Homework #2  
COMPSCI 104 - Computer Organization, Design and Programming  
Due in class, February 10th 2005

PART 1: Problems – 30 Points Total

1. [10 points] Using standard MIPS instructions write code that has the same effect as:

   jal target_proc

   Do not use the “jal” or “j” instructions.

2. [10 points] Problem 2.29 from P&H.

3. [10 points] Write the minimum number of MIPS instructions that have the same effect as:

   nor $t0, $s0, $s0  
   addi $t0, $t0, 1  
   add $s2, $s1, $t0

   Do not use “nor”, “add” or “addi” instructions.

PART 2: Programming – 70 Points Total

Pack the source code for all problems in a single zip archive and submit it using the DropBox in the COMPSCI 104 Blackboard page. Add a README file if necessary. If Blackboard/DropBox is offline, send it to the TA by e-mail.

You should use SPIM or XSPIM to execute and debug your program. We will run each program through SPIM when grading. The CD that accompanies the P&H book has a good amount of information on SPIM that you can use. The simulator itself as well as some tutorials can also be found at http://www.cs.wisc.edu/~larus/spim.html.

To receive full credit, the code should be well commented.

P1. [30 points] Write a MIPS program that reads two integer values (one at a time) and then prints the one with the largest value. If they are equal, it should print one of the two. The program should implement an user interface similar to the following:

   Enter first integer: 92  
   Enter second integer: 5  
   The larger integer is: 92

P2. [40 points] A Fibonacci sequence is an infinite sequence of numbers where each element is the sum of its two preceding numbers:

   0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Below is a C function that computes the n-th number in the sequence:

```c
int Fibonacci (int n) {
  if (n == 0) return 0;
  if (n == 1) return 1;
  return Fibonacci(n-1) + Fibonacci(n-2);
}
```

Write a MIPS program that implements the above function. Your program should read the value of “n” from the user and then display the n-th Fibonacci number.

Example:

```
Enter term to compute: 7
The term is: 13
```

You must implement the recursive algorithm in order to receive credit. You do not need to make speed optimizations.