

Computer Architecture (CSIE3340)

Lab 1 Report

B10605023 生機四 詹育晟

一、開發環境

- **作業系統**：Windows 10 (WSL Ubuntu 環境)
- **模擬器**：Icarus Verilog (vvp) + GTKWave 3.3.75
- **執行方式**：透過 make run 指令執行模擬流程，由 Docker 容器進行自動化建構、模擬、比對輸出與 VCD 生成。
- **GTKWave 使用方式**：使用 gtkwave waveform_1.vcd 查看波形變化，點選左側 TestBench 展開後可手動加入觀察訊號至波形區。

二、模組功能實作說明

1. CPU.v

此模組整體控制主體，負責各個子模組之連接，整合 datapath 流程，包括：取指 (Instruction Fetch)、解碼 (Decode)、執行 (Execute)、回寫 (Write Back)。

- 透過 PC 模組取得目前指令記憶體位置。
- 解開 instruction 各欄位後餵給 Control、Registers、ALU_Control、Sign_Extend 等模組。
- 設計 MUX 以根據 ALUSrc 控制訊號，選擇 Register Read 或 Sign-Extend 的 Immediate 作為 ALU 的第二個輸入。
- 經由 ALU 執行計算，結果回寫至 Register File 中 rd 所指定之暫存器。

2. Control.v

本模組根據 opcode 判別指令類型，輸出 RegWrite、ALUSrc、ALUOp 三個控制訊號：

- RegWrite 控制是否將 ALU 結果寫入 Register File。

- ALUSrc 控制 ALU 的第二個輸入是否為 Immediate (R-type 為 0, I-type 為 1)。
- ALUOp 為兩位元訊號，輸入 ALU_Control 進行運算類型細分。

3. ALU_Control.v

本模組根據 ALUOp、funct3、funct7 判斷具體運算指令。

- ALUOp 00：加法（如 lw/sw）
- ALUOp 01：減法（如 beq）
- ALUOp 10：根據 funct3/funct7 決定 AND/OR/SUB/ADD 等 ALU 操作。

4. Sign_Extend.v

針對 I-type、S-type、B-type 等指令，將 12~13 位元的 Immediate 進行符號擴展 (Sign Extension) 為 32 位元。

- 模組會根據 opcode 自動選擇需要擴展的欄位區段。

5. ALU.v

執行實際運算（加減乘除與邏輯運算）。輸出包括：

- 運算結果 data_o
- 判斷是否為零 Zero_o，提供給分支控制使用。

6. MUX.v — 多工器模組 (Multiplexer)

MUX.v 是一個 2-to-1 的 32-bit 多工器 (Multiplexer)，它的主要功能是根據輸入的選擇訊號 select_i，在兩個 32-bit 資料來源之間選擇一個作為輸出。這個模組在 RISC-V 單循環 CPU 中的使用情境為 **決定 ALU 的第二個操作數來源**。