Ch 3. Arithmetic for Computers

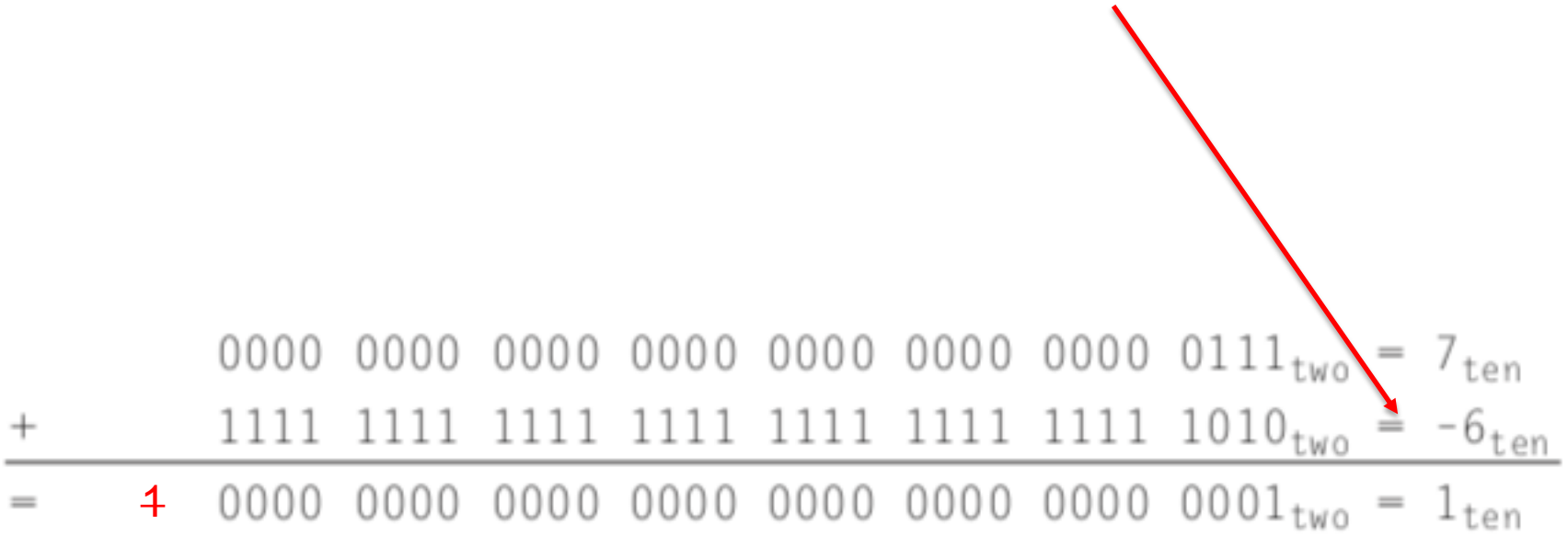
MK
MORGAN KAUFMANN

3.2 정수의 뺄셈

Signed Binary Subtraction : 7-6

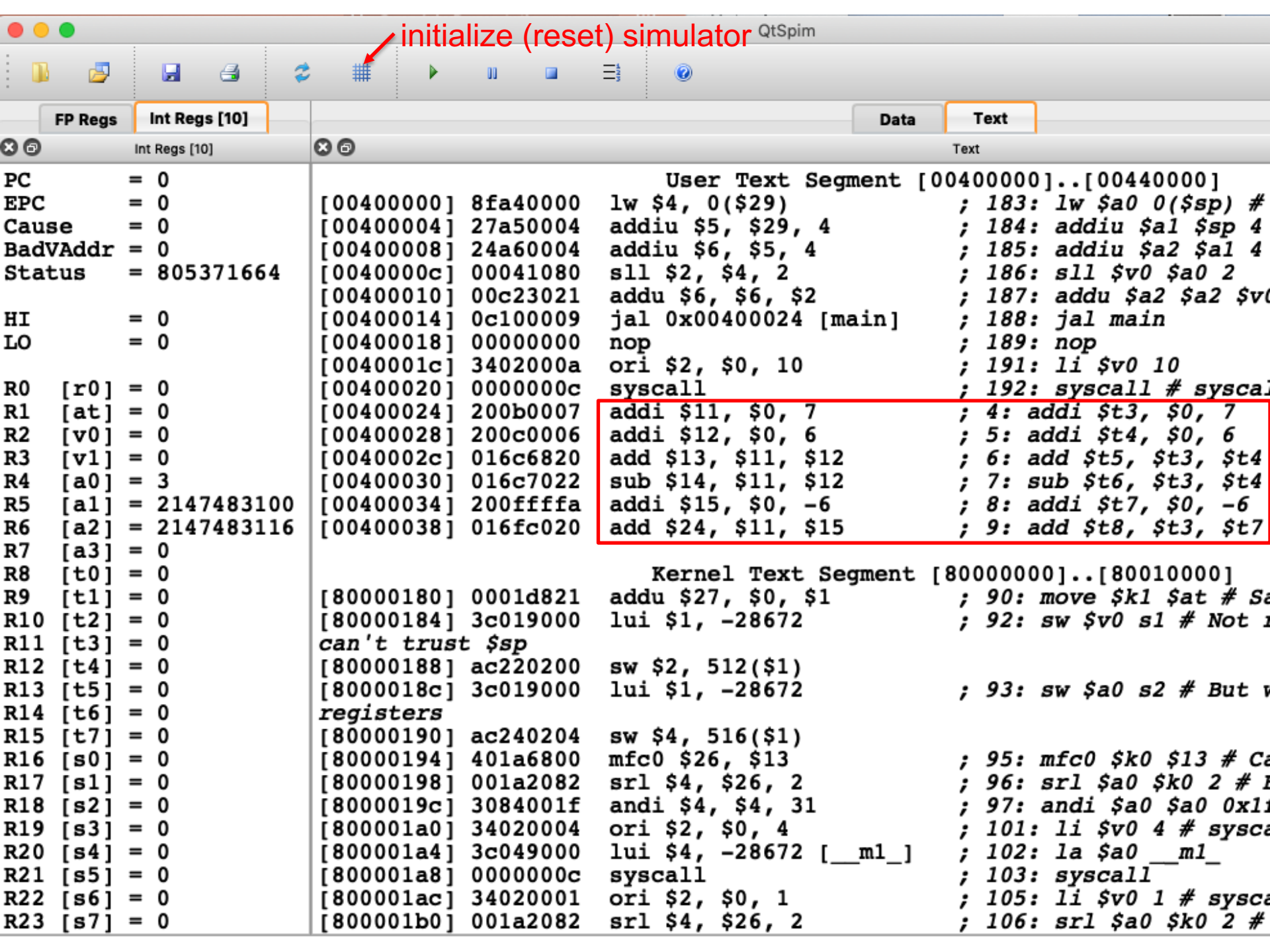
- **7 + (-6)**

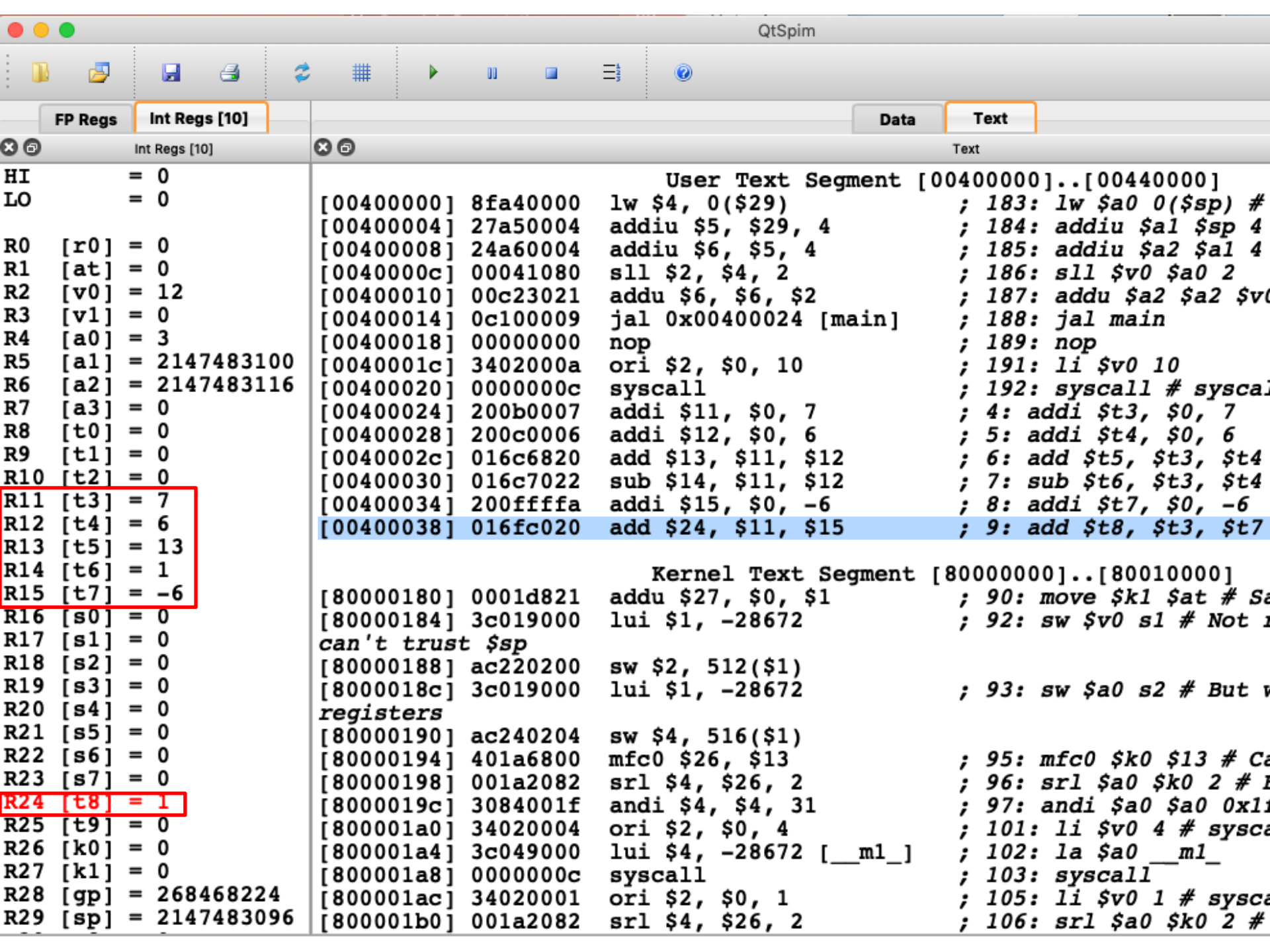
+6 0000 0000 0000 0000 0000 0000 0000 0110
의 1의 보수 1111 1111 1111 1111 1111 1111 1111 1001
\+ 1

