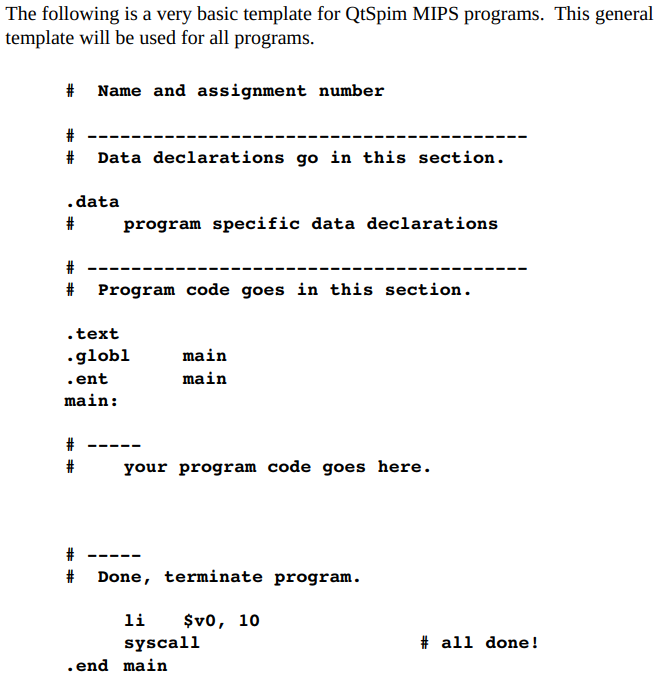
**MIPS SUMMARY**

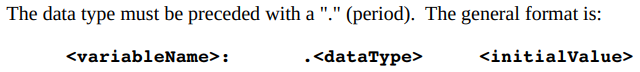
**(ASSEMBLY)**

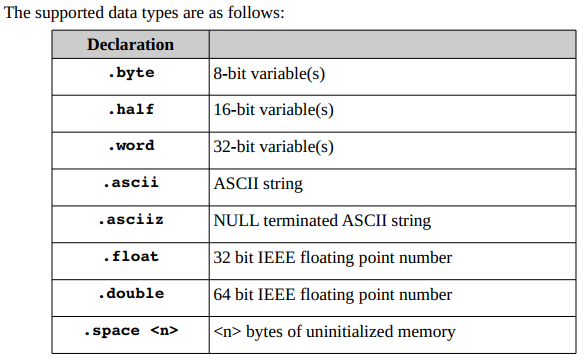
**Link tải phần mềm QTSpim:** [**Tại đây**](https://drive.google.com/file/d/12C6XUxFXr8cKqyvWHkZ4vuTR3MLWRAqx/view?usp=sharing)

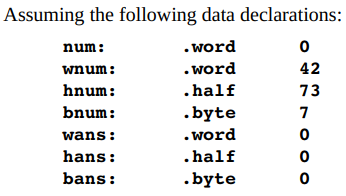
**1. Program Template**

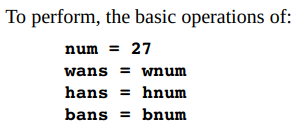
****

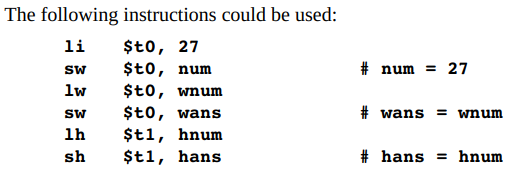
**2. Data Declarations**

****

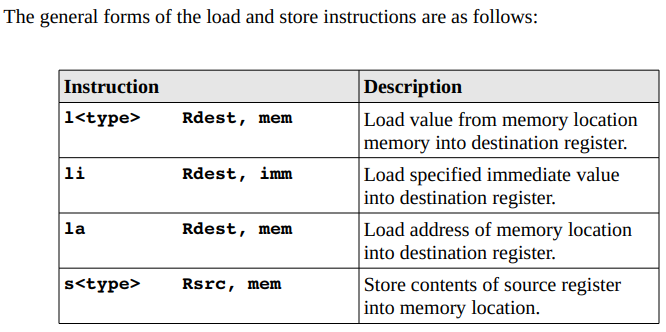
****

******

******



**3. Load / Store Instructions**

****

1. ***load: loading of data from memory (e.g., variables or arrays) into registers***

