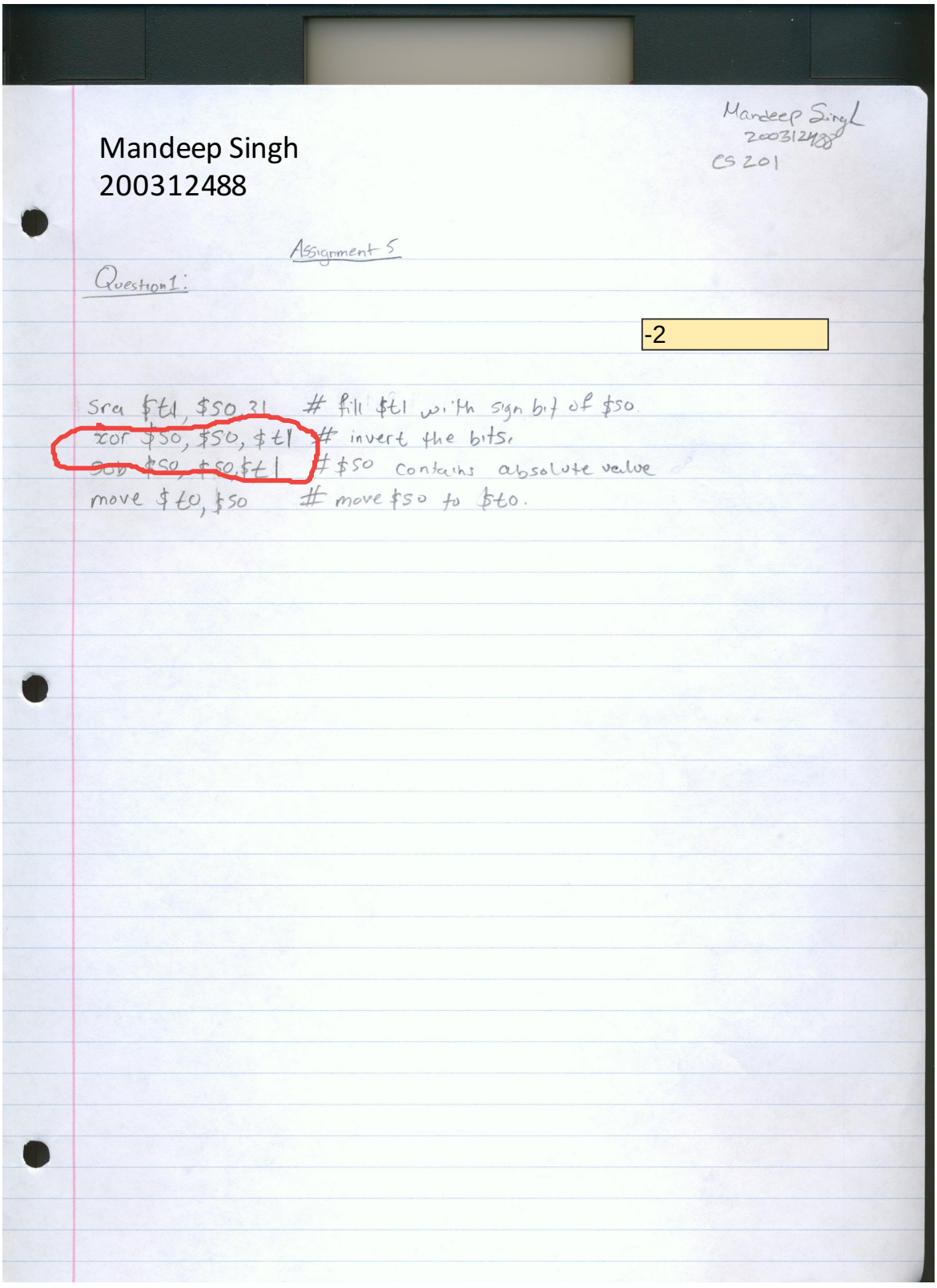
Assignment 5



Mandeep Singh
200312488

Mandeep Singh
200312488
CS 201

Assignment 5

Question 1:

-2

```
sra $t1, $s0, 31    # fill $t1 with sign bit of $s0.
xor $s0, $s0, $t1   # invert the bits.
sub $s0, $s0, $t1   # $s0 contains absolute value
move $t0, $s0       # move $s0 to $t0.
```

Question 2:

a) i) subu $s3, $t9, $v1

=> R[rd] = R[rs] - R[rt]

OP=0	rs	rt	rd	shamt=0	func=23
00000	$t9	$v1	$s3	0000	100011
6	5	5	5	5	

$s3 = 19 => 10011

$t9 = 25 => 11001

$v1 = 3 => 00011

	OP=0	$t9=25	$v1=3	$s3=19	shamt=0	func=23
=>	000000	11001	00011	10011	00000	100011

or: 0x03239823

ii) sra $t7, $a3, 2

=> R[rd] = R[rs] >> shamt.

