Institut for Matematik og Datalogi
Syddansk Universitet

September 14, 2004
JFB

Introduction to Computer Science
E04 – Lecture 3

Lecture, September 13

We covered up through section 1.7 of chapter 1 in the textbook. The textbook's interpretation of the mantissa in floating-point representations is not the same as the IEEE-standard and hence somewhat outdated: The book says that the mantissa 1010 means 0.1010 and that the first bit is always 1 in normalized numbers. IEEE-standard says that 1010 means 1.1010, meaning that the fixed normalization bit is a "hidden bit" or "implicit bit" before the radix point. This way the first bit in the mantissa may be 0. See the notes handed out about the IEEE standard.

Lecture, September 20

We will finish chapter 1 and cover chapter 2 and part of chapter 3.

Lecture, September 27

Note that we meet in U9 for the remainder of the semester. We will cover through section 3.4 in chapter 3 and begin on chapter 5.

Discussion section: week 39 – meet in the terminal room

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

Download the simulator in Java from the homepage for the textbook, which can be found through the homepage for the course:
http://www.imada.sdu.dk/Courses/DM35/. You will need to login to do this, and the first time, you will need a code in your textbook. Save the file

in the directory you created for this course 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 88 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 505 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 88. 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 88. 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 89. 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.

5. Design a program in this machine code which will have as input data a positive integer n, stored in memory cell `A0`, followed by n positive values in the next n memory locations, and will add these n numbers together. Try it with small data values. (Difficult) Would additional instructions in the machine language make this easier?

6. If you have time, try using the e-mail program `pine`. Read the e-mail from Frederik (or Holmer), and send e-mail to the other members of your group.

Assignment due 8:15, September 28

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.)

1. Write a program in the machine language from Appendix C which will read values stored in two memory cells `A0` and `A1`, and then create a new value which is the same the value in `A0`, except for the middle four bits of the value which should to be identical to the middle four bits of `A1`. For example if 01010101 is in `A0` and 10101010 is in `A1`, then the result should be 01101001. Write the result in memory cell `A2`.

2. What is the ASCII code for: "Five is 5." Note that the ASCII symbols are in Appendix A.