Brandon Nguyen

Professor Salloum

CS 264

16 November 2016

Lab #4

.data

space: .asciiz " "

newLine: .asciiz "\n"

newPara: .asciiz "\n\n\n\n\n\n"

.globl main

.text

main: li $t0, -4

li $t1, -2147483648

li $t2, 2147483648

li $t3, 65

srl $t4, $t0, 2

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

sra $t4, $t0, 2

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

sll $t4, $t0, 1

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

rol $t4, $t0, 2

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

ror $t4, $t0, 2

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

xor $t4, $t0, $t1

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

xor $t4, $t1, -8

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

add $t4,$t0, $t1

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

addu $t4, $t0,$t1

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

mul $t4, $t1,$t2

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

mulo $t4,$t1,$t2

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

mulou $t4,$t0,$t0

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

mulou $t4,$t3,$t3

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

div $t4, $t1,$t0

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

div $t4, $t3,$t0

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

divu $t4, $t1,$t0

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

sub $t4, $t1,$t0

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

subu $t4, $t1,$t0

move $a0, $t4

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

mult $t1,$t2

mfhi $a0

li $v0, 1

syscall

la $a0, newLine

li $v0, 4

syscall

mflo $a0

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

multu $t0, $t3

mfhi $a0

li $v0, 1

syscall

la $a0, newLine

li $v0, 4

syscall

mflo $a0

li $v0, 1

syscall

la $a0, newPara

li $v0, 4

syscall

end: li $v0, 10

syscall

Output:

1. srl $t4, $t0, 2

1073741823

$t0 = 11111111111111111111111111111100 = -4

$t4 = 00111111111111111111111111111111 = 2^29 + 2^28+…+2^0 = 1073741823 insert 0s

1. sra $t4, $t0,2

-1

$t0 = 11111111111111111111111111111100 = -4

$t4 = 11111111111111111111111111111111 = -1 inserts the sign $t2 bits which is one.

1. sll $t4, $t0,1

-8

$t0 = 11111111111111111111111111111100 = -4

$t4 = 11111111111111111111111111111000 = -8 insert 0s

1. rol $t4, $t0,2

-13

$t0 = 11111111111111111111111111111100 = -4

$t4 = 11111111111111111111111111110011 = -13 rotate to left two times

1. ror $t4, $t0,2

1073741823

$t0 = 11111111111111111111111111111100 = -4

$t4 = 00111111111111111111111111111111 = 1073741823 rotate to right two times

1. xor $t4, $t0,$t1

2147483644

$t0 = 11111111111111111111111111111100 = -4

$t1 = 10000000000000000000000000000000 = -2147483648

$t4 = 01111111111111111111111111111100 = -4 + 231 = 2147483644

1. xor $t4, $t1, -8

2147483640

$t1 = 10000000000000000000000000000000 = -2147483648

$t4 = 01111111111111111111111111111000 = 2^30 + 2^29+…+2^3 = 2147483640

1. add $t4,$t0, $t1

Exception 12 [Arithmetic overflow] occurred and ignored

2147483640

Overflow in 2’s compliment. Two negatives generated a positive

1. addu $t4, $t0,$t1

2147483644

$t0 = 11111111111111111111111111111100 = -4

$t1 = 10000000000000000000000000000000 = -2147483648

$t4 = 01111111111111111111111111111100 = -4 + 231 = 2147483644

10. mul $t4, $t1,$t2

-2147483648

$t1 = 10000000000000000000000000000000 = -2147483648

$t2 = 01111111111111111111111111111111 = 231 – 1 = 2147483647

$t4 = 10000000000000000000000000000000 = -2147483648

11. mulo $t4,$t1,$t2

Exception 9 [Breakpoint] occurred and ignored

-2147483648

$t4 = -231 \* 231 – 1 will be more than 32 bits so overflow is generated because mulo

12. mulou $t4,$t0,$t0

Exception 9 [Breakpoint] occurred and ignored

16

$t4 = mulou treats $t0 = 4294967292, which multiplied to itself will be more than 32 bits so overflow is generated

13. mulou $t4,$t3,$t3

4225

$t3 = 00000000000000000000000001000001 = 65

$t4 = 00000000000000000001000010000001 = 65 \* 65 = 4225

14. div $t4, $t1,$t0

536870912

$t1 = 10000000000000000000000000000000 = -2147483648

$t0 = 11111111111111111111111111111100 = -4

$t4 = 00100000000000000000000000000000 = 230 = 536870912

15. div $t4, $t3,$t0

-16

$t3 = 00000000000000000000000001000001 = 65

$t0 = 11111111111111111111111111111100 = -4

$t4 = 11111111111111111111111111110000 = -16 remainder = 1

16. divu $t4, $t1,$t0

0

$t1 = 10000000000000000000000000000000 = 2147483648 divu treats registers as unsigned

$t0 = 11111111111111111111111111111100 = 4294967292

$t4 = 00000000000000000000000000000000 = 0

17. sub $t4, $t1,$t0

-2147483644

$t1 = 10000000000000000000000000000000 = -2147483648

$t0 = 11111111111111111111111111111100 = -4

$t4 = 10000000000000000000000000000100 = -2147483644

18. subu $t4, $t1,$t0

-2147483644

$t1 = 10000000000000000000000000000000 = -2147483648

$t0 = 11111111111111111111111111111100 = -4

$t4 = 10000000000000000000000000000100 = -2147483644

19. mult $t1,$t2 and then print the hi and low register

-1073741824

-2147483648

$t1 = 10000000000000000000000000000000 = -2147483648

$t2 = 01111111111111111111111111111111 = 2147483647

HI = 11000000000000000000000000000000 = -231 + 230 = -1073741824

LO = 10000000000000000000000000000000 = -2147483648

20. multu $t0, $t3 and then print the hi and low registers.

64

-260

$t0 = 11111111111111111111111111111100 = 4294967292 multu treats registers as unsigned

$t3 = 00000000000000000000000001000001 = 65

HI = 00000000000000000000000001000000 = 26 = 64

LO = 11111111111111111111111011111100 = -4 – 28 = -260