

## BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH TUẦN 4

### Assignment 1

#### Code:

#Laboratory Exercise 4, Home Assignment 1

.text

start:

li \$s1,1

li \$s2,-2

li \$t0,0 #No Overflow is default status

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

xor \$t1,\$s1,\$s2 #Test if \$s1 and \$s2 have the same sign

bltz \$t1,EXIT #If not: t1<0, exit

slt \$t2,\$s3,\$s1 # if \$s3 < \$s1 => \$t2 = 1

bltz \$s1,NEGATIVE #Test if \$s1 and \$s2 is negative?

beq \$t2,\$zero,EXIT #if t2=0=> s3>s1 (s1 and \$s2 are positive)

# if \$s3 > \$s1 then the result is not overflow =>jump EXIT

j OVERFLOW

NEGATIVE:

bne \$t2,\$zero,EXIT #if t2!=0 => \$s3<s1 (s1 and \$s2 are negative)

# if \$s3 < \$s1 then the result is not overflow =>jump EXIT

OVERFLOW:

li \$t0,1 #the result is overflow1

EXIT:

## Nhận xét

- TH1: Cộng 2 số dương không tràn bộ nhớ
  - Lệnh: li \$s1,1  
li \$s2,2

Text Segment				
Bkpt	Address	Code	Basic	Source
	0x00400000	0x24110001	addiu \$17,\$0,1	4: li \$s1,1
	0x00400004	0x24120002	addiu \$18,\$0,2	5: li \$s2,2
	0x00400008	0x24080000	addiu \$8,\$0,0	7: li \$t0,0 #No Overflow is default status
	0x0040000c	0x02324821	addu \$19,\$17,\$18	8: addu \$s3,\$s1,\$s2 # s3 = s1 + s2
	0x00400010	0x02324826	xor \$9,\$17,\$18	9: xor \$t1,\$s1,\$s2 #Test if \$s1 and \$s2 have the same sign
	0x00400014	0x05200006	bltz \$9,6	10: bltz \$t1,EXIT #if not: t1<0, exit
	0x00400018	0x0271502a	slti \$10,\$19,\$17	12: slti \$t2,\$s3,\$s1 # if \$s3 < \$s1 => \$t2 = 1
	0x0040001c	0x06200002	bltz \$17,2	13: bltz \$s1,NEGATIVE #Test if \$s1 and \$s2 is negative?
	0x00400020	0x11400003	beq \$10,\$0,3	14: beq \$t2,\$zero,EXIT #if t2=0=> s3>s1 (s1 and \$s2 are positive)
	0x00400024	0x0810000b	j 0x0040002c	16: j OVERFLOW
	0x00400028	0x15400001	bne \$10,\$0,1	18: bne \$t2,\$zero,EXIT #if t2!=0 => \$s3<s1 (s1 and \$s2 are negative)
	0x0040002c	0x24080001	addiu \$8,\$0,1	21: li \$t0,1 #the result is overflow1

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	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	1
\$s2	18	2
\$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		4194312
hi		0
lo		0

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

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

- Lệnh: li \$t0,0 #No Overflow is default status  
⇒ Gán giá trị 0 là trạng thái không tràn bộ nhớ cho \$t0
- addu \$s3,\$s1,\$s2  
⇒ s3 = s1 + s2
- xor \$t1,\$s1,\$s2 #Test if \$s1 and \$s2 have the same sign

