

Homework I

Tom Kelliher, CS 220

Due Sept. 14, 2007

1. (10 pts.) 1.57
2. (10 pts.) Using the MIPS program in Exercise 2.34 (with bugs intact), determine the instruction format for each instruction and the decimal values of each instruction field.
3. (5 pts.) 2.2
4. (5 pts.) 2.3
5. (10 pts.) Consider the following code to implement the pseudo-instruction

```
sllv $s0, $s1, $s2
```

which uses the least significant five bits of the value in register `$s2` to specify the amount `$s1` should be shifted left:

```
mask:      .data
           .word 0xffff83f
           .text
start:     lw      $t0, mask
           lw      $s0, shifter
           and     $s0, $s0, $t0
           andi   $s2, $s2, 0x1f
           sll    $s2, $s2, 6
           or     $s0, $s0, $s2
           sw     $s0, shifter
shifter:   sll    $s0, $s1, 0
```

Add comments to the code and write a paragraph describing how it works. Note that the two `lw` instructions are pseudo-instructions that use a label to specify a memory address that contains the word of data to be loaded. (Similarly for the `sw` instruction.) Why do you suppose that writing “self-modifying code” such as this is a bad idea (and oftentimes not actually allowed)?

6. (10 pts.) The MIPS translation of the C segment

```
while (save[i] == k)
    i += 1;
```

on page 74 uses both a conditional branch and an unconditional jump each time through the loop. Only poor compilers would produce code with this loop overhead. Rewrite the assembly so that it uses at most one branch or jump each time through the loop. How many instructions are executed before and after the optimization if the number of iterations of the loop is 10 (i.e., `save[i + 10]` does not equal `k` and `save[i], ..., save[i + 9]` equal `k`)?