# LAB # 7

ARITHMETIC INSTRUCTIONS IN MIPS

|  |
| --- |
| **Understanding the concepts of Register Manipulation for different Operations** |
|  |

OBJECTIVE

Write a program using “addi”, “blez” and “bnez” instructions.

TIME BOXING

|  |  |  |
| --- | --- | --- |
| Activity Name | Activity Time | Total Time |
| Instruments Allocation + Setting up Lab | 10 mints | 10 mints |
| Walk through Theory & Tasks (Lecture) | 60 mints | 60 mints |
| Implementation & Practice time | 90 mints | 80 mints |
| Evaluation Time | 20 mints | 20 mints |
|  | Total Duration | 180 mints |

THEORY

addi Rdest,Rsrc1, imm

Rdest = Rsrc1 + 16-bit signed immediate. In case of an overflow, an overflow exception is generated.

blez Rs, Label

Branch if Less Than or Equal to Zero.

if ( Rs ≤ 0 ) go to Label.

bnez Rs, Label

Branch if Not Equal to Zero.

if (Rs != 0) go to Label

Program#1:

Sum of Positive Integers

# Objective: Computes the sum of all positive integers <= the input number.

# Input: Requests input number.

# Output: Outputs the sum.

################### Data segment ###################

.data

prompt: .asciiz "\n Please Input a value for N = "

result: .asciiz " The sum of the integers from 1 to N is "

bye: .asciiz "\n \*\*\*\* Have a good day \*\*\*\*"

################### Code segment ###################

.text

.globl main

main:

la $a0, prompt # load address of prompt into $a0

li $v0, 4 # system call code for Print String

syscall # print the prompt message

li $v0, 5 # system call code for Read Integer

syscall # reads the value of N into $v0

blez $v0, end # branch to end if $v0 <= 0

li $t0,0

loop:

add $t0, $t0, $v0 # sum of integers in register $t0

addi $v0, $v0, -1 # summing integers in reverse order

bnez $v0, loop # branch to loop if $v0 is != zero

la $a0, result # load address of message into $a0

li $v0, 4 # system call code for Print String

syscall # print the string

move $a0, $t0 # move value to be printed to $a0

li $v0, 1 # system call code for Print Integer

syscall # print sum of integers

end:

la $a0, bye # load address of message into $a0

li $v0, 4 # system call code for Print String

syscall # print the string

li $v0, 10 # terminate program

syscall # return control to system

## LAB TASK

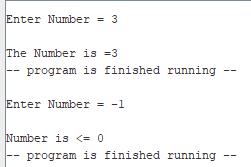
Task 1: Write a MIPS assembly language program that takes an input if value is zero or less than zero halt a program else print the number.

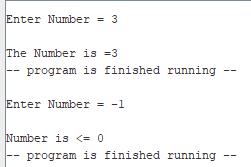
CODE:

.text

.globl main

OUTPUT:





main:

la $a0,a

li $v0,4

syscall

li $v0,5

syscall

move $t1,$v0

blez $v0,halt

la $a0,ans

li $v0,4

syscall

move $a0,$t1

li $v0,1

syscall

li $v0,10

syscall

halt:

la $a0,end\_result

li $v0,4

syscall

li $v0,10

syscall

Task 2: Write a MIPS assembly language program that takes an input if the input value is zero only halt a program else print the number.

CODE:

.data

#--- LABELS--------------------

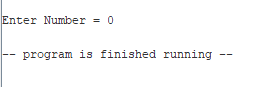
a: .asciiz"Enter Number = "

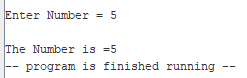
ans: .asciiz"\nThe Number is ="

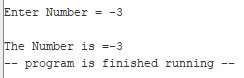
#-----------------------------

.text

OUTPUT:







.globl main

main:

la $a0,a

li $v0,4

syscall

li $v0,5

syscall

move $t1,$v0

bnez $v0,print

li $v0,10

syscall

print:

la $a0,ans

li $v0,4

syscall

move $a0,$t1

li $v0,1

syscall

li $v0,10

syscall