Text Segment							
Bkpt	Address	Code	Basic	Source	Name	Number	Value
<input type="checkbox"/>	0x00400000	0x24110001	addiu \$17,\$0,1	4: li \$s1,1	\$zero	0	0
<input type="checkbox"/>	0x00400004	0x24120002	addiu \$18,\$0,2	5: li \$s2,2	\$at	1	0
<input type="checkbox"/>	0x00400008	0x24080000	addiu \$8,\$0,0	7: li \$t0,0 #No Overflow is default status	\$v0	2	0
<input type="checkbox"/>	0x0040000c	0x02329821	addu \$19,\$17,\$18	8: addu \$s3,\$s1,\$s2 # s3 = s1 + s2	\$v1	3	0
<input type="checkbox"/>	0x00400010	0x02324826	xor \$9,\$17,\$18	9: xor \$t1,\$s1,\$s2 #test if \$s1 and \$s2 have the same sign	\$a0	4	0
<input type="checkbox"/>	0x00400014	0x02000006	bltz \$9,6	10: bltz \$t1,EXIT #if not: t1<0, exit	\$a1	5	0
<input type="checkbox"/>	0x00400018	0x0271502a	slt \$10,\$19,\$17	12: slt \$t2,\$s3,\$s1 # if \$s3 < \$s1 => \$t2 = 1	\$a2	6	0
<input type="checkbox"/>	0x0040001c	0x06200002	bltz \$17,2	13: bltz \$s1,NEGATIVE #Test if \$s1 and \$s2 is negative?	\$a3	7	0
<input type="checkbox"/>	0x00400020	0x11400003	beq \$10,\$0,3	14: beq \$t2,\$zero,EXIT #if t2=0=> s3>s1 (s1 and \$s2 are positive)	\$t0	8	0
<input type="checkbox"/>	0x00400024	0x0810000b	j 0x0040002c	16: j OVERFLOW	\$t1	9	3
<input type="checkbox"/>	0x00400028	0x15400001	bne \$10,\$0,1	18: bne \$t2,\$zero,EXIT #if t2!=0 => \$s3<s1 (s1 and \$s2 are negative)	\$t2	10	0
<input type="checkbox"/>	0x0040002c	0x24080001	addiu \$8,\$0,1	21: li \$t0,1 #the result is overflow	\$t3	11	0
<input type="checkbox"/>					\$t4	12	0
<input type="checkbox"/>					\$t5	13	0
<input type="checkbox"/>					\$t6	14	0
<input type="checkbox"/>					\$t7	15	0
<input type="checkbox"/>					\$s0	16	0
<input type="checkbox"/>					\$s1	17	1
<input type="checkbox"/>					\$s2	18	2
<input type="checkbox"/>					\$s3	19	3
<input type="checkbox"/>					\$s4	20	0
<input type="checkbox"/>					\$s5	21	0
<input type="checkbox"/>					\$s6	22	0
<input type="checkbox"/>					\$s7	23	0
<input type="checkbox"/>					\$s8	24	0
<input type="checkbox"/>					\$s9	25	0
<input type="checkbox"/>					\$k0	26	0
<input type="checkbox"/>					\$k1	27	0
<input type="checkbox"/>					\$gp	28	268468224
<input type="checkbox"/>					\$sp	29	2147479548
<input type="checkbox"/>					\$fp	30	0
<input type="checkbox"/>					\$ra	31	0
<input type="checkbox"/>					pc		4194324
<input type="checkbox"/>					hi		0
<input type="checkbox"/>					lo		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	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	

