**Final project report 100070023 資工系 楊雅琪**

這次實作並沒有做出來，想法大概是這樣：

1. 讀cache檔案存成字串，將字串各自存為bus,entries,associative.
2. 算出需要用到的k bits;(k=log entries/log2);
3. 由中間的bit開始向左右兩邊延伸顯出k個bit
4. 把.lst檔案中的地址寫進地址陣列內
5. 用第三步驟所選的k個bits找到對應的entries index[][](二維陣列[m][n],n即為associative的大小)
6. 找到對應的index後開始從第一個開始找(從index[][0]找到index[][n])，如果裡面內容是空的，就直接mapping上去並讓miss+1;並輸出此address miss到index.rpt;
7. 如果全部都滿了也沒有相符合的，就取代第一個(index[][0])，miss+1並輸出address miss到index.rpt；
8. 如果有找到相同的data就輸出此address hit到index.rpt中;
9. 如果剛剛.lst的檔案還沒讀完就跳回4,讓下一行的地址寫進地址陣列，讀完了就跳10
10. 將cache miss的總數輸出到index.rpt中
11. 如果還有cache.org未讀就跳第一步在讀下一個cache，如果都讀完就結束了

這次很可惜其實有大約想出來要怎麼實作，但自己時間分配的不好以致於沒有把project作完很難過…☹

只能盡量把想到的東西寫在report中，希望能讓助教 教授看到一點點點誠意QQ

想改進的地方就是先把這次作業寫成功QQQQQ