d	\$t0		45	0	
						0x0040003e	\$t0		46	0	
						0x0040003f	\$t0		47	0	
						0x00400040	\$t0		48	0	
						0x00400041	\$t0		49	0	
						0x00400042	\$t0		50	0	
						0x00400043	\$t0		51	0	
						0x00400044	\$t0		52	0	
						0x00400045	\$t0		53	0	
						0x00400046	\$t0		54	0	
						0x00400047	\$t0		55	0	
						0x00400048	\$t0		56	0	
						0x00400049	\$t0		57	0	
						0x0040004a	\$t0		58	0	
						0x0040004b	\$t0		59	0	
						0x0040004c	\$t0		60	0	
						0x0040004d	\$t0		61	0	
						0x0040004e	\$t0		62	0	
						0x0040004f	\$t0		63	0	
						0x00400050	\$t0		64	0	
						0x00400051	\$t0		65	0	
						0x00400052	\$t0		66	0	
						0x00400053	\$t0		67	0	
						0x00400054	\$t0		68	0	
						0x00400055	\$t0		69	0	
						0x00400056	\$t0		70	0	
						0x00400057	\$t0		71	0	
						0x00400058	\$t0		72	0	
						0x00400059	\$t0		73	0	
						0x0040005a	\$t0		74	0	
						0x0040005b	\$t0		75	0	
						0x0040005c	\$t0		76	0	
						0x0040005d	\$t0		77	0	
						0x0040005e	\$t0		78	0	
						0x0040005f	\$t0		79	0	
						0x00400060	\$t0		80	0	
						0x00400061	\$t0		81	0	
						0x00400062	\$t0		82	0	
						0x00400063	\$t0		83	0	
						0x00400064	\$t0		84	0	
						0x00400065	\$t0		85	0	
						0x00400066	\$t0		86	0	
						0x00400067	\$t0		87	0	
						0x00400068	\$t0		88	0	
						0x00400069	\$t0		89	0	
						0x0040006a	\$t0		90	0	
						0x0040006b	\$t0		91	0	
						0x0040006c	\$t0		92	0	
						0x0040006d	\$t0		93	0	
						0x0040006e	\$t0		94	0	
						0x0040006f	\$t0		95	0	
						0x00400070	\$t0		96	0	
						0x00400071	\$t0		97	0	
						0x00400072	\$t0		98	0	
						0x00400073	\$t0		99	0	
						0x00400074	\$t0		100	0	
						0x00400075	\$t0		101	0	
						0x00400076	\$t0		102	0	
						0x00400077	\$t0		103	0	
						0x00400078	\$t0		104	0	
						0x00400079	\$t0		105	0	
						0x0040007a	\$t0		106	0	
						0x0040007b	\$t0		107	0	
						0x0040007c	\$t0		108	0	
						0x0040007d	\$t0		109	0	
						0x0040007e	\$t0		110	0	
						0x0040007f	\$t0		111	0	
						0x00400080	\$t0		112	0	
						0x00400081	\$t0		113	0	
						0x00400082	\$t0		114	0	
						0x00400083	\$t0		115	0	
						0x00400084	\$t0		116	0	
						0x00400085	\$t0		117	0	
						0x00400086	\$t0		118	0	
						0x00400087	\$t0		119	0	
						0x00400088	\$t0		120	0	
						0x00400089	\$t0		121	0	
						0x0040008a	\$t0		122	0	
						0x0040008b	\$t0		123	0	
						0x0040008c	\$t0		124	0	
						0x0040008d	\$t0		125	0	
						0x0040008e	\$t0		126	0	
						0x0040008f	\$t0		127	0	
						0x00400090	\$t0		128	0	
						0x00400091	\$t0		129	0	
						0x00400092	\$t0		130	0	
						0x00400093	\$t0		131	0	
						0x00400094	\$t0		132	0	
						0x00400095	\$t0		133	0	
						0x00400096	\$t0		134	0	
						0x00400097	\$t0		135	0	
						0x00400098	\$t0		136	0	
						0x00400099	\$t0		137	0	
						0x0040009a	\$t0		138	0	
						0x0040009b	\$t0		139	0	
						0x0040009c	\$t0		140	0	
						0x0040009d	\$t0		141	0	
						0x0040009e	\$t0		142	0	
						0x0040009f	\$t0		143	0	
						0x004000a0	\$t0		144	0	
						0x004000a1	\$t0		145	0	
						0x004000a2	\$t0		146	0	
						0x004000a3	\$t0		147	0	
						0x004000a4	\$t0		148	0	
						0x004000a5	\$t0		149	0	
						0x004000a6	\$t0		150	0	
						0x004000a7	\$t0		151	0	
						0x004000a8	\$t0		152	0	
						0x004000a9	\$t0		153	0	
						0x004000aa	\$t0		154	0	
						0x004000ab	\$t0		155	0	
						0x004000ac	\$t0		156	0	
						0x004000ad	\$t0		157	0	
						0x004000ae	\$t0		158	0	
						0x004000af	\$t0		159	0	
						0x004000b0	\$t0		160	0	
						0x004000b1	\$t0		161	0	
						0x004000b2	\$t0		162	0	
						0x004000b3	\$t0		163	0	
						0x004000b4	\$t0		164	0	
						0x004000b5	\$t0		165	0	
						0x004000b6	\$t0		166	0	
						0x004000b7	\$t0		167	0	
						0x004					

## -TH4: Cộng hai số âm có tràn số

### \*Code:

```
1 .data
2 X: .word 0x80000000
3 Y: .word -1
4 .text
5     lw $s1,X
6     lw $s2,Y
7 start:
8     li $t0, 0 # trạng thái ban đầu không có tràn
9     addu $s3, $s1, $s2 # s3=s1+s2
10    xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0
11
12    bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
13    slt $t2, $s3, $s1 # t2 = s3 < s1 ? 1 : 0
14    bltz $s1, NEGATIVE #nếu s1 < 0 , nhảy đến NEGATIVE
15    beq $t2, $zero, EXIT # nếu t2 = 0 -> s3>s1 -> không tràn -> kết thúc kiểm tra
16    j OVERFLOW
17 NEGATIVE:
18    bne $t2, $zero, EXIT # nếu t2 = 1 -> s3<s1 -> không tràn -> kết thúc kiểm tra
19 OVERFLOW:
20    li $t0, 1 # có tràn số -> t0 = 1
21 EXIT:
22
```

### \*Kết quả chạy:

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c011001	lui \$1, 4097	5: lw \$s1,X
	0x00400004	0x8c310000	lw \$17,0(\$1)	
	0x00400008	0x3c011001	lui \$1, 4097	6: lw \$s2,Y
	0x0040000c	0x8c320004	lw \$18,4(\$1)	
	0x00400010	0x24060000	addu \$8,\$0,\$0	8: li \$t0, 0 # trạng thái ban đầu không có tràn
	0x00400014	0x02329802	addu \$19,\$17,\$18	9: addu \$s3, \$s1, \$s2 # s3=s1+s2
	0x00400018	0x02324826	xor \$9,\$17,\$18	10: xor \$t1,\$s1,\$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác d...
	0x0040001c	0x25200000	bltz \$9,6	12: bltz \$t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
	0x00400020	0x0271502a	slt \$10,\$19,\$17	13: slt \$t2,\$s3,\$s1 # t2 = s3 < s1 ? 1 : 0
	0x00400024	0x06200002	bltz \$17,2	14: bltz \$s1, NEGATIVE #nếu s1 < 0 , nhảy đến NEGATIVE
	0x00400028	0x11400003	beq \$10,\$0,3	15: beq \$t2, \$zero, EXIT # nếu t2 = 0 -> s3>s1 -> không tràn -> kết thúc kiểm tra
	0x0040002c	0x01000003	0x00400034	16: j OVERFLOW
	0x00400030	0x15400001	bne \$10,\$0,1	18: bne \$t2, \$zero, EXIT # nếu t2 = 1 -> s3<s1 -> không tràn -> kết thúc kiểm tra
	0x00400034	0x24060001	addu \$8,\$0,1	20: li \$t0, 1 # có tràn số -> t0 = 1

Labels

Label	Address
mips1.asm	
start	0x00400010
NEGATIVE	0x00400030
OVERFLOW	0x00400034
EXIT	0x00400038
Y	0x01000000
Y	0x01000004

Name	Number	Value
\$zero	0	0
\$t1	1	268500992
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	1
\$t1	9	2147483647
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$t8	16	0
\$t9	17	-2147483648
\$t10	18	-1
\$t11	19	2147483647
\$t12	20	0
\$t13	21	0
\$t14	22	0
\$t15	23	0
\$t16	24	0
\$t17	25	0
\$t18	26	0
\$t19	27	0
\$s0	28	268468224
\$s1	29	2147479548
\$s2	30	0
\$s3	31	0
\$pc		4194360
\$hi		0

