

Computer Architecture 2018 – Final Project 2

Team：地表計結最強的那組喔！懂？

1. Members and Team Work

Member			Team Work	
資工三	B05902023	李澤諺	50%	1. 修改 Project 1 的 pipelined CPU 以及助教提供的 dcache_top.v，並將兩者組合在一起。 2. Debug。
資工三	B05902121	黃冠博	50%	

2. Implementation

- (1) 將 Project 1 中的 PC.v、Registers.v、Instruction_Memory.v、Data_Memory.v、TestBench.v 更換為助教提供的 module。
- (2) 將 Project 1 中的 pipeline register 皆加上 stall 的 input port(當 cache miss 時，dcache_top 會輸出 stall 的 signal 給 PC 和 pipeline register，如此便可以 stall 整個 CPU)。
- (3) 將助教提供的 dcache_top.v、dcache_tag_sram.v、dcache_data_sram.v 接上。
- (4) 修改 dcache_top.v(實作如下)。
 - (i) 若 $p1_tag == sram_tag \ \&\& \ sram_valid$ 成立，即 address 中的 tag 和要 access 的 cache entry 所存的 tag 相同，且該 cache entry 為 valid，則代表 hit，否則為 miss。
 - (ii) 若 cache hit，則代表 cache entry 裡存有我們要 access 的 data，而若 cache miss，則必須將要 access 的 data block 自 data memory 帶到 cache 中，此時再 access cache entry 便會 hit，因此不論如何， r_hit_data 的值 assign 為 $sram_cache_data$ ，即要 access 的 cache entry 所存的值即可。
 - (iii) Read data 和 write data 中，必須找到要 access 的 data 在 block 中的哪一個位置，即 $r_hit_data[(p1_offset \gg 2) * 32]$ 開始的 32 個 bits(因為每次讀寫的 data 為 32 bits，即 4 bytes，為了 alignment，因此先將 $p1_offset$ 向右 shift 2 位再乘以 4，如此一來 byte offset 必為 4 的倍數，再乘以 8 後即為 bit offset，由此開始的 32 bits，即為要讀寫的 data)。
 - (iv) STATE_MISS 中，由於 cache miss 一定要 access data memory，因此將 mem_enable 設為 1'b1，若要 access 的 cache entry 為 dirty，則必須將該 entry 中所存有的 data 存回 data memory 中，即 write back，因此將 mem_write 和 $write_back$ 皆設為 1'b1，並進入 STATE_WRITEBACK，而若不為 dirty，便不需要

write back，因此將 mem_write 和 write_back 皆設為 1'b0，並直接進入 STATE_READMISS，將要 access 的 data block 自 data memory allocate 到 cache 中。

- (v) STATE_READMISS 中，若 mem_ack_i 為 1'b1，即 data memory 已經 ready，將 data 傳到 cache 中了，由於此時不用再 access data memory 而是要 access cache，將 data 存入要 access 的 cache entry 中，因此要將 mem_enable 和 cache_we 分別設為 1'b0 和 1'b1，並進入 STATE_READMISSOK，而若 mem_ack_i 為 1'b0，代表 data memory 還沒 ready，將需要的 data 傳回 cache，則繼續保持 STATE_READMISS。
- (vi) STATE_READMISSOK 中，代表需要的 data 已經存入 cache 中了，因此將 cache_we 設為 1'b0，並回到 STATE_IDLE。
- (vii) STATE_WRITEBACK 中，若 mem_ack_i 為 1'b1，代表 dirty data 已經 write back 回 data memory 中，可以將要 access 的 data 自 data memory allocate 到 cache 中了，因此將 mem_write 和 write_back 都設回 1'b0，並進入 STATE_READMISS，而若 mem_ack_i 為 1'b0，代表 dirty data 還沒存回 data memory 中，則繼續保持 STATE_WRITEBACK。

3. Problems and solutions

本次 project 其實並沒有遇到太困難的問題，比較大的問題應該是在於要理解 dcache_top.v 裡的 code 在做甚麼事情，並將其修改出來，理解之後就沒有其它太大的困難了。而在 debug 時，唯一的 bug 就是發現 Data_Memory 的 state 沒有初始化，後來將第 42 行改為 always @(posedge clk_i or negedge rst_i)後，就可以將 Data_Memory 的 state 初始為 STATE_IDLE，並順利執行了。