* ***syntax:***

*l <type> register\_destination*, *RAM\_source* *# Load integer number*

*l****.*** *<type> register\_destination, RAM\_source* *# Load float number*

1. ***load immediate:***

* ***syntax:***

*li register\_destination, value # Load immediate of a integer number*

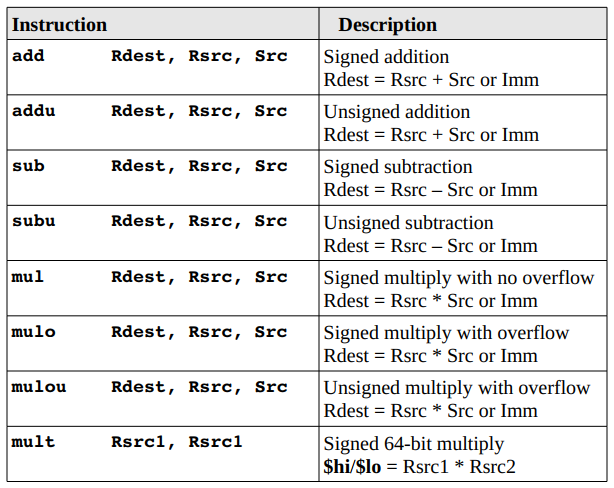
*li.<type> register\_destination, value* *# Load immediate of a float number*

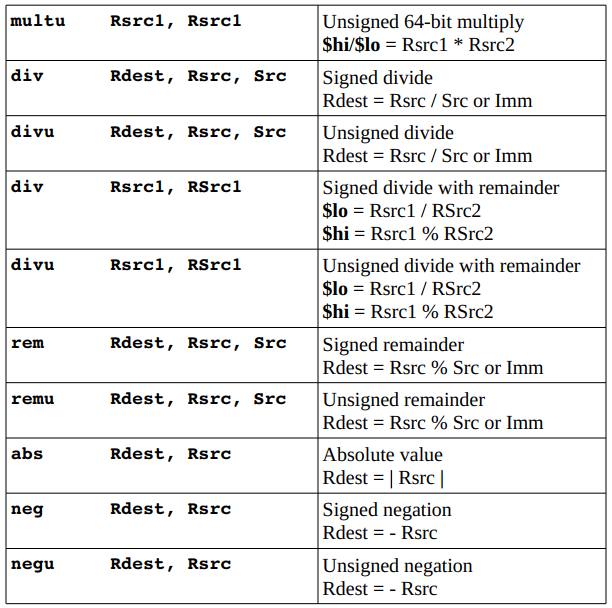
1. ***store: storing of data in register back to memory***

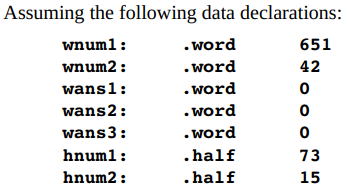
* ***syntax:*** *s <type> register\_source, RAM\_destination* *# Store integer number*

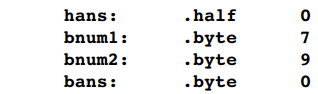
*s* ***.*** *<type> register\_source, RAM\_destination # Store float number*