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x01000000	-2147483648	-1	0	0	0	0	0	0
0x01000004	0	0	0	0	0	0	0	0
0x01000008	0	0	0	0	0	0	0	0
0x0100000c	0	0	0	0	0	0	0	0
0x01000010	0	0	0	0	0	0	0	0
0x01000014	0	0	0	0	0	0	0	0
0x01000018	0	0	0	0	0	0	0	0
0x0100001c	0	0	0	0	0	0	0	0
0x01000020	0	0	0	0	0	0	0	0
0x01000024	0	0	0	0	0	0	0	0
0x01000028	0	0	0	0	0	0	0	0
0x0100002c	0	0	0	0	0	0	0	0
0x01000030	0	0	0	0	0	0	0	0
0x01000034	0	0	0	0	0	0	0	0
0x01000038	0	0	0	0	0	0	0	0
0x0100003c	0	0	0	0	0	0	0	0
0x01000040	0	0	0	0	0	0	0	0
0x01000044	0	0	0	0	0	0	0	0
0x01000048	0	0	0	0	0	0	0	0
0x0100004c	0	0	0	0	0	0	0	0
0x01000050	0	0	0	0	0	0	0	0
0x01000054	0	0	0	0	0	0	0	0
0x01000058	0	0	0	0	0	0	0	0
0x0100005c	0	0	0	0	0	0	0	0
0x01000060	0	0	0	0	0	0	0	0
0x01000064	0	0	0	0	0	0	0	0
0x01000068	0	0	0	0	0	0	0	0
0x0100006c	0	0	0	0	0	0	0	0
0x01000070	0	0	0	0	0	0	0	0
0x01000074	0	0	0	0	0	0	0	0
0x01000078	0	0	0	0	0	0	0	0
0x0100007c	0	0	0	0	0	0	0	0
0x01000080	0	0	0	0	0	0	0	0
0x01000084	0	0	0	0	0	0	0	0
0x01000088	0	0	0	0	0	0	0	0
0x0100008c	0	0	0	0	0	0	0	0
0x01000090	0	0	0	0	0	0	0	0
0x01000094	0	0	0	0	0	0	0	0
0x01000098	0	0	0	0	0	0	0	0
0x0100009c	0	0	0	0	0	0	0	0
0x010000a0	0	0	0	0	0	0	0	0
0x010000a4	0	0	0	0	0	0	0	0
0x010000a8	0	0	0	0	0	0	0	0
0x010000ac	0	0	0	0	0	0	0	0
0x010000b0	0	0	0	0	0	0	0	0
0x010000b4	0	0	0	0	0	0	0	0
0x010000b8	0	0	0	0	0	0	0	0
0x010000bc	0	0	0	0	0	0	0	0
0x010000c0	0	0	0	0	0	0	0	0
0x010000c4	0	0	0	0	0	0	0	0
0x010000c8	0	0	0	0	0	0	0	0
0x010000cc	0	0	0	0	0	0	0	0
0x010000d0	0	0	0	0	0	0	0	0
0x010000d4	0	0	0	0	0	0	0	0
0x010000d8	0	0	0	0	0	0	0	0
0x010000dc	0	0	0	0	0	0	0	0
0x010000e0	0	0	0	0	0	0	0	0
0x010000e4	0	0	0	0	0	0	0	0
0x010000e8	0	0	0	0	0	0	0	0
0x010000ec	0	0	0	0	0	0	0	0
0x010000f0	0	0	0	0	0	0	0	0
0x010000f4	0	0	0	0	0	0	0	0
0x010000f8	0	0	0	0	0	0	0	0
0x010000fc	0	0	0	0	0	0	0	0
0x01000100	0	0	0	0	0	0	0	0
0x01000104	0	0	0	0	0	0	0	0
0x01000108	0	0	0	0	0	0	0	0
0x0100010c	0	0	0	0	0	0	0	0
0x01000110	0	0	0	0	0	0	0	0
0x01000114	0	0	0	0	0	0	0	0
0x01000118	0	0	0	0	0	0	0	0
0x0100011c	0	0	0	0	0	0	0	0
0x01000120	0	0	0	0	0	0	0	0
0x01000124	0	0	0	0	0	0	0	0
0x01000128	0	0	0	0	0	0	0	0
0x0100012c	0	0	0	0	0	0	0	0
0x01000130	0	0	0	0	0	0	0	0
0x01000134	0	0	0	0	0	0	0	0
0x01000138	0	0	0	0	0	0	0	0
0x0100013c	0	0	0	0	0	0	0	0
0x01000140	0	0	0	0	0	0	0	0
0x01000144	0	0	0	0	0	0	0	0
0x01000148	0	0	0	0	0	0	0	0
0x0100014c	0	0	0	0	0	0	0	0
0x01000150	0	0	0	0	0	0	0	0
0x01000154	0	0	0	0	0	0	0	0
0x01000158	0	0	0	0	0	0	0	0
0x0100015c	0	0	0	0	0	0	0	0
0x01000160	0	0	0	0	0	0	0	0
0x01000164	0	0	0	0	0	0	0	0
0x01000168	0	0	0	0	0	0	0	0
0x0100016c	0	0	0	0	0	0	0	0
0x01000170	0	0	0	0	0	0	0	0
0x01000174	0	0	0	0	0	0	0	0
0x01000178	0	0	0	0	0	0	0	0
0x0100017c	0	0	0	0	0	0	0	0
0x01000180	0	0	0	0	0	0	0	0
0x01000184	0	0	0	0	0	0	0	0
0x01000188	0	0	0	0	0	0	0	0
0x0100018c	0	0	0	0	0	0	0	0
0x01000190	0	0	0	0	0	0	0	0
0x01000194	0	0	0	0	0	0	0	0
0x01000198	0	0	0	0	0	0	0	0
0x0100019c	0	0	0	0	0	0	0	0
0x010001a0	0	0	0	0	0	0	0	0
0x010001a4	0	0	0	0	0	0	0	0
0x010001a8	0	0	0	0	0	0	0	0
0x010001ac	0	0	0	0	0	0	0	0
0x010001b0	0	0	0	0	0	0	0	0
0x010001b4	0	0	0	0	0	0	0	0
0x010001b8	0	0	0	0	0	0	0	0
0x010001bc	0	0	0	0	0	0	0	0
0x010001c0	0	0	0	0	0	0	0	0
0x010001c4	0	0	0	0	0	0	0	0
0x010001c8	0	0	0	0	0	0	0	0
0x010001cc	0	0	0	0	0	0	0	0
0x010001d0	0	0	0	0	0	0	0	0
0x010001d4	0	0	0	0	0	0	0	0
0x010001d8	0	0	0	0	0	0	0	0
0x010001dc	0	0	0	0	0	0	0	0
0x010001e0	0	0	0	0	0	0	0	0
0x010001e4	0	0	0	0	0	0	0	0
0x010001e8	0	0	0	0	0	0	0	0
0x010001ec	0	0	0	0	0	0	0	0
0x010001f0	0	0	0	0	0	0	0	0
0x010001f4	0	0	0	0	0	0	0	0
0x010001f8	0	0	0	0	0	0	0	0
0x010001fc	0	0	0	0	0	0	0	0
0x01000200	0	0	0	0	0	0	0	0
0x01000204	0	0	0	0	0	0	0	0
0x01000208	0	0	0	0	0	0	0	0
0x0100020c	0	0	0	0	0	0	0	0
0x01000210	0	0	0	0	0	0	0	0
0x01000214	0	0	0	0	0	0	0	0
0x01000218	0	0	0	0	0	0	0	0
0x0100021c	0	0	0	0	0	0	0	0
0x01000220	0	0	0	0	0	0	0	0
0x01000224	0	0	0	0	0	0	0	0
0x01000228	0	0	0	0	0	0	0	0
0x0100022c	0	0	0	0	0	0	0	0
0x01000230	0	0	0	0	0	0	0	0
0x01000234	0	0	0	0	0	0	0	0
0x01000238	0	0	0	0	0	0	0	0
0x0100023c	0	0	0	0	0	0	0	0
0x01000240	0	0	0	0	0	0	0	0
0x01000244	0	0	0	0	0	0	0	0
0x01000248	0							

-TH5: Cộng hai số âm không tràn số

\*Code:

```
1 .data
2 X: .word -10
3 Y: .word -1
4 .text
5     lw $s1,X
6     lw $s2,Y
7 start:
8     li $t0, 0 # trạng thái ban đầu không có tràn
9     addu $s3, $s1, $s2 # s3=s1+s2
10    xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0
11
12    bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
13    slt $t2, $s3, $s1 # t2 = s3 < s1 ? 1 : 0
14    bltz $s1, NEGATIVE #nếu s1 < 0 , nhảy đến NEGATIVE
15    beq $t2, $zero, EXIT # nếu t2 = 0 -> s3>s1 -> không tràn -> kết thúc kiểm tra
16    j OVERFLOW
17 NEGATIVE:
18    bne $t2, $zero, EXIT # nếu t2 = 1 -> s3<s1 -> không tràn -> kết thúc kiểm tra
19 OVERFLOW:
20    li $t0, 1 # có tràn số -> t0 = 1
21 EXIT:
22
```

\*Kết quả chạy:

