|  |
| --- |
| **CS -224Computer Architecture** |
| Course Instructor:**Eng. Qaiser Ijaz** |
| **Lab 3** |
| Issue Date: |
| **Name: Muhammad Talha Amin; Roll No13CS30:** |

**Objective:**To implement an algorithm of minimum-findingin a set of elementsusing MIPS instructions.

**Procedure:**

1. Start QtSpim on your workstation.
2. Analyze the state zero.
3. Download the source file (Lab03.s) from the course group.

Directory is Sp14CA =>Labs =>Lab03.s Tutorial MIPS

1. Follow the document, heading to heading.
2. Answer the questions or/and complete the task.
3. Include any other asked content.

**Filling in the Program:**

Assume that you have an array of 10 elements with base address in $s0. Append to the template code already given the code necessary to find the minimum value from the array (by iterating over the array) and store that value in register $t0.

**Note:**You are free to use any MIPS assembly and instructions and pseudo instructions that are recognized by QtSpim, but in order to receive full credit you must use a loop.

**Executing the Program:**

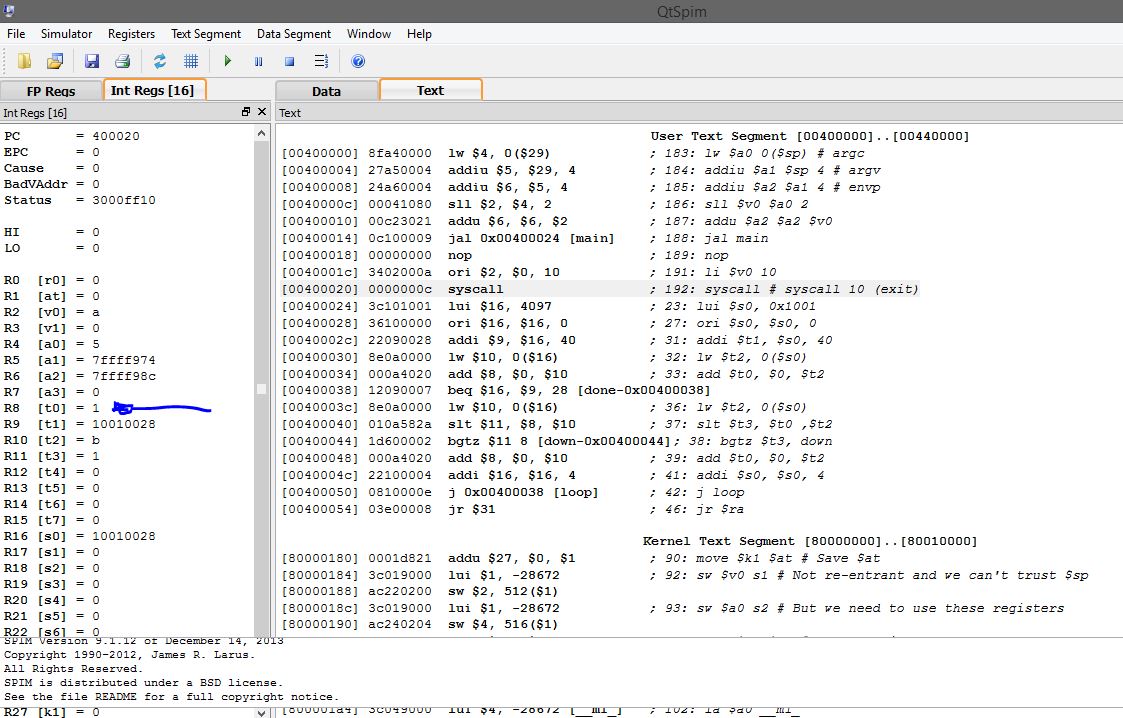
Once you have saved the program template, you can write the code, assuming that the base address is in $s0. There are 10 elements to be read, and the minimum of those to be found.Details about executing the program were covered in previous two labs.

**Note:**Once you find that you have to make changes to your code, make the changes in a text editor, then reinitialize the simulator [**Simulator -> Reinitialize**], and load the file again [**File->Open**]

**Further Task:**

In this lab, we tried writing and executing a program that finds the minimum of a given set of elements by looping over the set. Is there any other way of doing it, if yes then write the program.

***Result Proof***

******

***# Program code with loop #***

.data

.word 7

.word 3

.word 17

.word 12

.word 30

.word 2

.word 5

.word 1

.word 16

.word 11

.text

.globl main

main:

lui $s0, 0x1001

ori $s0, $s0, 0

addi $t1, $s0, 40

lw $t2, 0($s0)

add $t0, $0, $t2

loop:

beq $s0, $t1, done

lw $t2, 0($s0)

blt $t0 , $t2 , down

add $t0, $0, $t2

down:

addi $s0, $s0, 4

j loop

done:

jr $ra

**# Further task solution #**

**.data**

**.word 7**

**.word 3**

**.word 17**

**.word 12**

**.word 30**

**.word 2**

**.word 5**

**.word 1**

**.word 16**

**.word 11**

**.text**

**.globl main**

**main:**

**lui $s0, 0x1001**

**ori $s0, $s0, 0**

**lw $t2, 0($s0)**

**add $t0, $0, $t2**

**lw $t2, 4($s0)**

**blt $t0 , $t2 , down**

**add $t0, $0, $t2**

**down:**

**lw $t2, 8($s0)**

**blt $t0 , $t2 , down1**

**add $t0, $0, $t2**

**down1:**

**lw $t2, 12($s0)**

**blt $t0 , $t2 , down2**

**add $t0, $0, $t2**

**down2:**

**lw $t2, 16($s0)**

**blt $t0 , $t2 , down3**

**add $t0, $0, $t2**

**down3:**

**lw $t2, 20($s0)**

**blt $t0 , $t2 , down4**

**add $t0, $0, $t2**

**down4:**

**lw $t2, 24($s0)**

**blt $t0 , $t2 , down5**

**add $t0, $0, $t2**

**down5:**

**lw $t2, 28($s0)**

**blt $t0 , $t2 , down6**

**add $t0, $0, $t2**

**down6:**

**lw $t2, 32($s0)**

**blt $t0 , $t2 , down7**

**add $t0, $0, $t2**

**down7:**

**lw $t2, 36($s0)**

**blt $t0 , $t2 , down8**

**add $t0, $0, $t2**

**down8:**

**jr $ra**