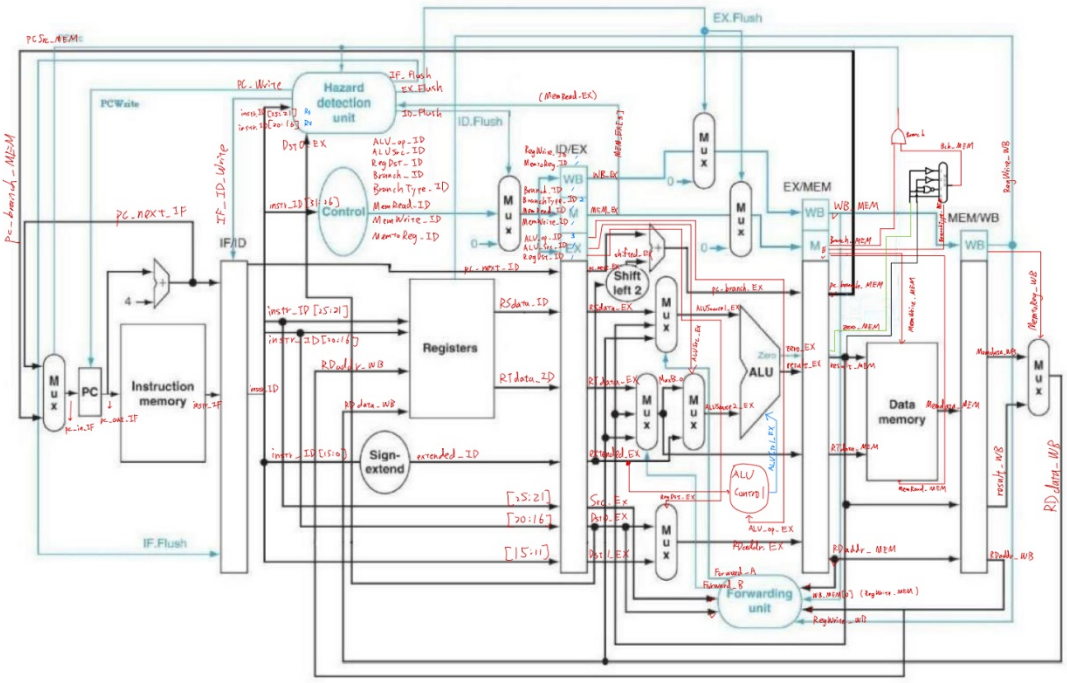
Computer Organization Lab5

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;

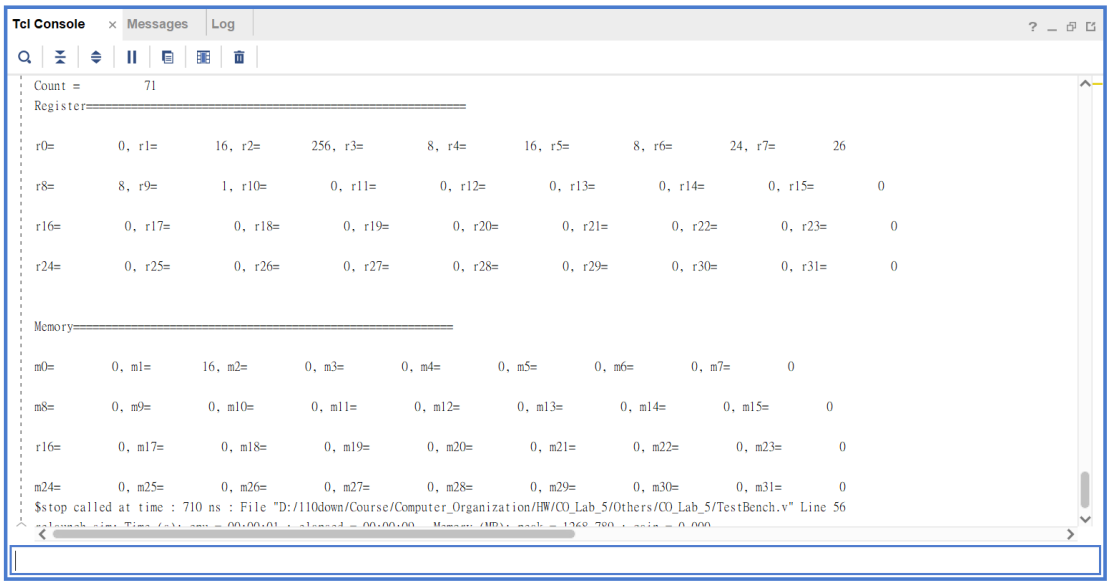
description:
HazardDetection主要是在偵測Load use data hazard或Branch是否發生，並作相對應的動作。

