/home/diana/Documents/Comp Org/Project/cache.h

```
1 #ifndef CACHE_H
2 #define CACHE_H
3
4 // Each block in a cache
5 struct Block {
6   int valid;    // track if block is valid
7   int dirty;    // track if block is dirty
8   unsigned long long tag;       // block's tag
9
10   // next block in LRU chain: if this is null, this is the LRU
11   struct Block *nextBlock;
12 };
13
14 // Each cache
15 struct Cache {
16   int cacheSize, blockSize, associativity;      // provided from config file
17
18   // how many rows in the cache
19   int lengthOfWay;
20
21   // size of index field & byte offset field
22   unsigned int indexFieldSize, byteOffsetSize;
23
24   // track number of hits/misses/references; need to be unsigned to handle possible sizes
25   unsigned long long hits, misses, writeRefs, readRefs, insRefs;
26
27   // track number of blocks that were invalidated during a flush
28   unsigned long long invalidates;
29
30   // track kick-outs, dirty kick-outs, flush kick-outs, and transfers
31   unsigned long long dirtyKickouts, kickouts, flushKickouts, transfers;
32
33   // track cycles for activities
34   unsigned long long instructionTime, readTime, writeTime, flushTime;
35
36   // how many cycles each hit/miss should add
37   unsigned int hitTime, missTime;
38
39   struct Block **blockArray;    // Array of pointers to blocks, implements LRU policy
40 };
41
42 #endif
43
```