

Computer Architecture 2018 Fall Project 2 Report

Team's Name

李胖一級胖

Members

B05902017 王盛立

- 工作比例: 33.333%
- 工作內容: 參與討論、接線、Debug，程式撰寫，整理Report

B05902033 高晟瑋

- 工作比例: 33.333%
- 工作內容: 參與討論、接線、Debug，程式撰寫，整理Report

B05902105 余友竹

- 工作比例: 33.333%
- 工作內容: 參與討論、接線、Debug，程式撰寫，整理Report

Pipelined CPU Implementation

我們參考了上課投影片以及作業指示，照著Data Path & Module的圖，一步一步串連CPU的接線

大體上的實作方向分成兩個部分：

1. Data Memory轉換成Data Cache

- 將原本data memory的module改成data cache，並把對應的線接好，原本的IF/ID ID/EX EX/MEM MEM/WB四個stage和pc都接上stall signal，用來處理當cache miss時從data memory拿資料那一段時間的stall
- 實作data cache這個module

2. 處理PJ1的小bug

- 原本的pj1太複雜了因此有些小狀況沒有完全的正確

Updated Modules

CPU

- 控制Modules輸入與輸出所對應的其他Modules

Dcache_top

- 主要負責處理data cache取代原來的data memory

Dcache_data_sram

- 把cache中的data輸出給memory

Dcache_tag_sram

- 取得cache中關於tag, valid bit, dirty bit等等的資料

Data_memory

- 根據enable跟write的信号決定是否能夠讀寫memory，在資料準備完成後由ack通知cache

Instruction_Memory

- 跟上次作業差不多，照著助教給的code沒動

PC

- 跟上次作業一樣，只是從data cache如果有傳stall signal的話就要stall

Registers

- 跟上次作業一樣，主要是從Register的addr.中讀出對應的Data，若有必須寫入的部分(寫入Register)，同樣也在這個Module處理。

Data Cache Detail

- 首先加上判斷什麼是hit，如果在cache中的資料現在是valid以及傳入的信号有要read或write就是hit
- 之後處理data格式的問題，在32bit跟256bit間去做轉換
- 最後照著每一個state的狀態去添加相對應的訊號讓他能做那個state該做的事
- 首先如果有request進來，而沒有hit的話則我們進入STATE_MISS，反之如果hit就持續STATE_IDLE
- 進入STATE_MISS後，如果cache中的資料是dirty，也就是被更動過的話，則我們把mem_enable, mem_write和write_back這幾個signal都設成 1，其中enable和write是直接連到memory的信号，

通知memory我們要做相對應的動作。而我們便進入STATE_WRITEBACK的狀態。相反如果資料不是dirty，我們則把enable設1，write和write_back設為0，因為我們沒有要寫，進入STATE_READMISS。

- 在STATE_WRITEBACK中，我們等待一個memory通知說他好了的signal，當他好了之後就跟上面一樣write和write_back設為0，進入STATE_READMISS。
- 在STATE_READMISS中，同樣我們等待memory通知說他好了，好了之後我們就把enable設為0，cache_we設為1，讓資料能夠正確的被載進來。並進入最後的STATE_READMISSOK。
- 最後我們把所有的signal都回復成0，避免他做不必要的事。

Implementation Difficulty

- 由於這次的要印的cycle數很多，所以執行完後要去找想看的cycle發生什麼錯誤，要往上滑很久
- 跟上次作業一樣，印出每個module的參數後會發現要印的東西太多，好不容易找到出錯的cycle還要看是哪邊出錯也很考驗眼力
- 助教給的data_memory.v，在always中加了negedge rst_i才跑得動
- pj1的小bug，例如在判斷原本的指令分別是什麼時，應該統一先從opcode判斷起，另外當初在傳遞control signal時因為剛開始不是很了解，我們是到了相對應的stage才做判斷，在ex/mem我們犯了一點判斷上的錯誤。
- Cache miss時，我們stall的cycle會比標準慢兩個