Instruction Cache Design

1043335 賴詩雨

## Source Code

* 實作一 ( 紅框內為填上的答案 )

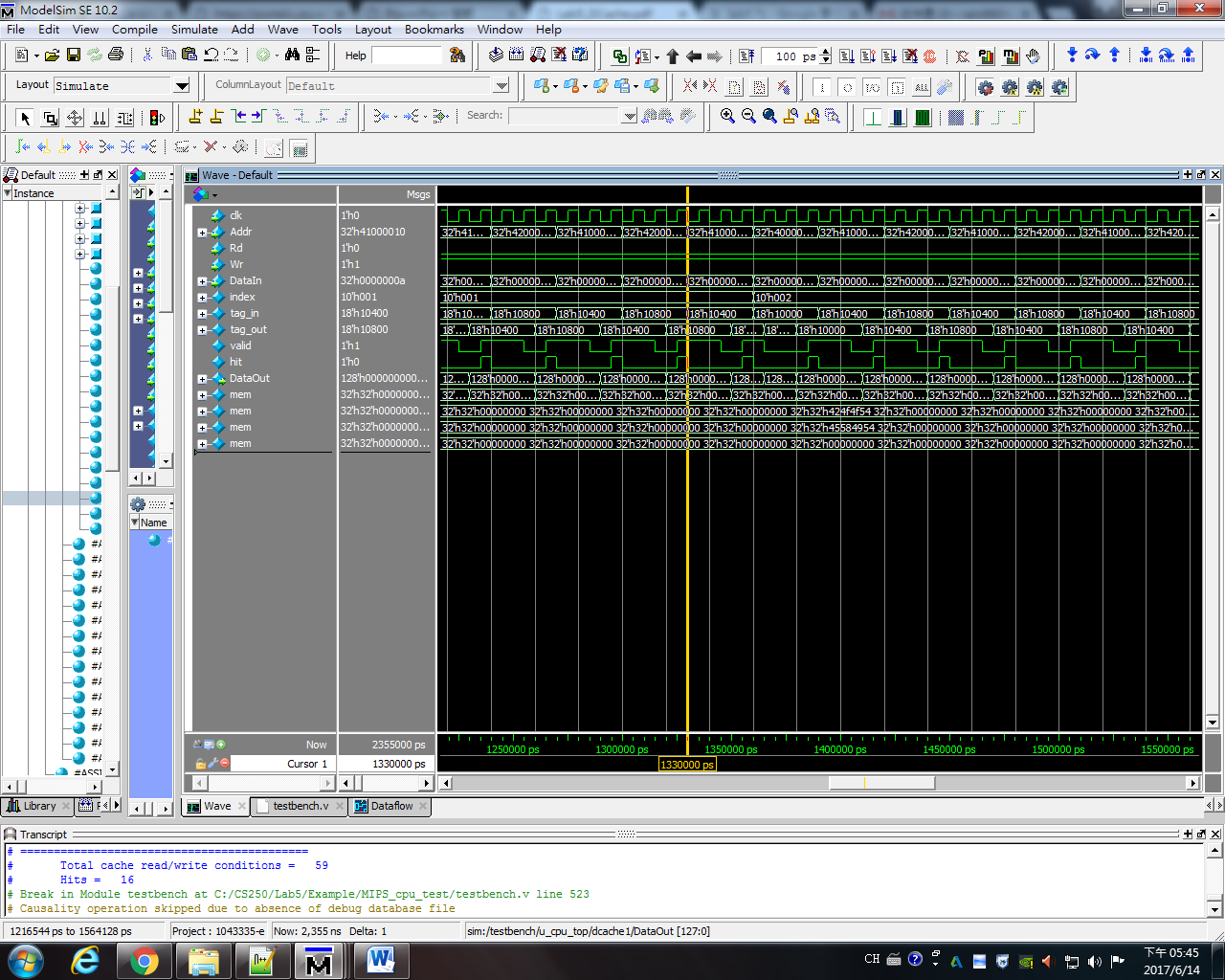
