Chris Wong; wongc
Hans Wun; wunh
Cwong030@ucr.edu, Hwun001@ucr.edu
Lab 7: Caches
I acknowledge that all content is of my own
Lab Report -7

I. Lab objective

In this lab, the main objective is to learn and implement several caches in VHDL. Our caches will vary in size and associativity. The Caches will not be holding any data, we are only looking weather or not a hit occurred, also, we are not interested in outputting where the hit occurred, and what data is stored at the cache line.

II. Personal Contributions

Chris 50%. Hans 50%. We both contributed to an equal amount of work on this lab. We each worked on the same parts and helped each other code and run tests.

III. Skills learned and knowledge gained

We learned how to code VHDL in C/C++ by using streams. We learned how caches worked and how to create a virtual one using Xilinx. IN addition, we learned the effects of modifying the cache's associtivity, physical size, and block size, and what impacts it has on the hit rate performance.

IV. Known bug locations

We were able to create the multiplexer.cc and decoder.cc generators, and the control block for the cache. We had problems with implementing the set and did not have enough time to complete it. With the small amount of time givent, we have created a 4KB DM my cache file, with all our components in there, but they are not wired together.

V. Feedback on lab

We had very little time (5 days instead of 14) to do this lab, and did the best we could in the allotted time.

VI. Case Study Report/Analysis

***Data on Cache A:**
Defaults: size = 16KB, Associativity = 4W, Block Size = 64 bytes,

Varying size (KB)	-> # of Hits
4	543081
8	689208
16	820243
32	899072
64	950073
128	974636
256	987616
512	994776
1024	998184
2048	999365
4096	999596

Varying Assoc.	-> # of Hits
DM (1)	733676
2W	788630
4W	820243
8W	826467
16W	833687
FA (-1)	843161

Varying Block size(Bytes)	-> # of Hits
4	970426
16	932612
32	885056
64	820243
128	720630
256	602308
512	460223

