計算機組織 期末專題報告

Pipelined MIPS CPU Simulator

A1105505 林彧頎

A1105524 吳雨宣

A1105545 潘妤揚

A1105549 杜佩真

一、工作分配

|  |  |
| --- | --- |
| A1105505 林彧頎 | 1.(Forwarding、stall、predict not taken)流程設計與除錯  2.MEM、WB  3.程式merge  4.github撰寫  5.報告撰寫  6. 程式註解 |
| A1105524 吳雨宣 | 1.ID  2.(Forwarding、stall、predict not taken)除錯  3. 程式註解 |
| A1105545 潘妤揚 | 1.EX  2.(Forwarding、stall、predict not taken)除錯  3. 程式註解 |
| A1105549 杜佩真 | 1.IF  2.(Forwarding、stall、predict not taken)除錯  3. 程式註解 |

二、問題與作法

1.架構設計:

我們一開始想了兩種方法來寫這個專題一個是暴力解，一個是按照PPT中會將OPCODE依照EX、MEM、WB依序傳入的格式做，因為此次專題中的結果需要輸出對應的opcode，因此我們大概是老師一開始公布，解答都還是錯的時候就開始做，我們分成兩人一組先試著暴力解(林彧頎、吳雨宣)，一組試著按照硬體邏輯做(杜佩真、潘妤揚)，而基本流程都是我先設計出來，然後再跟組員討論，但雙方在做到一半的時候都有發現問題，像是暴力解，容易有不知道該如何pipeline操作的問題，而硬體結構，則是要模擬的話，會有些困難，因為硬體架構中的指令其實比老師要求的還多，因此最終我們決定將兩種方法結合，我們透過將stage分開，並由後往前做(避免值被覆蓋)，並使用一個全域變數的class使可以共享狀態，避免忘記傳的情況，而且先定義好變數，規定好每個stage該做的事，像IF(抓指令)，ID(解析指令與分配值和一些forwarding等等…的判斷)、EX(運算和一些forwarding等等…的判斷)、MEM(主要進行MEMWrite、MEMRead一些forwarding等等…的判斷)、WB(判斷是否write back)。

2.分工方法與後續除錯:

分工的方法上面期時有提到，指示這裡最重要的是，我們其實是前期先各自做好各自應該要達成的架構，然後用一些指令和先自己存一些值的方式，來判斷自己的stage是否是對的，然而這樣還是會出錯，因此我們後期是一起將每個stage執行出來的結果做檢查，一起將forwarding、stall、predict not taken修好。

3.遇到的問題:

1. forwarding、stall、predict not taken:

這絕對是最主要的問題，因為我們即使一開始都把架構和ppt讀熟，甚至我為了讓大家更懂ppt，我還重新教ppt讓大家可以一次搞懂，但我們還是都有遺漏，像是我那時候一值認為predict not taken的branch判斷跳轉的時機是EX階段，但後來看PPT並且有組員的提醒我才發覺是要在ID階段就要判斷，或者是在forwarding中，有許多種的地方要判斷，時常會不小心拿錯變數，或改著改著就錯亂了，而且我們的程是碼是部分暴力解的，因此需要不斷檢查每個stage少了什麼，然後回去檢查，我們就像這樣不斷除錯，才解決全部的forwarding、stall、predict not taken。

2.分工:

我覺得這個專題就跟之前的組合語言很像，基本上很難分工，因為每個stage之間都息息相關，像我本來想說在暴力解中，直接將MEM印出來，卻沒想到MEM階段其實也有需要幫忙檢查的forwarding，這是我們組員之間互相檢查出來的結果

3.指令格式的問題:

其實我很猶豫要不要寫這個，因為老師題目裡基本上有一個規則存在，所以有些地方可以寫死的(判斷指令或抓數字)，但實際上的指令可能有常數、暫存器、Memory，我們一直在想辦法，要好好抓到準確的值，並且可以準確的判斷forwarding，因此我們後期製作的時候，其實就是在嘗試，老師題目沒給的可能性，所以過程中，反而除錯更多!

4.總結:

這個專題我覺得一個人很難做出來，因為不是每個人都能想的那麼周全，因此需要大家一起討論、檢查、分析，才可以有更好的作為，我們這組比較可惜的是，大家基本上都不會用github，因此我們是實體討論，比較需要大家的配合，整體來說學到很多，也更加認識老師教的MIPS架構與除錯訓練!

(下一頁心得)

三、心得

A1105505 林彧頎

我負責forwarding、predict not taken的除錯，更主要撰寫MEM、WB，並將程式註解，然後傳github，寫報告與統整，看起來很多，但過程中組員幫忙也不少，我們的工作可以說是很平均的，因為大家一起互相合作討論，大家都會趁有空時就幫忙檢查code，然後跟大家討論，我覺得我在這次專題中學到許多我上課忽略的地方，尤其最印象深刻的是，我一直很執著所有運算都要在EX結束，但潘妤揚因為有讀到，因此翻給我看，結果把我們整個程式中，最後一個branch forwarding很奇怪的地方給解決了，這讓我學到團隊合作是多麼重要，也更讓我知道forwarding、branch的流程，謝謝組員，最後做出成果絕對是大家的功勞，十分開心最後能做出來。

A1105524 吳雨宣

起初在上課時，就覺得第四章這部份真的有點複雜，後來把PPT讀完之後，就有比較懂期末專題的題目要求，雖然看懂專題題目的要求了，在考卷上也能寫出推導過程，但距離用程式寫出這個專題似乎還有一段距離，為了實作這個專題，我們反覆翻了好幾次上課的PPT。在五個階段當中，最為複雜的應該就是ID了，在ID的部分就會針對stall跟forwarding做判斷等等，雖然在寫程式的過程中遇到蠻多困難點的，但最後都順利解決了，看到了完成的成果，真的很開心。

A1105545 潘妤揚

在這次的專題當中，我負責的部分是EX、forwarding。我覺得主要難的地方在於把前一個指令或前前一個指令直接抓至EX當中，因為在每一個cycle之後那個值會直接被蓋掉，所以我們有想一個辦法，就是在ID先把東西存起來，再往下丟，丟到MEM，之後再拿那個值，這樣就可以完成forwarding了。透過這次的專題我也重新把MIPS的運作都學了一遍，雖然程式碼可能還有一些可以改進的地方，但因為這次的專題，我覺得我除錯的能力有因此提升。

A1105549 杜佩真

這次專題中，我負責的是IF部分的撰寫和最後stall和forwarding的除錯，藉由這次的專題，讓我更了解整個模擬器的運作，雖然還是有一些地方是因為看結果發現哪裡可能需要stall在停一回合，就在那個部分額外加上去，或是需要beq停在ID多久而寫了額外的變數判斷，感覺這些部分是之後可以改進的地方，讓整個邏輯再更順暢一點。