|  |
| --- |
|  |

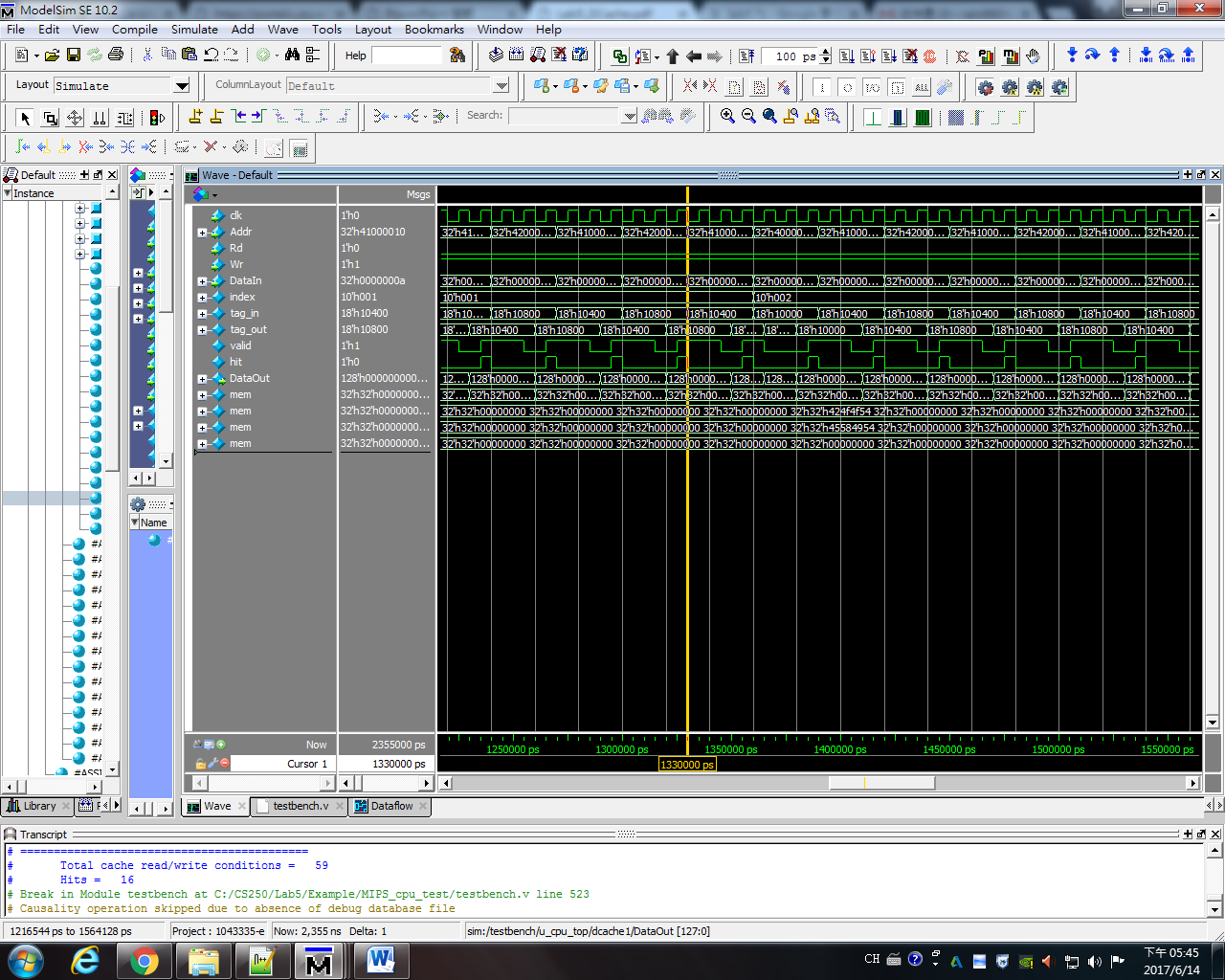
* 實作二 ( 紅框內為填上的答案 )

