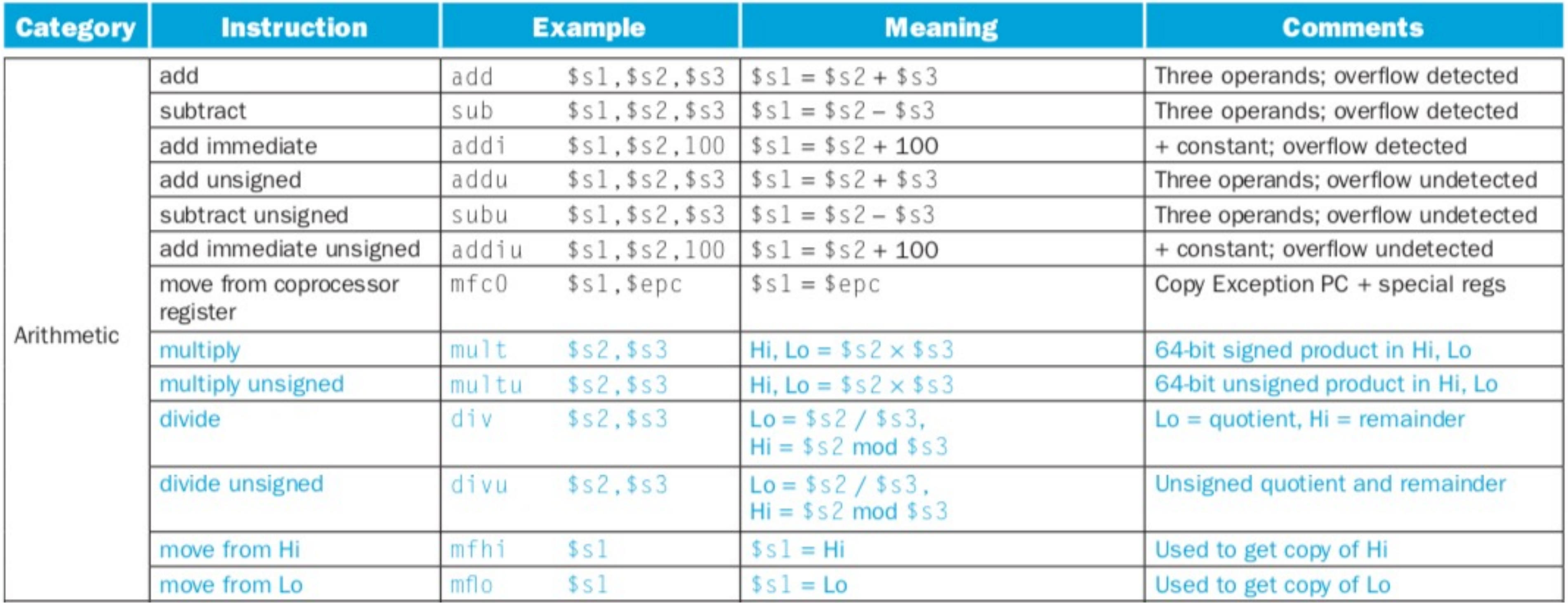
2.6 Logical Operations

MIPS Arithmetic Operations



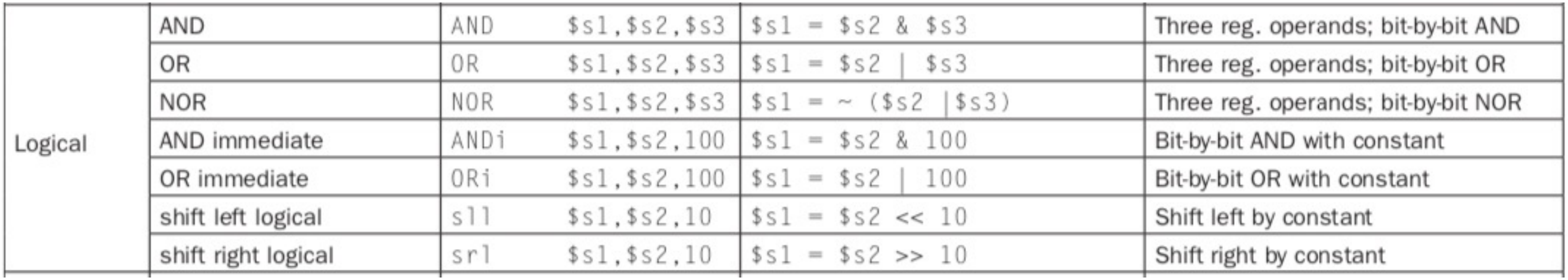
Category	Instruction	Example	Meaning	Comments
Arithmetic	add	add $s1,$s2,$s3	$s1 = $s2 + $s3	Three operands; overflow detected
	subtract	sub $s1,$s2,$s3	$s1 = $s2 – $s3	Three operands; overflow detected
	add immediate	addi $s1,$s2,100	$s1 = $s2 + 100	+ constant; overflow detected
	add unsigned	addu $s1,$s2,$s3	$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

2.6 MIPS logical operations



Logical	AND	AND $s1,$s2,$s3	$s1 = $s2 & $s3	Three reg. operands; bit-by-bit AND
	OR	OR $s1,$s2,$s3	$s1 = $s2 | $s3	Three reg. operands; bit-by-bit OR
	NOR	NOR $s1,$s2,$s3	$s1 = ~ ($s2 |$s3)	Three reg. operands; bit-by-bit NOR
	AND immediate	ANDi $s1,$s2,100	$s1 = $s2 & 100	Bit-by-bit AND with constant
	OR immediate	ORi $s1,$s2,100	$s1 = $s2 | 100	Bit-by-bit OR with constant
	shift left logical	sll $s1,$s2,10	$s1 = $s2 << 10	Shift left by constant
	shift right logical	srl $s1,$s2,10	$s1 = $s2 >> 10	Shift right by constant

논리 연산

논리 연산	NOT(논리부정)		AND(논리곱)			OR(논리합)		
수식	$X = A' = \bar{A} = \sim A$		$X = A \cdot B = A \wedge B = AB$			$X = A + B = A \vee B$		
진리표	A	X	A	B	X	A	B	X
	0	1	0	0	0	0	0	0
			0	1	0	0	1	1
	1	0	1	0	0	1	0	1
			1	1	1	1	1	1

- A : 1 은 짝수다 (거짓)
- B : 3 은 짝수다 (거짓)

논리 연산

논리 연산	NOT(논리부정)		AND(논리곱)			OR(논리합)		
수식	$X = A' = \bar{A} = \sim A$		$X = A \cdot B = A \wedge B = AB$			$X = A + B = A \vee B$		
진리표	A	X	A	B	X	A	B	X
	0	1	0	0	0	0	0	0
			0	1	0	0	1	1
	1	0	1	0	0	1	0	1
			1	1	1	1	1	1

- A : 1 은 짝수다 (거짓)
- B : 3 은 홀수다 (참)

논리 연산

논리 연산	NOT(논리부정)		AND(논리곱)			OR(논리합)		
수식	$X = A' = \bar{A} = \sim A$		$X = A \cdot B = A \wedge B = AB$			$X = A + B = A \vee B$		
진리표	A	X	A	B	X	A	B	X
	0	1	0	0	0	0	0	0
			0	1	0	0	1	1
	1	0	1	0	0	1	0	1
			1	1	1	1	1	1

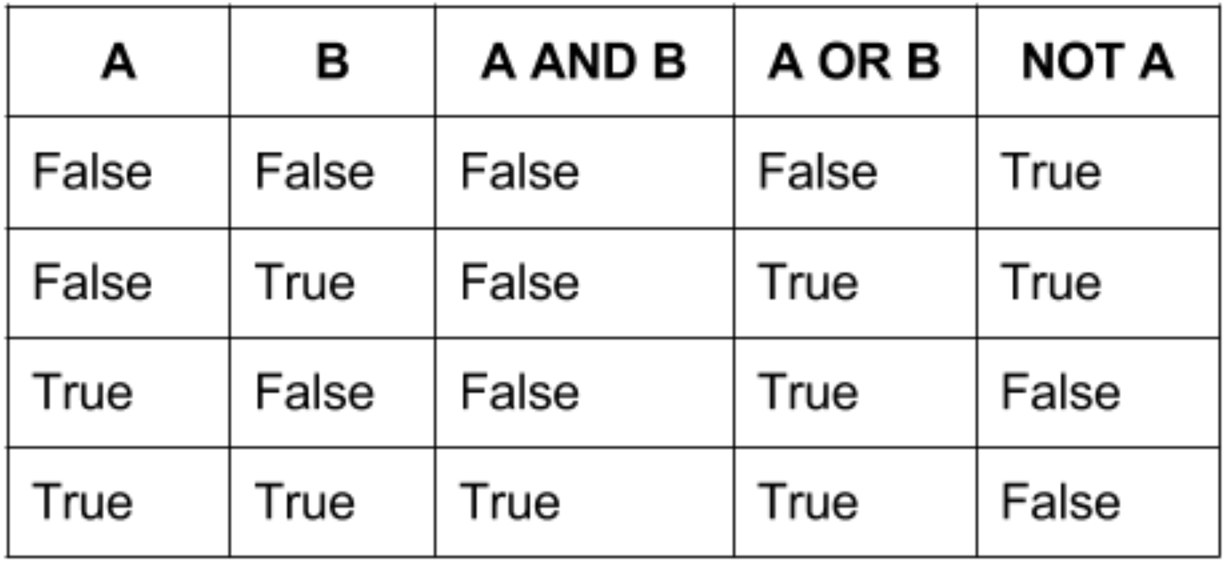
- A : 1 은 홀수다 (참)
- B : 3 은 짝수다 (거짓)

논리 연산

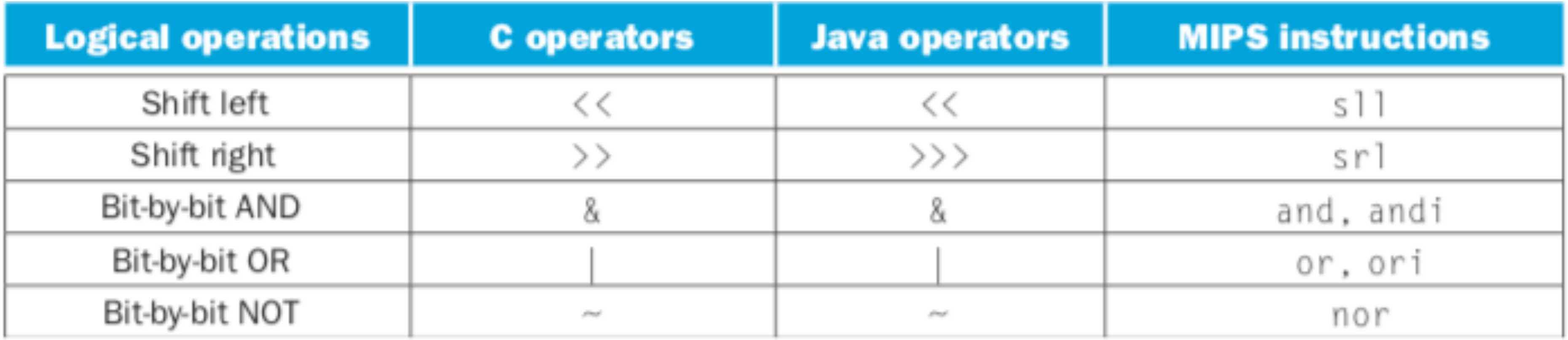
논리 연산	NOT(논리부정)		AND(논리곱)			OR(논리합)		
수식	$X = A' = \bar{A} = \sim A$		$X = A \cdot B = A \wedge B = AB$			$X = A + B = A \vee B$		
진리표	A	X	A	B	X	A	B	X
	0	1	0	0	0	0	0	0
			0	1	0	0	1	1
	1	0	1	0	0	1	0	1
			1	1	1	1	1	1

- A : 1 은 홀수다 (참)
- B : 3 은 홀수다 (참)

