COE 308 – Computer Architecture

Assignment 2 : MIPS Instructions and Assembly Language

Due Sunday, October 1st, 2006

1. (3 pts) Add comments to the following MIPS code and describe in one sentence what it computes. Assume that `$a0` is used for the input and initially contains `n`, a positive integer. Assume that `$v0` is used for the output.

```
begin:    addi $t0, $zero, 0
          addi $t1, $zero, 1
loop:     slt  $t2, $a0, $t1
          bne  $t2, $zero, finish
          add  $t0, $t0, $t1
          addi $t1, $t1, 2
          j    loop
finish:   add  $v0, $t0, $zero
```

2. (4 pts) The following code fragment processes an array and produces two important values in registers $v0 and $v1. Assume that the array consists of 5000 words indexed 0 through 4999, and its base address is stored in $a0 and its size (5000) in $a1. Describe in one sentence what this code does. Specifically, what will be returned in $v0 and $v1?

```
          add  $a1, $a1, $a1
          add  $a1, $a1, $a1
          add  $v0, $zero, $zero
          add  $t0, $zero, $zero
outer:    add  $t4, $a0, $t0
          lw   $t4, 0($t4)
          add  $t5, $zero, $zero
          add  $t1, $zero, $zero
inner:    add  $t3, $a0, $t1
          lw   $t3, 0($t3)
          bne  $t3, $t4, skip
          addi $t5, $t5, 1
skip:     addi $t1, $t1, 4
          bne  $t1, $a1, inner
          slt  $t2, $t5, $v0
          bne  $t2, $zero, next
          add  $v0, $t5, $zero
          add  $v1, $t4, $zero
next:     addi $t0, $t0, 4
          bne  $t0, $a1, outer
```

3. (5 pts) For each pseudo-instruction in the following table, produce a minimal sequence of actual MIPS instructions to accomplish the same thing. You may use the `$at` for some of the sequences. In the following table, `big` refers to a specific number that requires 32 bits to represent and `small` to a number that can be expressed using 16 bits.

Pseudo-instruction
`move $t1, $t2`
`clear $t5`
`li $t5, small`
`li $t5, big`
`lw $t5, big($t3)`
`addi $t5, $t3, big`
`beq $t5, small, L`
`beq $t5, big, L`
`ble $t5, $t3, L`
`bgt $t5, $t3, L`
`bge $t5, $t3, L`

4. (5 pts) Consider the following fragment of C code:

```
for (i=0; i<=100; i=i+1) { a[i] = b[i] + c; }
```

Assume that `a` and `b` are arrays of words and the base address of `a` is in `$a0` and the base address of `b` is in `$a1`. Register `$t0` is associated with variable `i` and register `$s0` with `c`. Write the code in MIPS. How many instructions are executed during the running of this code? How many memory data references will be made during execution?

5. (3 pts) Write a procedure, `bfind`, in MIPS assembly language. The procedure should take a single argument that is a pointer to a null-terminated string in register `$a0`. The `bfind` procedure should locate the first `b` character in the string and return its address in register `$v0`. If there are no `b`'s in the string, then `bfind` should return a pointer to the null character at the end of the string.