Text Segment					Labels		Registers		
Bkpt	Address	Code	Basic	Source	Label	Address	Name	Number	Value
0	0x00400000	0x3c011001	lui \$1, 4097	5: lw \$s1,X	mips1.asm	0x00400000	\$zero	0	0
1	0x00400004	0xc3100001	lw \$1, 0(\$1)			0x00400001	\$at	1	268509992
2	0x00400008	0x3c011001	lui \$1, 4097	6: lw \$s2,Y		0x00400002	\$v0	2	0
3	0x0040000c	0xc3100001	lw \$1, 0(\$1)			0x00400003	\$t1	3	0
4	0x00400010	0x24080000	addiu \$s, \$0, 0	8: li \$t0, 0 # trạng thái ban đầu không có tràn		0x00400004	\$a0	4	0
5	0x00400014	0x0232982e	addu \$s, \$1, \$18	9: addu \$s3, \$s1, \$s2 # s3=s1+s2		0x00400005	\$a1	5	0
6	0x00400018	0x0232982e	xor \$s, \$1, \$18	10: xor \$t1,\$s1,\$s2 # kiểm tra dấu của s1 và s2. nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác d...		0x00400006	\$a2	6	0
7	0x0040001c	0xc3100001	lw \$1, 0(\$1)	12: bltz \$t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra		0x00400007	\$a3	7	0
8	0x00400020	0x0271502a	slt \$t2, \$s3, \$s1	13: slt \$t2, \$s3, \$s1 # t2 = s3 < s1 ? 1 : 0		0x00400008	\$t0	8	0
9	0x00400024	0xc3100001	lw \$1, 0(\$1)	14: bltz \$s1, NEGATIVE #nếu s1 < 0 , nhảy đến NEGATIVE		0x00400009	\$t1	9	0
10	0x00400028	0xc3100001	lw \$1, 0(\$1)	15: beq \$t2, \$zero, EXIT # nếu t2 = 0 -> s3>s1 -> không tràn -> kết thúc kiểm tra		0x0040000a	\$t2	10	1
11	0x0040002c	0xc3100001	lw \$1, 0(\$1)	16: j OVERFLOW		0x0040000b	\$t3	11	0
12	0x00400030	0x15400001	bne \$t2, \$zero, EXIT # nếu t2 = 1 -> s3<s1 -> không tràn -> kết thúc kiểm tra	18: bne \$t2, \$zero, EXIT # nếu t2 = 1 -> s3<s1 -> không tràn -> kết thúc kiểm tra		0x0040000c	\$t4	12	0
13	0x00400034	0x24080000	addiu \$s, \$0, 1	20: li \$t0, 1 # có tràn số -> t0 = 1		0x0040000d	\$t5	13	0
14	0x00400038	0xc3100001	lw \$1, 0(\$1)			0x0040000e	\$t6	14	0
15	0x0040003c	0xc3100001	lw \$1, 0(\$1)			0x0040000f	\$t7	15	0
16	0x00400040	0xc3100001	lw \$1, 0(\$1)			0x00400010	\$t8	16	0
17	0x00400044	0xc3100001	lw \$1, 0(\$1)			0x00400011	\$t9	17	-10
18	0x00400048	0xc3100001	lw \$1, 0(\$1)			0x00400012	\$t10	18	-11
19	0x0040004c	0xc3100001	lw \$1, 0(\$1)			0x00400013	\$t11	19	-11
20	0x00400050	0xc3100001	lw \$1, 0(\$1)			0x00400014	\$t12	20	0
21	0x00400054	0xc3100001	lw \$1, 0(\$1)			0x00400015	\$t13	21	0
22	0x00400058	0xc3100001	lw \$1, 0(\$1)			0x00400016	\$t14	22	0
23	0x0040005c	0xc3100001	lw \$1, 0(\$1)			0x00400017	\$t15	23	0
24	0x00400060	0xc3100001	lw \$1, 0(\$1)			0x00400018	\$t16	24	0
25	0x00400064	0xc3100001	lw \$1, 0(\$1)			0x00400019	\$t17	25	0
26	0x00400068	0xc3100001	lw \$1, 0(\$1)			0x0040001a	\$t18	26	0
27	0x0040006c	0xc3100001	lw \$1, 0(\$1)			0x0040001b	\$t19	27	0
28	0x00400070	0xc3100001	lw \$1, 0(\$1)			0x0040001c	\$t20	28	268468224
29	0x00400074	0xc3100001	lw \$1, 0(\$1)			0x0040001d	\$t21	29	2147475548
30	0x00400078	0xc3100001	lw \$1, 0(\$1)			0x0040001e	\$t22	30	0
31	0x0040007c	0xc3100001	lw \$1, 0(\$1)			0x0040001f	\$t23	31	0
32	0x00400080	0xc3100001	lw \$1, 0(\$1)			0x00400020	\$t24	32	4194360
33	0x00400084	0xc3100001	lw \$1, 0(\$1)			0x00400021	\$t25	33	0
34	0x00400088	0xc3100001	lw \$1, 0(\$1)			0x00400022	\$t26	34	0
35	0x0040008c	0xc3100001	lw \$1, 0(\$1)			0x00400023	\$t27	35	0
36	0x00400090	0xc3100001	lw \$1, 0(\$1)			0x00400024	\$t28	36	0
37	0x00400094	0xc3100001	lw \$1, 0(\$1)			0x00400025	\$t29	37	0
38	0x00400098	0xc3100001	lw \$1, 0(\$1)			0x00400026	\$t30	38	0
39	0x0040009c	0xc3100001	lw \$1, 0(\$1)			0x00400027	\$t31	39	0
40	0x004000a0	0xc3100001	lw \$1, 0(\$1)			0x00400028	\$t32	40	0
41	0x004000a4	0xc3100001	lw \$1, 0(\$1)			0x00400029	\$t33	41	0
42	0x004000a8	0xc3100001	lw \$1, 0(\$1)			0x0040002a	\$t34	42	0
43	0x004000ac	0xc3100001	lw \$1, 0(\$1)			0x0040002b	\$t35	43	0
44	0x004000b0	0xc3100001	lw \$1, 0(\$1)			0x0040002c	\$t36	44	0
45	0x004000b4	0xc3100001	lw \$1, 0(\$1)			0x0040002d	\$t37	45	0
46	0x004000b8	0xc3100001	lw \$1, 0(\$1)			0x0040002e	\$t38	46	0
47	0x004000bc	0xc3100001	lw \$1, 0(\$1)			0x0040002f	\$t39	47	0
48	0x004000c0	0xc3100001	lw \$1, 0(\$1)			0x00400030	\$t40	48	0
49	0x004000c4	0xc3100001	lw \$1, 0(\$1)			0x00400031	\$t41	49	0
50	0x004000c8	0xc3100001	lw \$1, 0(\$1)			0x00400032	\$t42	50	0
51	0x004000cc	0xc3100001	lw \$1, 0(\$1)			0x00400033	\$t43	51	0
52	0x004000d0	0xc3100001	lw \$1, 0(\$1)			0x00400034	\$t44	52	0
53	0x004000d4	0xc3100001	lw \$1, 0(\$1)			0x00400035	\$t45	53	0
54	0x004000d8	0xc3100001	lw \$1, 0(\$1)			0x00400036	\$t46	54	0
55	0x004000dc	0xc3100001	lw \$1, 0(\$1)			0x00400037	\$t47	55	0
56	0x004000e0	0xc3100001	lw \$1, 0(\$1)			0x00400038	\$t48	56	0
57	0x004000e4	0xc3100001	lw \$1, 0(\$1)			0x00400039	\$t49	57	0
58	0x004000e8	0xc3100001	lw \$1, 0(\$1)			0x0040003a	\$t50	58	0
59	0x004000ec	0xc3100001	lw \$1, 0(\$1)			0x0040003b	\$t51	59	0
60	0x004000f0	0xc3100001	lw \$1, 0(\$1)			0x0040003c	\$t52	60	0
61	0x004000f4	0xc3100001	lw \$1, 0(\$1)			0x0040003d	\$t53	61	0
62	0x004000f8	0xc3100001	lw \$1, 0(\$1)			0x0040003e	\$t54	62	0
63	0x004000fc	0xc3100001	lw \$1, 0(\$1)			0x0040003f	\$t55	63	0
64	0x00400100	0xc3100001	lw \$1, 0(\$1)			0x00400040	\$t56	64	0
65	0x00400104	0xc3100001	lw \$1, 0(\$1)			0x00400041	\$t57	65	0
66	0x00400108	0xc3100001	lw \$1, 0(\$1)			0x00400042	\$t58	66	0
67	0x0040010c	0xc3100001	lw \$1, 0(\$1)			0x00400043	\$t59	67	0
68	0x00400110	0xc3100001	lw \$1, 0(\$1)			0x00400044	\$t60	68	0
69	0x00400114	0xc3100001	lw \$1, 0(\$1)			0x00400045	\$t61	69	0
70	0x00400118	0xc3100001	lw \$1, 0(\$1)			0x00400046	\$t62	70	0
71	0x0040011c	0xc3100001	lw \$1, 0(\$1)			0x00400047	\$t63	71	0
72	0x00400120	0xc3100001	lw \$1, 0(\$1)			0x00400048	\$t64	72	0
73	0x00400124	0xc3100001	lw \$1, 0(\$1)			0x00400049	\$t65	73	0
74	0x00400128	0xc3100001	lw \$1, 0(\$1)			0x0040004a	\$t66	74	0
75	0x0040012c	0xc3100001	lw \$1, 0(\$1)			0x0040004b	\$t67	75	0
76	0x00400130	0xc3100001	lw \$1, 0(\$1)			0x0040004c	\$t68	76	0
77	0x00400134	0xc3100001	lw \$1, 0(\$1)			0x0040004d	\$t69	77	0
78	0x00400138	0xc3100001	lw \$1, 0(\$1)			0x0040004e	\$t70	78	0
79	0x0040013c	0xc3100001	lw \$1, 0(\$1)			0x0040004f	\$t71	79	0
80	0x00400140	0xc3100001	lw \$1, 0(\$1)			0x00400050	\$t72	80	0
81	0x00400144	0xc3100001	lw \$1, 0(\$1)			0x00400051	\$t73	81	0
82	0x00400148	0xc3100001	lw \$1, 0(\$1)			0x00400052	\$t74	82	0
83	0x0040014c	0xc3100001	lw \$1, 0(\$1)			0x00400053	\$t75	83	0
84	0x00400150	0xc3100001	lw \$1, 0(\$1)			0x00400054	\$t76	84	0
85	0x00400154	0xc3100001	lw \$1, 0(\$1)			0x00400055	\$t77	85	0
86	0x00400158	0xc3100001	lw \$1, 0(\$1)			0x00400056	\$t78	86	0
87	0x0040015c	0xc3100001	lw \$1, 0(\$1)			0x00400057	\$t79	87	0
88	0x00400160	0xc3100001	lw \$1, 0(\$1)			0x00400058	\$t80	88	0
89	0x00400164	0xc3100001	lw \$1, 0(\$1)			0x00400059	\$t81	89	0
90	0x00400168	0xc3100001	lw \$1, 0(\$1)			0x0040005a	\$t82	90	0
91	0x0040016c	0xc3100001	lw \$1, 0(\$1)			0x0040005b	\$t83	91	0
92	0x00400170	0xc3100001	lw \$1, 0(\$1)			0x0040005c	\$t84	92	0
93	0x00400174	0xc3100001	lw \$1, 0(\$1)			0x0040005d	\$t85	93	0
94	0x00400178	0xc3100001	lw \$1, 0(\$1)			0x0040005e	\$t86	94	0
95	0x0040017c	0xc3100001	lw \$1, 0(\$1)			0x0040005f	\$t87	95	0
96	0x00400180	0xc3100001	lw \$1, 0(\$1)			0x00400060	\$t88	96	0
97	0x00400184	0xc3100001	lw \$1, 0(\$1)			0x00400061	\$t89	97	0
98	0x00400188	0xc3100001	lw \$1, 0(\$1)			0x00400062	\$t90	98	0
99	0x0040018c	0xc3100001	lw \$1, 0(\$1)			0x00400063	\$t91	99	0
100	0x00400190	0xc3100001	lw \$1, 0(\$1)			0x00400064	\$t92	100	0
101	0x00400194	0xc3100001	lw \$1, 0(\$1)			0x00400065	\$t93	101	0
102	0x00400198	0xc3100001	lw \$1, 0(\$1)			0x00400066	\$t94	102	0
103	0x0040019c	0xc3100001	lw \$1, 0(\$1)			0x00400067	\$t95	103	0
104	0x004001a0	0xc3100001	lw \$1, 0(\$1)			0x00400068	\$t96	104	0
105	0x004001a4	0xc3100001	lw \$1, 0(\$1)			0x00400069	\$t97	105	0
106	0x004001a8	0xc3100001	lw \$1, 0(\$1)			0x0040006a	\$t98	106	0
107	0x004001ac	0xc3100001	lw \$1, 0(\$1)			0x0040006b	\$t99	107	0
108	0x004001b0	0xc3100001	lw \$1, 0(\$1)			0x0040006c	\$t100	108	0
109	0x004001b4	0xc3100001	lw \$1, 0(\$1)			0x0040006d	\$t101	109	0
110	0x004001b8	0xc3100001	lw \$1, 0(\$1)			0x0040006e	\$t102	110	0
111	0x004001bc	0xc3100001	lw \$1, 0(\$1)			0x0040006f	\$t103	111	0
112	0x004001c0	0xc3100001	lw \$1, 0(\$1)			0x00400070	\$t104	112	0
113	0x004001c4	0xc3100001	lw \$1, 0(\$1)			0x00400071	\$t105	113	0
114	0x004001c8	0xc3100001	lw \$1, 0(\$1)			0x00400072	\$t106	114	0
115	0x004001cc	0xc3100001	lw \$1, 0(\$1)			0x00400073	\$t107	115	0
116	0x004001d0	0xc3100001	lw \$1, 0(\$1)			0x00400074	\$t108	116	0

