Lab 05

September 28, 2016

MIPS loops (again)

Implement a function `sum(n)` that sums up all integers from `0` to `n` (exclusive) using a loop. Use your knowledge of MIPS and the MIPS-related PDFs from Piazza (in the Resources section) to code up this solution. Use `helloworld.s` from the last lab to get you started.

FizzBuzz

Using MIPS assembly, write the Fizz Buzz program. That is, print the integers 1 through 20 with the following conditions: if the number is divisible by 3, output “Fizz” instead of the number, “Buzz” if it’s divisible by 5, and “FizzBuzz” if it is divisible by both 3 and 5. Division in MIPS can be tricky: one has to use the `div` instruction like so: `div $t0, $t1`. This looks unlike any other instruction we have seen as there is no destination register. The `div` instruction produces two results: the quotient and the remainder. These values are stored *outside* of the normal registers in registers named `HI` and `LO`. After the `div` instruction `LO` will contain the quotient and `HI` will contain the remainder. You can access them with the `mfhi` and `mflo` instructions (move from `HI` and move from `LO`, respectively). The following code snippet divides 5 by 3: 1 with a remainder of 2.

```
li   $t0, 5
li   $t1, 3
div  $t0, $t1
mfhi $t2
mflo $t3
```

Fibonacci in MIPS

Implement the `fib(n)` function in MIPS that returns the nth Fibonacci number using recursive function calls. The example from the slide, “Non-Leaf Procedure Example” may be useful.