Lab #07

Decelerating Applications by Disabling Caching

Assignment Type:	Individual
Assignment Duration:	1 week
Marks (of course total):	3%
Submission Method:	https://tcd.blackboard.com
Submission Deadline:	23:59 on Friday the 25th of March 2022

Introduction

This lab aims to highlight the importance of caching in modern computing systems by running an application with/without caches enabled to compare the performance of the application in both scenarios. You should take the code you wrote for lab #06 (Accelerating Applications using Multiple Cores) and extend it so that you can measure the performance of your code running under the following conditions and:

1. Using a single CPU core with caches enabled
2. Using a single CPU core with caches disabled
3. Using both CPU cores with caches enabled
4. Using both CPU cores with caches disabled

For each scenario, measure the execution time for the following:

1. The single-precision floating-point function
2. The double-precision floating-point function
3. The total execution time of the application

Implement the following template functions to get and set the correct bit of the register that controls the functionality of the XIP cache (it is defined as *XIP_CTRL_BASE* in the Raspberry Pi Pico SDK). You should refer to both the SDK documentation and the RP2040 SOC Datasheet for more details.

```
// Function to get the enable status of the XIP cache
bool get_xip_cache_en();

// Function to set the enable status of the XIP cache
bool set_xip_cache_en(bool cache_en);
```

After completing this lab, you should have successfully created and compiled a C application to gather the 3 different data-points for each of 4 different scenarios (so 12 data-points in total) described above.

Instructions

1. Make sure that your "pico-apps" repository is up to date before starting the lab.
2. Copy the code you wrote from "labs/lab06/lab06.c" into "labs/lab07/lab07.c" in the *"labs/lab07"* template folder.
3. Extend your code so that you can enable and disable the RP2040 XIP cache
4. Using your code, capture the results for each of the 12 datapoints outlined in the introduction section
5. Document and graph your datapoints in a new file called "lab07_rslt"
    - Calculate the performance gained by using the cache when running on 1 core and when running on 2 cores and add to your results file
    - Briefly comment on and discuss any interesting observations you make based on the various execution times you have logged
6. Make sure that the code is fully commented (Doxygen format is preferred)
7. Commit the changes push them back to your "*pico-apps*" main repository

Grading Scheme

Complete all steps in the instructions section and then upload the completed "**lab07.elf**", "**lab07.uf2**", "**lab07.c**" and the "**lab07_rslt**" graph and results files to the lab07 assignment in Blackboard before the specified deadline to complete the lab. This lab is worth a total of 3% of your year-end results for the CSU232021 module and the marking scheme is as follows.

	INCOMPLETE [0%]	COMPETENT [50%]	PROFICIENT [100%]
SUBMISSION [10%]	Student has failed to submit the lab exercise.	Student has submitted the lab exercise but some of the required deliverables are missing.	Student has submitted the lab exercise and has included all the required deliverables.
IMPLEMENTATION [65%]	Student has not made a sufficient attempt at completing the lab exercise.	Student has completed the lab exercise however the implementation does not work as expected.	Student has completed the lab exercise and the implementation works as expected.
QUALITY [25%]	The code is badly structured or has not been well commented. Some or all of the specified tasks were not completed.	The code is adequately structured and there are some comments present. Some of the specified tasks were not completed.	The code is well structured with good and clear commenting throughout. All of the specified tasks were completed.