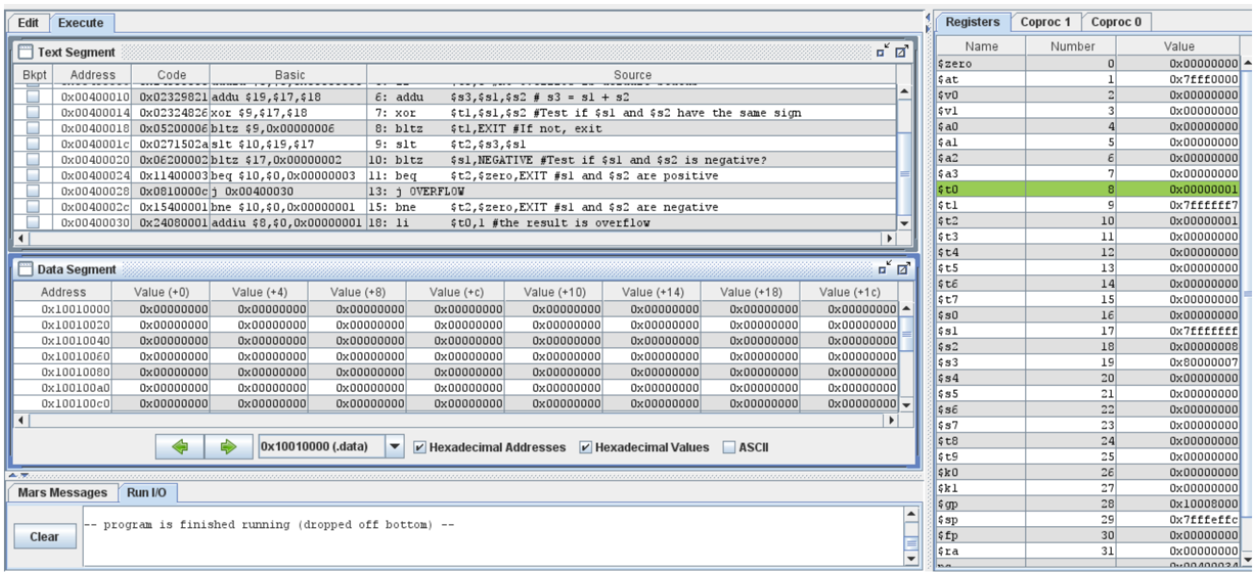
Phạm Duy Hưng – 20225850

Assignment 1:

```
.text
start:
li  $s1,0x7FFFFFFF
li  $s2,8
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, exit
slt     $t2,$s3,$s1
bltz    $s1,NEGATIVE #Test if $s1 and $s2 is negative?
beq     $t2,$zero,EXIT #s1 and $s2 are positive
    # if $s3 > $s1 then the result is not overflow
j OVERFLOW
NEGATIVE:
bne     $t2,$zero,EXIT #s1 and $s2 are negative
 # if $s3 < $s1 then the result is not overflow
OVERFLOW:
li  $t0,1 #the result is overflow
EXIT:
```



Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400010	0x02329821	addu $19,$17,$18	6: addu $s3,$s1,$s2 # s3 = s1 + s2
☐	0x00400014	0x02324826	xor $9,$17,$18	7: xor $t1,$s1,$s2 #Test if $s1 and $s2 have the same sign
☐	0x00400018	0x05200006	bltz $9,0x00000006	8: bltz $t1,EXIT #If not, exit
☐	0x0040001c	0x0271502a	slt $10,$19,$17	9: slt $t2,$s3,$s1
☐	0x00400020	0x06200002	bltz $17,0x00000002	10: bltz $s1,NEGATIVE #Test if $s1 and $s2 is negative?
☐	0x00400024	0x11400003	beq $10,$0,0x00000003	11: beq $t2,$zero,EXIT #s1 and $s2 are positive
☐	0x00400028	0x0810000c	j 0x00400030	13: j OVERFLOW
☐	0x0040002c	0x15400001	bne $10,$0,0x00000001	15: bne $t2,$zero,EXIT #s1 and $s2 are negative
☐	0x00400030	0x24080001	addiu $8,$0,0x00000001	18: li $t0,1 #the result is overflow

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

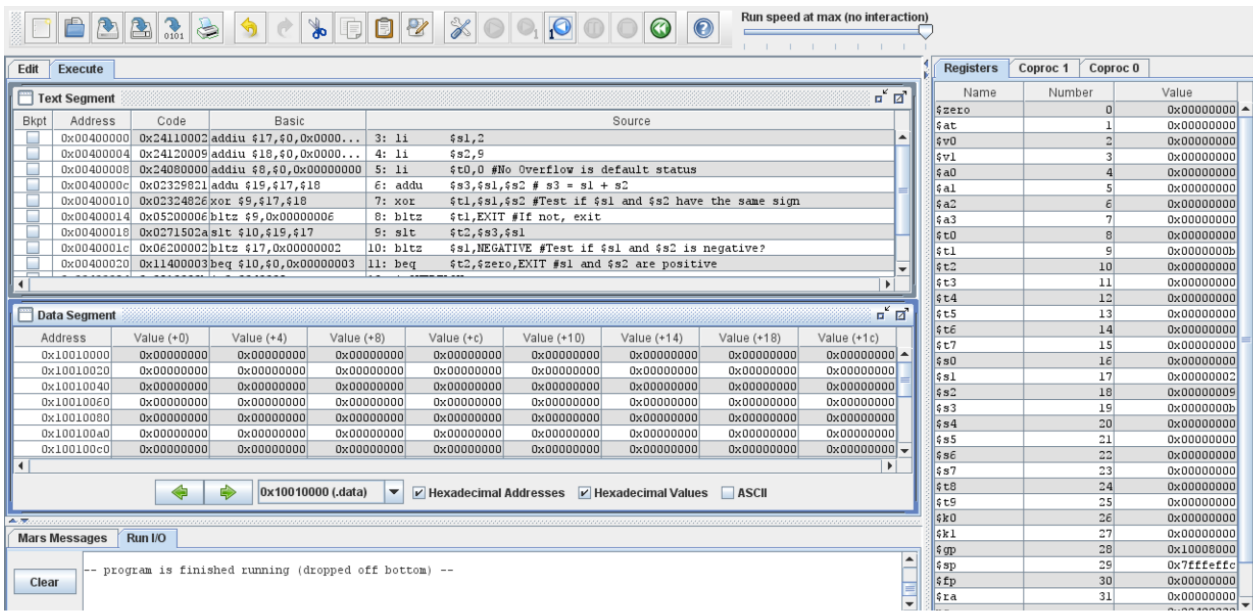
0x10010000 (.data) ✓ Hexadecimal Addresses ✓ Hexadecimal Values ☐ ASCII

