

Computer Architecture (Lab). Week 8

Vladislav, Artem, Hamza, Manuel

Innopolis University

v.ostankovich@innopolis.ru

a.burmyakov@innopolis.ru

m.rodriguez.osuna@innopolis.university

h.salem@innopolis.university

October 22, 2019

- MIPS Arithmetic Operation Summary
- Floating point operations
- Exercises

MIPS Arithmetic Instruction Summary

● Addition and subtraction

- `add`, `addi`, `sub` raise exceptions on overflow.
- `addu`, `addiu`, `subu` do not raise exceptions on overflow.
- `addi`, `addiu` both sign extend their immediate fields.

● Comparisons

- `slt`, `slti` are for signed comparisons.
- `sltu`, `sltiu` do unsigned comparisons.
- `slti`, `sltiu` both sign extend their immediate fields.

● Data transfers

- `lb`, `lbu` and `sb` for data transfer in terms of byte.
- `lw` and `sw` for loading data in terms of word.

● Shifting

- `sll`, `srl` are used for logical shifts of up to 31 bits.

● Bitwise operators

- `andi`, `ori` do not sign extend their immediate field.

Example # 1

```
# Pseudocode:  
# c = (a < b) || ((a+b) == 10)  
# Register mappings:  
# a: t0, b: t1, c: t2
```

Example # 1 – Solution

```
# Pseudocode:  
# c = (a < b) || ((a+b) == 10)  
# Register mappings:  
# a: t0, b: t1, c: t2  
  
add $t3, $t0, $t1    # tmp = a+b  
li $t4, 10           # tmp = tmp == 10  
seq $t3, $t3, $t4     # set equal  
slt $t2, $t0, $t1     # c = a < b  
or $t2, $t2, $t3      # c = c | tmp
```

Example # 2

- Compute the surface area and volume of a sphere.
- The formulas for surface area and volume of a sphere are as follows:

$$\begin{aligned}\text{surfaceArea} &= 4.0 \times \pi \times \text{radius}^2 \\ \text{volume} &= 4.0 \times \pi / 3.0 \times \text{radius}^3\end{aligned}$$

Example # 2 – Solution (1/3)

```
# Data Declarations
.data
pi: .float 3.14159
fourPtZero: .float 4.0
threePtZero: .float 3.0
radius: .float 17.25
surfaceArea: .float 0.0
volume: .float 0.0
#-----
# text/code section
.text
.globl main
main:
```

Example # 2 – Solution (2/3)

```
#-----  
# Compute:(4.0 * pi) which is used for both equations.  
l.s $f2, fourPtZero  
l.s $f4, pi  
mul.s $f4, $f2, $f4 # 4.0 * pi  
l.s $f6, radius # radius  
  
#-----  
# Calculate surface area of a sphere.  
# surfaceArea = 4.0 * pi * radius^2  
mul.s $f8, $f6, $f6 # radius^2  
mul.s $f8, $f4, $f8 # 4.0 * pi * radius^2  
s.s $f8, surfaceArea # store final answer
```


Example # 2 – Solution (3/3)

```
#-----
# Calculate volume of a sphere.
# volume = (4.0 * pi / 3.0) * radius^3
l.s $f8, threePtZero
div.s $f2,$f4, $f8      # (4.0 * pi / 3.0)
mul.s $f10, $f6, $f6    # radius^2
mul.s $f10, $f10, $f6    # radius^3
mul.s $f12, $f2, $f10    # 4.0*pi/3.0*radius^3
s.s $f12, volume        # store final answer
#-----
# Done, terminate program.
li $v0, 10              # terminate call code
syscall                 # system call
.end main
```

Example # 3

- Compare and branch example

```
.data
    pi: .float 3.1415
    alert: .asciiz "Greater than zero"
.text
    mtc1 $zero, $f0 # move zero value to $f0
    l.s $f1, pi      # load pi into $f1
    c.lt.s $f0, $f1 # compare
    bc1t pi_greater_than_zero # branch if 0 < pi
    j end
pi_greater_than_zero:
    li $v0, 4        # print message
    la $a0, alert
    syscall
end:
```

Exercise 1

- Write a MIPS code that asks for Fahrenheit temperature and returns the Celsius value which is printed on the console. The conversion formula is:

$$\text{Celsius} = (\text{Fahrenheit} - 32.0) \times 5.0 / 9.0$$

Exercise 2: Recursive Function

Write a **recursive** function that will calculate the sum of natural numbers.

Exercise 3 * Optional

Write a **function** that calculates the following expression:

$$\frac{e^2}{\pi i}$$

Hints:

- Assume $e = 2.71828$, $\pi = 3.1415$

Useful Links

- <https://courses.cs.washington.edu/courses/cse378/09wi/lectures/lec05.pdf>
- <http://people.cs.pitt.edu/~xujie/cs447/Mips/sub.html>
- <https://www.youtube.com/watch?v=y9Wv1RVbbNA>
- http://chortle.ccsu.edu/assemblytutorial/Chapter-25/ass25_8.html
- https://chortle.ccsu.edu/AssemblyTutorial/Chapter-31/ass31_1.html
- https://chortle.ccsu.edu/AssemblyTutorial/Chapter-32/ass32_1.html

Acknowledgements

- This lab was created and maintained by Vitaly Romanov, Aidar Gabdullin, Munir Makhmutov, Ruzilya Mirgalimova, Muhammad Fahim, Vladislav Ostankovich, Alena Yuryeva and Artem Burmyakov