**\*Code:**

[illegible]

**\*Code:**

The screenshot displays the Immunity Debugger's MIPS2 assembly window. The assembly code is as follows:

```

0: lui $t0, 0x00000100
4: lw $t0, x($t0)
8: ori $t0, $t0, 0x000000ff
12: lw $t0, x($t0)

```

The register window on the right shows the state of the registers. The \$t0 register contains the value 0x000000ff. The data segment window at the bottom shows memory addresses from 0x00000000 to 0x0000000f.

**\*Code:**

### Text Segment

Offset	Address	Code	Basic	Source
0	0x00400000	0x3cd11001	lui \$1, 0x00000101	
4	0x00400004	0x8c300000	lw \$16, 0x00000000(\$1)	lw \$16, x # \$16 * x
8	0x00400008	0x32100000	andi \$16, \$16, 0x0000...	andi \$16, \$16, 0 * \$16 = 0

### Labels

Label	Address
mps2.asm	0x10010000

### Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x12345678	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
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

### Memory View

Address	Value
0x00000000	0x00000000
0x00000004	0x00000000
0x00000008	0x00000000
0x0000000c	0x00000000
0x00000010	0x00000000
0x00000014	0x00000000
0x00000018	0x00000000
0x0000001c	0x00000000
0x00000020	0x00000000
0x00000024	0x00000000
0x00000028	0x00000000
0x0000002c	0x00000000
0x00000030	0x00000000
0x00000034	0x00000000
0x00000038	0x00000000
0x0000003c	0x00000000
0x00000040	0x00000000
0x00000044	0x00000000
0x00000048	0x00000000
0x0000004c	0x00000000
0x00000050	0x00000000
0x00000054	0x00000000
0x00000058	0x00000000
0x0000005c	0x00000000
0x00000060	0x00000000
0x00000064	0x00000000
0x00000068	0x00000000
0x0000006c	0x00000000
0x00000070	0x00000000
0x00000074	0x00000000
0x00000078	0x00000000
0x0000007c	0x00000000
0x00000080	0x00000000
0x00000084	0x00000000
0x00000088	0x00000000
0x0000008c	0x00000000
0x00000090	0x00000000
0x00000094	0x00000000
0x00000098	0x00000000
0x0000009c	0x00000000
0x000000a0	0x00000000
0x000000a4	0x00000000
0x000000a8	0x00000000
0x000000ac	0x00000000
0x000000b0	0x00000000
0x000000b4	0x00000000
0x000000b8	0x00000000
0x000000bc	0x00000000
0x000000c0	0x00000000

### Bài 3:

a, abs \$s0,\$s1

\*Code:

```
1 # abs $s0, $s1
2 .text
3 addi $s1,$zero -5 # $s1 = -5
4 slt $t0, $s1, $zero # $t0 = $s1 < 0 ? 1 : 0
5 beq $t0, $zero, ABS # $t0 = 0 -> $s1>0 -> nhảy den ABS
6 sub $s0, $zero, $s1 # $s0 = -$s1
7 j EXIT
8 ABS:
9 add $s0, $zero, $s1 # $s0 = $s1
10 EXIT:
```

\*Kết quả chạy:

Text Segment					Labels		Name			Number	Value
Bkpt	Address	Code	Basic	Source	Label	Address					
	0x00400000	0x2011ffff	addi \$17,\$0,-5	3: addi \$s1,\$zero -5 # \$s1 = -5	mips3_1.asm			\$zero	0	0	
	0x00400004	0x2220402a	slt \$t0,\$17,\$0	4: slt \$t0, \$s1, \$zero # \$t0 = 0 ? 1 : 0	ABS	0x00400014		\$t0	1	0	
	0x00400008	0x11000002	beq \$t0,\$0,2	5: beq \$t0, \$zero, ABS # \$t0 = 0 -> \$s1>0 -> nay den ABS	EXT	0x00400018		\$t1	2	0	
	0x0040000c	0x00118022	sub \$16,\$0,\$17	6: sub \$s0, \$zero, \$s1 # \$s0 = -\$s1				\$t2	3	0	
	0x00400010	0x08100006	j EXIT	7: j EXIT				\$t3	4	0	
	0x00400014	0x00118020	add \$16,\$0,\$17	9: add \$s0, \$zero, \$s1 # \$s0 = \$s1				\$t4	5	0	
								\$t5	6	0	
								\$t6	7	0	
								\$t7	8	1	
								\$t8	9	0	
								\$t9	10	0	
								\$t10	11	0	
								\$t11	12	0	
								\$t12	13	0	
								\$t13	14	0	
								\$t14	15	0	
								\$t15	16	0	
								\$t16	17	0	
								\$t17	18	0	
								\$t18	19	0	
								\$t19	20	0	
								\$t20	21	0	
								\$t21	22	0	
								\$t22	23	0	
								\$t23	24	0	
								\$t24	25	0	
								\$t25	26	0	
								\$t26	27	0	
								\$t27	28	0	
								\$t28	29	268468224	
								\$t29	30	2147479548	
								\$t30	31	0	
								\$t31	32	0	
								\$t32	33	0	
								\$t33	34	0	
								\$t34	35	0	
								\$t35	36	0	
								\$t36	37	0	
								\$t37	38	0	
								\$t38	39	0	
								\$t39	40	0	
								\$t40	41	0	
								\$t41	42	0	
								\$t42	43	0	
								\$t43	44	0	
								\$t44	45	0	
								\$t45	46	0	
								\$t46	47	0	
								\$t47	48	0	
								\$t48	49	0	
								\$t49	50	0	
								\$t50	51	0	
								\$t51	52	0	
								\$t52	53	0	
								\$t53	54	0	
								\$t54	55	0	
								\$t55	56	0	
								\$t56	57	0	
								\$t57	58	0	
								\$t58	59	0	
								\$t59	60	0	
								\$t60	61	0	
								\$t61	62	0	
								\$t62	63	0	
								\$t63	64	0	
								\$t64	65	0	
								\$t65	66	0	
								\$t66	67	0	
								\$t67	68	0	
								\$t68	69	0	
								\$t69	70	0	
								\$t70	71	0	
								\$t71	72	0	
								\$t72	73	0	
								\$t73	74	0	
								\$t74	75	0	
								\$t75	76	0	
								\$t76	77	0	
								\$t77	78	0	
								\$t78	79	0	
								\$t79	80	0	
								\$t80	81	0	
								\$t81	82	0	
								\$t82	83	0	
								\$t83	84	0	
								\$t84	85	0	
								\$t85	86	0	
								\$t86	87	0	
								\$t87	88	0	
								\$t88	89	0	
								\$t89	90	0	
								\$t90	91	0	
								\$t91	92	0	
								\$t92	93	0	
								\$t93	94	0	
								\$t94	95	0	
								\$t95	96	0	
								\$t96	97	0	
								\$t97	98	0	
								\$t98	99	0	
								\$t99	100	0	
								\$t100	101	0	
								\$t101	102	0	
								\$t102	103	0	
								\$t103	104	0	
								\$t104	105	0	
								\$t105	106	0	
								\$t106	107	0	
								\$t107	108	0	
								\$t108	109	0	
								\$t109	110	0	
								\$t110	111	0	
								\$t111	112	0	
								\$t112	113	0	
								\$t113	114	0	
								\$t114	115	0	
								\$t115	116	0	
								\$t116	117	0	
								\$t117	118	0	
								\$t118	119	0	
								\$t119	120	0	
								\$t120	121	0	
								\$t121	122	0	
								\$t122	123	0	
								\$t123	124	0	
								\$t124	125	0	
								\$t125	126	0	
								\$t126	127	0	
								\$t127	128	0	
								\$t128	129	0	
								\$t129	130	0	
								\$t130	131	0	
								\$t131	132	0	
								\$t132	133	0	
								\$t133	134	0	
								\$t134	135	0	
								\$t135	136	0	
								\$t136	137	0	
								\$t137	138	0	
								\$t138	139	0	
								\$t139	140	0	
								\$t140	141	0	
								\$t141	142	0	
								\$t142	143	0	
								\$t143	144	0	
								\$t144	145	0	
								\$t145	146	0	
								\$t146	147	0	
								\$t147	148	0	
								\$t148	149	0	
								\$t149	150	0	
								\$t150	151	0	
								\$t151	152	0	
								\$t152	153	0	
								\$t153	154	0	
								\$t154	155	0	
								\$t155	156	0	
								\$t156	157	0	
								\$t157	158	0	
								\$t158	159	0	
								\$t159	160	0	
								\$t160	161	0	
								\$t161	162	0	
								\$t162	163	0	
								\$t163	164	0	
								\$t164	165	0	
								\$t165	166	0	
								\$t166	167	0	
								\$t167	168	0	
								\$t168	169	0	
								\$t169	170	0	
								\$t170	171	0	
								\$t171	172	0	
								\$t172	173	0	
								\$t173	174	0	
								\$t174	175	0	
								\$t175	176	0	
								\$t176	177	0	
								\$t177	178	0	
								\$t178	179	0	
								\$t179	180	0	
								\$t180	181	0	
								\$t181	182	0	
								\$t182	183	0	
								\$t183	184	0	
								\$t184	185	0	
								\$t185	186	0	
								\$t186	187	0	
								\$t187	188	0	
								\$t188	189	0	
								\$t189	190	0	
								\$t190	191	0	
								\$t191	192	0	
								\$t192	193	0	
								\$t193	194	0	
								\$t194	195	0	
								\$t195	196	0	
								\$t196	197	0	

c, not \$s0, \$s1

\*Code:

```
1 #not $s0, $s1
2 .text
3     addi $s1, $zero, 3
4     nor $s0, $s1, $zero
5
```

