

Defining the associativity:

To use n-way associativity just pass use “-a n” as a command line argument.

To use a direct mapped cache use “-a 1”.

To use a fully associative cache use “-a 0”.

Outputted data:

The program outputs the passed information, as well as information such as total number of lines, blocks and sets etc. that is derived using the command line arguments.

The total number of hits and misses, as well as the ratio of hits is outputted last.

Results:

To better simulate reuse of address I simply copied the address file twice, so that the cache will access addresses more than once.

With size = 128 and line size = 16:

Direct mapped: 35.9% hit rate.

2 way associative: 40.6% hit rate.

4 way associative: 42.2% hit rate.

Fully associative: 46.9% hit rate.

With size = 256 and line size = 16:

Direct mapped: 54.7% hit rate.

2 way associative: 62.5% hit rate.

4 way associative: 67.2% hit rate.

Fully associative: 76.6% hit rate.

Notes:

Addresses are stored in an array. The size of this array is calculated automatically using *sizeof(addresses) / sizeof(int)*, so there is no need to change the size variable when adding/removing addresses from the array.

When using associativity > 1, ensure the size is larger enough to support the required number of blocks/sets. The program will print an error if this is a problem. As example consider size = 128, line size = 16 and associativity = 16. This requires 16 sets per block, but only 8 lines are available (128/16).