

**Course Name: Computer Architecture and Assembly Lab**

**Course Number and Section: 14:332:333:02**

**Experiment:** [Experiment # [4] – RISC-V Assembly ]

**Lab Instructor:** Jalal Abdulbaqi

**Date Performed:** 10/24/2018

**Date Submitted:** 11/6/2018

**Submitted by:** [Jahidul Islam - 171001155]

**Course Name: Computer Architecture and Assembly Lab**

**Course Number and Section: 14:332:333:02**

**! Important: Please include this page in your report if the submission is a paper submission. For electronic submission (email or Sakai) please omit this page.**

**-----For Lab Instructor Use ONLY-----**

**GRADE: \_\_\_\_\_**

**COMMENTS:**

**Electrical and Computer Engineering Department**

**School of Engineering**

**Rutgers University, Piscataway, NJ 08854**

**ECE Lab Report Structure**

- 1. Purpose / Introduction / Overview – describe the problem and provide background information**
- 2. Approach / Method – the approach took, how problems were solved**
- 3. Results – present your data and analysis, experimental results, etc.**
- 4. Conclusion / Summary – what was done and how it was done**

.

## 1. RISC-V functions (Code included separately as well)

```
1) triple:    add a0,a0,a0
              add a0,a0,a0
              jr ra
```

### In simple RISC-V given by Venus:

```
add x10,x10,x10
add x10,x10,x10
jalr x0,x1,0
```

```
2)power:     li t0, 0
              addi t1, a0, 0

loop:        bge t0, a1, end
              mul a0, a0, t1
              addi t0, t0, 1
              jal x0, loop

end:         jr ra
```

### In simple RISC-V given by Venus:

```
addi x5,x0,0
addi x6,x10,0
bge x5,x11,16
mul x10,x10,x6
addi x5,x5,1
jal x0,-12
jalr x0,x1,0
```

## 2. RISC-V Arrays and Lists

- 1) This first part of the code sets arr[1] to arr[0] + arr[2].
- 2) Now it proceeds to negate all elements in arr
- 3) Finally, for the last part it increments all values in the linked list by 1

## 3. RISC-V Calling Conventions

- 1) How do we pass arguments into functions?

In order to pass arguments into functions, use the 8 arguments registers which are a0 - a7.

**2) How are values returned by functions?**

In order to have values returned by functions, use a0 and a1 as the return value registers.

**3) What is sp and how should it be used in the context of RISC-V functions?**

The term “sp” stands for stack pointer. We have to subtract from the sp in order to create more space and add to free space. The stack is mainly used to save values of those registers that are allowed to be overwritten and it can also be restored.

**4) Which values need to be saved before using jal?**

The values that need to be saved before using jal are registers a0 - a7, t0 - t6, and ra.

**5) Which values need to be restored before using jr to return from a function?**

Before using jr to return from a function, these values need to be restored; Registers sp, gp, and s0 - s11

**4. Writing RISC-V Functions (Code included separately as well)**

sumSquare:

```
addi sp sp -12
sw ra 0(sp)
sw s0 4(sp)
sw s1 8(sp)
add s0 a0 x0
add s1 x0 x0
```

```
loop: bge x0 s0 end
      add a0 s0 x0
      jal ra square
      add s1 s1 a0
      addi s0 s0 -1
      jal x0 loop
```

```
end:  add a0,s1,x0
      lw ra 0(sp)
      lw s0 4(sp)
      lw s1 8(sp)
      addi sp sp 12
      jr ra
```