

# Mini project 1

## Introduction

In this project, you will be writing assembly language programs that compute Fibonacci numbers and factorial. The basic MIPS assembly language instructions should be familiar to you after reading Chapter 2 of *Computer Organization and Design*. Refer to Appendix B for a complete list of all the available instructions along with descriptions of their functions.

## Fibonacci Numbers

Recall that each number in the Fibonacci series is the sum of the previous two numbers. Here are the first few numbers in the series:

<b>n</b>	1	2	3	4	5	6	7	8	9	10	...
<b>fib(n)</b>	0	1	1	2	3	5	8	13	21	34	...

We can also define the fib function for negative values of  $n$ . To be consistent with the definition of the Fibonacci series, what would the values of fib(0) and fib(-1) be? These values are useful when writing a loop to compute fib( $n$ ) for all non-negative values of  $n$ .

To execute the fib program that you write, you will be using the PCSpim application. PCSpim allows you to step through one line of code at a time while displaying the values of all registers and memory, so that you can watch what the program is doing and catch any bugs that appear. PCSpim is included on the companion CD; see the Software page on the CD if you have not already installed this software.

## Writing the Code

For a template for the assembly program that you should write refer to SPIM lecture notes available on the web page of the course. To get you started, here is a template for the assembly program that you should write:

```
main:  addi $a0,$0,8           # n = 8
      addi $t0,$0,-1       # jump-start loop with
      addi $t1,$0,1        # fib(-1) and fib(0)
      loop: <add your code here>  # <your comments here>
      ...
done:  j done # infinite loop
```

Use Notepad (or your favorite text editor) to enter this code into a text file named fibo\_xx.asm (where xx are your initials). Note that the first line of code places the value 8 into register a0. This is the  $n$  argument to the fib function. You should test your program on several different values (not just 8). The third and fourth lines of code load the values of fib(-1) and fib(0) into registers t0 and t1, respectively. As mentioned above, these can be used to “bootstrap” your loop. After looping the number of times specified by the value placed in register a0, your code should leave the resulting value of fib( $n$ ) in register \$v0 and branch to done. The last line of code is simply a way of ensuring that your program terminates in a predictable manner.

Comments begin with the ‘#’ character and continue to the end of the line. You should use comments to include your name and the date at the top of the file. Please add comments after every line of code also, explaining clearly what the code does. Follow the steps explained during SPIM lecture (available on the web page of the course) and run your project on the SPIM. Use system calls if required. Submit your results and your code by February 25<sup>th</sup>.

## Factorial

Write an assembly language program to compute the factorial function: fact( $n$ ) =  $n!$ . Name your text file fac\_xx.asm (where xx are your initials). Submit your results and your code by February 25<sup>th</sup>.