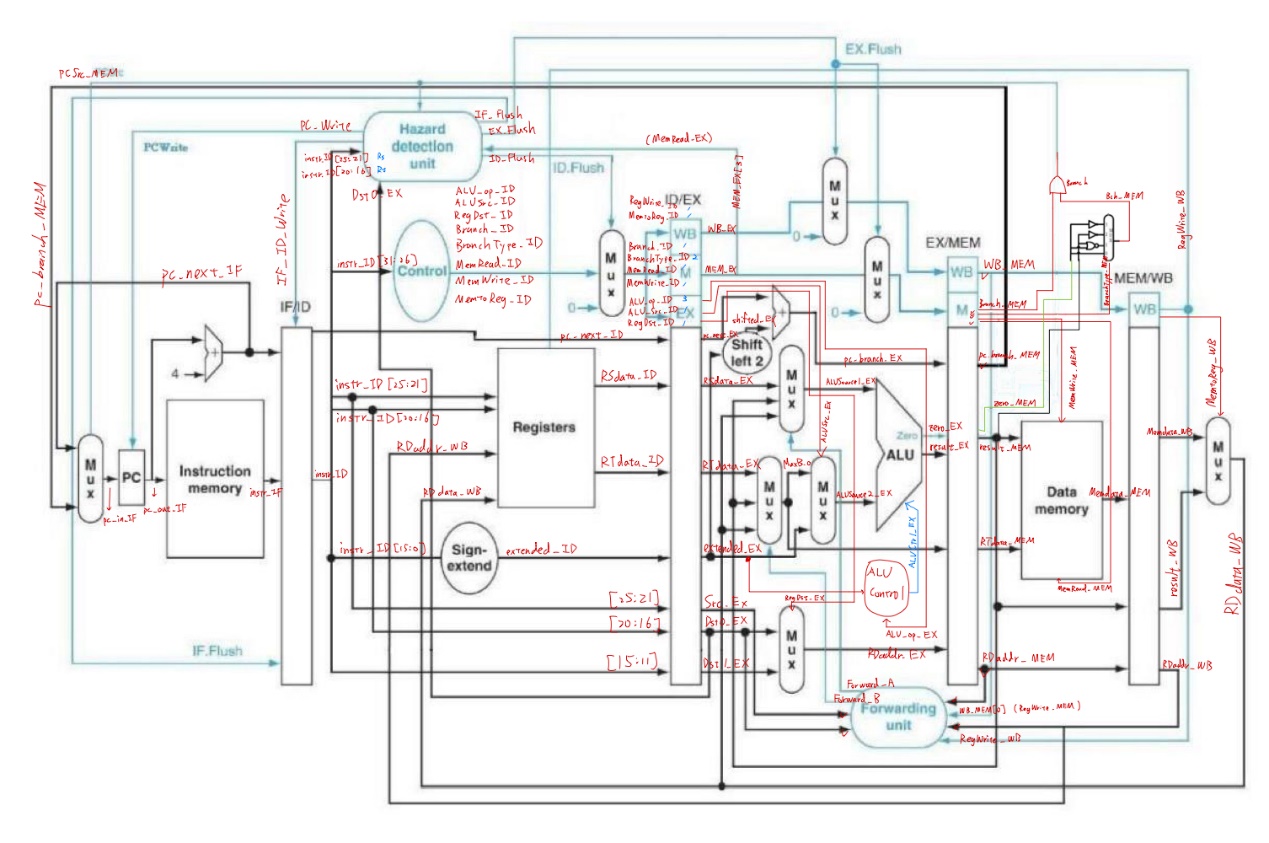
**Computer Organization Lab5**

**Name: 陳子祈**

**ID: 0819823**

**Architecture diagrams:**



**Hardware module analysis:**

1. forwarding:

input [4:0] EX\_Rs;

input [4:0] EX\_Rt;

input [4:0] MEM\_Rd;

input MEM\_RegWrite;

input [4:0] WB\_Rd;

input WB\_RegWrite;

output [1:0] Forward\_A;

output [1:0] Forward\_B;

description:

1. 檢查EX\_Rs、EX\_Rt是否為MEM\_RD及MEM\_RegWrite是否是1及MEM\_RD是否為非0，如果以上條件都成立則使用Forwarding，以此來解決MEM的data hazard。
2. 若MEM\_Rd == EX\_Rs或EX\_Rt不成立，接著再檢查EX\_Rs、EX\_Rt是否為WB\_Rd且WB\_RegWrite是否為1及WB\_Rd是否為非0，若上述條件都成立則繼續使用Forwarding，以此來解決WB的data hazard。在發現WB的data hazard並做Forwarding之前，要先排除MEM的data hazard的發生才能使用Forwarding，避免double data hazard的發生。

2. HazardDetection:

input EX\_MemRead;

input [4:0]EX\_Rt;

input [4:0] ID\_Rs;

input [4:0] ID\_Rt;

input PCSrc;

output PC\_Write;

output IF\_ID\_Write;

output IF\_Flush;

output ID\_Flush;

output EX\_Flush;

description:

HazardDetection主要是在偵測Load use data hazard或Branch是否發生，並作相對應的動作。

1. 如果EX\_MemRead ==1 且 EX\_Rt==ID\_Rs或ID\_Rt 則代表Load use data hazard發生，這時要stall one block，將PC及IF/ID\_Reg保存起來，就是設定result\_PC\_Write=0且result\_IF\_ID\_Write=0，除此之外，還需要製造一個bubble，也就是設定result\_ID\_Flush=1。
2. 當PCSrc為1代表Branch發生，此時要防止前面幾個不需要執行的指令更新暫存器或記憶體，將IF、ID、EX正在執行的指令都Flush，也就是把IF/ID、ID/EX、EX/MEM的Pipeline Register全部變成0，因此輸出IF\_Flush、ID\_Flush、EX\_Flush皆為1。

其他module之前都已經有詳細說明，在此LAB就不贅述。

**Finished part:**

請助教在測試testbench.v的時候可以將MAX\_COUNT調成64，如下:

一張含有 文字 的圖片

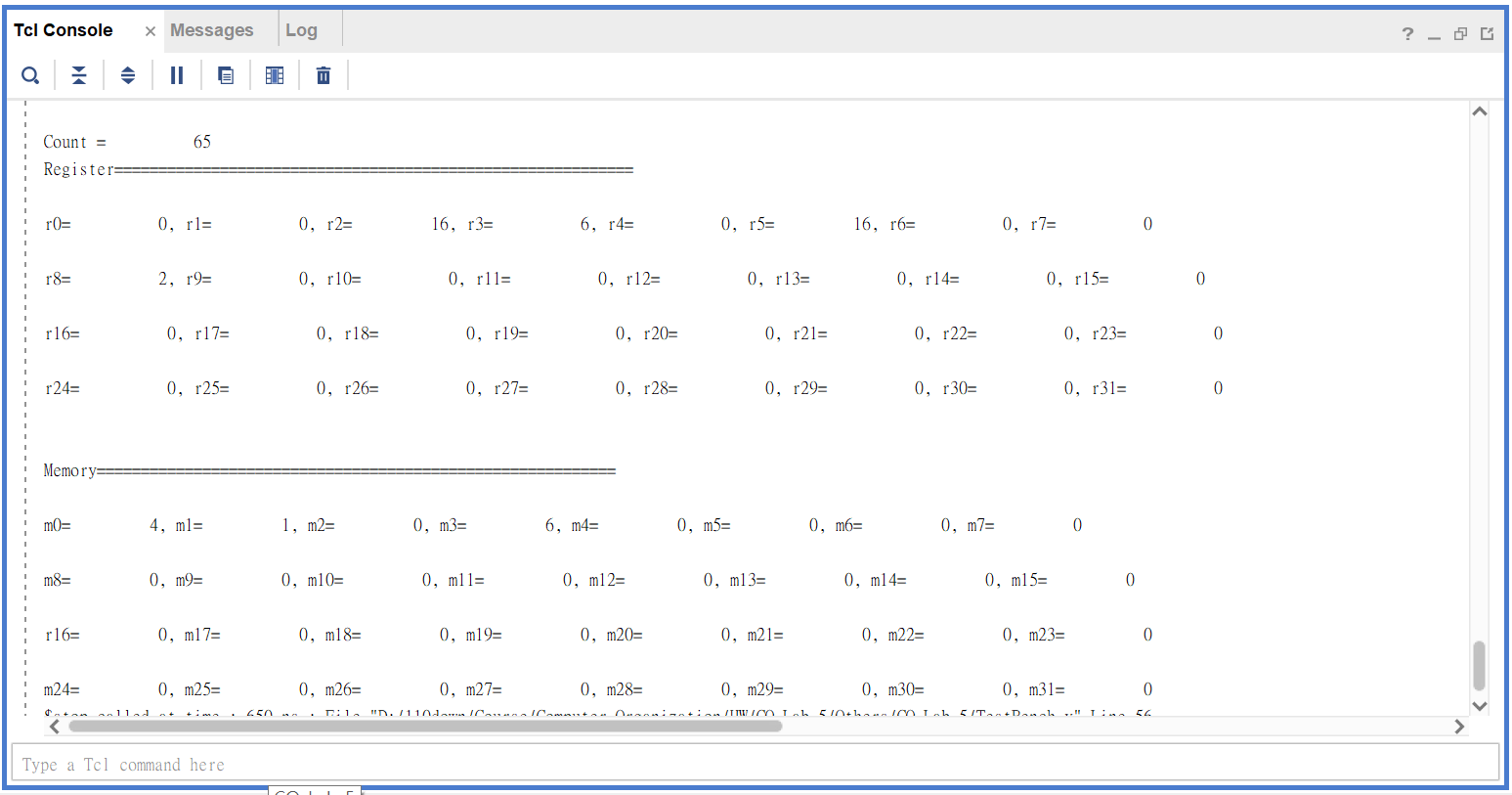
自動產生的描述

For CO\_P5\_test\_1.txt:

一張含有 文字, 螢幕擷取畫面, 電腦, 膝上型電腦 的圖片

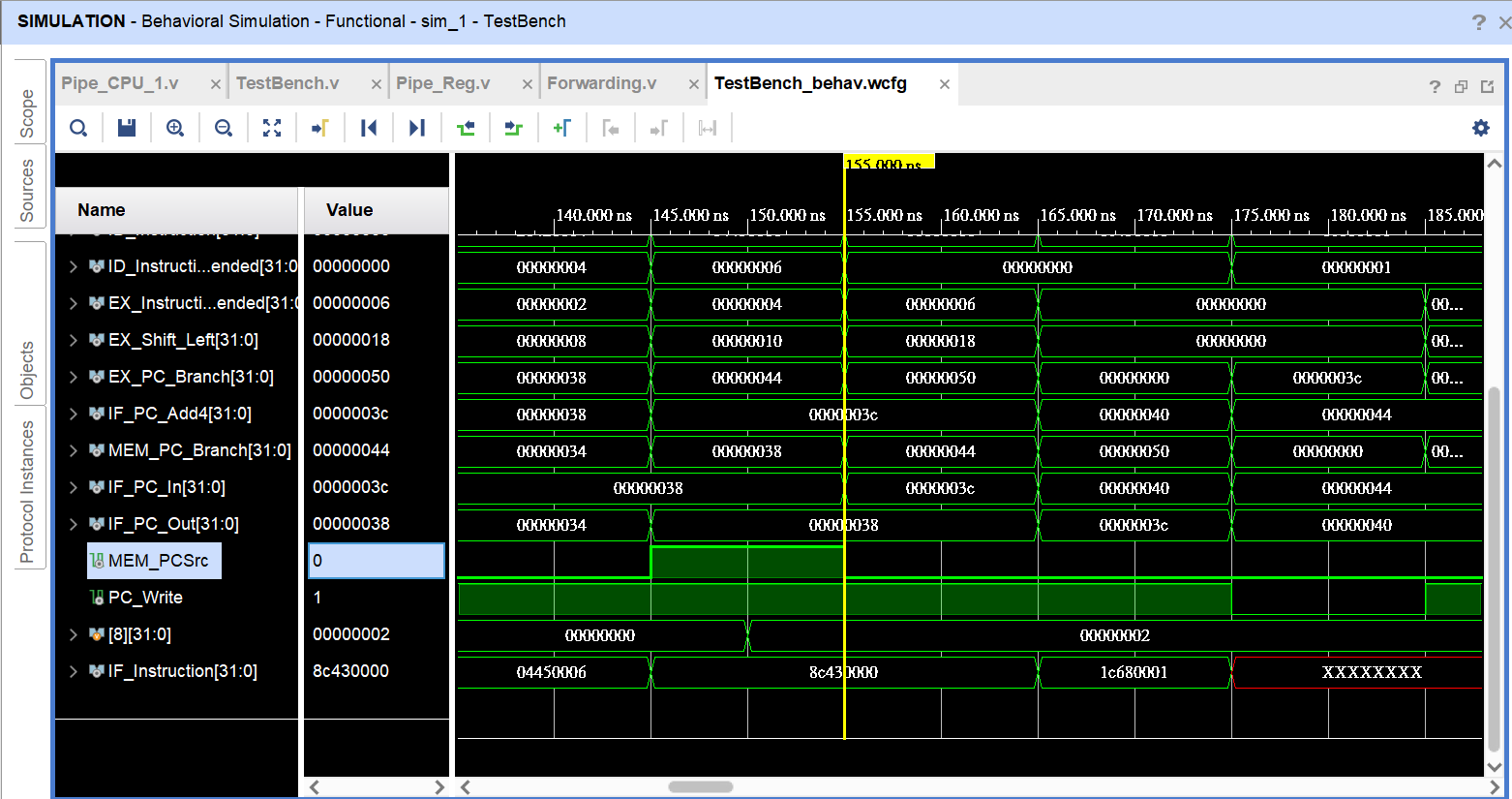
自動產生的描述

For CO\_P5\_test\_2.txt:



**Problems you met and solutions:**

Q1: 發現IF\_Instruction會讀不到值。



A1: 因為我原本instruction\_file設為15，但實際上指令有21個，所以後面會有一些指令讀不到

一張含有 文字 的圖片

自動產生的描述

Q2: bgt、bne寫錯，找很久才發現有錯。

A2: 原本以為branch指令跳到錯誤的指令位置，後來發現是我自己bgt、bne寫錯，改正確就好了。

一張含有 文字 的圖片

自動產生的描述

**Summary:**

這次的LAB比之前難很多，雖然只有加Forward及Hazard detection兩個模組，可是要注意很多細節，尤其是判斷hazard的部分。經過學長提醒，我才知道我後面很多指令讀不到原來是因為Instruction\_file設太少。另外，我還以為遇到Branch的問題，結果原來是自己寫錯bgt及bge。這次學到很多，希望暑假再好好精進實作的能力，我覺得自己很長粗心，找錯要花很多時間，我花了整整三天才搞清楚問題在哪裡，希望下次可以好好加油。