Institut for Matematik og Datalogi
Syddansk Universitet

September 6, 2005
JFB

Introduction to Computer Science
E05 – Lecture 2

Announcement

The discussion section on September 12 will be in the terminal room. Remember to bring your login information (IMADA computers).

Lecture, September 5

We began with an introduction to the course, covering chapter 0 in the textbook, but skipping section 0.2. We also covered "Boolean operations" and "gates and flip-flops" from section 1.1 of chapter 1.

Lecture, September 8

We will cover more of chapter 1 in the textbook, probably up through section 1.7. 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of the mantissa in floating-point representations is not the same as the IEEE-standard and hence somewhat outdated: The book says that the mantissa 1010 means 0.1010 and that the first bit is always 1 in normalized numbers.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 in the first lecture about the IEEE standard.

Lecture, September 12

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

Discussion section: week 37 – meet in the terminal room, with your login information

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/DM501/. 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. (It is probably easier to use the applet directly, since it is also available through the course's homepage. The above instructions are only there in case this doesn't work.)

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.

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

Page 34: Problem 4.

Page 40: Problems 1, 2 (see the appendix on page 495).

Page 51: Problems 1c, 2c, 3b, 4c, 6.

Page 56: 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 68: Problem 43 (again use the format discussed in class).

Pages 70–71: Problems 1, 5, 7.

Assignment due 8:15, September 19

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.) Show how you obtained your answers.

1. Convert 10101011 from two's complement to its equivalent base ten form.
2. Decode 11101001 from the floating-point representation described above.
3. 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 two bits of the value which should to be identical to the first two bits of `A1`. For example if 01010101 is in `A0` and 11001100 is in `A1`, then the result should be 01011101. Write the result in memory cell `A2`.