Cache基本  
write back

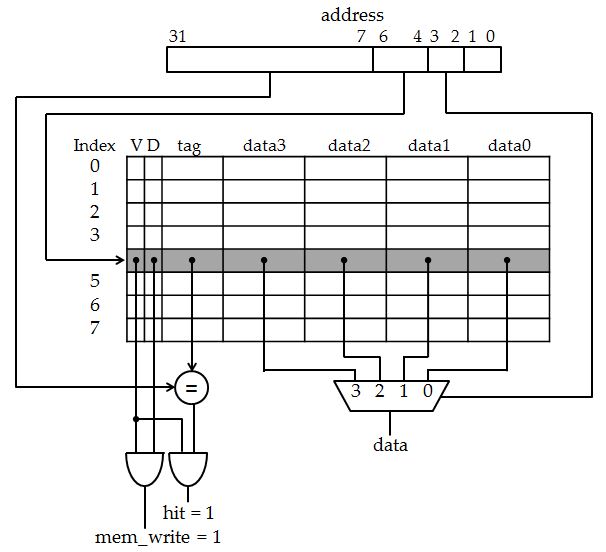
L1: 32words = 4words \* 8blocks

L2: 256words =

Cache優化

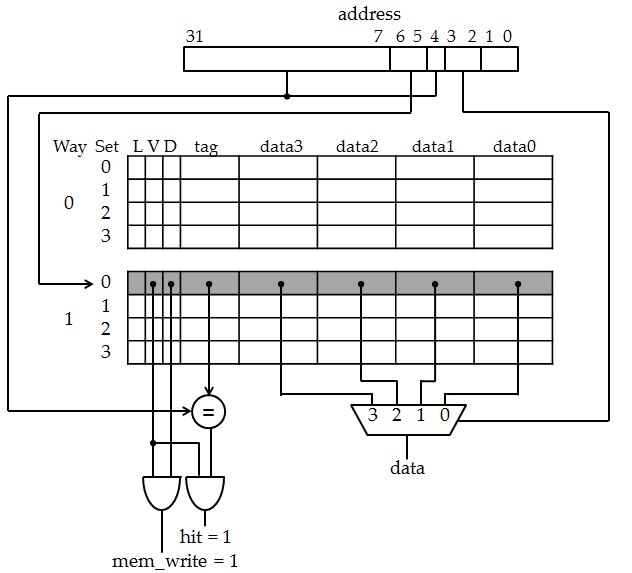
1. I cache and D cache

一般來說，I cache的資料會比D cache的資料還要照順序拿取，因此本來設計是I cache為direct mapped，D cache為N way set associative cache，但是後來發現如果是依照助教給的指令，I cache比較容易跳來跳去，所以反而是I cache會需要N way set associative cache。



2. N way set associative

對N way set associative的cache來說，很重要的一點是它的replacement policy，在這次實作中是採用LRU (Least recently used)，而要判斷各個way之間的新舊，其中一個方式是在每個block加上一個counter，每操作一次，counter就加一，進而排出不同way之間的新舊順序，但是這種要確切得知先後順序的方式既浪費面積又耗時，因此對於4 way以上的cache採用pseudo LRU，在每個block加上一個LRUbit，當使用到的時候把LRUbit拉起來變成1，因此LRUbit為1的block會被優先替換，如果同時遇到多個block的LRUbit都是1的話，則取第一個block替換，雖然可能沒有辦法確切得知每個block的新舊順序，但是考量到面積、時間等等的整體效益，pseudo LRU是較可行的方式。



3. L2 cache

4.

