

## Pipelined CPU

10801128 陳俊鴻

Q1: 請寫出在你的電路中，各 pipeline register中所設計的欄位。

A1:

IF/ID	ID/EX	EX/MEM	MEM/WB
PCadder1_sum	RegDst	RegWrite	RegWrite
Instruction	RegWrite	MemRead	MemtoReg
	ALUOp	MemWrite	FUResult
	ALUSrc	MemtoReg	RD_addr
	Branch	FUResult	
	BranchType	RT_data	
	Jump	RD_addr	
	MemRead		
	MemWrite		
	MemtoReg		
	Instruction		
	PCadder1_sum		
	RS_data		
	RT_data		
	SignExtend		
	ZeroFilled		

Q2: (a) 請說明為了將 CPU設計成 pipelined形式，對原 single cycle CPU中那些模組做了甚麼修改、或新增了甚麼功能的模組。(b) 請說明為了以 stall方式處理 data hazard而新增或修改了哪些電路。

A2:

(a) 新增了 pipeline register, hazard detection, 和 forwarding 。(b) 用 hazard detection 在需要 stall 時傳送 flush 訊號給 ID/EX 和 EX/MEM 的 pipeline register 。

Q3：請拍下你 在 “CO\_Lab4\_testdata2.txt ” 這個測資中 的 Data Hazard Detection Unit 以及 Control Hazard Detection Unit 的波形圖 貼入報告中 ，並說明 其波形 (Hazard detect) 是否正確 。

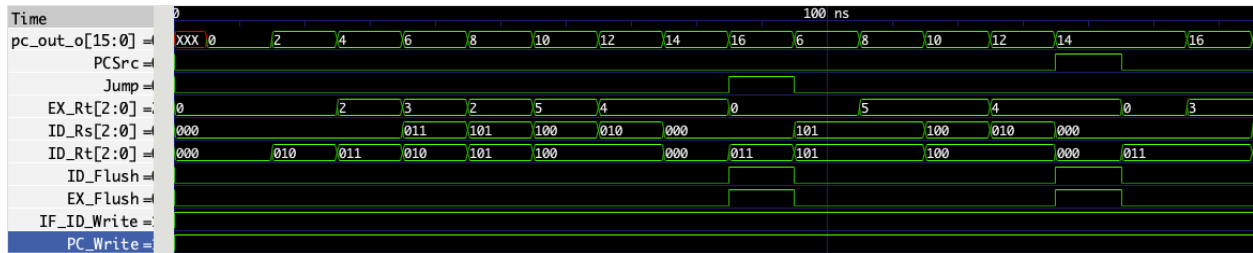
A3:

Data hazard unit (forwarding):



將 EX 之 RS 和 RT 與 MEM 和 WB 之 RD 對應便能得到正確的 forwardA 和 forwardB 。

Control hazard detection unit (hazard detection):



Q4：請寫出你在 Lab4 中所碰到的困難，以及實作完成後的心得。

因為之前是 single cycle CPU，不需要擔心 next PC 的問題，但這次 pipeline 的設定出現了一個新的問題，如何讓前幾個 cycle 無條件先採取 PC+2，這個問題之所以重要是因為 PCSrc 是在 EX stage 決定，因此若沒有在 pipeline register 先寫好初始化為 0 可能會遇到 PC 無從定義的窘境，然而即使初始化還是只能應付一個週期，因此我還是寫了偵測 x 訊號的條件來解決跨兩週期的未定義訊號問題。另外因為這次線比上次多了好幾倍，寫的時候不免覺得真是考驗程式碼的可讀性，否則一不小心線接錯或代碼大小寫寫錯又要花不知道

幾倍時間來檢查彌補錯誤。最後是一個我至今還是不太確定的問題，因為我的開發環境不是 Vivado 而是 iverilog，因此我不知道是不是只有我的波形圖在跑完所有 cycle 後還會繼續跑好幾個不知道從哪來的 instruction，詭異的是在題目規範的cycle內結果是完全正確的，但因為電腦多跑了幾個不明的cycle，導致部分結果反而被這些亂碼覆蓋了，這個問題我找了很久依然找不到成因，如果閱卷者知道是什麼原因拜託告訴我，謝謝。