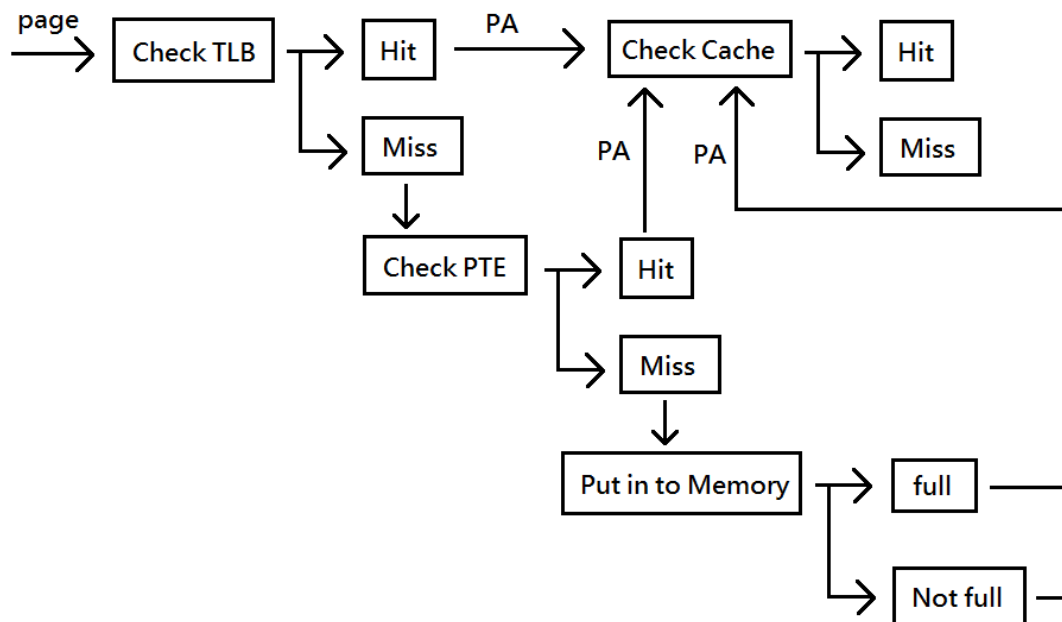


1) Project Description

1-1) Program Flow chart



1-2) Detailed Description

l 和 d 基本上一模一樣，唯一的不同就是 l 是用 PC，d 是用 address，以下統一用 VA 稱呼 PC 和 address。

首先 page 就是 VA 除以 page_size，用來檢查 TLB、PTE 和放入 Memory。

PA 則是 VA 取 page_size 的餘數，再加上 PPN 乘以 page_size，用來檢查 cache。

檢查都是從 TLB 開始，以下分為幾種情況：

- (1) TLB_hit → Cache_hit。
- (2) TLB_hit → Cache_miss。
- (3) TLB_miss → PTE_hit → Cache_hit。
- (4) TLB_miss → PTE_hit → Cache_miss。
- (5) TLB_miss → PTE_miss → 放入 Memory → Cache_miss。

再來說明每個部分是怎麼放入的。

TLB:

TLB 先檢查 page，如果找到的話，就不用放入，將那個位置的 LRU 更新就好。如果沒找到，就先找 invalid 的放入，如果沒有，就放入 LRU 最大的，並更新。

PTE&Memory:

因為 PTE 大於或等於 Memory 的 size，所以不會有需要覆蓋的情況。

只要記錄目前有哪幾個 page 有在 Memory 中就好。

PTE 檢查 page 時，有找到的話就不用做事。
沒找到則要將該 page 設為正在 Memory 中，並將該 page 放入 Memory。
而放入 Memory 時，則分為未滿和已滿兩種情況。
未滿的時候就找 invalid 的放，沒有覆蓋的情形。
已滿的話則用 LRU 來判斷覆蓋哪個 page，
且被覆蓋的 page 的內容物有出現在 PTE、TLB、Cache 的話，就要將它們刪除。
而有沒有在裡面的判斷，PTE 和 TLB 是看 page，Cache 則是看 PPN。
另外刪除時要更新 TLB 的 LRU 及 Cache 的 MRU 及 Valid_bit。

Cache:

Index 決定放入的 block，而 tag 則是內容物，如果兩個都相同就 hit。
Index 為 $PA/block_size \% set_num$ 。
Tag 為 $PA/block_size/set_num$ 。
而放入是依據 MRU，如果放下去時整個 set 都是 1 的話，就要將該 set 更新。
另外如果出現因為 Memory 覆蓋而導致刪除時，將該 block 的 Valid_bit 設為 0。
放入時 Valid_bit 的檢查優先於 MRU。

2) Test case Design

2-1) Detail Description of Test case

以下是我的 testcase:

```
lw $0 0($0)
lw $0 16($0)
lw $0 0($0)
lw $0 32($0)
lw $0 0($0)
lw $0 64($0)
lw $0 128($0)
lw $0 192($0)
lw $0 256($0)
lw $0 0($0)
halt
```

因為 i 和 d 基本上一樣，所以我只測了 d 的部分，全部都是用 lw。
我的 testcase 分為兩部分，一個是前五行，另一個是第五行到最後。

前五行是針對 default 參數設計的，因為 default 的 dMemory 大小只有 2。
所以一開始放了 0 和 16 兩個 page 後，在第四行放 32 的時候，會將 0 的 page 刪除，因此第五行會 miss。
比較特別的是，因為第三行 TLB_hit，所以沒有更新 memory 的 LRU，所以第四行時被刪除的 page 還是 0 的。

而第五行以後則是想測 TLB_miss 但 PTE_hit 的情況，但這種情況不會出現在 default，只有可能出現在 memory_size 大於 TLB 的 entries 的時候。

以助教公布的第三個參數為例，512 1024 128 64 64 4 8 32 4 4。

TLB 只有 4 個 entries，而 Memory 和 PTE 則有 16 個。

因此在放完 0、64、128、192 後，在放 256 的時候，TLB 會將 0 的 page 覆蓋。

但 PTE 有 16 個 entries，所以不會覆蓋，且 Memory 也沒覆蓋，因此不會刪除 PTE 裡的 0 的 page。

所以在倒數第二行，再次出現 0 的時候，TLB 會 miss，而 PTE 會 hit。