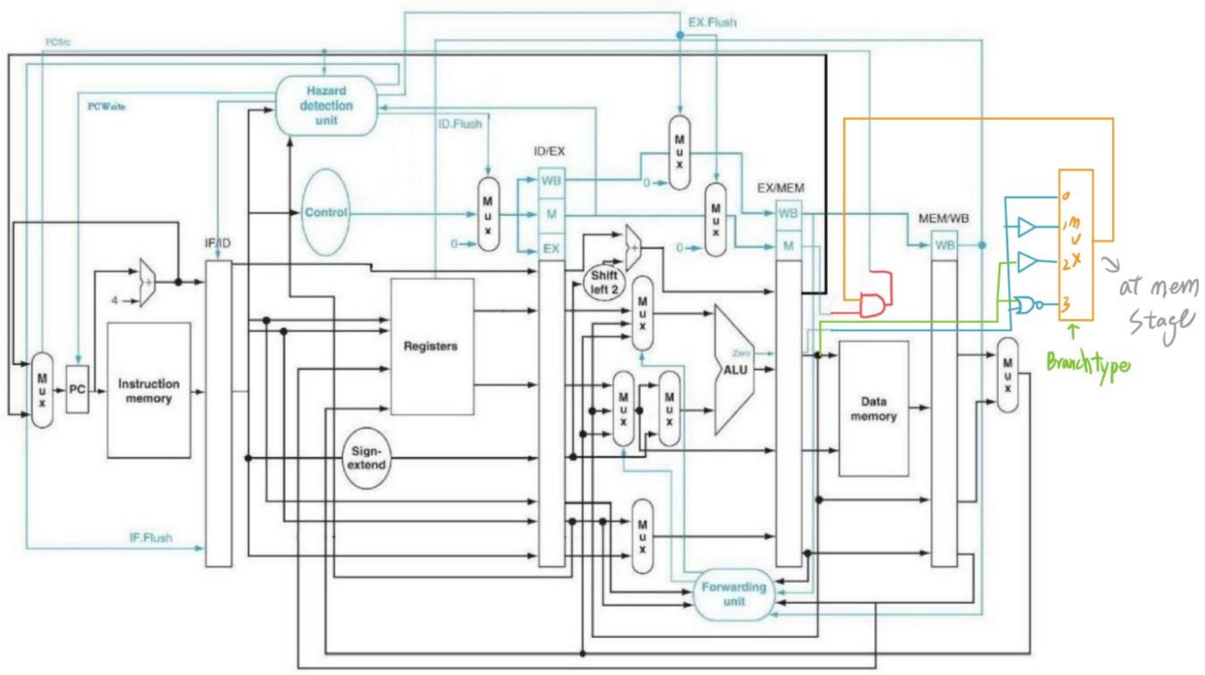
Computer Organization Lab5

Name: 梁詠晴

ID: 109550134

Architecture diagrams:



比起 lab4 新增 hazard detection unit 和 forwarding unit

Hardware module analysis:

Hazard detection : 在 branch、hazard 時決定 flush、保留 pc

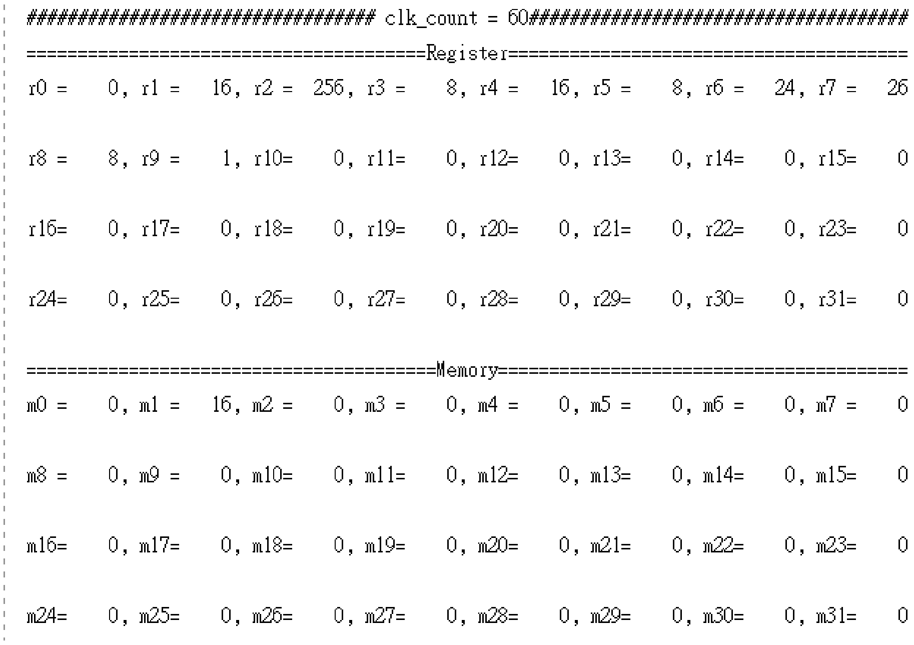
Forwarding : 在可以 forward 解決的 hazard 情況下，偵測並透過 forward a、b 解決 hazard

