Institut for Matematik og Datalogi
Syddansk Universitet

September 2, 2008
JFB

Introduction to Computer Science
E08 – Week 2

Lecture, August 26

Rolf Fagerberg lectured. He began with an introduction to the course, covering chapter 0 in the textbook, but skipping section 0.2.

Lecture, August 27

Kim Skak Larsen lectured. He covered through section 1.7 of chapter 1 in the textbook. The IEEE standard for floating point numbers was included.

Lecture, September 1

We will start with chapters 2 in the textbook and return to chapter 1 if there is time.

Lecture, September 3

We will finish chapter 1 in the textbook and begin on chapter 3.

Lecture, September 8

We will continue with chapter 3 and introduce LaTeX.

Laboratory: first in week 37. Meet in IMADA's terminal room with your login information.

Discussion in groups (only two, or possibly three, people per group, since you will sit at a computer for the first part):

There are two ways to get started. The faster one is to use the applet directly; it is available from the course's homepage:
http://www.imada.sdu.dk/~joan/intro/index.html.
If for some reason this doesn't work, try the slower method: Download the simulator in Java from the homepage for the textbook, which can be found through the homepage for the course. You will need to login to do this, and the first time, you will need a code from your textbook. Create a directory for this course and save the file as `Simulator.java`. Compile the program using `javac Simulator.java`. Start running it using `java Simulator`.
Clicking on `Help` will tell you how to set the contents of some memory cells.

1. Do problem 1 on page 116 of the textbook. To input the data and program, type `[00] 14 02 34 17 C0 00` in the white field at the top of the window. Click on `Load Data`. Use Appendix C, starting on page 619 of the textbook, to figure out what should happen. Then, `Single Step` through the execution to see that it does happen.

2. Do problem 2 on page 116. To load the value `B0` into the program counter, type `[PC] B0` in the `Data Input Window` and click on `Load Data`. Why does register 3 get the values it does when you step through the program?

3. Do problem 3 on page 117. Note that the operation `B` is usually referred to as a *conditional* branch, and there is usually also an *unconditional* branch instruction, which always causes the program counter to get the specified value (without checking the values of any registers). How is the conditional branch instruction used here to get the effect of an unconditional branch?

4. Do problem 4 on page 117. This is strange in that it is an example of how a program can modify itself when there is no distinction between program and data. Discuss the security implications of this.

Discuss the following problems from the textbook in groups of three or four:

Page 55: Problem 4.

Page 62: Problems 1, 2 (see the appendix on page 615).

Pages 74–75: Problems 1c, 2c, 3b, 4c, 6.

Page 80: Problems 1b, 1c, 2b, 2d (for these problems use the floating-point format discussed in class, which is the same as in the textbook except that it uses an implicit bit in the mantissa).

Page 94: Problem 39 (again use the format discussed in class).

Pages 96–97: Problems 1, 5, 6.

Assignment due 12:15, September 15

Late assignments will not be accepted. Working together is not allowed. (You may write this either in English or Danish, but write clearly if you do it by hand.) Next to your name, write the section you are in. Show how you obtained your answers, and explain what your program does.

You may either do problems 1, 2, and 3, or problems 2 and 4 (a little more challenging).

1. Convert 11010111 from two's complement to its equivalent base ten form.

2. Decode 11101110 from the floating-point representation described in class (and on the first weekly note).

3. Write a program in the machine language from Appendix C which will read values stored in two memory cells `A1` and `B1`, and then create a new value which is the same the value in `A1`, except for the middle four bits of the value which should to be identical to the last four bits of `B1`. For example if 01010101 is in `A1` and 11001100 is in `B1`, then the result should be 01110001. Write the result in memory cell `A2`.

4. Read an integer n in location C0 in memory. You may assume that n is at most 8. Copy n memory locations, starting at D0 to the n memory locations starting at F0, but set the low order two bits of each value to zero. (Write this in the machine language from Appendix C.)