⇒ \$t1 =3 => \$s1 và \$s2 cùng dấu

- bltz \$t1,EXIT #If not: t1<0, exit  
⇒ vì \$t1=3 >0 => chạy tiếp lệnh dưới

- slt \$t2,\$s3,\$s1 # if \$s3 < \$s1 => \$t2 = 1

Text Segment							
Bkpt	Address	Code	Basic	Source	Name	Number	Value
<input type="checkbox"/>	0x00400000	0x24110001	addiu \$17,\$0,1	4: li \$s1,1	\$zero	0	0
<input type="checkbox"/>	0x00400004	0x24120002	addiu \$18,\$0,2	5: li \$s2,2	\$at	1	0
<input type="checkbox"/>	0x00400008	0x24080000	addiu \$8,\$0,0	7: li \$t0,0 #No Overflow is default status	\$v0	2	0
<input type="checkbox"/>	0x0040000c	0x02329821	addu \$19,\$17,\$18	8: addu \$s3,\$s1,\$s2 # s3 = s1 + s2	\$a0	4	0
<input type="checkbox"/>	0x00400010	0x02324826	xor \$9,\$17,\$18	9: xor \$t1,\$s1,\$s2 #test if \$s1 and \$s2 have the same sign	\$a1	5	0
<input type="checkbox"/>	0x00400014	0x02000006	bltz \$9,6	10: bltz \$t1,EXIT #if not: t1<0, exit	\$a2	6	0
<input type="checkbox"/>	0x00400018	0x0271502a	slt \$10,\$19,\$17	12: slt \$t2,\$s3,\$s1 # if \$s3 < \$s1 => \$t2 = 1	\$a3	7	0
<input type="checkbox"/>	0x0040001c	0x06200002	bltz \$17,2	13: bltz \$s1,NEGATIVE #Test if \$s1 and \$s2 is negative?	\$t0	8	0
<input type="checkbox"/>	0x00400020	0x11400003	beq \$10,\$0,3	14: beq \$t2,\$zero,EXIT #if t2=0=> s3>s1 (s1 and \$s2 are positive)	\$t1	9	3
<input type="checkbox"/>	0x00400024	0x0810000b	j 0x0040002c	16: j OVERFLOW	\$t2	10	0
<input type="checkbox"/>	0x00400028	0x15400001	bne \$10,\$0,1	18: bne \$t2,\$zero,EXIT #if t2!=0 => \$s3<s1 (s1 and \$s2 are negative)	\$t3	11	0
<input type="checkbox"/>	0x0040002c	0x24080001	addiu \$8,\$0,1	21: li \$t0,1 #the result is overflow	\$t4	12	0
<input type="checkbox"/>					\$t5	13	0
<input type="checkbox"/>					\$t6	14	0
<input type="checkbox"/>					\$t7	15	0
<input type="checkbox"/>					\$s0	16	0
<input type="checkbox"/>					\$s1	17	1
<input type="checkbox"/>					\$s2	18	2
<input type="checkbox"/>					\$s3	19	3
<input type="checkbox"/>					\$s4	20	0
<input type="checkbox"/>					\$s5	21	0
<input type="checkbox"/>					\$s6	22	0
<input type="checkbox"/>					\$s7	23	0
<input type="checkbox"/>					\$s8	24	0
<input type="checkbox"/>					\$s9	25	0
<input type="checkbox"/>					\$k0	26	0
<input type="checkbox"/>					\$k1	27	0
<input type="checkbox"/>					\$gp	28	268468224
<input type="checkbox"/>					\$sp	29	2147479548
<input type="checkbox"/>					\$fp	30	0
<input type="checkbox"/>					\$ra	31	0
<input type="checkbox"/>					pc		4194324
<input type="checkbox"/>					hi		0
<input type="checkbox"/>					lo		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	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	

⇒ Vì \$s3>\$s1 => \$t2 =0

- bltz \$s1,NEGATIVE #Test if \$s1 and \$s2 is negative?  
⇒ Vì \$s1 =1 không <0 => tiếp tục chạy lệnh bên dưới

- beq \$t2,\$zero,EXIT #if  $t2=0 \Rightarrow s3>s1$  ( $s1$  and  $s2$  are positive)

