Project #2 (Due November 10)

NOTE: This assignment can be done by a team of two students (probably the same team of project 1).

The goal of this assignment is to study the effect of caches on CPU performance by adding a cache simulator either to your pipeline simulator developed in Project 1, or to the single cycle CPU simulator given to you in *CPU.c* (if you could not successfully complete Project 1). Your CPU+cache simulator will assume that instructions always hit in the instruction cache and only simulate the memory hierarchy for data access. The simulator should accept the same command line arguments as *CPU.c* and read the following memory hierarchy configuration parameters from a configuration file "cache_config.txt":

1. the size of the L1 cache in Kilo Bytes
2. the block size for the L1 cache (which is also the block size for L2)
3. the associativity of the L1 cache
4. the size of the L2 cache in Kilo Bytes
5. the associativity of the L2 cache
6. the L2 access time
7. the memory access time

All 8 parameters are integers and each of the first six parameters should be a power of 2. The access time for the L1 cache is assumed to be zero (access is completed within the execution cycle). An L2 size of 0 means the absence of an L2. For example, the following file configures a hierarchy with only an L1 cache and no L2 cache.

A skeleton for the extension of *CPU.c* to include the data cache hierarchy is given in *CPU+cache.c*. This skeleton is very rough and may not even compile, but includes the file *cache.h* that contains a template for the data structures and functions that you should use to implement a cache. Your task is to complete the implementation of *CPU+cache.c* and *cache.h* assuming the following:

- All the blocks in the caches are initially invalid.
- The "write allocate" policy is applied on a write miss
- The "write back" policy applies
- LRU block replacement.
- The L2 is "inclusive" in the sense that a block is not evicted from L2 if it resides in L1

In addition to the total number of execution cycles, the simulator should report the total number of memory accesses and the miss rate (for each of L1 and L2).

To test the correctness of your simulator, you may want to use your own short traces that are composed of only a few instructions designed to test specific cases. Given that you cannot construct a binary trace file, you may rewrite the function *trace_get_item()* to read instructions from a text file in which each line represents an instruction (four "char" and two "int" to specify `type, sReg_a, sReg_b, dReg, PC, Addr).` You can then write your own sequence of instructions. In fact, to test the cache simulator, you only need load and store instructions (for which you only need `type` and `addr).`

After completing and testing your implementation, you should use the same traces as in project 1 and conduct the following experiments, assuming a memory access time of 100 cycles:

Experiment 1: To examine the effect of the block size, consider a system with no L2 and three different L1 cache sizes (1KB, 16KB and 128KB) with 4-way set associativity and four possible block sizes (4B, 16B, 64B, and

256B). For each of the two long traces, build two tables similar to the one shown below, one to report the L1 miss rate and the other to report the total number of cycles needed to complete the execution of the trace.

	1KB	16KB	128KB
4B			
16B			
64B			
256B			

Experiment 2: Repeat experiment 1 after adding a 1 MB L2 cache with the same associativity as the L1 cache, assuming an L2 access time of 6 cycles.

Experiment 3: To examine the effect of associativity, consider a 256KB L2 and a 4KB L1 with 32B blocks. Use a bar graph to plot the miss rate for associativity = 1, 4 and 8. This plot will have 6 bars, three for each long trace.

What to submit (email to TA):

1) One file with your source code for the *CPU+cache.c* and *cache.h* (which includes all functions and declarations).
2) One file with
    a. The result of running your simulation with *trace_view_on* = 0 for each long trace file and the parameters specified in each experiment
    b. The tables and bar graph specified in the experiments
    c. Your remarks and explanation of the results and the conclusions that you draw from the experiments.