

計組期末專題報告

MIPS CPU Pipeline

with Forwarding

第九組

A1095509 李品妤(Forwarding)

A1095514 朱祐誼(Forwarding)

A1095550 莊郁誼(non-Forwarding)

A1095551 廖怡誠(non-Forwarding)

一、 工作分配

姓名	工作內容
李品妤	架構設計、程式碼實作、readme撰寫
朱祐誼	架構設計、程式碼實作、程式註解撰寫、報告撰寫

二、 遇到的問題

● 架構設計

我們這次從一開始就是想利用模擬實際硬體架構的方式去寫，所以程式架構就變得十分重要，每個stage該做什麼要存什麼，哪些stage需要用到另外stage的資料等等都要和實際的一樣，還有因為實際上的電路是同時執行的，而用程式模擬只有一個thread所以很像沒有pipeline的狀況，所以就要想清楚如何去模擬出pipeline，這些東西一開始規劃沒想好就會造成後續不斷的東補西補，進而造成越來越難寫。

所以我們其實有寫了兩種架構，然後大概寫到一個段落的時候比較一下，選出了比較好的架構繼續寫下去，雖然說是比較好的架構，但是其實寫的途中還是有發現一開始沒想到的地方，像是我們把branch放在最後寫，結果完全忘了branch也有hazard的問題，還好架構的部分有寫好，才沒有需要大改程式碼。

● 程式分工

因為這份程式碼相依性很高，也就是要拆成不同的工作比較難，因為各自的工作都會影響到其他人的部分，所以變成很難去分工，這也是前面有提到的架構很重要，如果架構可以很早定出來然後都不會去更動，其實可以解決部分的分工問題，但我們不太可能做到這樣，所以我們才預先分成Forwarding跟n

on-Forwarding，雖然可以減少分工的問題，但多少還是會造成同一時間只有一個人能做事的情況，在製作效率上比較差，所以其實我們也有想規劃做一個自動檢查答案的腳本或程式，不過礙於事情太多，做了一些就沒有繼續做下去了。

● 實作問題

前面有說到我們照著實際硬體的架構去做的，但其實有部分還是沒有像實際的一樣，像是每個pipeline register存的東西其實不太一樣，但實作上我們是用一個struct，所以每個pipeline register存的東西都是一樣的，另外我們也礙於程式模擬，所以還另外加了一個pipeline register，方便我們去實作。

另外遇到最多的就是執行順序的問題，因為我們實際上是由後往前執行，也就是WB先IF最後，所以像是在寫EX stage時就要考慮到mem、wb其實已經執行完，到下一個pipeline register了，我們常常被這個執行順序給搞混，像是在凍結或更改PC時，常常搞混是前一個還是現在這個。

最後一個是輸出格式，因為我們覺得教授PPT上輸出的東西不夠多，為了要方便我們檢查及Debug，所以我們有自訂了輸出格式，但跟PPT上的有個很大的不同，我們的Stage輸出的是執行結束後，但PPT的是執行之前，所以造成我們在跟同組non-Forwarding及PPT上的比較就有差異，所以最後還是統一換成執行之前，不過因為又跟原本寫的不一樣，所以在檢查的時候三不五時就會錯亂。

三、心得

這次我們是做有Forwarding的，但我覺得應該沒有比non-Forwarding的難多少，比較有挑戰性的應該是想要依照硬體上的架構去寫，為了達成這個我們製作的時候不知道翻了幾次の上課的PPT，最後寫出來的成果也讓我們蠻開心的，不過我們還有看到之前有學長用verilog寫這個專題，真的是大吃一驚。

另外這次專題也讓我更加了解程式架構的重要性，好的架構不僅實作起來更簡單，在出問題時也比較好去Debug，甚至未來自己或是其他人在看的時候也更容易去了解，這次也是第一次寫了一份完整的readme跟整份專題的註解，實在是另一個大挑戰，因為其實已經距離寫完有一點時間了，不過也跟前面架構有關，讓我們在回想的時候可以更快的想起來，最後要寫一個讓人快速上手的readme比我想像中的還困難，在自己的環境下去測都不會有問題，但要給任何人去使用就要考量到OS、Liberty、編譯、輸出輸入等等的，甚至還要介紹程式的架構邏輯等等的，不過也因為把readme寫好才能讓不清楚的人看到這個專題就可以快速理解，所以我們也會考慮把這學期、甚至是之前的專題補上完整的readme。