CS 349, Summer 2002
Architecture Lab (Part B): Extending a Sequential Y86 Processor
Assigned: Tue May 28, Due: Monday June 3, 11:59PM

Dave O'Hallaron (droh@cs.cmu.edu) is the lead person for this assignment.

1 Introduction

In Part B of the Architecture Lab, you will learn about the implementation of a sequential Y86 processor. In particular, you will extend the SEQ processor with two new instructions.

2 Logistics

- You will work on this lab alone. Any clarifications and revisions to the assignment will be posted on the course Web page.
- In the following, `CLASSDIR` refers to

  ```
  /afs/cs/academic/class/15349-s02
  ```

3 Handout Instructions

All files you need are in the directory

```
CLASSDIR/archlabb
```

1. Start by copying the file `archlabb-handout.tar` from that directory to a (protected) directory in which you plan to do your work.
2. Then give the command: `tar xvf archlabb.tar`. This will cause the following files to be unpacked into the directory: `README`, `Makefile`, `y86.tar`, `archlabb.ps`, and `archlabb.pdf`.
3. Next, give the command `tar xvf y86.tar`. This will create the directory `./sim`, which contains your personal copy of the Y86 simulators.

4. Finally, change to the `.sim` direcctory and build the Y86 simulators and utility routines:

```
unix>  cd ./sim
unix>  make clean
unix>  make
unix>  make gui
```

4 Your Task

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

```
./sim/seq/seq-full.hcl
```

which implements the version of SEQ described in your 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 Andrew ID.
- A description of the computations required for the `iaddl` instruction. Use the descriptions of `irmovl` and `OPl` in Figure 4.16 as a guide.
- A description of the computations required for the `leave` instruction. Use the description of `popl` in Figure 18 as a guide.

5 Building and Testing Your Solution

Once you have finished modifying the `seq-full.hcl` file, then you will need to build new instances of the simulator and test it:

- *Building a new simulator.* You can use `make` to build new instances of the simulator:

  ```
  unix>  cd ./sim/seq
  unix>  make VERSION=full
  ```

  This builds new instances of `seq_tty` (the TTY form of the simulator) and `seq_tk` (the GUI form of the simulator).

- *Testing your solution using tiny Y86 programs.* For your initial testing, we recommend running the GUI form of the simulator (`seq_tk`) on some of your own tiny Y86 programs.
- *Testing your solution using some benchmark programs.* Once your simulator is able to correctly execute small programs, then you can automatically test it on the Y86 benchmark programs in `sim/y86-code`:

```
unix>  cd ./sim/y86-code
unix>  make seq
```

This will run `seq_tty` on the benchmark programs and compare results with `YIS` instruction set simulator. See the `README` file for more details.

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

  ```
  unix>  cd ./sim/ptest
  unix>  make SIM=../seq/seq_tty
  ```

  To test your implementation of `iaddl`:

  ```
  unix>  make SIM=../seq/seq_tty TFLAGS=-i
  ```

  To test your implementation of `leave`:

  ```
  unix>  make SIM=../seq/seq_tty TFLAGS=-l
  ```

  To test both `iaddl` and `leave`:

  ```
  unix>  make SIM=../seq/seq_tty TFLAGS=-il
  ```

For more information on the SEQ simulator, refer to the handout "CS:APP Guide to Y86 Processor Simulators."

6 Evaluation

This part of the lab is worth 60 points:

- 10 points for your description of the computations required for `iaddl`.
- 10 points for your description of the computations required by `leave`.
- 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 synthetic regression tests for `iaddl` in `ptest`.
- 15 points for passing the synthetic regression tests for `leave` in `ptest`.

7 Handin Instructions

- You will be handing in the file `seq-full.hcl`.
- Make sure you have included your name and Andrew ID in a comment at the top of each of your handin files.
- To handin your solution, go to your protected directory that contains your `sim` directory (i.e., the directory in which you unpacked the `archlabb-handout.tar` file), and type:

  ```
  make handin TEAM=teamname
  ```

  where `teamname` is your Andrew ID.
- After the handin, if you discover a mistake and want to submit a revised copy, type

  ```
  make handin TEAM=teamname VERSION=2
  ```

  Keep incrementing the version number with each submission.
- You can verify your handin by looking in

  ```
  CLASSDIR/archlabb/handin
  ```

  You have list and insert permissions in this directory, but no read or write permissions.