Text Segment						Name	Number	Value
Bkpt	Address	Code	Basic	Source				
<input type="checkbox"/>	0x00400000	0x24110001	addiu \$17,\$0,1	4: li \$s1,1		\$zero	0	0
<input type="checkbox"/>	0x00400004	0x24120002	addiu \$18,\$0,2	5: li \$s2,2		\$at	1	0
<input type="checkbox"/>	0x00400008	0x24080000	addiu \$8,\$0,0	7: li \$t0,0 #No Overflow is default status		\$v0	2	0
<input type="checkbox"/>	0x0040000c	0x2329821	addu \$19,\$17,\$18	8: addu \$s3,\$s1,\$s2 # s3 = s1 + s2		\$v1	3	0
<input type="checkbox"/>	0x00400010	0x02324826	xor \$9,\$17,\$18	9: xor \$t1,\$s1,\$s2 #Test if \$s1 and \$s2 have the same sign		\$a0	4	0
<input type="checkbox"/>	0x00400014	0x05200006	bltz \$9,6	10: bltz \$t1,EXIT #if not: t1<0, exit		\$a1	5	0
<input type="checkbox"/>	0x00400018	0x0271502a	sllt \$10,\$19,\$17	12: sllt \$t2,\$s3,\$s1 # if \$s3 < \$s1 => \$t2 = 1		\$a2	6	0
<input type="checkbox"/>	0x0040001c	0x06200002	bltz \$17,2	13: bltz \$s1,NEGATIVE #Test if \$s1 and \$s2 is negative?		\$a3	7	0
<input type="checkbox"/>	0x00400020	0x11400003	beq \$10,\$0,3	14: beq \$t2,\$zero,EXIT #if t2==> s3=s1 (\$s1 and \$s2 are positive)		\$t0	8	0
<input type="checkbox"/>	0x00400024	0x0810000b	j 0x0040002c	16: j OVERFLOW		\$t1	9	3
<input type="checkbox"/>	0x00400028	0x15400001	bne \$10,\$0,1	18: bne \$t2,\$zero,EXIT #if t2!=0 => \$s3<\$s1 (\$s1 and \$s2 are negative)		<b>\$t2</b>	10	0
<input type="checkbox"/>	0x0040002c	0x24080001	addiu \$8,\$0,1	21: li \$t0,1 #the result is overflow!		\$t3	11	0
						\$t4	12	0
						\$t5	13	0
						\$t6	14	0
						\$t7	15	0
						\$a0	16	0
						\$a1	17	1
						\$a2	18	2
						\$a3	19	3
						\$a4	20	0
						\$a5	21	0
						\$a6	22	0
						\$a7	23	0
						\$t8	24	0
						\$t9	25	0
						\$k0	26	0
						\$k1	27	0
						\$gp	28	268468224
						\$fp	29	2147479548
						\$sp	30	0
						\$ra	31	0
						pc		4194352
						hi		0
						lo		0

- ⇒ Vì  $t2 = 0 \Rightarrow$  Nhảy tới EXIT để kết thúc chương trình
- ⇒  $t0 = 0 \Rightarrow$  Không bị tràn bộ nhớ

- **TH2 : Cộng 2 số dương tràn bộ nhớ**
  - Tương tự như chương trình cộng 2 số dương nhưng tới lệnh:
    - `addu $s3,$s1,$s2`      # \$s1=0x7fffffff, \$s2
    - ⇒  $s3 = s1 + s2 = -2$  ( overflow )

EditExecute

Text Segment

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x3c017fff	lui \$1,32767	4: li \$a1,0x7FFFFFFF
<input type="checkbox"/>	0x00400004	0x3431ffff	ori \$17,\$1,65535	
<input type="checkbox"/>	0x00400008	0x3c017fff	lui \$1,32767	5: li \$a2,0x7FFFFFFF
<input type="checkbox"/>	0x0040000c	0x3432ffff	ori \$18,\$1,65535	
<input type="checkbox"/>	0x00400010	0x24080000	addiu \$8,\$0,0	7: li \$t0,0 #No Overflow is default status
<input type="checkbox"/>	0x00400014	0x02329821	addu \$19,\$17,\$18	8: addu \$a3,\$a1,\$a2 # a3 = a1 + a2
<input type="checkbox"/>	0x00400018	0x02324826	xor \$9,\$17,\$18	9: xor \$t1,\$a1,\$a2 #test if \$a1 and \$a2 have the same sign
<input type="checkbox"/>	0x0040001c	0x05200006	bltz \$9,\$6	10: bltz \$t1,EXIT #if not: t1<0, exit
<input type="checkbox"/>	0x00400020	0x0271502a	sle \$10,\$19,\$17	12: sle \$t2,\$a3,\$a1 # if \$a3 < \$a1 => \$a2 = 1
<input type="checkbox"/>	0x00400024	0x0a620000	bltz \$17,2	13: bltz \$a1,NOARRIVE #test if \$a1 and \$a2 is negative?
<input type="checkbox"/>	0x00400028	0x11400003	beg \$10,\$0,\$3	14: beg \$t2,\$zero,EXIT #if t2==> a3>a1 (\$a1 and \$a2 are positive)
<input type="checkbox"/>	0x0040002c	0x0810000d	j 0x00400034	16: j OVERFLOW