(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。

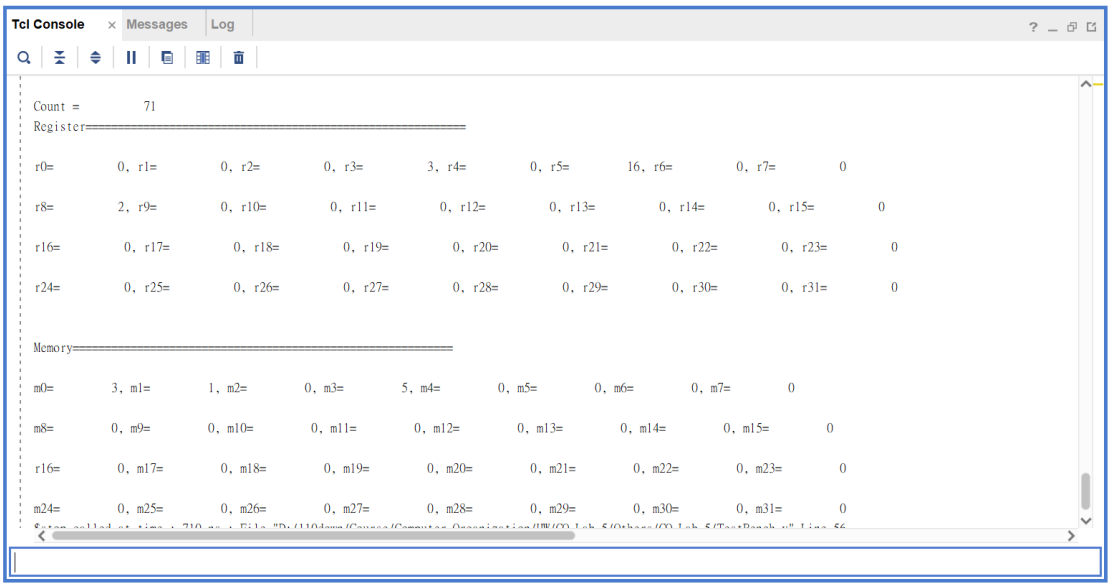
(2) 當PCSrc為1代表Branch發生，此時要防止前面幾個不需要執行的指令更新暫存器或記憶體，將IF、ID、EX正在執行的指令都Flush，也就是把IF/ID、ID/EX、EX/MEM的Pipeline Register全部變成0，因此輸出IF_Flush、ID_Flush、EX_Flush皆為1。

Finished part:

For CO_P5_test_1.txt:

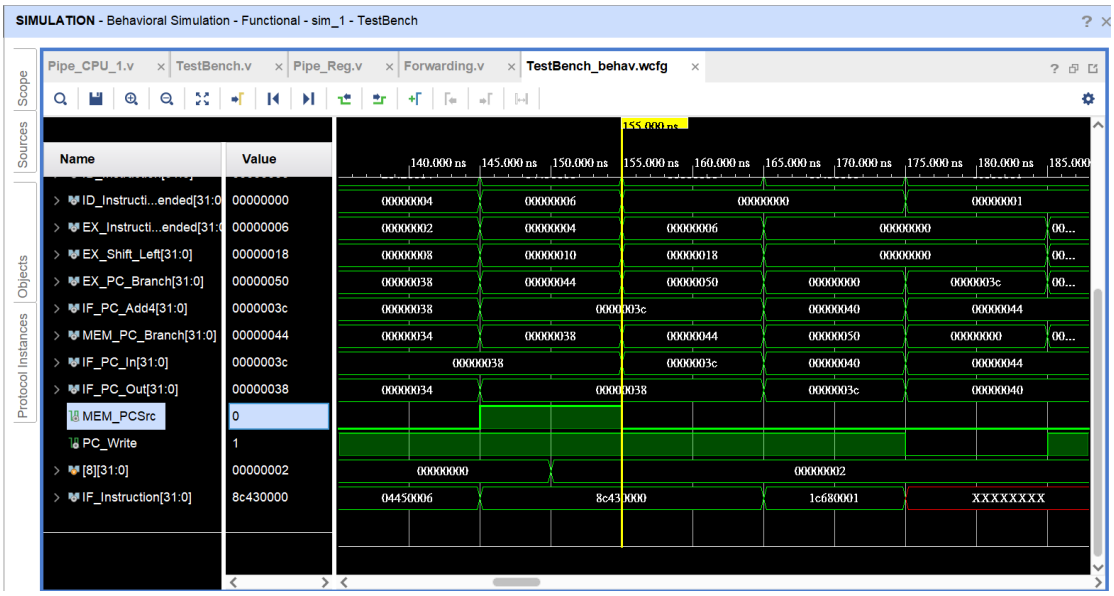


For CO_P5_test_2.txt:



Problems you met and solutions:

Q: 我發現如果 branch 有跳的話，MEM_PCSrc 在 MEM stage 更新為 1，MEM_PC_Branch 也會在 MEM stage 更新為要跳的 PC，可是不知道為什麼兩個更新的時間不同。MEM_PCSrc 比 MEM_PC_Branch 早一個 cycle time 就更新了。



A:

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。我還遇到 Branch 的問題，本來想說上次怎麼沒遇到這個問題，看了上次給的測資才發現上次的 beq 也沒有真的跳，所以 PC 都沒遇到問題，不過這次 beq 有跳，我才發現沒有完全處理好 Branch 的 PC。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。