CPSC 359 – Fall 2014
Assignment 4: Microarchitecture and Microprogramming

based on J. Kawash - 2014

due: 2359h 5-Dec-2014

Background

Microarchitecture refers to the organization of a computer processor to implement its instruction set. Microprogramming is a means to implement a state machine to control the microarchitecture. This assignment introduces you to these concepts by experimenting with an architecture that implements an integer Java virtual machine (IJVM), a simplified version of the Java virtual machine (JVM) than interprets Java Assembly (JAS) language.

Objectives

Your goal is to become familiar with an IJVM simulator by augmenting its instruction set and writing/modifying some simple JAS code to test the augmented instruction set.

1 Add a new ISA instruction (2 points)

In this part, we will walk you through the steps to modify the microprogram. We will add a new JAS instruction INEG that negates (in 2's complement) the value at the top of the stack. INEG is an actual instruction in the JVM, but is not included in our IJVM. (For a listing of JVM opcodes see http://docs.oracle.com/javase/specs/jvms/se7/html/jvms-6.html).

Locate the file ijvm.conf, within the Mic-1 MMV files. Add to it the following line:

```
0x74 INEG \\ write an appropriate comment
```

We are using the opcode *0x74* since it is the actual opcode of INEG in JVM and it is not used by any other existing instruction in IJVM.

Locate the file *mic1ijvm.mal*, which contains the Mic-1 microprogram. At the beginning of the file, there are few labels defined, one for each starting address of the mircoinstrucions that interprets a given JAS instruction. Add to these labels the following line:

```
.label ineg1 0x74
```

(Make sure the value (0x74) is the same as the one you defined in ijvm.conf.)

At the end of the file add the following microinstructions:

```
ineg1 H = TOS // read top of stack to H
ineg2 MAR = SP
ineg3 TOS = MDR = -H; wr; goto Main1    // negate TOS and write
                                        // back to stack
```

Add appropriate comments to these microinstructions.

Run MMV from a folder that contains *ijvm.conf*. From MMV, open *mic1ijvm.mal* assemble it, and load it. If it contains errors, correct them. (The code provided here has been tested and is error-free).

You have modified the microporgam so that it supports the new ISA (or JAS) instruction INEG. You will need it for the next part.

2 Use the new instruction (4 points)

Write a JAS program that contains the method *iabs(int x)*, which returns the absolute value of its argument, x.

Since IJVM does not have a native instruction to calculate the absolute value of an integer, you have to code it using the existing IJVM instructions, in addition to the new INEG. The algorithm is straightforward: if the value at the top of the stack is positive, then return it as is; otherwise negate it, using INEG, and return it.

3 Add some IO for testing (4 points)

In the file *add.jas* contained in the examples folder of Mic-1 MMV, there are I/O methods to read and print numbers. You may use these methods for testing; however, **they do not work with negative numbers**. Write your own *getnum()* and *print()* methods that work with negative numbers.

Team Work

You may work individually, or with a partner, but no larger groups.

Programs that do not compile cannot receive more than 3 points. Programs that compile, but do not implement any of the functionality described above can receive a maximum of 5 points.

Submit a .tar.gz file of your entire project directory (including source code, make file, build objects, kernel.img, etc) to your TA via the appropriate dropbox on Desire2Learn. If you are working with group members in different tutorial sections, choose just one TA to submit to.

Late work

After the deadline and up to 24hrs late: -2pts. After 24hrs and up to 48hrs late: -5pts. Over 48hrs late: -10pts, i.e., no assignment will be accepted beyond 48hrs after the deadline.

Plagiarism

Work must me the sole work of the individual(s) submitting the work.

You may find that the task for this assignment is similar to work found elsewhere. Nevertheless, you should go through the process yourself, and submit your own work. If you do borrow from other sources, cite the source and clearly indicate what you have borrowed, keeping in mind the design must be substantially your own. If you cite your sources, worst case you may receive a reduced grade for borrowing too much. If you borrow, but do not cite, that is plagiarism and academic misconduct, and carries severe penalties as determined by the Faculty of Science.

As a guideline, consider the 20-minute rule. Talk with your colleagues and consult other sources (cite them please). Wait at least 20 minutes, then do your work to be sure that it is your own. Less then 20 minutes usually means that you are merely copying work from the original source.