# **FOR LOOP; CONTROL STRUCTURE IN MIPS**

## **OBJECTIVE**

Study how to implement translation of “for loop” control structure in MIPS assembly

langage.

## **THEORY**

Translation of a “FOR LOOP” Control Structure

Obviously a “for loop” control structure is very useful. Let us suppose that a

programmer initially developed an algorithm containing the following pseudo code.

In one sentence, can you describe what this algorithm accomplishes?

$a0 = 0;

for ( $t0 =10; $t0 > 0; $t0= $t0 -1)

do {$a0 = $a0 + $t0}

The following is a translation of the above “for loop” pseudo code to MIPS assembly

language code.

* li $a0, 0 # $a0 = 0
* li $t0, 10 # Initialize loop counter to 10
* loop:
* add $a0, $a0, $t0
* addi $t0,$t0, -1 # Decrement loop counter
* bgtz $t0, loop # If ($t0 > 0) Branch to loop BRANCH IF GREATER THAN ZERO
* bgtz $t0,loop # branch to loop if $t0>0

## **Program#1:**

Translation of a FOR LOOP Control Structure

# Objective: Translates a FOR LOOP control structure.

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

.data counter: .asciiz "\n value of count,

$t0: " total: .asciiz "value of sum,

$a0: " tab: .asciiz "\t"

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

.text

.globl main

main:

li $a2,0 # $a2=0

li $t0,10 #initialize loop variable counter $t0=10

loop:

add $a2,$a2,$t0

la $a0,counter # print message of counter

li $v0,4

syscall

move $a0,$t0 # print value of $t0

li $v0,1

syscall

addi $t0,$t0,-1 # decrement loop variable counter

la $a0,tab # print tab

li $v0,4

syscall

la $a0,total # print message of total

li $v0,4

syscall

move $a0,$a2 # print value of $a2

li $v0,1

syscall

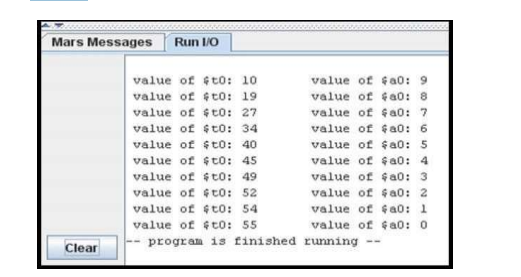
bgtz $t0,loop # if($t0>0) branch to loop

end:

li $v0,10

syscall

## **Output:**



## **LAB TASK**

Task 1: Write a program in MIPS assembly language that takes input and display whether number is prime or not.

Task 2: Write a program in MIPS assembly language that provide the sum from 1 to 99 using for Loop