In main:

ParseFile()

{

RWvar;

address;

if “RWvar == read”

ReadFunction(numReads, numHits, address) TODO check if numHits or numMisses

else if “RWvar == write”

writeFunction(address) TODO track hits and misses here?

}

Have a function that builds and returns a CacheMatrix

Have a write Address function, which calculates where to store an element, set the tag, valid bit~~, and dirty bit~~.

Figure out what a dirty bit is and why it’s needed Not needed, that’s for virtual memory You actually do need a dirty bit, it’s used in the write back policy

Have a read address function, which takes in an address, calculates what the index and tag should be, checks the tag at that index. (If not direct mapped, checks multiple tags.) If valid bit is 0 or tag(s) doesn’t match, miss. Store total number of misses, also increment every read. Can probably pass in totalReads variable or something and just increment at the top of read method.

First, figure out how to simulate a cache of one type.

Then, make a class/method which will build the cache based on the 4 relevant variables.

Then, simulate reads/writes on the cache. Recall we don’t actually care what the data is

Finally, read in the file and interpret “read” for read method and “write” for write method, and interpret hex address.

Then we need to track certain things, plot those things, etc.