

# 計算機網路 HW4

資工三 b05902001 廖彥綸

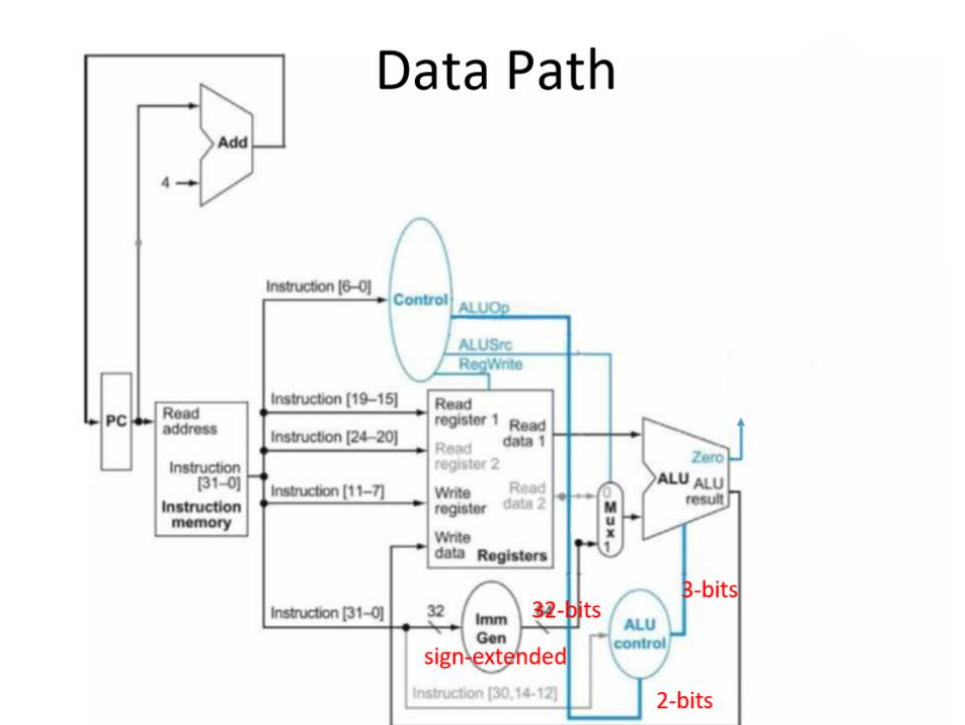
## 一、 架設環境

作業系統：Window10

模擬環境：ModelSim

編輯：WordPad

## 二、 模組實作解釋



### 1. Adder.v

參考投影片方式，宣告後直接 `assign data_o = data1_in + data2_in`。

### 2. Control.v

根據流程圖input為7位元的 `inst[6:0]` (opcode)，output 有 `ALUOp_o`、`ALUSrc_o`、`RegWrite_o` 三個，`RegDst_o` 目前不會用到暫時不予理會。

`ALUOp_o` 為兩個 bits，負責協助 `ALUcontrol`，所以將opcode不同的addi (I-format)的 `ALUOp_o`設為11其他為00。

只有 I-format會用到 IMM，所以 `ALUSrc_o`設1，反之設為0。而所有的運算都會將資料存在記憶體，`RegWrite_o`設定恆等於1。其中使用`Op_i[5]`判斷是否為 I-format。

### 3. ALU\_Control.v

稍微修改程式，以3bits的 funct3和7bits的 funct3為input，先以 ALUOp\_i判斷是否為 addi，接著用 funct3和 funct7分離各個運算式。每一個運算式assign 3bits不同的 ALUCtrl\_o。

### 4. Sign\_Extend.v

將原本 32bits 輸入只取前面 12bits 的 imm 重新 extend 到 32bits。

### 5. ALU.v

根據 ALUCtrl\_o 連結到的 ALUCtrl\_i 為 input，處理各個定義型別。

### 6. MUX32.v

根據 select\_i 決定輸出的值。

### 7. CPU.v

依照圖表填寫，input 端括弧內填透過宣告項(inst, inst\_addr, clk, start 等...)數值；output 端填入之後連結到的 input。