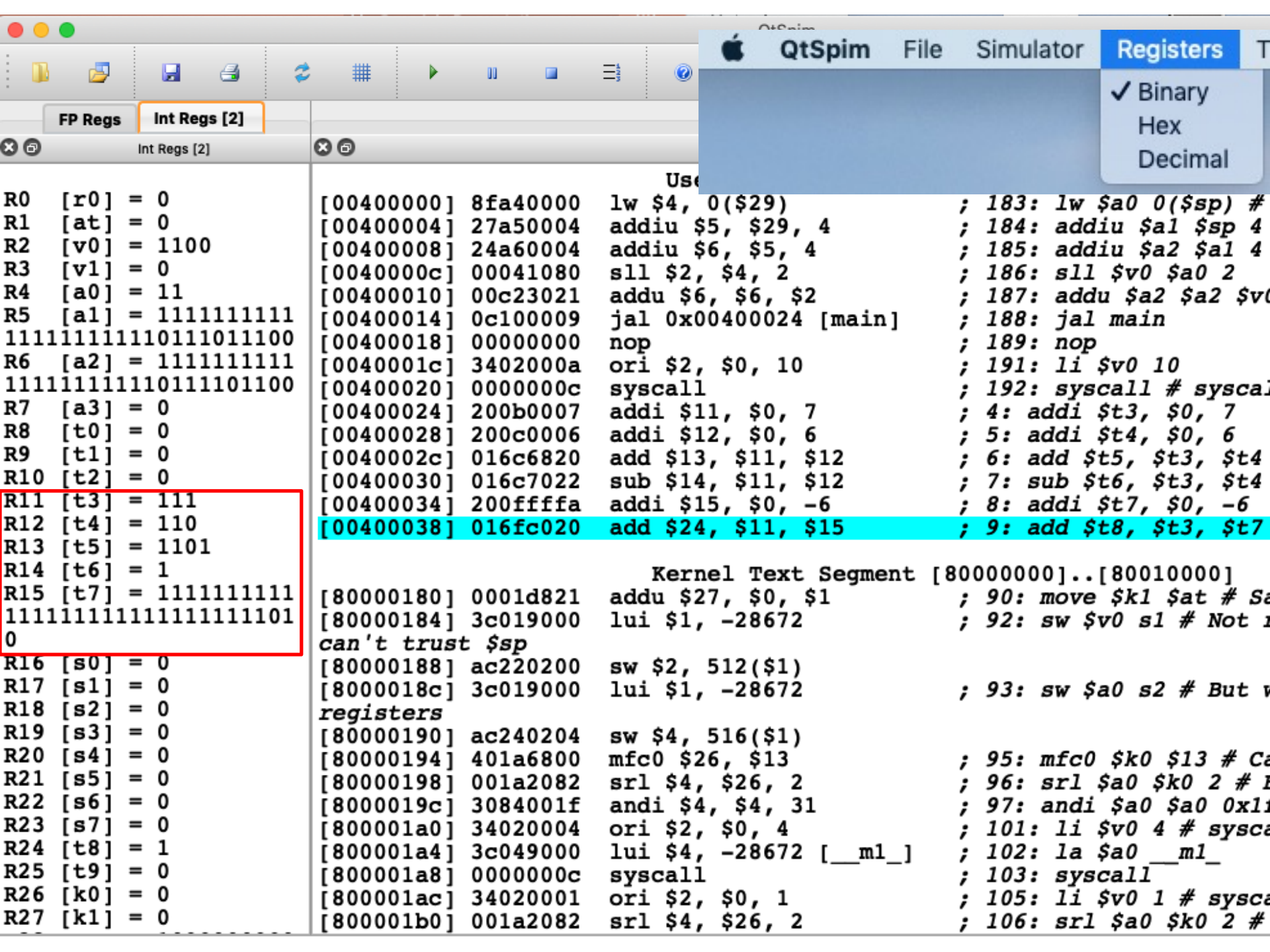


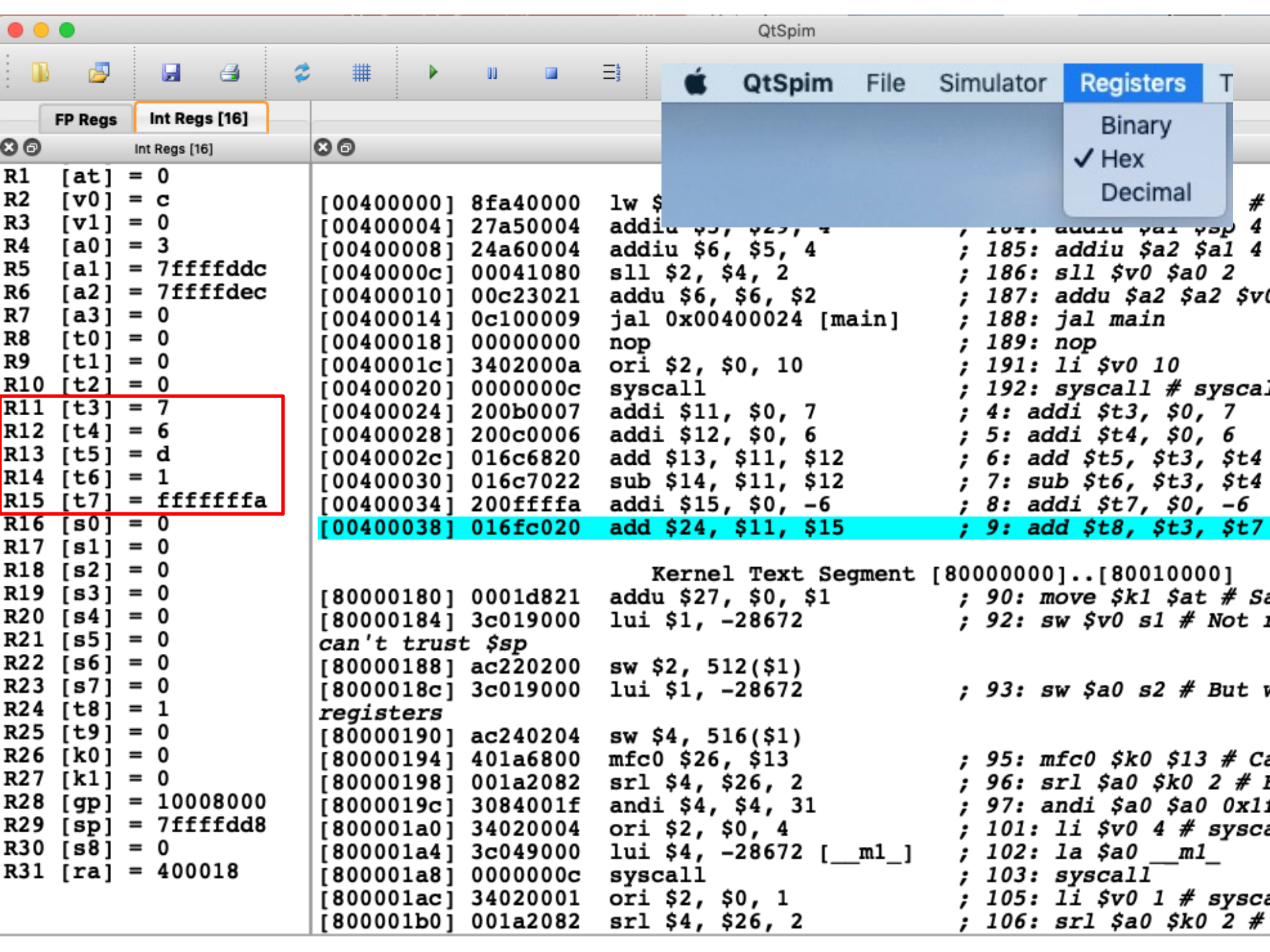
MIPS source file

```
.text
.globl main
main:
    addi $t3, $0, 7
    addi $t4, $0, 6
    add  $t5, $t3, $t4 # 7 + 6
    sub  $t6, $t3, $t4 # 7 - 6
    addi $t7, $0, -6
    add  $t8, $t3, $t7 # 7 + (-6)
```


initialize (reset) simulator
QtSpim
FP Regs
Int Regs [10]
Data
Text
Int Regs [10]
Text
PC = 0
EPC = 0
Cause = 0
BadVAddr = 0
Status = 805371664
HI = 0
LO = 0
R0 [r0] = 0
R1 [at] = 0
R2 [v0] = 0
R3 [v1] = 0
R4 [a0] = 3
R5 [a1] = 2147483100
R6 [a2] = 2147483116
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp) #
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp 4
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1 4
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2 $v
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # sysca
[00400024] 200b0007 addi $11, $0, 7 ; 4: addi $t3, $0, 7
[00400028] 200c0006 addi $12, $0, 6 ; 5: addi $t4, $0, 6
[0040002c] 016c6820 add $13, $11, $12 ; 6: add $t5, $t3, $t4
[00400030] 016c7022 sub $14, $11, $12 ; 7: sub $t6, $t3, $t4
[00400034] 200ffffa addi $15, $0, -6 ; 8: addi $t7, $0, -6
[00400038] 016fc020 add $24, $11, $15 ; 9: add $t8, $t3, $t7
Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821 addu $27, $0, $1 ; 90: move $k1 $at # Sa
[80000184] 3c019000 lui $1, -28672 ; 92: sw $v0 s1 # Not
can't trust $sp
[80000188] ac220200 sw $2, 512($1)
[8000018c] 3c019000 lui $1, -28672 ; 93: sw $a0 s2 # But
registers
[80000190] ac240204 sw $4, 516($1)
[80000194] 401a6800 mfc0 $26, $13 ; 95: mfc0 $k0 $13 # Ca
[80000198] 001a2082 srl $4, $26, 2 ; 96: srl $a0 $k0 2 #
[8000019c] 3084001f andi $4, $4, 31 ; 97: andi $a0 $a0 0x1
[800001a0] 34020004 ori $2, $0, 4 ; 101: li $v0 4 # sysca
[800001a4] 3c049000 lui $4, -28672 [__m1_] ; 102: la $a0 __m1_
[800001a8] 0000000c syscall ; 103: syscall
[800001ac] 34020001 ori $2, $0, 1 ; 105: li $v0 1 # sysca
[800001b0] 001a2082 srl $4, $26, 2 ; 106: srl $a0 $k0 2 #


QtSpim
FP Regs
Int Regs [10]
Data
Text
Int Regs [10]
Text
HI = 0
LO = 0
R0 [r0] = 0
R1 [at] = 0
R2 [v0] = 12
R3 [v1] = 0
R4 [a0] = 3
R5 [a1] = 2147483100
R6 [a2] = 2147483116
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 0
R10 [t2] = 0
R11 [t3] = 7
R12 [t4] = 6
R13 [t5] = 13
R14 [t6] = 1
R15 [t7] = -6
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
R24 [t8] = 1
R25 [t9] = 0
R26 [k0] = 0
R27 [k1] = 0
R28 [gp] = 268468224
R29 [sp] = 2147483096
User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp) #
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp 4
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1 4
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2 $v
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # syscal
[00400024] 200b0007 addi $11, $0, 7 ; 4: addi $t3, $0, 7
[00400028] 200c0006 addi $12, $0, 6 ; 5: addi $t4, $0, 6
[0040002c] 016c6820 add $13, $11, $12 ; 6: add $t5, $t3, $t4
[00400030] 016c7022 sub $14, $11, $12 ; 7: sub $t6, $t3, $t4
[00400034] 200ffffa addi $15, $0, -6 ; 8: addi $t7, $0, -6
[00400038] 016fc020 add $24, $11, $15 ; 9: add $t8, $t3, $t7
Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821 addu $27, $0, $1 ; 90: move $k1 $at # Sa
[80000184] 3c019000 lui $1, -28672 ; 92: sw $v0 s1 # Not
can't trust $sp
[80000188] ac220200 sw $2, 512($1)
[8000018c] 3c019000 lui $1, -28672 ; 93: sw $a0 s2 # But
registers
[80000190] ac240204 sw $4, 516($1)
[80000194] 401a6800 mfc0 $26, $13 ; 95: mfc0 $k0 $13 # Ca
[80000198] 001a2082 srl $4, $26, 2 ; 96: srl $a0 $k0 2 #
[8000019c] 3084001f andi $4, $4, 31 ; 97: andi $a0 $a0 0x1
[800001a0] 34020004 ori $2, $0, 4 ; 101: li $v0 4 # sysca
[800001a4] 3c049000 lui $4, -28672 [__m1_] ; 102: la $a0 __m1_
[800001a8] 0000000c syscall ; 103: syscall
[800001ac] 34020001 ori $2, $0, 1 ; 105: li $v0 1 # sysca
[800001b0] 001a2082 srl $4, $26, 2 ; 106: srl $a0 $k0 2 #


QtSpim
File
Simulator
Registers
Binary
Hex
Decimal
FP Regs
Int Regs [2]
Int Regs [2]
R0 [r0] = 0
R1 [at] = 0
R2 [v0] = 1100
R3 [v1] = 0
R4 [a0] = 11
R5 [a1] = 1111111111
11111111110111011100
R6 [a2] = 1111111111
11111111110111101100
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 0
R10 [t2] = 0
R11 [t3] = 111
R12 [t4] = 110
R13 [t5] = 1101
R14 [t6] = 1
R15 [t7] = 1111111111
11111111111111111101
0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
R24 [t8] = 1
R25 [t9] = 0
R26 [k0] = 0
R27 [k1] = 0
[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp) #
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp 4
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1 4
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2 $v
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # syscal
[00400024] 200b0007 addi $11, $0, 7 ; 4: addi $t3, $0, 7
[00400028] 200c0006 addi $12, $0, 6 ; 5: addi $t4, $0, 6
[0040002c] 016c6820 add $13, $11, $12 ; 6: add $t5, $t3, $t4
[00400030] 016c7022 sub $14, $11, $12 ; 7: sub $t6, $t3, $t4
[00400034] 200ffffa addi $15, $0, -6 ; 8: addi $t7, $0, -6
[00400038] 016fc020 add $24, $11, $15 ; 9: add $t8, $t3, $t7
Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821 addu $27, $0, $1 ; 90: move $k1 $at # Sa
[80000184] 3c019000 lui $1, -28672 ; 92: sw $v0 s1 # Not
can't trust $sp
[80000188] ac220200 sw $2, 512($1)
[8000018c] 3c019000 lui $1, -28672 ; 93: sw $a0 s2 # But
registers
[80000190] ac240204 sw $4, 516($1)
[80000194] 401a6800 mfc0 $26, $13 ; 95: mfc0 $k0 $13 # Ca
[80000198] 001a2082 srl $4, $26, 2 ; 96: srl $a0 $k0 2 #
[8000019c] 3084001f andi $4, $4, 31 ; 97: andi $a0 $a0 0x1
[800001a0] 34020004 ori $2, $0, 4 ; 101: li $v0 4 # sysca
[800001a4] 3c049000 lui $4, -28672 [__m1_] ; 102: la $a0 __m1_
[800001a8] 0000000c syscall ; 103: syscall
[800001ac] 34020001 ori $2, $0, 1 ; 105: li $v0 1 # sysca
[800001b0] 001a2082 srl $4, $26, 2 ; 106: srl $a0 $k0 2 #


QtSpim
QtSpim File Simulator Registers T
Binary
Hex
Decimal
FP Regs
Int Regs [16]
Int Regs [16]
R1 [at] = 0
R2 [v0] = c
R3 [v1] = 0
R4 [a0] = 3
R5 [a1] = 7ffffddc
R6 [a2] = 7ffffdec
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 0
R10 [t2] = 0
R11 [t3] = 7
R12 [t4] = 6
R13 [t5] = d
R14 [t6] = 1
R15 [t7] = fffffffa
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
R24 [t8] = 1
R25 [t9] = 0
R26 [k0] = 0
R27 [k1] = 0
R28 [gp] = 10008000
R29 [sp] = 7ffffdd8
R30 [s8] = 0
R31 [ra] = 400018
[00400000] 8fa40000 lw $
[00400004] 27a50004 addiu
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1 4
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2 $v0
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # syscal
[00400024] 200b0007 addi $11, $0, 7 ; 4: addi $t3, $0, 7
[00400028] 200c0006 addi $12, $0, 6 ; 5: addi $t4, $0, 6
[0040002c] 016c6820 add $13, $11, $12 ; 6: add $t5, $t3, $t4
[00400030] 016c7022 sub $14, $11, $12 ; 7: sub $t6, $t3, $t4
[00400034] 200ffffa addi $15, $0, -6 ; 8: addi $t7, $0, -6
[00400038] 016fc020 add $24, $11, $15 ; 9: add $t8, $t3, $t7
Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821 addu $27, $0, $1 ; 90: move $k1 $at # Sa
[80000184] 3c019000 lui $1, -28672 ; 92: sw $v0 s1 # Not r
can't trust $sp
[80000188] ac220200 sw $2, 512($1)
[8000018c] 3c019000 lui $1, -28672 ; 93: sw $a0 s2 # But w
registers
[80000190] ac240204 sw $4, 516($1)
[80000194] 401a6800 mfc0 $26, $13 ; 95: mfc0 $k0 $13 # Ca
[80000198] 001a2082 srl $4, $26, 2 ; 96: srl $a0 $k0 2 # E
[8000019c] 3084001f andi $4, $4, 31 ; 97: andi $a0 $a0 0x1f
[800001a0] 34020004 ori $2, $0, 4 ; 101: li $v0 4 # sysca
[800001a4] 3c049000 lui $4, -28672 [__m1_] ; 102: la $a0 __m1_
[800001a8] 0000000c syscall ; 103: syscall
[800001ac] 34020001 ori $2, $0, 1 ; 105: li $v0 1 # sysca
[800001b0] 001a2082 srl $4, $26, 2 ; 106: srl $a0 $k0 2 #

3.2 정수의 뺄셈

- a – b = a + (-b) 로 계산
- 즉, b 를 양수/음수 관계없이 –b (2의 보수) 로 만들어 덧셈을 한다.