|  |
| --- |
|  |

## 實作結果

* Example波形圖

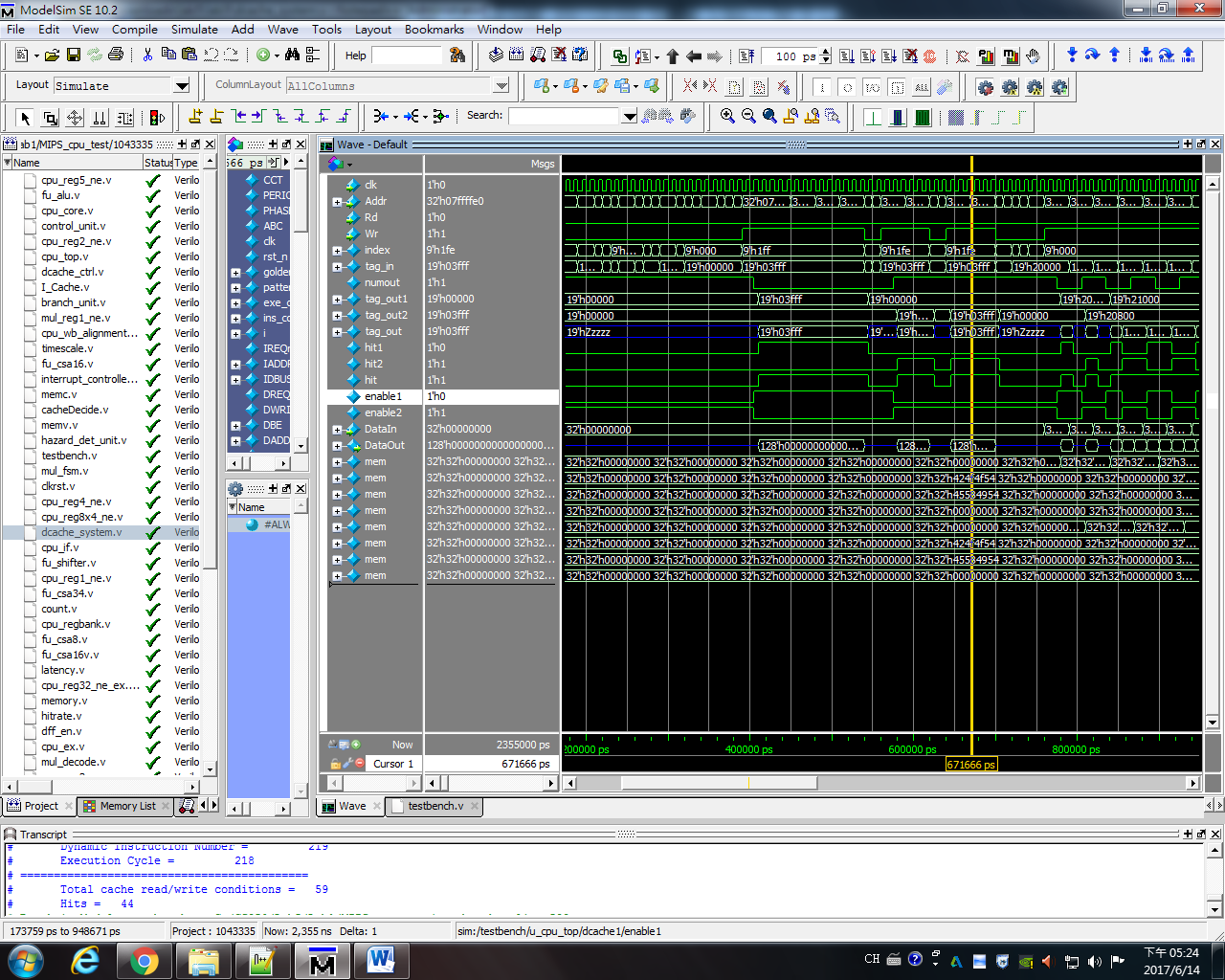


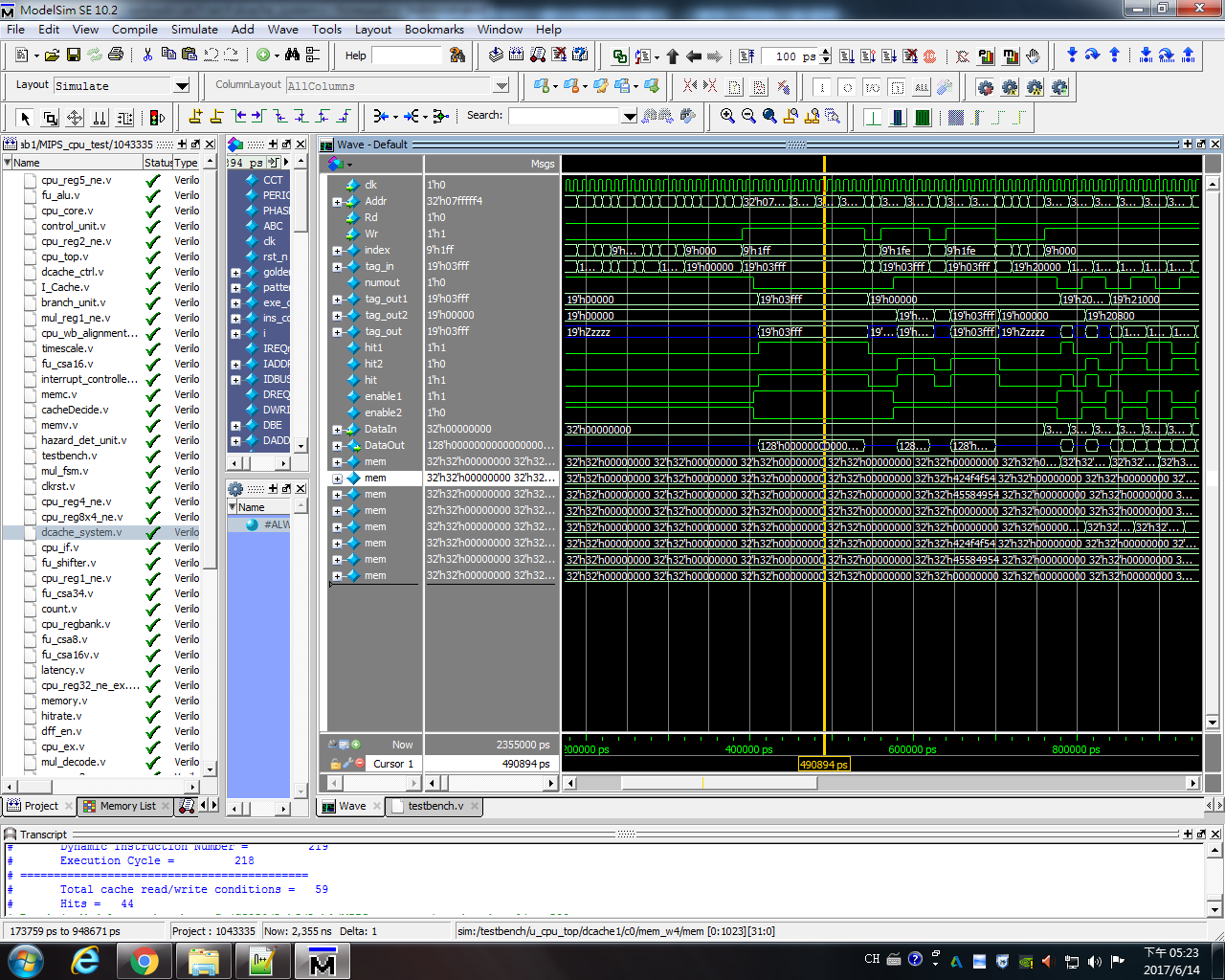


