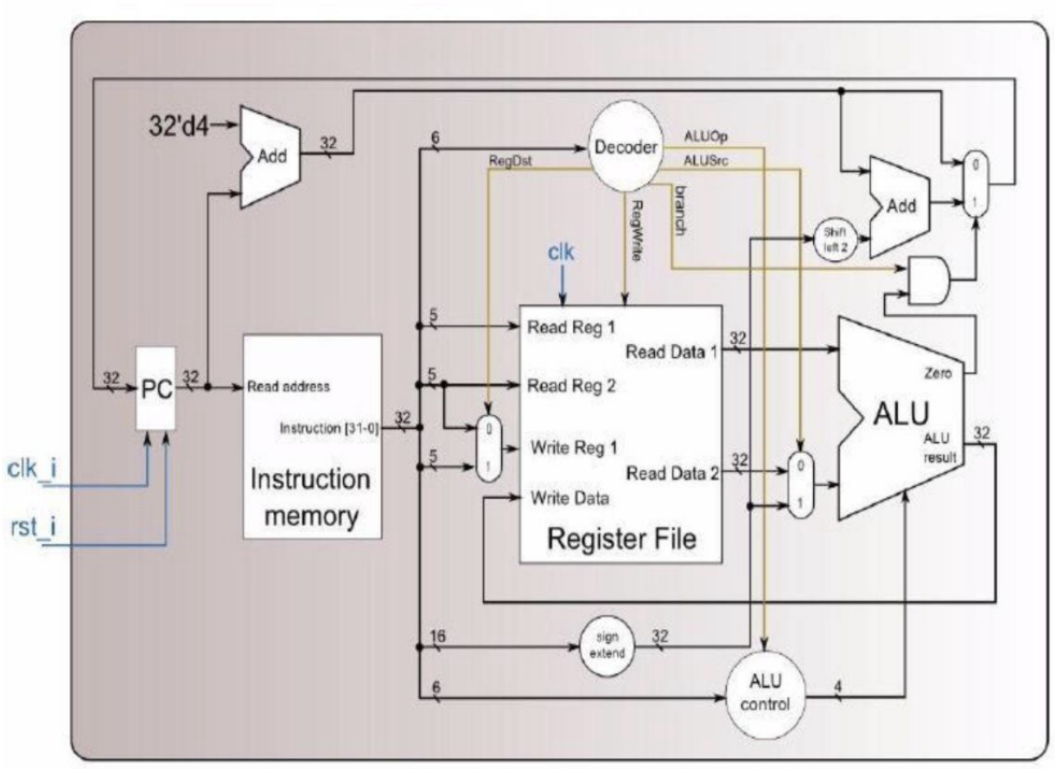
Computer Organization
Lab2: Single Cycle CPU

student ID:110550093 Name: 蔡師睿

1. Architecture diagrams



Single-cycle CPU

2. Hardware module analysis

Program Counter

當 reset 不爲 0 時，輸出輸入的訊號；若爲 0，則輸出 0。

Instruction Memory

讀取.txt 檔中的指令。

Register File

讀取輸入的暫存器的地址並輸出其儲存的值。另外根據 RegWrite 判斷，是否需要對暫存器寫入新的值。

Adder

將輸入的兩個 32-bit 數字相加輸出。

Decoder

根據 Instruction 第 31 到 26 bit 的值，輸出相對應的輸出。

Instruction[31:26]	**Output ={RegWrite, ALU_op, ALUSrc, RegDst, Branch}**
6'b000000	7'b1010010
6'b000100	7'b00010x1
6'b001000	7'b1000100
6'b001010	7'b1011100
else	7'bxxxxxxx

MUX 2 to 1

若 select 爲 1，輸出 data1；若爲 0，輸出 data0。

Shift left two

將輸入的 32-bit 數字往左移兩位補 0 再輸出。

Sign Extended

輸入的 16-bit 二進位數字的最高位若爲 1，在此數字前補 16 個 1，若爲 0，則在前補 16 個 0，最後將補完位滿足 32-bit 的數字輸出。

ALU Control

根據由 Decoder 傳入的 ALUOp 和 Instruction 第 5 到 0 bit 的值，輸出相對應的 ALU Control 值。

{ALUOp, Instruction[5:0]}	ALU Control
9'b010_100100	4'b0000
9'b010_100101	4'b0001
9'b000_xxxxxx	4'b0010
9'b010_100000	4'b0010
9'b001_xxxxxx	4'b0110
9'b010_100010	4'b0110
9'b010_101010	4'b0111
9'b011_xxxxxx	4'b0111

ALU

根據 ALU Control 傳進來的值做相對應的 And、Or、Add、Sub、Slt 或 Nor 操作。

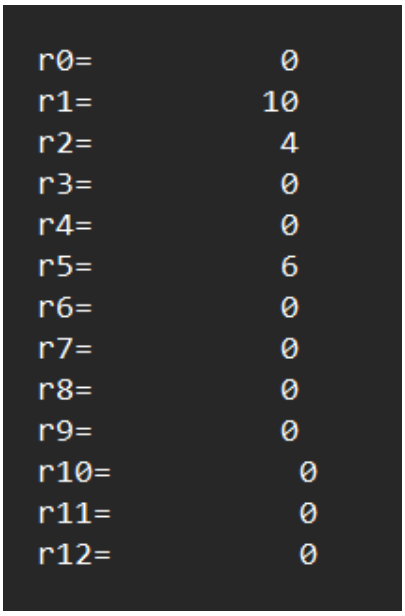
ALU action	ALU input control ={A_invert, B_invert, Operation}
And	4'b0000
Or	4'b0001
Add	4'b0010
Sub	4'b0110
Nor	4'b1100
Slt	4'b0111

Single cycle CPU

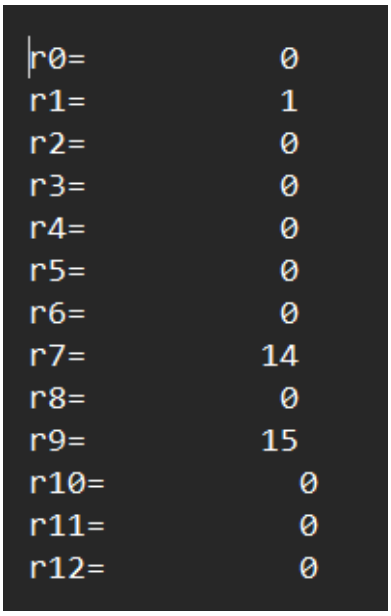
根據 Architecture diagram 和以上的 modules 組出正確的 Single-cycle CPU。

3. Finished part

Result of data1 & data2



data1



data2

4. Problems and solutions

1. 實作部分.v 檔時，一開始是用 dataflow 的形式，結果 vivado 編譯報錯，後來才知道使用 dataflow 時，assign 的等號左邊不能爲 reg。
2. Decoder 和 ALU Control 未考慮 addi 和 slti 的可能性，結果跑 data1 和 data2 時出現錯誤的結果，最後讓 Decoder 偵測到 addi 和 slti 後，ALU Control 不判斷 instruction[5:0]，才得出正確結果。

5. Summary

將 Single-cycle CPU 拆成小單位後，更能清楚地了解 Single-cycle CPU 的運作，尤其是在寫 verilog 時，由小單位組成一整體的過程也比較不容易搞混其中的邏輯，或是可讀性下降造成 debug 不易。