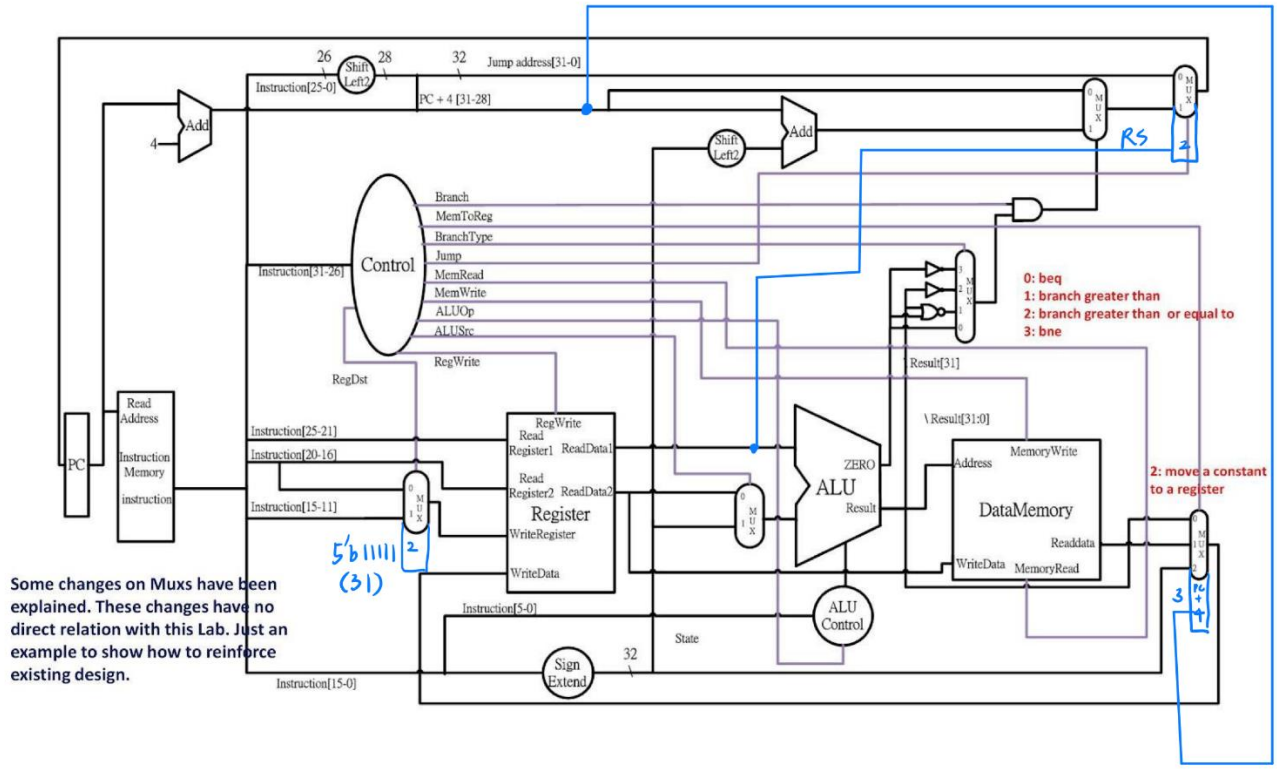
Computer Organization Lab2

Name: 梁詠晴

ID:109550134

Architecture diagrams:



新增結構連線:

1. WriteData input :
   Case(MemToReg)
   0 : alu result
   1 : DM read data
   2 : sign extend
   3 : pc+4 (adder1 result)

2. WriteRegister input :
   Case(RegDst)
   0 : instruction[20-16]
   1 : instruction[15-11]
   2 : 5’b11111

3. PC input :
   Case(Jump)
   0 : Jump addr
   1 : Branch mux result
   2 : RS data

Hardware module analysis:

PC : 表示程式執行的位置，每回合加 4，jump 時則會跳到指定的位置
Adder : 將 2 個 32bit 的 source 相加
IM : 讀 pc 的位置，並依照 I-format、R-rormat 不同輸出相對應 address 的指令
MUX_2/3/4 to1: 根據輸入條件決定要輸出哪一個 input
Decoder : 根據 opcode 和 function code 決定指令需不需要 RegWrite_o, ALU_op_o, ALUSrc_o, RegDst_o, Branch_o, BranchType_o, Jump_o, MemToReg_o, MemRead_o, MemWrite
ALU_Ctrl : 根據 decoder 輸出決定 ALU 要做甚麼運算
DM：根據 mem read/write 決定要不要從 memory 中讀寫資料
ALU : 根據 ALU ctrl 對兩個輸入運算

Finished part:

Test data 1:

```
PC =        128
Data Memory =          1,         2,         0,         0,         0,         0,         0,         0
Data Memory =          0,         0,         0,         0,         0,         0,         0,         0
Data Memory =          0,         0,         0,         0,         0,         0,         0,         0
Data Memory =          0,         0,         0,         0,         0,         0,         0,         0
Registers
R0 =          0, R1 =          1, R2 =          2, R3 =          3, R4 =          4, R5 =          5, R6 =          1, R7 =          2
R8 =          4, R9 =          2, 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 =       128, R30 =         0, R31 =         0
$stop called at time : 705 ns : File "D:/colab_2022/CO_LAB3_new/lab3_new/lab3_new.srcs/sim_1/imports/testcase/testbench.v" Line 36
INFO: [USF-XSim-96] XSim completed. Design snapshot 'testbench_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:07 ; elapsed = 00:00:10 . Memory (MB): peak = 1452.758 ; gain = 0.000
```

Test data 2:

```
Data Memory =          0,          0,          0,          0,          0,          0,          0,          0
Data Memory =          0,          0,          0,          0,          0,          0,          0,          0
Data Memory =          0,          0,          0,          0,         68,          2,          1,         68
Data Memory =          2,          1,         68,          4,          3,         16,          0,          0
Registers
R0 =          0, R1 =          0, R2 =          5, R3 =          0, R4 =          0, R5 =          0, R6 =          0, R7 =          0
R8 =          0, 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 =        128, R30 =          0, R31 =         16
$stop called at time : 705 ns : File "D:/colab_2022/CO_LAB3_new/lab3_new/lab3_new.srcs/sim_1/imports/testcase/testbench.v" Line 36
xsim: Time (s): cpu = 00:00:13 ; elapsed = 00:00:06 . Memory (MB): peak = 1452.758 ; gain = 0.000
INFO: [USF-XSim-96] XSim completed. Design snapshot 'testbench_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:16 ; elapsed = 00:00:13 . Memory (MB): peak = 1452.758 ; gain = 0.000
```

Problems you met and solutions:

認為邏輯正確，卻一直跑不出正確答案。後來發現可能是因為有些 decoder、alu ctrl 運算時使用 case，但未給定正確的 default 值才導致答案錯誤。

Summary:

這次 lab 的 decoder 比上次多處理很多輸出跟 instruction。在這樣需要處理更多細節的情況時，我覺得如何把程式寫得好讀就變得很重要。本來想要直接在上次的基礎上添加變數，但未整理的 code 讓我花了很多額外的時間 debug，而整理過後效率就有因此提升。