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

Name	Number	Value
$zero	0	0x00000000
$at	1	0x7fff0000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000001
$t1	9	0x7ffffff7
$t2	10	0x00000001
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x7fffffff
$s2	18	0x00000008
$s3	19	0x80000007
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000

- Trường hợp 1: Xảy ra overflow với s1 = 7FFFFFFF , s2 = 8 => s3 = 80000007

```
.text
start:
li  $s1,2
li  $s2,9
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, exit
slt     $t2,$s3,$s1
bltz    $s1,NEGATIVE #Test if $s1 and $s2 is negative?
beq     $t2,$zero,EXIT #s1 and $s2 are positive
    # if $s3 > $s1 then the result is not overflow
j OVERFLOW
NEGATIVE:
bne     $t2,$zero,EXIT #s1 and $s2 are negative
 # if $s3 < $s1 then the result is not overflow
OVERFLOW:
li  $t0,1 #the result is overflow
EXIT:
```



Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400000	0x24110002	addiu $17,$0,0x0000...	3: li $s1,2
☐	0x00400004	0x24120009	addiu $18,$0,0x0000...	4: li $s2,9
☐	0x00400008	0x24080000	addiu $8,$0,0x00000000	5: li $t0,0 #No Overflow is default status
☐	0x0040000c	0x02329821	addu $19,$17,$18	6: addu $s3,$s1,$s2 # s3 = s1 + s2
☐	0x00400010	0x02324826	xor $9,$17,$18	7: xor $t1,$s1,$s2 #Test if $s1 and $s2 have the same sign
☐	0x00400014	0x05200006	bltz $9,0x00000006	8: bltz $t1,EXIT #If not, exit
☐	0x00400018	0x0271502a	slt $10,$19,$17	9: slt $t2,$s3,$s1
☐	0x0040001c	0x06200002	bltz $17,0x00000002	10: bltz $s1,NEGATIVE #Test if $s1 and $s2 is negative?
☐	0x00400020	0x11400003	beq $10,$0,0x00000003	11: beq $t2,$zero,EXIT #s1 and $s2 are positive

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

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

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	0x00000000
$t1	9	0x0000000b
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000002
$s2	18	0x00000009
$s3	19	0x0000000b
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000

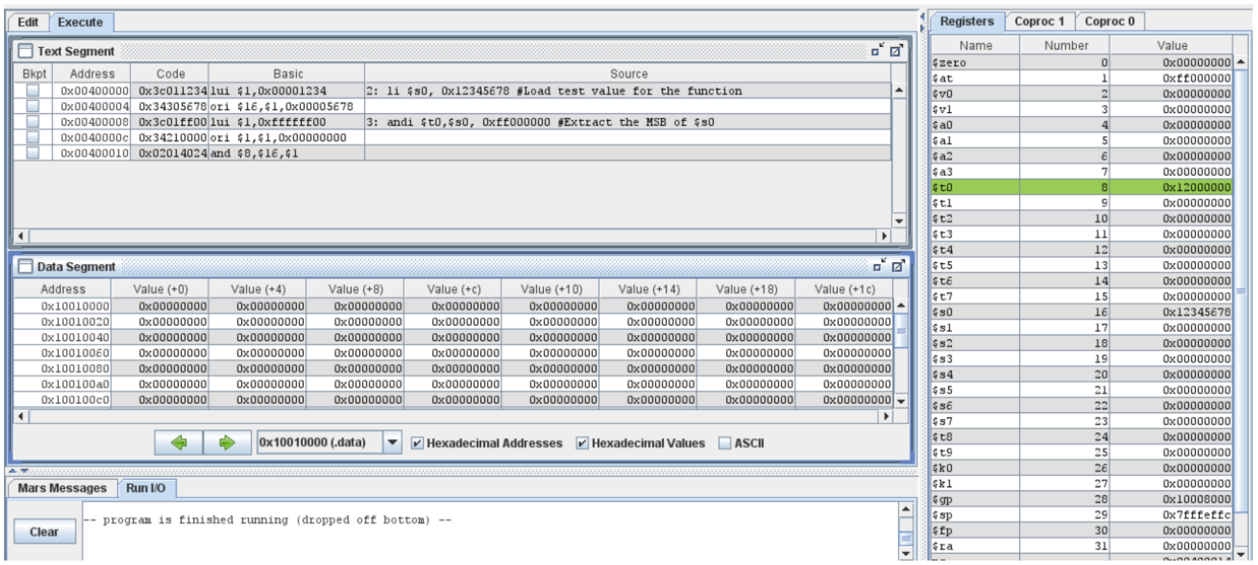
```
-- program is finished running (dropped off bottom) --
```

- Trường hợp 2: Không xảy ra overflow với s1 = 2, s2 = 9 => s3 = 11

Assignment 2:

- Extract MSB of $s0

```
.text
li $s0, 0x12345678 #Load test value for the function
andi $t0,$s0, 0xff000000 #Extract the MSB of $s0
```



Edit | Execute

Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400000	0x3c011234	lui $1,0x00001234	2: li $s0, 0x12345678 #Load test value for the function
☐	0x00400004	0x34305678	ori $16,$1,0x00005678	
☐	0x00400008	0x3c01ff00	lui $1,0xffffff00	3: andi $t0,$s0, 0xff000000 #Extract the MSB of $s0
☐	0x0040000c	0x34210000	ori $1,$1,0x00000000	
☐	0x00400010	0x02014024	and $8,$16,$1	

