

Lab7:

Objectives:

The main objective of this lab is to understand control structures in MIPS. This lab will teach students the concept of decision making in assembly language. Students will be able to do translation of high level decision making pseudo code in MIPS.

Flow Control:

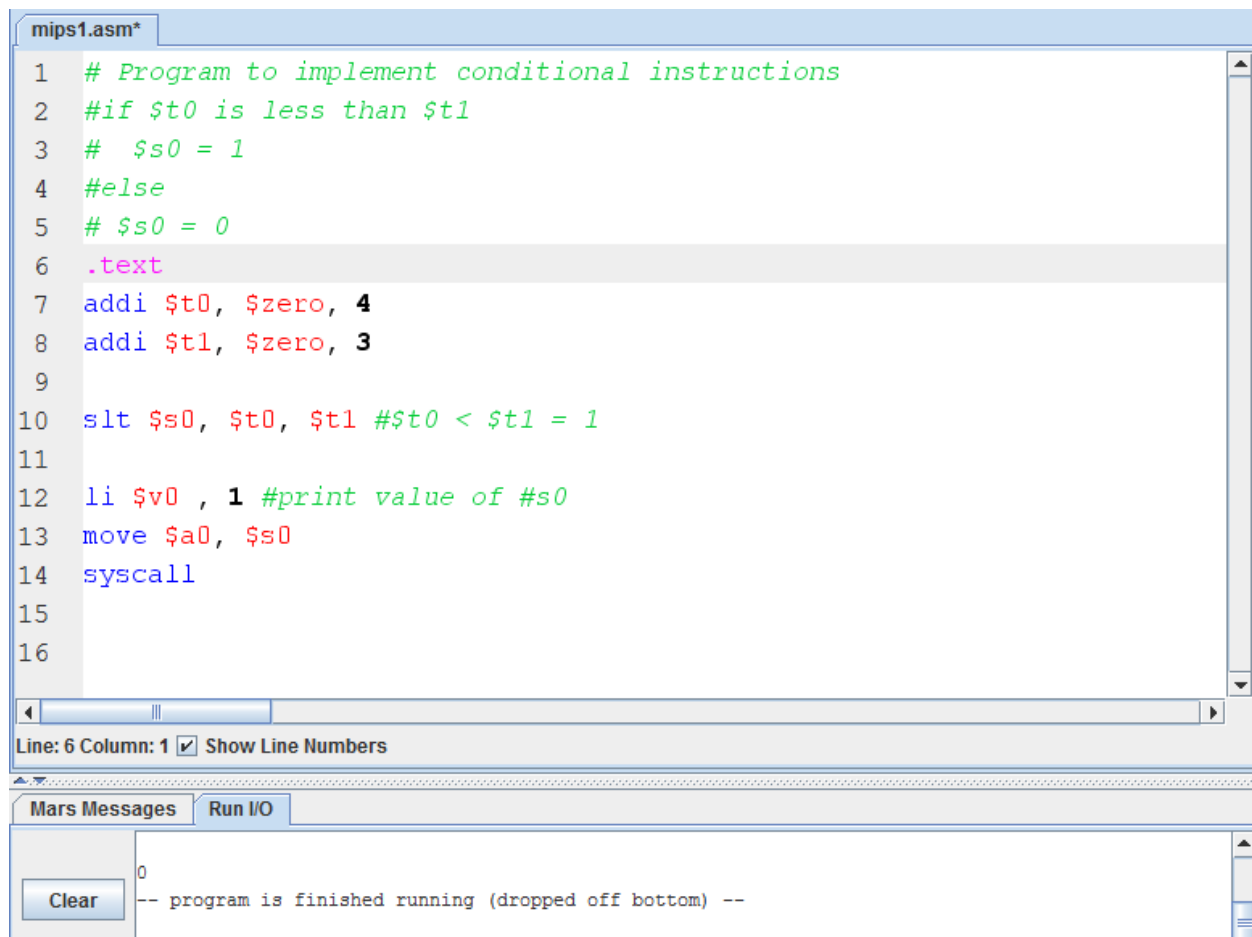
In previous labs, programs made in assembly were purely sequential (each instruction executed in a sequential order). In order to make assembly language more useful, MIPS contains control instructions, which allow programs to execute in a non-sequential manner. In a high-level language, such as C, control structures have the form if-then-else, goto, for, or do-while. In assembly language, the flow control statements are generally simpler and are of the form branch, jump, call, and return. Each has the ability to alter the flow of the program.

