CSCE 614 (Fall 2017) Eun Jung Kim

Computer Architecture

Homework # 2 (Branch Predictor)

Due: 10/03 2:20 pm (Report must be submitted in the class time)

1 Objective

This project is to help you familiar with zsim, a Pin-based x86-64 simulator that implements out-of-order issue superscalar processor with detailed memory hierarchy for large-scale multicore architectures.

2 System Requirement

Linux operating system is required in order to use the zsim. **Do not use Cygwin, Mac OS X is not supported**. If you dont have any linux machine, please use linux.cse.tamu.edu with your CSE account. If you dont have CSE account, contact HelpDesk located in the third floor.

3 Procedure

3.1 Setup Your TAMU Git Account

Please login to github.tamu.edu using your NetID and password. We will use Github for the programming homework assignment as well as project. Please try to learn and get familiar with git commands.

3.2 Fork and Setup ZSim

We have support neccessary librariers and setup for your ease of use. It is put at https://github.tamu.edu/jyhuang/casim.

1. Fork zsim from **casim** repository. https://github.tamu.edu/jyhuang/casim.
2. Add TA **jyhuang** as a collaborator in your forked repository (check setting).
3. Read the README.md file under casim.
4. Setup the simulator according to the instructions:

```
$ git clone https://github.tamu.edu/netid/casim
$ cd casim
$ source setup_env
$ cd zsim
$ scons -j4
```

5. After you get the simulator, have a look at the system configuration files under `zsim/tests`, they show how system parameters are configured and the command to simulate.

3.3 Get the Benchmark Inputs

1. Download input files for PARSEC benchmarks from the following link:
   http://students.cse.tamu.edu/jyhuang/csce614/parsec_smallinputs.tar.gz
2. Move the downloaded file to `casim/benchmarks/parsec-2.1/`
   ```
   $ mv parsec_simsmallinputs.tar.gz casim/benchmarks/parsec-2.1/
   $ tar xzvf parsec_simsmallinputs.tar.gz
   ```

3.4 Get run script and config files

1. Download files from the following links:
   http://students.cse.tamu.edu/jyhuang/csce614/configs.tar.gz
   http://students.cse.tamu.edu/jyhuang/csce614/hw2runscript
2. Move them to `casim/zsim` and untar the files using tar command
3. The runscript is used to run each benchmark, check how to use it
   ```
   $ ./hw2runscript
   ```

3.5 Run the following benchmarks using ZSim

blackschoels, bodytrack, canneal, dedup, fluidanimate, freqmine, streamcluster, swaptions, x264

1. Use `hw2runscript` to run the benchmarks
   Note: The config files and runscript must be in the `casim/zsim`.
   ```
   $ ./hw2runscript <benchmark> <automaton>
   ```
   Example: `$ ./hw2runscript blackscholes A2`
2. Check the results in `outputs` directory (`zsim.out`)

If you are running zsim in linux.cse.tamu.edu, be sure you are not monopolizing computational resources on the machine. **Do not run more than 1 instance at a time in linux.cse.tamu.edu.** It is violation of section 3.3 of the Appropriate Use of Computer Science Computing Resources Policy, located here:
https://engineering.tamu.edu/cse/cse-internal/computing-resources

Don't run more than one instance of any benchmark simultaneous in the same machine. It may cause errors. **Run one instance at a time per benchmark.**

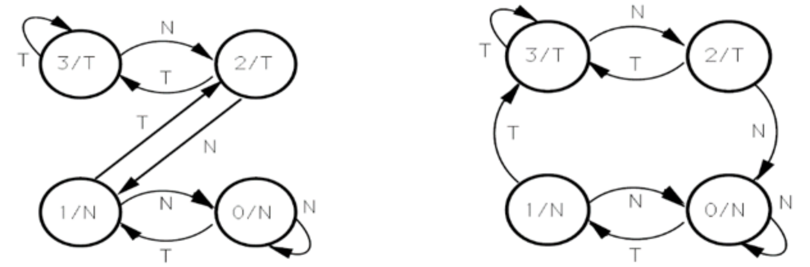
4 Assignment

4.1 Reading

1. Tse-Yu Yeh and Yale N. Patt, "Alternative Implementations of Two-level Adaptive Branch Prediction," ISCA 1992
2. Tse-Yu Yeh and Yale N. Patt, "A Comparison of Dynamic Branch Predictors that use Two Levels of Branch History," ISCA 1993

4.2 Part A: Automaton Implementation

Compare performance of two different state diagrams in two-level branch prediction. According to the papers, these two state diagrams have the nearly same performance. Show how close these state diagrams. A2 is used in zsim already, you need to add codes to implement the A3 state diagram.



(a) A2 (Original zsim Implementation) (b) A3 (Same as shown in the textbook)

Figure 1: Two different State Diagrams

Use `hw2runscript` to run both A2 and A3

Note: You will need to modify `ooo_core.h` to implement the state diagram

4.3 Part B: Configuration Derivation

Branch predictors can have different configurations as follows:

1. GAg: 1 global history register and 1 global prediction table
2. GAp: 1 global history register and 8 per-address prediction tables
3. PAg: 8 per-address history registers and 1 global prediction table
4. PAp: 8 per-address history registers and 8 per-address prediction tables

We will use the following format for the configurations.

```
-bpred:2lev <l1size> <l2size> <hist_size>

Configurations:  N, M, W

        N  # entries in first level     (# of shift register(s))
        M  # entries in 2nd level       (# of counters, or other FSM)
        W  width of shift register(s)   (# of bits in each shift register)


Sample predictors:
```

```
    GAg:  1, M, W  where $M = 2^W$
    GAp:  1, M, W  where $M = C * 2^W$, C is # of per-address prediction tables
    PAg:  N, M, W  where $M = 2^W$
    PAp:  N, M, W  where $M = N * 2^W$
```

Requirements: Show the configurations for the above 4 branch predictors using **N, M, W** and write in the report. Assume for each branch predictor, the number of predictors (saturation counters) is 512. (No simulation is required for Part B)

5 Turning Instruction

1. Make all your files including **modified source codes, simulation results and the report** into one zipped file. **We accept zip files only.** If you send a different file format, you may receive **0 point** for the assignment. Please **DON'T turn in the whole simulator**.

Your report must contain simulation results (You should include output files in the zipped file, **draw figures of results** in your report.) and analysis of them. Any result you consider important can be used. Only **PDF** is acceptable for the report.

2. Submit your zipped file through **CSNET**. Your file name should be **hw2_#UIN.zip**.

3. Also commit your change of src code to your Forked repository in github.tamu.edu, TA will check your code history.

4. IMPORTANT: Be sure to turn in the hard copy of your report including simulation results in the class time.

5. Penalty of late submission: 5% deduction per day.

Any questions, please contact Jiayi Huang (jyhuang@cse.tamu.edu)