The Cache Project Part 2: L2 Cache
Due Friday, December 8

The second part of the project is to simulate a direct-mapped, write-back L2 cache in C. As with the main memory part that you have already completed, in this simulated system data is written to and read from the L2 cache as entire 8-word cache lines, not individual words (which are 64 bits).

A detailed description of the L2 cache is found in the l2_cache.c file that you already downloaded at the start of the project. I have provided extensive comments in l2_cache.c describing what the code should do. Your job is to fill in the missing code.

Remember, you need to work on your own on this project. Be sure to check the discussion board for helpful information.

Here are the steps you should take:

- **Step 1**
  You should see what the output of testing my compiled L2 code is by typing

  ```
  ./ben_test_l2
  ```

- **Step 2**
  Open the file l2_cache.c in an editor and read every line closely. You'll see that I describe the organization of the L2 cache and that there are three procedures, l2_initialize(), l2_cache_access(), and l2_insert_line(), that you have to implement. You should also look closely at the related header files, memory_subsystem_constants.h and l2_cache.h.

  In l2_cache.c, the comments describe exactly what you have to do to implement the above three procedures to 1) initialize the L2 cache by clearing the valid bits in each cache entry, 2) support reads from and writes to the L2 cache, and 3) insert a new cache line into the L2 cache (possibly evicting another cache line).

  You should also take a look at my code in test_l2.c, so you can see how l2_initialize(), l2_cache_access(), and l2_insert_line() are called.

  The only file you need to modify for this part of the project is l2_cache.c.

- **Step 3**
  To test your code in l2_cache.c, you can use my test_l2.c file. To do so, compile them together by typing

  ```
  make test_l2
  ```

and, if it compiles correctly, run the program by typing

```
./test_l2
```

The output when you run the program should be the same as the output when using my compiled version (./ben_test_l2). Feel free to read and modify the code in test_l2.c to aid in your debugging.

- **Step 4**
  Upload your l2_cache.c file and get started on part 3!