**DSD\_Report\_HW3**

R13921072 何家祥

1. **Cycle time use in cache\_syn.sdc and tb\_cache.v**

由於這次的實驗沒有要進行AT ranking，所以cycle time設為10有過之後就沒再另外測試其他，因此合成用的設定就是10.0 ns。

1. **General specification of the cache unit**

**dm\_cache**

Size : 32words，依照規定含有8blocks，且每個block能夠存放4words

Placement policy: Direct-Mapped，也就是當快取沒有所需資料時，直

接將整組block替換成記憶體中的內容。

**2way\_cache**

Size : 32words，依照規拆成4個sets、每個set有2條way能夠各存放

4個words

Placement policy: 2-way set associative，也就是當快取沒有所需資料

時，在同一個index下，可以存放兩組數據，不需要

每次都把所有資料洗掉。

1. **Read/Write policy**

兩種方法，dm\_cache、2way-cache，都是使用read allocate的策略，當CPU要讀取資料但是miss的時候，會先將記憶體資料搬運到快取中，並標示valid，才再把資料給CPU作運用。

至於書寫策略都是用write back + write allocate，也就是說當快取裡的資料要被改變時，會只改快取中的資料並標註dirty，並不會浪費時間直接寫回記憶體，而是等到之後用不到，要被更換時，才順便把書寫的資料改道記憶體裡頭。還有稍微不一樣的是，2way cache在替換的策略上，會用LRU進行分析，會選擇這個set中比較不常被用到的拿去做資料更換。

1. **Design architecture and FSM**

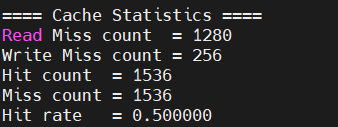
整體架構跟講義的方法是差不多的，依據題目所定義的IO port，來設計兩種cache，並且使用wrtie back的方式寫回記憶體單元，並在下方解釋所使用的FSM設計。

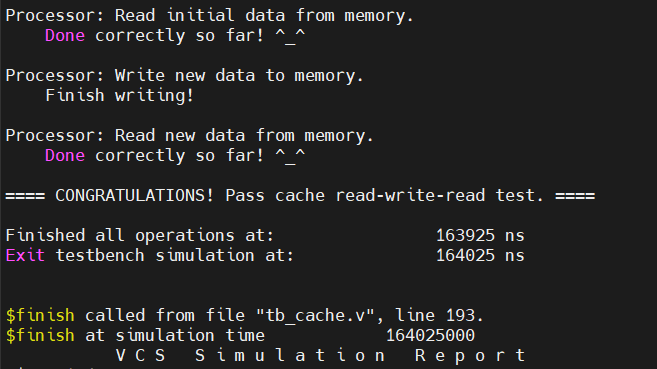
**dm\_cache**

使用四種狀態來設計，IDLE、CHECK、WRITE、READ、UPDATE

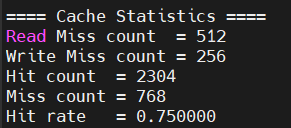
1. IDLE : 初始或閒置狀態，等待CPU發送read/write指令，當訊號來時代表要做事，因此切換到check。
2. CHECK :在此判是否hit，如果有就直接對快取進行資料處理，若是沒有就會根據dirty bit決定要進行WRITE還是READ。
3. WRITE：若這個block是dirty的，就會要先將快取資料放回記憶體，也就是寫入記憶體完成後才讓CPU讀取，也就是回到READ。
4. READ : 讀取記憶以內的資料進入快取中，也就是進行read allocate
5. UPDATE：
6. **Performance**

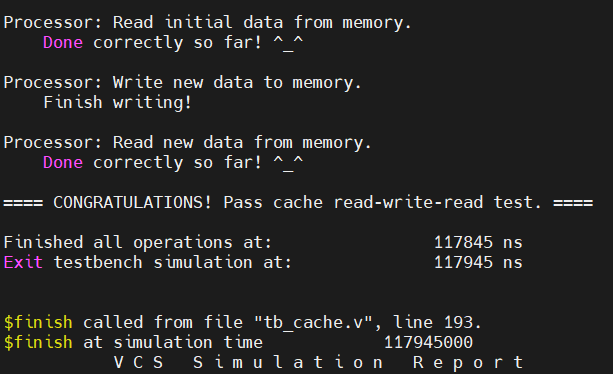
**dm\_cache**





**2way\_cache**





由於2way的方法我有使用ＬＲＵ進行挑選替換的資料內容，因此從結果來看，可以看出更有效率，能夠有更高的命中率以及較低的miss機率，也因此能用更少的時間完成整個過程。