

計算機組織 HW1

1. Implement a 32-bits Complete ALU

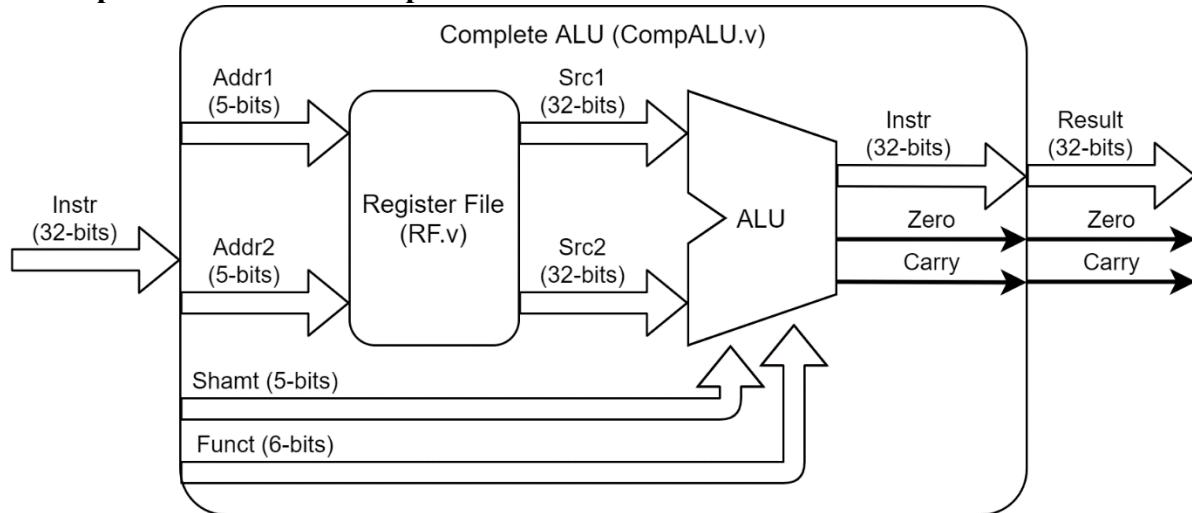


Figure 1: Path of Complete ALU

實作一位寬為 32 位元之 Complete ALU 模組，其內部由一 Register File 模組及一 ALU 模組組成。其 Register File 模組為 32 個 32 位寬唯讀暫存器組成。而 ALU 模組為 32 位寬算術邏輯單元。

2. 32-bits Read Only Register File

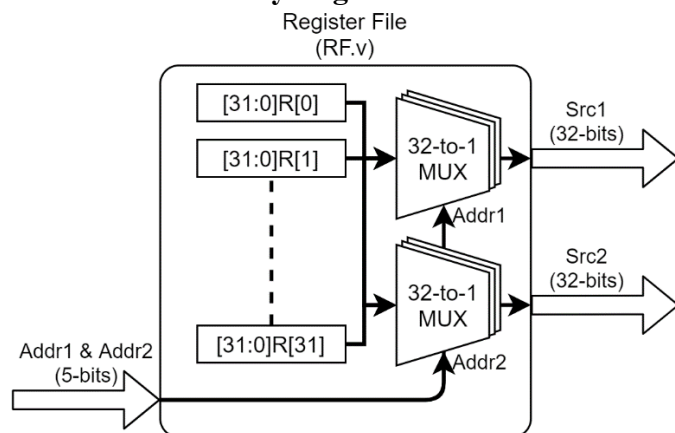


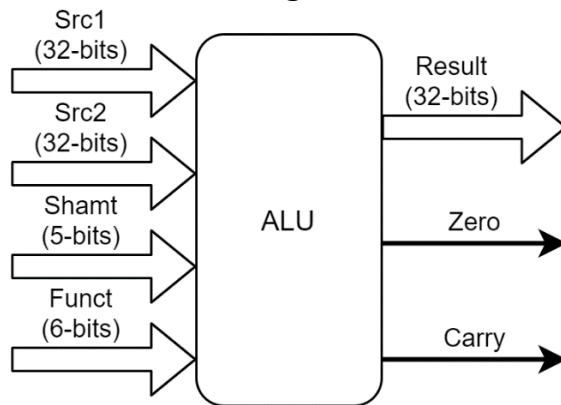
Figure 2: Path of Register File

I/O Interface

```
module RF (  
    input [4:0] Addr1,  
    input [4:0] Addr2,  
    output [31:0] Src1,  
    output [31:0] Src2  
);
```

本模組為模擬 MIPS 架構之暫存器，但由於本次作業無需實作寫入動作，故本模組只含讀取功能，其值於初始化時賦予。本模組內含 32 個 32 位寬之暫存器，由 5 位元作為定址。本模組提供兩路匯流排平行讀取動作，其記作“Src1”及“Src2”。各暫存器之初始值以 16 進制方式存於“testbench”文件夾內“RF.dat”文件內。模擬時由“tb_RF”或“tb_CompALU”根據文件對暫存器進行初始化。

