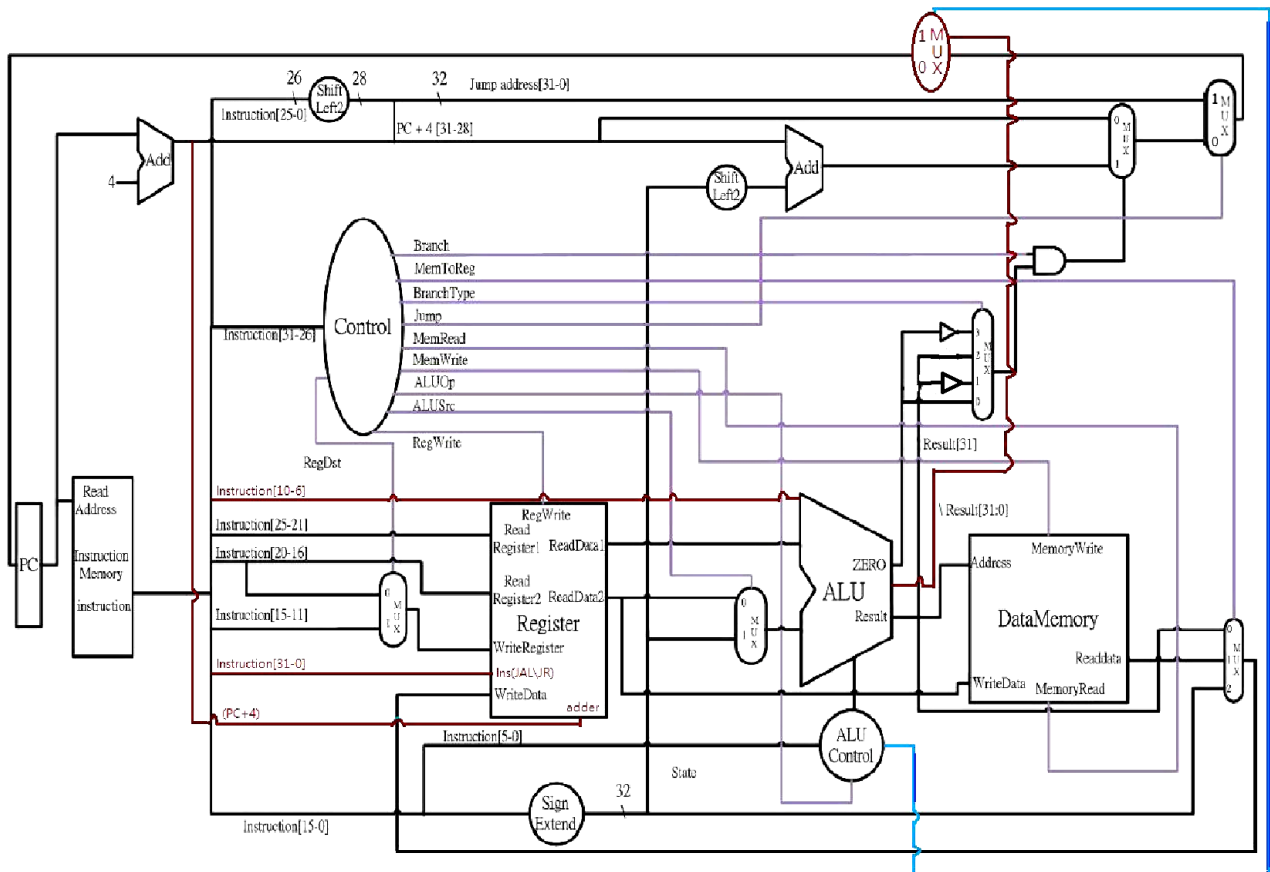


Computer Organization

0316056 蔡妮芳 0316038 石瑾旋

Architecture diagram:



Detailed description of the implementation:

1. Decoder :
多加了 mul/jump/blt/bnez/bgez/jal/jr(R-type)的 instruction_ OPcode 轉成 ALU_op。
2. ALU_Ctrl :
(1)R-type 加了 mul 和 jr 的 cases,還要另外加一個 bgez 的情況。
(2)加一條線為 Jr_o,只有在 jr 發生時=1。
3. ALU :
(1)加上一個 jr_addr 的 output,只有在 jr 發生時把 src1_i 的值給 jr_addr。
(2)加一個 mul 的 case。
4. Simple_single_cpu :
(1)因為多了 jr ,所以需要多用一個 2to1mux 來控制 PC ,此 MUX 利用 ALU

control 所拉出來的 Jr_o 來判斷是否為 JR，若為 JR 則 MUX_2to1 的 output 為 ALU 所拉出來的 jr_address，接入 PC 而做到跳轉的動作。

(2)Branch_Source 要由 4to1mux 來控制。

5. Branch 的設計：

(1)blt 設 $ALU_res = src1 - src2$ ，如果 $ALU_res[31] == 1$ 就 branch。

(2)bnez 設 $ALU_res = src1 - src2$ (是 00000)，如果 $zero_o == 1$ 就 branch。

(3)bgez 設 $ALU_res = src1 \text{ or } src2$ (是 00001)，如果 $ALU_res[31] == 0$ 就 branch。

Problems encountered and solutions:

起先照著老師給的圖做，發生問題後才發現最右上角那個 2to1 mux 應該把 0 和 1 的順序調換。

這次作業中犯下了不少很基本、不該犯下的錯，例如：沒有把測資(.txt 檔)跟 project 放到同一個資料夾，或是沒有按下” run all”，使得跑出的結果與應有的結果不同。這些基礎的錯誤都讓我們花了好幾個小時去 debug，下次會更小心，不會再犯。

Lesson learnt (if any):

1. 學會如何 implement jump,jr,jal,blt,bnez,bgez。
2. 了解 function call 如何在 MIPS 中運作。