Lab Sheet 3 for CS F342 Computer Architecture
Semester 1 : 2023-24

Explanation: the basic program in MIPS (Storing and loading to and from memory, Use of mfhi and mflo registers, and arithmetic instructions

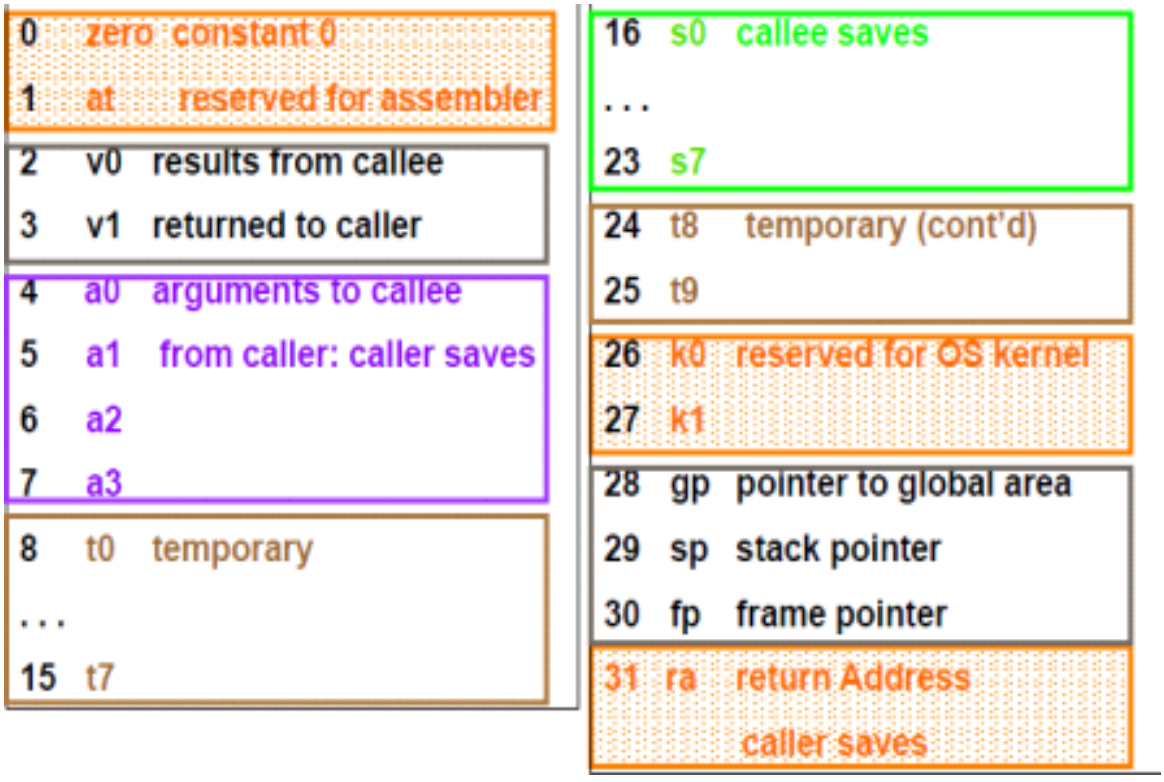
Goal : To get introduced to QTSPIM and implement some code related to - System calls and instruction for reading integer input and store it in memort and registers.

Reference for MIPS assembly – refer to the **MIPS Reference Data Card ("Green sheet")** uploaded in CMS. Further, some of the QTSPIM assembly instructions are beyond this data card (e.g. the pseudo instruction ***la***).

Additionally use Appendix A (HP_AppA.pdf) from Patterson and Hennessey "Assemblers, Linkers and the SPIM Simulator" for gaining background knowledge of SPIM.

In this lab we focus on reversing only integer based instructions

(add, or, subi etc.). **Reference for Registers:**



System calls as well as functions (in later part of the semester) should take care of using the registers in proper sequence. Especially take note of V0, V1 [R2, R3 in QTSPIM] and a0-a3 [R4- R7 in QTSPIM] registers.

Reference for System Calls:

Service	Code (put in $v0)	Arguments	Result
print_int	1	$a0=integer	
print_float	2	$f12=float	
print_double	3	$f12=double	
print_string	4	$a0=addr. of string	
read_int	5		int in $v0
read_float	6		float in $f0
read_double	7		double in $f0
read_string	8	$a0=buffer, $a1=length	
sbrk	9	$a0=amount	addr in $v0
exit	10		

Reference for Data directives:

.word w1, …, wn
-store n 32-bit quantities in successive memory words

.half h1, …, hn
-store n 16-bit quantities in successive memory half words

.byte b1, …, bn
-store n 8-bit quantities in successive memory bytes

.ascii str

-store the string in memory but do not null-terminate it
-strings are represented in double-quotes “str”
-special characters, eg. \n, \t, follow C convention

.asciiz str
-store the string in memory and null-terminate it

.float f1, …, fn
-store n floating point single precision numbers in successive memory locations

.double d1, …, dn
-store n floating point double precision numbers in successive memory locations

.space n
-reserves n successive bytes of space

Class Exercises

1. Declare two 32-bit integers in the data segment using the appropriate data directive. Then, ask the user to add the integers and then print the respective integer after loading it from memory.

Hint: Declare two memory variables in data section say
num1: .word 20
num2: .word 10

Use store word and load word instruction, store is used for storing from reg to mem and load from mem to register.

2) Take two integers A, B as input and print their products(A*B), quotients(A/B) and remainders(A%B)

Hint: Use hi and lo registers.

3) Take an integer length and a string as input, the string should be truncated at the input length, then print this string. Note: this length includes the null character.

Hint: Use a1 register to fix the length of the string.

4) Write a program that takes two integer inputs, A and B, from the user. Compare the product of A and B with a predefined integer value. If the product equals the predefined integer, the program should return 0. Otherwise, it should return the remainder of the product when divided by the predefined integer.

Practice Questions:

1) Make a calculator, given two operands from the user at run time, perfrom addition, subtraction, multiplication and division operation.

2) Calculate the average of first N consecutive positive integer numbers and print the output, take the value on N at run time by the user.