

# MOESI Simulator report

---

102062111 林致民

## 實作方式

---

實作的細節都已經寫在source code的註解上面，這裡大概整理一下實作方式：

1. 先把p0.tr ~ p3.tr所有的trace按照timestamp從小到大排好，如果時間一樣，就按照processor ID從小到大排好。
2. 確保執行的優先序之後，我設計了兩個主要的function：`read` `write`，`read`負責處理當值讀近cache時應該採取的策略，而`write`則是把直寫進某個變數所採取的策略。
3. Write 的作法：
  1. 偵測是否為write MISS：先判斷當前這個block的tag是否一致，如果在cache的block的tag與要寫入的block tag不同，代表這個block將會被換掉，因此在這個階段判斷當前的block是否為M 或 O State，如果是的話就把值write back回Memory，累計一次writeback次數
  2. 接下來看與其他processor中cache的狀態（tag一樣），假設當前的狀態是I，如果可以找到其他狀態是M或O的cache block，就計算一次從其他processor到當前processor的transfer。
  3. Invalid 其他 processor中的狀態，並且統計除了Invalid以外的State轉移到Invalid state的個數
  4. 之後要更改這個block，所以把這個block改成M state，代表現在已經更改，但還沒寫回memory。
4. Read的作法：
  1. 偵測是否為read MISS：跟Write偵測block方式差不多，如果當前的state是M or O，累計write back。
  2. 偵測其他processor中相同block，假設有找到其他相同的block，而且其state為M or O，就把資料copy到這個processor來，累計一次，如果是M的state把它設成O
  3. 偵測是否有其他processor中是否相同的block，有的話讓這個block變成S state，否則就是E State。如果其他processor是E State，就讓他變成S State。

5. 要把所有dirty全部write回去，因此當前如果是O or M state，就需要把這些block writeBack回Memory