Signed Binary Subtraction : 7-6

- 7 + (-6)

$$
\begin{array}{lll}
 & 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0111_{two} & = 7_{ten} \\
+ & 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1010_{two} & = -6_{ten} \\
\hline
= \quad \cancel{1} & 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001_{two} & = 1_{ten}
\end{array}
$$

2,147,483,647 – (-2)

- a – b = a + (-b) 로 계산
- 즉, b 를 양수/음수 관계없이 –b (2의 보수) 로 만들어 덧셈을 한다.

0111 1111 1111 1111 1111 1111 1111 1111 = 2147483647
1111 1111 1111 1111 1111 1111 1111 1110 = -2

0111 1111 1111 1111 1111 1111 1111 1111 = 2147483647
0000 0000 0000 0000 0000 0000 0000 0010 = -2 의 2's complement

---

1000 0000 0000 0000 0000 0000 0000 0001 → ? = (-2147483647
not +2147483649)

Overflow

- 연산의 결과가 32-bit word 로 표현될 수 없을 때

Operation	Operand A	Operand B	Result indicating overflow
$A + B$	≥ 0	≥ 0	< 0
$A + B$	< 0	< 0	≥ 0
$A - B$	≥ 0	< 0	< 0
$A - B$	< 0	≥ 0	≥ 0

0111 1111 1111 1111 1111 1111 1111 1111 = 2147483647

0111 1111 1111 1111 1111 1111 1111 1111 = 2147483647

1111 1111 1111 1111 1111 1111 1111 1110 = -2

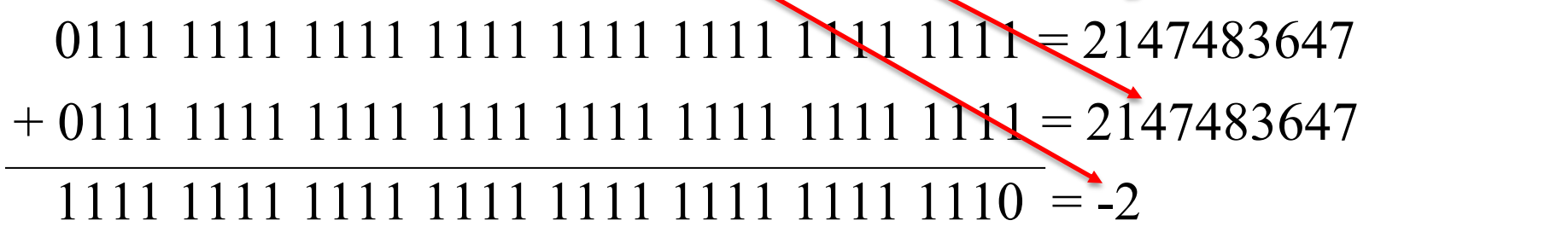
→ overflow exception 발생

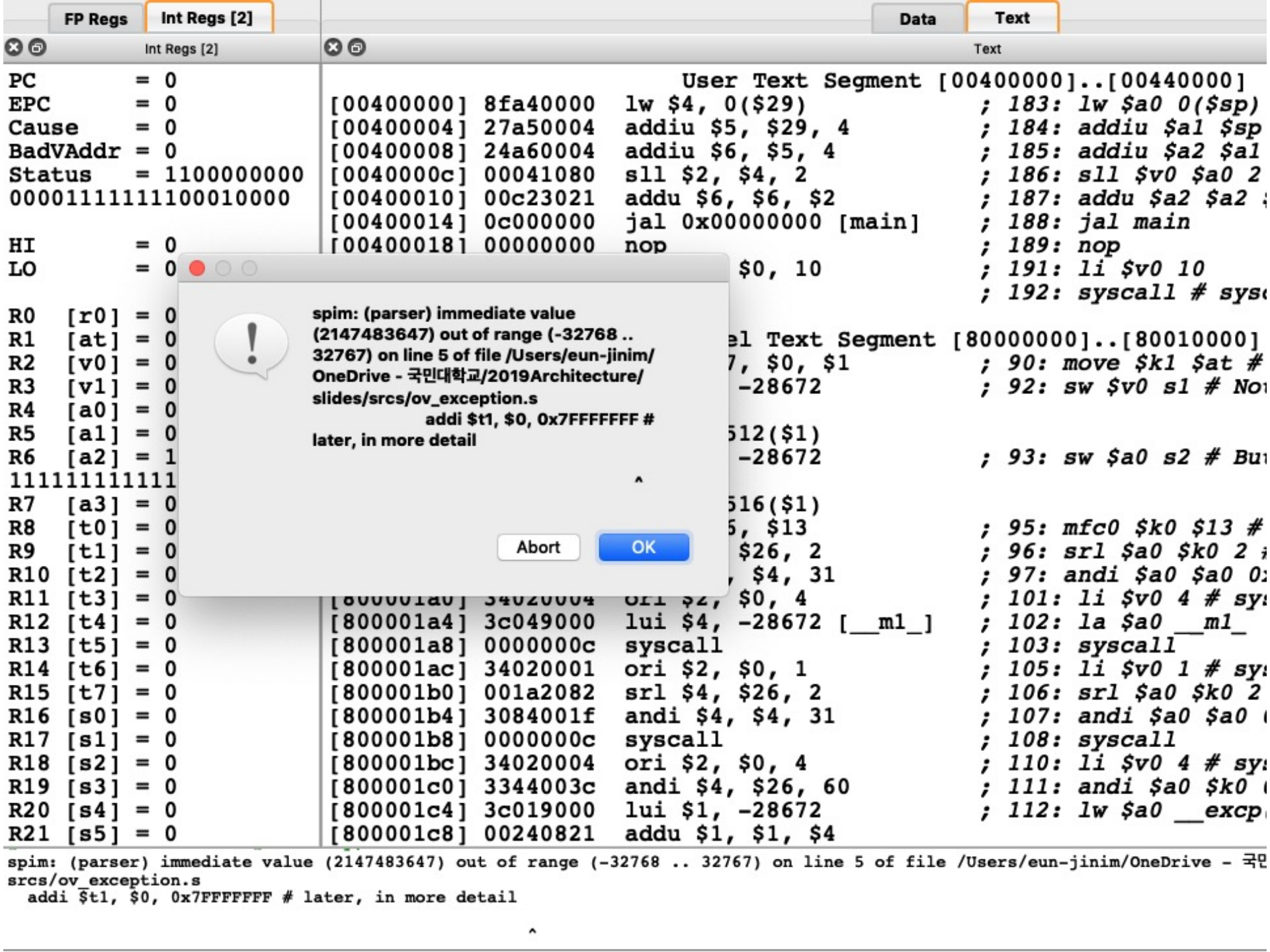
add/addi/sub instructions

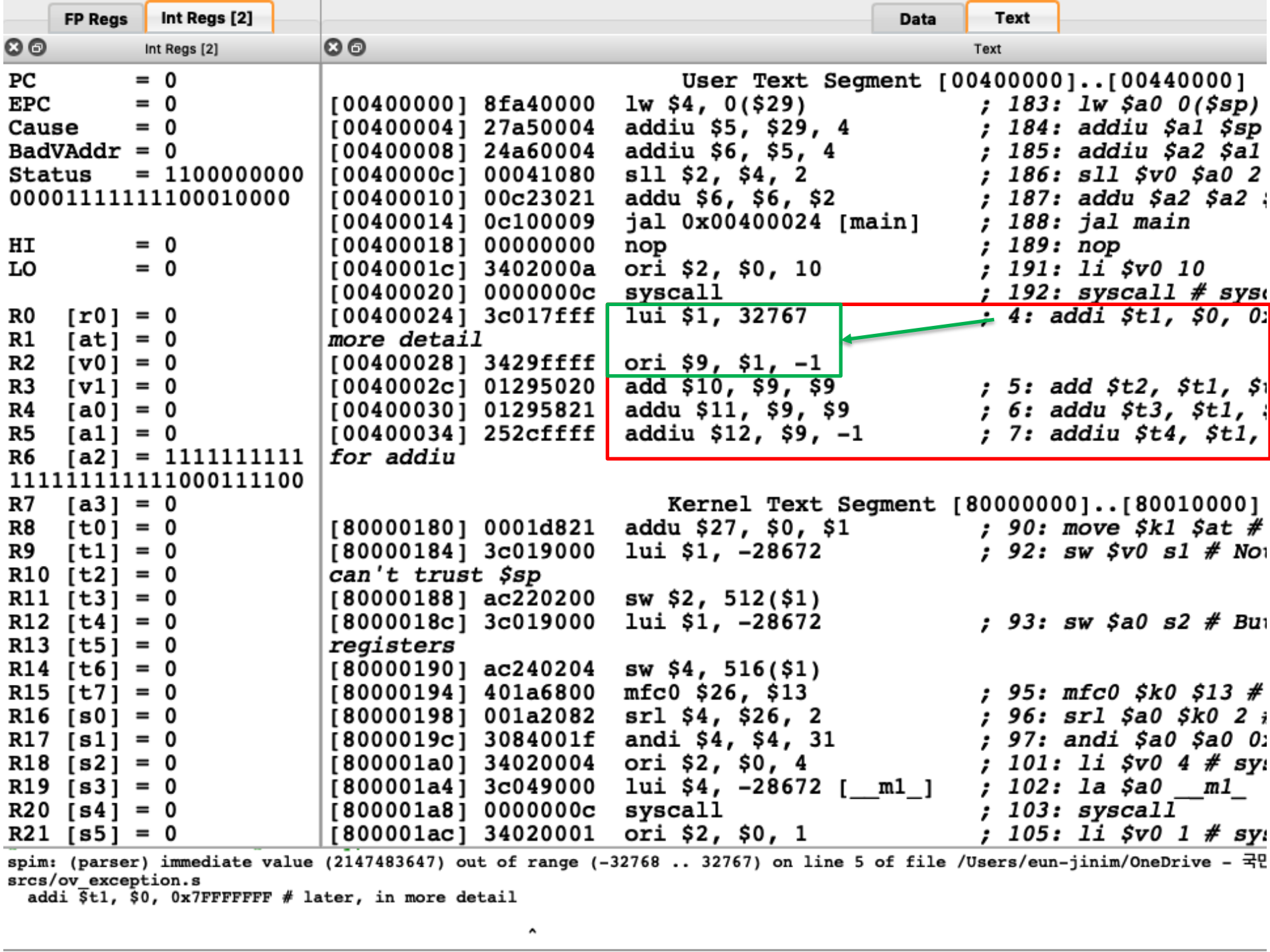
- add/addi/sub 명령어는 연산 결과 overflow 가 되면 exception 을 발생한다.
- exception (예외) : 프로그램의 정상적인 수행을 방해하는 계획되지 않은 사건

MIPS source file

```
.text
.globl main
main:
    addi $t1, $0, 0x7FFFFFFF # later, in more detail
    add  $t2, $t1, $t1 # overflow
    addu $t3, $t1, $t1 # no exception
    addiu $t4, $t1, -1 # negative constant for addiu
```




FP Regs
Int Regs [2]
Data
Text
Int Regs [2]
Text
PC = 0
EPC = 0
Cause = 0
BadVAddr = 0
Status = 1100000000
00001111111100010000
HI = 0
LO = 0
R0 [r0] = 0
R1 [at] = 0
R2 [v0] = 0
R3 [v1] = 0
R4 [a0] = 0
R5 [a1] = 0
R6 [a2] = 1
11111111111
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp)
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2
[00400014] 0c000000 jal 0x00000000 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
$0, 10 ; 191: li $v0 10
; 192: syscall # sys
el Text Segment [80000000]..[80010000]
7, $0, $1 ; 90: move $k1 $at #
-28672 ; 92: sw $v0 s1 # No
512($1)
-28672 ; 93: sw $a0 s2 # Bu
516($1)
5, $13 ; 95: mfc0 $k0 $13 #
$26, 2 ; 96: srl $a0 $k0 2
, $4, 31 ; 97: andi $a0 $a0 0
$0, 4 ; 101: li $v0 4 # sy
[800001a4] 3c049000 lui $4, -28672 [__m1_] ; 102: la $a0 __m1_
[800001a8] 0000000c syscall ; 103: syscall
[800001ac] 34020001 ori $2, $0, 1 ; 105: li $v0 1 # sy
[800001b0] 001a2082 srl $4, $26, 2 ; 106: srl $a0 $k0 2
[800001b4] 3084001f andi $4, $4, 31 ; 107: andi $a0 $a0
[800001b8] 0000000c syscall ; 108: syscall
[800001bc] 34020004 ori $2, $0, 4 ; 110: li $v0 4 # sy
[800001c0] 3344003c andi $4, $26, 60 ; 111: andi $a0 $k0
[800001c4] 3c019000 lui $1, -28672 ; 112: lw $a0 __excp
[800001c8] 00240821 addu $1, $1, $4
spim: (parser) immediate value (2147483647) out of range (-32768 .. 32767) on line 5 of file /Users/eun-jinim/OneDrive - 국민
srcs/ov_exception.s
addi $t1, $0, 0x7FFFFFFF # later, in more detail
^
spim: (parser) immediate value (2147483647) out of range (-32768 .. 32767) on line 5 of file /Users/eun-jinim/OneDrive - 국민대학교/2019Architecture/slides/srcs/ov_exception.s
addi $t1, $0, 0x7FFFFFFF # later, in more detail
^
Abort
OK


