

Programming Assignment 3 Part A: Load bypass

1 Objectives

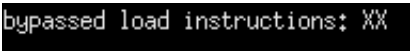
The objective is to let you step into gpgpu-sim simulator, and prepare for your final project.

2 Ground Rules

- (1) All students must work alone. Sharing code between students will be considered as cheating and receive appropriate action based on the University Policy.
- (2) A Q&A forum has been created on Moodle for posting questions, discussing and debating issues, but please do not share your code directly on Moodle.
- (3) **Every student should follow the submission requirement exactly, otherwise you will receive points deduction.**

3 Description

For this task, you need to implement the **load bypass mechanism** in gpgpusim and let all the load instructions in the address range of **0xc0000000-0xc00ffff** bypass the L1 data cache. The benchmark you should use is BFS, and you should dump out how many load instructions bypassed the L1 data cache in the following format



```
bypassed load instructions: XX
```

Figure 1: Output format

Note that BFS is a multi-kernel benchmark, you should dump out your results for each kernel

4 Hints

1. Read the code about load store unit in shader.cc
2. Try to understand where the L1 data cache access request are generated.
3. Try to understand where the response of these bypassed load instructions

are handled in the load store unit.

4. To verify your implementation, you can run the BFS benchmark with unmodified code to get the statistics about L1 data cache access, and then run the BFS benchmark with your implementation, compare the L1 data cache statistics with the unmodified code. For the correct implementation, you should be able to see that the L1 data cache access statistics are changed.

5 Submission

1. Write report to describe your implementation and report the counters about bypassed load instructions for BFS benchmark.
2. Zip the files you modified and your report as `unity_id.zip`. Please make sure you follow the submission requirements correctly.