Rose-Hulman Institute of Technology
Computer Science and Software Engineering
Profs. Laurence Merkle and Archana Chidanandan

Homework 2 Solutions (Assembly and Machine Languages)

This assignment is due Monday, September 29, 2003 for Sections 1 and 3 and on Tuesday, September 30, 2003 for Section 2. Submit your solutions on a separate sheet of paper.

Learning Objectives

In the process of completing this homework assignment, students will develop their abilities to

- Translate assembly language instructions into machine language.
- Interpret a sequence of bits and determine what it represents.
- Determine addressing in branches and jumps.
- Interpret instruction formats and fields.

Problems

1. [5 pts] List the machine language fields and their binary values for these MIPS instructions:

```
add    $t0, $s0, $s1
op    = 000000
rs    =  10000
rt    =  10001
rd    =  01000
shamt =  00000
funct = 100000

addi   $t0, $s0, 150
op    = 001000
rs    =  10000
rt    =  01000
imm   = 0000 0000 1001 0110

lw     $t0, 150($s0)
op    = 100011
rs    =  10000
rt    =  01000
imm   = 0000 0000 1001 0110

slt    $t0, $s0, $s1
op    = 000000
rs    =  10000
rt    =  10001
rd    =  01000
shamt =  00000
funct = 101010
```

```
jr      $t0
op    = 000000
rs    =  01000
rt    =  00000
rd    =  00000
shamt =  00000
funct = 001000
```

2. [4 pts] Assume that the MIPS instruction `beq $t0, $s0, Label` is located at address `0x0400 1234`, and that `Label` is located at address `0x0400 5678`. What will the binary value of the address field be? *Hint*: remember that the offset is relative to the instruction following the branch, and that all branch targets must be word aligned.

   The instruction branches forward 0x4444 bytes, i.e. 0x4440 bytes = 0x1110 instructions relative to the next instruction. Thus, the address field will contain

   `0001 0001 0001 0000`

3. [4 pts] Assume that the MIPS instruction `j Label` is located at address `0x0400 1234`, and that `Label` is located at address `0x0400 5678`. What will the binary value of the target address field be? *Hint*: remember that all branch targets must be word aligned.

   The jump target is on the same "page" as the instruction following the jump, so the problem is well posed. The address field will contain bits 27 – 2 of the label's address, i.e.

   `0100 0000 0101 0110 0111 10`

4. [2 pts] What sequence of two MIPS instructions starting at address `0x0400 1234` could be used to branch to address `0x4400 5678`?

   The jump target is not on the same "page" as the instruction following the starting address (or the one after it), so the `j` instruction will not work, and the two instruction sequence must specify all 32 bits of the jump target address. The following sequence will work:

```
li   $t0, 0x4400 5678
jr   $t0
```

5. [4 pts] A computer has a 32-bit word length, and all the instructions are one word in length. Every instruction has a two bit field which specifies the instruction type. The number of registers (the register file) in the architecture of the computer is 64.
   a. For a format that has an opcode field to determine the function of the instruction and three register fields, what is the maximum number of opcodes possible?

      The type field requires 2 bits, and each of the three register fields requires lg 64 = 6 bits, so there are 32 – 2 – 18 = 12 bits left for the opcode field. Thus, there are 2^{12} possible opcodes (i.e. 4096).

   b. For a format with two register fields, one immediate/address field, and a maximum of 256 opcodes, what is the maximum number of bits available for the immediate/address field?

      The type field requires 2 bits, the opcode field requires lg 256 = 8 bits, and each of the two register fields requires 6 bits, so there are 32 – 2 – 8 – 12 = 10 bits left for the immediate/address field.

6. [6 pts] Bits have no inherent meaning. However, given the rules to interpret them, a sequence of bits can represent different values and have meaning. Given the following 32-bit pattern:

   1000 1101 0010 1001 0000 0000 0000 0000

   what does it represent, assuming that it is

   a. a two's complement integer?

      The sign bit is negative. To find the additive inverse, we can "flip all the bits"and add one, giving 0111 0010 1101 0111 0000 0000 0000 0000, which represents 1,926,692,864, so the original bit pattern represents

      -1,926,692,864

   b. an unsigned integer?

      2,368,274,432

   c. a MIPS instruction?

      The opcode is 100011, which represents 35 decimal, so this is a lw instruction. Regrouping into I-type fields gives

      100011 01001 01001 0000000000000000

      The rs and rt fields are both 01001, which represents 9 decimal, i.e. $t1. The immediate field is zero. Thus, the instruction is

      ```
      lw $t1, 0($t1)
      ```