FP Regs
Int Regs [2]
Data
Text
Int Regs [2]
Text
PC = 0
EPC = 0
Cause = 0
BadVAddr = 0
Status = 1100000000
00001111111100010000
HI = 0
LO = 0
R0 [r0] = 0
R1 [at] = 0
R2 [v0] = 0
R3 [v1] = 0
R4 [a0] = 0
R5 [a1] = 0
R6 [a2] = 1111111111
11111111111000111100
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp)
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # sys
[00400024] 3c017fff lui $1, 32767 ; 4: addi $t1, $0, 0
more detail
[00400028] 3429ffff ori $9, $1, -1
[0040002c] 01295020 add $10, $9, $9 ; 5: add $t2, $t1, $t
[00400030] 01295821 addu $11, $9, $9 ; 6: addu $t3, $t1,
[00400034] 252cffff addiu $12, $9, -1 ; 7: addiu $t4, $t1,
for addiu
Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821 addu $27, $0, $1 ; 90: move $k1 $at #
[80000184] 3c019000 lui $1, -28672 ; 92: sw $v0 s1 # No
can't trust $sp
[80000188] ac220200 sw $2, 512($1)
[8000018c] 3c019000 lui $1, -28672 ; 93: sw $a0 s2 # Bu
registers
[80000190] ac240204 sw $4, 516($1)
[80000194] 401a6800 mfc0 $26, $13 ; 95: mfc0 $k0 $13 #
[80000198] 001a2082 srl $4, $26, 2 ; 96: srl $a0 $k0 2
[8000019c] 3084001f andi $4, $4, 31 ; 97: andi $a0 $a0 0
[800001a0] 34020004 ori $2, $0, 4 ; 101: li $v0 4 # sy
[800001a4] 3c049000 lui $4, -28672 [__m1_] ; 102: la $a0 __m1_
[800001a8] 0000000c syscall ; 103: syscall
[800001ac] 34020001 ori $2, $0, 1 ; 105: li $v0 1 # sy
spim: (parser) immediate value (2147483647) out of range (-32768 .. 32767) on line 5 of file /Users/eun-jinim/OneDrive - 국민
srcs/ov_exception.s
addi $t1, $0, 0x7FFFFFFF # later, in more detail

FP Regs | Int Regs [16] | Data | Text

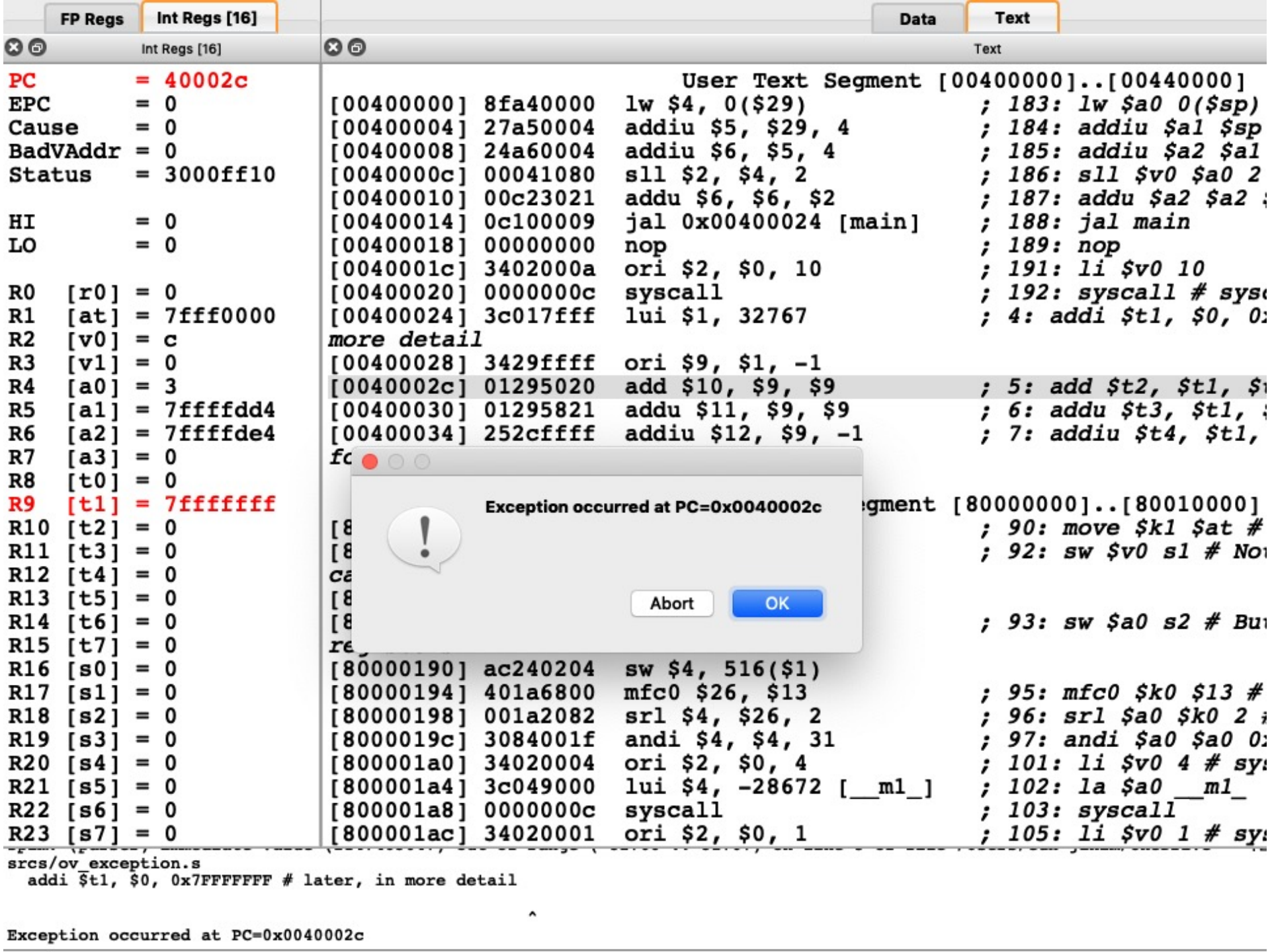
Int Regs [16]

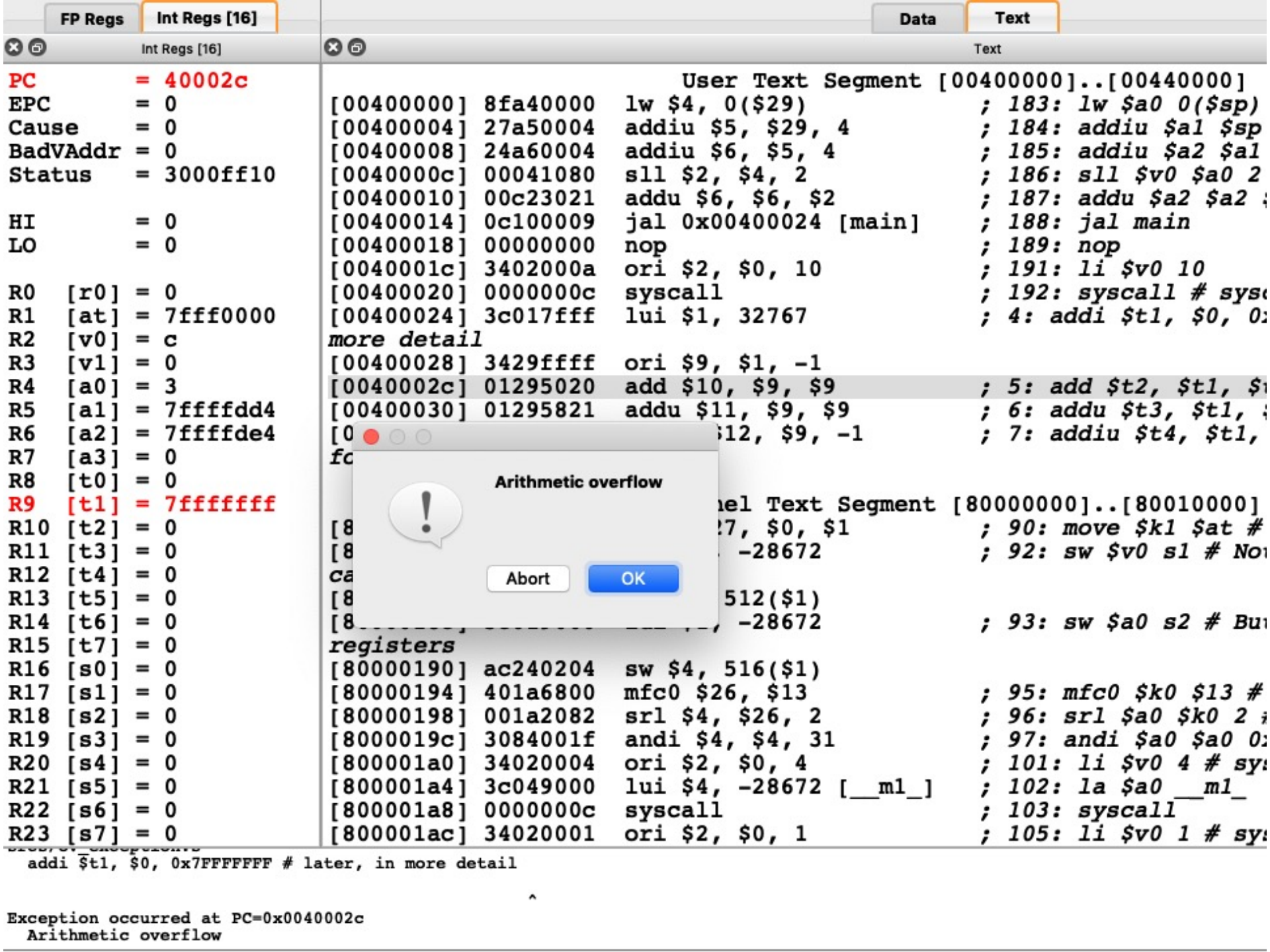
```
PC        = 40002c
EPC       = 0
Cause     = 0
BadVAddr  = 0
Status    = 3000ff10

HI        = 0
LO        = 0

R0  [r0]  = 0
R1  [at]  = 7fff0000
R2  [v0]  = c
R3  [v1]  = 0
R4  [a0]  = 3
R5  [a1]  = 7ffffdd4
R6  [a2]  = 7ffffde4
R7  [a3]  = 0
R8  [t0]  = 0
R9  [t1]  = 7fffffff
R10 [t2]  = 0
R11 [t3]  = 0
R12 [t4]  = 0
R13 [t5]  = 0
R14 [t6]  = 0
R15 [t7]  = 0
R16 [s0]  = 0
R17 [s1]  = 0
R18 [s2]  = 0
R19 [s3]  = 0
R20 [s4]  = 0
R21 [s5]  = 0
R22 [s6]  = 0
R23 [s7]  = 0
```

Text

```
                    User Text Segment [00400000]..[00440000]
[00400000] 8fa40000  lw $4, 0($29)              ; 183: lw $a0 0($sp)
[00400004] 27a50004  addiu $5, $29, 4          ; 184: addiu $a1 $sp
[00400008] 24a60004  addiu $6, $5, 4           ; 185: addiu $a2 $a1
[0040000c] 00041080  sll $2, $4, 2             ; 186: sll $v0 $a0 2
[00400010] 00c23021  addu $6, $6, $2           ; 187: addu $a2 $a2 $
[00400014] 0c100009  jal 0x00400024 [main]     ; 188: jal main
[00400018] 00000000  nop                       ; 189: nop
[0040001c] 3402000a  ori $2, $0, 10            ; 191: li $v0 10
[00400020] 0000000c  syscall                   ; 192: syscall # sysc
[00400024] 3c017fff  lui $1, 32767             ; 4: addi $t1, $0, 0x
more detail
[00400028] 3429ffff  ori $9, $1, -1
[0040002c] 01295020  add $10, $9, $9           ; 5: add $t2, $t1, $t
[00400030] 01295821  addu $11, $9, $9          ; 6: addu $t3, $t1, $
[00400034] 252cffff  addiu $12, $9, -1         ; 7: addiu $t4, $t1,
for addiu

                    Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821  addu $27, $0, $1          ; 90: move $k1 $at #
[80000184] 3c019000  lui $1, -28672            ; 92: sw $v0 s1 # Not
can't trust $sp
[80000188] ac220200  sw $2, 512($1)
[8000018c] 3c019000  lui $1, -28672            ; 93: sw $a0 s2 # But
registers
[80000190] ac240204  sw $4, 516($1)
[80000194] 401a6800  mfc0 $26, $13             ; 95: mfc0 $k0 $13 #
[80000198] 001a2082  srl $4, $26, 2            ; 96: srl $a0 $k0 2 #
[8000019c] 3084001f  andi $4, $4, 31           ; 97: andi $a0 $a0 0x
[800001a0] 34020004  ori $2, $0, 4             ; 101: li $v0 4 # sys
[800001a4] 3c049000  lui $4, -28672 [__m1_]    ; 102: la $a0 __m1_
[800001a8] 0000000c  syscall                   ; 103: syscall
[800001ac] 34020001  ori $2, $0, 1             ; 105: li $v0 1 # sys
```