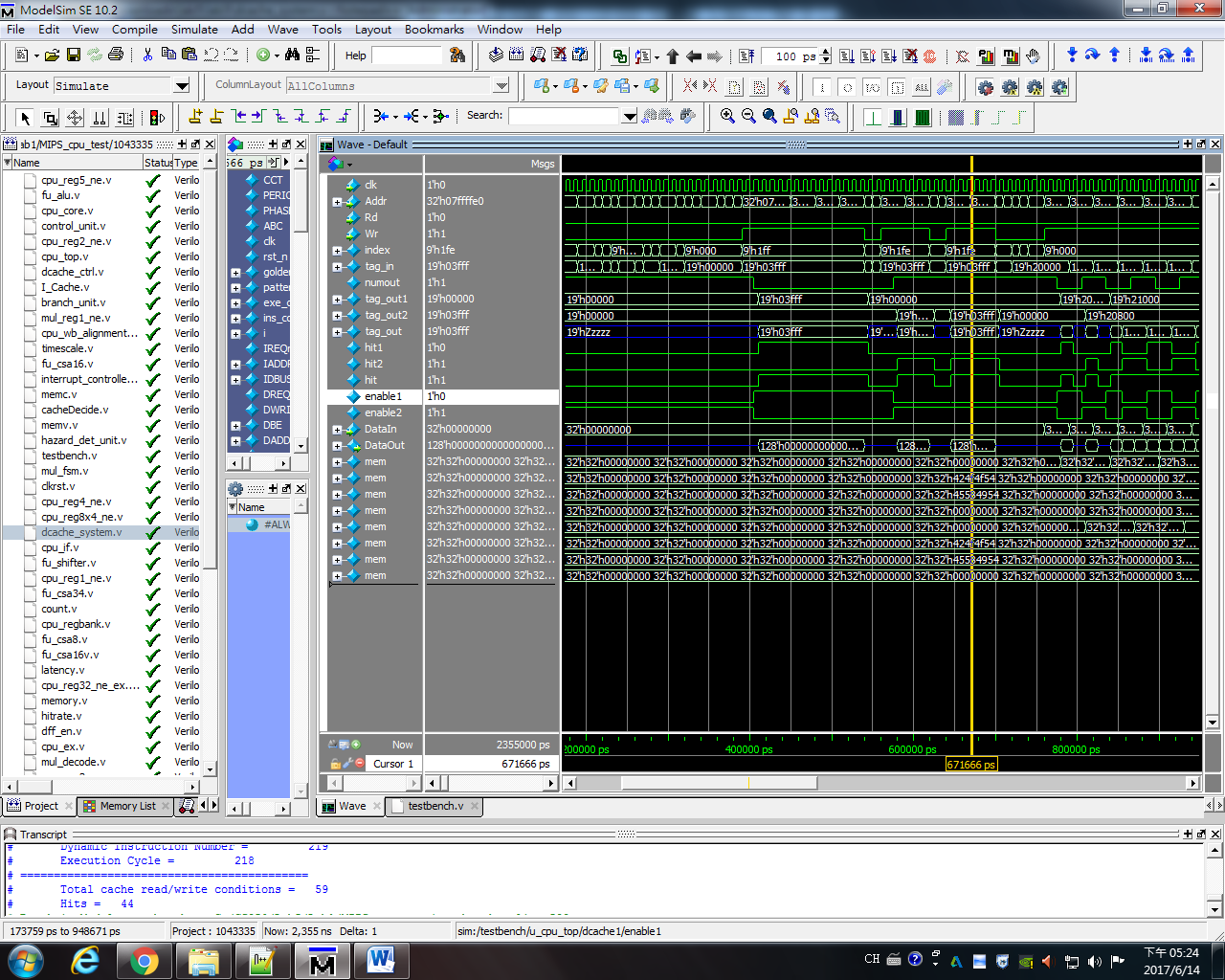
* Example分析

1. Hit rate = 16/59 = 0.271

* 實作一波形圖





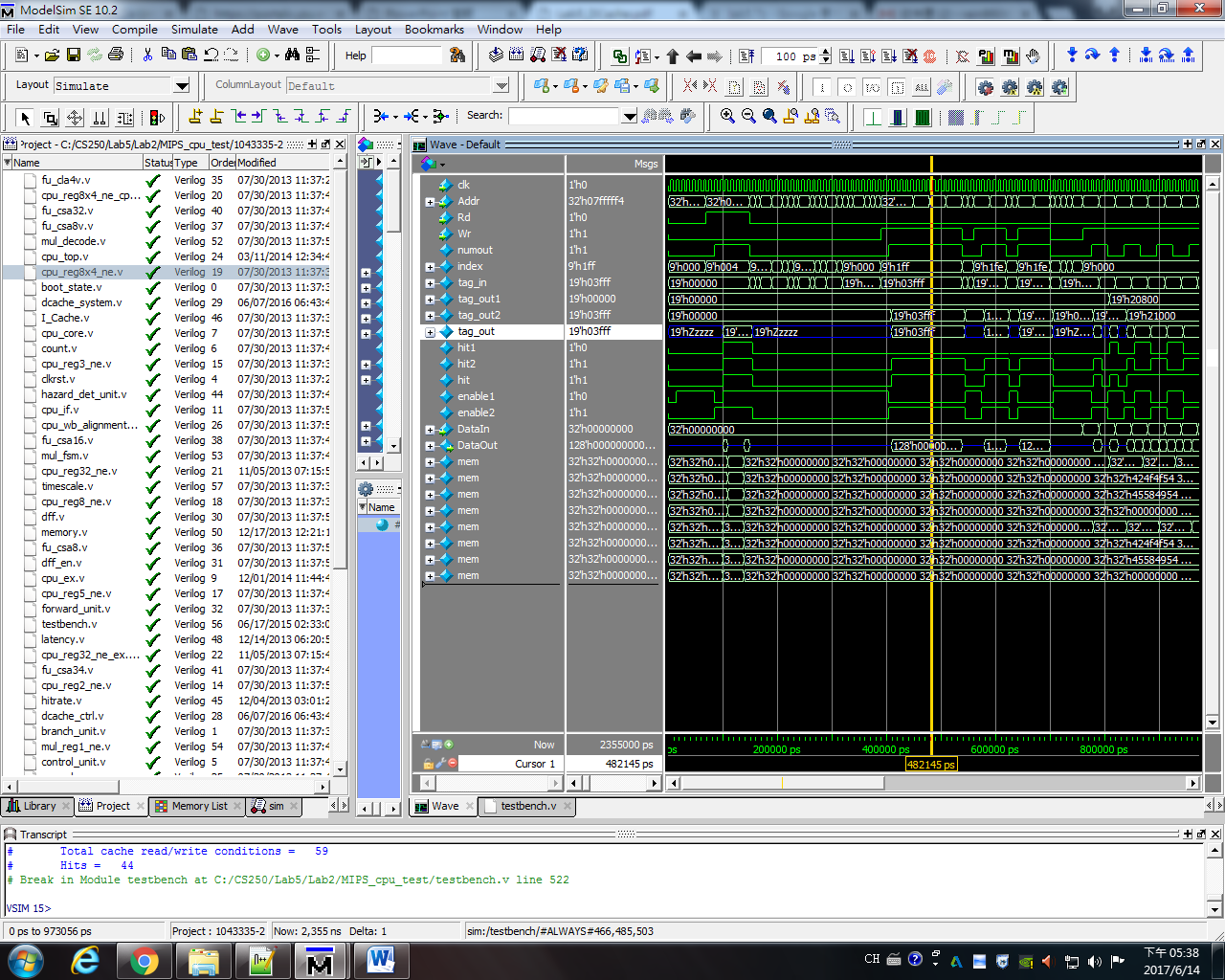