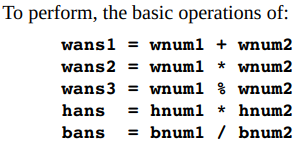
**4. Integer Arithmetic Operations**

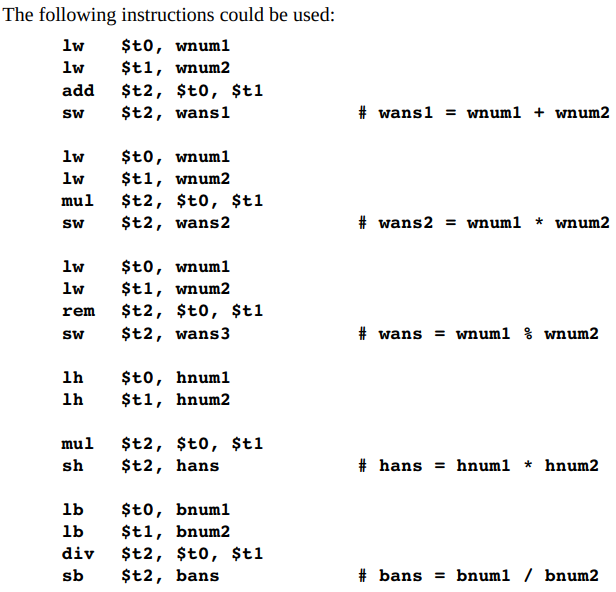
****

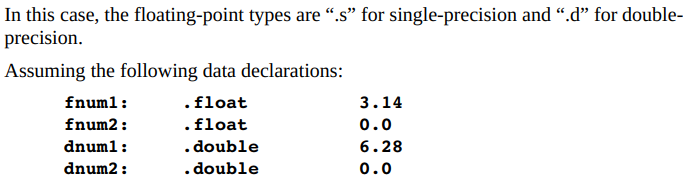
****

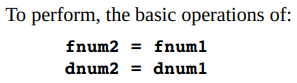
****

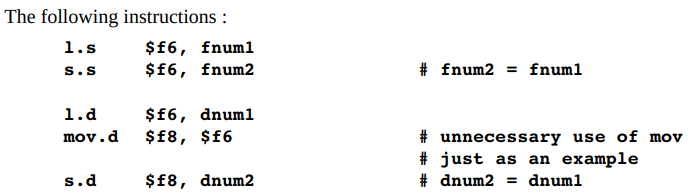
****

****

****

****

****

****

**5. Convert số nguyên thành số thực (Xem chi tiết mục 5.7.3)**

Giả sử để chia 2 số nguyên x, y và kết quả sau khi chia phải hiển thị là kiểu số thực thì chúng ta phải convert 2 số nguyên x, y thành số thực trước khi chia.

**VD:**

lw $t1, 3 # y=2

lw $t2, 2 # x=3

mtc1 $t1, $f1 # switch register $t1 to $f1

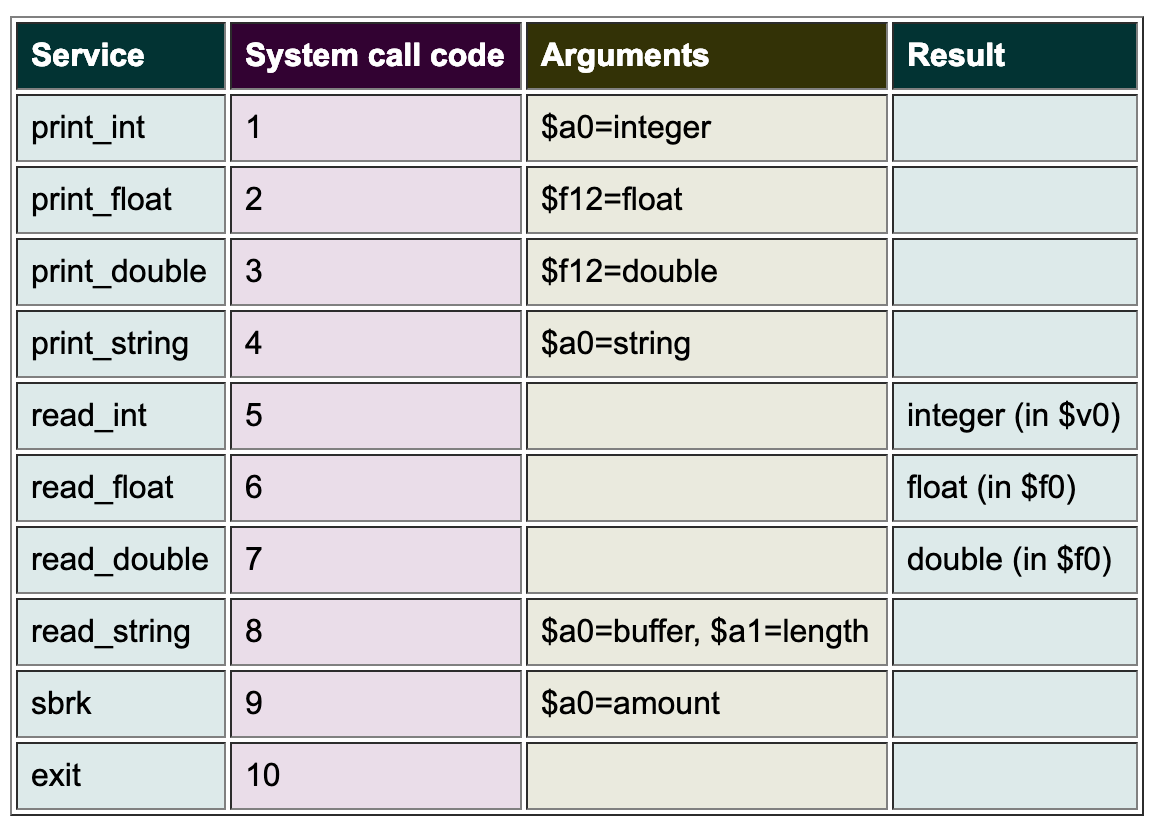
mtc1 $t2, $f2 # switch register $t2 to $f2