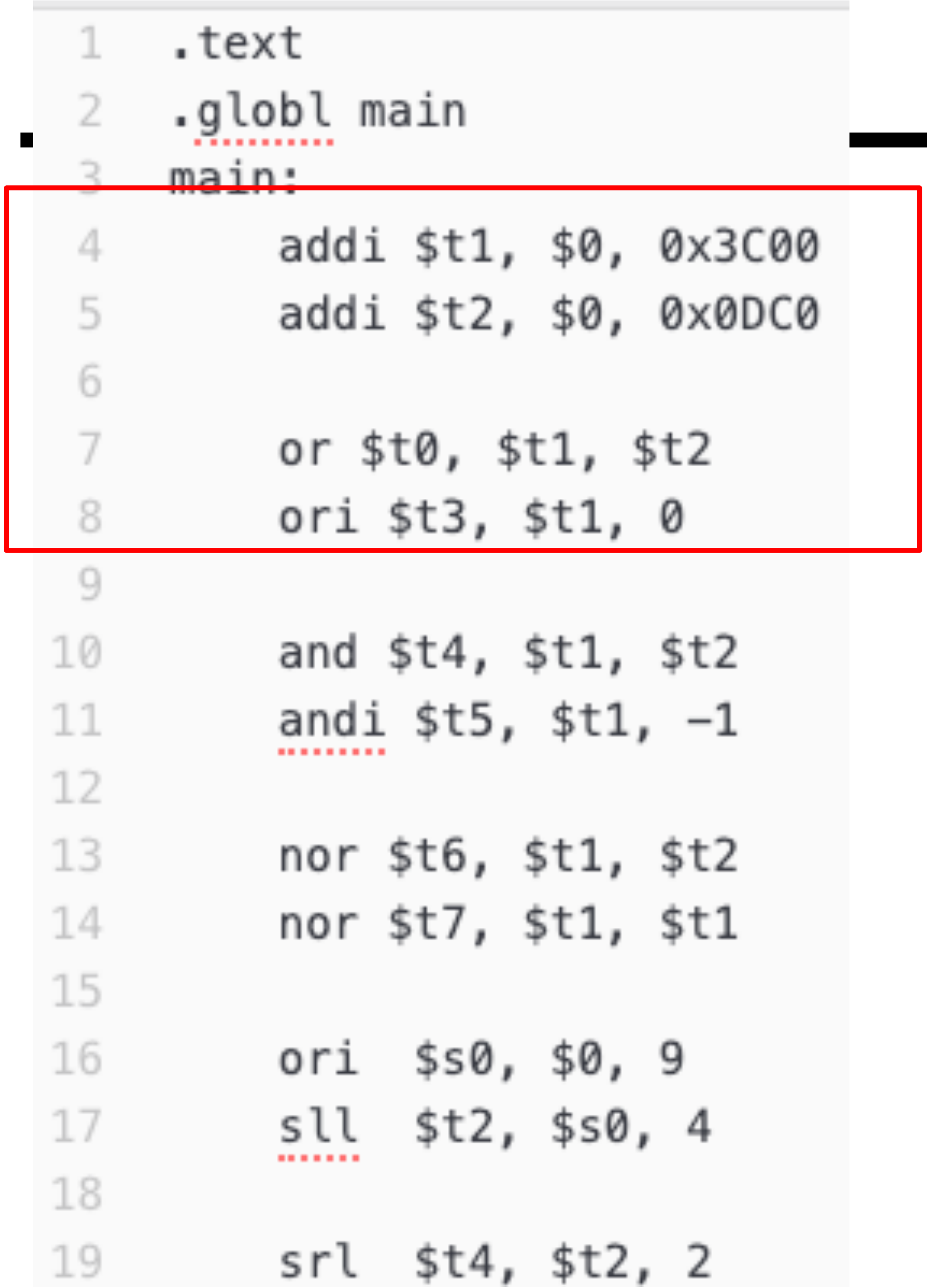
bitwise logical operators



A	B	A AND B	A OR B	NOT A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False



Logical operations	C operators	Java operators	MIPS instructions
Shift left	<<	<<	`sll`
Shift right	>>	>>>	`srl`
Bit-by-bit AND	&	&	`and`, `andi`
Bit-by-bit OR	|	|	`or`, `ori`
Bit-by-bit NOT	~	~	`nor`



```
.text
.globl main
main:
    addi $t1, $0, 0x3C00
    addi $t2, $0, 0x0DC0

    or $t0, $t1, $t2
    ori $t3, $t1, 0

    and $t4, $t1, $t2
    andi $t5, $t1, -1

    nor $t6, $t1, $t2
    nor $t7, $t1, $t1

    ori  $s0, $0, 9
    sll  $t2, $s0, 4

    srl  $t4, $t2, 2
```

or / ori

$t1 : 0000 0000 0000 0000 0011 1100 0000 0000$_{two}$

$t2 : 0000 0000 0000 0000 0000 1101 1100 0000$_{two}$

```
or  $t0, $t1, $t2
```

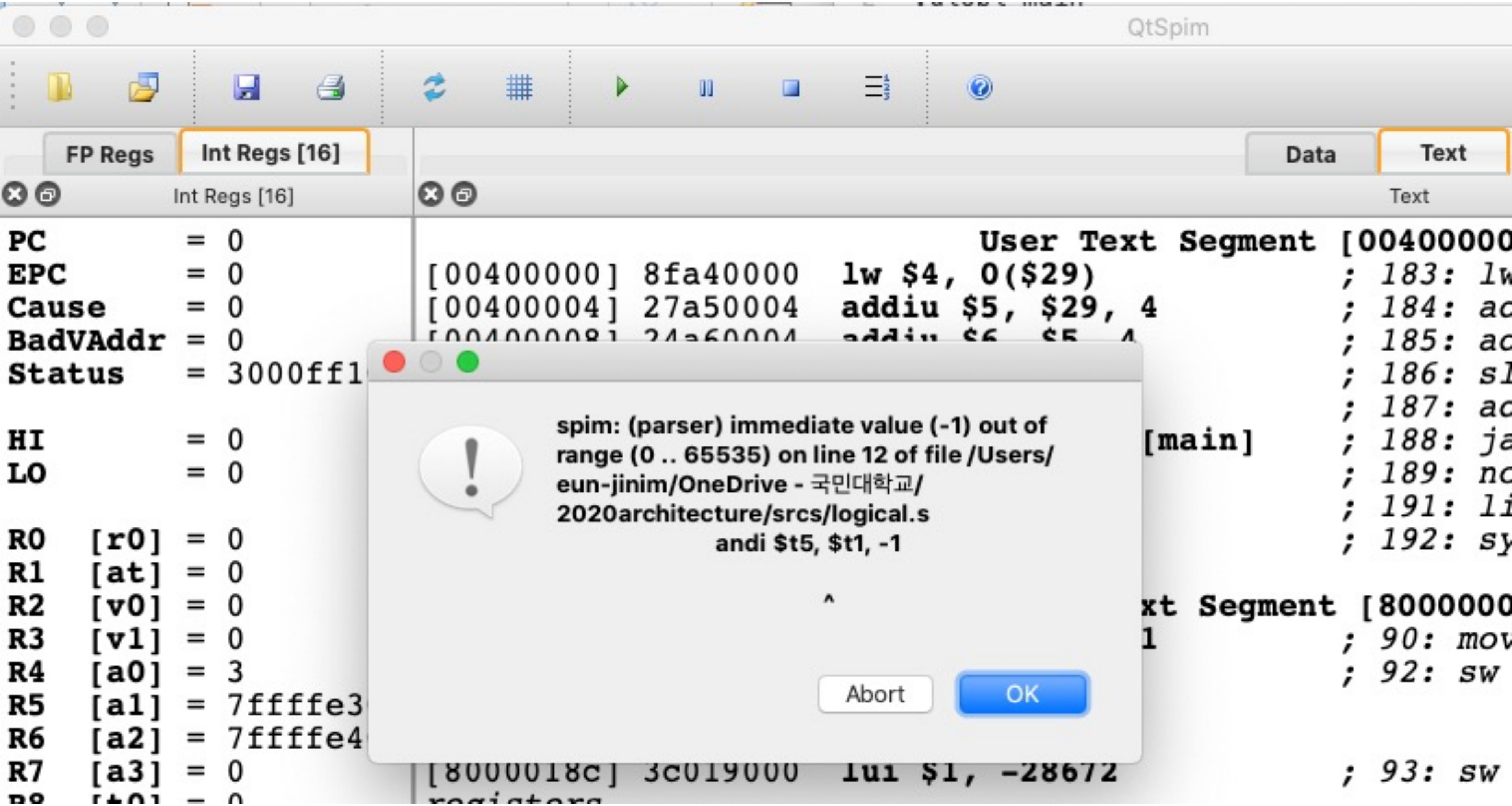
$t0 : 0000 0000 0000 0000 0011 1101 1100 0000$_{two}$

```
ori $t3, $t1, 0
```

A or 0 = A

```
      0000 0000 0000 0000 0011 1100 0000 0000
      0000 0000 0000 0000 0000 0000 0000 0000
      ---------------------------------------
$t3 : 0000 0000 0000 0000 0011 1100 0000 0000
```

당분간 out of range error는 무시하고 실행하세요.



```
PC        = 0
EPC       = 0
Cause     = 0
BadVAddr  = 0
Status    = 3000ff1

HI        = 0
LO        = 0

R0   [r0] = 0
R1   [at] = 0
R2   [v0] = 0
R3   [v1] = 0
R4   [a0] = 3
R5   [a1] = 7ffffe3
R6   [a2] = 7ffffe4
R7   [a3] = 0
R8   [t0] = 0
```

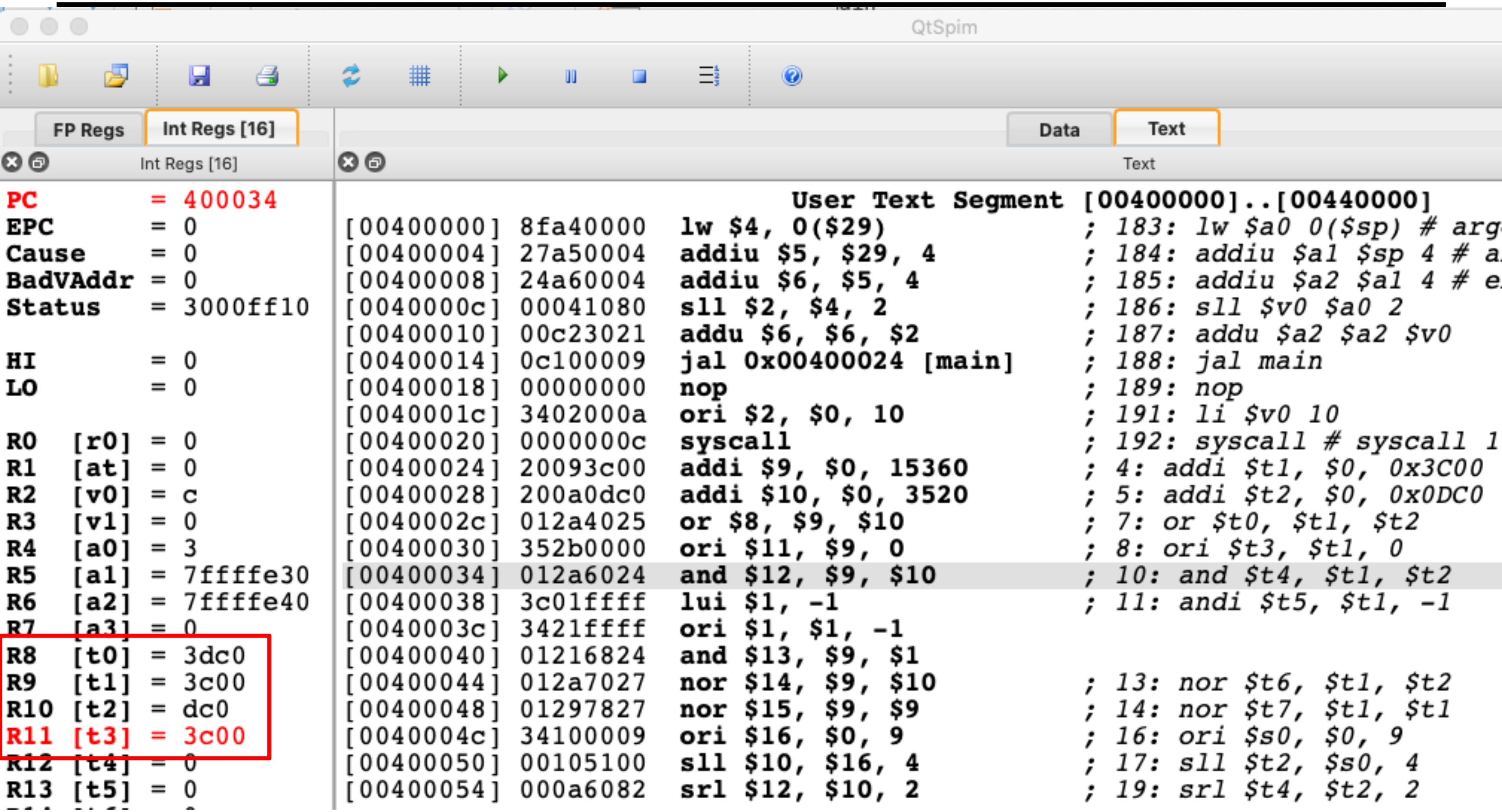
```
                              User Text Segment [00400000
[00400000] 8fa40000  lw $4, 0($29)            ; 183: lw
[00400004] 27a50004  addiu $5, $29, 4         ; 184: ad
[00400008] 24a60004  addiu $6, $5, 4          ; 185: ad
                                              ; 186: sl
                                              ; 187: ad
                                      [main]  ; 188: ja
                                              ; 189: no
                                              ; 191: li
                                              ; 192: sy

                                    xt Segment [8000000
                                    1         ; 90: mov
                                              ; 92: sw

[8000018c] 3c019000  lui $1, -28672           ; 93: sw
registers
```

spim: (parser) immediate value (-1) out of range (0 .. 65535) on line 12 of file /Users/eun-jinim/OneDrive - 국민대학교/2020architecture/srcs/logical.s

```
	andi $t5, $t1, -1
	                ^
```