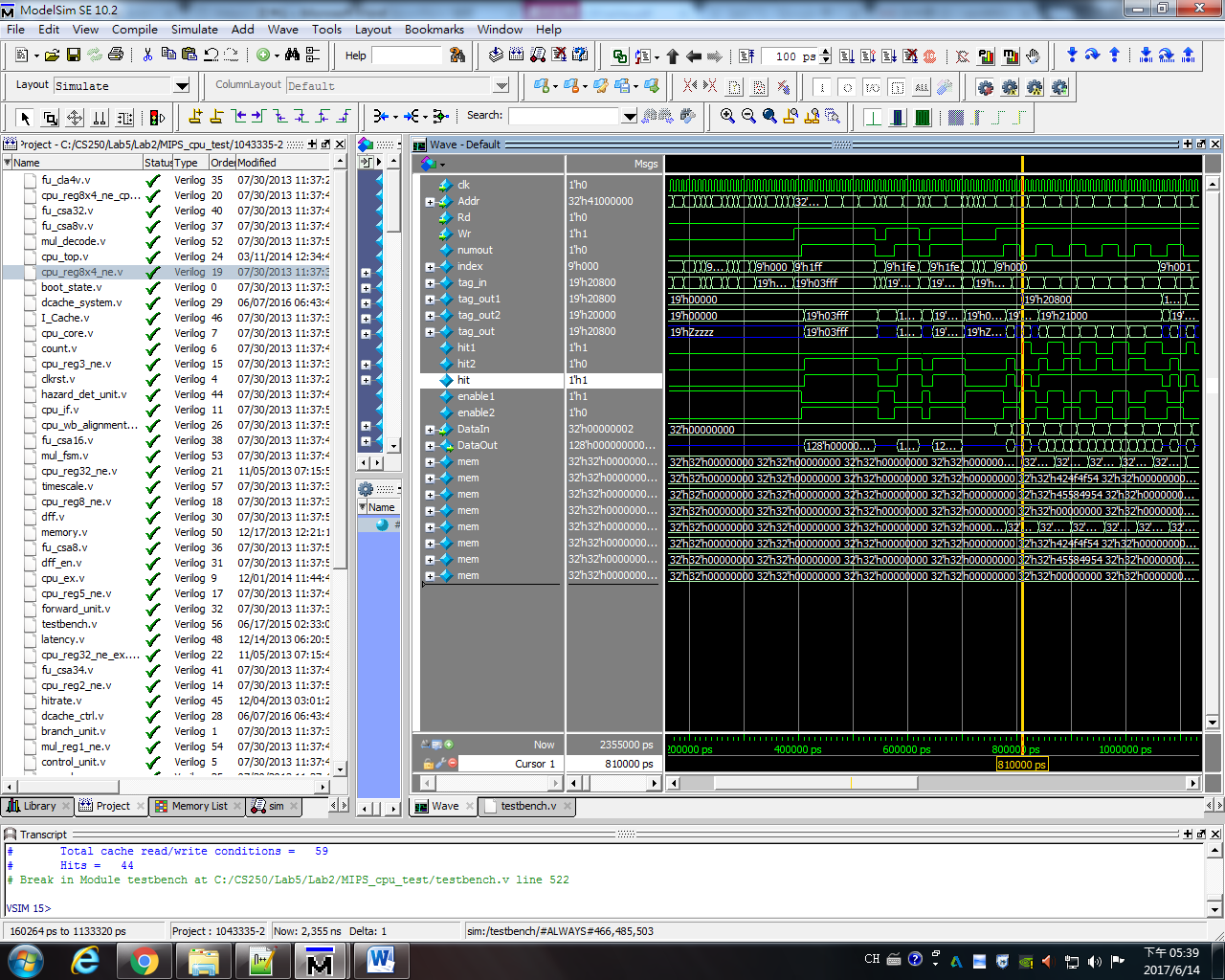
* 實作一分析

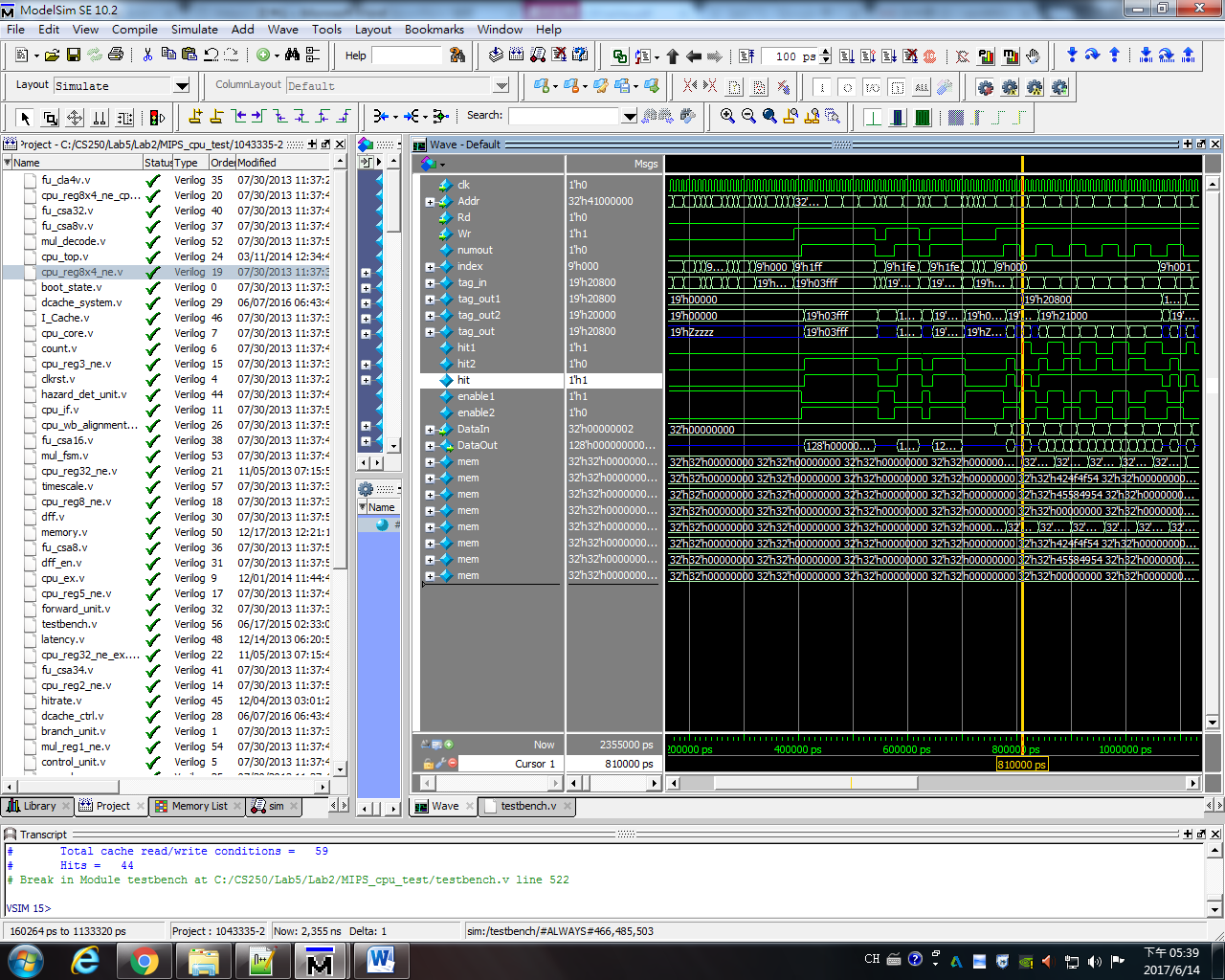
1. 第一張圖的黃色框框中，可以看到way1發生hit ( hit1 = 1 )，所以tag\_out的值是tag\_out1的值。
2. 第一張圖的兩個紅色箭頭，當numout = 0 ( 決定way1離開 ) 時，enable1 = 1 ( way1 要寫入 ) 。
3. 第二張圖的黃色框框中，可以看到way2發生hit ( hit2 = 1 )，所以tag\_out的值是tag\_out2的值。
4. 第二張圖的兩個紅色箭頭，當numout = 1 ( 決定way2離開 ) 時，enable2 = 1 ( way2 要寫入 ) 。
5. Hit rate = 44/59 = 0.7457