OP=0	rs	rt=0	rd	shamt=2	func=3
000000	$a3	00000	$t7	00010	000011
6	5	5	5	5	6

$t7 = 15 => 01111

$a3 = 7 => 00111

	OP=0	$a3=7	rt=0	$t7=15	shamt=2	func=3
=>	000000	00111	00000	01111	00010	000011

or: 0x00E07883

b) 0x144C0012

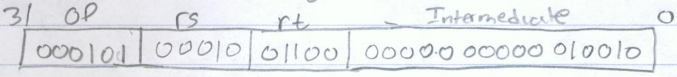
=> 0001 0100 0100 1100 0000 0000 0001 0010
1 4 4 C 0 0 1 2

The first 6 digits always represent OP. code.

OP:

000101 = 5 => bne

=> I Register



if(R[rs] != R[rt])

rs: 00010 => 2 => $v0

rt: 01100 => 12 => $t4

=> bne $v0, $t4 #

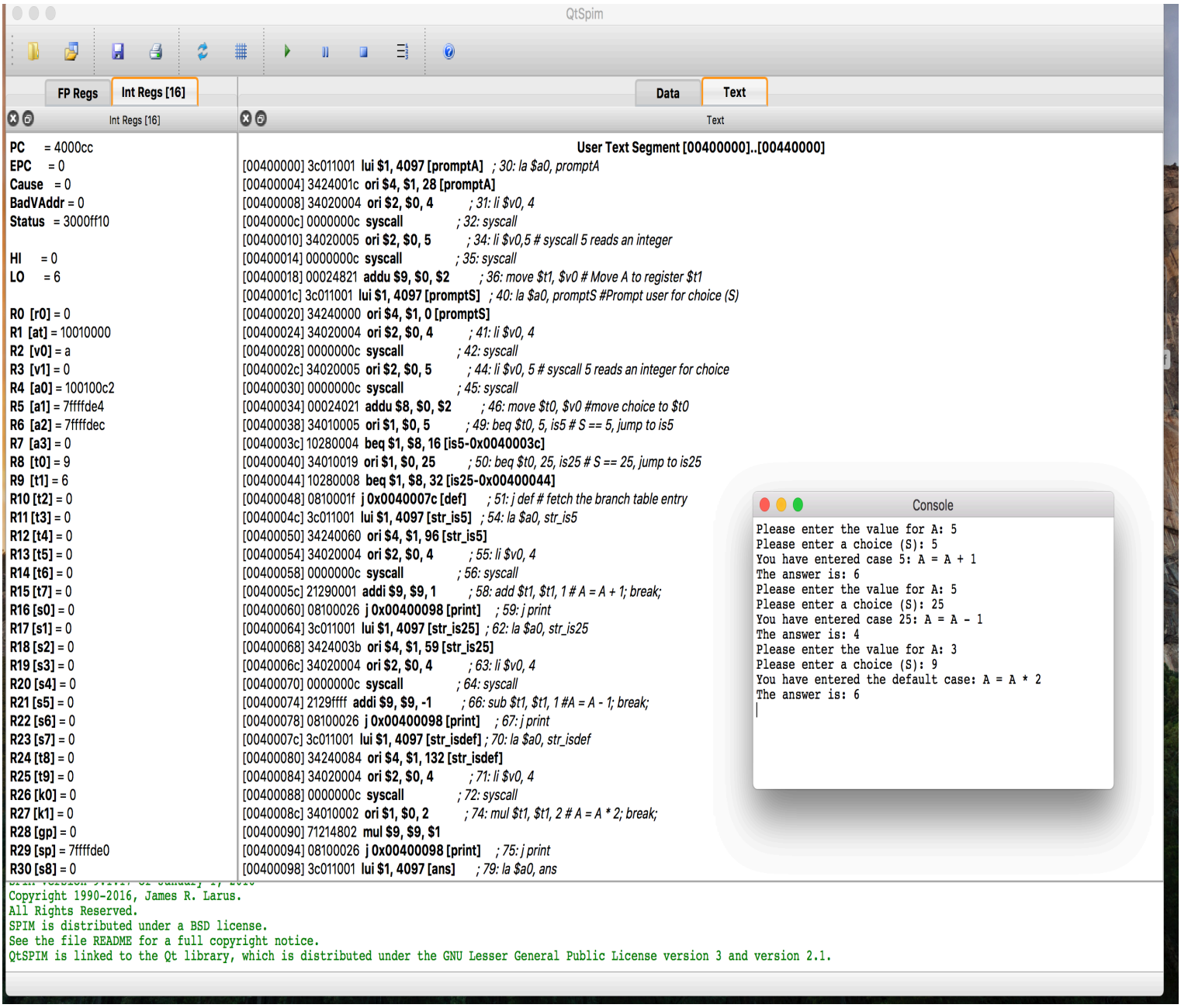
-2

c) li => loads a specific value into a register

la => loads an address into a register

li $t0, 0 => addi, $rt, $rs, 0

Pseudo instruction exist to make it easier to program and increasing the clarity of the code


