1 Outline

Each element of the Cache is a "Block". A Block is simply a class with the tag, valid bit and dirty bit. The Cache contains A 2D vector of Blocks, Replacement Policy, Write Policy and associativity. For each address in the access file, the access method of Cache is called. To handle all the policies in the same class, the dimensions of the 2D cache vector are set as per the cache type. For Direct-Map, the associativity or the second dimension is simply 1. For Fully-Associative, the number of sets or the outer dimension is set to 1.

```
class cache
{
public:
    int cacheSize, blockSize, associativity;
    string replacementPolicy, write;
    int numLines;                    // Number of sets
    vector<vector<block>> dataBlock; // 2D vector to store the cache
```

Figure 1: Cache

2 Bit-Masking

I use the inbuilt log2 function to find the offset, index and tag bits. Now, I re-use my maskInRange function from the last assignment to obtain the index and tag from the number.

```
// Extract out bits from start to end
long long int maskInRange(long long int hex, int start, int end)
{
    // hex >> (start) -> erase the bits upto start
    // ((1 << (end-start)) - 1) -> we get end-start set bits(2^(end-start)-1)
    // Taking bitwise and of the two numbers we get the required bits[start, end).
    return ((hex >> (start))) & ((1 << (end - start)) - 1);
}
```

Figure 2: MaskInRange

3 Replacement-Policy

For LRU replacement, I bring the block to the end of the line on every hit. In this way, the blocks get ordered from the least recently used to the most recently used. Now, for replacement, I can simply remove the first element and add the new element to the end for both LRU and FIFO(Ordered by when they were brought to the cache). For Random, I simply generate a random number between 0 and associativity-1 and replace it.

4 Handling Writes

I did not allocate for a miss in case of write operation on Write Through policy. In the case of the Write-Back policy, I set the dirty bit to one whenever I get a hit on write-mode to indicate a change in data. Initially, both valid and dirty bits are set to zero.