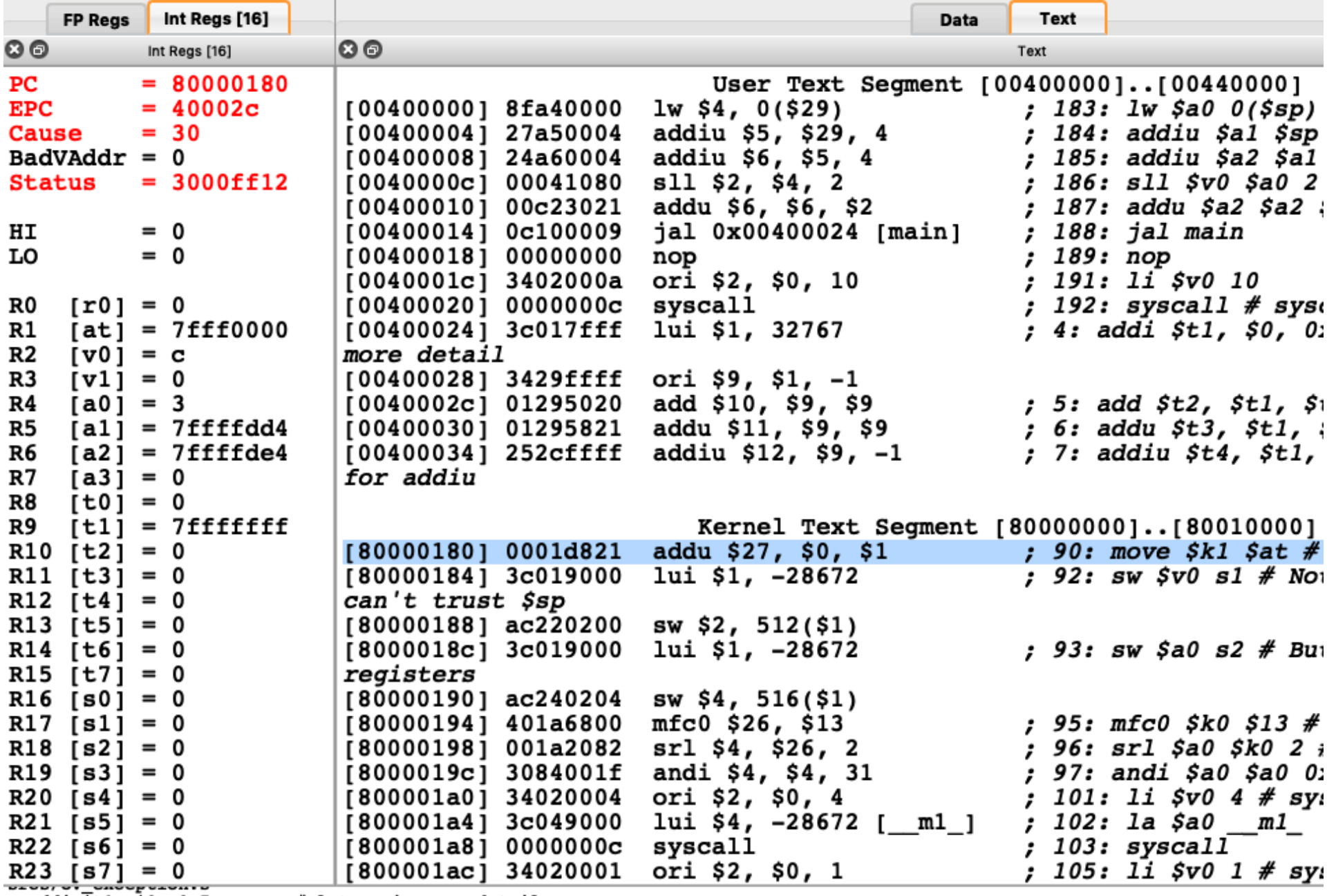
```
spim: (parser) immediate value (2147483647) out of range (-32768 .. 32767) on line 5 of file /Users/eun-jinim/OneDrive - 국민
srcs/ov_exception.s
  addi $t1, $0, 0x7FFFFFFF # later, in more detail
                           ^
```

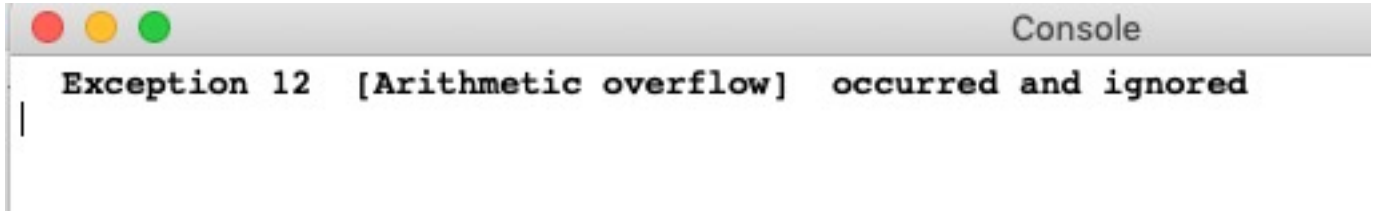

FP Regs
Int Regs [16]
Data
Text
Int Regs [16]
Text
PC = 40002c
EPC = 0
Cause = 0
BadVAddr = 0
Status = 3000ff10
HI = 0
LO = 0
R0 [r0] = 0
R1 [at] = 7fff0000
R2 [v0] = c
R3 [v1] = 0
R4 [a0] = 3
R5 [a1] = 7fffdd4
R6 [a2] = 7fffde4
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 7fffffff
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp)
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # sys
[00400024] 3c017fff lui $1, 32767 ; 4: addi $t1, $0, 0
more detail
[00400028] 3429ffff ori $9, $1, -1
[0040002c] 01295020 add $10, $9, $9 ; 5: add $t2, $t1, $
[00400030] 01295821 addu $11, $9, $9 ; 6: addu $t3, $t1,
[00400034] 252cffff addiu $12, $9, -1 ; 7: addiu $t4, $t1,
Exception occurred at PC=0x0040002c
Abort
OK
gment [80000000]..[80010000]
; 90: move $k1 $at #
; 92: sw $v0 s1 # No
; 93: sw $a0 s2 # Bu
[80000190] ac240204 sw $4, 516($1)
[80000194] 401a6800 mfc0 $26, $13 ; 95: mfc0 $k0 $13 #
[80000198] 001a2082 srl $4, $26, 2 ; 96: srl $a0 $k0 2
[8000019c] 3084001f andi $4, $4, 31 ; 97: andi $a0 $a0 0
[800001a0] 34020004 ori $2, $0, 4 ; 101: li $v0 4 # sy
[800001a4] 3c049000 lui $4, -28672 [__m1_] ; 102: la $a0 __m1_
[800001a8] 0000000c syscall ; 103: syscall
[800001ac] 34020001 ori $2, $0, 1 ; 105: li $v0 1 # sy
srcs/ov_exception.s
addi $t1, $0, 0x7FFFFFFF # later, in more detail
Exception occurred at PC=0x0040002c


FP Regs
Int Regs [16]
Data
Text
Int Regs [16]
Text
PC = 40002c
EPC = 0
Cause = 0
BadVAddr = 0
Status = 3000ff10
HI = 0
LO = 0
R0 [r0] = 0
R1 [at] = 7fff0000
R2 [v0] = c
R3 [v1] = 0
R4 [a0] = 3
R5 [a1] = 7fffdd4
R6 [a2] = 7fffde4
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 7fffffff
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp)
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # sys
[00400024] 3c017fff lui $1, 32767 ; 4: addi $t1, $0, 0
more detail
[00400028] 3429ffff ori $9, $1, -1
[0040002c] 01295020 add $10, $9, $9 ; 5: add $t2, $t1, $
[00400030] 01295821 addu $11, $9, $9 ; 6: addu $t3, $t1,
[0 512, $9, -1 ; 7: addiu $t4, $t1,
fc
Arithmetic overflow
Abort
OK
el Text Segment [80000000]..[80010000]
7, $0, $1 ; 90: move $k1 $at #
-28672 ; 92: sw $v0 s1 # No
ca
512($1)
-28672 ; 93: sw $a0 s2 # Bu
registers
[80000190] ac240204 sw $4, 516($1)
[80000194] 401a6800 mfc0 $26, $13 ; 95: mfc0 $k0 $13 #
[80000198] 001a2082 srl $4, $26, 2 ; 96: srl $a0 $k0 2
[8000019c] 3084001f andi $4, $4, 31 ; 97: andi $a0 $a0 0
[800001a0] 34020004 ori $2, $0, 4 ; 101: li $v0 4 # sy
[800001a4] 3c049000 lui $4, -28672 [__m1_] ; 102: la $a0 __m1_
[800001a8] 0000000c syscall ; 103: syscall
[800001ac] 34020001 ori $2, $0, 1 ; 105: li $v0 1 # sy
addi $t1, $0, 0x7FFFFFFF # later, in more detail
Exception occurred at PC=0x0040002c
Arithmetic overflow

Exception Handler 수행

- source code 의 6번 행으로 돌아올 때까지 계속 single step


FP Regs
Int Regs [16]
Data
Text
Int Regs [16]
Text
PC = 80000180
EPC = 40002c
Cause = 30
BadVAddr = 0
Status = 3000ff12
HI = 0
LO = 0
R0 [r0] = 0
R1 [at] = 7fff0000
R2 [v0] = c
R3 [v1] = 0
R4 [a0] = 3
R5 [a1] = 7fffdd4
R6 [a2] = 7fffde4
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 7fffffff
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp)
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # sys
[00400024] 3c017fff lui $1, 32767 ; 4: addi $t1, $0, 0
more detail
[00400028] 3429ffff ori $9, $1, -1
[0040002c] 01295020 add $10, $9, $9 ; 5: add $t2, $t1, $
[00400030] 01295821 addu $11, $9, $9 ; 6: addu $t3, $t1,
[00400034] 252cffff addiu $12, $9, -1 ; 7: addiu $t4, $t1,
for addiu
Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821 addu $27, $0, $1 ; 90: move $k1 $at #
[80000184] 3c019000 lui $1, -28672 ; 92: sw $v0 s1 # No
can't trust $sp
[80000188] ac220200 sw $2, 512($1)
[8000018c] 3c019000 lui $1, -28672 ; 93: sw $a0 s2 # Bu
registers
[80000190] ac240204 sw $4, 516($1)
[80000194] 401a6800 mfc0 $26, $13 ; 95: mfc0 $k0 $13 #
[80000198] 001a2082 srl $4, $26, 2 ; 96: srl $a0 $k0 2
[8000019c] 3084001f andi $4, $4, 31 ; 97: andi $a0 $a0 0
[800001a0] 34020004 ori $2, $0, 4 ; 101: li $v0 4 # sy
[800001a4] 3c049000 lui $4, -28672 [__m1_] ; 102: la $a0 __m1_
[800001a8] 0000000c syscall ; 103: syscall
[800001ac] 34020001 ori $2, $0, 1 ; 105: li $v0 1 # sy
addi $t1, $0, 0x7FFFFFFF # later, in more detail
Exception occurred at PC=0x0040002c
Arithmetic overflow


Console
Exception 12 [Arithmetic overflow] occurred and ignored

Unsigned Add/Sub

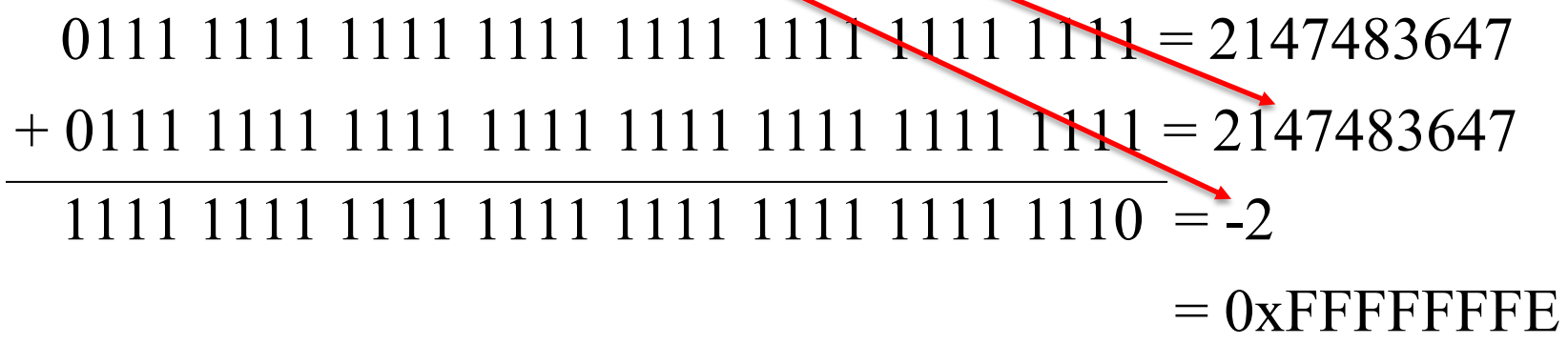
Category	Instruction	Example	Meaning	Comments
Arithmetic	add	`add $s1,$s2,$s3`	$s1 = $s2 + $s3	Three operands; overflow detected
	subtract	`sub $s1,$s2,$s3`	$s1 = $s2 – $s3	Three operands; overflow detected
	add immediate	`addi $s1,$s2,100`	$s1 = $s2 + 100	+ constant; overflow detected
	add unsigned	`addu $s1,$s2,$s3`	$s1 = $s2 + $s3	Three operands; overflow undetected
	subtract unsigned	`subu $s1,$s2,$s3`	$s1 = $s2 – $s3	Three operands; overflow undetected
	add immediate unsigned	`addiu $s1,$s2,100`	$s1 = $s2 + 100	+ constant; overflow undetected
	move from coprocessor register	`mfc0 $s1,$epc`	$s1 = $epc	Copy Exception PC + special regs
	multiply	`mult $s2,$s3`	Hi, Lo = $s2 × $s3	64-bit signed product in Hi, Lo
	multiply unsigned	`multu $s2,$s3`	Hi, Lo = $s2 × $s3	64-bit unsigned product in Hi, Lo
	divide	`div $s2,$s3`	Lo = $s2 / $s3, Hi = $s2 mod $s3	Lo = quotient, Hi = remainder
	divide unsigned	`divu $s2,$s3`	Lo = $s2 / $s3, Hi = $s2 mod $s3	Unsigned quotient and remainder
	move from Hi	`mfhi $s1`	$s1 = Hi	Used to get copy of Hi
	move from Lo	`mflo $s1`	$s1 = Lo	Used to get copy of Lo

