計算機組織期末專題報告

第九組

成員

A1095509 李品妤

A1095514 朱祐誼

A1095550 莊郁誼

A1095551 廖怡誠

1、遇到問題&解決

◆ 指令讀取處理

問題:

正常處理流程，傳入MIPS指令後需要先將指令組譯成機械碼，再放入pipeline中開始循環，但因為我們在轉換成機械碼後會是用字串儲存，反而在ID階段要獲取暫存器值時，會因為要將二進位字串轉回整數，變得更難處理。

解決:

最後，我們放棄原本的方式，不把傳入的指令轉換成二進制，直接將指令做字串切割，分出要獲取的數值，就不需要再做轉換的動作。

◆ Data Hazard

問題1:

在課堂的內容中，只有提及在有forwarding的情況下，EX hazard和MEM hazard是如何判斷及處理，但這次的專題是實做沒有forwarding的版本，所以當初對於要放在哪個階段和怎麼設計猶豫了很久。

解決:

最後，我們決定參考forwarding電路圖的架構，將判斷機制放在ID階段，與原本的差異在於在判別後，我們不會提前把值傳回去做更新，只會再出現情況後，將傳入EX的值改為null，模擬stall。

問題2:

使用軟體模擬硬體環境最大的痛點是無法模擬同步執行，這導致判別data hazard的操作沒辦法一起放在ID中，因為程式的流程是從後往前執行，代表在ID階段獲取EX和MEM階段的值，實際上已經是下一個cycle的值，會造成判斷錯誤，如果要從前往後執行的話，又會導致前面更新的值蓋過舊的值，要更改架構的話，也只會變得更複雜。

解決:

因此，最後我們決定這個部份就不按照硬體設計，等全部的階段執行完，在一次做判斷，決定下一個cycle是否需要stall。

◆ Beq的判斷

問題:

Beq原本是在ID中做判斷的，但會因為執行先後順序的影響，破壞到指令運行的步驟。

解決:

因此，我們把它移到循環的地方做判斷，因為那邊可以直接處理指令運行的順序。

◆ lw & sw Offset

在寫的時候，指令lw, sw的offset我們是直接讀取存入，但取得register或memory值之前，需要先除以四，位置才會對，因

為單位是word。網路上查了很多資料，有些沒有除，有些有除，所以當時我們花了許多時間討論。

2、分工

因為這次的專題程式碼為一體成型，所以我們工作分配的方式為輪流寫，在討論完整體架構後，我們將工作分為前半部，架構撰寫包含pipelined的五個階段，後半部為beq和data hazard的判斷及彙整程式碼。會這樣分配的原因是從頭建立架構比較麻煩，所以將較麻煩的stall處理拆出來，順便進行程式碼的檢查並彙整。

其餘的報告、Readme、makefile均為兩人合力討論寫出來的內容。

莊郁誼	程式架構設計、撰寫、寫報告、Readme、makefile
廖怡誠	程式架構設計、撰寫、寫報告、Readme、makefile

3、心得

莊郁誼:

一開始在寫的時候不知道要從哪裡下手，所以就先把他全部都寫在主程式，之後再分成不同階段的class。而程式在進入每個階段後可以得到不同的值，而這個就要花很多時間去研究、搞懂其運行的原理，再把他寫成程式，是一項挑戰。但經過這次專題我也更理解什麼時候需要加stall，還有每個階段的功用，會產生哪些值，什麼時後會影響暫存器、記憶體...，還有signal的運作！總覺得應該要把每個階段所有的元件都寫成一個物件來處理，但這樣好像又太大費周章，所以最後沒有這樣處理。在處理傳送到每個階段的時候，我覺得

特別酷，因為不像是以我們學到的去讓他照順序處理下來，而是先從後面一個階段往後推，再往前一個階段後，再往後推。

還有在寫的過程中，有用到git，git真的相當方便，終於不是在用LINE來做版本更新了呵呵呵，而且也更熟悉git的指令。還有學到怎麼寫readme，還有makefile。

總之寫完，學到了很多，也感到很有成就感，相信這個專題帶給我相當多的知識！

廖怡誠:

我認為最困難的部份是架構設計，因為要用軟體模擬硬體的實際情況有很多的條件限制，像是無法同步處理、模擬元件的大小較難設計成一樣的，且上課教過比較完整的電路架構只有forwarding的版本，因此在data　hazard如何處理的部份，花了許多的心思，最後也只能突發奇想的設計一下。雖然在實做之前已經有好好想過要怎麼設計，但開始做之後又會發現許多的問題，像是stall和beq的判別，等一輪的cycle結束後再一次做判別，才不會出現問題，否則就會有遇到困難中提到的問題發生，在這個環節改了很多次的架構，有幾次的更動反而讓整體的結構都被影響到，越寫越亂，經過一番波折才決定用現在的處理方式，這部份應該是整份專題最有挑戰性的地方，所以在做出來之後覺得很有成就感。

在Readme的設計中，為了要讓助教可以快速編譯執行，還嘗試寫了Makefile，覺得還蠻有趣的，在其他課程上看到其他助教使用，就覺得是一個能夠避免環境問題的好方法，所以這次硬是抽出時間把Makefile做出來，成果也相當不錯。