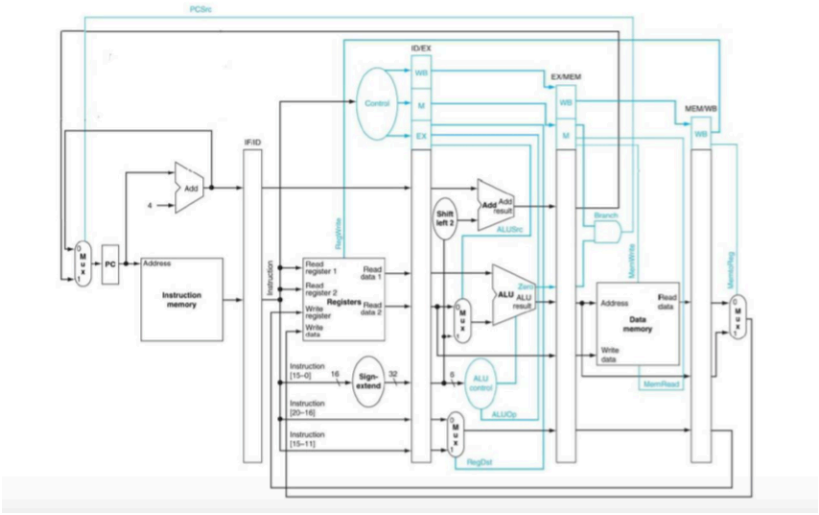
作業名稱：Lab 4: Pipelined CPU I

系統架構：




設計模組分析：

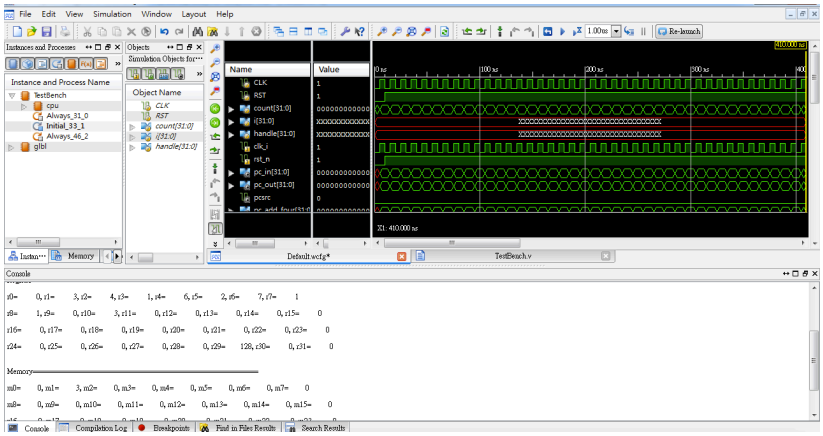
在 pipe_cpu_1 裡這個 module 主要是要讓我們接線，這次的 cpu 裡多了五個 stage 是這次作業主要的目的：pipeline，而 pipeline 把 cpu 分成了 5 個 stage：IF,ID,EX,MEM,WB，而每兩個 stage 中都會有一個 register，裡面存著每個 stage 做完事之後的值，在沒有 forwarding 的情況下，為了避免因為 data dependency 所造成的 data hazard，這次我們使用了修改測資的方式，來使其順利進行，而在 alu 這個 module 就是在做一些運算，其他大概都和上次 lab 差不多。

設計結果與功能說明：

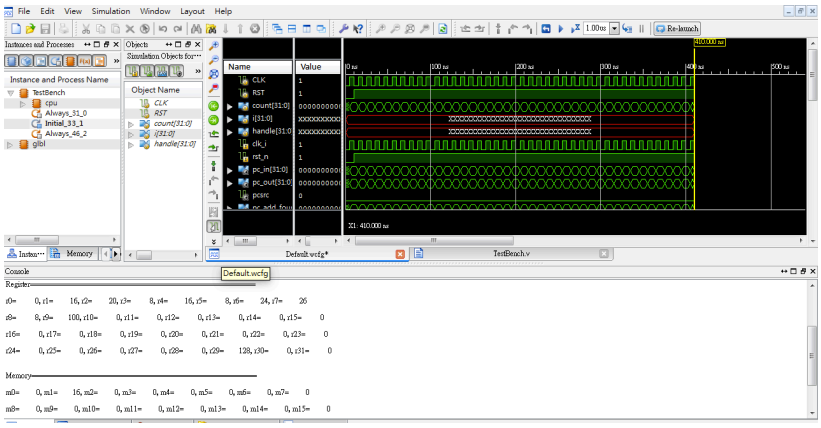
Pipelined cpu因為被分成了5個stage，所以每一層都是有關的，其中五個stage分別為IF, ID, EXE, MEM, WB, 而也因為共5個 stage，所以該 cpu 的整體效能共能比一般的 single cycle 的 cpu 加快5倍，而 IF stage 的主要作用是輸出 PC 值並且讀出 instruction，而 ID stage 的主要作用是將 IF stage 的 instruction 分解，而 EXE stage 的作用是將 IF stage 所分解的訊號進行處理， MEM stage 主要為偵測資料有沒有需要寫回記憶體或讀出記憶體，最後的WB stage 為偵測看所算出的資訊有沒有需要寫回暫存器。

Test1:

註解 [sy1]:



Test2:



遭遇困難與解決方法：

這次 lab 裡遭遇的困能我記得有兩個，第一個是因為上次的 lab 在 data_memory 傳出 data 的時候 mux 是 1，而這次傳出的時候 mux 是 0 導致一開始我們的值都是 xx，而第二個地方就是我們的 alu_result，在沒有做事就是在 else 的地方應該要給 0，如果一開始沒給 0 會導致沒有做事時 rs 會被 don't care 掉會導致都是 xx，改過後就成功了。

作業心得討論：

這次的 lab 比上次多增加了一些 wire，而這些 wire 的目的是為了減少 cycle time，讓我們得以使用 pipeline。