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

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

Mars Messages | Run I/O

Clear

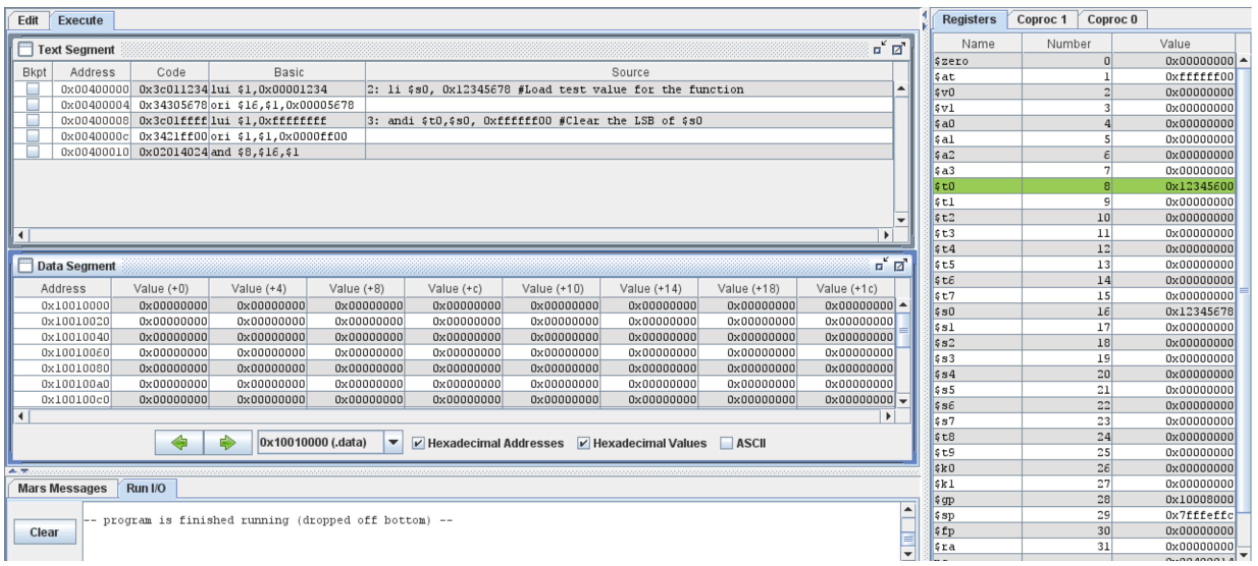
```
-- program is finished running (dropped off bottom) --
```

Registers | Coproc 1 | Coproc 0

Name	Number	Value
$zero	0	0x00000000
$at	1	0xff000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x12000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x12345678
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000

- Clear LSB of $s0

```
.text
li $s0, 0x12345678 #Load test value for the function
andi $t0,$s0, 0xffffff00 #Clear the LSB of $s0
```



Edit | Execute

Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400000	0x3c011234	lui $1,0x00001234	2: li $s0, 0x12345678 #Load test value for the function
☐	0x00400004	0x34305678	ori $16,$1,0x00005678	
☐	0x00400008	0x3c01ffff	lui $1,0xffffffff	3: andi $t0,$s0, 0xffffff00 #Clear the LSB of $s0
☐	0x0040000c	0x3421ff00	ori $1,$1,0x0000ff00	
☐	0x00400010	0x02014024	and $8,$16,$1	

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

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

Mars Messages | Run I/O

Clear

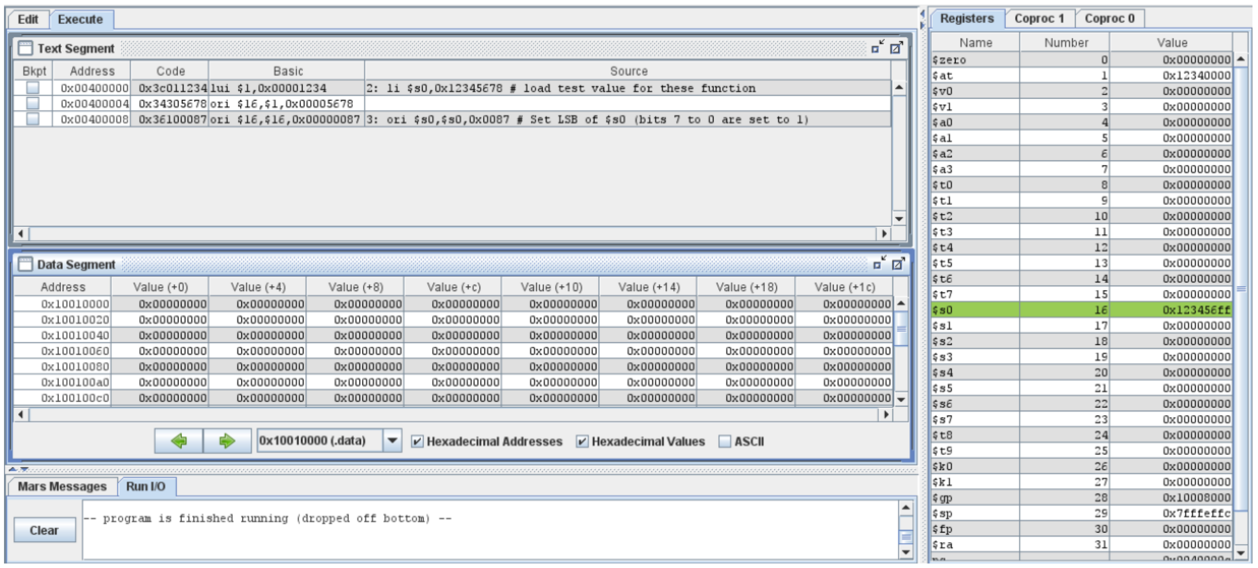
```
-- program is finished running (dropped off bottom) --
```

Registers | Coproc 1 | Coproc 0

Name	Number	Value
$zero	0	0x00000000
$at	1	0xffffff00
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x12345600
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x12345678
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000