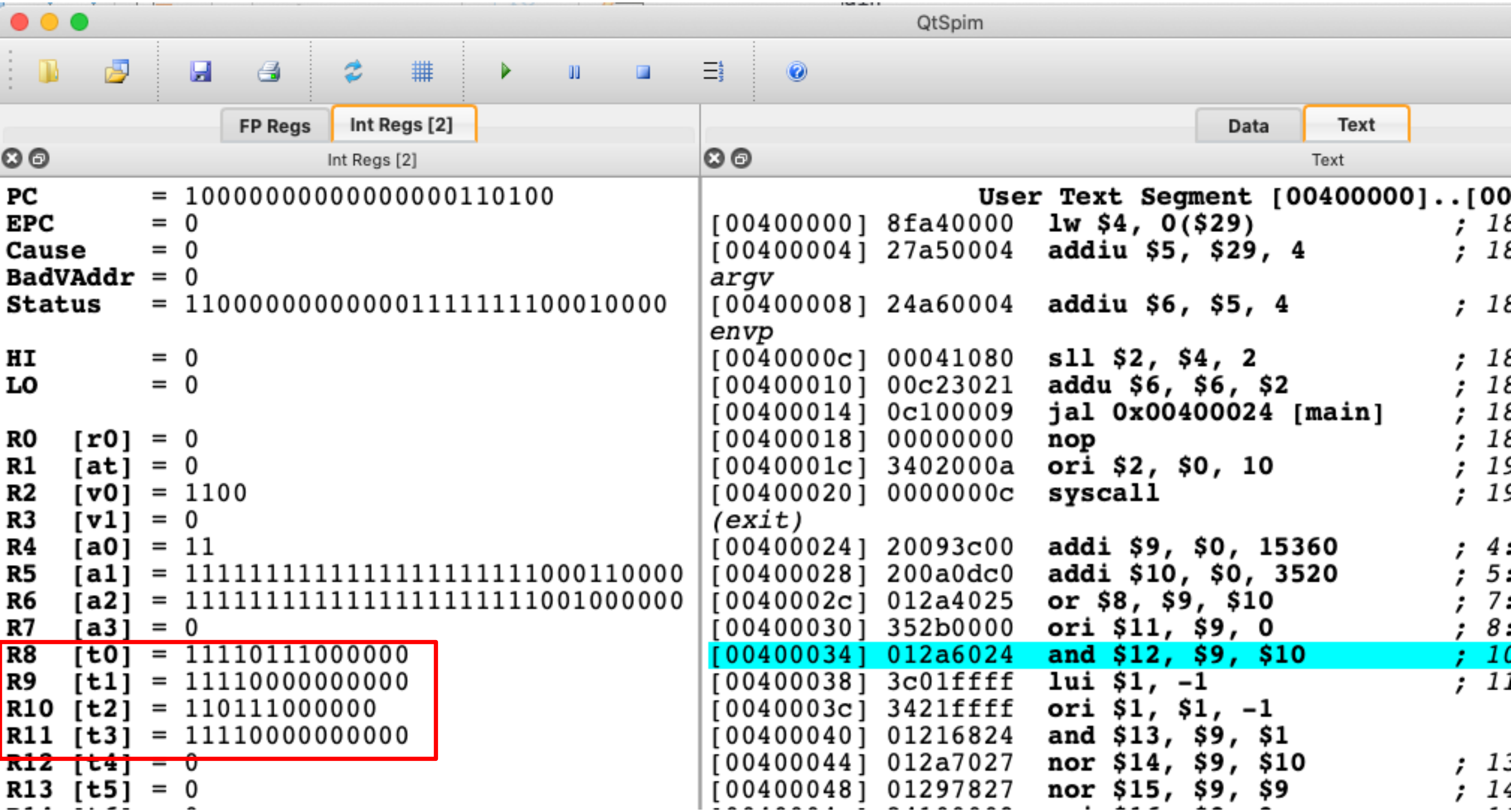


```
PC       = 400034
EPC      = 0
Cause    = 0
BadVAddr = 0
Status   = 3000ff10

HI       = 0
LO       = 0

R0   [r0] = 0
R1   [at] = 0
R2   [v0] = c
R3   [v1] = 0
R4   [a0] = 3
R5   [a1] = 7ffffe30
R6   [a2] = 7ffffe40
R7   [a3] = 0
R8   [t0] = 3dc0
R9   [t1] = 3c00
R10  [t2] = dc0
R11  [t3] = 3c00
R12  [t4] = 0
R13  [t5] = 0
R14  [t6] = 0
```

```
                          User Text Segment [00400000]..[00440000]
[00400000] 8fa40000  lw $4, 0($29)             ; 183: lw $a0 0($sp) # arg
[00400004] 27a50004  addiu $5, $29, 4          ; 184: addiu $a1 $sp 4 # a
[00400008] 24a60004  addiu $6, $5, 4           ; 185: addiu $a2 $a1 4 # e
[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 # syscall 1
[00400024] 20093c00  addi $9, $0, 15360        ; 4: addi $t1, $0, 0x3C00
[00400028] 200a0dc0  addi $10, $0, 3520        ; 5: addi $t2, $0, 0x0DC0
[0040002c] 012a4025  or $8, $9, $10            ; 7: or $t0, $t1, $t2
[00400030] 352b0000  ori $11, $9, 0            ; 8: ori $t3, $t1, 0
[00400034] 012a6024  and $12, $9, $10          ; 10: and $t4, $t1, $t2
[00400038] 3c01ffff  lui $1, -1                ; 11: andi $t5, $t1, -1
[0040003c] 3421ffff  ori $1, $1, -1
[00400040] 01216824  and $13, $9, $1
[00400044] 012a7027  nor $14, $9, $10          ; 13: nor $t6, $t1, $t2
[00400048] 01297827  nor $15, $9, $9           ; 14: nor $t7, $t1, $t1
[0040004c] 34100009  ori $16, $0, 9            ; 16: ori $s0, $0, 9
[00400050] 00105100  sll $10, $16, 4           ; 17: sll $t2, $s0, 4
[00400054] 000a6082  srl $12, $10, 2           ; 19: srl $t4, $t2, 2
```

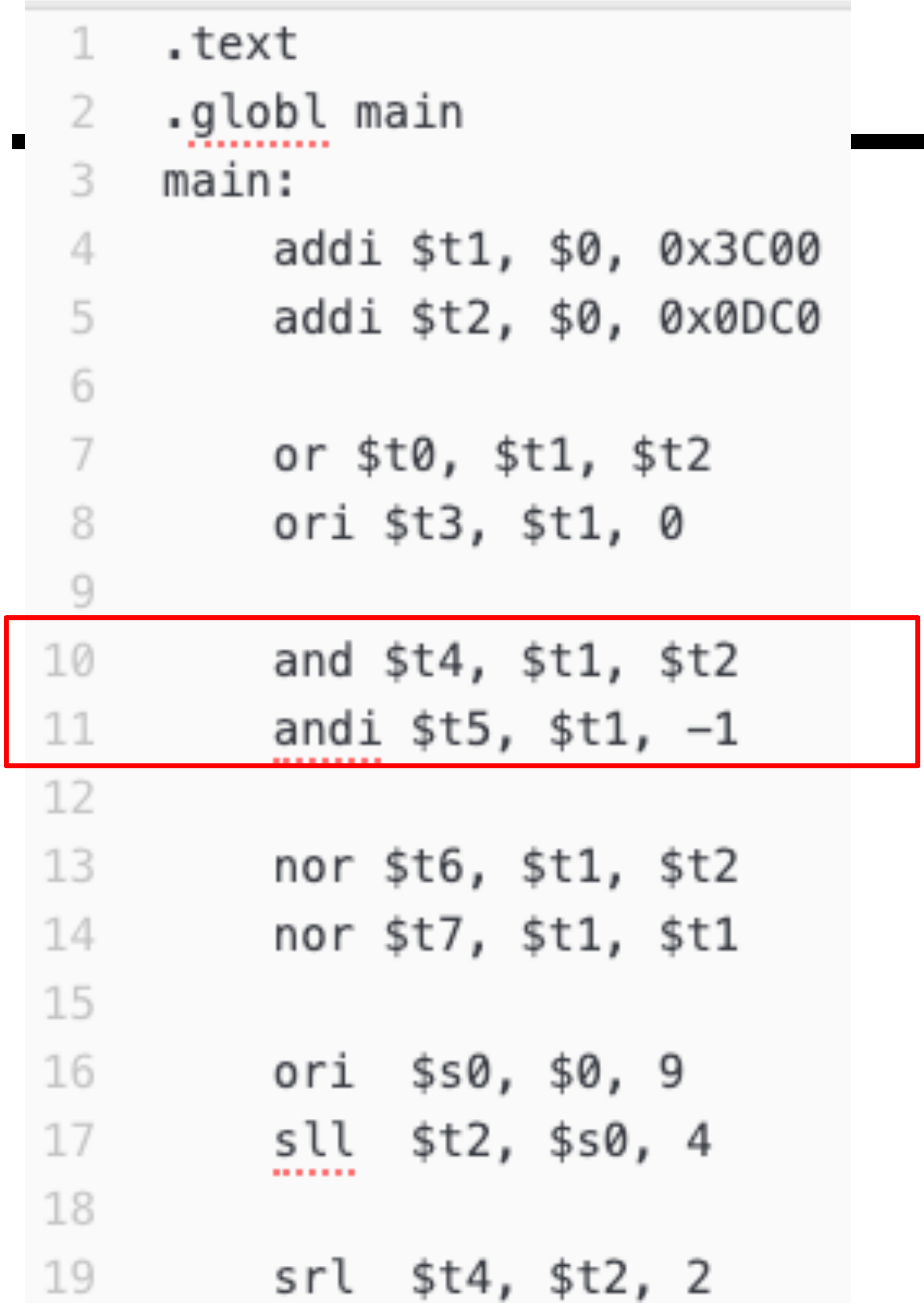


```
PC       = 1000000000000000000110100
EPC      = 0
Cause    = 0
BadVAddr = 0
Status   = 11000000000000011111111100010000

HI       = 0
LO       = 0

R0  [r0] = 0
R1  [at] = 0
R2  [v0] = 1100
R3  [v1] = 0
R4  [a0] = 11
R5  [a1] = 11111111111111111111111000110000
R6  [a2] = 11111111111111111111111001000000
R7  [a3] = 0
R8  [t0] = 11110111000000
R9  [t1] = 11110000000000
R10 [t2] = 110111000000
R11 [t3] = 11110000000000
R12 [t4] = 0
R13 [t5] = 0
```

```
                    User Text Segment [00400000]..[00
[00400000] 8fa40000  lw $4, 0($29)                ; 18
[00400004] 27a50004  addiu $5, $29, 4             ; 18
argv
[00400008] 24a60004  addiu $6, $5, 4              ; 18
envp
[0040000c] 00041080  sll $2, $4, 2                ; 18
[00400010] 00c23021  addu $6, $6, $2              ; 18
[00400014] 0c100009  jal 0x00400024 [main]        ; 18
[00400018] 00000000  nop                          ; 18
[0040001c] 3402000a  ori $2, $0, 10               ; 19
[00400020] 0000000c  syscall                      ; 19
(exit)
[00400024] 20093c00  addi $9, $0, 15360           ; 4:
[00400028] 200a0dc0  addi $10, $0, 3520           ; 5:
[0040002c] 012a4025  or $8, $9, $10               ; 7:
[00400030] 352b0000  ori $11, $9, 0               ; 8:
[00400034] 012a6024  and $12, $9, $10             ; 10
[00400038] 3c01ffff  lui $1, -1                   ; 11
[0040003c] 3421ffff  ori $1, $1, -1
[00400040] 01216824  and $13, $9, $1
[00400044] 012a7027  nor $14, $9, $10             ; 13
[00400048] 01297827  nor $15, $9, $9              ; 14
```



```
.text
.globl main
main:
    addi $t1, $0, 0x3C00
    addi $t2, $0, 0x0DC0

    or $t0, $t1, $t2
    ori $t3, $t1, 0

    and $t4, $t1, $t2
    andi $t5, $t1, -1

    nor $t6, $t1, $t2
    nor $t7, $t1, $t1

    ori  $s0, $0, 9
    sll  $t2, $s0, 4

    srl  $t4, $t2, 2
```

