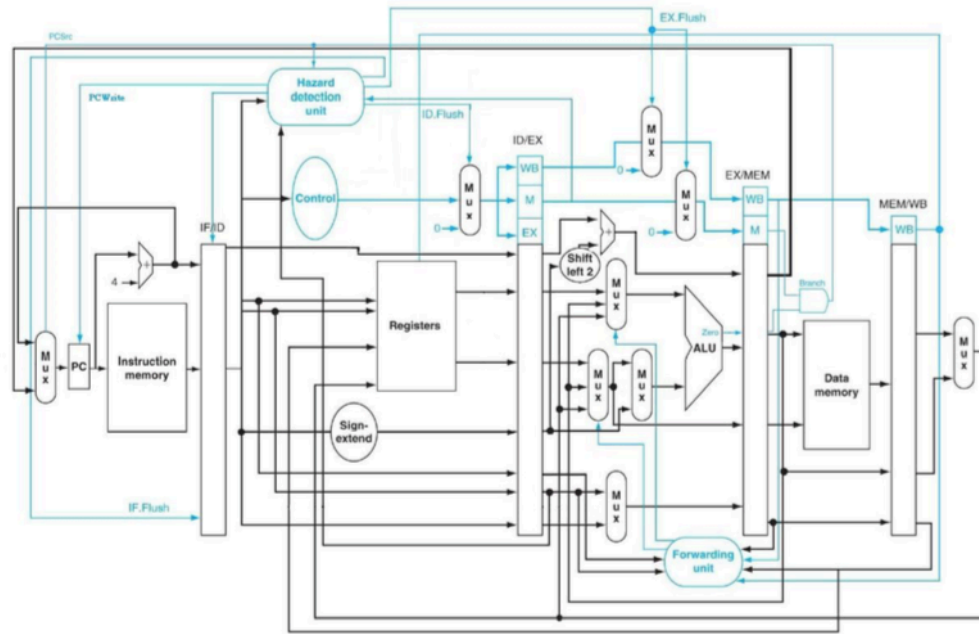


作業名稱: LAB5 The Arithmetic Logic Unit

系統架構:

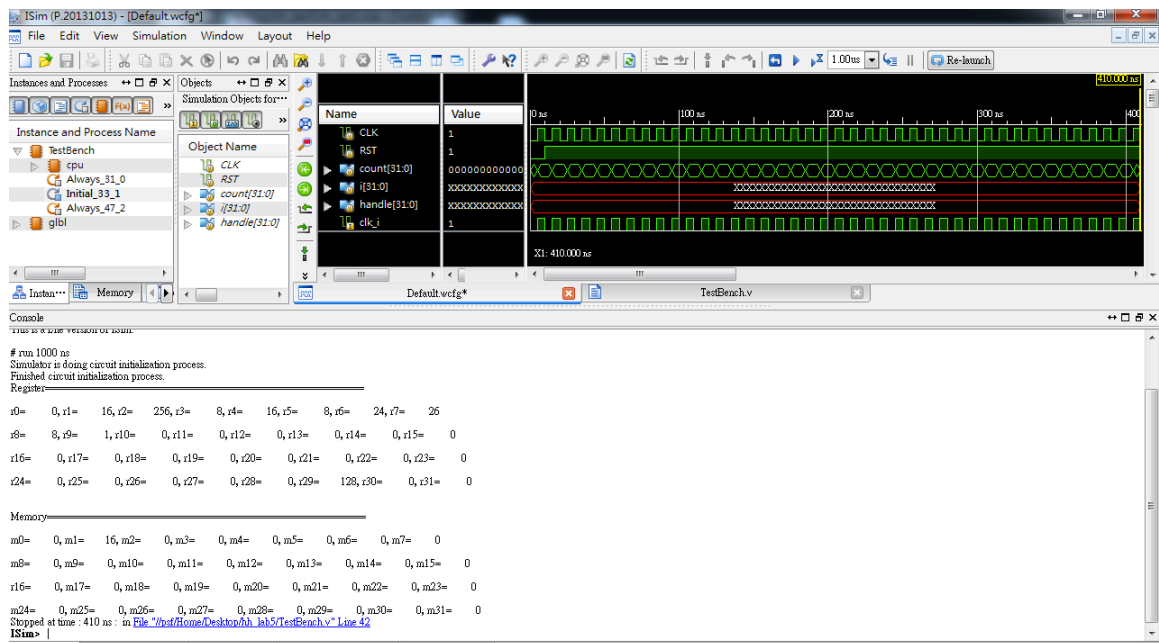


設計模組分析:

在 pipe_cpu_1 裡這個 module 主要是要讓我們接線，而這次比上次多了幾個 module 也就是 hazard detection unit 還有 forwarding unit 這兩個 module，hazard detection unit 主要是在偵測 load-use 的情況，如果遇到 load-use 就 stall the pipeline，而 forwarding unit 這個 module 是用來控制 mux，用 forwarding 回來的值。

設計結果與功能說明:

hazard detection unit 主要是在偵測 load-use 的情況，如果遇到 load-use 就 stall the pipeline，而 forwarding unit 這個 module 是用來控制 mux，用 forwarding 回來的值，如果是在 ex 這一級遇到 hazard 也就是 $ex/mem.regwrite = 1$ ，而且 $ex/mem.registerrd \neq 0$ ，而且 $ex/mem.registerrd == id/ex.registerrr$ ，就讓這裡的 mux 選擇 forwarding 回來的值，而在 mem 這一級遇到 hazard，為了防止遇到和前述一樣的情況，使得 mux 不知道要填哪一個值，所以不等於前述情況外， $mem/wd.regwrite = 1$ ，而且 $mem/WB.registerrd \neq 0$ ，而且 $mem/wb.registerrd == id/ex.registerrr$ ，如果符合上述情況，就讓 mux 選擇 wb forwarding 回來得值。



遭遇困難與解決方法:

在此次遭遇的困難就是因為多增加了 hazard detection unit 的 if flush，id flush 之類的東西，讓每一級和每一級之間的 register 多增加了幾個 bits，而我們是利用 lab4 再來擴增的，所以導致有些 bit 數算錯，在 debug 這方面有一點困難。

作業心得討論:

這次的 lab 讓我們更了解了 hazard detection unit 與 forwarding unit 的運作情況，像是當什麼情況要 forwarding 哪一級回傳的值，才不會導致 alu 運算出錯，還有什麼情況要 stall the pipeline，都讓我們有更深一層的了解。