- Set LSB of $s0 (bits 7 to 0 are set to 1)

```
.text
li $s0,0x12345678 # load test value for these function
ori $s0,$s0,0x0087 # Set LSB of $s0 (bits 7 to 0 are set to 1)
```



Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400000	0x3c011234	lui $1,0x00001234	2: li $s0,0x12345678 # load test value for these function
☐	0x00400004	0x34305678	ori $16,$1,0x00005678	
☐	0x00400008	0x36100087	ori $16,$16,0x00000087	3: ori $s0,$s0,0x0087 # Set LSB of $s0 (bits 7 to 0 are set to 1)

Data Segment

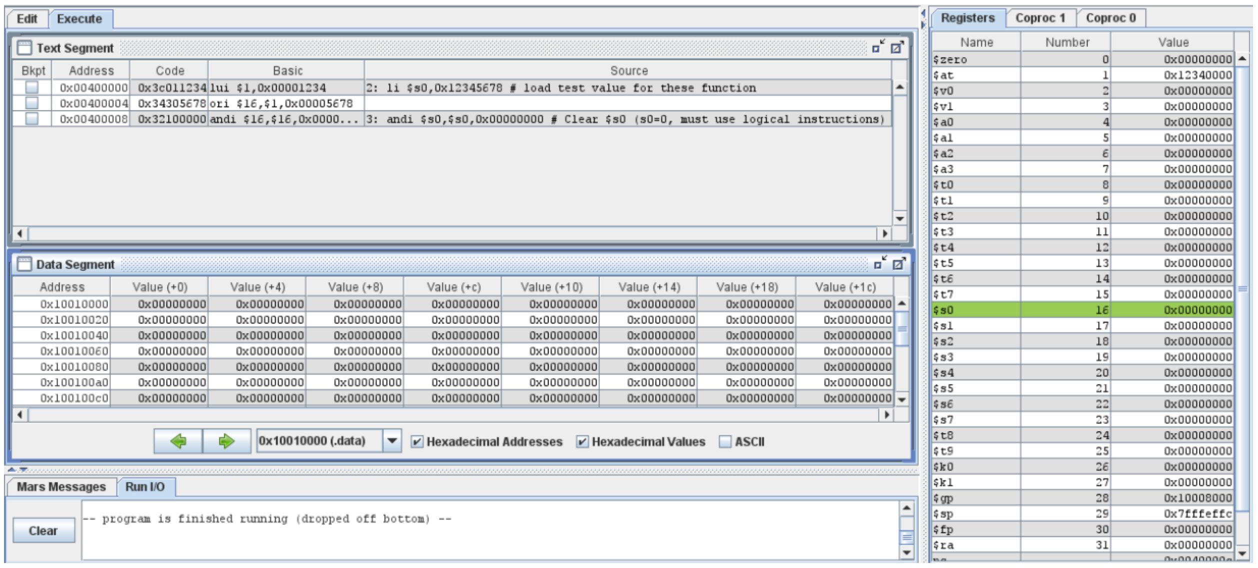
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

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

Name	Number	Value
$zero	0	0x00000000
$at	1	0x12340000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	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
$s0	16	0x123456ff
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000

- Clear $s0 (s0=0, must use logical instructions)

```
.text
li $s0,0x12345678 # load test value for these function
andi $s0,$s0,0x00000000 # Clear $s0 (s0=0, must use logical instructions)
```



Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400000	0x3c011234	lui $1,0x00001234	2: li $s0,0x12345678 # load test value for these function
☐	0x00400004	0x34305678	ori $16,$1,0x00005678	
☐	0x00400008	0x32100000	andi $16,$16,0x0000...	3: andi $s0,$s0,0x00000000 # Clear $s0 (s0=0, must use logical instructions)

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

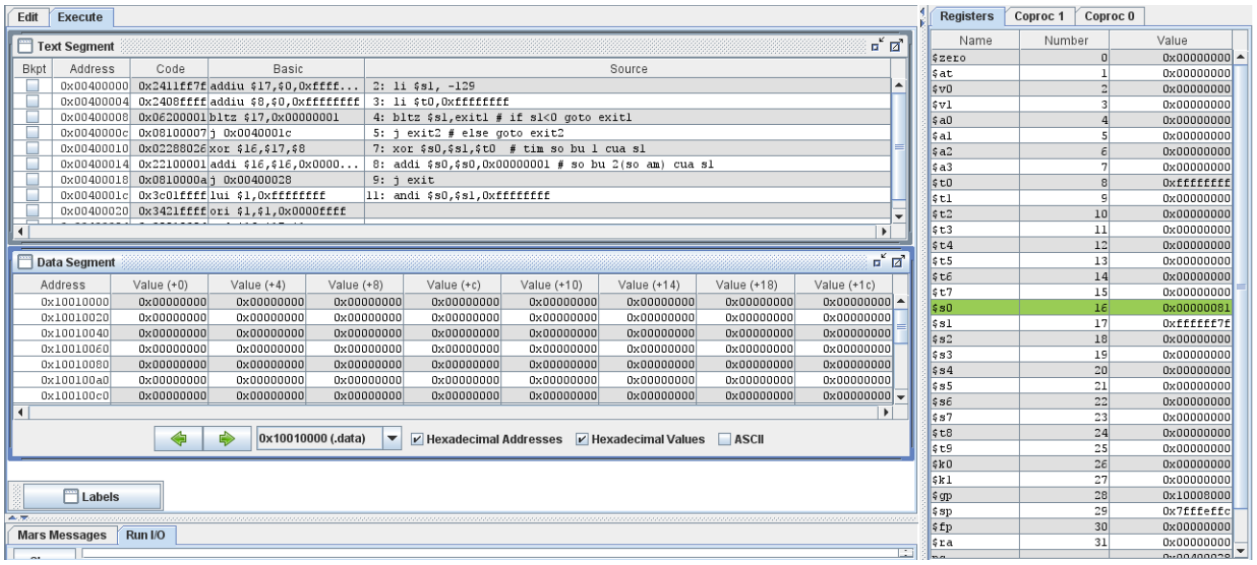
```
-- program is finished running (dropped off bottom) --
```