and / andi

$t1 : 0000 0000 0000 0000 0011 1100 0000 0000$_{two}$

$t2 : 0000 0000 0000 0000 0000 1101 1100 0000$_{two}$

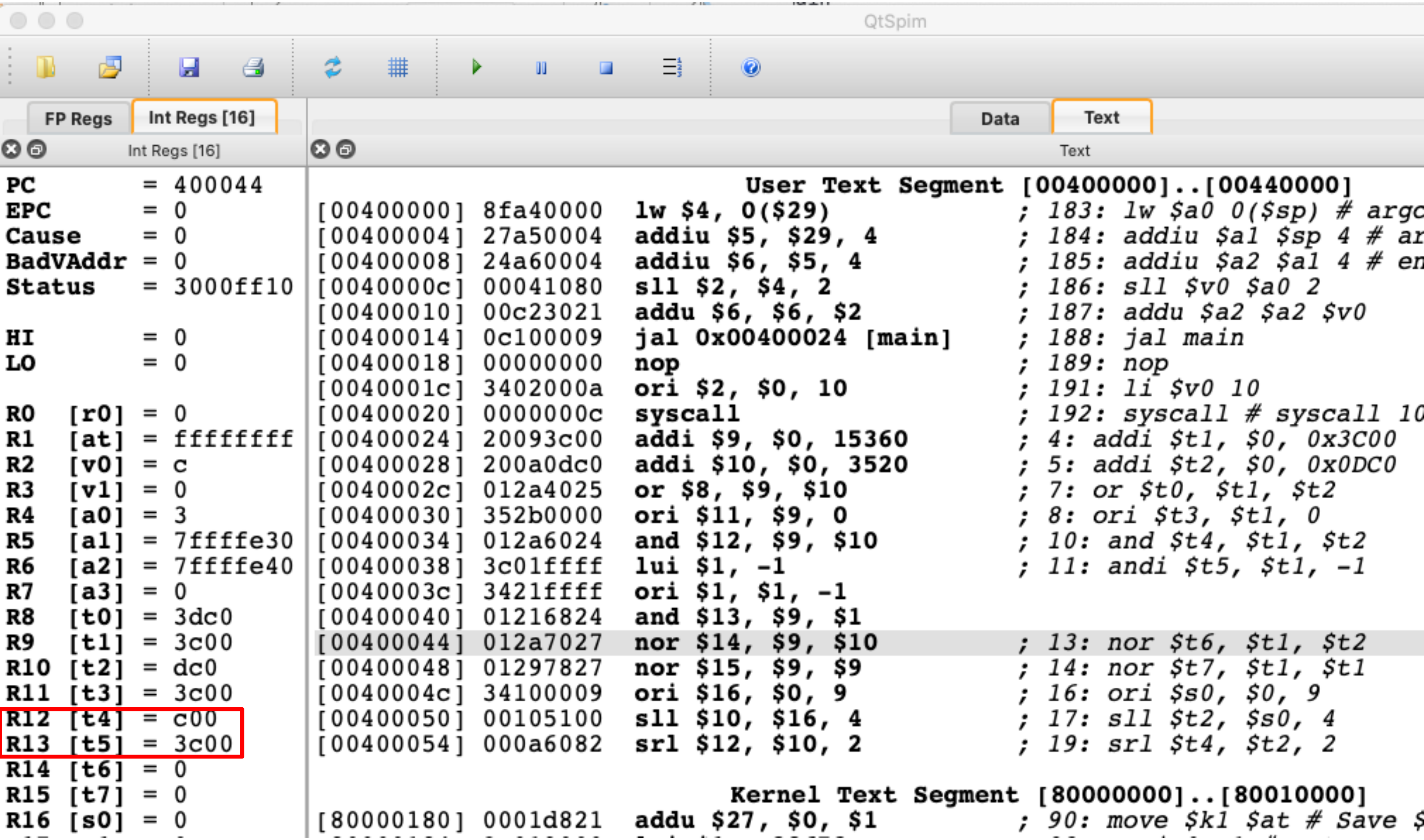
```
and $t4, $t1, $t2
```

$t4 : 0000 0000 0000 0000 0000 1100 0000 0000$_{two}$

```
andi $t5, $t1, -1
```

A and 1 = A

```
        0000 0000 0000 0000 0011 1100 0000 0000
        1111 1111 1111 1111 1111 1111 1111 1111
      ------------------------------------------
$t5 :   0000 0000 0000 0000 0011 1100 0000 0000
```



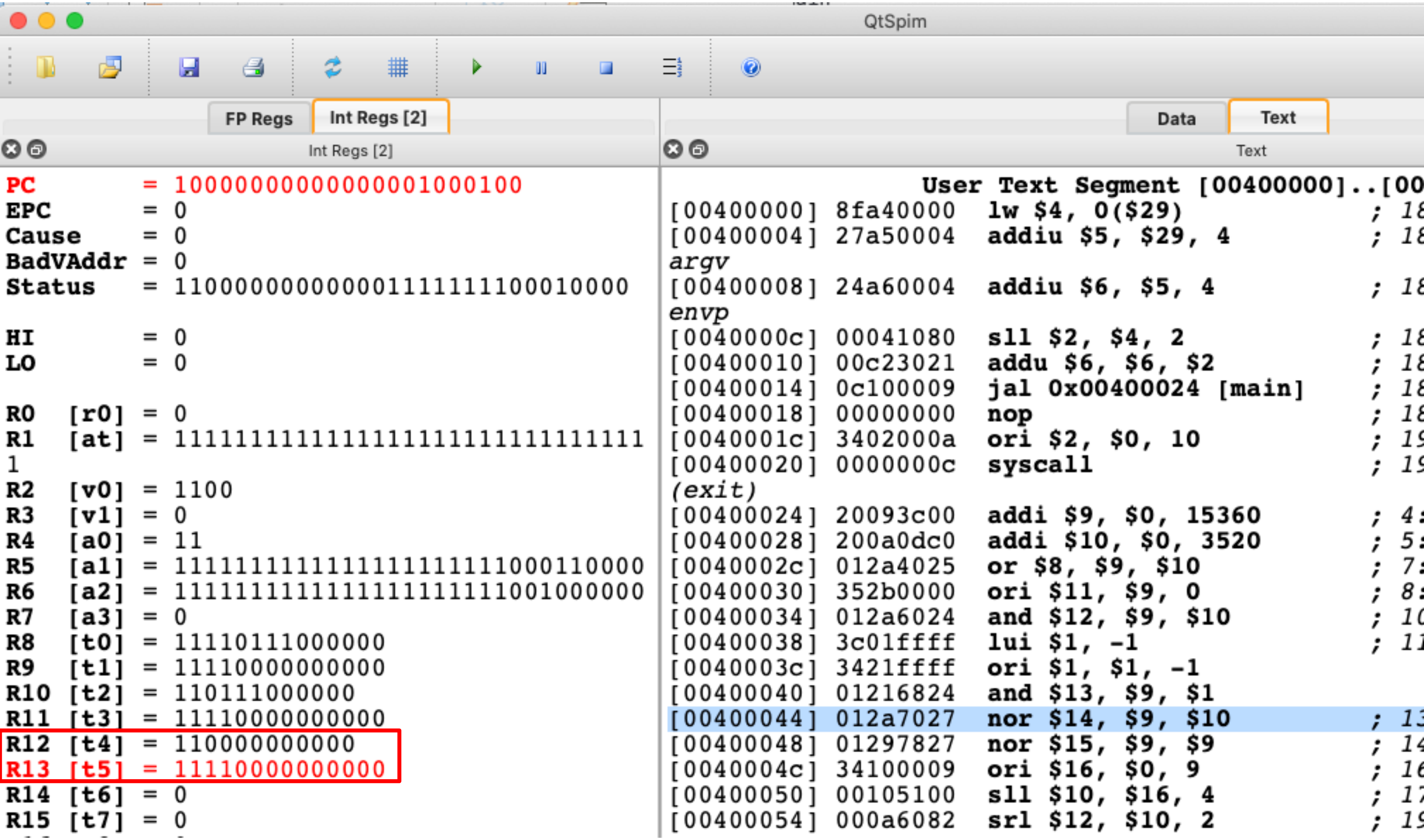
```
PC       = 400044
EPC      = 0
Cause    = 0
BadVAddr = 0
Status   = 3000ff10

HI       = 0
LO       = 0

R0  [r0] = 0
R1  [at] = ffffffff
R2  [v0] = c
R3  [v1] = 0
R4  [a0] = 3
R5  [a1] = 7ffffe30
R6  [a2] = 7ffffe40
R7  [a3] = 0
R8  [t0] = 3dc0
R9  [t1] = 3c00
R10 [t2] = dc0
R11 [t3] = 3c00
R12 [t4] = c00
R13 [t5] = 3c00
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
```

```
                                    User Text Segment [00400000]..[00440000]
[00400000] 8fa40000  lw $4, 0($29)            ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004  addiu $5, $29, 4         ; 184: addiu $a1 $sp 4 # ar
[00400008] 24a60004  addiu $6, $5, 4          ; 185: addiu $a2 $a1 4 # en
[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 # syscall 10
[00400024] 20093c00  addi $9, $0, 15360       ; 4: addi $t1, $0, 0x3C00
[00400028] 200a0dc0  addi $10, $0, 3520       ; 5: addi $t2, $0, 0x0DC0
[0040002c] 012a4025  or $8, $9, $10           ; 7: or $t0, $t1, $t2
[00400030] 352b0000  ori $11, $9, 0           ; 8: ori $t3, $t1, 0
[00400034] 012a6024  and $12, $9, $10         ; 10: and $t4, $t1, $t2
[00400038] 3c01ffff  lui $1, -1               ; 11: andi $t5, $t1, -1
[0040003c] 3421ffff  ori $1, $1, -1
[00400040] 01216824  and $13, $9, $1
[00400044] 012a7027  nor $14, $9, $10         ; 13: nor $t6, $t1, $t2
[00400048] 01297827  nor $15, $9, $9          ; 14: nor $t7, $t1, $t1
[0040004c] 34100009  ori $16, $0, 9           ; 16: ori $s0, $0, 9
[00400050] 00105100  sll $10, $16, 4          ; 17: sll $t2, $s0, 4
[00400054] 000a6082  srl $12, $10, 2          ; 19: srl $t4, $t2, 2

                                    Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821  addu $27, $0, $1         ; 90: move $k1 $at # Save $
```

