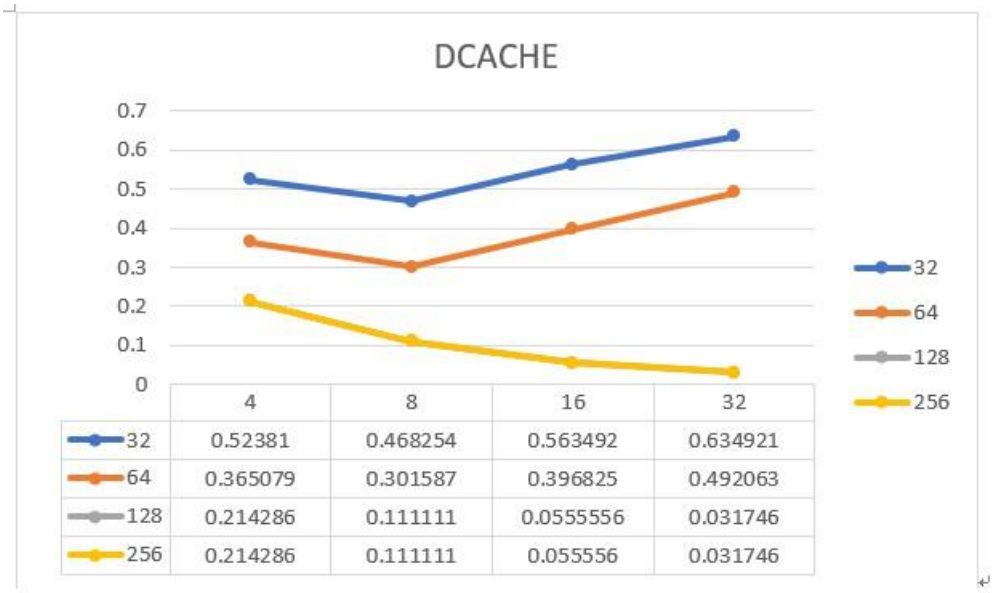
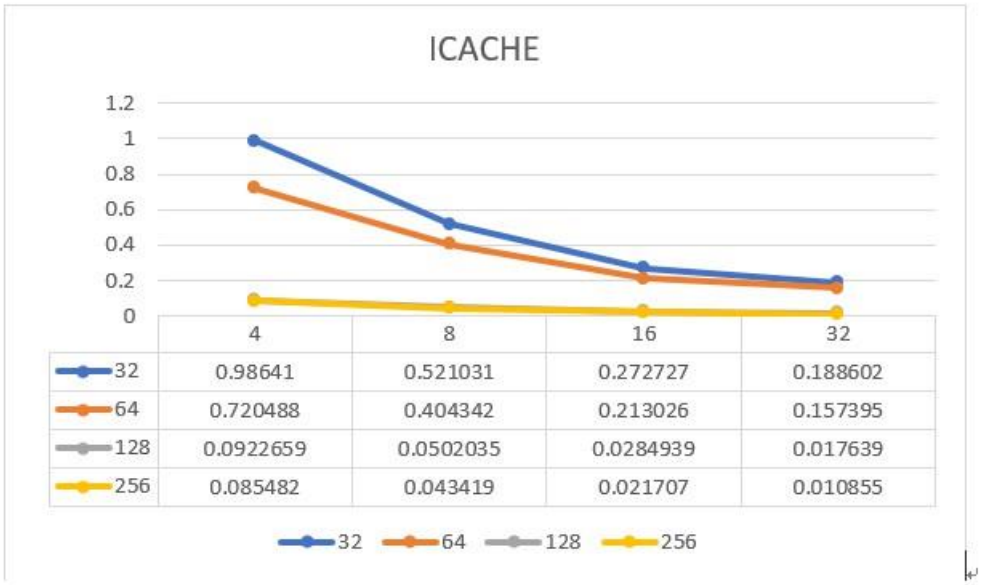
HW4 Report

姓名學號: 林哲宇 0616018, 張哲銓 0616032



	4	8	16	32
32	0.52381	0.468254	0.563492	0.634921
64	0.365079	0.301587	0.396825	0.492063
128	0.214286	0.111111	0.0555556	0.031746
256	0.214286	0.111111	0.055556	0.031746



	4	8	16	32
32	0.98641	0.521031	0.272727	0.188602
64	0.720488	0.404342	0.213026	0.157395
128	0.0922659	0.0502035	0.0284939	0.017639
256	0.085482	0.043419	0.021707	0.010855