Name	Number	Value
$zero	0	0x00000000
$at	1	0x12340000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	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
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000

Assignment 3:

a. abs $s0,s1
   s0 <= | $s1 |

```
.text
li $s1, -129
li $t0,0xffffffff
bltz $s1,exit1 # if s1<0 goto exit1
j exit2 # else goto exit2
exit1:
xor $s0,$s1,$t0  # tim so bu 1 cua s1
addi $s0,$s0,0x00000001 # so bu 2(so am) cua s1
j exit
exit2:
andi $s0,$s1,0xffffffff
exit:
```


Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x2411ff7f	addiu $17,$0,0xffff...	2: li $s1, -129
	0x00400004	0x2408ffff	addiu $8,$0,0xffffffff	3: li $t0,0xffffffff
	0x00400008	0x06200001	bltz $17,0x00000001	4: bltz $s1,exit1 # if s1<0 goto exit1
	0x0040000c	0x08100007	j 0x0040001c	5: j exit2 # else goto exit2
	0x00400010	0x02288026	xor $16,$17,$8	7: xor $s0,$s1,$t0 # tim so bu 1 cua s1
	0x00400014	0x22100001	addi $16,$16,0x0000...	8: addi $s0,$s0,0x00000001 # so bu 2(so am) cua s1
	0x00400018	0x0810000a	j 0x00400028	9: j exit
	0x0040001c	0x3c01ffff	lui $1,0xffffffff	11: andi $s0,$s1,0xffffffff
	0x00400020	0x3421ffff	ori $1,$1,0x0000ffff	

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

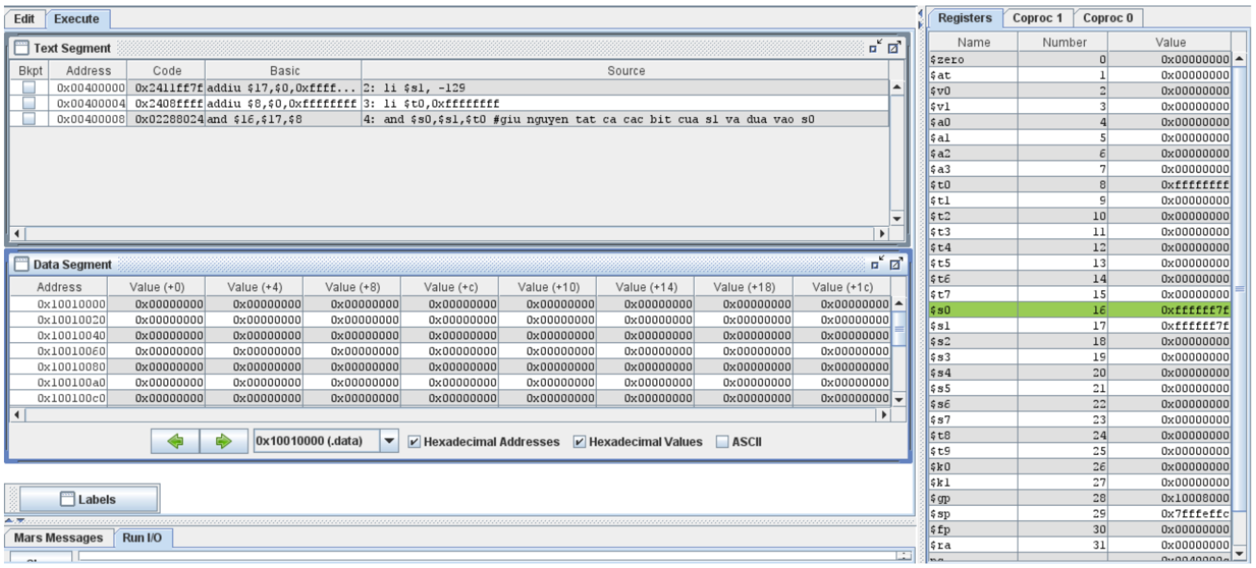
0x10010000 (.data) ✓ Hexadecimal Addresses ✓ Hexadecimal Values ☐ ASCII

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	0xffffffff
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000081
$s1	17	0xffffff7f
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000

b. move $s0,s1

s0 <= $s1

```
.text
li $s1, -129
li $t0,0xffffffff
and $s0,$s1,$t0 #giu nguyen tat ca cac bit cua s1 va dua vao s0
```


Edit | Execute

Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400000	0x2411ff7f	addiu $17,$0,0xffff...	2: li $s1, -129
☐	0x00400004	0x2408ffff	addiu $8,$0,0xffffffff	3: li $t0,0xffffffff
☐	0x00400008	0x02288024	and $16,$17,$8	4: and $s0,$s1,$t0 #giu nguyen tat ca cac bit cua s1 va dua vao s0