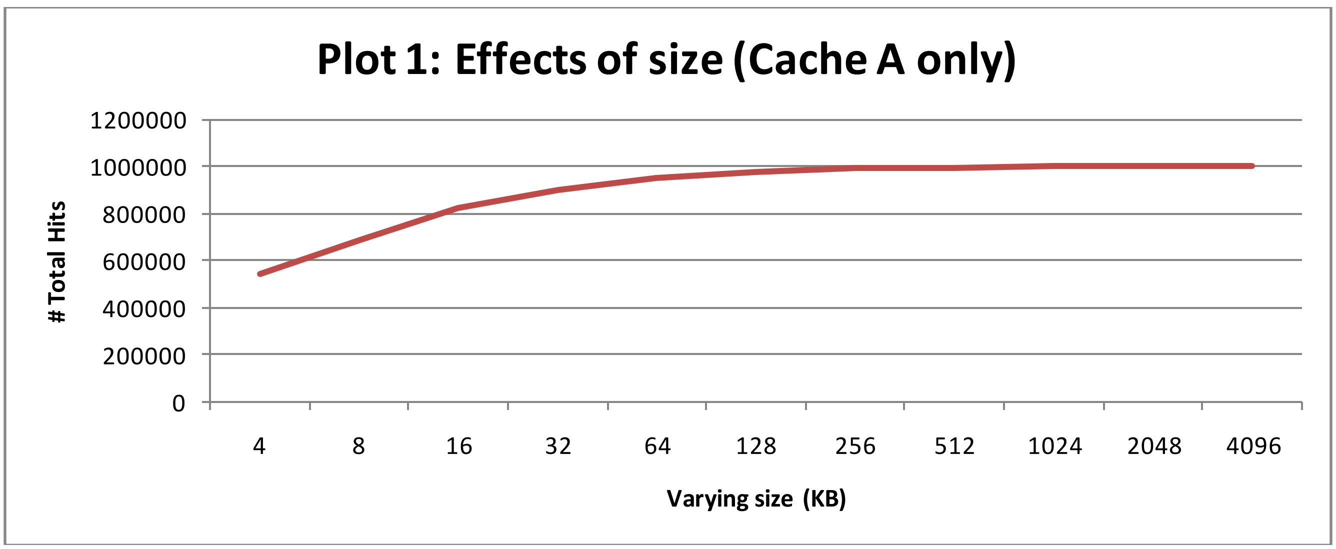
***Data on Cache B:**

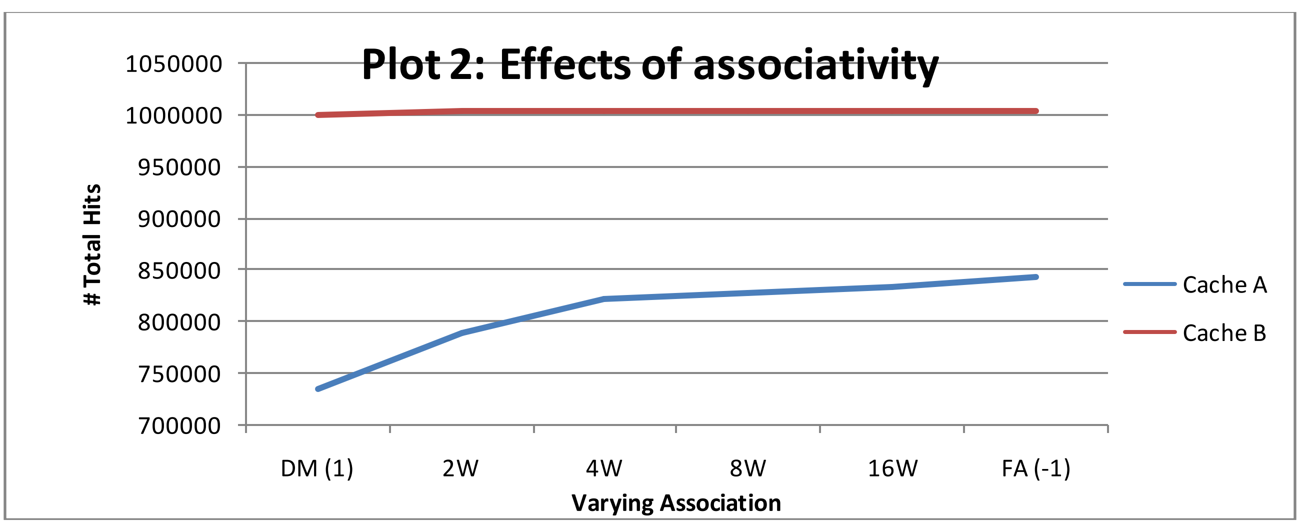
Defaults: size = 16MB, Associativity = 16W, Block size = 512 bytes,

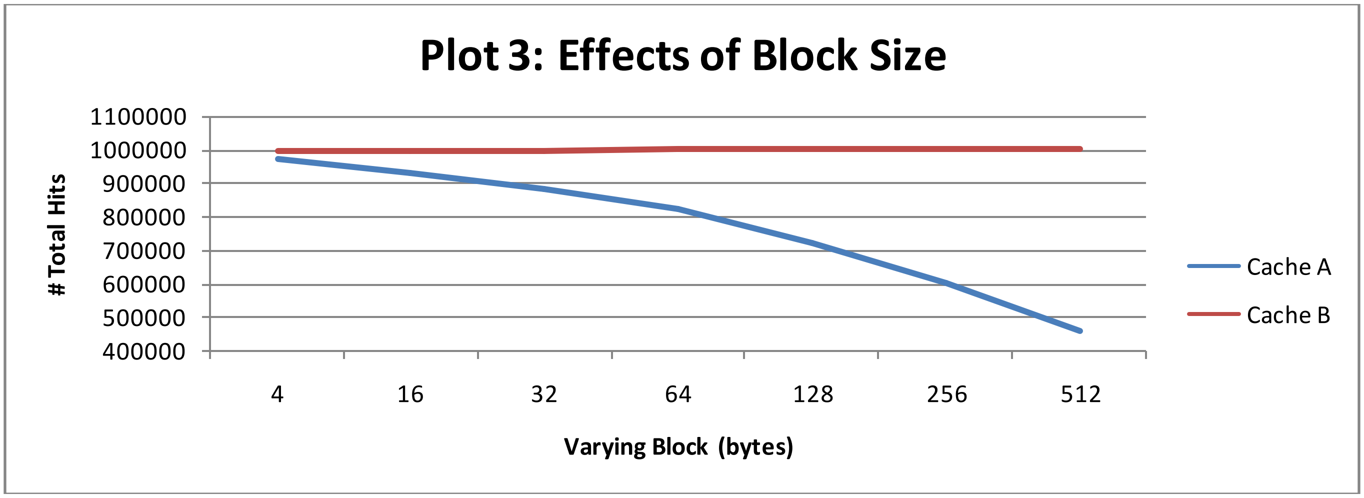
Varying Assoc.	-> Num of Hits
DM (1)	998729
2W	1002661
4W	1002930
8W	1002984
16W	1002999
FA (-1)	1003001

Varying Block size(Bytes)	-> Num of Hits
4	993714
16	996805
32	998304
64	999644
128	1000925
256	1002168
512	1002999

***Plots and Graphs For Case Study**







***Discussion for case study**

Namely the effects of various parameters on performance. Relate your discussion to the physical constraints. The cache emulator does not take in to consideration the time of access caches. From your experience building caches, discuss the physical implications of various parameters on the performance of caches. Relate these to physical constraints.

First we'll take a look at the effects of physical size, a graphical interpretation can be seen in our plot 1. From the graph and the raw data, we can conclude total hits increase as the size increases, but at a logarithmic growth rate. Thus, it will eventually hit a ceiling of diminishing returns; one cannot take advantage of the spatial locality once it hits a certain point. From the data collected, we can deduce that the 'sweet' spot is somewhere around 1KB to 4KB.

Next we'll discuss the second graph, where we vary the associativy number. Here both Cache B and Cache A have the exact same trends; like the trend in the first graph, they both have a logarithmic growth. Here the sweet spot is either 8-way assoitivity or 16-way assositivity.

The effects of block size however are different from the other two varying variables. Here, we notice that Cache B grows logrithically like the other two graphs, but Cache A actually decreases. This is probably because the ratio of block size to physical size is much larger than that of Cache B. Thus, we see a decrease number of hit performance. It's like finding a needle in a aircraft carrier, compared to finding a needle in a shoebox.