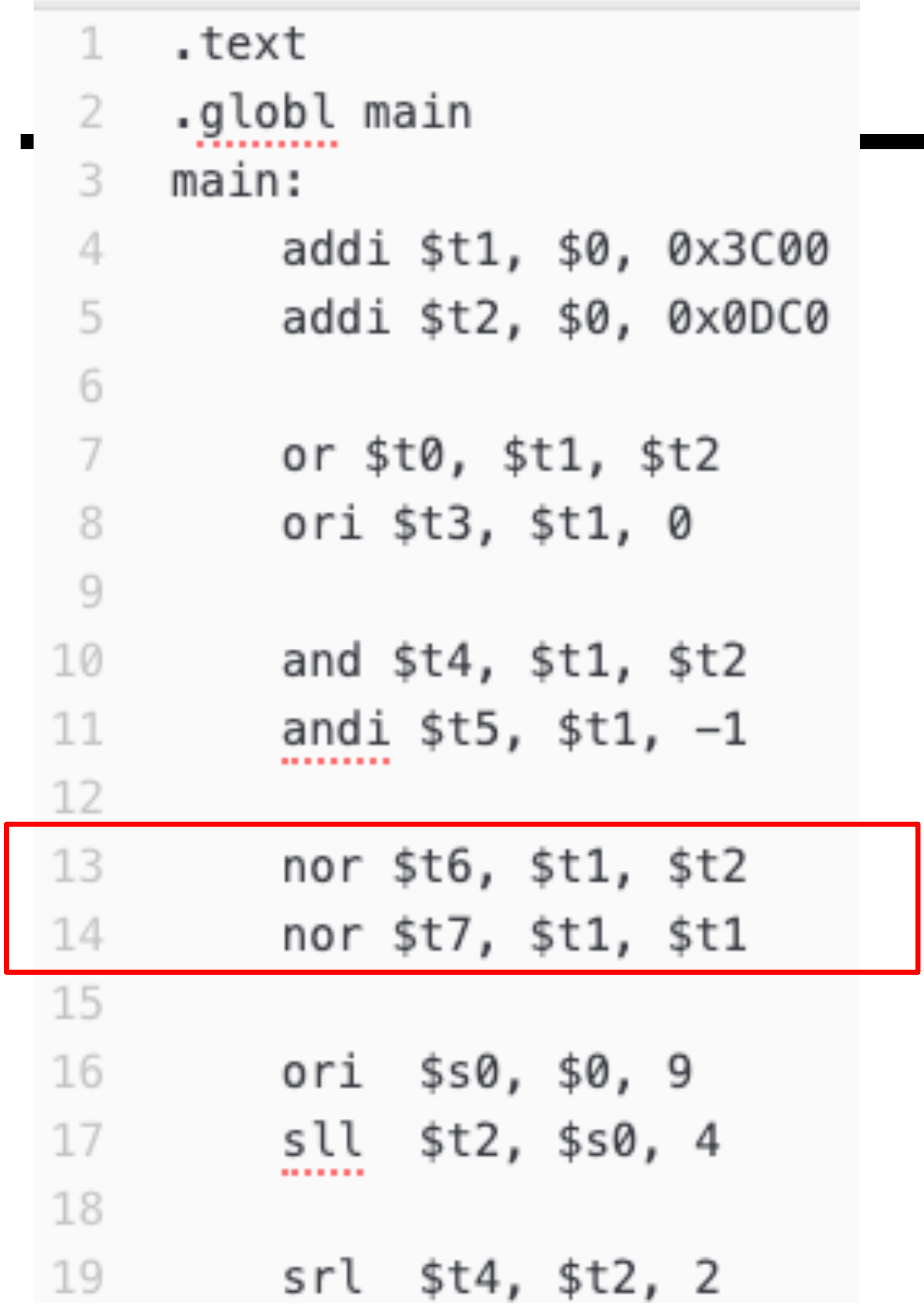


```
PC       = 10000000000000001000100
EPC      = 0
Cause    = 0
BadVAddr = 0
Status   = 110000000000001111111100010000

HI       = 0
LO       = 0

R0  [r0] = 0
R1  [at] = 1111111111111111111111111111111
1
R2  [v0] = 1100
R3  [v1] = 0
R4  [a0] = 11
R5  [a1] = 1111111111111111111111000110000
R6  [a2] = 1111111111111111111111001000000
R7  [a3] = 0
R8  [t0] = 11110111000000
R9  [t1] = 11110000000000
R10 [t2] = 110111000000
R11 [t3] = 11110000000000
R12 [t4] = 110000000000
R13 [t5] = 11110000000000
R14 [t6] = 0
R15 [t7] = 0
```

```
                    User Text Segment [00400000]..[00
[00400000] 8fa40000  lw $4, 0($29)                      ; 18
[00400004] 27a50004  addiu $5, $29, 4                   ; 18
argv
[00400008] 24a60004  addiu $6, $5, 4                    ; 18
envp
[0040000c] 00041080  sll $2, $4, 2                      ; 18
[00400010] 00c23021  addu $6, $6, $2                    ; 18
[00400014] 0c100009  jal 0x00400024 [main]              ; 18
[00400018] 00000000  nop                                ; 18
[0040001c] 3402000a  ori $2, $0, 10                     ; 19
[00400020] 0000000c  syscall                            ; 19
(exit)
[00400024] 20093c00  addi $9, $0, 15360                 ; 4:
[00400028] 200a0dc0  addi $10, $0, 3520                 ; 5:
[0040002c] 012a4025  or $8, $9, $10                     ; 7:
[00400030] 352b0000  ori $11, $9, 0                     ; 8:
[00400034] 012a6024  and $12, $9, $10                   ; 10
[00400038] 3c01ffff  lui $1, -1                         ; 11
[0040003c] 3421ffff  ori $1, $1, -1
[00400040] 01216824  and $13, $9, $1
[00400044] 012a7027  nor $14, $9, $10                   ; 13
[00400048] 01297827  nor $15, $9, $9                    ; 14
[0040004c] 34100009  ori $16, $0, 9                     ; 16
[00400050] 00105100  sll $10, $16, 4                    ; 17
[00400054] 000a6082  srl $12, $10, 2                    ; 19
```



```
.text
.globl main
main:
    addi $t1, $0, 0x3C00
    addi $t2, $0, 0x0DC0

    or $t0, $t1, $t2
    ori $t3, $t1, 0

    and $t4, $t1, $t2
    andi $t5, $t1, -1

    nor $t6, $t1, $t2
    nor $t7, $t1, $t1

    ori  $s0, $0, 9
    sll  $t2, $s0, 4

    srl  $t4, $t2, 2
```

nor = not or

$t1 : 0000 0000 0000 0000 0011 1100 0000 0000_{two}

$t2 : 0000 0000 0000 0000 0000 1101 1100 0000_{two}

```
nor $t6, $t1, $t2
```

$t6 : 1111 1111 1111 1111 1100 0010 0011 1111

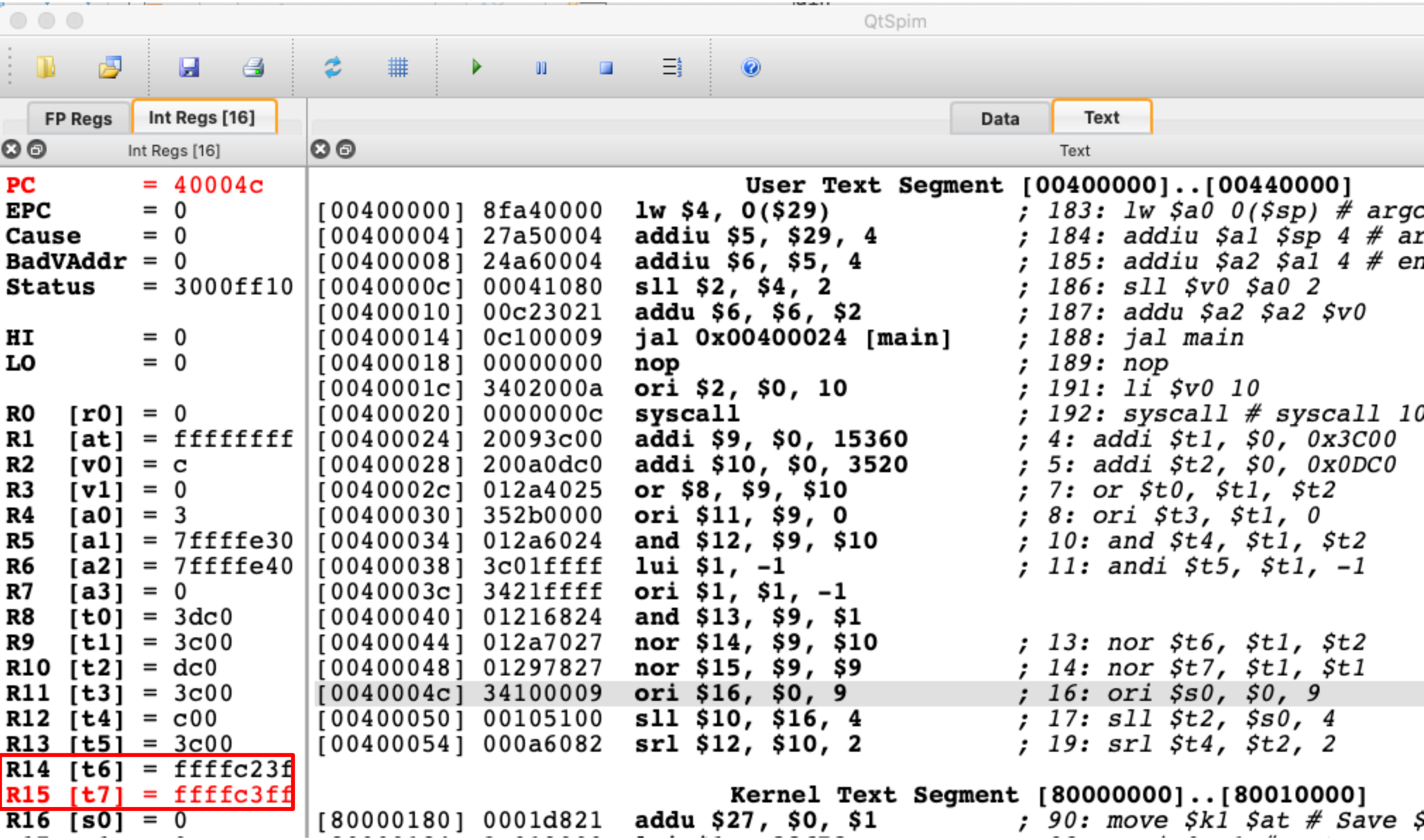
not(A) = not(A or A) = A nor A
= not (A or 0) = A nor 0

```
nor $t7, $t1, $t1 # $t7 = ~$t1
```

0000 0000 0000 0000 0011 1100 0000 0000

0000 0000 0000 0000 0011 1100 0000 0000

$t7 : 1111 1111 1111 1111 1100 0011 1111 1111



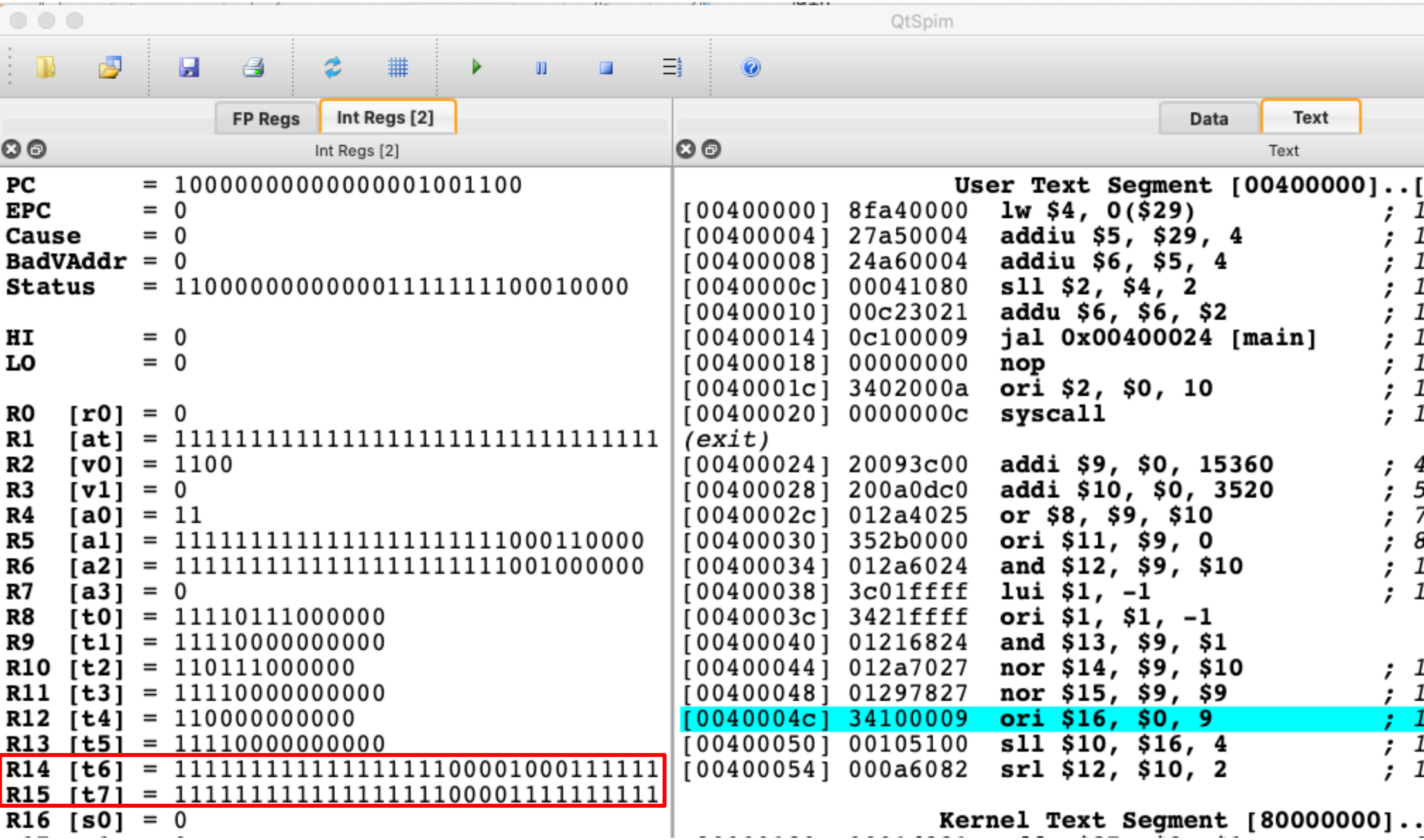
```
PC         = 40004c
EPC        = 0
Cause      = 0
BadVAddr   = 0
Status     = 3000ff10

HI         = 0
LO         = 0

R0   [r0] = 0
R1   [at] = ffffffff
R2   [v0] = c
R3   [v1] = 0
R4   [a0] = 3
R5   [a1] = 7ffffe30
R6   [a2] = 7ffffe40
R7   [a3] = 0
R8   [t0] = 3dc0
R9   [t1] = 3c00
R10  [t2] = dc0
R11  [t3] = 3c00
R12  [t4] = c00
R13  [t5] = 3c00
R14  [t6] = ffffc23f
R15  [t7] = ffffc3ff
R16  [s0] = 0
```

```
                              User Text Segment [00400000]..[00440000]
[00400000] 8fa40000  lw $4, 0($29)              ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004  addiu $5, $29, 4           ; 184: addiu $a1 $sp 4 # ar
[00400008] 24a60004  addiu $6, $5, 4            ; 185: addiu $a2 $a1 4 # en
[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 # syscall 10
[00400024] 20093c00  addi $9, $0, 15360         ; 4: addi $t1, $0, 0x3C00
[00400028] 200a0dc0  addi $10, $0, 3520         ; 5: addi $t2, $0, 0x0DC0
[0040002c] 012a4025  or $8, $9, $10             ; 7: or $t0, $t1, $t2
[00400030] 352b0000  ori $11, $9, 0             ; 8: ori $t3, $t1, 0
[00400034] 012a6024  and $12, $9, $10           ; 10: and $t4, $t1, $t2
[00400038] 3c01ffff  lui $1, -1                 ; 11: andi $t5, $t1, -1
[0040003c] 3421ffff  ori $1, $1, -1
[00400040] 01216824  and $13, $9, $1
[00400044] 012a7027  nor $14, $9, $10           ; 13: nor $t6, $t1, $t2
[00400048] 01297827  nor $15, $9, $9            ; 14: nor $t7, $t1, $t1
[0040004c] 34100009  ori $16, $0, 9             ; 16: ori $s0, $0, 9
[00400050] 00105100  sll $10, $16, 4            ; 17: sll $t2, $s0, 4
[00400054] 000a6082  srl $12, $10, 2            ; 19: srl $t4, $t2, 2

                              Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821  addu $27, $0, $1           ; 90: move $k1 $at # Save $
```