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

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

Labels

Mars Messages | Run I/O

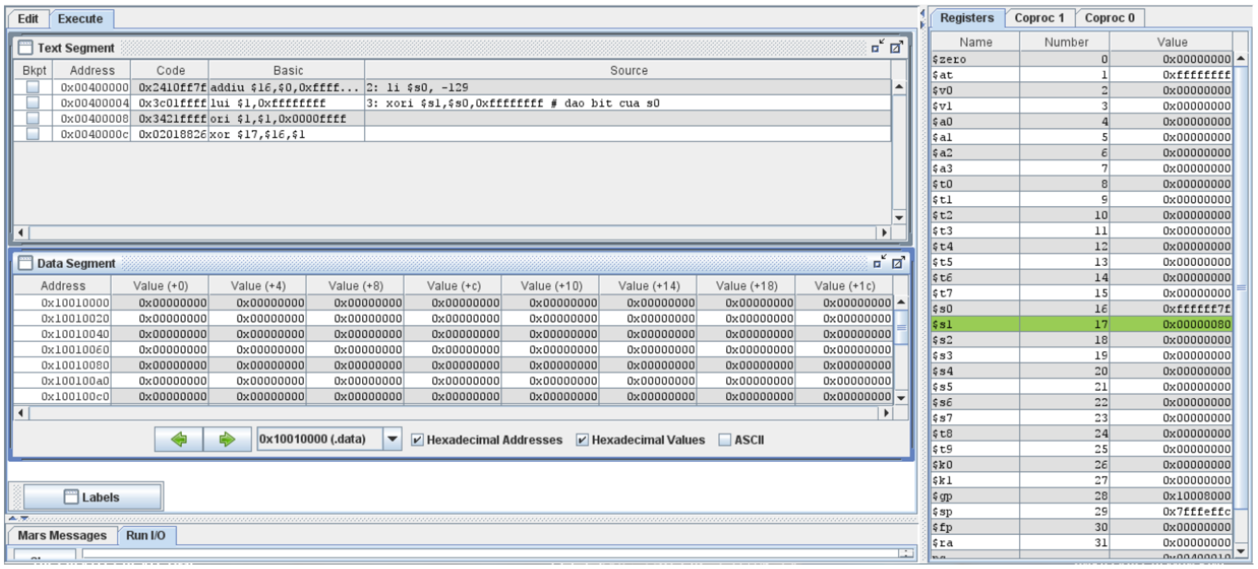
Registers | Coproc 1 | Coproc 0

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0xffffffff
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0xffffff7f
$s1	17	0xffffff7f
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000

c. not $s0

s0 <= bit invert (s0)

```
.text
li $s0, -129
xori $s1,$s0,0xffffffff # dao bit cua s0
```


Edit | Execute

Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400000	0x2410ff7f	addiu $16,$0,0xffff...	2: li $s0, -129
☐	0x00400004	0x3c01ffff	lui $1,0xffffffff	3: xori $s1,$s0,0xffffffff # dao bit cua s0
☐	0x00400008	0x3421ffff	ori $1,$1,0x0000ffff	
☐	0x0040000c	0x02018826	xor $17,$16,$1	

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

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

Labels

Mars Messages | Run I/O

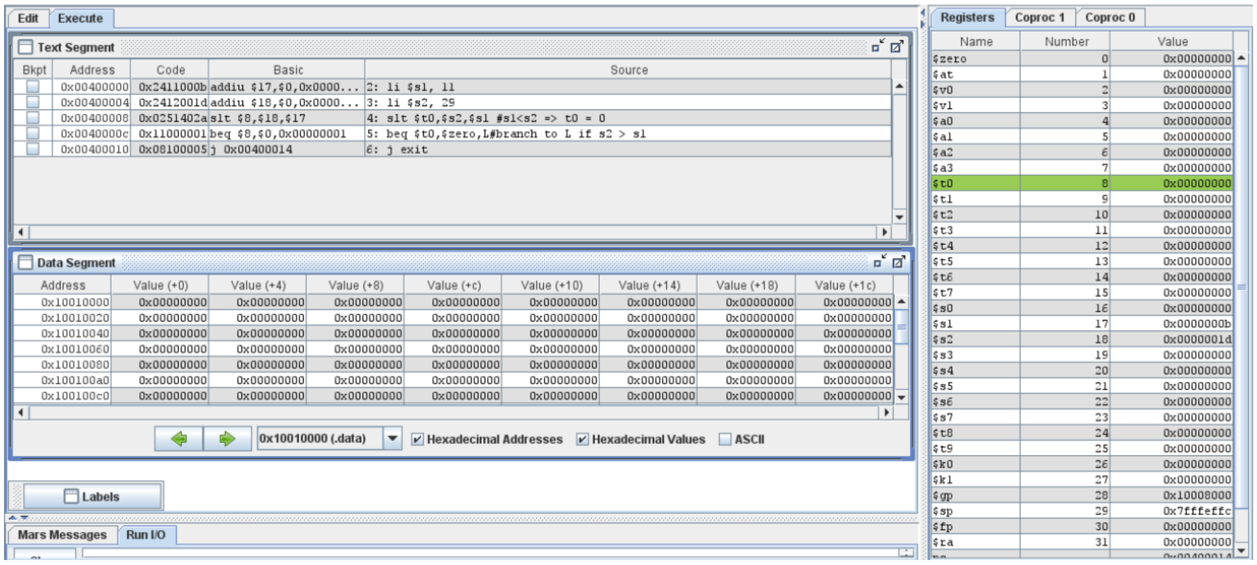
Registers | Coproc 1 | Coproc 0

Name	Number	Value
$zero	0	0x00000000
$at	1	0xffffffff
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	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
$s0	16	0xffffff7f
$s1	17	0x00000080
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000

d. ble $s1,s2,L

if (s1 <= $s2) j L

```
.text
li $s1, 11 #s1 = 11
li $s2, 29 #s2 = 29
slt $t0,$s2,$s1 #s1<s2 => t0 = 0
beq $t0,$zero,L#branch to L if s2 > s1
j exit
L:
exit:
```



Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400000	0x2411000b	addiu $17,$0,0x0000...	2: li $s1, 11
☐	0x00400004	0x2412001d	addiu $18,$0,0x0000...	3: li $s2, 29
☐	0x00400008	0x0251402a	slt $8,$18,$17	4: slt $t0,$s2,$s1 #s1<s2 => t0 = 0
☐	0x0040000c	0x11000001	beq $8,$0,0x00000001	5: beq $t0,$zero,L#branch to L if s2 > s1
☐	0x00400010	0x08100005	j 0x00400014	6: j exit

