CSE261 Project 3: Building a Cache Simulator

Deadline: 12/6 (TUE) 23:59

Overview

The main goal of this project is building a **4-way set associative cache.**

Detail specification of the cache is:

- **32-byte block(cacheline) size**
- **8 sets**
- **Write back**
- **LRU replacement scheme**
- Total (8 set * 4 block/set * 32byte/block) = 1KB cache size

```
# An empty cache layout

|index| v |  tag   |              way0               | v |  tag   |              way1               | v |  tag   |              way2               | v |  tag   |              way3               |
+-----+---+--------+---------------------------------+---+--------+---------------------------------+---+--------+---------------------------------+---+--------+---------------------------------+
| 000 | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... |
| 001 | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... |
| 010 | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... |
| 011 | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... |
| 100 | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... |
| 101 | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... |
| 110 | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... |
| 111 | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... | 0 | 000000 | ...cache block data (32byte)... |
```

- **Address translation layout**
  - 32-byte cacheline: log2(32) = 5 bit for offset
  - 8-set: log2(8) = 3 bit for index
  - tag: 32-(5+3) = 24 bit

```
 ```
 31   8 7   5 4   0
  +------------------//---------------+-----------+---------------------+
  | tag(24bit) | index(3bit) | offset(5bit) |
  +-----------------//----------------+-----------+---------------------+
 ```
```

This project is an extension of Project2. If you did not completed Project2, you can't proceed this project. Before you start this project, fill `run.py` and `parse.py` files that you implemented at Project2.

Your job is implementing `cache_read()` and `cache_write()` function in `cache.py`. Below diagram will be helpful when you are doing this project.

- Data transfer call flow

```
                                           [word-size transfer]             [block-size transfer]
                                                 (4byte)                          (32byte)
+-------------+                   +-----------+                     +-----------+                     +----------+
|             |  load/store instr | processor |     cache_read()    |           |     block_read()    |          |
|   program   | ----------------> |           | <------------------ |   cache   | <------------------ |  memory  |
|             |                   | (register)| ------------------> |           | ------------------> |          |
+-------------+                   +-----------+     cache_write()   +-----------+     blcok_write()   +----------+
```

How to Execute

```
python3 main.py [-d debug_interval] [-n num_instr] inputBinary
```

- Parameters
  - `-d` : Print the cache contents and memory for every `debug_interval` cycles.
  - `-n` : The number of instructions to be simulated.
- We provide C source code for testcases for letting you know how to these programs operate.
- In assembly(.s) file, There are some instructions that was not included in the previous projects. But they are properly translated to object(.o) file so that you can also run these testcases in project2.

Grading Policy

We will evaluate your cache and memory state at runtime using our reference code. So, we do not provide grading program. But, we guarantee that If you pass all tests in `./test.sh` , you will get full score.

Submission

The grading will be on the code in Master branch at the deadline. Any other branches are not considered as submission of the assignments. Please make sure that you can see the same result on the submission on the master branch as on your local machine.

Be careful!

- If there are any updates to the project, including additional tools/inputs/outputs, or changes, we will post a notice on the BB and Piazza. Please check the notice for any updates. One concern is that we are using Python as an assignment language, there is plenty of packages you can leverage. Therefore, **we restrict you not to import external packages**.
- **Be aware of plagiarism!** Although it is encouraged to discuss with others and refer to extra materials, copying other students or opened code is strictly banned. The TAs will compare your source code with other students' code. If you are caught, you will receive a penalty for plagiarism. Last semester, we found a couple of plagiarism cases through an automated tool. Please do not try to cheat TAs. If you have any requests or questions regarding administrative issues (such as late submission due to an unfortunate accident, git push or repository not working) please ask TAs through Piazza. Be careful not to post your own code when posting questions on Piazza. If you post your own code, it will be considered cheating. And please keep your questions private if your questions can be a hint for other students to do their assignment. If possible, it is better to ask questions through Piazza (assignment - #3), but if it is urgent, you can send an email to TA (aero5010@unist.ac.kr).