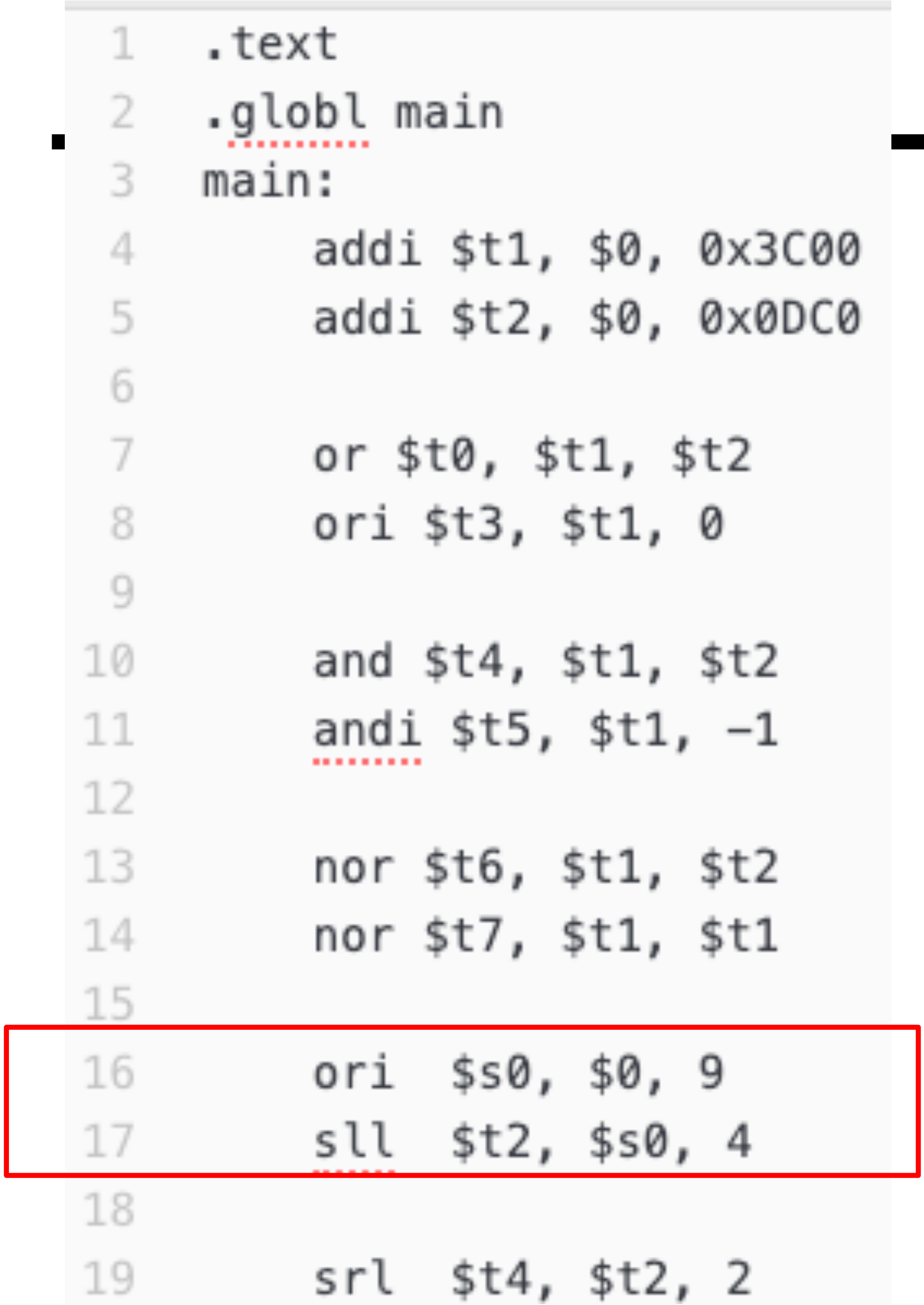
```
PC        = 10000000000000001001100
EPC       = 0
Cause     = 0
BadVAddr  = 0
Status    = 110000000000001111111100010000

HI        = 0
LO        = 0

R0  [r0] = 0
R1  [at] = 11111111111111111111111111111111
R2  [v0] = 1100
R3  [v1] = 0
R4  [a0] = 11
R5  [a1] = 1111111111111111111111000110000
R6  [a2] = 1111111111111111111111001000000
R7  [a3] = 0
R8  [t0] = 11110111000000
R9  [t1] = 11110000000000
R10 [t2] = 110111000000
R11 [t3] = 11110000000000
R12 [t4] = 110000000000
R13 [t5] = 11110000000000
R14 [t6] = 11111111111111111100001000111111
R15 [t7] = 11111111111111111100001111111111
R16 [s0] = 0
```

```
                          User Text Segment [00400000]..[
[00400000] 8fa40000  lw $4, 0($29)                ; 1
[00400004] 27a50004  addiu $5, $29, 4             ; 1
[00400008] 24a60004  addiu $6, $5, 4              ; 1
[0040000c] 00041080  sll $2, $4, 2                ; 1
[00400010] 00c23021  addu $6, $6, $2              ; 1
[00400014] 0c100009  jal 0x00400024 [main]        ; 1
[00400018] 00000000  nop                          ; 1
[0040001c] 3402000a  ori $2, $0, 10               ; 1
[00400020] 0000000c  syscall                      ; 1
(exit)
[00400024] 20093c00  addi $9, $0, 15360           ; 4
[00400028] 200a0dc0  addi $10, $0, 3520           ; 5
[0040002c] 012a4025  or $8, $9, $10               ; 7
[00400030] 352b0000  ori $11, $9, 0               ; 8
[00400034] 012a6024  and $12, $9, $10             ; 1
[00400038] 3c01ffff  lui $1, -1                   ; 1
[0040003c] 3421ffff  ori $1, $1, -1
[00400040] 01216824  and $13, $9, $1
[00400044] 012a7027  nor $14, $9, $10             ; 1
[00400048] 01297827  nor $15, $9, $9              ; 1
[0040004c] 34100009  ori $16, $0, 9               ; 1
[00400050] 00105100  sll $10, $16, 4              ; 1
[00400054] 000a6082  srl $12, $10, 2              ; 1

                        Kernel Text Segment [80000000]..
```



```
.text
.globl main
main:
    addi $t1, $0, 0x3C00
    addi $t2, $0, 0x0DC0

    or $t0, $t1, $t2
    ori $t3, $t1, 0

    and $t4, $t1, $t2
    andi $t5, $t1, -1

    nor $t6, $t1, $t2
    nor $t7, $t1, $t1

    ori  $s0, $0, 9
    sll  $t2, $s0, 4

    srl  $t4, $t2, 2
```

sll : shift left logical

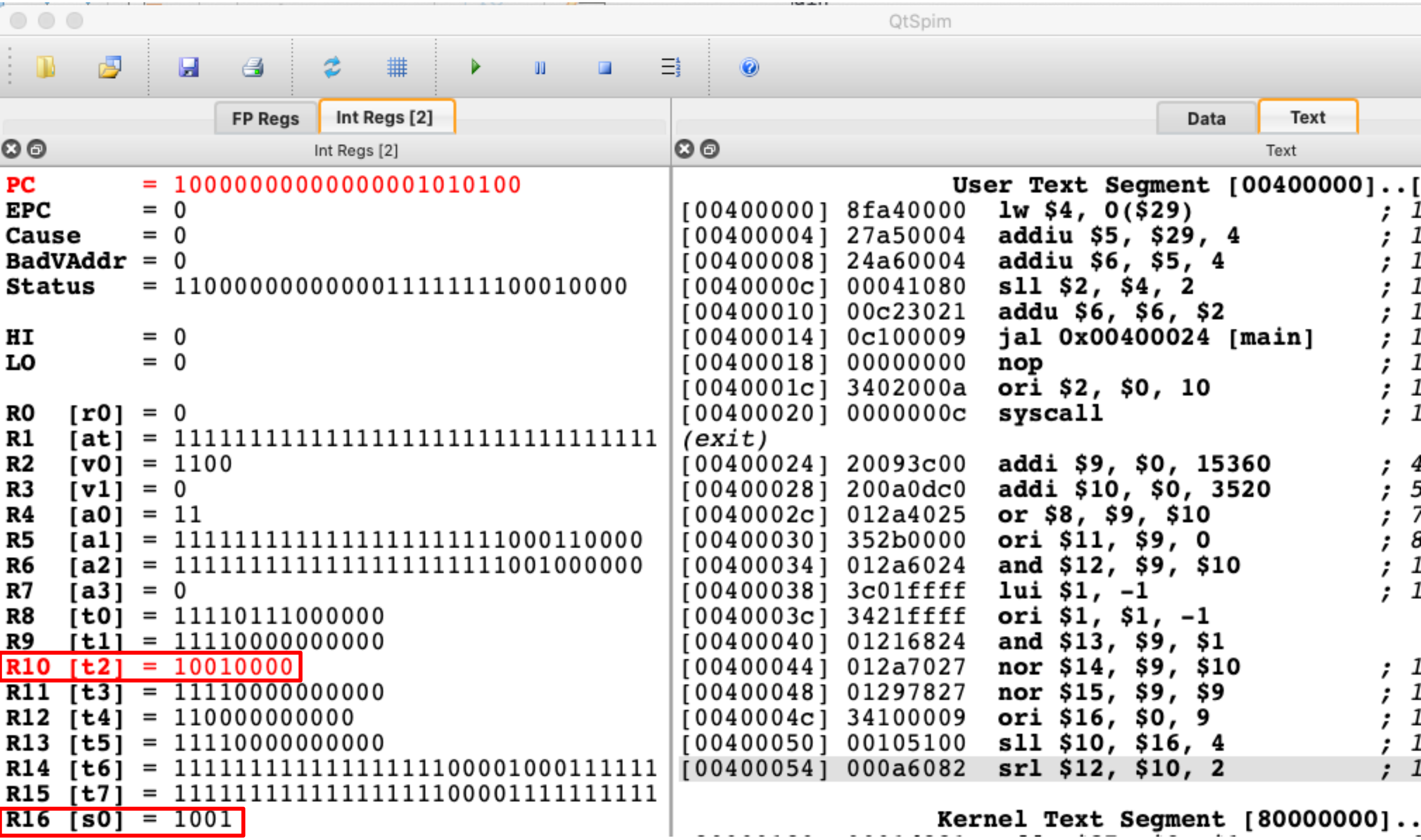
\$s0 : 0000 0000 0000 0000 0000 0000 0000 1001$_{two}$ = 9$_{ten}$

←←←←

```
sll  $t2, $s0, 4
```

\$t2 : 0000 0000 0000 0000 0000 0000 1001 0000$_{two}$ = 144$_{ten}$

왼쪽으로 i bits shift 하는 것은 2^{i} 만큼 곱하는 것과 같은데 연산 속도는 곱셈보다 빠르다.



