COL 718 Architecture of High-Performance Computers

Assignment 2

Add Support for Instructions in Tejas

September 8, 2022

In this assignment, you have to add support for ***system call (syscall)*** and ***clflush*** instructions in the Tejas architectural simulator for the Out of Order pipeline. You have to provide the simulation statistics. This assignment has to be done in a group of two. The evaluation will be based on the demos.

Add support for the following instructions:

- ### System calls:

  A system call is a way for a user program to interface with the operating system. System calls are predefined functions that the operating system may directly invoke if a high-level language is used. You have to add support for a *syscall* in Tejas, which flushes the *iTLB* and *dTLB* whenever a syscall is encountered in the pipeline.

  The functionality of system call that is needed to be added in Tejas:

  Whenever a syscall is encountered by Tejas, it should be able to identify the instruction as a syscall and should flush the TLBs. For this assignment, only make changes in the /src/simulator folder. **No changes** need to be made in /src/emulator.

  Hint: See how add or mov instructions are implemented in Tejas. Implement on a similar basis. A function needs to be added in TLB.java to flush the TLBs.

- **Clfush**: Flush cache-line

  Clflush instruction takes the address of a cache line as an operand. Clflush invalidates the cache line from every level of the cache hierarchy in the cache coherence domain. If that cache line contains modified data at any level of the cache hierarchy, that data is written back to memory.

  The functionality of clflush that is needed to be added in Tejas:

  Whenever a clflush is encountered by Tejas, it should be able to identify the instruction as clflush and subsequently flush the respective cache-line. For this assignment, only make changes in the /src/simulator folder. **No changes** need to be made in /src/emulator.

  Hint: A function needs to be added in Cache.java to flush the respective cache-line.

Framework:

- Download the assignment and extract it.
- The folder traces contain the trace files of syscall and clflush instructions, which will be used as an input to the config file in /src/simulator/config/config.xml file in Tejas. Copy the trace folder in Tejas home folder.
- Follow the instructions to set up Tejas as provided in assignment 1.
- Additional changes for running the Tejas through trace files:
    - In the config file (/src/simulator/config/config.xml) change the ***emulatorType*** to **none** and ***communicationType*** to **file.** In ***BenchmarkPath*** give the path of the absolute path of the trace file.
    - For running the simulation in test.sh file use command;
        - Java –jar <jar file> <config file> <output file>

Evaluating your Design:

Run the following commands:

- ant clean: to clean the class files and jar files from previous builds.
- ant make-jar: to create a jar file (akin to an X86 executable).
- Java –jar <jar file> <config file> <output file>

This should run without errors and your simulation results can be seen in the output file.

Compare the following simulation statistics: IPC, cache stats, and TLB stats with and without the addition of the instructions and **submit a proper report** with the implementation and screenshots of the simulation results.

Submitting your Assignment:

We will be evaluating your submission based on the demos and the report. So, make sure the submission can be built and executed successfully, otherwise you will be awarded **zero marks**. Create a .tar.gz archive of the Tejas home folder. Name it <entry_number1_entry_number_2". Submit only the.tar.gz archive to Moodle. The deadline is **October 5, 2022**.