

# Computer Architecture HW4

B04705003 資工三 林子雋

## 1. Coding Environment :

本次作業我使用 Windows 10 x64-based 64bit OS，並且使用 Icarus Iverilog 來編譯。

## 2. Module Implementation :

—Adder.v：如同作業投影片的實作

—ALU.v：我使用 always block 來實作，並使用 case block 來選擇 Zero\_o 要放 0 或是 1，而 data\_o 要放經過+, -, \*, /, | 運算過後的值。

—ALU\_Control.v：使用 always block 實作，先判斷 ALUOp\_i 的值後再決定要放甚麼值給 ALUCtrl\_o。

—Control.v：RegDst\_o 的部分我使用 product of sum 來實作(判斷是不是 R-type)；RegWrite\_o, ALUSrc\_o, ALUOp\_o 的部分我使用 always block 來實作，判斷是不是 R-type 或是 addi。

—CPU.v：照著投影片的方式把所有 module 的線接好。

—Instruction\_Memory.v：助教已實作好

—MUX32.v：使用 question mark operator 來實作 multiplexer

—MUX5.v：使用 question mark operator 來實作 multiplexer

—PC.v：助教已實作好

—Registers.v：助教已實作好

—Sign\_Extend.v：使用 always block 將 data\_o[15:0]內的資料直接從 data\_i[15:0]拿，剩下的 16 之後的 bit 用 data\_i[15]的 bit 來填補。