

CSCI 2321 (Computer Design), Spring 2021

Homework 2

Credit: 30 points.

1 Reading

Be sure you have read, or at least skimmed, the assigned readings from Chapter 2 up through 2.7.

2 Problems

Answer the following questions. You may write out your answers by hand and scan them, or you may use a word processor or other program, but please submit a PDF or plain text via e-mail to my TMail address. (No links to shared files on Google Drive please.) Please use a subject line that mentions the course and the assignment (e.g., “csci 2321 hw 2” or “computer design hw 2”).

Tips:

- If a question requires you to do calculations, please show enough work to help me understand how you got the answer you did, so if you make a mistake I can give partial credit for anything you did get right.
- For some of the problems you’re asked to do something that involves converting between hexadecimal (base 16) and binary. Remember that this is a straightforward process for which you don’t need a calculator — each hexadecimal digit represents four binary digits. (There’s a discussion of this at the start of section 2.5 of the textbook.)
- Some problems ask you to write MIPS code. You can find a complete list of instructions in the appendix, section A-10, *but note* that this list includes many pseudoinstructions, which for these problems you should not use. (I want you to learn the instructions actually covered in this chapter first.)

1. (5 points) Translate the following line of C into MIPS assembler:

```
B[6] = A[i] + A[j];
```

assuming that

- all variables are 32-bit integers,
 - `i` and `j` are associated with registers `$s1` and `$s2`, and
 - `$s3` and `$s4` contain the base addresses of `A` and `B` respectively.
2. (5 points) For each of the following MIPS instructions, translate it into machine language, first listing all the fields (e.g., opcode) in binary and then giving the 32-bit instruction in hexadecimal.

- `sub` `$t0, $s1, $s2`
- `addi` `$s1, $s1, -1`

(We didn't do an example of `addi` in class, but like `lw` and `sw` it's an I-format instruction.)

3. (5 points) Given a machine-language instruction `0x02108020` (`0x` denotes a base-16 value), what is the corresponding MIPS assembler-language instruction?
4. (5 points) Given the following initial contents for registers `$t1` and `$t2`:

```
$t1 0xFFFFFFFF
$t2 0x12345678
```

For each of the following sequences of MIPS instructions, if `$t1` and `$t2` are as above, what does `$t0` contain, in hexadecimal, after it is executed?

- `sll` `$t0, $t1, 16`
 `and` `$t0, $t0, $t2`
- `srl` `$t0, $t1, 16`
 `or` `$t0, $t0, $t2`
- `sra` `$t0, $t1, 16`
 `and` `$t0, $t0, $t2`
- `ori` `$t0, $t2, 0x00FF`

(Assume that the assembler is smart enough to convert `0x00FF` to an appropriate 16-bit constant.)

5. (10 points) Reverse-compile the following MIPS assembly code into equivalent C (without use of `goto`), using integer variable `i` to represent the value in `$t1` and integer variable `result` to represent the value in `$s2`. (You can use `int` to represent a 32-bit integer; that's what a "word" is in MIPS. So `MemArray` is an array of 100 ints.)

```

        .text
        addi    $t1, $0, 0
        la     $s0, MemArray
        addi    $s2, $0, 0
LOOP:   lw     $s1, 0($s0)
        add    $s2, $s2, $s1
        addi    $s0, $s0, 4
        addi    $t1, $t1, 1
        slti   $t2, $t1, 100
        bne    $t2, $0, LOOP

```

```

        .data
MemArray: .space 400 # reserve space for 400 bytes, i.e., 100 words

```

3 Pledge

For programming assignments, this section should go in the body of the e-mail or in a plain-text file `pledge.txt` (no word-processor files please). For written assignments, please put it in the text or PDF file with your answers.

Include the Honor Code pledge or just the word “pledged”, *plus* at least one of the following about collaboration and help (as many as apply). Text *in italics* is explanatory or something for you to fill in.

- I did not get outside help *aside from course materials, including starter code, readings, sample programs, the instructor.*
- I worked with *names of other students* on this assignment.
- I got help with this assignment from *source of help — ACM tutoring, another student in the course, etc. (Here, “help” means significant help, beyond a little assistance with tools or compiler errors.)*
- I got help from *outside source — a book other than the textbook (give title and author), a Web site (give its URL), etc.. (Here too, you only need to mention significant help — you don’t need to tell me that you looked up an error message on the Web, but if you found an algorithm or a code sketch, tell me about that.)*
- I provided help to *names of students* on this assignment. *(And here too, you only need to tell me about significant help.)*

4 Essay

For programming assignments, this section should go in the body of the e-mail or in a plain-text file `pledge.txt` (no word-processor files please). For written assignments, please put it in the text or PDF file with your answers.

Include a brief essay (a sentence or two is fine, though you can write as much as you like) telling me what if anything you think you learned from the assignment, and what if anything you found interesting, difficult, or otherwise noteworthy.