

Computer Architecture-Project1

b06209027 李冠勳

Development environment

- linux
- iverilog

Implement each module

比較值得提的是以下三個module

Hazard_Detection module + ID stage的Adder

- 在ID stage的adder因為jump的PC有可能是負的所以需要宣告成signed reg
- Hazard Detection判斷是否用到需要從memory讀值的情況，如果有需要告知IF/ID pipeline registers保留上一個clock的數值;通知PC module不要增加PC的數值並且將Control module的數值全部設為0

Forwarding_Unit module + 2個MUX4 module

- 藉由判斷EX stage是否要用到MEM和WB stage要存回相同的register來決定要使用從ID讀取到的數值或MEM stage的數值或是WB stage的數值
- 通過兩個MUX4來判斷要取用的數值

IF_ID_Registers

- 其他的pipeline registers只需要將數值存起來下一個clock時將數值釋放即可，但是IF/ID這部分需要邏輯判斷stall和flush
- 處理非同步問題將always分兩部分，一部分判斷是否有新的value進來進而更新registers數值，另一部分監聽clock，在clock來時將數值釋放
- 此外同時發生flush和stall的情況應以flush為優先，因為flush之後就不會去考慮前一個cycle stall的情況

Difficulties

- 接線比作業四複雜許多要很細心不然就會接錯
- IF/ID Pipeline 非同步的問題困擾很久，最後通過將邏輯判斷與clock分開的方式解決
- 希望可以至少兩人一組，遇到問題多一人可以討論不然上網查資料查很久QQ