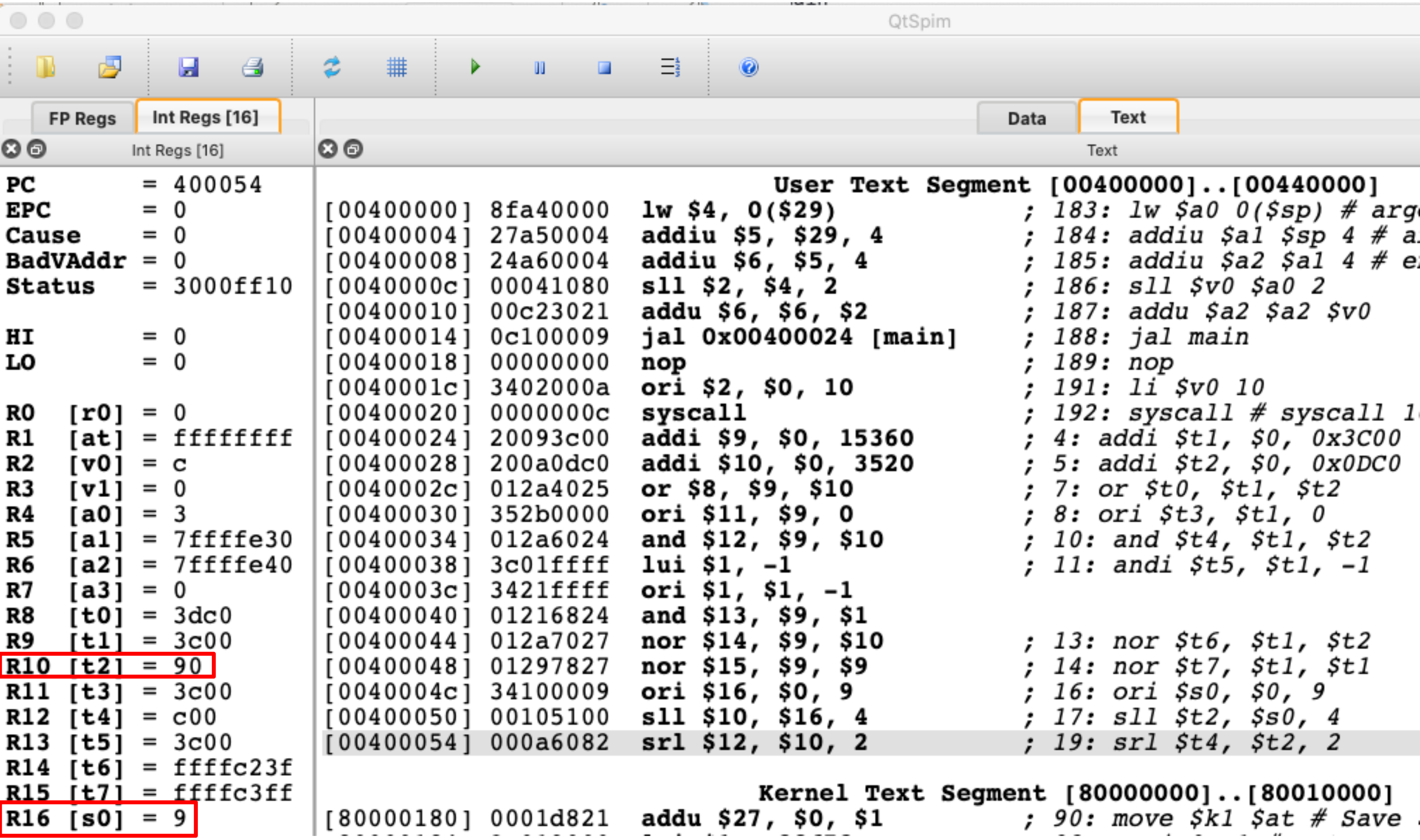
```
PC          = 10000000000000000001010100
EPC         = 0
Cause       = 0
BadVAddr    = 0
Status      = 110000000000000111111100010000

HI          = 0
LO          = 0

R0   [r0] = 0
R1   [at] = 11111111111111111111111111111111
R2   [v0] = 1100
R3   [v1] = 0
R4   [a0] = 11
R5   [a1] = 1111111111111111111111000110000
R6   [a2] = 1111111111111111111111001000000
R7   [a3] = 0
R8   [t0] = 1111011100000
R9   [t1] = 1111000000000
R10  [t2] = 10010000
R11  [t3] = 1111000000000
R12  [t4] = 11000000000
R13  [t5] = 1111000000000
R14  [t6] = 11111111111111111110000100011111
R15  [t7] = 11111111111111111110000111111111
R16  [s0] = 1001
```

```
                    User Text Segment [00400000]..[
[00400000] 8fa40000  lw $4, 0($29)                 ; 1
[00400004] 27a50004  addiu $5, $29, 4              ; 1
[00400008] 24a60004  addiu $6, $5, 4               ; 1
[0040000c] 00041080  sll $2, $4, 2                 ; 1
[00400010] 00c23021  addu $6, $6, $2               ; 1
[00400014] 0c100009  jal 0x00400024 [main]         ; 1
[00400018] 00000000  nop                           ; 1
[0040001c] 3402000a  ori $2, $0, 10                ; 1
[00400020] 0000000c  syscall                       ; 1
(exit)
[00400024] 20093c00  addi $9, $0, 15360            ; 4
[00400028] 200a0dc0  addi $10, $0, 3520            ; 5
[0040002c] 012a4025  or $8, $9, $10                ; 7
[00400030] 352b0000  ori $11, $9, 0                ; 8
[00400034] 012a6024  and $12, $9, $10              ; 1
[00400038] 3c01ffff  lui $1, -1                    ; 1
[0040003c] 3421ffff  ori $1, $1, -1
[00400040] 01216824  and $13, $9, $1
[00400044] 012a7027  nor $14, $9, $10              ; 1
[00400048] 01297827  nor $15, $9, $9               ; 1
[0040004c] 34100009  ori $16, $0, 9                ; 1
[00400050] 00105100  sll $10, $16, 4               ; 1
[00400054] 000a6082  srl $12, $10, 2               ; 1

                   Kernel Text Segment [80000000]..
```



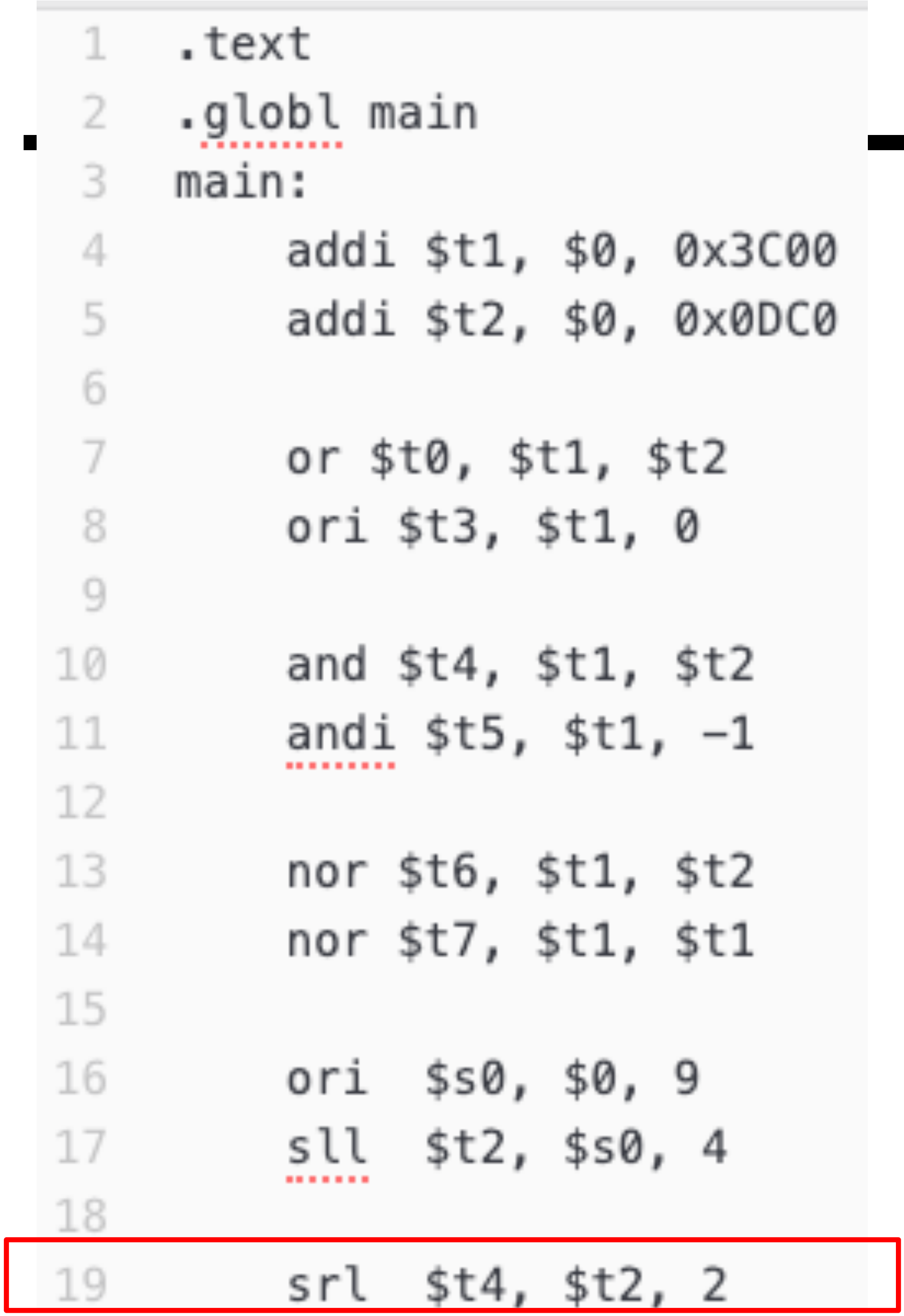
```
PC       = 400054
EPC      = 0
Cause    = 0
BadVAddr = 0
Status   = 3000ff10

HI       = 0
LO       = 0

R0  [r0] = 0
R1  [at] = ffffffff
R2  [v0] = c
R3  [v1] = 0
R4  [a0] = 3
R5  [a1] = 7ffffe30
R6  [a2] = 7ffffe40
R7  [a3] = 0
R8  [t0] = 3dc0
R9  [t1] = 3c00
R10 [t2] = 90
R11 [t3] = 3c00
R12 [t4] = c00
R13 [t5] = 3c00
R14 [t6] = ffffc23f
R15 [t7] = ffffc3ff
R16 [s0] = 9
```

```
                        User Text Segment [00400000]..[00440000]
[00400000] 8fa40000  lw $4, 0($29)              ; 183: lw $a0 0($sp) # arg
[00400004] 27a50004  addiu $5, $29, 4           ; 184: addiu $a1 $sp 4 # a
[00400008] 24a60004  addiu $6, $5, 4            ; 185: addiu $a2 $a1 4 # e
[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 # syscall 1
[00400024] 20093c00  addi $9, $0, 15360         ; 4: addi $t1, $0, 0x3C00
[00400028] 200a0dc0  addi $10, $0, 3520         ; 5: addi $t2, $0, 0x0DC0
[0040002c] 012a4025  or $8, $9, $10             ; 7: or $t0, $t1, $t2
[00400030] 352b0000  ori $11, $9, 0             ; 8: ori $t3, $t1, 0
[00400034] 012a6024  and $12, $9, $10           ; 10: and $t4, $t1, $t2
[00400038] 3c01ffff  lui $1, -1                 ; 11: andi $t5, $t1, -1
[0040003c] 3421ffff  ori $1, $1, -1
[00400040] 01216824  and $13, $9, $1
[00400044] 012a7027  nor $14, $9, $10           ; 13: nor $t6, $t1, $t2
[00400048] 01297827  nor $15, $9, $9            ; 14: nor $t7, $t1, $t1
[0040004c] 34100009  ori $16, $0, 9             ; 16: ori $s0, $0, 9
[00400050] 00105100  sll $10, $16, 4            ; 17: sll $t2, $s0, 4
[00400054] 000a6082  srl $12, $10, 2            ; 19: srl $t4, $t2, 2

                        Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821  addu $27, $0, $1           ; 90: move $k1 $at # Save
```



```
.text
.globl main
main:
    addi $t1, $0, 0x3C00
    addi $t2, $0, 0x0DC0

    or $t0, $t1, $t2
    ori $t3, $t1, 0

    and $t4, $t1, $t2
    andi $t5, $t1, -1

    nor $t6, $t1, $t2
    nor $t7, $t1, $t1

    ori  $s0, $0, 9
    sll  $t2, $s0, 4

    srl  $t4, $t2, 2
```

