

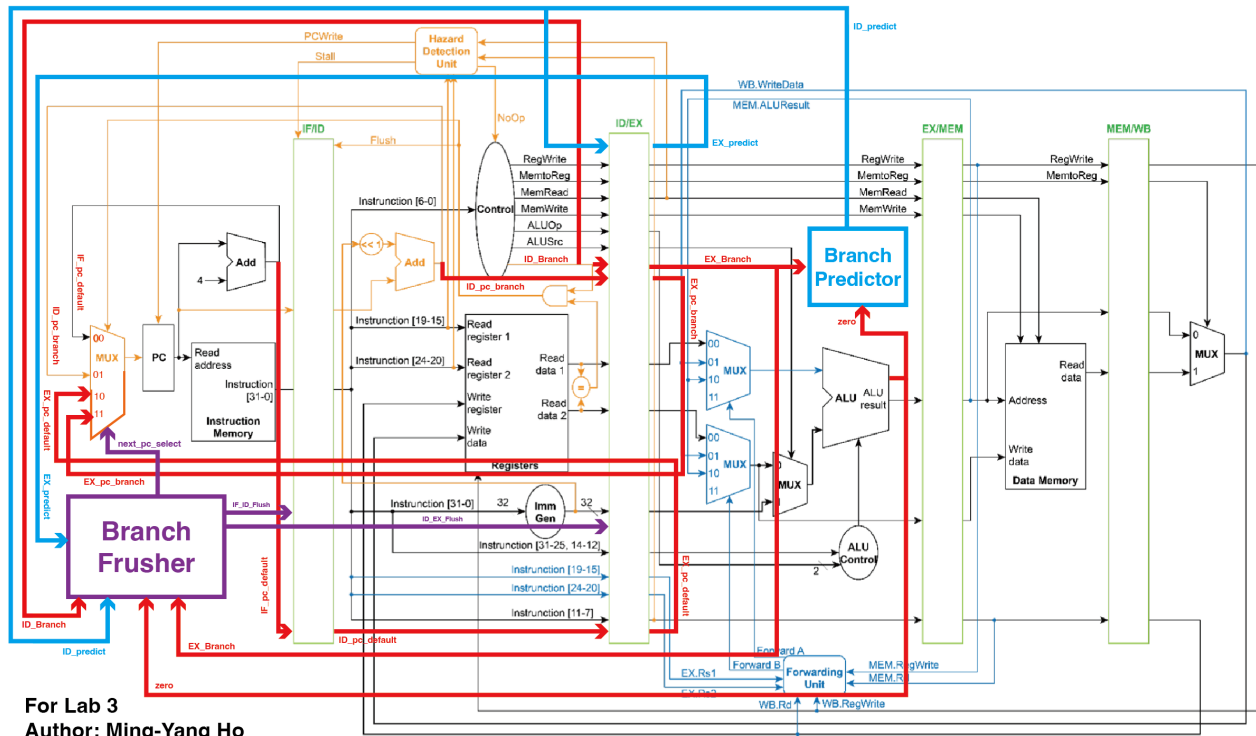
Computer Architecture Lab3 Report

Name: 何明洋

Student ID: r11922208

Designed Datapath

主要增加了 Branch Predictor 與 Branch Flusher 來處理prediction與flush等流程。



Modules Explanation

- Note: 以下針對增加或更動(from Lab 2)的modules進行解釋

New Modules

Branch Predictor

根據在EX stage的 Branch (是否是一個branch指令)，以及 zero (是否taken)，來決定要predict什麼。並且裡面還有一個2-bit state，一開始的值是 11 (STRONGLY_TAKEN)，若EX目前是 Branch，則會受到 zero 改變其中的state，改變後的狀態包含 10 (WEAKLY_TAKEN)，01 (WEAKLY_NON_TAKEN)，00 (STRONGLY_NON_TAKEN)，改變的方式如同題目給定的說明。而prediction (給ID stage) 的結果即是最高位：1 表示 taken，0 表示non-taken。

Branch Flusher

這個module會根據predict與真實的結果來給flush訊號，並且傳出一個selector來選pc。

- 先針對EX stage，input EX_Branch (在EX是否是branch)、EX_predict (在上一步ID傳到EX stage的預測值)、EX_zero (正確答案)，來做對應處理。假設是 EX_Branch 且 EX_predict != EX_zero，代表預測錯誤，IF_ID 與 ID_EX 都要flush，而要置換的pc則根據：
 - EX_predict 表示predict taken但預測錯誤，pc換成EX_pc_default(一路傳過來的pc + 4)
 - !EX_predict 表示predict non-taken但預測錯誤，EX_pc_branch(在ID stage算出的位置)
- 若無以上問題，那就確認在ID stage中是否是branch by ID_Branch，並且若branch predictor也傳來的 ID_predict 也預測要taken，那 IF_ID flush，並也把pc換成 ID_pc_branch，亦即在ID stage算出的位置。
- 若以上都不是，那就是原封不動地使用default pc，也不用flush

MUX_PCSrc

是一個 MUX32_Double，由上述 Branch Flusher 給的選擇訊號 next_pc_select 來選擇到底pc要使用 IF_pc_default，ID_pc_branch，EX_pc_default，EX_pc_branch。

Modified Modules

ALU

多回傳 Zero_o，即src1 - src2是否為0。

Pipeline_IF_ID

增加了傳遞 pc_default 的功能。

Pipeline_ID_EX

增加了傳遞 pc_default，pc_branch，predict，branch 的功能，也能透過 flush 訊號flush。

CPU

把上述的新modules接上。

No modification

Adder，ALU_Control，Control，Forwarder，Hazard_Detection，Imm_Gen，MUX32_Double，MUX32，Pipeline_EX_MEM，Pipeline_MEM_WB，and Sign_Extend

Difficulties Encountered and Solutions in This Lab

- 花了非常多時間才把Datapath畫出來，原本以為2-bit branch predictor就是改變state而已沒什麼，上課講義也畫得非常簡單，但因為predict錯誤的影響深遠，得到處接線並縝密思考到底下一步要做什麼事。
- 原本用ipad畫datapath，但要打字很麻煩換用Microsoft PowerPoint，但線太混亂難以處理，最後不得已用了Adobe Illustrator。
- 比起Lab2是需要實作大量新module的體力活，Lab3是簡單但混亂考驗邏輯與專注度的作業。

Development Environment

這個作業是直接 Ubuntu 20.04 環境中起 Ubuntu 22.04 Docker container(本題提供的 dockerfile)開發與測試。