CS 429H, Spring 2011
Extending a sequential Y86 processor
Assigned: Fri Mar 4, Due: Monday Mar 21, 11:59PM

Christian Miller (`ckm@cs.utexas.edu`) is the lead person for this assignment.

1 Introduction

In this lab, you will extend a Y86 simulator with two new instructions: `iaddl` and `leave`.

2 Logistics

You will work on this lab alone.

Any clarifications and revisions to the assignment will be posted on the course Web page.

3 Handout Instructions

The code handout can be downloaded from the class webpage at:

`http://www.cs.utexas.edu/~fussell/courses/cs429h/labs/labs.shtml`

1. Start by copying the file `seq-handout.tar` to a (protected) directory in which you plan to do your work.
2. Then give the command: `tar xvf seqlab-handout.tar`.
3. Next, give the command `tar xvf sim.tar`. This will create the directory `sim`, which contains your personal copy of the Y86 tools. You will be doing all of your work inside this directory.
4. Finally, change to the `sim` directory and build the Y86 tools:

```
unix>  cd sim
unix>  make clean; make
```

4 Assignment

You will be working in directory `sim/seq` in this part.

Your task is to extend the SEQ processor to support two new instructions: `iaddl` (described in homework problems 4.32 and 4.34) and `leave` (described in homework problems 4.33 and 4.35). To add these instructions, you will modify the file `seq-full.hcl`, which implements the version of SEQ described in the CS:APP textbook. In addition, it contains declarations of some constants that you will need for your solution.

Your HCL file must begin with a header comment containing the following information:

- Your name and UTCS ID.
- A description of the computations required for the `iaddl` instruction. Use the descriptions of `irmovl` and `OPl` in Figure 4.16 in the CS:APP text as a guide.
- A description of the computations required for the `leave` instruction. Use the description of `popl` in Figure 4.18 in the CS:APP text as a guide.

Building and Testing Your Solution

Once you have finished modifying the `seq-full.hcl` file, then you will need to build a new instance of the SEQ simulator (`ssim`) based on this HCL file, and then test it:

- *Building a new simulator.* You can use `make` to build a new SEQ simulator:

  ```
  unix>  make VERSION=full
  ```

  This builds a version of `ssim` that uses the control login you specified in `seq-full.hcl`. To save typing, you can assign `VERSION=full` in the Makefile.

- *Testing your solution on a simple Y86 program.* For your initial testing, we recommend running a simple program such as `asum.yo` in TTY mode, comparing the results against the ISA simulation:

  ```
  unix>  ./ssim -t asum.yo
  ```

  If the ISA test fails, then you should debug your implementation by single stepping the simulator in GUI mode:

  ```
  unix>  ./ssim -g asum.yo
  ```

- *Testing your solution using the benchmark programs.* Once your simulator is able to correctly execute small programs, then you can automatically test it on the Y86 benchmark programs in `../y86-code`:

  ```
  unix>  (cd ../y86-code; make testssim)
  ```

This will run `ssim` on the benchmark programs and check for correctness by comparing the resulting processor state with the state from a high-level ISA simulation. See file `../y86-code/README` file for more details.

- *Performing regression tests.* Once you can execute the benchmark programs correctly, then you should run the extensive set of regression tests in `../ptest`. To test everything except `iaddl` and `leave`:

  ```
  unix> (cd ../ptest; make SIM=../seq/ssim)
  ```

  To test your implementation of `iaddl`:

  ```
  unix> (cd ../ptest; make SIM=../seq/ssim TFLAGS=-i)
  ```

  To test your implementation of `leave`:

  ```
  unix> (cd ../ptest; make SIM=../seq/ssim TFLAGS=-l)
  ```

  To test both `iaddl` and `leave`:

  ```
  unix> (cd ../ptest; make SIM=../seq/ssim TFLAGS=-il)
  ```

For more information on the SEQ simulator refer to the handout *CS:APP Guide to Y86 Processor Simulators* (`simguide.pdf`).

5 Evaluation

This lab is worth 60 points:

- 10 points for your description of the computations required for the `iaddl` instruction.
- 10 points for your description of the computations required for the `leave` instruction.
- 10 points for passing the benchmark regression tests in `y86-code`, to verify that your simulator still correctly executes the benchmark suite.
- 15 points for passing the regression tests in `ptest` for `iaddl`.
- 15 points for passing the regression tests in `ptest` for `leave`.

6 Handin Instructions

To submit your code, use the following command:

```
turnin --submit ckm seqlab seq-full.hcl
```

Make to include your name and UTCS ID in a comment at the top of your file!