Data Segment

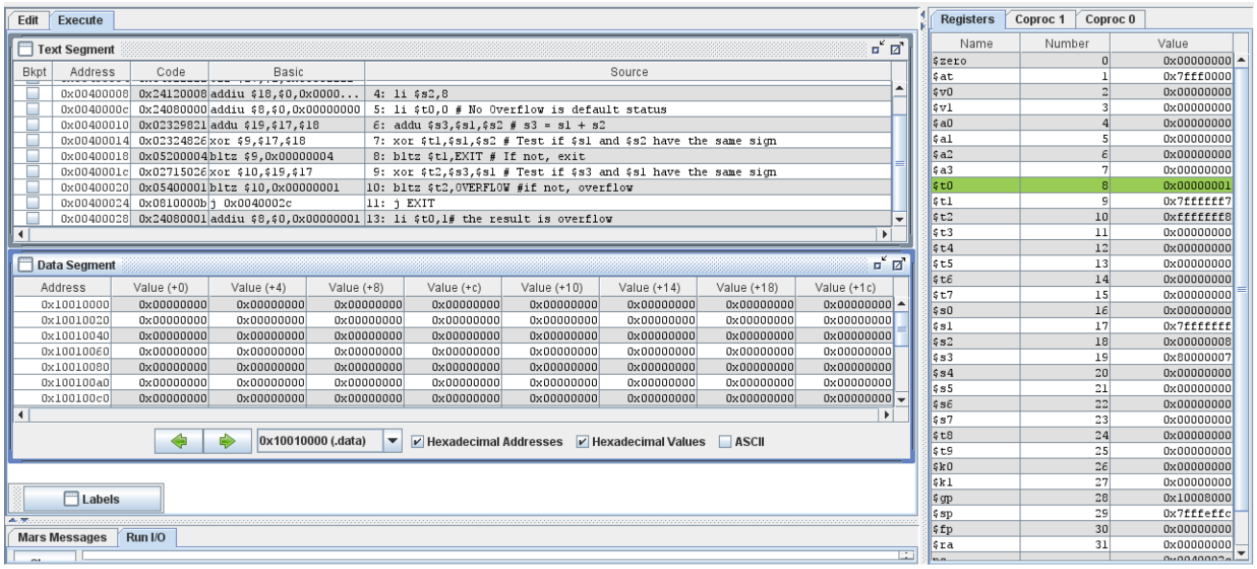
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

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

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	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x0000000b
$s2	18	0x0000001d
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000

Assignment 4:

```
.text
start:
li $s1,0x7FFFFFFF
li $s2,8
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, exit
xor $t2,$s3,$s1 # Test if $s3 and $s1 have the same sign
bltz $t2,OVERFLOW #if not, overflow
j EXIT
OVERFLOW:
li $t0,1# the result is overflow
EXIT:
```



Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400008	0x24120008	addiu $18,$0,0x0000...	4: li $s2,8
☐	0x0040000c	0x24080000	addiu $8,$0,0x00000000	5: li $t0,0 # No Overflow is default status
☐	0x00400010	0x02329821	addu $19,$17,$18	6: addu $s3,$s1,$s2 # s3 = s1 + s2
☐	0x00400014	0x02324826	xor $9,$17,$18	7: xor $t1,$s1,$s2 # Test if $s1 and $s2 have the same sign
☐	0x00400018	0x05200004	bltz $9,0x00000004	8: bltz $t1,EXIT # If not, exit
☐	0x0040001c	0x02715026	xor $10,$19,$17	9: xor $t2,$s3,$s1 # Test if $s3 and $s1 have the same sign
☐	0x00400020	0x05400001	bltz $10,0x00000001	10: bltz $t2,OVERFLOW #if not, overflow
☐	0x00400024	0x0810000b	j 0x0040002c	11: j EXIT
☐	0x00400028	0x24080001	addiu $8,$0,0x00000001	13: li $t0,1# the result is overflow

Data Segment

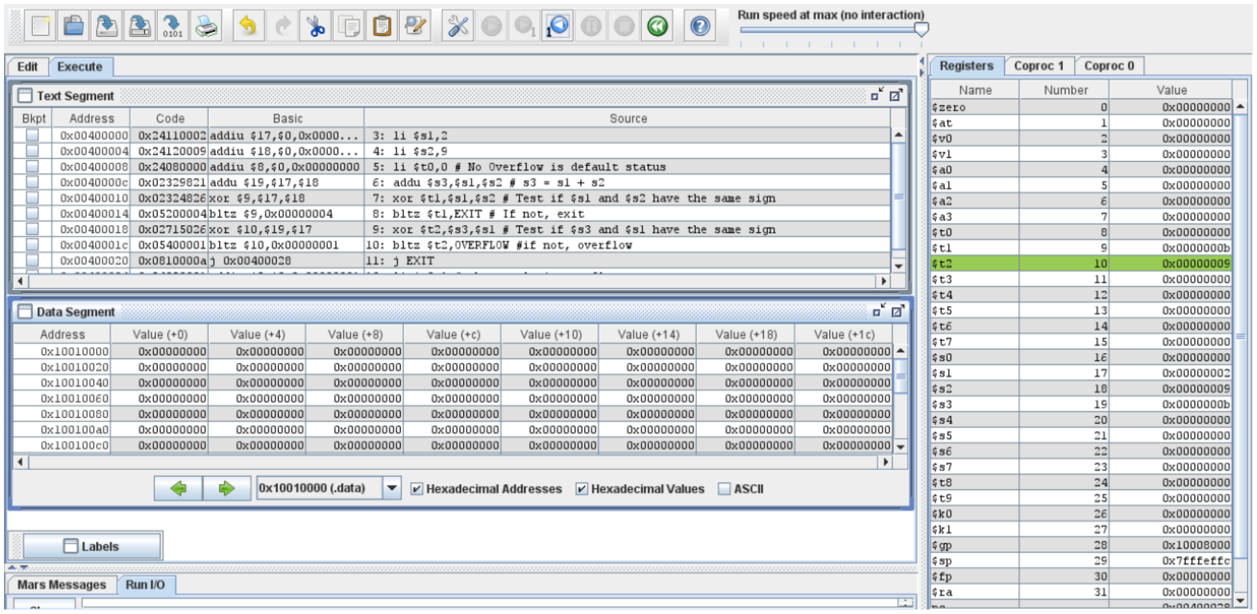
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

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

Name	Number	Value
$zero	0	0x00000000
$at	1	0x7fff0000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000001
$t1	9	0x7ffffff7
$t2	10	0xfffffff8
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x7fffffff
$s2	18	0x00000008
$s3	19	0x80000007
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000

