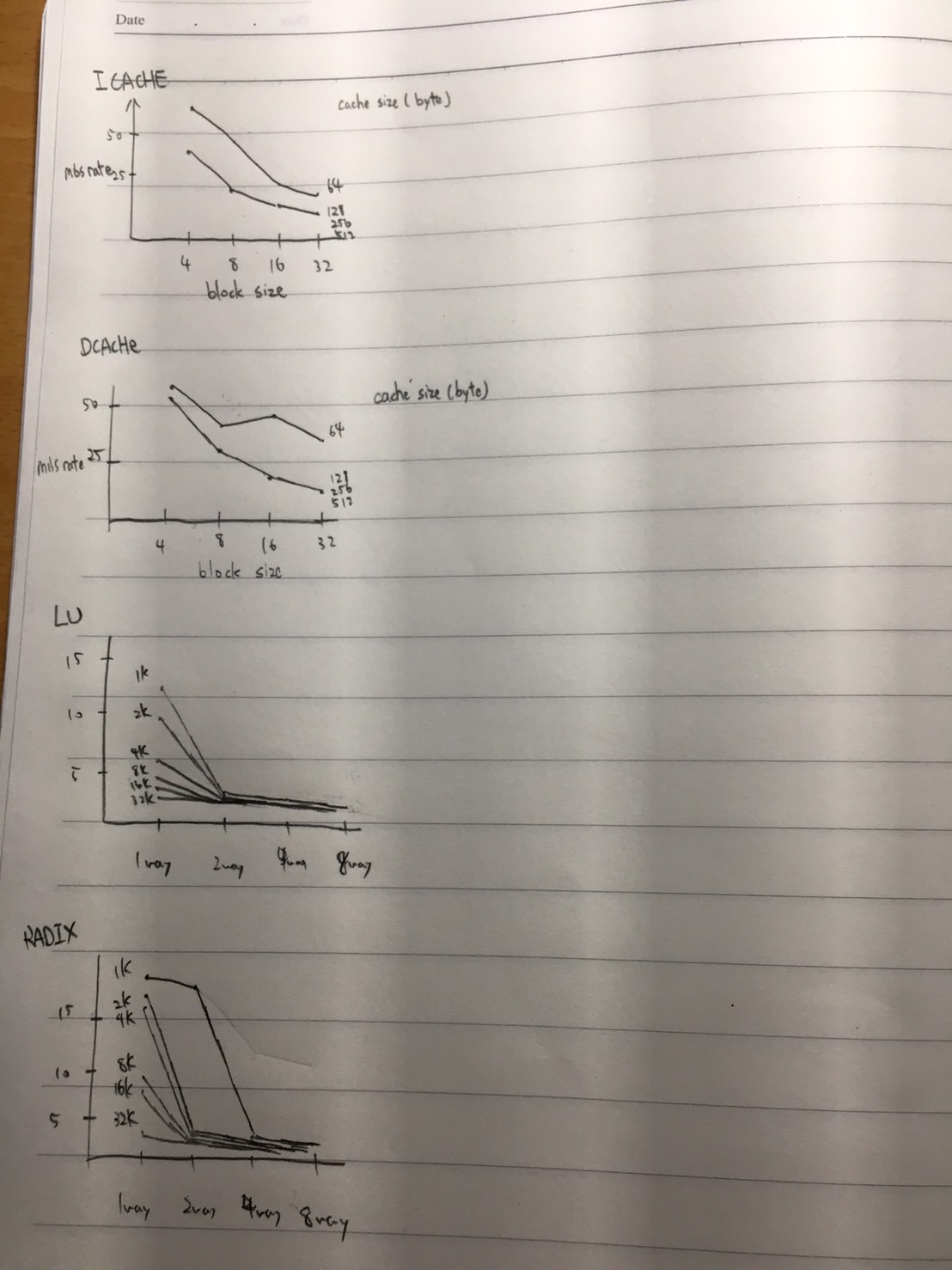
Computer Organization, Spring 2018 Lab 4: Cache Simulator

0516225 0516016

1. ICACHE simulation(忽略斜直線)
2. DCACHE simulation(忽略斜直線)



討論:

在同樣的cache size下，提昇block size可以降低miss rate，因為提昇了spatial locality。

但由於實驗結果實在太奇怪了，因為竟然都在同一條線上。跟很多同學討論後都發現大家都一樣，於是我們這組做了另外一個實驗，把cache size變小，如上圖，結果發現變兩條了，經討論後，我們認為cache size愈小愈容易影響miss rate，因為目標地址與cache裡不同的機率變高，也就是miss rate會因為不同的instruction而產生波動。我們也發現在固定的cache size下，太大的block Size 會使block數目變少而miss rate上升。

1. miss rate of LU :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Associativity  Cache size | 1-way | 2-way | 4-way | 8-way |
| 1K | 11.0681 | 4.0303 | 2.5424 | 2.3407 |
| 2K | 8.2777 | 3.1623 | 2.3407 | 2.3252 |
| 4K | 5.472 | 2.5422 | 2.3252 | 2.2787 |
| 8K | 4.0303 | 2.3407 | 2.2787 | 2.2787 |
| 16K | 3.1623 | 2.3252 | 2.2787 | 2.2787 |
| 32K | 2.5422 | 2.2787 | 2.2787 | 2.2787 |

1. miss rate of RADIX :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Associativity  Cache size | 1-way | 2-way | 4-way | 8-way |
| 1K | 25.0864 | 8.4317 | 1.2312 | 1.0109 |
| 2K | 21.2169 | 7.3559 | 1.0109 | 0.7517 |
| 4K | 19.2569 | 1.2312 | 0.7517 | 0.7517 |
| 8K | 8.4317 | 1.0109 | 0.7517 | 0.7517 |
| 16K | 7.3599 | 0.7517 | 0.7517 | 0.7517 |
| 32K | 1.2312 | 0.7517 | 0.7517 | 0.7517 |

討論:

LU分解是矩陣分解的一種，可以將一個矩陣分解為一個下三角矩陣和一個上三角矩陣的乘積

* cache size增加，miss rate會減少
* 同一個cache size中，當n增加，miss rate也會跟著下降，但最低也只能到2.3%

RADIX 是 一種SORT

* miss rate最低可以降到0.75%，效益非常高。

Total bits 與 CACHE SIZE呈正相關

**Detailed description of the implementation:**

用hw3的CPU來做，因為testbench裡面有些接線不一樣所以還要再做一些修改。跑完會有ICACHE跟DCACHE兩個檔案，再用提供的C++檔案來更改blocksize和cachesize即可得到數據，使用FOR 迴圈。然後要做LRU，我是先用一個變數used來計算現在跑到哪裡每讀一次數據就+1，然後假設miss就把它存到我另外設的tagused裡面，然後之後只要判斷tagused最小的就是符合最久沒用過的。過程中為了方便建構圖表，我用了C++ 的 OFSTREAM 寫CSV檔。