# Computer Organization 2020

## HOMEWORK 4

系級: 資訊甲 111　　學號: H34071136　　姓名: 呂兆凱

## 問題(Question)

Q1. How do you know the number of block from input file?

the number of block = Cache size (KB) * 1024 / block size (Byte)

Q2. How do you know how many set in this cache?

Read the file to determine how many ways based on the number of representing associative first.

Associative is 0, way=1.

Associative is 1, way=4.

Associative is 2, way=the number of block. (fully)

Then calculate the number of set by “the number of block / way.

Q3. How do you know the bits of the width of the Tag ?

Offset bits = $\log_2$(block size)

Index bits = $\log_2$(the number of set)

Tag bits = 32 – Offset bits – Index bits

Q4. Briefly describe your data structure of your cache.

先將讀進的16進位轉為10進位，然後將數字除以block size 所得的商即為 block address

將 block address 除以 number of set ，餘數為 index ，商為 tag (要存進 cache 裡面的數字)

最後根據不同的 associative 和 replacement policy 作相對應的處理

Q5. Briefly describe your algorithm of LRU.

Cache為一個二維陣列，row為way數，col為set數。

row從0~最後一row為最近使用到的順序，最近使用的會放在最後一個row。

所以當需要剔除時就是從row的第0個剔除，然後全部往前進一格，最新的data放在row的最後一個。

如果是way是滿的話，就把當次的tag放置在最後一個，如果way還沒放滿，則在全部往前一個後找到第一個不為空的row放入tag。

Q6. Briefly describe your algorithm of your policy.

My policy : MRU(Most Recently Used)

最近用到的檔案先被刪除

row從0~最後為最近使用到的順序，最近使用的會放在最後一個。

所以當需要剔除時就是從row的最後一個剔除，然後全部往前進一格，最新的data放在row的最後一個。

如果是way是滿的話，就把當次的tag放置在最後一個，如果way還沒放滿，則在全部往前一個後找到第一個不為空的row放入tag。

Q7. Run trace2.txt, trace3.txt and then makefile to get the miss rate and put it in your report.

Trace2.txt Miss Rate : 0.000597

Trace3.txt Miss Rate : 0.000076

## 心得(Report)

Realize how cache works based on different associative and replacement policy clearly, including FIFO, LRU, etc.

I think the hard part is the implementation of LRU(How to judge which one is used last).