Conditional Instructions:

op-	Meaning	Description
slt slti	Set if less than	If the leftOperand is less than the rightOperand, the < <i>destination</i> > is set to 1, and 0 otherwise.
sgt	Set if greater than	If the leftOperand is greater than the rightOperand, the < <i>destination</i> > is set to 1, and 0 otherwise.
sle	Set if less than or equal	If leftOperand is less than or equal to rightOperand, the < <i>destination</i> > is set to 1, and 0 otherwise.
sge	Set if greater than or equal	If leftOperand is greater than or equal to rightOperand, the < <i>destination</i> > is set to 1, and 0 otherwise.
seq	Set if equal	If leftOperand equal to rightOperand, the < <i>destination</i> > is set to 1, and 0 otherwise.
sne	Set if not equal	If leftOperand not equal to rightOperand, the < <i>destination</i> > is set to 1, and 0 otherwise.

Sample Program:

Program to print value of \$s0, if \$t0 is less than \$t1, then set \$s0 to 1, otherwise 0.



```
mips1.asm*
1  # Program to implement conditional instructions
2  #if $t0 is less than $t1
3  #  $s0 = 1
4  #else
5  # $s0 = 0
6  .text
7  addi $t0, $zero, 4
8  addi $t1, $zero, 3
9
10 slt $s0, $t0, $t1 # $t0 < $t1 = 1
11
12 li $v0, 1 # print value of $s0
13 move $a0, $s0
14 syscall
15
16
```

Line: 6 Column: 1 ☒ Show Line Numbers

Mars Messages Run I/O

0
Clear -- program is finished running (dropped off bottom) --

Explanation:

`slt $s0, $t0, $t1` is used to compare values of `$t0` and `$t1`, if `$t0 < $t1`, then `$s0` is set to 1, otherwise `$s0` is set to 0.

Task1:

Write a program to take two integers from user. Store them in registers `$t0` and `$t1` respectively. Compare them using `slt $s0, $t0, $t1` instruction and print the value of `$s0`.

Sample Output:

Enter first value:5

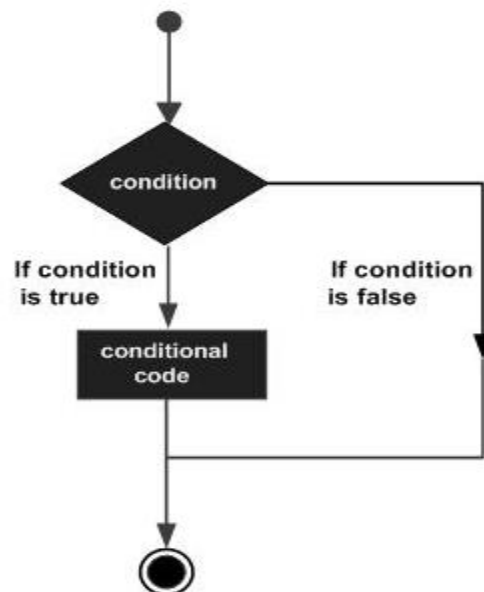
Enter Second value: 4

The value of `$s0` is: 0

Conditional Branch Instructions:

Instruction	Description
blt R_s, R_t, label	if ($R_s < R_t$) branch to label
bltz R_s, label	if ($R_s < 0$) branch to label
bgt R_s, R_t, label	if ($R_s > R_t$) branch to label
bgtz R_s, label	if ($R_s > 0$) branch to label
beq R_s, R_t, label	if ($R_s == R_t$) branch to label
beqz R_s, label	if ($R_s = 0$) branch to label
bne R_s, R_t, label	if ($R_s \neq R_t$) branch to label
bnez R_s, label	if ($R_s \neq 0$) branch to label
ble R_s, R_t, label	if ($R_s \leq R_t$) branch to label
blez R_s, label	if ($R_s \leq 0$) branch to label
bge R_s, R_t, label	if ($R_s \geq R_t$) branch to label
bgez R_s, label	if ($R_s \geq 0$) branch to label

IF Statement:



Sample Program:

If \$t0 contains a negative number then put -1 in \$s0.

```
1  .data
2  msg1:.asciiz" Enter a number : "
3  .text
4  li $v0,4
5  la $a0,msg1
6  syscall
7  li $v0,5
8  syscall
9  move $t0,$v0
10 bgez $t0,exit
11 li $s0,-1
12 li $v0,1
13 move $a0,$s0
14 syscall
15 exit:
16 li $v0,10
17 syscall
```

Explanation:

In the above code `bgez $t0,exit` is used. `bgez` stands for branch if greater or equal to zero. This instruction helps in decision making.

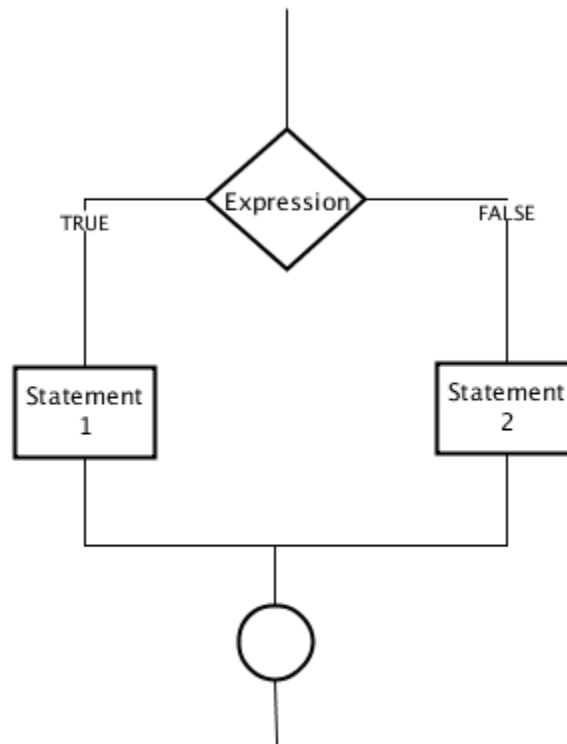
Task2:

Write a program to take three integers from user, find the maximum of three numbers and print it on the screen.

Task3:

Write a program to take integer from user. If it is greater than 18 then it should print "Eligible to Vote"

IF ELSE Statement:



Sample Program:

Suppose \$t0 and \$t1 contains signed numbers. Write some code to put the biggest one in \$s0 and display it.

```

1  .data
2  msg1:.asciiz" Enter a number : "
3  msg2:.asciiz"\n Enter a second number: "
4  .text
5  li $v0,4
6  la $a0,msg1
7  syscall
8  li $v0,5
9  syscall
10 move $t0,$v0
11 li $v0,4
12 la $a0,msg2
13 syscall
14 li $v0,5
15 syscall
16 move $t1,$v0
17 blt $t0,$t1,next
18 move $s0,$t0
19 b display
20 next:
21 move $s0,$t1

22 display:
23 li $v0,1
24 move $a0,$s0
25 syscall
26 exit:
27 li $v0,10
28 syscall

```

Task4:

Write an assembly language program that takes 2 inputs from user. If first number is greater than the second then add first and second number, else subtract them.