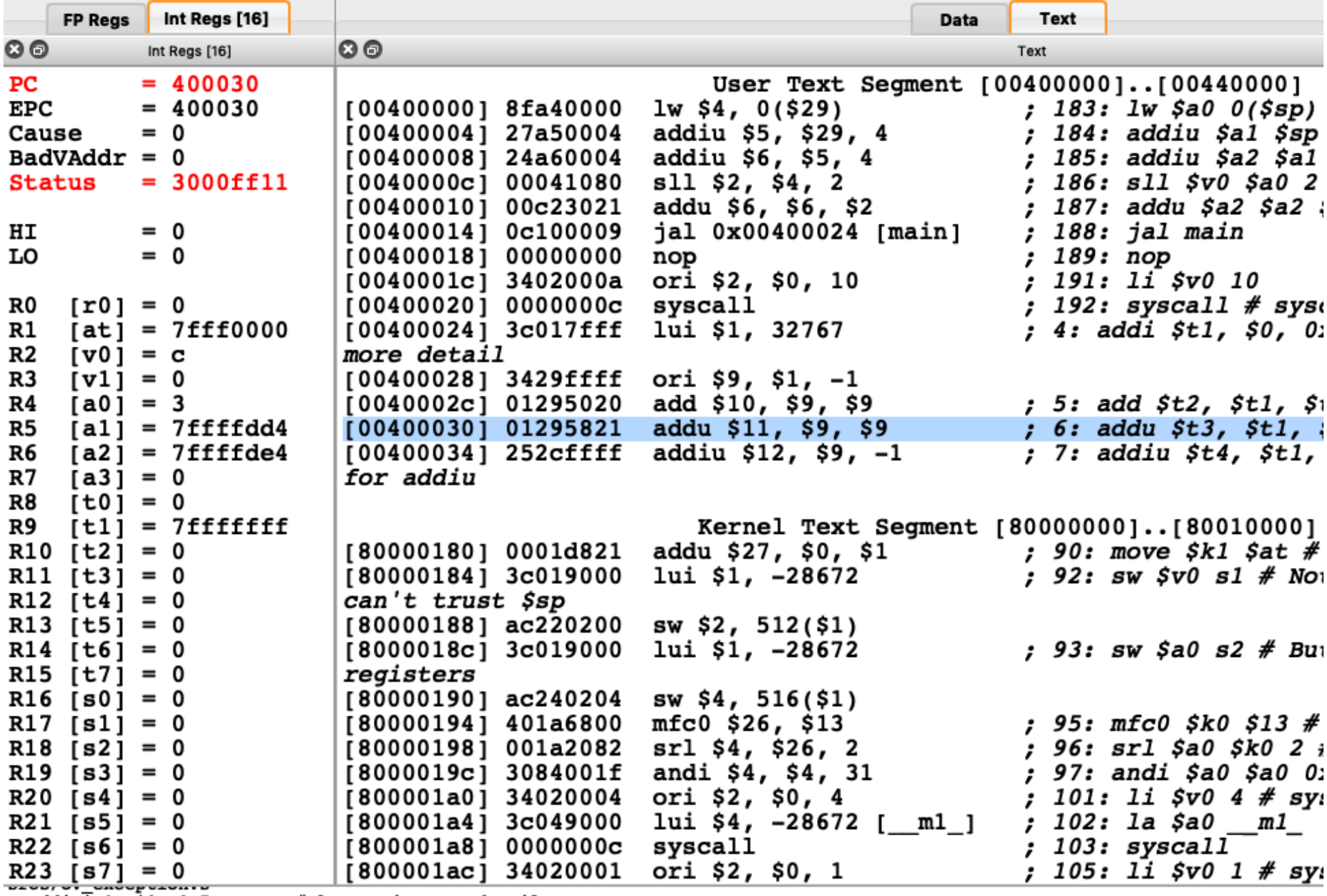
addu/addiu/subu instructions

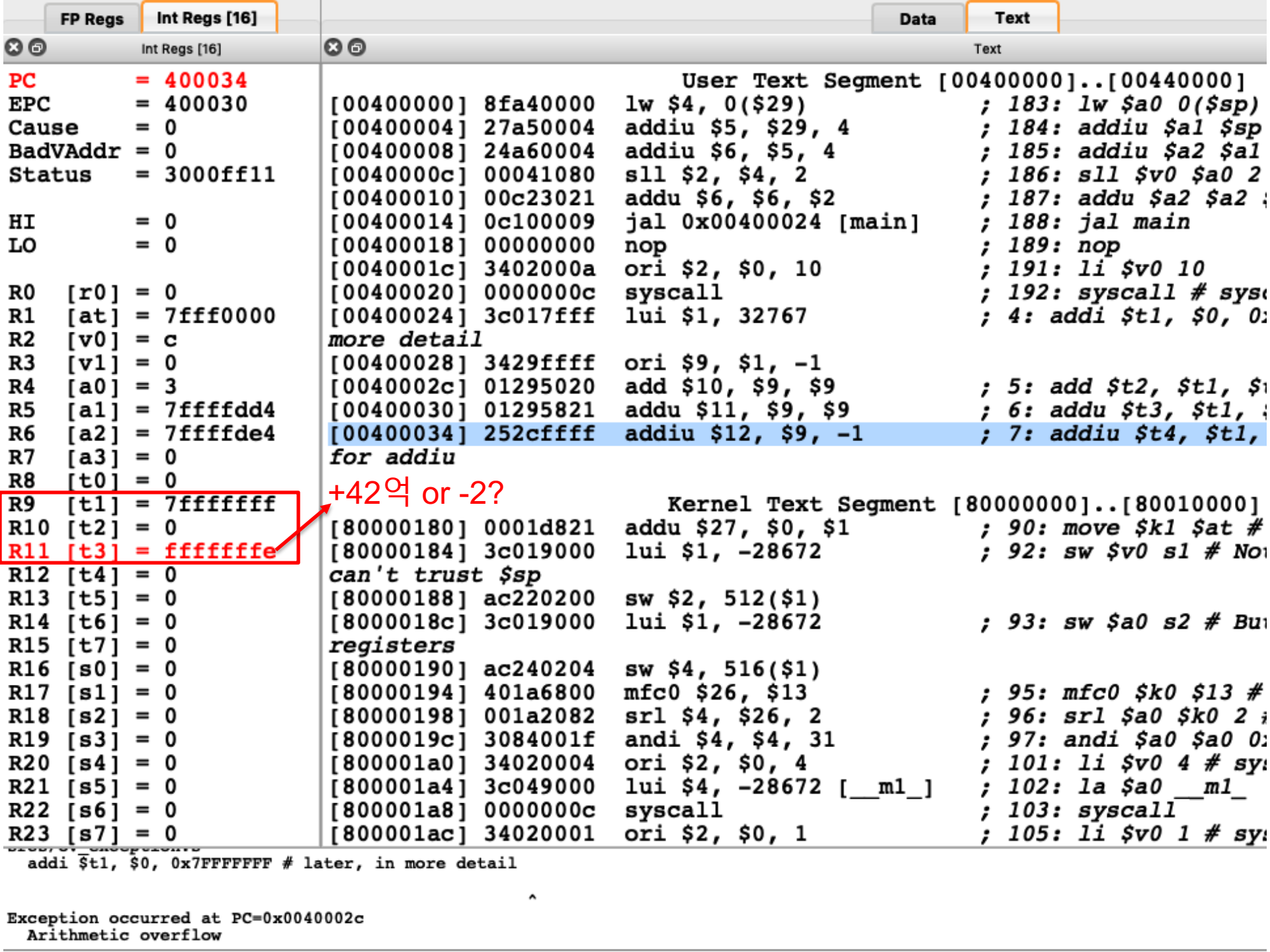
- MIPS unsigned 명령어의 의미 : 연산의 결과 overflow 가 되어도 exception 이 발생하지 않는다.
- add/addi/sub 명령어는 overflow 가 되면 exception 을 발생한다.
- exception (예외) : 프로그램의 정상적인 수행을 방해하는 계획되지 않은 사건

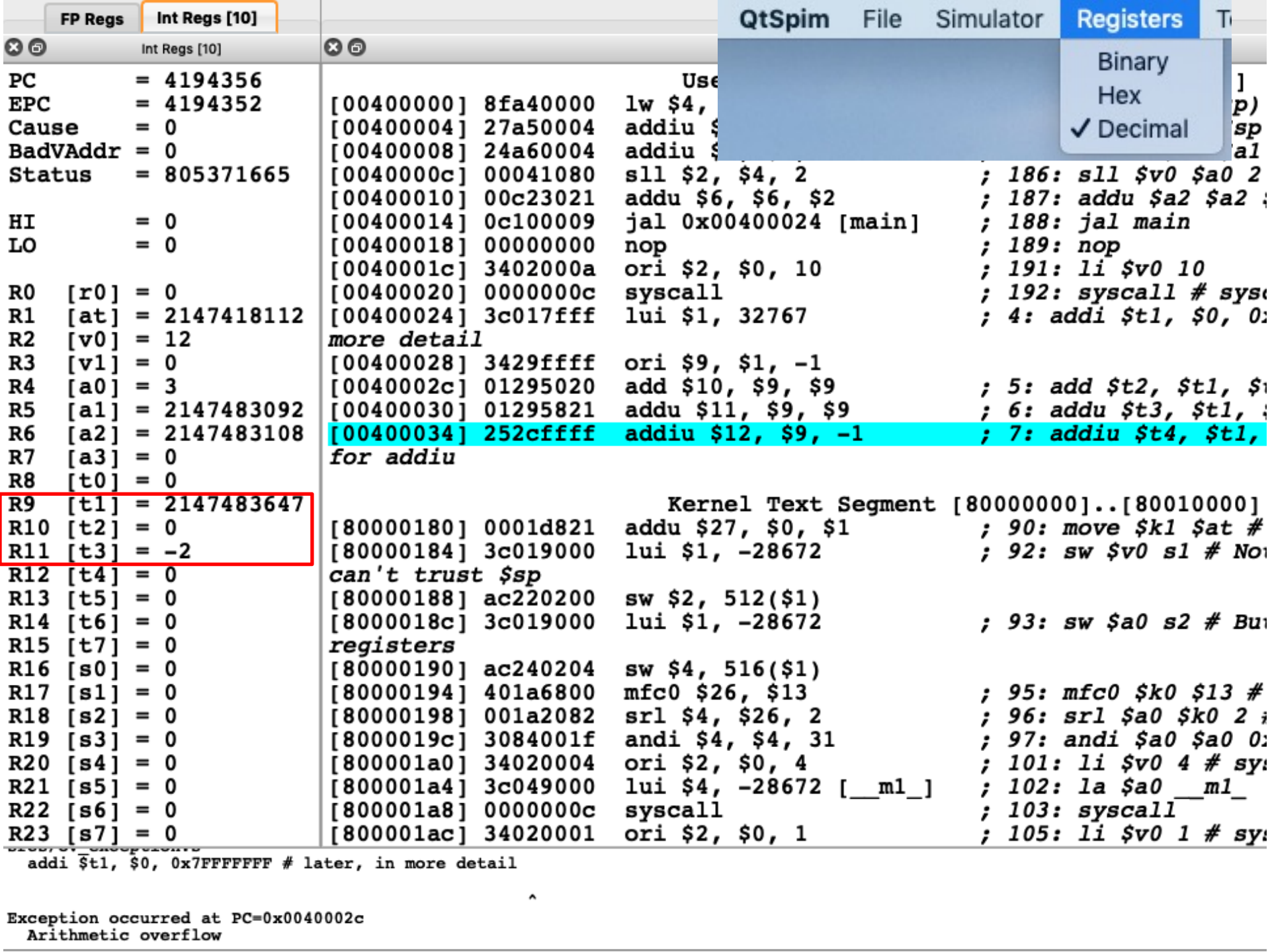
MIPS source file

```
.text
.globl main
main:
    addi $t1, $0, 0x7FFFFFFF # later, in more detail
    add  $t2, $t1, $t1 # overflow
    addu $t3, $t1, $t1 # no exception
    addiu $t4, $t1, -1 # negative constant for addiu
```




