Joshuan Jimenez
Cristian Caro
CDA4101 - RVBB
Assignment #1

Assignment #1

1. Adding Three Numbers

1. .data # variable declarations follow this line
2. .text # instructions follow this line
3. main:
4. ## Code Part 1: Get first number from user, put into $t0.
5. ori $v0, $0, 5 # OUR CODE BEGINS HERE: load syscall read_int into $v0.
6. syscall # make the syscall.
7. addu $t0, $0, $v0 # move the number read into $t0.
8. ## Get second number from user, put into $t1.
9. ori $v0, $0, 5 # load syscall read_int into $v0.
10. syscall # make the syscall.
11. addu $t1, $0, $v0 # move the number read into $t1.
12. add $t2, $t0, $t1 # compute the sum.
13. ori $v0, $0, 5 # load syscall read_int into $v0.
14. syscall # make the syscall.
15. addu $t3, $0, $v0 # move the number read into $t3.
16. add $t4, $t3, $t2 # compute the sum.
17. ## Print out $t4.
18. addu $a0, $0, $t4 # move the number to print into $a0.
19. ori $v0, $0, 1 # load syscall print_int into $v0.
20. syscall # make the syscall.
21. ori $v0, $0, 10 # syscall code 10 is for exit.
22. syscall # make the syscall.
23. ## end of add2.asm.

2. In this code, we had to repeat the procedure of obtaining the third number using ori and syscall, to then be able to move the new number into register $t3. From there, we can add the result from the first computation, which is stored in $t2 with the new input ($t3), which is then loaded into $t4. Then, we can move the number in $t4 to $a0 to then print the number.

3. In this code, the main thing it does is add two immediate values into a set memory location.

4. The result is 5 and it is stored into the location in register $s7 + 4. At first, a memory location is loaded into $s7, which is then shifted by the sll line. Then, the value 2 is put into register $s0, taken out, put back, complemented, then put back again. Afterwards, the value 3 is put into $s1. Registers $s0 and $s1 are added together and the result is put into $s2. Then, since $s2 is not equal to $zero (0 != 5), the branch is not taken, and the result is then stored into $s7 + 4 due to the sw line. This then goes to the j exit line which loads the exit and ends the program.

Joshuan Jimenez
Cristian Caro
CDA4101 - RVBB
Assignment #1

5. The ori instruction is being used to load the immediate values into the appropriate registers.

6. The complement operation was used to get the complement of the value in register $s0, which entails performing an OR operation.

7. This program does the sum of all elements in the array. In register $s1, the value held is 30, which represents the sum of all the elements in that given array.

8.

```
.data #
A: .asciiz "Please enter the value of A:"
B: .asciiz "Please enter the value of B:"
S: .asciiz " "
.text # Put program here
.globl main # globally define 'main'
main: li $v0, 4
la $a0, A # Ask for input A
syscall
ori $v0, $0, 5
syscall
addu $t0, $0, $v0 # Store A
li $v0, 4
la $a0, B # Ask for input B
syscall
ori $v0, $0, 5
syscall
addu $t1, $0, $v0 # Store B
blez $t1, stop # b <= 0, stop
mul $t2, $t0, $t1 # S = A * B
addu $t3, $t0, $t3 # m = A
loop: li $v0, 1
move $a0, $t3
syscall # Print m
beq $t2, $t3, stop # S == m, stop
addu $t3, $t3, $t0 # Add m + A
li $v0, 4
la $a0, S # Ask for input A
syscall
j loop
stop: li $v0, 10 # syscall to exit cleanly from main only
syscall # ends execution
.end
```