Institut for Matematik og Datalogi
Syddansk Universitet

September 3, 2003
JFB

Introduction to Computer Science
E03 – Lecture 2

Lecture, September 1

We began with an introduction to the course, covering chapter 0 in the textbook, but skipping section 0.2, which is more relevant to your History of Computing course. We also covered section 1.1 of chapter 1.

Lecture, September 8

We will cover sections 1.2, 1.3, and 1.4 of chapter 1 in the textbook. Then we will cover sections 2.1, 2.2, and 2.3 of chapter 2, plus Appendix C.

Lecture, September 15

Edmund Christiansen will lecture on sections 1.5, 1.6, and 1.7.

Discussion section: week 37

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/. 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 86 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 86. 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 86. 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. 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 will small data values.

5. Do problem 4 on page 86. 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.

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

Assignment due 8:15, September 16

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 a value stored in memory cell `A0`, change the middle four bits to ones without changing the other four bits, and write the result in memory cell `A1`.

2. Do question 1 on page 42 in the textbook. Note that the ASCII symbols are in Appendix A.