report

小組名稱：期末hazard

組員：B07902007伍柏豪、B07902015江律旻、B07902113陳品鈞

Modules Explanation:

dcache\_sram:

Input: clk\_i, rst\_i,addr\_i, tag\_i, data\_i, enable\_i, write\_i,

Output: tag\_o, data\_o, hit\_o

Memory:

[24:0 ] tag [0:15][0:1]

[255:0] data[0:15][0:1]

[1:0] lru [0:15] (lru為 least recently used 如lru[8] = 2 代表第2個 block是最舊的)

write: 如果write且enable，則比對在addr\_i的兩個tag和tag\_i有沒有相同，相同代表write hit，直接覆寫該block並回傳被覆寫的block，更新lru。若miss，則依據lru決定哪一個block要被覆寫，同樣回傳被覆寫的block並更新lru。無論hit/miss被寫的block dirty bit 都設為1。

read: 比對在addr\_i的兩個tag和tag\_i有沒有相同，相同且valid bit 為1代表read hit，直接回傳該block並更新lru，並且設hit\_o為1；若無相同則為miss，設hit\_o為0，依據lru決定哪一個block要allocate出來，回傳該block。

dcache\_contoller:

input:clock signals, reset bit, data and ack from memory, data, address, read bit, write bit from cpu

output:data, address, enable bit, write bit to data memory, stall bit, data to cpu

idle state:等到有新的request且資料沒有hit時跳到miss state

miss state:若dirty bit on則mem\_write, mem\_enable, write\_back on並跳到writeback，其他狀況則mem\_enable on, write\_back, mem\_write off並跳到readmiss

readmiss state:等待data memory傳入ack bit，cache\_write bit on, mem\_enable off跳到readmissok

readmissok state:cache\_write off跳到idle state

writeback state:等待data memory傳入ack bit，mem\_write, cache\_write off跳到readmiss state

pipeline registers(IF/ID、ID/EX….):每過一個cycle就將從上一個state讀入的資料傳到下一個state，但若收到來自dcashe\_controller的stall訊號則不動。

testbench:和project1一樣修改pipeline registers的initialization部分。

CPU:將stall線接上各個pipeline registers以及PC，加入dcache\_contoller取代data\_memory，data\_memory的部分改為宣告在testbench裡。

Members and Teamwork:

江律旻:dcache\_controller

陳品鈞: dcache\_sram ，debug

伍柏豪: 整合前部分接線及其他register修改，debug

report 分別撰寫負責部分

Difficulties and Solutions:

debug很困難，資料出錯可能是cache沒處理好可能是從data memory讀錯，看波形圖很久才de完。

Development Environment:

Ubuntu16.04 on Windows， 但在工作站也測過

compiler: iverilog