srl : shift right logical

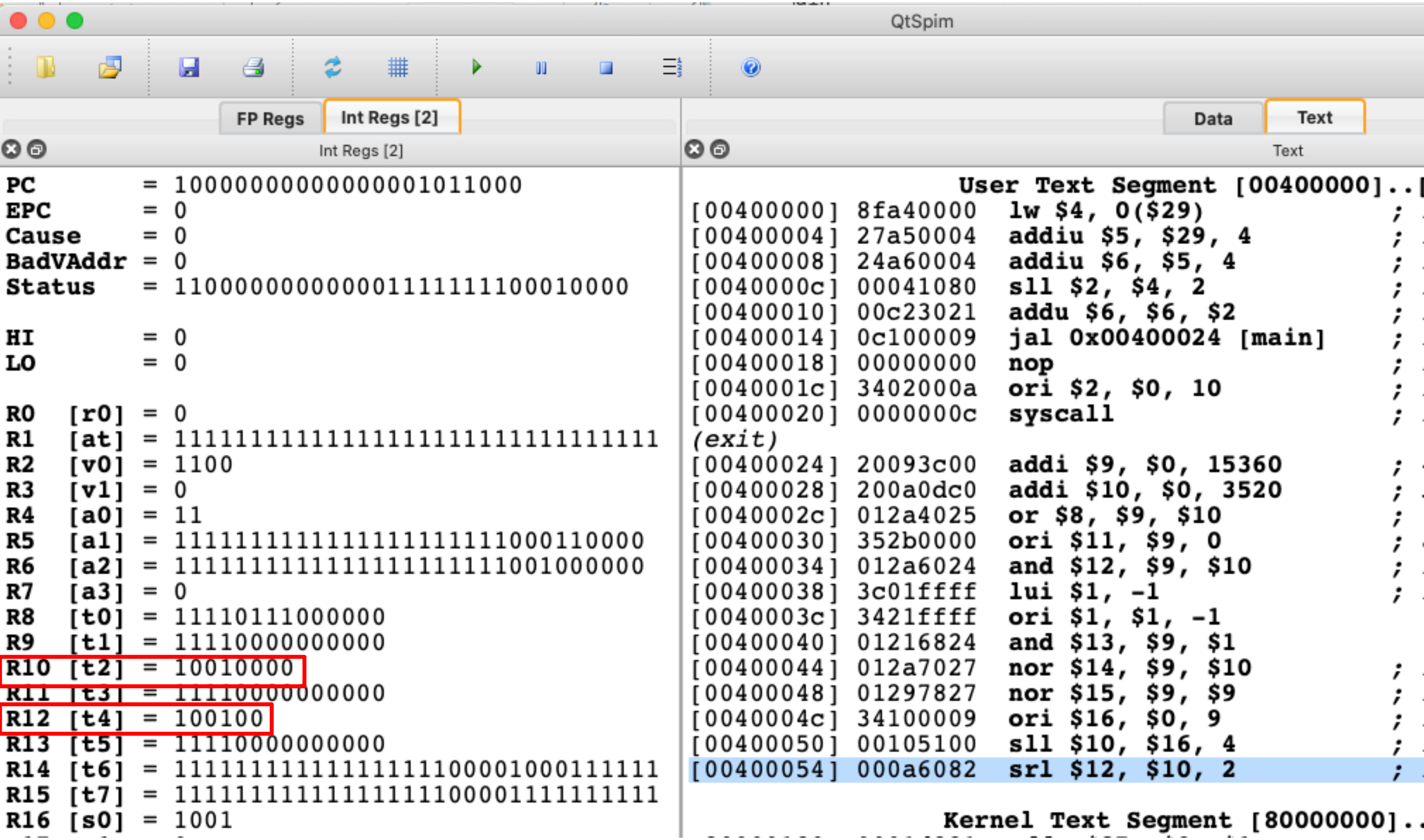
$t2 : 0000 0000 0000 0000 0000 0000 1001 0000 = 0x90

→→

```
srl   $t4, $t2, 2
```

$t4 : 0000 0000 0000 0000 0000 0000 0010 0100 = 0x24

오른쪽으로 i bits shift 하는 것은 2^i 로 나누는 것과 같은데 연산 속도는 나눗셈보다 빠르다.



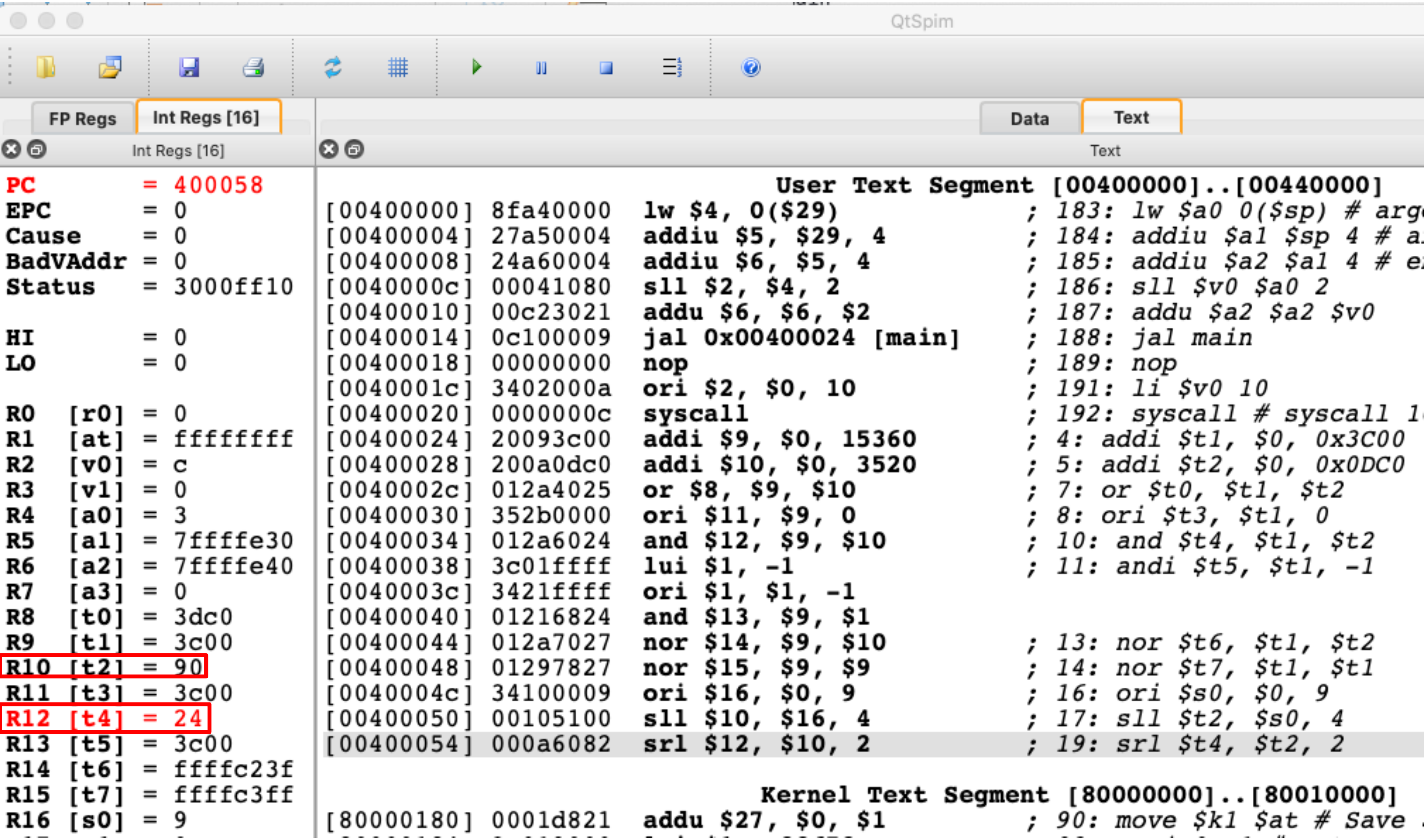
```
PC       = 10000000000000001011000
EPC      = 0
Cause    = 0
BadVAddr = 0
Status   = 110000000000001111111100010000

HI       = 0
LO       = 0

R0  [r0] = 0
R1  [at] = 11111111111111111111111111111111
R2  [v0] = 1100
R3  [v1] = 0
R4  [a0] = 11
R5  [a1] = 1111111111111111111111000110000
R6  [a2] = 1111111111111111111111001000000
R7  [a3] = 0
R8  [t0] = 11110111000000
R9  [t1] = 11110000000000
R10 [t2] = 10010000
R11 [t3] = 11110000000000
R12 [t4] = 100100
R13 [t5] = 11110000000000
R14 [t6] = 11111111111111111100001000111111
R15 [t7] = 11111111111111111100001111111111
R16 [s0] = 1001
```

```
                    User Text Segment [00400000]..[
[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                          ;
(exit)
[00400024] 20093c00  addi $9, $0, 15360               ;
[00400028] 200a0dc0  addi $10, $0, 3520               ;
[0040002c] 012a4025  or $8, $9, $10                   ;
[00400030] 352b0000  ori $11, $9, 0                   ;
[00400034] 012a6024  and $12, $9, $10                 ;
[00400038] 3c01ffff  lui $1, -1                       ;
[0040003c] 3421ffff  ori $1, $1, -1
[00400040] 01216824  and $13, $9, $1
[00400044] 012a7027  nor $14, $9, $10                 ;
[00400048] 01297827  nor $15, $9, $9                  ;
[0040004c] 34100009  ori $16, $0, 9                   ;
[00400050] 00105100  sll $10, $16, 4                  ;
[00400054] 000a6082  srl $12, $10, 2                  ;

                  Kernel Text Segment [80000000]..
```
]



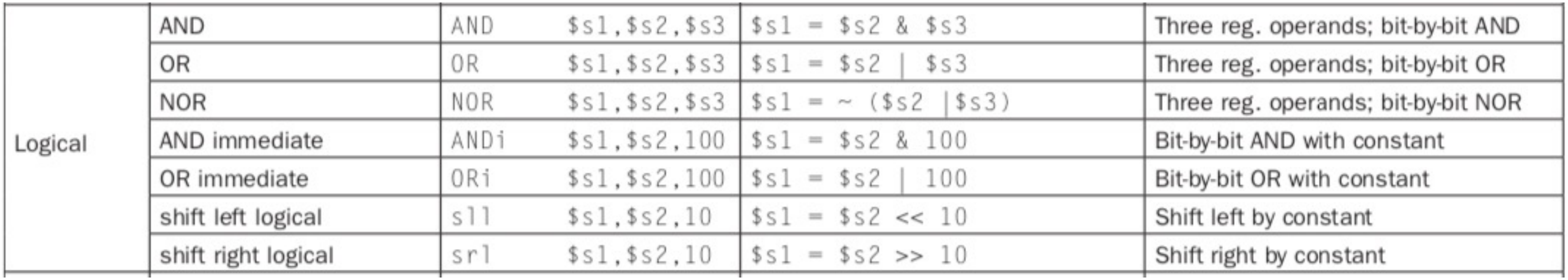
```
PC          = 400058
EPC         = 0
Cause       = 0
BadVAddr    = 0
Status      = 3000ff10

HI          = 0
LO          = 0

R0   [r0] = 0
R1   [at] = ffffffff
R2   [v0] = c
R3   [v1] = 0
R4   [a0] = 3
R5   [a1] = 7ffffe30
R6   [a2] = 7ffffe40
R7   [a3] = 0
R8   [t0] = 3dc0
R9   [t1] = 3c00
R10  [t2] = 90
R11  [t3] = 3c00
R12  [t4] = 24
R13  [t5] = 3c00
R14  [t6] = ffffc23f
R15  [t7] = ffffc3ff
R16  [s0] = 9
```

```
                        User Text Segment [00400000]..[00440000]
[00400000] 8fa40000  lw $4, 0($29)            ; 183: lw $a0 0($sp) # arg
[00400004] 27a50004  addiu $5, $29, 4         ; 184: addiu $a1 $sp 4 # a
[00400008] 24a60004  addiu $6, $5, 4          ; 185: addiu $a2 $a1 4 # e
[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 # syscall 1
[00400024] 20093c00  addi $9, $0, 15360       ; 4: addi $t1, $0, 0x3C00
[00400028] 200a0dc0  addi $10, $0, 3520       ; 5: addi $t2, $0, 0x0DC0
[0040002c] 012a4025  or $8, $9, $10           ; 7: or $t0, $t1, $t2
[00400030] 352b0000  ori $11, $9, 0           ; 8: ori $t3, $t1, 0
[00400034] 012a6024  and $12, $9, $10         ; 10: and $t4, $t1, $t2
[00400038] 3c01ffff  lui $1, -1               ; 11: andi $t5, $t1, -1
[0040003c] 3421ffff  ori $1, $1, -1
[00400040] 01216824  and $13, $9, $1
[00400044] 012a7027  nor $14, $9, $10         ; 13: nor $t6, $t1, $t2
[00400048] 01297827  nor $15, $9, $9          ; 14: nor $t7, $t1, $t1
[0040004c] 34100009  ori $16, $0, 9           ; 16: ori $s0, $0, 9
[00400050] 00105100  sll $10, $16, 4          ; 17: sll $t2, $s0, 4
[00400054] 000a6082  srl $12, $10, 2          ; 19: srl $t4, $t2, 2

                        Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821  addu $27, $0, $1         ; 90: move $k1 $at # Save
```



Logical	AND	AND $s1,$s2,$s3	$s1 = $s2 & $s3	Three reg. operands; bit-by-bit AND
	OR	OR $s1,$s2,$s3	$s1 = $s2 | $s3	Three reg. operands; bit-by-bit OR
	NOR	NOR $s1,$s2,$s3	$s1 = ~ ($s2 |$s3)	Three reg. operands; bit-by-bit NOR
	AND immediate	ANDi $s1,$s2,100	$s1 = $s2 & 100	Bit-by-bit AND with constant
	OR immediate	ORi $s1,$s2,100	$s1 = $s2 | 100	Bit-by-bit OR with constant
	shift left logical	sll $s1,$s2,10	$s1 = $s2 << 10	Shift left by constant
	shift right logical	srl $s1,$s2,10	$s1 = $s2 >> 10	Shift right by constant