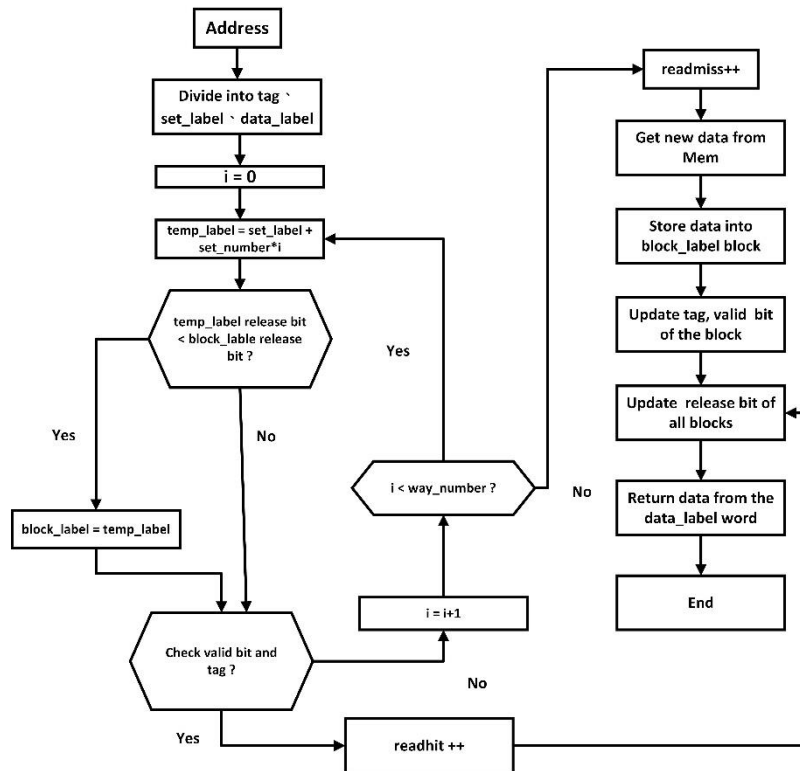


計算機結構與數位系統設計 作業四

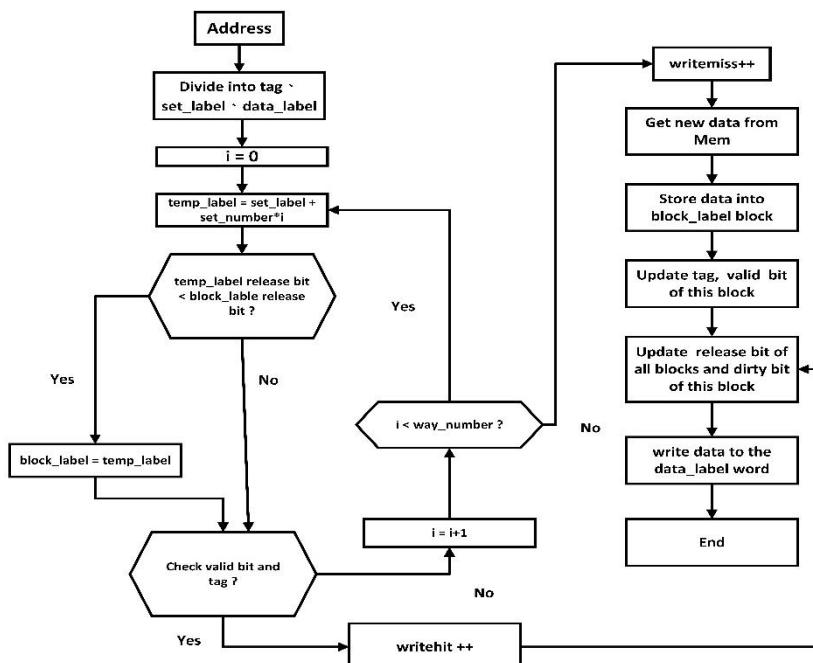
B03901026 許凱傑

- I. Q1: Please draw the flow chart to describe the read/write behavior of your cache. Start from address sent to cache, and end in return data. Vice versa.

Read



Write



計算機結構與數位系統設計 作業四

B03901026 許凱傑

II. Q2: Please modify the test bench, and compare the hit rate and miss rate of different structure of cache (direct-mapped, 2-way, 4-way, or fully-associative).

1 Q2-1: Build a table of hit/miss as TABLE I/II for both quicksort and mergesort.

hit rate(%)_	Mergesort	Quicksort
direct-mapped	80.5049	88.9901
2-way associative	84.993	91.1867
4-way associative	86.9693	90.9992
fully-associative	86.8418	90.9992

2 Q2-2: Discuss why there's difference of hit rate between different structures?

因為如果一個 set 有多個 block 可以使得不會每次不同的 address 就一定會 miss 而有緩衝。舉例來說 cache 有編號 0-7 的 block，如果現在需要取得記憶體位置是 1、9、1、9、17、25、17、33、9、1 的資料，direct-mapped 的 miss 是 10，但 2-way 是 7，4-way 是 6，fully 是 5。

3 Q2-3: Is hit rate keeps going up as the way_number goes up, why or why not?

並非絕對會升高，因為如果不是一直取到在較低 way_number 時會歸屬同個 set 的資料，fully-associative(或較高 way_number)反而會全部放到同一個 set 無法做有效的區分，導致 data 更頻繁地被替換，

III. Q3: Please describe the difference between quicksort algorithm and merge sort algorithm.

1 Q3-1: What the computation complexity of quicksort and merge sort algorithm?

	Time			Space
	worst	average	best	
Mergesort	$O(n * \log(n))$	$O(n * \log(n))$	$O(n * \log(n))$	$O(n)$
Quicksort	$O(n^2)$	$O(n * \log(n))$	$O(n * \log(n))$	$O(\log(n))$

2 Q3-2: Which algorithm has the higher hit rate? Why?

Quicksort 有比較高的 hit rate，但並非一定，只是因為這次實作是透過 array，所以記憶體位置都在附近，若是 linked-list 可能不一定有如此明顯的差別。因為 quicksort 是 in-place 也就是比較、排序、放入都是在同一塊記憶體位置，但是 Mergesort 是先全部打斷後，再一一排序放到另一塊空間，所以相較來說比較容易發生 miss。