* Example V.S.實作一 Hit rate 不同之原因

Example為1 way，而實作一為2 ways，因此Example沒有實作一擁有的way2，因此只要進來的資料Cache內無，就要replace，這也是為什麼Example的hit較低的原因; 反之，實作一因為有2 ways的Cache，所以hit機率比較高。

* 實作二波形圖



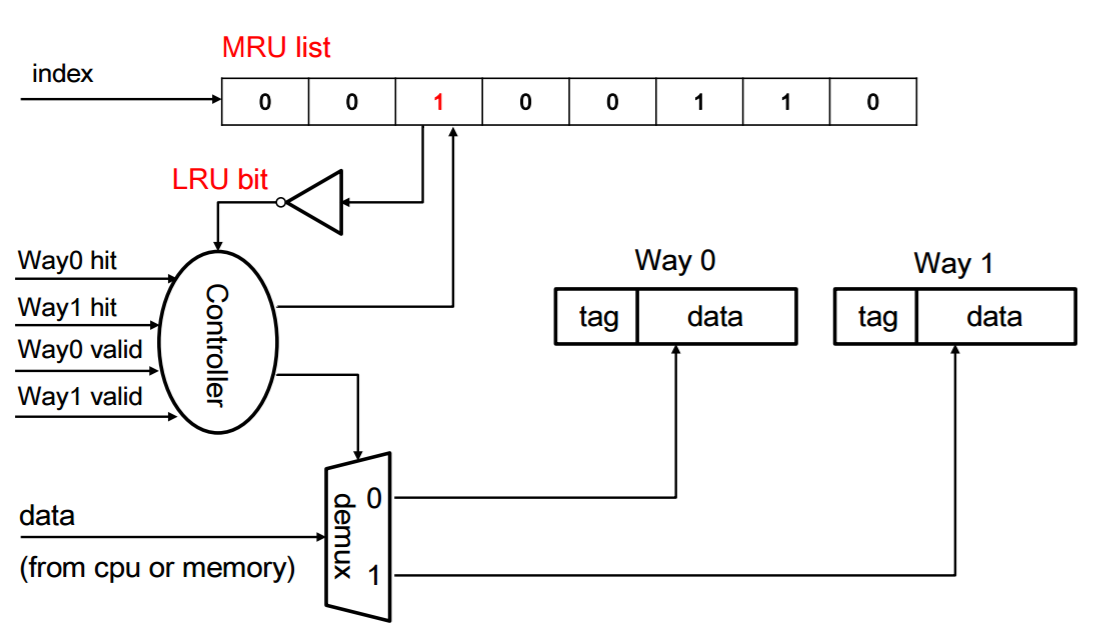


* 實作二分析

1. 第一張圖的黃色框框中，可以看到way2發生hit ( hit2 = 1 )，所以tag\_out的值是tag\_out2的值。
2. 第一張圖的兩個紅色箭頭，當numout = 1 ( 決定way2離開 ) 時，enable2 = 1 ( way2 要寫入 ) 。
3. 第二張圖的黃色框框中，可以看到way1發生hit ( hit1 = 1 )，所以tag\_out的值是tag\_out1的值。
4. 第二張圖的兩個紅色箭頭，當numout = 0 ( 決定way1離開 ) 時，enable1 = 1 ( way1 要寫入 ) 。
5. Hit rate = 44/59 = 0.7457

* 實作一V.S.實作二實施方法不同之處

實作一的replacement是使用random的方法，numout去決定要寫入哪一個way，way1有資料就寫入way2，反之亦然，但若兩個都有資料，就輪流寫入。

實作二則是使用LRU的方法，有一行MRU-list，若他的index為0，代表最近用到way1，因此要把way2換掉，反之亦然，因為他是2 way，因此不是0就是1所以就這樣一值判斷到他結束。

## 心得

* 實驗心得:

這次實驗是上禮拜實驗的延伸，再加上這禮拜的youtube影片就是教本次實驗課的相關內容，所以在實作起來時相對比較順利一點，而在作完以後，對於replace的方法以及cache存在的功用又更加熟悉了。不過在作實驗時，發現實作一和實作二的hit rate一模一樣實真的太驚訝了，剛開始還以為自己做錯，後還再三檢查以後才發現真的是一模一樣，真的是一件很巧的事!

* 對本次Lab的改進建議:

這次的lab是本學期的最後一次lab了，後面的幾次實驗都沒有一開始實的緊湊，不過一樣有學到很多東西，還是很謝謝老師和助教給予我們的幫助，雖然不確定未來會不會在工作或是生活上再次接觸到這門課學習的東西或是需要實作，但是我想這門課學到的知識和能力應該是不太容易忘了。