Data Segment

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

←

→

0x10010000 (.data)

☒ Hexadecimal Addresses☐ Hexadecimal Values☐ ASCII

RegistersCoproc 1Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	2147418112
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	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	2147483647
\$s2	18	2147483647
\$a3	19	-3
\$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	260468224
\$ap	29	2147479548
\$fp	30	0
\$ra	31	0
\$f		4194328
\$hi		0
\$lo		0

Mars MessagesRun I/O

- slt \$t2,\$s3,\$s1

**Text Segment**

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c017fff	lui \$1,32767	4: li \$s1,0x7fffffff
	0x00400004	0x3431ffff	ori \$17,\$1,65535	
	0x00400008	0x3c017fff	lui \$1,32767	5: li \$s2,0x7fffffff
	0x0040000c	0x3432ffff	ori \$18,\$1,65535	
	0x00400010	0x24080000	addiu \$8,\$0,0	7: li \$t0,0 #No Overflow is default status
	0x00400014	0x02329821	addu \$19,\$17,\$18	8: addu \$s3,\$s1,\$s2 # s3 = s1 + s2
	0x00400018	0x02324826	xor \$9,\$17,\$18	9: xor \$t1,\$s1,\$s2 #test if \$s1 and \$s2 have the same sign
	0x0040001c	0x05200006	bltz \$9,\$6	10: bltz \$t1,EXIT #if not: t1<0, exit
	0x00400020	0x0271502a	slt \$10,\$19,\$17	12: slt \$t2,\$s3,\$s1 # if \$s3 < \$s1 => \$t2 = 1
	0x00400024	0x06200002	bltz \$17,\$2	13: bltz \$s1,NEGATIVE #test if \$s1 and \$s2 is negative?
	0x00400028	0x11400003	beq \$10,\$0,\$3	14: beq \$t2,\$zero,EXIT #if t2=0=> s3>s1 (s1 and \$s2 are positive)
	0x0040002c	0x0810000d	j 0x00400034	16: j OVERFLOW

**Data Segment**

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

Name	Number	Value
\$zero	0	0
\$at	1	2147418112
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	1
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$a0	16	0
\$a1	17	2147483647
\$a2	18	2147483647
\$a3	19	-2
\$a4	20	0
\$a5	21	0
\$a6	22	0
\$a7	23	0
\$a8	24	0
\$a9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
\$f		4194348
hi		0
lo		0

⇒ Vì \$s3 < \$s1( -2 < 0x7fffffff) => \$t2 = 1

- beq \$t2,\$zero,EXIT

**Text Segment**

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c017fff	lui \$1,32767	4: li \$s1,0x7fffffff
	0x00400004	0x3431ffff	ori \$17,\$1,65535	
	0x00400008	0x3c017fff	lui \$1,32767	5: li \$s2,0x7fffffff
	0x0040000c	0x3432ffff	ori \$18,\$1,65535	
	0x00400010	0x24080000	addiu \$8,\$0,0	7: li \$t0,0 #No Overflow is default status
	0x00400014	0x02329821	addu \$19,\$17,\$18	8: addu \$s3,\$s1,\$s2 # s3 = s1 + s2
	0x00400018	0x02324826	xor \$9,\$17,\$18	9: xor \$t1,\$s1,\$s2 #test if \$s1 and \$s2 have the same sign
	0x0040001c	0x05200006	bltz \$9,\$6	10: bltz \$t1,EXIT #if not: t1<0, exit
	0x00400020	0x0271502a	slt \$10,\$19,\$17	12: slt \$t2,\$s3,\$s1 # if \$s3 < \$s1 => \$t2 = 1
	0x00400024	0x06200002	bltz \$17,\$2	13: bltz \$s1,NEGATIVE #test if \$s1 and \$s2 is negative?
	0x00400028	0x11400003	beq \$10,\$0,\$3	14: beq \$t2,\$zero,EXIT #if t2=0=> s3>s1 (s1 and \$s2 are positive)
	0x0040002c	0x0810000d	j 0x00400034	16: j OVERFLOW

