CCE1011 Introduction to Computer Systems

Online Assignment from William Stallings website

Cache Memory

Use http://williamstallings.com/COA/Animation/Links.html

From the page choose Cache Simulator. It is important to move down the page and press Return to Main Menu. The screen below, Figure 1, should appear. This diagram gives us the ability to see various aspects of the cache before attempting a full simulation.

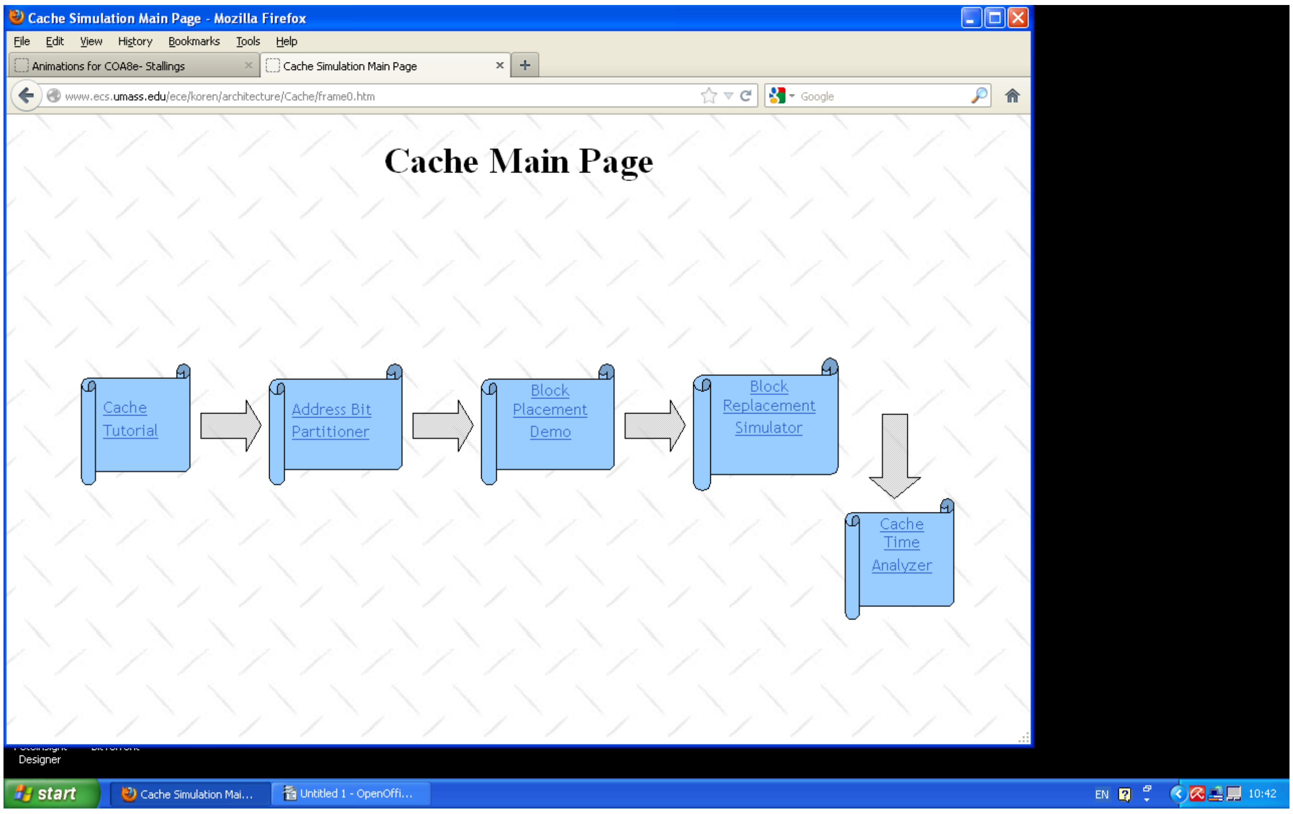

Figure 1

Press Address Bit Partitioner.

Section A - Address Bit Partitioner

This is used to divide a Main Memory Address into the TAG, SET or BLOCK, and WORD fields. Stallings is using here TAG, INDEX and OFFSET

Note on the right (i) the setup for MM size, Cache size, Block size
(ii) the cache scheme
(iii) the **show** button that gives you the address with its three fields.

Do the following:
A Set cache scheme to **direct** and
(a) MM 256KB, Cache 64KB, Block 16B
(b) MM 1MB, Cache 64KB, Block 16B

(c) MM 4MB, Cache 256KB, Block 32B
(d) MM 16MB, Cache 256KB, Block 2B

For the four cases try to obtain the answer BEFORE looking at it using **SHOW**.

B. Repeat the four cases for

(i) Set Associative with	2 Blocks per set
(ii)	4 Blocks per set
(iii)	8 Blocks per set

You can try any other combinations to make sure you understand the MM address partitioning.

Return to Main Menu

Choose **Block Placement**

Section B Block Mapping from main memory to cache

Note that in this window the parameters are set from the top part. This exercise assumes a Block Number (in binary or decimal) as input. Set up the cache system as follows:

A.

direct scheme; cache size 128B; block size 8 B; from right bottom window choose **decimal** and in the box for the MM block number write 77.

Press MAP to get the answer.
Note (i) number of cache blocks is 128/8 = 16 (Block 0 to Block 15)
(ii) mapping is to rem (77/16) = 13
(iii) the MM address in binary with the OFFSET field set to all zeroes since we are considering only a block in this exercise.
The output shows the mapping of MM blocks to the Cache blocks.

B. Repeat using the following MM block numbers - 201, 205, 256

C. Now change the block type to **binary** and give the following patterns:
100111 , 100001, 1111110000000, 000000111.
Note that in this case the OFFSET field contain the lower bits, depending on the block size.

D. Work out a binary address that maps into block 7 of the cache and check.
E. Now change to set associative, 2 blocks per set, leaving same cache size and block size and repeat for the same block numbers (77, 201, 205, 256) and same binary values
F. Repeat for set associative 8 way.

At the end Return to **Main Menu**

Press Block Replacement Simulator

Section C Block Replacement in the Cache

A.

(i)Choose Direct Mapping by choosing Cache Size and Sets to have the same number. Start by choosing 16.
(ii)Use the **LRU** replacement algorithm

(iii) In the **Enter Query Sequence** write down the numbers from 0 to 20 spaced with commas. Increase the **limit query** at the bottom of the left hand window to 30.
Note that the repeat 2 cycles should be OFF while the SHOW TAGS should be ON. Note also that we are using only one processor. So the second box is left empty.
Press the SHOW CACHE to look at the result. Look at it carefully, noting the final blocks in the cache, the misses and sequence trace.

B. Now set the repeat cycles to 2 and the limit query to 50 and Repeat. Again look carefully at the output of both the first cycle and the second cycle. In particular see how the blocks are being replaced and the overall hits and misses in second cycle.

C Now change to FIFO and see whether there is any difference.

D. Change to Cache Size 16 and Sets 8 (This means 2 blocks per set) and repeat A, B and C above, and contrast with the original results obtained.

E. Finally change to cahe Size 16 and Sets 4 (4 blocks per set) and Repeat. Again compare results.

F. If the MRU replacement algorithm is used do you think the results will be better?

Try to experiment some more and look carefully at the results.

Section D Cache Time Analysis

This last section includes writeback or write through as a new parameter, and makes a statistical analysis that results in hits and misses for a given cache and main memory system. Look at the details of the calculation of read and write hits and miss time allocations to see how the average is obtained.