- Trường hợp 1: Xảy ra overflow với s1 = 7FFFFFFF , s2 = 8 => s3 = 80000007

```
.text
start:
li $s1,2
li $s2,9
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, exit
xor $t2,$s3,$s1 # Test if $s3 and $s1 have the same sign
bltz $t2,OVERFLOW #if not, overflow
j EXIT
OVERFLOW:
li $t0,1 # the result is overflow
EXIT:
```



Bkpt	Address	Code	Basic	Source
	0x00400000	0x24110002	addiu $17,$0,0x0000...	3: li $s1,2
	0x00400004	0x24120009	addiu $18,$0,0x0000...	4: li $s2,9
	0x00400008	0x24080000	addiu $8,$0,0x00000000	5: li $t0,0 # No Overflow is default status
	0x0040000c	0x02329821	addu $19,$17,$18	6: addu $s3,$s1,$s2 # s3 = s1 + s2
	0x00400010	0x02324826	xor $9,$17,$18	7: xor $t1,$s1,$s2 # Test if $s1 and $s2 have the same sign
	0x00400014	0x05200004	bltz $9,0x00000004	8: bltz $t1,EXIT # If not, exit
	0x00400018	0x02715026	xor $10,$19,$17	9: xor $t2,$s3,$s1 # Test if $s3 and $s1 have the same sign
	0x0040001c	0x05400001	bltz $10,0x00000001	10: bltz $t2,OVERFLOW #if not, overflow
	0x00400020	0x0810000a	j 0x00400028	11: j EXIT

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

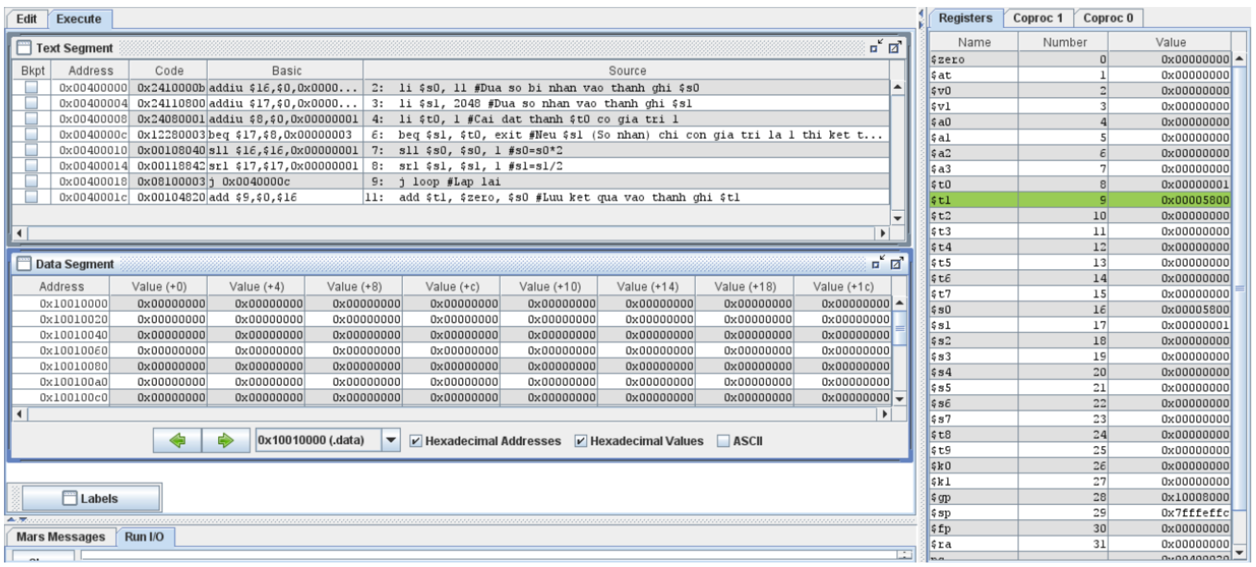
0x10010000 (.data)

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	0x00000000
$t1	9	0x0000000b
$t2	10	0x00000009
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000002
$s2	18	0x00000009
$s3	19	0x0000000b
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000

- Trường hợp 2: Không xảy ra overflow với s1 = 2, s2 = 9 => s3 = 11

Assignment 5:

```
.text
 li $s0, 11 #Dua so bi nhan vao thanh ghi $s0
 li $s1, 2048 #Dua so nhan vao thanh ghi $s1
 li $t0, 1 #Cai dat thanh $t0 co gia tri 1
loop:
 beq $s1, $t0, exit #Neu $s1 (So nhan) chi con gia tri la 1 thi ket thuc vong
lap=
 sll $s0, $s0, 1 #s0=s0*2
 srl $s1, $s1, 1 #s1=s1/2
 j loop #Lap lai
exit:
 add $t1, $zero, $s0 #Luu ket qua vao thanh ghi $t1
```



Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400000	0x2410000b	addiu $16,$0,0x0000...	2: li $s0, 11 #Dua so bi nhan vao thanh ghi $s0
☐	0x00400004	0x24110800	addiu $17,$0,0x0000...	3: li $s1, 2048 #Dua so nhan vao thanh ghi $s1
☐	0x00400008	0x24080001	addiu $8,$0,0x00000001	4: li $t0, 1 #Cai dat thanh $t0 co gia tri 1
☐	0x0040000c	0x12280003	beq $17,$8,0x00000003	6: beq $s1, $t0, exit #Neu $s1 (So nhan) chi con gia tri la 1 thi ket t...
☐	0x00400010	0x00108040	sll $16,$16,0x00000001	7: sll $s0, $s0, 1 #s0=s0*2
☐	0x00400014	0x00118842	srl $17,$17,0x00000001	8: srl $s1, $s1, 1 #s1=s1/2
☐	0x00400018	0x08100003	j 0x0040000c	9: j loop #Lap lai
☐	0x0040001c	0x00104820	add $9,$0,$16	11: add $t1, $zero, $s0 #Luu ket qua vao thanh ghi $t1

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

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

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	0x00000001
$t1	9	0x00005800
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00005800
$s1	17	0x00000001
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000