Q: 關於作業輸出?

A:

Input file (輸入檔名)

Demand fetch (需要處理的指令數)

Cache hit (有在快取找到資料的指令數)

Cache miss (沒有在快取找到資料所以需要另外處裡的指令數)

所以Demand fetch = Cache hit + Cache miss

Miss rate (失誤比例，Cache miss / Demand fetch) (精準到四位小數)

Read data (要讀資料的指令數，也就是lable 0數量)

Write data (要寫資料的指令數，也就是lable 1數量)

Bytes from Memory (總共從記憶體傳輸了多少byte到Cache)

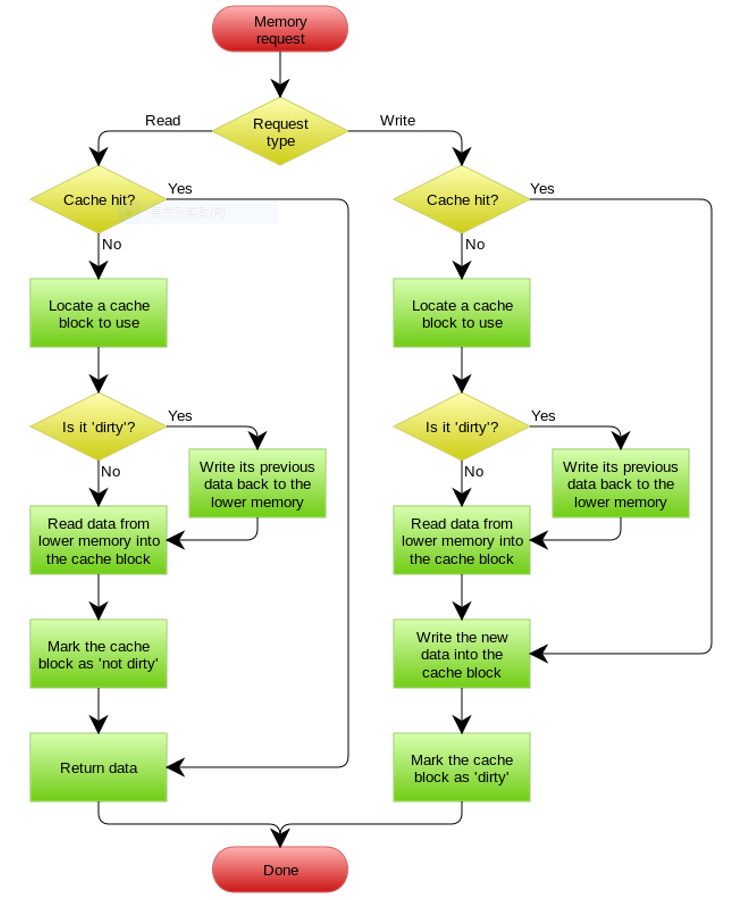
Byte to memory (總共從Cache傳輸了多少byte資料到記憶體)

有跟記憶體溝通的才算

然後讀一次跟寫一次(就Cache Miss時)的資料量跟Block Size一樣

Q: 要怎麼寫資料?

A: 看這個流程圖：



寫的東西都是寫到快取 不管有沒有Miss

然後只要有Miss 也要把要寫的記憶體位置本來的資料先讀進來 再改

只要有寫過資料 Dirty Bit就改成1

老師上課有講假如Cache Miss也可以直接往記憶體送

但這次作業的要求不是這樣…

Q: 為什麼gcd.din明明就有100003個指令 範例解答裡面只有100002？

A: 最後一個2 ffffffff是多的 可以看到ffffffff就忽略 或是直接把最後一行給砍掉

Q:說明測資(.din檔)

A:

Label 0(Read Data): 用對應的Address從cache讀取資料。

Label 1(Write Data): 寫入資料到對應的Address。

Label 2(Read Instruction): 用對應的Address從cache讀取指令。

遇到Label 2 時的行為: 以對應的Address 到cache 讀取指令，若有找到則

cache hit。若沒有找到則cache miss，若cache miss，則根據address

將對應的Cache entry中的資訊取代或更新。

Address 都是32bit長，圖中未滿32bit視為前面補16進位的0。

Ex: 408ed4 實際上是00408ed4。

Q: 通通都執行完之後 要不要把Cache有修改過的資料通通寫回記憶體?

A: 要！

Q: 使用語言

A: 盡量用C/C++

Q: 使用的替換法則?

A: FIFO 和 LRU

Q: 想要更多的範例答案!

A: 終於來了!

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 測試條件 | 256 128 f LRU spice.din | 8 4 8 LRU spice.din | 16 16 2 LRU spice.din | 256 32 2 LRU spice.din |
| Input file | spice.din | spice.din | spice.din | spice.din |
| Demand fetch | 1000001 | 1000001 | 1000001 | 1000001 |
| Cache hit | 999299 | 950702 | 984871 | 997911 |
| Cache miss | 702 | 49299 | 15130 | 2090 |
| Miss rate | 0.0007 | 0.0493 | 0.0151 | 0.0021 |
| Read date | 150699 | 150699 | 150699 | 150699 |
| Write data | 66538 | 66538 | 66538 | 66538 |
| Bytes from Mem | 89856 | 197196 | 242080 | 66880 |
| Bytes to Mem | 19456 | 24648 | 31584 | 14496 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 測試條件 | 8 32 1 FIFO gcc.din | 32 8 f FIFO gcc.din | 32 8 f LRU gcc.din | 32 8 4 FIFO gcc.din | 32 8 4 LRU gcc.din |
| Input file | gcc.din | gcc.din | gcc.din | gcc.din | gcc.din |
| Demand fetch | 1000002 | 1000002 | 1000002 | 1000002 | 1000002 |
| Cache hit | 924129 | 964755 | 969624 | 962302 | 966355 |
| Cache miss | 75873 | 35247 | 30378 | 37700 | 33647 |
| Miss rate | 0.0759 | 0.0352 | 0.0304 | 0.0377 | 0.0336 |
| Read date | 159631 | 159631 | 159631 | 159631 | 159631 |
| Write data | 83030 | 83030 | 83030 | 83030 | 83030 |
| Bytes from Mem | 2427936 | 281976 | 243024 | 301600 | 269176 |
| Bytes to Mem | 352192 | 47744 | 42032 | 49536 | 43576 |