FP Regs
Int Regs [16]
Data
Text
Int Regs [16]
Text
PC = 400030
EPC = 400030
Cause = 0
BadVAddr = 0
Status = 3000ff11
HI = 0
LO = 0
R0 [r0] = 0
R1 [at] = 7fff0000
R2 [v0] = c
R3 [v1] = 0
R4 [a0] = 3
R5 [a1] = 7ffffdd4
R6 [a2] = 7ffffde4
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 7fffffff
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp)
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # sys
[00400024] 3c017fff lui $1, 32767 ; 4: addi $t1, $0, 0
more detail
[00400028] 3429ffff ori $9, $1, -1
[0040002c] 01295020 add $10, $9, $9 ; 5: add $t2, $t1, $
[00400030] 01295821 addu $11, $9, $9 ; 6: addu $t3, $t1,
[00400034] 252cffff addiu $12, $9, -1 ; 7: addiu $t4, $t1,
for addiu
Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821 addu $27, $0, $1 ; 90: move $k1 $at #
[80000184] 3c019000 lui $1, -28672 ; 92: sw $v0 s1 # No
can't trust $sp
[80000188] ac220200 sw $2, 512($1)
[8000018c] 3c019000 lui $1, -28672 ; 93: sw $a0 s2 # Bu
registers
[80000190] ac240204 sw $4, 516($1)
[80000194] 401a6800 mfc0 $26, $13 ; 95: mfc0 $k0 $13 #
[80000198] 001a2082 srl $4, $26, 2 ; 96: srl $a0 $k0 2
[8000019c] 3084001f andi $4, $4, 31 ; 97: andi $a0 $a0 0
[800001a0] 34020004 ori $2, $0, 4 ; 101: li $v0 4 # sy
[800001a4] 3c049000 lui $4, -28672 [__m1_] ; 102: la $a0 __m1_
[800001a8] 0000000c syscall ; 103: syscall
[800001ac] 34020001 ori $2, $0, 1 ; 105: li $v0 1 # sy
addi $t1, $0, 0x7FFFFFFF # later, in more detail
Exception occurred at PC=0x0040002c
Arithmetic overflow

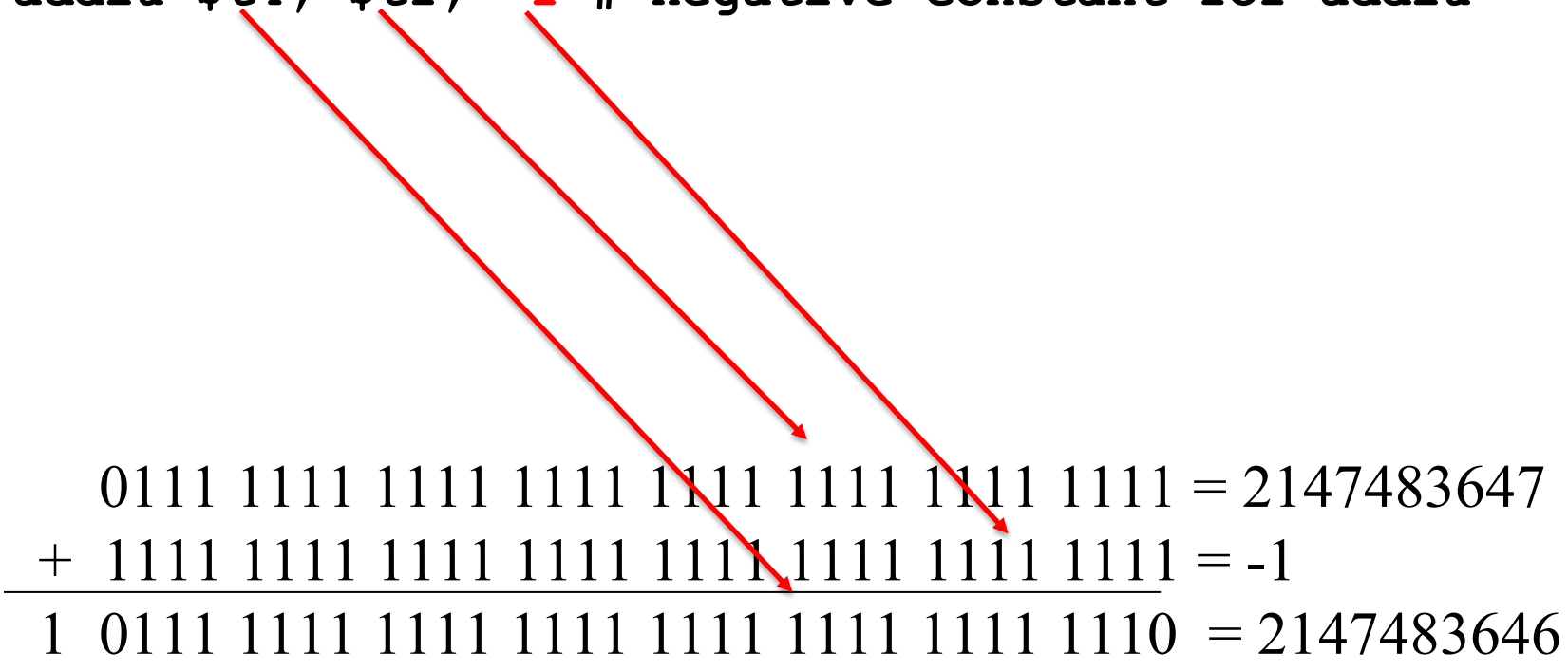

FP Regs
Int Regs [16]
Data
Text
Int Regs [16]
Text
PC = 400034
EPC = 400030
Cause = 0
BadVAddr = 0
Status = 3000ff11
HI = 0
LO = 0
R0 [r0] = 0
R1 [at] = 7fff0000
R2 [v0] = c
R3 [v1] = 0
R4 [a0] = 3
R5 [a1] = 7ffffdd4
R6 [a2] = 7ffffde4
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 7fffffff
R10 [t2] = 0
R11 [t3] = fffffffe
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
+42억 or -2?
User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp)
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # sys
[00400024] 3c017fff lui $1, 32767 ; 4: addi $t1, $0, 0
more detail
[00400028] 3429ffff ori $9, $1, -1
[0040002c] 01295020 add $10, $9, $9 ; 5: add $t2, $t1, $t
[00400030] 01295821 addu $11, $9, $9 ; 6: addu $t3, $t1,
[00400034] 252cffff addiu $12, $9, -1 ; 7: addiu $t4, $t1,
for addiu
Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821 addu $27, $0, $1 ; 90: move $k1 $at #
[80000184] 3c019000 lui $1, -28672 ; 92: sw $v0 s1 # No
can't trust $sp
[80000188] ac220200 sw $2, 512($1)
[8000018c] 3c019000 lui $1, -28672 ; 93: sw $a0 s2 # Bu
registers
[80000190] ac240204 sw $4, 516($1)
[80000194] 401a6800 mfc0 $26, $13 ; 95: mfc0 $k0 $13 #
[80000198] 001a2082 srl $4, $26, 2 ; 96: srl $a0 $k0 2
[8000019c] 3084001f andi $4, $4, 31 ; 97: andi $a0 $a0 0
[800001a0] 34020004 ori $2, $0, 4 ; 101: li $v0 4 # sy
[800001a4] 3c049000 lui $4, -28672 [__m1_] ; 102: la $a0 __m1_
[800001a8] 0000000c syscall ; 103: syscall
[800001ac] 34020001 ori $2, $0, 1 ; 105: li $v0 1 # sy
addi $t1, $0, 0x7FFFFFFF # later, in more detail
Exception occurred at PC=0x0040002c
Arithmetic overflow

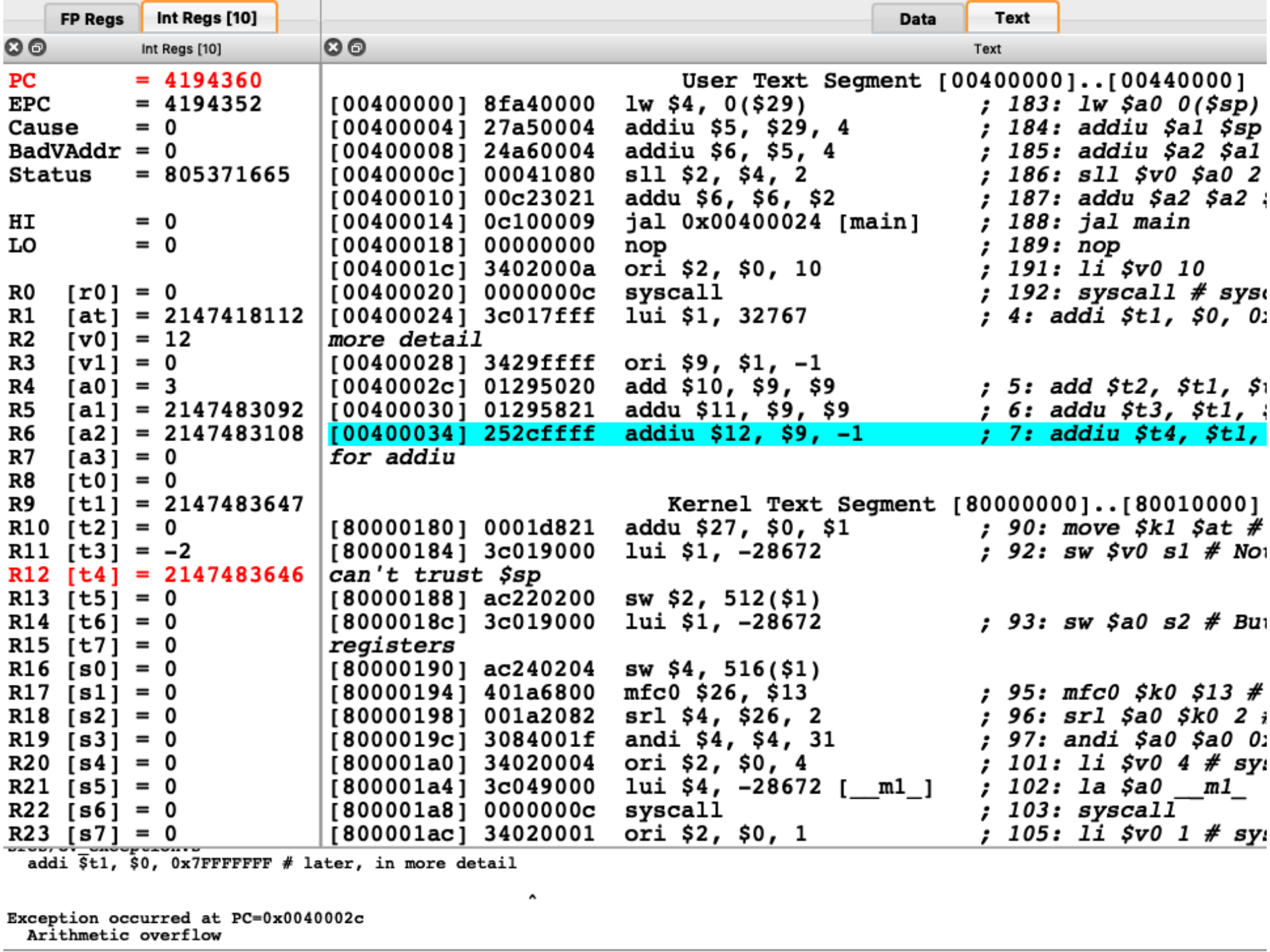

FP Regs
Int Regs [10]
Int Regs [10]
QtSpim
File
Simulator
Registers
Binary
Hex
✓ Decimal
PC = 4194356
EPC = 4194352
Cause = 0
BadVAddr = 0
Status = 805371665
HI = 0
LO = 0
R0 [r0] = 0
R1 [at] = 2147418112
R2 [v0] = 12
R3 [v1] = 0
R4 [a0] = 3
R5 [a1] = 2147483092
R6 [a2] = 2147483108
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 2147483647
R10 [t2] = 0
R11 [t3] = -2
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
[00400000] 8fa40000 lw $4,
[00400004] 27a50004 addiu $
[00400008] 24a60004 addiu $
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # sys
[00400024] 3c017fff lui $1, 32767 ; 4: addi $t1, $0, 0
more detail
[00400028] 3429ffff ori $9, $1, -1
[0040002c] 01295020 add $10, $9, $9 ; 5: add $t2, $t1, $t
[00400030] 01295821 addu $11, $9, $9 ; 6: addu $t3, $t1,
[00400034] 252cffff addiu $12, $9, -1 ; 7: addiu $t4, $t1,
for addiu
Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821 addu $27, $0, $1 ; 90: move $k1 $at #
[80000184] 3c019000 lui $1, -28672 ; 92: sw $v0 s1 # No
can't trust $sp
[80000188] ac220200 sw $2, 512($1)
[8000018c] 3c019000 lui $1, -28672 ; 93: sw $a0 s2 # Bu
registers
[80000190] ac240204 sw $4, 516($1)
[80000194] 401a6800 mfc0 $26, $13 ; 95: mfc0 $k0 $13 #
[80000198] 001a2082 srl $4, $26, 2 ; 96: srl $a0 $k0 2
[8000019c] 3084001f andi $4, $4, 31 ; 97: andi $a0 $a0 0
[800001a0] 34020004 ori $2, $0, 4 ; 101: li $v0 4 # sy
[800001a4] 3c049000 lui $4, -28672 [__m1_] ; 102: la $a0 __m1_
[800001a8] 0000000c syscall ; 103: syscall
[800001ac] 34020001 ori $2, $0, 1 ; 105: li $v0 1 # sy
addi $t1, $0, 0x7FFFFFFF # later, in more detail
Exception occurred at PC=0x0040002c
Arithmetic overflow