\*Kết quả chạy:

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x20110003	addi \$t7,\$0,0x00000003	3: addi \$s1, \$zero, 3
	0x00400004	0x02208027	nor \$t6,\$t7,\$0	4: nor \$s0, \$s1, \$zero

Data Segment

Labels

Name	Number	Value
\$zero	0	0x00000000
\$t1	1	0x00000000
\$t0	2	0x00000000
\$t1	3	0x00000000
\$t0	4	0x00000000
\$t1	5	0x00000000
\$t2	6	0x00000000
\$t3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$t0	16	0xffffffff
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000

☒ Data

☒ Text

d, ble \$s1, \$s2, label

\*Code:

```
1 #ble $s1, $s2, label
2 .text
3     addi $s1, $zero, 5 # $s1 =5
4     addi $s2, $zero, 5 # $s2 =5
5     slt $t0, $s2, $s1 # $t0 = $s2 < $s1 ? 1 : 0
6     beq $t0, $zero, label # $t0=0 -> $s1 <= $s2 nhay den label
7     add $s1, $s1, $s2 # $s1 = $s1 + $s2
8     j exit
9 label:
10    add $s2, $s1, $s2 # $s2 = $s1 + $s2
11    exit:
```

\*Kết quả chạy:

Text Segment

Bkpt

Address

Code

Basic

Source

0x00400000

0x20110005

addi \$t0, \$zero, 5 # \$s1 =5

3:

addi \$s1, \$zero, 5 # \$s1 =5

0x00400004

0x20110005

addi \$t0, \$zero, 5 # \$s2 =5

4:

addi \$s2, \$zero, 5 # \$s2 =5

0x00400008

0x0251402a

slt \$t0, \$s2, \$s1 # \$t0 = \$s2 < \$s1 ? 1 : 0

5:

slt \$t0, \$s2, \$s1 # \$t0 = \$s2 < \$s1 ? 1 : 0

0x0040000c

0x11000002

beq \$t0, \$zero, label # \$t0=0 -> \$s1 <= \$s2 nhay den label

6:

beq \$t0, \$zero, label # \$t0=0 -> \$s1 <= \$s2 nhay den label

0x00400010

0x02208027

add \$s1, \$s1, \$s2 # \$s1 = \$s1 + \$s2

7:

add \$s1, \$s1, \$s2 # \$s1 = \$s1 + \$s2

0x00400014

0x08100007

j exit

8:

j exit

0x00400018

0x02208027

add \$s2, \$s1, \$s2 # \$s2 = \$s1 + \$s2

10:

add \$s2, \$s1, \$s2 # \$s2 = \$s1 + \$s2

Labels

Label

Address

mips3\_4.asm

0x00400018

label

0x0040001c

exit

0x0040001c

Name

Number

Value

\$zero

0

0

\$at

1

0

\$v0

2

0

\$a0

3

0

\$a1

4

0

\$a2

5

0

\$a3

6

0

\$a4

7

0

\$a5

8

0

\$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

0

\$s3

19

0

\$s4

20

0

\$s5

21

0

\$s6

22

0

\$s7

23

0

\$t8

24

0

\$t9

25

0

\$s8

26

0

\$t1

27

0

\$s9

28

268468224

\$t0

29

2147479548

\$s9

30

0

\$ra

31

0

PC

4194332

\$1

0

\$e

0

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

0x10010020

0

0

0

0

0

0

0

0x10010040

0

0

0

0

0

0

0

0x10010060

0

0

0

0

0

0

0

0x10010080

0

0

0

0

0

0

0

0x100100a0

0

0

0

0

0

0

0

0x100100c0

0

0

0

0

0

0

0

0x100100e0

0

0

0

0

0

0

0

0x10010100

0

0

0

0

0

0

0

0x10010120

0

0

0

0

0

0

0

0x10010140

0

0

0

0

0

0

0

0x10010160

0

0

0

0

0

0

0

0x10010180

0

0

0

0

0

0

0



## Bài 4:

### -TH1: Cộng hai số khác dấu

#### \*Code:

```
1 .data
2 X: .word -5
3 Y: .word 5
4 .text
5     lw $s1,X
6     lw $s2,Y
7 start:
8     li $t0, 0 # trạng thái ban đầu không có tràn
9     addu $s3, $s1, $s2 # s3=s1+s2
10    xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0
11
12    bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
13    xor $t2, $s3, $s1 # $s3 và $s1 khác dấu $t2 < 0, ngược lại $t2 > 0
14    bltz $t2,OVERFLOW # nếu $t2 < 0, say ra tràn số, nhảy đến OVERFLOW
15    j EXIT
16 OVERFLOW:
17    li $t0, 1 # có tràn số -> t0 = 1
18 EXIT:
19
```

#### \*Kết quả chạy:

Text Segment

Block

Address

Code

Basic

Source

0x00400000

0x3c011001

lui \$1, 4097

5: lw \$s1, X

0x00400004

0x8c310000

lw \$17, 0(\$1)

0x00400008

0x3c011001

lui \$1, 4097

6: lw \$s2, Y

0x0040000c

0x8c320004

lw \$18, 4(\$1)

0x00400010

0x24080000

addiu \$8, \$0, 0

8: li \$t0, 0 # trạng thái ban đầu không có tràn

0x00400014

0x232282c2

addu \$19, \$17, \$18

9: addu \$s3, \$s1, \$s2 # s3=s1+s2

0x00400018

0x02324826

xor \$9, \$17, \$18

10: xor \$t1, \$s1, \$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác d...

0x0040001c

0x05200004

bltz \$9, 4

12: bltz \$t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra

0x00400020

0x02715028

xor \$10, \$19, \$17

13: xor \$t2, \$s3, \$s1 # \$s3 và \$s1 khác dấu \$t2 < 0, ngược lại \$t2 > 0

0x00400024

0x0540000c

bltz \$10, 1

14: bltz \$t2, OVERFLOW # nếu \$t2 < 0, say ra tràn số, nhảy đến OVERFLOW

0x00400028

0x0810000c

j 0x00400030

15: j EXIT

0x0040002c

0x24080001

addiu \$8, \$0, 1

17: li \$t0, 1 # có tràn số -> t0 = 1

<

-TH2: Cộng hai số dương không tràn số

\*Code:

```
1 .data
2 X: .word 10
3 Y: .word 5
4 .text
5 lw $s1,X
6 lw $s2,Y
7 start:
8 li $t0, 0 # trạng thái ban đầu không có tràn
9 addu $s3, $s1, $s2 # s3=s1+s2
10 xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0
11
12 bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
13 xor $t2, $s3, $s1 # $s3 và $s1 khác dấu $t2 < 0, ngược lại $t2 > 0
14 bltz $t2,OVERFLOW # nếu $t2 < 0, say ra tràn so, nhảy den OVERFLOW
15 j EXIT
16 OVERFLOW:
17 li $t0, 1 # có tràn số -> t0 = 1
18 EXIT:
19
```

\*Kết quả chạy:

Text Segment

Blockpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x3c011001	lui \$1,4097	5: lw \$s1,X
<input type="checkbox"/>	0x00400004	0x3c210000	lw \$17,0(\$1)	
<input type="checkbox"/>	0x00400008	0x3c011001	lui \$1,4097	6: lw \$s2,Y
<input type="checkbox"/>	0x0040000c	0x3c320004	lw \$18,4(\$1)	
<input type="checkbox"/>	0x00400010	0x24080000	addu \$8,\$0,\$0	8: li \$t0, 0 # trạng thái ban đầu không có tràn
<input type="checkbox"/>	0x00400014	0x02329821	addu \$19,\$17,\$18	9: addu \$s3, \$s1, \$s2 # s3=s1+s2
<input type="checkbox"/>	0x00400018	0x02324826	xor \$9,\$17,\$18	10: xor \$t1,\$s1,\$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác d...
<input type="checkbox"/>	0x0040001c	0x05200004	bltz \$9,4	12: bltz \$t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
<input type="checkbox"/>	0x00400020	0x02715026	xor \$10,\$19,\$17	13: xor \$t2, \$s3, \$s1 # \$s3 và \$s1 khác dấu \$t2 < 0, ngược lại \$t2 > 0
<input type="checkbox"/>	0x00400024	0x05400001	bltz \$10,1	14: bltz \$t2,OVERFLOW # nếu \$t2 < 0, say ra tràn so, nhảy den OVERFLOW
<input type="checkbox"/>	0x00400028	0x0810000c	j \$0x04000030	15: j EXIT
<input type="checkbox"/>	0x0040002c	0x24080000	addu \$8,\$0,\$1	17: li \$t0, 1 # có tràn số -> t0 = 1

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	10	5	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
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

Labels

Label	Address
mips4.asm	
start	0x00400010
OVERFLOW	0x0040002c
EXIT	0x00400030
X	0x10010000
Y	0x10010004

Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	15
\$t2	10	3
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	10
\$s2	18	5
\$s3	19	15
\$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		4194392
n1		0
lo		0

### -TH3: Cộng hai số dương có tràn số

#### \*Code:

```
1 .data
2 X: .word 0x7fffffff
3 Y: .word 5
4 .text
5     lw $s1,X
6     lw $s2,Y
7 start:
8     li $t0, 0 # trạng thái ban đầu không có tràn
9     addu $s3, $s1, $s2 # s3=s1+s2
10    xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0
11
12    bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
13    xor $t2, $s3, $s1 # $s3 và $s1 khác dấu $t2 < 0, ngược lại $t2 > 0
14    bltz $t2,OVERFLOW # nếu $t2 < 0, say ra tràn số, nhảy đến OVERFLOW
15    j EXIT
16 OVERFLOW:
17     li $t0, 1 # có tràn số -> t0 = 1
18 EXIT:
19
```