3. 32-bits Arithmetic Logic Unit



I/O Interface

```

module ALU (
    input [31:0] Src1,
    input [31:0] Src2,
    input [4:0] Shamt,
    input [5:0] Funct,
    output [31:0] Result,
    output Zero,
    output Carry
);
  
```

Figure 3: I/O Interface of ALU

本模組為算術邏輯單元，負責對暫存器做相關運算，並輸出其結果及相關旗號。本模組輸入端含有兩路 32 位寬的資料匯流排，用以輸入相關暫存器資料，及一 6 位寬的控制匯流排，用以輸入相關動作的控制指令。其輸出端則為一 32 位寬資料匯流排以輸出運算結果，及兩個 1 位元的旗號，分別為“Zero”與“Carry”。Zero 旗號用於表示其運算結果是否為零，而 Carry 旗號則用於運算結果的進/借位旗號。本次作業需實作下表運算，由於本次作業無需寫入暫存器，故運算結果直接輸出於“Result”即可。

Instruction	Example	Meaning	Funct Code
Add unsigned	addu Result, Src1, Src2	Result = Src1 + Src2	001001
Subtract unsigned	subu Result, Src1, Src2	Result = Src1 - Src2	001010
And	and Result, Src1, Src2	Result = Src1 & Src2	010001
Shift left logical	sll Result, Src1, 10	Result = Src1 << 10	100001

Note: Shift 動作所需值存放於“Shamt”。

4. 32-bits Complete ALU

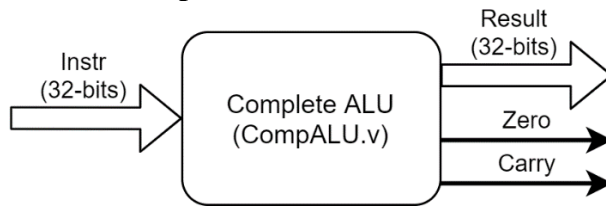


Figure 4: I/O Interface of Complete ALU

I/O Interface

```
module CompALU (
    input [31:0] Instr,
    output [31:0] Result,
    output Zero,
    output Carry
);
```

本模組由前二模組組成，以 32 位寬指令作為輸入，運算後輸出 32 位寬結果、Zero 旗號及 Carry 旗號。為模擬 MIPS 架構指令，故本次作業的指令格式如下：

OP Code	Source Register	Target Register	Destination Register	Shamt	Funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

OP Code: 用以表示執行的指令，本次作業設為與 Funct 相同。

Source Register: 用以表示需執行的第一暫存器。

Target Register: 用以表示需執行的第二暫存器。

Destination Register: 用以存放執行結果的暫存器，本次作業無需使用，可設為零。

Shamt: 用以表示“Shift”動作的次數。

Funct: 用以表示執行的指令，本次作業設為與 OP Code 相同。

Examples:

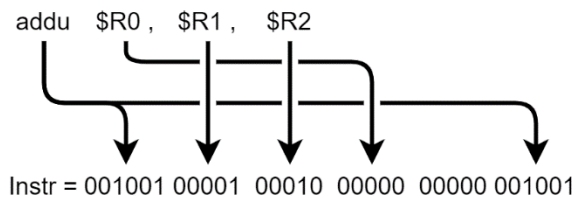


Figure 5: Example of Add Unsigned Instruction

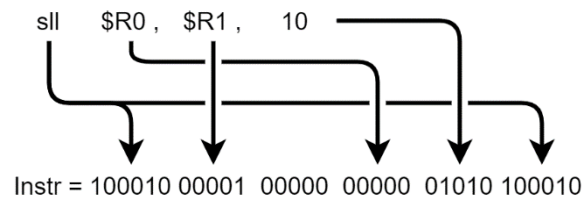


Figure 6: Example of Shift Left Logical Instruction

5. Functional Simulation

a. tb_RF

測試平台“tb_RF”首先會根據“testbench/RF.dat”初始化暫存器，然後順序輸入地址於 Addr1，逆序於 Addr2。其輸出波形（16 進制）將類似於下圖，應截圖並說明於報告之中。