QtSpim
FP Regs
Int Regs [16]
Data
Text
Int Regs [16]
Text
PC = 4000cc
EPC = 0
Cause = 0
BadVAddr = 0
Status = 3000ff10
HI = 0
LO = 6
R0 [r0] = 0
R1 [at] = 10010000
R2 [v0] = a
R3 [v1] = 0
R4 [a0] = 100100c2
R5 [a1] = 7ffffde4
R6 [a2] = 7ffffdec
R7 [a3] = 0
R8 [t0] = 9
R9 [t1] = 6
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
R24 [t8] = 0
R25 [t9] = 0
R26 [k0] = 0
R27 [k1] = 0
R28 [gp] = 0
R29 [sp] = 7ffffde0
R30 [s8] = 0
User Text Segment [00400000]..[00440000]
[00400000] 3c011001 lui $1, 4097 [promptA] ; 30: la $a0, promptA
[00400004] 3424001c ori $4, $1, 28 [promptA]
[00400008] 34020004 ori $2, $0, 4 ; 31: li $v0, 4
[0040000c] 0000000c syscall ; 32: syscall
[00400010] 34020005 ori $2, $0, 5 ; 34: li $v0,5 # syscall 5 reads an integer
[00400014] 0000000c syscall ; 35: syscall
[00400018] 00024821 addu $9, $0, $2 ; 36: move $t1, $v0 # Move A to register $t1
[0040001c] 3c011001 lui $1, 4097 [promptS] ; 40: la $a0, promptS #Prompt user for choice (S)
[00400020] 34240000 ori $4, $1, 0 [promptS]
[00400024] 34020004 ori $2, $0, 4 ; 41: li $v0, 4
[00400028] 0000000c syscall ; 42: syscall
[0040002c] 34020005 ori $2, $0, 5 ; 44: li $v0, 5 # syscall 5 reads an integer for choice
[00400030] 0000000c syscall ; 45: syscall
[00400034] 00024021 addu $8, $0, $2 ; 46: move $t0, $v0 #move choice to $t0
[00400038] 34010005 ori $1, $0, 5 ; 49: beq $t0, 5, is5 # S == 5, jump to is5
[0040003c] 10280004 beq $1, $8, 16 [is5-0x0040003c]
[00400040] 34010019 ori $1, $0, 25 ; 50: beq $t0, 25, is25 # S == 25, jump to is25
[00400044] 10280008 beq $1, $8, 32 [is25-0x00400044]
[00400048] 0810001f j 0x0040007c [def] ; 51: j def # fetch the branch table entry
[0040004c] 3c011001 lui $1, 4097 [str_is5] ; 54: la $a0, str_is5
[00400050] 34240060 ori $4, $1, 96 [str_is5]
[00400054] 34020004 ori $2, $0, 4 ; 55: li $v0, 4
[00400058] 0000000c syscall ; 56: syscall
[0040005c] 21290001 addi $9, $9, 1 ; 58: add $t1, $t1, 1 # A = A + 1; break;
[00400060] 08100026 j 0x00400098 [print] ; 59: j print
[00400064] 3c011001 lui $1, 4097 [str_is25] ; 62: la $a0, str_is25
[00400068] 3424003b ori $4, $1, 59 [str_is25]
[0040006c] 34020004 ori $2, $0, 4 ; 63: li $v0, 4
[00400070] 0000000c syscall ; 64: syscall
[00400074] 2129ffff addi $9, $9, -1 ; 66: sub $t1, $t1, 1 #A = A - 1; break;
[00400078] 08100026 j 0x00400098 [print] ; 67: j print
[0040007c] 3c011001 lui $1, 4097 [str_isdef] ; 70: la $a0, str_isdef
[00400080] 34240084 ori $4, $1, 132 [str_isdef]
[00400084] 34020004 ori $2, $0, 4 ; 71: li $v0, 4
[00400088] 0000000c syscall ; 72: syscall
[0040008c] 34010002 ori $1, $0, 2 ; 74: mul $t1, $t1, 2 # A = A * 2; break;
[00400090] 71214802 mul $9, $9, $1
[00400094] 08100026 j 0x00400098 [print] ; 75: j print
[00400098] 3c011001 lui $1, 4097 [ans] ; 79: la $a0, ans
Console
Please enter the value for A: 5
Please enter a choice (S): 5
You have entered case 5: A = A + 1
The answer is: 6
Please enter the value for A: 5
Please enter a choice (S): 25
You have entered case 25: A = A - 1
The answer is: 4
Please enter the value for A: 3
Please enter a choice (S): 9
You have entered the default case: A = A * 2
The answer is: 6
Copyright 1990-2016, James R. Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.