#### \*Kết quả chạy:

Text Segment				Labels		Name			Number	Value
Bkpt	Address	Code	Basic	Source	Label	Address				
	0x00400000	0x3c011001	lui \$1,4097				\$zero	0	0	
	0x00400004	0x8c310000	lw \$17,0(\$1)	5: lw \$s1,X			\$a1	1	268500992	
	0x00400008	0x3c011001	lui \$1,4097				\$a0	2	0	
	0x0040000c	0x8c320004	lw \$18,4(\$1)	6: lw \$s2,Y			\$v1	3	0	
	0x00400010	0x24080000	addiu \$8,\$0,0	8: li \$t0, 0 # trạng thái ban đầu không có tràn	start	0x00400010	\$a0	4	0	
	0x00400014	0x02229821	addu \$19,\$17,\$18	9: addu \$s3, \$s1, \$s2	OVERFLOW	0x0040002c	\$a1	5	0	
	0x00400018	0x02224826	xor \$9,\$17,\$18	10: xor \$t1,\$s1,\$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác d...	EXIT	0x00400030	\$a2	6	0	
	0x0040001c	0x05200004	bltz \$9,4	12: bltz \$t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra	X	0x10010000	\$a3	7	0	
	0x00400020	0x02715026	xor \$10,\$19,\$17	13: xor \$t2, \$s3, \$s1 # \$s3 và \$s1 khác dấu \$t2 < 0, ngược lại \$t2 > 0	Y	0x10010004	\$t0	8	0	
	0x00400024	0x05400001	bltz \$10,1	14: bltz \$t2,OVERFLOW # nếu \$t2 < 0, say ra tràn số, nhảy đến OVERFLOW			\$t1	9	2147483642	
	0x00400028	0x0810000c	j EXIT	15: j EXIT			\$t2	10	-5	
	0x0040002c	0x24080001	addiu \$8,\$0,1	17: li \$t0, 1 # có tràn số -> t0 = 1			\$t3	11	0	
							\$t4	12	0	
							\$t5	13	0	
							\$t6	14	0	
							\$t7	15	0	
							\$t8	16	0	
							\$t9	17	0	
							\$s0	18	2147483647	
							\$s1	19	-2147483644	
							\$s2	20	0	
							\$s3	21	0	
							\$s4	22	0	
							\$s5	23	0	
							\$s6	24	0	
							\$s7	25	0	
							\$s8	26	0	
							\$s9	27	0	
							\$sp	28	268488224	
							\$fp	29	2147479548	
							\$tp	30	0	
							\$ra	31	0	
							pc		4194352	
							\$1		0	
							\$e		0	

Data Segment	Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
	0x10010000	2147483647	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
	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

**\*Code:**

```

1  .data
2  X: .word -6
3  Y: .word -10
4  .text
5      lw $s1,X
6      lw $s2,Y
7  start:
8      li $t0, 0 # trạng thái ban đầu không có tràn
9      addu $s3, $s1, $s2 # s3=s1+s2
10     xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0
11
12     bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
13     xor $t2, $s3, $s1 # $s3 và $s1 khác dấu $t2 < 0, ngược lại $t2 > 0
14     bltz $t2,OVERFLOW # nếu $t2 < 0, say ra tràn số, nhảy đến OVERFLOW
15     j EXIT
16 OVERFLOW:
17     li $t0, 1 # có tràn số -> t0 = 1
18 EXIT:
19

```

[illegible]

## -TH5: Cộng hai số âm có tràn số

### \*Code:

```
1 .data
2 X: .word 0x80000000
3 Y: .word -10
4 .text
5 lw $s1,X
6 lw $s2,Y
7 start:
8 li $t0, 0 # trạng thái ban đầu không có tràn
9 addu $s3, $s1, $s2 # s3=s1+s2
10 xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0
11
12 bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
13 xor $t2, $s3, $s1 # $s3 và $s1 khác dấu $t2 < 0, ngược lại $t2 > 0
14 bltz $t2,OVERFLOW # nếu $t2 < 0, say ra tràn số, nhảy đến OVERFLOW
15 j EXIT
16 OVERFLOW:
17 li $t0, 1 # có tràn số -> t0 = 1
18 EXIT:
19
```

### \*Kết quả chạy:

The screenshot displays the Mips4.asm assembly viewer with three main panels: Text Segment, Labels, and Data Segment.

**Text Segment:** Shows the assembly code with addresses and sources. The code is as follows:

```
0x00400000: 0x3c011001 lui $1,4097
0x00400004: 0x8c310000 lw $17,0($1)
0x00400008: 0x3c011001 lui $1,4097
0x0040000c: 0x8c320004 lw $18,4($1)
0x00400010: 0x24080000 addiu $8,$0,0
0x00400014: 0x02320921 addu $19,$17,$18
0x00400018: 0x023204826 xor $9,$17,$18
0x0040001c: 0x05200004 bltz $9,4
0x00400020: 0x02725026 xor $10,$19,$17
0x00400024: 0x05400001 bltz $10,1
0x00400028: 0x0810000c j EXIT
0x0040002c: 0x24080001 addiu $8,$0,1
```

**Labels:** Shows the labels and their addresses:

```
start 0x00400010
OVERFLOW 0x0040002c
EXIT 0x0040000c
X 0x10010000
Y 0x10010004
```

**Data Segment:** Shows the data segment with addresses and values. The data is as follows:

```
0x10010000: 2147483648
0x10010020: 0
0x10010040: 0
0x10010060: 0
0x10010080: 0
0x100100a0: 0
0x100100c0: 0
0x100100e0: 0
0x10010100: 0
0x10010120: 0
0x10010140: 0
0x10010160: 0
0x10010180: 0
0x100101a0: 0
0x100101c0: 0
```

