

CA2022 Lab2 Report

• 2.1

- 以下簡單介紹一下與lab1不同的部分。
- **ALU**
 - ALU新增了zero，只有1 bit方便使用，如果是ALU Result是0的話zero = 1，反之為 0。
- **branch_predictor**
 - 每一個cycle，如果在EXE stage的instruction是beq，就看ALU的Result，來更新state以及predict。
- **IF_ID**
 - 需要flush的時機為以下兩種。
 - 在ID stage，如果為beq，且predict taken。
 - 在EXE stage為beq，且predict錯誤。
- **ID_EX**
 - 新增了以下的reg
 - flush
 - predict錯了的話，flush = 1，讓下一個cycle的registers全變0。
 - Branch
 - 在EXE stage 知道是否為branch instruction
 - pc_next、Branch_address
 - 儲存了beq的下一個指令，如果在EXE stage知道predict錯了的話，可以restore回去。
 - prev_predict
 - 紀錄了上一次的predict值，如果在EXE stage發現predict錯了的話，可以flush IF、ID stage。
- **CPU**
 - 新增了一個pc的multiplexer，在predict錯誤時，需要restore回去，如果當初是taken，則restore pc_next，當初是nontaken的話，則restore branch_address。

• 2.2

- 最一開始因為沒有電路圖加上題目敘述過於簡略，因此想了蠻久，然後忘了 predict taken 要 flush，branch_address 有時候不知道為什麼跑不出來，大概花了兩天做出一坨垃圾，最後全部重寫一遍，減少一些 module，直接在 cpu 的 module 裡面 assign 一些值，再把需要的值，像是 pc_next、branch_address 直接存起來，方便使用。

• 2.3

- OS: MacOS 13.0 ventura on M1 chip
- IDE: vscode
- compiler: Iverilog