**Data Segment**

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

Name	Number	Value
\$zero	0	0
\$at	1	2147418112
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	1
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$a0	16	0
\$a1	17	2147483647
\$a2	18	2147483647
\$a3	19	-2
\$a4	20	0
\$a5	21	0
\$a6	22	0
\$a7	23	0
\$a8	24	0
\$a9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
\$f		4194348
hi		0
lo		0

⇒ Vì \$t2 = 1 != 0 => chương trình sẽ jump tới OVERFLOW

- OVERFLOW:  
li \$t0,1 #the result is overflow is 1

Text Segment							
Bkpt	Address	Code	Basic	Source	Name	Number	Value
<input type="checkbox"/>	0x00400008	0x3e017fff	lui \$1,32767	5: li \$a2,0x7FFFFFFF	\$zero	0	0
<input type="checkbox"/>	0x0040000c	0x3432ffff	ori \$18,\$1,65535		\$at	1	2147418112
<input type="checkbox"/>	0x00400010	0x24080000	addiu \$8,\$0,0	7: li \$t0,0 #No Overflow is default status	\$v0	2	0
<input type="checkbox"/>	0x00400014	0x02329821	addu \$19,\$17,\$18	8: addu \$a3,\$a1,\$a2 # a3 = a1 + a2	\$v1	3	0
<input type="checkbox"/>	0x00400018	0x02324826	xor \$9,\$17,\$18	9: xor \$t1,\$a1,\$a2 #test if \$a1 and \$a2 have the same sign	\$a0	4	0
<input type="checkbox"/>	0x0040001c	0x05200006	bltz \$9,\$6	10: bltz \$t1,EXIT #if not: t1<0, exit	\$a1	5	0
<input type="checkbox"/>	0x00400020	0x0271502a	sllt \$10,\$19,\$17	12: sllt \$t2,\$a3,\$a1 # if \$a3 < \$a1 => \$t2 = 1	\$a2	6	0
<input type="checkbox"/>	0x00400024	0x06200002	bltz \$17,\$2	13: bltz \$a1,NEGATIVE #test if \$a1 and \$a2 is negative?	\$a3	7	0
<input type="checkbox"/>	0x00400028	0x11400003	beq \$10,\$0,3	14: beq \$t2,\$zero,EXIT #if t2=0=> a3>a1 (a1 and \$a2 are positive)	\$t0	8	0
<input type="checkbox"/>	0x0040002c	0x0810000d	j 0x00400034	16: j OVERFLOW	\$t1	9	0
<input type="checkbox"/>	0x00400030	0x15400001	hne \$10,\$0,1	18: hne \$t2,\$zero,EXIT #if t2!=0 => \$a3<a1 (a1 and \$a2 are negative)	\$t2	10	1
<input type="checkbox"/>	0x00400034	0x24080001	addiu \$8,\$0,1	21: li \$t0,1 #the result is overflow	\$t3	11	0
					\$t4	12	0
					\$t5	13	0
					\$t6	14	0
					\$t7	15	0
					\$a0	16	0
					\$a1	17	2147483647
					\$a2	18	2147483647
					\$a3	19	-2
					\$a4	20	0
					\$a5	21	0
					\$a6	22	0
					\$a7	23	0
					\$a8	24	0
					\$a9	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

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

⇒ Gán giá trị 1 cho biến \$t0=> Chương trình bị tràn bộ nhớ

- TH3 : Cộng 2 số khác dấu
- Tương tự chương trình cộng 2 số dương cho đến lệnh:
- xor \$t1,\$s1,\$s2