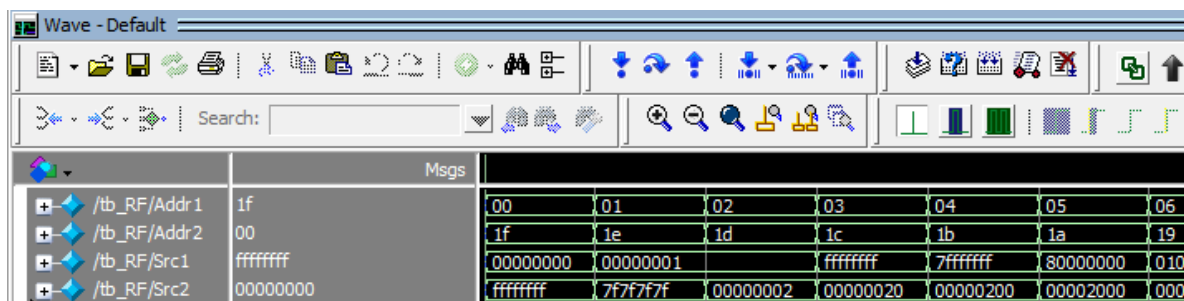


Figure 7: Waveform of tb_RF

b. tb_ALU

測試平台“tb_ALU”會初始化輸入訊號為零，等待一段時間後將逐行執行“testbench/tb_ALU.in”內指令。其輸出波形（16 進制）將類似於下圖，應截圖並說明指令與結果於報告之中。

Note: “tb_ALU.in” 只提供 “sll” 範例，應自行添加測試指令並說明於報告書。

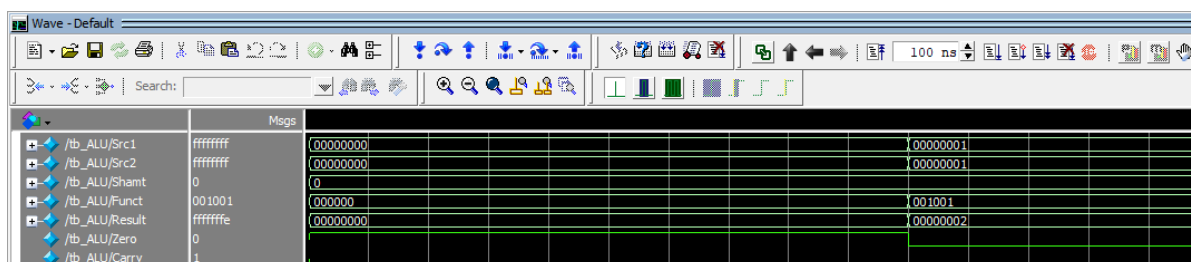


Figure 8: Waveform of tb_ALU

c. tb_CompALU

測試平台“tb_CompALU”會初始化輸入訊號為零以及暫存器，等待一段時間後將逐行執行“testbench/tb_CompALU.in”內指令。其輸出波形（16 進制）將類似於下圖，應截圖並說明指令與結果於報告之中。

Note: “tb_CompALU.in” 只提供各運算範例，可自行添加測試指令並說明於報告書。

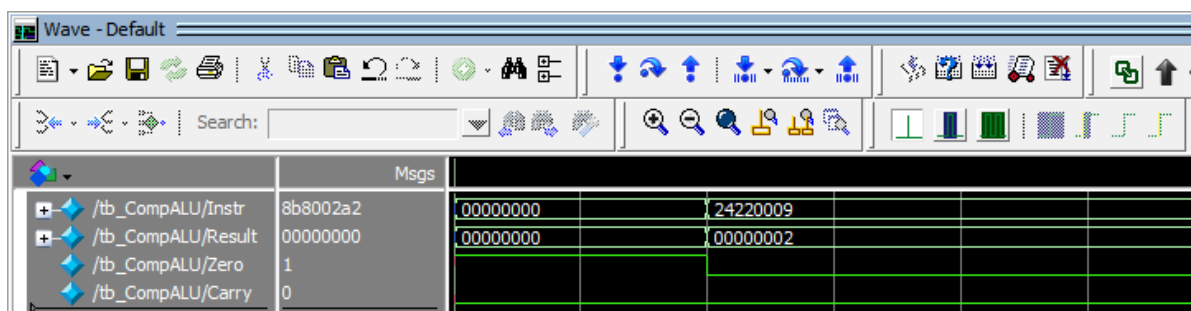


Figure 9: Waveform of tb_CompALU

6. Submission

報告書：

- a. 封面。
- b. 各模組程式碼截圖並說明。
- c. 各模組測試指令 (.in 檔) 截圖並說明。
- d. 各模組測試結果 (16 進制波形) 截圖並說明。
- e. 作業總結與心得。
- f. 存為 PDF 檔，並以學號命名——“B10XXXXXX.pdf”。

壓縮檔：將所有模組的.v 檔 (RF.v、ALU.v、CompALU.v)、所有測試 .in 檔 (tb_ALU.in、tb_CompALU.in) 及報告書放於資料夾中，並壓縮成檔案 B10XXXXXX.zip。

評分：

- a. RF：需完整輸出 RF.dat 內容，共 20 分。
- b. ALU：以另外產生 testbench 進行測試，各 Instruction 5 分，共 20 分。
- c. CompALU：以另外產生 testbench 進行測試，各 Instruction 5 分，共 20 分。
- d. 報告書：共 40 分。
- e. 抄襲則以零分計算。

繳交時間：110/03/18 13:00 前上傳至 Moodle。