The Data Segment also shows the values of the registers \$s1, \$s2, \$s3, \$t0, \$t1, \$t2, \$t3, \$t4, \$t5, \$t6, \$t7, \$t8, \$t9, \$t10, \$t11, \$t12, \$t13, \$t14, \$t15, \$t16, \$t17, \$t18, \$t19, \$t20, \$t21, \$t22, \$t23, \$t24, \$t25, \$t26, \$t27, \$t28, \$t29, \$t30, \$t31, \$t32, \$t33, \$t34, \$t35, \$t36, \$t37, \$t38, \$t39, \$t40, \$t41, \$t42, \$t43, \$t44, \$t45, \$t46, \$t47, \$t48, \$t49, \$t50, \$t51, \$t52, \$t53, \$t54, \$t55, \$t56, \$t57, \$t58, \$t59, \$t60, \$t61, \$t62, \$t63, \$t64, \$t65, \$t66, \$t67, \$t68, \$t69, \$t70, \$t71, \$t72, \$t73, \$t74, \$t75, \$t76, \$t77, \$t78, \$t79, \$t80, \$t81, \$t82, \$t83, \$t84, \$t85, \$t86, \$t87, \$t88, \$t89, \$t90, \$t91, \$t92, \$t93, \$t94, \$t95, \$t96, \$t97, \$t98, \$t99, \$t100, \$t101, \$t102, \$t103, \$t104, \$t105, \$t106, \$t107, \$t108, \$t109, \$t110, \$t111, \$t112, \$t113, \$t114, \$t115, \$t116, \$t117, \$t118, \$t119, \$t120, \$t121, \$t122, \$t123, \$t124, \$t125, \$t126, \$t127, \$t128, \$t129, \$t130, \$t131, \$t132, \$t133, \$t134, \$t135, \$t136, \$t137, \$t138, \$t139, \$t140, \$t141, \$t142, \$t143, \$t144, \$t145, \$t146, \$t147, \$t148, \$t149, \$t150, \$t151, \$t152, \$t153, \$t154, \$t155, \$t156, \$t157, \$t158, \$t159, \$t160, \$t161, \$t162, \$t163, \$t164, \$t165, \$t166, \$t167, \$t168, \$t169, \$t170, \$t171, \$t172, \$t173, \$t174, \$t175, \$t176, \$t177, \$t178, \$t179, \$t180, \$t181, \$t182, \$t183, \$t184, \$t185, \$t186, \$t187, \$t188, \$t189, \$t190, \$t191, \$t192, \$t193, \$t194, \$t195, \$t196, \$t197, \$t198, \$t199, \$t200, \$t201, \$t202, \$t203, \$t204, \$t205, \$t206, \$t207, \$t208, \$t209, \$t210, \$t211, \$t212, \$t213, \$t214, \$t215, \$t216, \$t217, \$t218, \$t219, \$t220, \$t221, \$t222, \$t223, \$t224, \$t225, \$t226, \$t227, \$t228, \$t229, \$t230, \$t231, \$t232, \$t233, \$t234, \$t235, \$t236, \$t237, \$t238, \$t239, \$t240, \$t241, \$t242, \$t243, \$t244, \$t245, \$t246, \$t247, \$t248, \$t249, \$t250, \$t251, \$t252, \$t253, \$t254, \$t255, \$t256, \$t257, \$t258, \$t259, \$t260, \$t261, \$t262, \$t263, \$t264, \$t265, \$t266, \$t267, \$t268, \$t269, \$t270, \$t271, \$t272, \$t273, \$t274, \$t275, \$t276, \$t277, \$t278, \$t279, \$t280, \$t281, \$t282, \$t283, \$t284, \$t285, \$t286, \$t287, \$t288, \$t289, \$t290, \$t291, \$t292, \$t293, \$t294, \$t295, \$t296, \$t297, \$t298, \$t299, \$t300, \$t301, \$t302, \$t303, \$t304, \$t305, \$t306, \$t307, \$t308, \$t309, \$t310, \$t311, \$t312, \$t313, \$t314, \$t315, \$t316, \$t317, \$t318, \$t319, \$t320, \$t321, \$t322, \$t323, \$t324, \$t325, \$t326, \$t327, \$t328, \$t329, \$t330, \$t331, \$t332, \$t333, \$t334, \$t335, \$t336, \$t337, \$t338, \$t339, \$t340, \$t341, \$t342, \$t343, \$t344, \$t345, \$t346, \$t347, \$t348, \$t349, \$t350, \$t351, \$t352, \$t353, \$t354, \$t355, \$t356, \$t357, \$t358, \$t359, \$t360, \$t361, \$t362, \$t363, \$t364, \$t365, \$t366, \$t367, \$t368, \$t369, \$t370, \$t371, \$t372, \$t373, \$t374, \$t375, \$t376, \$t377, \$t378, \$t379, \$t380, \$t381, \$t382, \$t383, \$t384, \$t385, \$t386, \$t387, \$t388, \$t389, \$t390, \$t391, \$t392, \$t393, \$t394, \$t395, \$t396, \$t397, \$t398, \$t399, \$t400, \$t401, \$t402, \$t403, \$t404, \$t405, \$t406, \$t407, \$t408, \$t409, \$t410, \$t411, \$t412, \$t413, \$t414, \$t415, \$t416, \$t417, \$t418, \$t419, \$t420, \$t421, \$t422, \$t423, \$t424, \$t425, \$t426, \$t427, \$t428, \$t429, \$t430, \$t431, \$t432, \$t433, \$t434, \$t435, \$t436, \$t437, \$t438, \$t439, \$t440, \$t441, \$t442, \$t443, \$t444, \$t445, \$t446, \$t447, \$t448, \$t449, \$t450, \$t451, \$t452, \$t453, \$t454, \$t455, \$t456, \$t457, \$t458, \$t459, \$t460, \$t461, \$t462, \$t463, \$t464, \$t465, \$t466, \$t467, \$t468, \$t469, \$t470, \$t471, \$t472, \$t473, \$t474, \$t475, \$t476, \$t477, \$t478, \$t479, \$t480, \$t481, \$t482, \$t483, \$t484, \$t485, \$t486, \$t487, \$t488, \$t489, \$t490, \$t491, \$t492, \$t493, \$t494, \$t495, \$t496, \$t497, \$t498, \$t499, \$t500, \$t501, \$t502, \$t503, \$t504, \$t505, \$t506, \$t507, \$t508, \$t509, \$t510, \$t511, \$t512, \$t513, \$t514, \$t515, \$t516, \$t517, \$t518, \$t519, \$t520, \$t521, \$t522, \$t523, \$t524, \$t525, \$t526, \$t527, \$t528, \$t529, \$t530, \$t531, \$t532, \$t533, \$t534, \$t535, \$t536, \$t537, \$t538, \$t539, \$t540, \$t541, \$t542, \$t543, \$t544, \$t545, \$t546, \$t547, \$t548, \$t549, \$t550, \$t551, \$t552, \$t553, \$t554, \$t555, \$t556, \$t557, \$t558, \$t559, \$t560, \$t561, \$t562, \$t563, \$t564, \$t565, \$t566, \$t567, \$t568, \$t569, \$t570, \$t571, \$t572, \$t573, \$t574, \$t575, \$t576, \$t577, \$t578, \$t579, \$t580, \$t581, \$t582, \$t583, \$t584, \$t585, \$t586, \$t587, \$t588, \$t589, \$t590, \$t591, \$t592, \$t593, \$t594, \$t595, \$t596, \$t597, \$t598, \$t599, \$t600, \$t601, \$t602, \$t603, \$t604, \$t605, \$t606, \$t607, \$t608, \$t609, \$t610, \$t611, \$t612, \$t613, \$t614, \$t615, \$t616, \$t617, \$t618, \$t619, \$t620, \$t621, \$t622, \$t623, \$t624, \$t625, \$t626, \$t627, \$t628, \$t629, \$t630, \$t631, \$t632, \$t633, \$t634, \$t635, \$t636, \$t637, \$t638, \$t639, \$t640, \$t641, \$t642, \$t643, \$t644, \$t645, \$t646, \$t647, \$t648, \$t649, \$t650, \$t651, \$t652, \$t653, \$t654, \$t655, \$t656, \$t657, \$t658, \$t659, \$t660, \$t661, \$t662, \$t663, \$t664, \$t665, \$t666, \$t667, \$t668, \$t669, \$t670, \$t671, \$t672, \$t673, \$t674, \$t675, \$t676, \$t677, \$t678, \$t679, \$t680, \$t681, \$t682, \$t683, \$t684, \$t685, \$t686, \$t687, \$t688, \$t689, \$t690, \$t691, \$t692, \$t693, \$t694, \$t695, \$t696, \$t697, \$t698, \$t699, \$t700, \$t701, \$t702, \$t703, \$t704, \$t705, \$t706, \$t707, \$t708, \$t709, \$t710, \$t711, \$t712, \$t713, \$t714, \$t715, \$t716, \$t717, \$t718, \$t719, \$t720, \$t721, \$t722, \$t723, \$t724, \$t725, \$t726, \$t727, \$t728, \$t729, \$t730, \$t731, \$t732, \$t733, \$t734, \$t735, \$t736, \$t737, \$t738, \$t739, \$t740, \$t741, \$t742, \$t743, \$t744, \$t745, \$t746, \$t747, \$t748, \$t749, \$t750, \$t751, \$t752, \$t753, \$t754, \$t755, \$t756, \$t757, \$t758, \$t759, \$t760, \$t761, \$t762, \$t763, \$t764, \$t765, \$t766, \$t767, \$t768, \$t769, \$t770, \$t771, \$t772, \$t773, \$t774, \$t775, \$t776, \$t777, \$t778, \$t779, \$t780, \$t781, \$t782, \$t783, \$t784, \$t785, \$t786, \$t787, \$t788, \$t789, \$t790, \$t791, \$t792, \$t793, \$t794, \$t795, \$t796, \$t797, \$t798, \$t799, \$t800, \$t801, \$t802, \$t803, \$t804, \$t805, \$t806, \$t807, \$t808, \$t809, \$t810, \$t811, \$t812, \$t813, \$t814, \$t815, \$t816, \$t817, \$t818, \$t819, \$t820, \$t821, \$t822, \$t823, \$t824, \$t825, \$t826, \$t827, \$t828, \$t829, \$t830, \$t831, \$t832, \$t833, \$t834, \$t835, \$t836, \$t837, \$t838, \$t839, \$t840, \$t841, \$t842, \$t843, \$t844, \$t845, \$t846, \$t847, \$t848, \$t849, \$t850, \$t851, \$t852, \$t853, \$t854, \$t855, \$t856, \$t857, \$t858, \$t859, \$t860, \$t861, \$t862, \$t863, \$t864, \$t865, \$t866, \$t867, \$t868, \$t869, \$t870, \$t871, \$t872, \$t873, \$t874, \$t875, \$t876, \$t877, \$t878, \$t879, \$t880, \$t881, \$t882, \$t883, \$t884, \$t885, \$t886, \$t887, \$t888, \$t889, \$t890, \$t891, \$t892, \$t893, \$t894, \$t895, \$t896, \$t897, \$t898, \$t899, \$t900, \$t901, \$t902, \$t903, \$t904, \$t905, \$t906, \$t907, \$t908, \$t909, \$t910, \$t911, \$t912, \$t913, \$t914, \$t915, \$t916, \$t917, \$t918, \$t919, \$t920, \$t921, \$t922, \$t923, \$t924, \$t925, \$t926, \$t927, \$t928, \$t929, \$t930, \$t931, \$t932, \$t933, \$t934, \$t935, \$t936, \$t937, \$t938, \$t939, \$t940, \$t941, \$t942, \$t943, \$t944, \$t945, \$t946, \$t947, \$t948, \$t949, \$t950, \$t951, \$t952, \$t953, \$t954, \$t955, \$t956, \$t957, \$t958, \$t959, \$t960, \$t961, \$t962, \$t963, \$t964, \$t965, \$t966, \$t967, \$t968, \$t969, \$t970, \$t971, \$t972, \$t973, \$t974, \$t975, \$t976, \$t977, \$t978, \$t979, \$t980, \$t981, \$t982, \$t983, \$t984, \$t985, \$t986, \$t987, \$t988, \$t989, \$t990, \$t991, \$t992, \$t993, \$t994, \$t995, \$t996, \$t997, \$t998, \$t999, \$t1000, \$t1001, \$t1002, \$t1003, \$t1004, \$t1005, \$t1006, \$t1007, \$t1008, \$t1009, \$t1010, \$t1011, \$t1012, \$t1013, \$t1014, \$t1015, \$t1016, \$t1017, \$t1018, \$t1019, \$t1020, \$t1021, \$t1022, \$t1023, \$t1024, \$t1025, \$t1026, \$t1027, \$t1028, \$t1029, \$t1030, \$t1031, \$t1032, \$t1033, \$t1034, \$t1035, \$t1036, \$t1037, \$t1038, \$t1039, \$t1040, \$t1041, \$t1042, \$t1043, \$t1044, \$t1045, \$t1046, \$t1047, \$t1048, \$t1049, \$t1050, \$t1051, \$t1052, \$t1053, \$t1054, \$t1055, \$t1056, \$t1057, \$t1058, \$t1059, \$t1060, \$t1061, \$t1062, \$t1063, \$t1064, \$t1065, \$t1066, \$t1067, \$t1068, \$t1069, \$t1070, \$t1071, \$t1072, \$t1073, \$t1074, \$t1075, \$t1076, \$t1077, \$t1078, \$t1079, \$t1080, \$t1081, \$t1082, \$t1083, \$t1084, \$t1085, \$t1086, \$t1087, \$t1088, \$t1089, \$t1090, \$t1091, \$t1092, \$t1093, \$t1094, \$t1095, \$t1096, \$t1097, \$t1098, \$t1099, \$t1100, \$t1101, \$t1102, \$t1103, \$t1104, \$t1105, \$t1106, \$t1107, \$t1108, \$t1109, \$t1110, \$t1111, \$t1112, \$t1113, \$t1114, \$t1115, \$t1116, \$t1117, \$t1118, \$t1119, \$t1120, \$t1121, \$t1122, \$t1123, \$t1124, \$t1125, \$t1126, \$t1127, \$t1128, \$t1129, \$t1130, \$t1131, \$t1132, \$t1133, \$t1134, \$t1135, \$t1136, \$t1137, \$t1138, \$t1139, \$t1140, \$t1141, \$t1142, \$t1143, \$t1144, \$t1145, \$t1146, \$t1147, \$t1148, \$t1149, \$t1150, \$t1151, \$t1152, \$t1153, \$t1154, \$t1155, \$t1156, \$t1157, \$t1158, \$t1159, \$t1160, \$t1161, \$t1162, \$t1163, \$t1164, \$t1165, \$t1166, \$t1167, \$t1168, \$t1169, \$t1170, \$t1171, \$t1172, \$t1173, \$t1174, \$t1175, \$t1176, \$t1177, \$t1178, \$t1179, \$t1180, \$t1181, \$t1182, \$t1183, \$t1184, \$t1185, \$t1186, \$t1187, \$t1188, \$t1189, \$t1190, \$t1191, \$t1192, \$t1193, \$t1194, \$t1195, \$t1196, \$t1197, \$t1198, \$t1199, \$t1200, \$t1201, \$t1202, \$t1203, \$t1204, \$t1205, \$t1206, \$t1207, \$t1208, \$t1209, \$t1210, \$t1211, \$t1212, \$t1213, \$t1214, \$t1215, \$t1216, \$t1217, \$t1218, \$t1219, \$t1220, \$t1221, \$t1222, \$t1223, \$t1224, \$t1225, \$t1226, \$t1227, \$t1228, \$t1229, \$t1230, \$t1231, \$t1232, \$t1233, \$t1234, \$t1235, \$t1236, \$t1237, \$t1238, \$t1239, \$t1240, \$t1241, \$t1242, \$t1243, \$t1244, \$t1245, \$t1246, \$t1247, \$t1248, \$t1249, \$t1250, \$t1251, \$t1252, \$t1253, \$t1254, \$t1255, \$t1256, \$t1257, \$t1258, \$t1259, \$t1260, \$t1261, \$t1262, \$t1263, \$t1264, \$t1265, \$t1266, \$t1267, \$t1268, \$t1269, \$t1270, \$t1271, \$t1272, \$t1273, \$t1274, \$t1275, \$t1276, \$t1277, \$t1278, \$t1279, \$t1280, \$t1281, \$t1282, \$t1283, \$t1284, \$t1285, \$t1286, \$t1287, \$t1288, \$t1289, \$t1290, \$t1291, \$t1292, \$t1293, \$t1294, \$t1295, \$t1296, \$t1297, \$t1298, \$t1299, \$t1300, \$t1301, \$t1302, \$t1303, \$t1304, \$t1305, \$t1306, \$t1307, \$t1308, \$t1309, \$t1310, \$t1311, \$t1312, \$t1313, \$t1314, \$t1315, \$t1316, \$t1317, \$t1318, \$t1319, \$t1320, \$t1321, \$t1322, \$t1323, \$t1324, \$t1325, \$t1326, \$t1327, \$t1328, \$t1329, \$t1330, \$t1331, \$t1332, \$t1333, \$t1334, \$t1335, \$t1336, \$t1337, \$t1338, \$t1339, \$t1340, \$t1341, \$t1342, \$t1343, \$t1344, \$t1345, \$t1346, \$t1347, \$t1348, \$t1349, \$t1350, \$t1351, \$t1352, \$t1353, \$t1354, \$t1355, \$t1356, \$t1357, \$t1358, \$t1359, \$t1360, \$t1361, \$t1362, \$t1363, \$t1364, \$t1365, \$t1366, \$t1367, \$t1368, \$t1369, \$t1370, \$t1371, \$t1372, \$t1373, \$t1374, \$t1375, \$t1376, \$t1377, \$t1378, \$t1379, \$t1380, \$t1381, \$t1382, \$t1383, \$t1384, \$t1385, \$t1386, \$t1387, \$t1388, \$t1389, \$t1390, \$t1391, \$t1392, \$t1393, \$t1394, \$t1395, \$t1396, \$t1397, \$t1398, \$t1399, \$t1400, \$t1401, \$t1402, \$t1403, \$t1404, \$t1405, \$t1406, \$t1407, \$t1408, \$t1409, \$t1410, \$t1411, \$t1412, \$t1413, \$t1414, \$t1415, \$t1416, \$t1417, \$t1418, \$t1419, \$t1420, \$t1421, \$t1422, \$t1423, \$t1424, \$t1425, \$t1426, \$t1427, \$t1428, \$t1429, \$t1430, \$t1431, \$t1432, \$t1433, \$t1434, \$t1435, \$t1436, \$t1437, \$t1438, \$t1439, \$t1440, \$t1441, \$t1442, \$t1443, \$t1444, \$t1445, \$t1446, \$t1447, \$t1448, \$t1449, \$t1450, \$t1451, \$t1452, \$t1453, \$t1454, \$t1455, \$t1456, \$t1457, \$t1458, \$t1459, \$t1460, \$t1461, \$t1462, \$t1463, \$t1464, \$t1465, \$t1466, \$t1467, \$t1468, \$t1469, \$t1470, \$t1471, \$t1472, \$t1473, \$t1474, \$t1475, \$t1476, \$t1477, \$t1478, \$t1479, \$t1480, \$t1481, \$t1482, \$t1483, \$t1484, \$t1485, \$t1486, \$t1487, \$t1488, \$t1489, \$t1490, \$t1491, \$t1492, \$t1493, \$t1494, \$t1495, \$t1496, \$t1497, \$t1498, \$t1499, \$t1500, \$t1501, \$t1502, \$t1503, \$t1504, \$t1505, \$t1506, \$t1507, \$t1508, \$t1509, \$t1510, \$t1511, \$t1512, \$t1513, \$t1514, \$t1515, \$t1516, \$t1517, \$t1518, \$t1519, \$t1520, \$t1521, \$t1522, \$t1523, \$t1524, \$t1525, \$t1526, \$t1527, \$t1528, \$t1529, \$t1530, \$t1531, \$t1532, \$t1533, \$t1534, \$t1535, \$t1536, \$t1537, \$t1538, \$t1539, \$t1540, \$t1541, \$t1542, \$t1543, \$t1544, \$t1545, \$t1546, \$t1547, \$t1548, \$t1549, \$t1550, \$t1551, \$t1552, \$t1553, \$t1554, \$t1555, \$t1556, \$t1557, \$t1558, \$t1559, \$t1560, \$t1561, \$t1562, \$t1563, \$t1564, \$t1565, \$t1566, \$t1567, \$t1568, \$t1569, \$t1570, \$t1571, \$t1572, \$t1573, \$t1574, \$t1575, \$t1576, \$t1577, \$t1578, \$t1579, \$t1580, \$t1581, \$t1582, \$t1583, \$t1584, \$t1585, \$t1586, \$t1587, \$t1588, \$t1589, \$t1590, \$t1591, \$t1592, \$t1593, \$t1594, \$t1595, \$t1596, \$t1597, \$t1598, \$t1599, \$t1600, \$t1601, \$t1602, \$t1603, \$t1604, \$t1605, \$t1606, \$t1607, \$t1608, \$t1609, \$t1610, \$t1611, \$t1612, \$t1613, \$t1614, \$t1615, \$t1616, \$t1617, \$t1618, \$t1619, \$t1620, \$t1621, \$t1622, \$t1623, \$t1624, \$t1625, \$t1626, \$t1627, \$t1628, \$t1629, \$t1630, \$t1631, \$t1632, \$t1633, \$t1634, \$t1635, \$t1636, \$t1637, \$t1638, \$t1639, \$t1640, \$t1641, \$t1642, \$t1643, \$t1644, \$t1645, \$t1646, \$t1647, \$t1648, \$t1649, \$t1650, \$t1651, \$t1652, \$t1653, \$t1654, \$t1655, \$t1656, \$t1657, \$t1658, \$t1659, \$t1660, \$t1661, \$t1662, \$t1663, \$t1664, \$t1665, \$t1666, \$t1667, \$t1668, \$t1669, \$t1670, \$t1671, \$t1672, \$t1673, \$t1674, \$t1675, \$t1676, \$t1677, \$t1678, \$t1679, \$t1680, \$t1681, \$t1682, \$t1683, \$t1684, \$t1685, \$t1686, \$t1687, \$t1688, \$t1689, \$t1690, \$t1691, \$t1692, \$t1693, \$t1694, \$t1695, \$t1696, \$t1697, \$t1698, \$t1699, \$t1700, \$t1701, \$t1702, \$t1703, \$t1704, \$t1705, \$t1706, \$t1707, \$t1708, \$t1709, \$t1710, \$t1711, \$t1712, \$t1713, \$t1714, \$t1715, \$t1716, \$t1717, \$t1718, \$t1719, \$t1720, \$t1721, \$t1722, \$t1723, \$t1724, \$t1725, \$t1726, \$t1727, \$t1728, \$t1729, \$t1730, \$t1731, \$t1732, \$t1733, \$t1734, \$t1735, \$t1736, \$t1737, \$t1738, \$t1739, \$t1740, \$t1741, \$t1742, \$t1743, \$t1744, \$t1