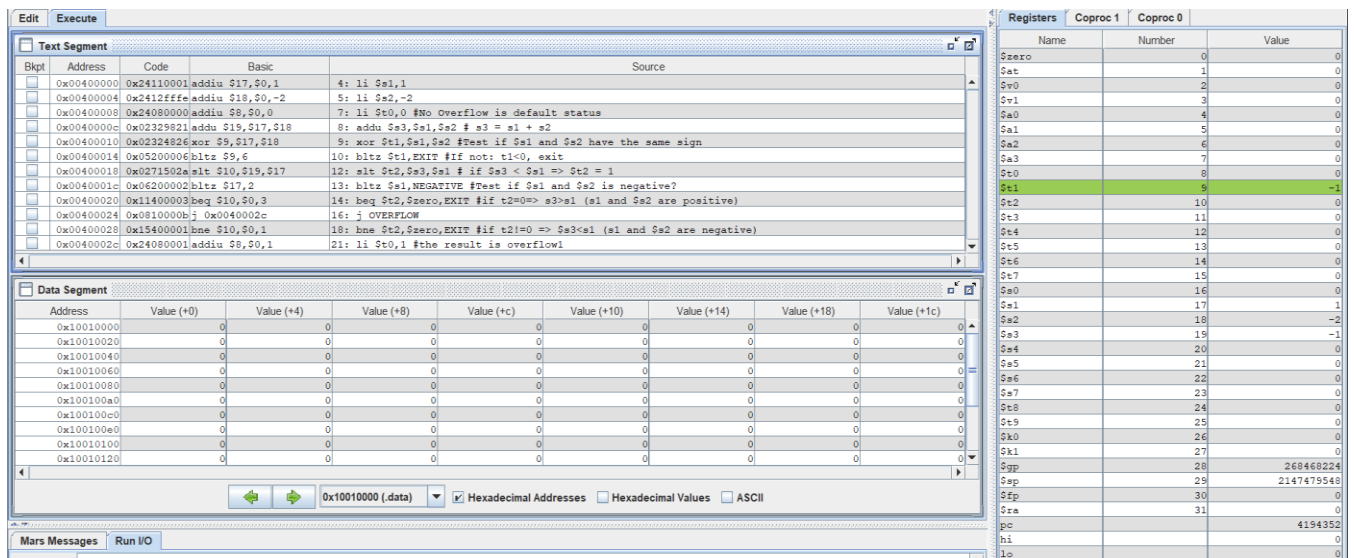
Text Segment							
Bkpt	Address	Code	Basic	Source	Name	Number	Value
<input type="checkbox"/>	0x00400008	0x24110001	addiu \$17,\$0,1	4: li \$a1,1	\$zero	0	0
<input type="checkbox"/>	0x0040000c	0x2411ffff	addiu \$19,\$0,-2	5: li \$a2,-2	\$at	1	0
<input type="checkbox"/>	0x00400010	0x24080000	addiu \$8,\$0,0	7: li \$t0,0 #No Overflow is default status	\$v0	2	0
<input type="checkbox"/>	0x00400014	0x02329821	addu \$19,\$17,\$18	8: addu \$a3,\$a1,\$a2 # a3 = a1 + a2	\$v1	3	0
<input type="checkbox"/>	0x00400018	0x02324826	xor \$9,\$17,\$18	9: xor \$t1,\$a1,\$a2 #test if \$a1 and \$a2 have the same sign	\$a0	4	0
<input type="checkbox"/>	0x0040001c	0x05200006	bltz \$9,\$6	10: bltz \$t1,EXIT #if not: t1<0, exit	\$a1	5	0
<input type="checkbox"/>	0x00400020	0x0271502a	sllt \$10,\$19,\$17	12: sllt \$t2,\$a3,\$a1 # if \$a3 < \$a1 => \$t2 = 1	\$a2	6	0
<input type="checkbox"/>	0x00400024	0x06200002	bltz \$17,\$2	13: bltz \$a1,NEGATIVE #test if \$a1 and \$a2 is negative?	\$a3	7	0
<input type="checkbox"/>	0x00400028	0x0810000d	j 0x00400034	14: beq \$t2,\$zero,EXIT #if t2=0=> a3>a1 (a1 and \$a2 are positive)	\$t0	8	0
<input type="checkbox"/>	0x0040002c	0x15400001	hne \$10,\$0,1	18: hne \$t2,\$zero,EXIT #if t2!=0 => \$a3<a1 (a1 and \$a2 are negative)	\$t1	9	-1
<input type="checkbox"/>	0x00400034	0x24080001	addiu \$8,\$0,1	21: li \$t0,1 #the result is overflow	\$t2	10	0
					\$t3	11	0
					\$t4	12	0
					\$t5	13	0
					\$t6	14	0
					\$t7	15	0
					\$a0	16	0
					\$a1	17	1
					\$a2	18	-2
					\$a3	19	-1
					\$a4	20	0
					\$a5	21	0
					\$a6	22	0
					\$a7	23	0
					\$a8	24	0
					\$a9	25	0
					\$k0	26	0
					\$k1	27	0
					\$gp	28	268468224
					\$sp	29	2147479548
					\$fp	30	0
					\$ra	31	0
					pc		4194324
					hi		0
					lo		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	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	