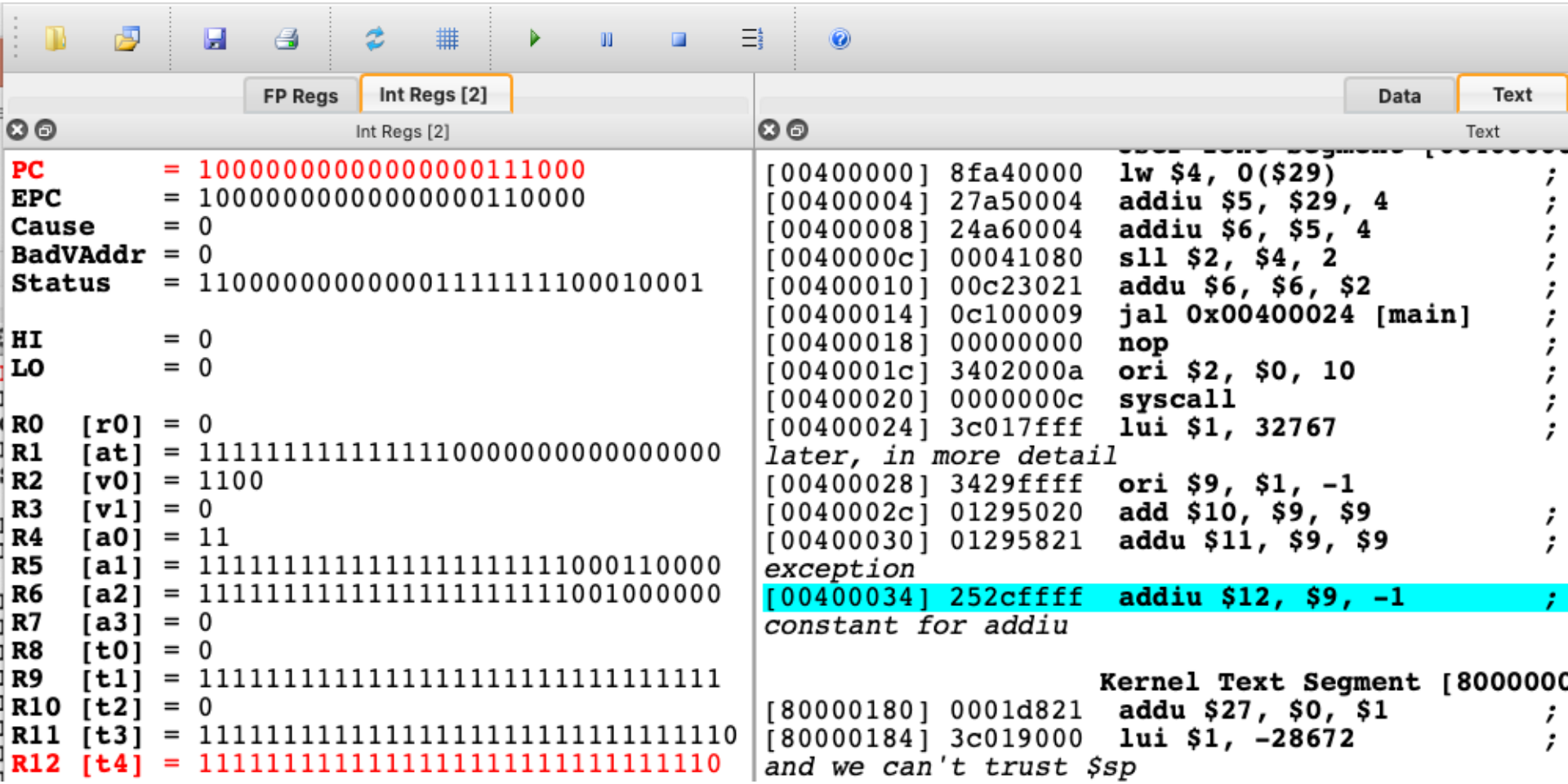
addiu instruction

- addiu 명령의 상수 operand 는 음수일 수 있다.

```
addiu $t4, $t1, -1 # negative constant for addiu
```




FP Regs
Int Regs [10]
Data
Text
Int Regs [10]
Text
PC = 4194360
EPC = 4194352
Cause = 0
BadVAddr = 0
Status = 805371665
HI = 0
LO = 0
R0 [r0] = 0
R1 [at] = 2147418112
R2 [v0] = 12
R3 [v1] = 0
R4 [a0] = 3
R5 [a1] = 2147483092
R6 [a2] = 2147483108
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 2147483647
R10 [t2] = 0
R11 [t3] = -2
R12 [t4] = 2147483646
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp)
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # sys
[00400024] 3c017fff lui $1, 32767 ; 4: addi $t1, $0, 0
more detail
[00400028] 3429ffff ori $9, $1, -1
[0040002c] 01295020 add $10, $9, $9 ; 5: add $t2, $t1, $t
[00400030] 01295821 addu $11, $9, $9 ; 6: addu $t3, $t1,
[00400034] 252cffff addiu $12, $9, -1 ; 7: addiu $t4, $t1,
for addiu
Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821 addu $27, $0, $1 ; 90: move $k1 $at #
[80000184] 3c019000 lui $1, -28672 ; 92: sw $v0 s1 # No
can't trust $sp
[80000188] ac220200 sw $2, 512($1)
[8000018c] 3c019000 lui $1, -28672 ; 93: sw $a0 s2 # Bu
registers
[80000190] ac240204 sw $4, 516($1)
[80000194] 401a6800 mfc0 $26, $13 ; 95: mfc0 $k0 $13 #
[80000198] 001a2082 srl $4, $26, 2 ; 96: srl $a0 $k0 2
[8000019c] 3084001f andi $4, $4, 31 ; 97: andi $a0 $a0 0
[800001a0] 34020004 ori $2, $0, 4 ; 101: li $v0 4 # sy
[800001a4] 3c049000 lui $4, -28672 [__m1_] ; 102: la $a0 __m1_
[800001a8] 0000000c syscall ; 103: syscall
[800001ac] 34020001 ori $2, $0, 1 ; 105: li $v0 1 # sy
addi $t1, $0, 0x7FFFFFFF # later, in more detail
Exception occurred at PC=0x0040002c
Arithmetic overflow


FP Regs
Int Regs [2]
Int Regs [2]
PC = 10000000000000000111000
EPC = 10000000000000000110000
Cause = 0
BadVAddr = 0
Status = 11000000000000111111100010001
HI = 0
LO = 0
R0 [r0] = 0
R1 [at] = 11111111111111100000000000000000
R2 [v0] = 1100
R3 [v1] = 0
R4 [a0] = 11
R5 [a1] = 11111111111111111111111000110000
R6 [a2] = 11111111111111111111111001000000
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 11111111111111111111111111111111
R10 [t2] = 0
R11 [t3] = 11111111111111111111111111111110
R12 [t4] = 11111111111111111111111111111110
Data
Text
Text
[00400000] 8fa40000 lw $4, 0($29) ;
[00400004] 27a50004 addiu $5, $29, 4 ;
[00400008] 24a60004 addiu $6, $5, 4 ;
[0040000c] 00041080 sll $2, $4, 2 ;
[00400010] 00c23021 addu $6, $6, $2 ;
[00400014] 0c100009 jal 0x00400024 [main] ;
[00400018] 00000000 nop ;
[0040001c] 3402000a ori $2, $0, 10 ;
[00400020] 0000000c syscall ;
[00400024] 3c017fff lui $1, 32767 ;
later, in more detail
[00400028] 3429ffff ori $9, $1, -1
[0040002c] 01295020 add $10, $9, $9 ;
[00400030] 01295821 addu $11, $9, $9 ;
exception
[00400034] 252cffff addiu $12, $9, -1 ;
constant for addiu
Kernel Text Segment [8000000
[80000180] 0001d821 addu $27, $0, $1 ;
[80000184] 3c019000 lui $1, -28672 ;
and we can't trust $sp

HW2: due 9/14 (목) 자정

- subu instruction 을 사용하여 0번 레지스터($0)에서 $t1을 뺄셈을 했을 때 오버플로우가 발생하도록 $t1 의 값을 초기화한 다음 뺄셈을 하는 MIPS assembly source code 를 작성한 다음, SPIM 에서 수행하여 작성한 코드가 load 된 text segment 부분과 연산 결과 레지스터 내용이 16진수로 함께 보이는 화면을 캡춰하여 ecampus 에 제출 (단 명령어 실행 시 overflow exception 이 발생하지 않도록 한다.)
- source register로는 $0 와 $t1 를 사용하고 destination register로 $t2를 사용한다.

Operation	Operand A	Operand B	Result indicating overflow
$A + B$	≥ 0	≥ 0	< 0
$A + B$	< 0	< 0	≥ 0
$A - B$	≥ 0	< 0	< 0
$A - B$	< 0	≥ 0	≥ 0

숙제 제출 시 주의할 점

마지막 화면 1개만 캡춰하여 제출할 것.

여러 개 화일을 압축하여 제출하지 마세요.

HW2 는 화일 첨부가 아니라 직접 작성으로 제출 방식을 바꾸었습니다.

그림 화일의 크기는 1M 미만으로.

HW2 제출 방법

hw2

• subu instruction 을 사용하여 0번 레지스터($0)에서 $t1을 뺄셈을 했을 때 오버플로우가 발생하도록 $t1 의 값을 초기화한 다음 뺄셈을 하는 MIPS assembly source code 를 작성한 다음, SPIM 에서 수행하여 작성한 코드가 load 된 text segment 부분과 연산 결과 레지스터 내용이 16진수로 함께 보이는 화면을 캡쳐하여 ecampus 에 제출 (단 명령어 실행 시 overflow exception 이 발생하지 않도록 한다.)

• source register로는 $0 와 $t1 를 사용하고 destination register로 $t2를 사용한다.

제출 상황

제출 여부	제출 안 함
채점 상황	채점되지 않음
종료 일시	2020-09-18 11:00
마감까지 남은 기한	8 일 11 시간
최종 수정 일시	-
제출물 설명	▶ 댓글 (0)

과제 제출하기

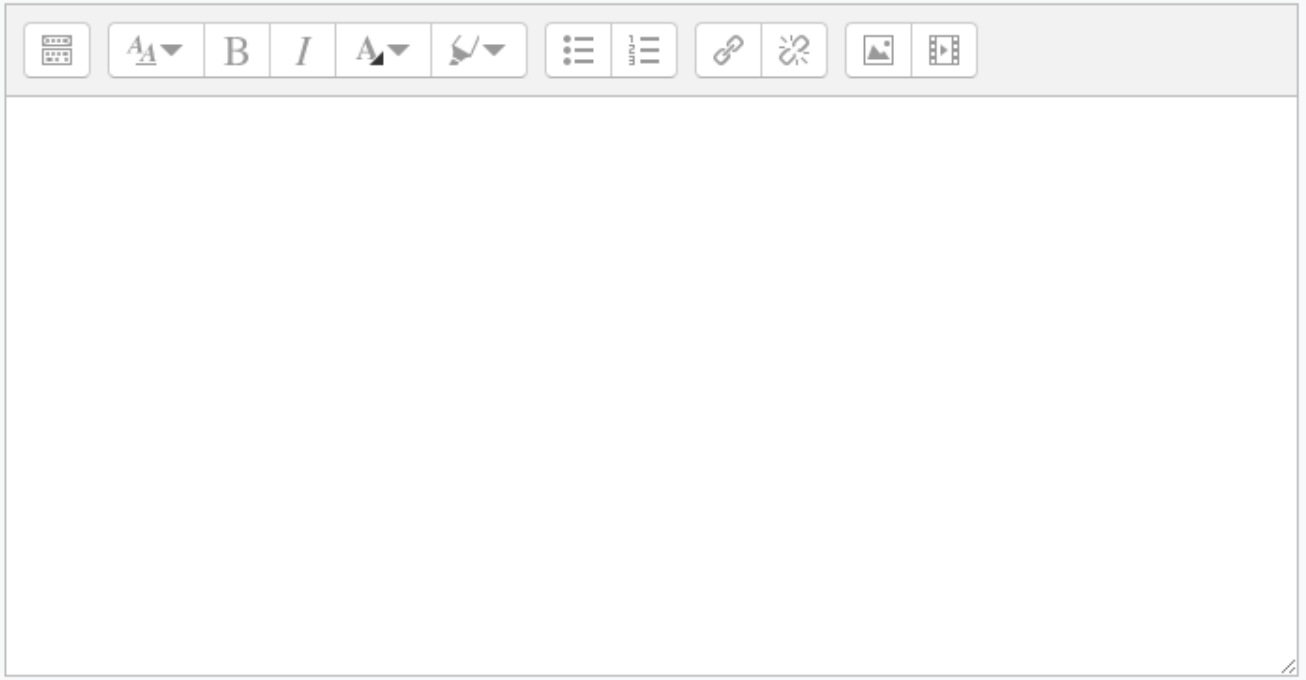
캡춰한 그림 화일을 drag-and-drop

hw2

• subu instruction 을 사용하여 0번 레지스터($0)에서 $t1을 뺄셈을 했을 때 오버플로우가 발생하도록 $t1 의 값을 초기화한 다음 뺄셈을 하는 MIPS assembly source code 를 작성한 다음, SPIM 에서 수행하여 작성한 코드가 load 된 text segment 부분과 연산 결과 레지스터 내용이 16진수로 함께 보이는 화면을 캡춰하여 ecampus 에 제출 (단 명령어 실행 시 overflow exception 이 발생하지 않도록 한다.)

• source register로는 $0 와 $t1 를 사용하고 destination register로 $t2를 사용한다.

직접 작성



저장 취소

[저장] 버튼을 클릭 시 최종 과제 제출일이 업데이트 되오니 주의하세요.