cvt.s.w $f1, $f1 # convert integer $f1 to float $f1

cvt.s.w $f2, $f2 # convert integer $f2 to float $f2

div.s $f3, $f2, $f1 # Divide float $f3 = x/y

**6. System Calls and I/O**



* **Read a integer number**

*li $v0, 5 # load code 5 vào $v0 để hệ thống nhận biết nhập số nguyên*

*syscall # Gọi hệ thống để nhập số nguyên và lưu số vừa nhập vào thanh ghi $v0*

* **Read a float**

*li $v0, 6 # load code 6 vào $v0 để hệ thống nhận biết nhập số float*

*syscall # Gọi hệ thống để nhập số float và lưu số vừa nhập vào thanh ghi $f0*

* **Read a double**

*li $v0, 7 # load code 7 vào $v0 để hệ thống nhận biết nhập số double*

*syscall # Gọi hệ thống để nhập số double và lưu số vừa nhập vào thanh ghi $f0*

* **Print a integer**

*li $v0, 1 # load code 1 vào $v0 để hệ thống nhận biết xuất số nguyên*

*lw $a0, number # Load số cần xuất vào thanh ghi $a0*

*syscall # Gọi hệ thống để xuất giá trị chưa trong thanh ghi $a0*

* **Print a float**

*li $v0, 2 # load code 2 vào $v0 để hệ thống nhận biết xuất số float*

*mov.s $f12, $f0 # Di chuyển giá trị trong $f0 vào $f12*

*syscall # Gọi hệ thống để xuất giá trị float chứa trong thanh ghi $f12*

* **Print a double**

*li $v0, 3 # load code 3 vào $v0 để hệ thống nhận biết xuất số double*

*mov.s $f12, $f0 # Di chuyển giá trị trong $f0 vào $f12*

*syscall # Gọi hệ thống để xuất giá trị double chứa trong thanh ghi $f12*

* **Print a string**

*li $v0, 4 # load code 4 vào $v0 để hệ thống nhận biết xuất chuỗi*

*la $a0, msg # Load địa chỉ của chuỗi cần xuất vào thanh ghi $a0*

*syscall # Gọi hệ thống để xuất giá trị chuỗi chứa trong thanh ghi $a0*

* **Exit**

*li $v0, 10 # load code 10 vào $v0 để hệ thống nhận biết thoát chương trình*

*syscall # Gọi hệ thống để thoát*

**Ví dụ 1: Viết chương trình Xuất một chuỗi “Xin chào các bạn” ra màn hình**

Mở Editor soạn thảo code, lưu lại với đuôi .s (VD mở Notepad++, soạn code và Save: Bai1.s)