⇒ \$t1 = -1 ( \$s1 = 1 , \$s2 = -2)

- bltz \$t1,EXIT      #If not: t1<0, exit



- ⇒ Vì trong TH này \$t1 = <0 => Nhảy tới lệnh EXIT
- ⇒ Vì \$t0 = 0 => Chương trình không bị tràn bộ nhớ

- TH cộng 2 số âm không bị tràn bộ nhớ và bị tràn bộ nhớ sẽ lọt vào NEGATIVE, kết quả tương tự như TH1 và TH2

## Assignment 2

Kết quả được chứa ở \$s1:

- Lấy ra MSB của \$s0

.text

li \$s0, 0x312ACD81 # Set value to s0

li \$t0, 0xFF000000 # Mask to get MSB

and \$s1, \$s0, \$t0 # Get MSB

- Xóa LSB của \$s0

.text

```
li $s0, 0x312ACD81 # Set value to s0
li $t0, 0xffffffff # Mask to clear LSB
and $s1, $s0, $t0 # Clear LSB
```

- Đặt 8 bit bên phải của \$s0 là 1

```
.text
```

```
li $s0, 0x312ACD81 # Set value to s0
li $t0, 0x000000ff # Mask to Set LSB
or $s1, $s0, $t0 # Set LSB
```

- Xóa \$s0 (\$s0 = 0)

```
.text
```

```
li $s0, 0x312ACD81 # Set value to s0
and $s0, $s0, $zero # Clear $s0
```

### Assignment 3

a.

```
.text
li $s1, 10
bltz $s1, NEGSET # Branch to set $s0 = -$s1 if s1 < 0
add $s0, $s1, $zero # s0 = s1 if s1 > 0
j EXIT
NEGSET:
li $t0, 0xffffffff # Load Mask to t0
xor $s3, $s1, $t0 # Set s0 = -s1
addi $s0, $s3, 1
EXIT:
```

b.



```
.text
add $s0, $s1, $zero # s0 = s1
```

c.

```
.text
li $s2, 0xffffffff
xor $s0, $s1, $s2 # s0 = not(s1)
```

d.

```
.text
slt $t1, $s1, $s2
bne $t1, $zero, label
beq $s1, $s2, label
```

## Assignment 4

```
.text
li $s1, 0x7fffffff
li $s2, 0x7fffffff
li $t0, 0 # Default status is no overflow
xor $t1, $s1, $s2 # Check s1 and s2 have the same sign
bltz $t1, EXIT # Exit if $t1 < 0 <=> different sign

# Else s1 and s2 have same sign
addu $s3, $s1, $s2 # s3 = s1 + s2
xor $t1, $s3, $s1 # Check sum and s1 have the same sign
bltz $t1, OVERFLOW # t1 < 0 <=> sum and s1 does not have
the same sign => overflow
j EXIT
OVERFLOW:
```

```
li $t0, 1 # Overflow
```

```
EXIT:
```

## **Assignment 5**

```
.text
```

```
li $s0, 1 # set first result s0 = 1
```

```
li $s1, 8 # set second factor s1 = 8 (small power of X2)
```

```
li $s2, 1 # value to determine stop time of loop
```

```
LOOP:
```

```
beq $s1, $s2, END # End if s1 = 1
```

```
sll $s0, $s0, 1 # s0 = s0 * 2
```

```
srl $s1, $s1, 1 # s1 = s1/2
```

```
j LOOP
```

```
END:
```