## Bài 5:

### \*Code:

```
1 .data
2 x: .word 6 # x = 6
3 y: .word 8 # y = 8
4 p: .word 0
5 i: .word 0 # i = 0
6 .text
7 lw $s1, x # $s1 = x
8 lw $s2, y # $s2 = y
9 lw $s3, i # $s3 = i
10 la $t8, p
11 li $t0, 1
12 loop:
13 beq $s2, $t0, enloop # y = 1, ket thuc vong lap
14 srl $s2, $s2, 1 # y = y : 2
15 addi $s3, $s3, 1 # i = i + 1
16 j loop
17 enloop:
18 sllv $t1, $s1, $s3 # p = x*y
19 sw $t1, 0($t8)
```

### \*Kết quả chạy:

The screenshot displays a MIPS assembly simulator interface. The main window is divided into several panes:

- Text Segment:** Shows the assembly code being executed. The code is as follows:

```
0x00400000: 0x3c011001 lui $1, 0x0001
0x00400004: 0x8c310000 lw $t1, 0($1)
0x00400008: 0x3c011001 lui $1, 0x0001
0x0040000c: 0x8c320004 lw $t8, 4($1)
0x00400010: 0x3c011001 lui $1, 0x0001
0x00400014: 0x8c33000c lw $t0, 12($1)
0x00400018: 0x3c011001 lui $1, 0x0001
0x0040001c: 0x34380008 ori $t4, $1, 8
0x00400020: 0x24080001 addiu $8, $0, 1
0x00400024: 0x12480003 beq $t8, $8, 3
0x00400028: 0x00299424 srl $s2, $s2, 1 # y = y : 2
0x0040002c: 0x22730001 addi $s3, $s3, 1 # i = i + 1
0x00400030: 0x08100009 j loop
0x00400034: 0x02714804 sllv $t1, $s1, $s3 # p = x*y
0x00400038: 0xa1090000 sw $t1, 0($t8)
```
- Data Segment:** Shows the memory layout. The value at address 0x10010000 (register \$s1) is 48.
- Registers:** Shows the state of registers. The register \$s1 contains the value 48, and \$t1 contains 268468224.