Pipe reg : 新增 flush 和 new，讓 flush 時值為 0、new=0 時保留原本值

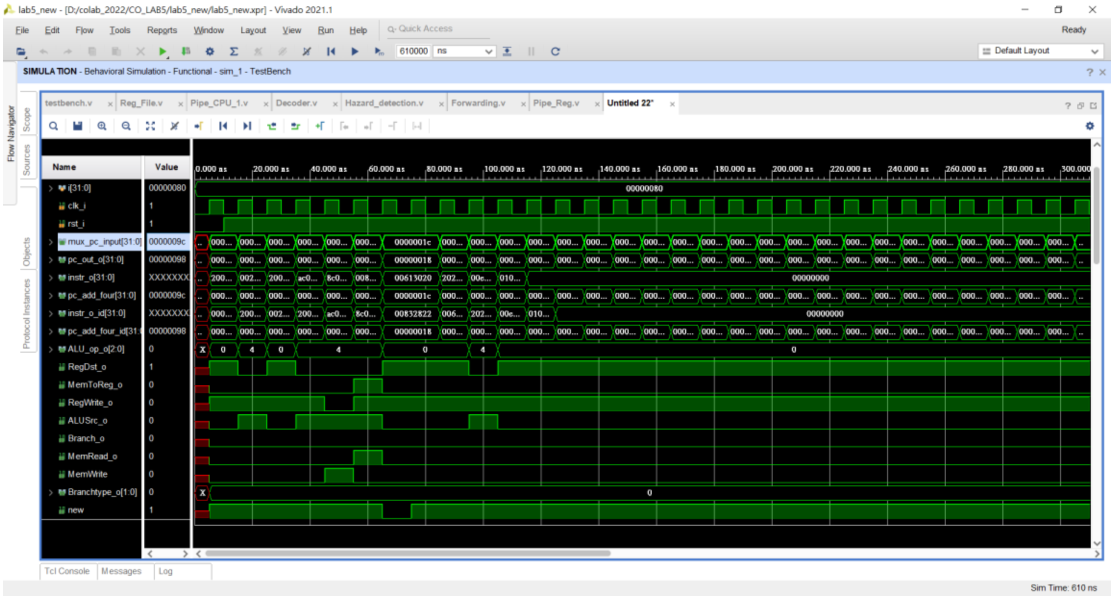
Branch type : lab5 新增 BGE、BGT 等指令，透過 branch type 選擇不同 pc source

Finished part:

Testcase 1:

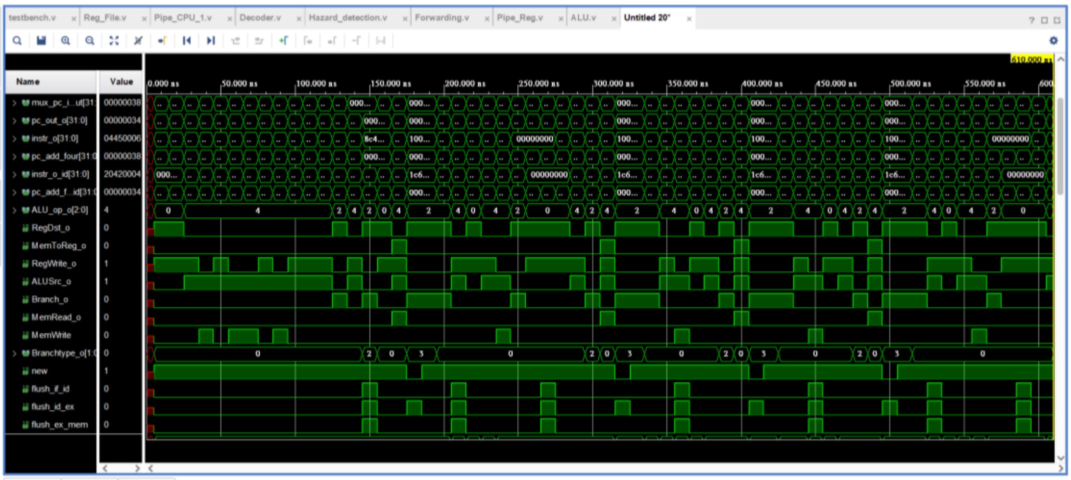


```
################################## clk_count = 60####################################
=======================================Register=======================================
r0 =    0, r1 =   16, r2 =  256, r3 =    8, r4 =   16, r5 =    8, r6 =   24, r7 =   26

r8 =    8, r9 =    1, r10=    0, r11=    0, r12=    0, r13=    0, r14=    0, r15=    0

r16=    0, r17=    0, r18=    0, r19=    0, r20=    0, r21=    0, r22=    0, r23=    0

r24=    0, r25=    0, r26=    0, r27=    0, r28=    0, r29=    0, r30=    0, r31=    0

========================================Memory========================================
m0 =    0, m1 =   16, m2 =    0, m3 =    0, m4 =    0, m5 =    0, m6 =    0, m7 =    0

m8 =    0, m9 =    0, m10=    0, m11=    0, m12=    0, m13=    0, m14=    0, m15=    0

m16=    0, m17=    0, m18=    0, m19=    0, m20=    0, m21=    0, m22=    0, m23=    0

m24=    0, m25=    0, m26=    0, m27=    0, m28=    0, m29=    0, m30=    0, m31=    0
```



```
i[31:0]              00000080
clk_i                1
rst_i                1
mux_pc_input[31:0]   0000009c
pc_out_o[31:0]       00000098
instr_o[31:0]        XXXXXXXX
pc_add_four[31:0]    0000009c
instr_o_id[31:0]     XXXXXXXX
pc_add_four_id[31:0] 00000098
ALU_op_o[2:0]        0
RegDst_o             1
MemToReg_o           0
RegWrite_o           0
ALUSrc_o             0
Branch_o             0
MemRead_o            0
MemWrite             0
Branchtype_o[1:0]    0
new                  1
```

Testcase 2:



Name	Value
mux_pc_i...ut[31:	00000038
pc_out_o[31:0]	00000034
instr_o[31:0]	04450006
pc_add_four[31:0	00000038
instr_o_id[31:0]	20420004
pc_add_f...id[31:0	00000034
ALU_op_o[2:0]	4
RegDst_o	0
MemToReg_o	0
RegWrite_o	1
ALUSrc_o	1
Branch_o	0
MemRead_o	0
MemWrite	0
Branchtype_o[1:0	0
new	1
flush_if_id	0
flush_id_ex	0
flush_ex_mem	0

```
################################### clk_count = 60######################################
========================================Register========================================
r0 =      0, r1 =      0, r2 =     12, r3 =      6, r4 =      0, r5 =     16, r6 =      0, r7 =      0

r8 =      2, r9 =      0, r10=      0, r11=      0, r12=      0, r13=      0, r14=      0, r15=      0

r16=      0, r17=      0, r18=      0, r19=      0, r20=      0, r21=      0, r22=      0, r23=      0

r24=      0, r25=      0, r26=      0, r27=      0, r28=      0, r29=      0, r30=      0, r31=      0

=========================================Memory=========================================
m0 =      4, m1 =      1, m2 =      0, m3 =      6, m4 =      0, m5 =      0, m6 =      0, m7 =      0

m8 =      0, m9 =      0, m10=      0, m11=      0, m12=      0, m13=      0, m14=      0, m15=      0

m16=      0, m17=      0, m18=      0, m19=      0, m20=      0, m21=      0, m22=      0, m23=      0

m24=      0, m25=      0, m26=      0, m27=      0, m28=      0, m29=      0, m30=      0, m31=      0
```

Problems you met and solutions:

這次 lab 一樣遇到了接錯線的問題，後來藉由比對波形圖，成功找到因為變數名稱太像而出錯的地方

Summary:

一直到 Lab4 要寫 report 時我才發現可以透過 add to window 將 cpu 的波形顯示出來，並對比波形找出錯誤，也因此 Lab5 debug 的時間相對以前較少。

下次在寫 code 前一定要先備齊工具再開始呀!