*.data*

*input: .asciiz "Xin chao cac ban"*

*.text*

*.globl main*

*main:*

*# print input*

*li $v0, 4 # Load code =4 vào thanh ghi $v0 để hệ thống biết print chuỗi*

*la $a0, input # Load địa chỉ của chuỗi input cần xuất vào $a0*

*syscall # Gọi hệ thống để xuất nội dung trong $a0 ra*

*# exit*

*li $v0,10 # Load code =10 vào thanh ghi $v0 để hệ thống biết là exit*

*syscall # Gọi hệ thống để exit*

**Ví dụ 2: Viết chương trình nhập vào 2 số nguyên. Tính tổng và xuất kết quả**

Mở Editor soạn thảo code, lưu lại với đuôi .s (VD mở Notepad++, soạn code và Save: Bai1.s)

.data

msg: .asciiz "Nhap 2 so nguyen de tinh tong, cach nhau boi Enter: \n"

.text

.globl main

main:

# Print string msg

li $v0, 4

la $a0, msg

syscall

# Nhap so nguyen thu nhat

li $v0, 5 # Load code =5 vào thanh ghi $v0 để hệ thống biết là nhập số nguyên

syscall # Gọi hệ thống để cho phép nhập số nguyên, và lưu số vừa nhập vào $v0

move $t1, $v0 # Di chuyển giá trị của thanh ghi $v0 vào $t1 ($t1 = số thứ 1)

#Nhao so nguyen thu hai

li $v0, 5 # Load code =5 vào thanh ghi $v0 để hệ thống biết là nhập số nguyên

syscall # Gọi hệ thống để cho phép nhập số nguyên, và lưu số vừa nhập vào $v0

move $t2, $v0 # Di chuyển giá trị của thanh ghi $v0 vào $t2 ($t2=số thứ 2)

# Tính tổng

add $t3, $t1, $t2 # $t3 = $t1+$t2

#print a interger

li $v0, 1 # Load code=1 để hệ thống nhận biết print a integer

move $a0, $t3 # Di chuyển giá trị trong $t3 vào $a0 để xuất ra

syscall # Gọi hệ thống để xuất giá trị trong $a0 ra ngoài

# Thoát

li $v0, 10

syscall

**Ví dụ 3: Viết chương trình nhập vào 02 số thực. Tính tổng 02 số thực vừa nhập**

.data

msg: .asciiz "Nhap 2 so thuc de tinh tong, cach nhau boi Enter: \n"

.text

.globl main

main:

# Print string msg

li $v0, 4

la $a0, msg

syscall

# Nhap so float thu 1

li $v0, 6 # Load code =6 vào thanh ghi $v0 để hệ thống biết là nhập số float

syscall # Gọi hệ thống để cho phép nhập số float, và lưu số vừa nhập vào $f0

mov.s $f1, $f0 # Di chuyển giá trị của thanh ghi $f0 vào $f1 ($f1 = số thứ 1)

#Nhao so float thu 2

li $v0, 6 # Load code =6 vào thanh ghi $v0 để hệ thống biết là nhập số float

syscall # Gọi hệ thống để cho phép nhập số float, và lưu số vừa nhập vào $f0

mov.s $f2, $f0 # Di chuyển giá trị của thanh ghi $f0 vào $f2 ($f2=số thứ 2)

# Tính tổng

add.s $f3, $f1, $f2 # $f3 = $f1+$f2

#print float

li $v0, 2 # Load code=2 để hệ thống nhận biết print a float

mov.s $f12, $f3 # Di chuyển giá trị trong $f3 vào $f12 để xuất ra

syscall # Gọi hệ thống để xuất giá trị trong $f12 ra ngoài

# Thoát

li $v0, 10

syscall

**Ví dụ 3: Viết chương trình tính diện tích, chu vi**

# 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

.globl main

main:

l.s $f2, fourPtZero

l.s $f4, pi

mul.s $f4, $f2, $f4 # 4.0 \* pi

l.s $f6, radius # radius

mul.s $f8, $f6, $f6 # radius^2

mul.s $f8, $f4, $f8 # 4.0 \* pi \* radius^2

s.s $f8, surfaceArea # store final answer

l.s $f8, threePtZero

div.s $f2, $f4, $f8 # (4.0 \* pi / 3.0)

mul.s $f10, $f2, $f2

mul.s $f10, $f10, $f6 # radius^3

mul.s $f12, $f6, $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