CSA-115 – Fall 2004 Handout: Exam2-Info
Introduction to Computer Science
Michael H. Goldwasser
Saint Louis University Thursday, 28 October 2004

Midterm 2 Information

- The second midterm will be on Thursday, 4 Nov 2004, from 12:45–2:00 p.m.
- Included with this packet is a copy of the relevant "help" files associated with the Super Simple CPU instructions. You will be allowed to bring these prepared notes into the exam with you.
- In addition to the notes provided by us, you may prepare in advance *the back of the SSCPU page* with whatever notes you wish to place on it, and you may use those during the exam. When the exam is over, submit that sheet with the rest of your exam.
- The midterm will likely consist of 8 questions and you will have 75 minutes to complete the exam.
- The exam will primarily cover lectures from Thursday, 23 Sep 2004 through Tuesday, 26 Oct 2004, as well as the associated readings and homeworks for those topics. (please see the class schedule for exact coverage)
- As a sample, we will provide you with the second midterm which was used in a past offering of this course, as well as with the solutions to that exam. Keep in mind that the exact coverage of topics varies from semester to semester, so this sample is meant mostly as an example of the *style* of the questions, as opposed to the precise content of the questions.

MORE ABOUT THE SUPER SIMPLE CPU INSTRUCTIONS

```
==================================================================
1111 STP  -- this stops the computer, no more fetch/decode/execute
             cycles until you reset.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
0001 ADD  -- fetch a number from memory and add it to the
             contents of the accumulator, replacing the value
             value in the accumulator.

  E.g.  0001 000000001111  -- get the value at memory location 15
                and add that to accumulator.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
0010 SUB  -- just like ADD, only subtract.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
0011 LOD  -- fetch a number from memory and store it into
             the accumulator, replacing its old value.

  E.g.  0011 000000001111  -- get the value at memory location 15
                and store that value into the accumulator.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
0100 LDI  -- load immediate; the value to be put into the
             accumulator is the rightmost 12 bits of the
             instruction; do not go to memory like LOD

  E.g.  0100 000000001111  -- store the value 15 into the
                accumulator.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
0101 STO  -- store the accumulator’s value into memory at the
             indicated location.

  E.g.  0101 000000001111  -- store the accumulator’s value
                into memory location 15.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
0110 INP  -- ask the user for one number and store that into
             the accumulator.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
0111 OUT  -- copy the value in the accumulator to the output
             area.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1000 JMP  -- jump to the instruction at the indicated memory
             address.

  E.g.  1000 000000001111  -- put the value 15 into the PC
                which will cause the next instruction to
                be taken from location 15 of memory.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1001 JNG  -- jump to the instruction at the indicated memory
             location if the accumulator’s value is negative;
             otherwise just add 1 to the PC.

  E.g.  1001 000000001111  -- put the value 15 into the PC
                if accumulator < 0, otherwise go to the
                next instruction.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1010 JZR  -- jump to the instruction at the indicated memory
             location if the accumulator’s value is zero;
             otherwise just add 1 to the PC.

  E.g.  1010 000000001111  -- put the value 15 into the PC
                if accumulator = 0, otherwise go to the
                next instruction.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```