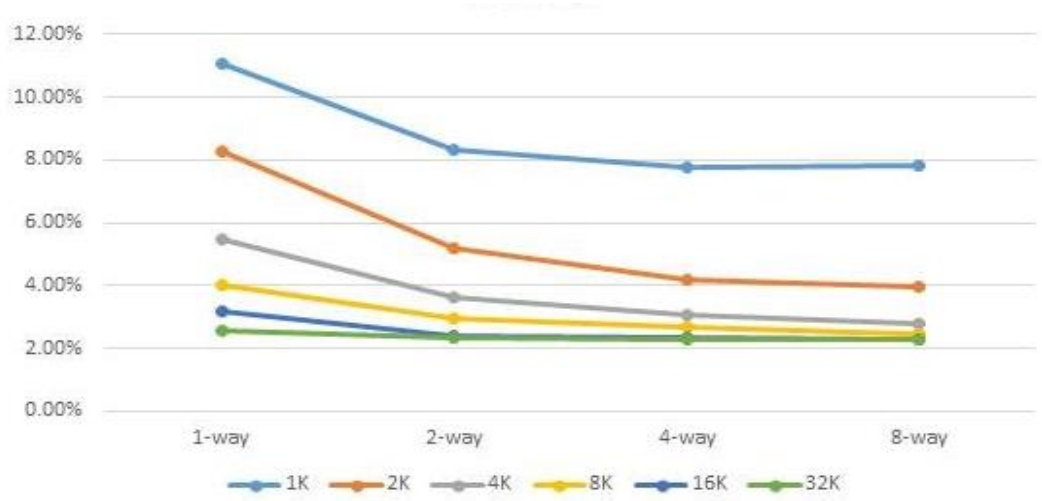
程式的部分我們把一些 module 和 entry 名稱能對應到 testbench。Cpp 檔我們把就計算總次數和 miss 的次數最後相除結果就是 miss rate。

關於 ICACHE 和 DCACHE 的 cache size 和 block size，我們觀察 Verilog

執行後生成的 ICACHE.txt 和 DCACHE.txt 之後，發現只有後兩個 bytes 有被用到，所以我們把 cache size 調成 32, 64, 128, 256，並且把 block size 調成 4, 8, 16, 32，單位為 B。

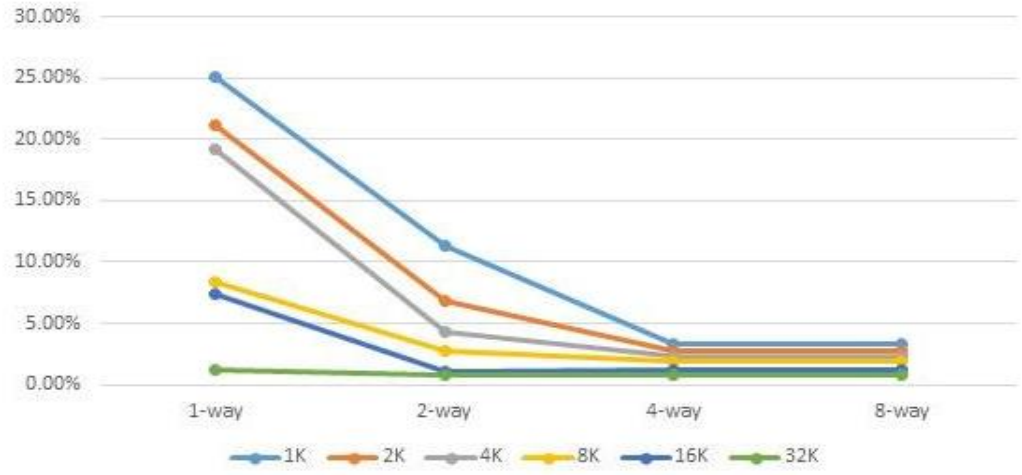
最後分析的部分，當 block size 增大，會因為 spatial locality 的特性降低 miss rate，因此可以解釋上圖折線下降；然而，如果在同個 cache size，一直提高 block size，則會導致 pollution，提高 miss penalty，因此可以解釋上圖折線圖上升。

LU:



	1-way	2-way	4-way	8-way
1K	11.07%	8.36%	7.78%	7.83%
2K	8.28%	5.18%	4.19%	3.98%
4K	5.47%	3.63%	3.07%	2.81%
8K	4.03%	2.98%	2.67%	2.45%
16K	3.16%	2.37%	2.34%	2.29%
32K	2.54%	2.33%	2.28%	2.28%

RADIX:



	1-way	2-way	4-way	8-way
1K	25.09%	11.37%	3.35%	3.33%
2K	21.22%	6.87%	2.73%	2.70%
4K	19.26%	4.37%	2.40%	2.37%
8K	8.43%	2.71%	1.86%	1.93%
16K	7.36%	1.12%	1.17%	1.24%
32K	1.23%	0.81%	0.78%	0.76%

Advanced 的部分加了 n-way set-associative，所以在程式中，需要 struct 中的 tag 變成陣列，儲存每個 record。一開始先把每個 tag 初始化成 -1，每次判斷是否為 hit，就掃過去 tag 陣列，如果是 hit，就把對應的 tag 移到陣列的第零項，之前的就依照原本順序往後順移一個；如果是 miss，就把最後一項改成新的 tag，並把它移到第零項，之前的就依照原本順序往後順移一個。這樣就可以順便解決每一筆 record 最近被用過的順序了。

分析的部分，由上圖可以很明顯發現每條線隨著 n 越大，miss rate 就越小，這是因為 block size 都固定，但是隨著 n 越大，能存的 tag 數越多，miss rate 自然就下降。Cache size 上升也會讓 miss rate 下降是因為 line 上升，能存的 index 也越多。

	1-way	2-way	4-way	8-way
1K	8560	8576	8592	8608
2K	17088	17120	17152	17184
4K	34112	34176	34240	34304
8K	68096	68224	68352	68480
16K	135936	136192	136448	136704
32K	271360	271872	272384	272896

total bits =

line * (block size *8 + 32 – log2(line) + log2(n) – log2(block size) +1)