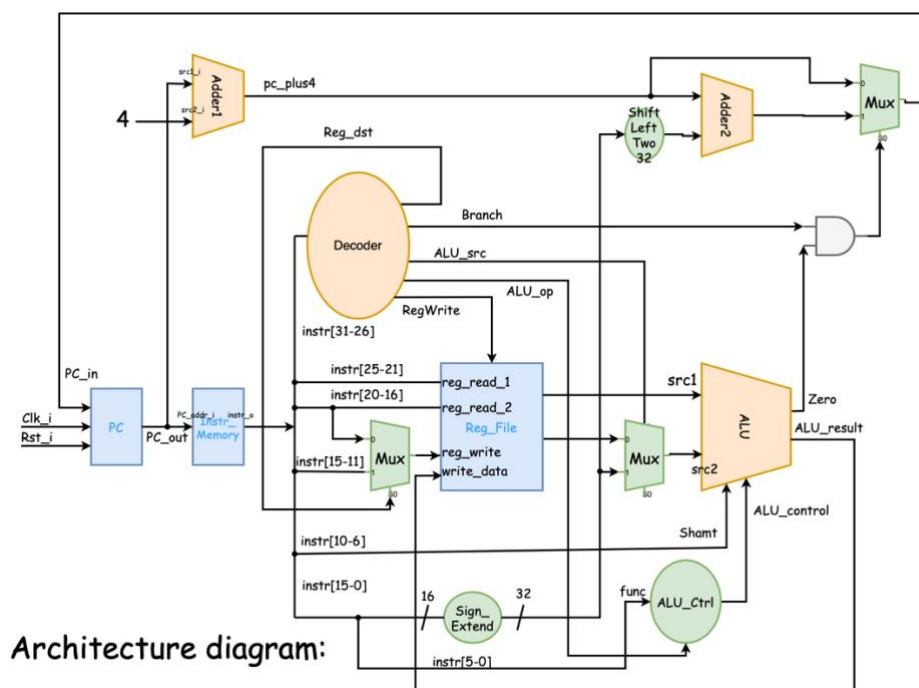


Computer Organization Lab 2

Simplified Single-cycle CPU

0716055 王耀德

- Architecture diagram:

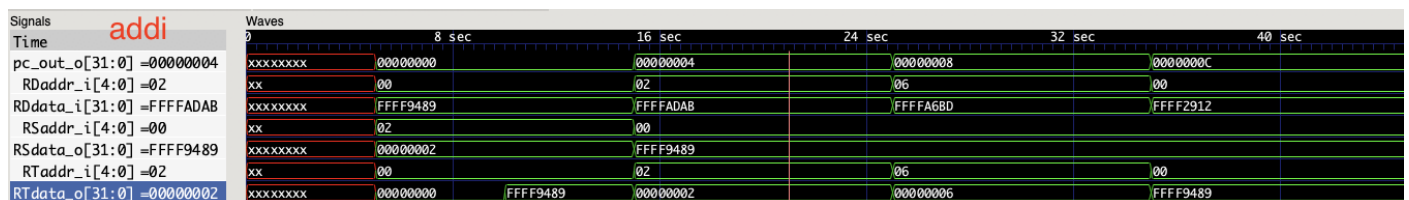


- Module description:

1. Adder：對 PC 作加法 (PC+4 (Adder1) 或 branch (Adder2))。
2. ALU_Ctrl：控制 ALU 要做什麼運算。
3. ALU：對傳入的資料執行各種運算(or, add, shift ...)。
4. Decoder：透過 Instr_memory 傳來的 op_code 控制 Reg_write、Reg_dst、ALU_src 與 ALU_op，決定要執行的操作。
5. Instr_Memory：將 32-bit Instruction 分別傳進各個 module。

6. Mux_2to1(write_reg): 判斷 src1 是否要 write。
7. Mux_2to1(ALU_src): 判斷是 reg2 或 sign-extend 的部分進入 ALU。
8. Mux_2to1(PC_source): 判斷 PC 接下來是 PC+4 或是 Branch 的結果。
9. ProgramCounter: 紀錄現在指令在 Instr_Memory 的位置。
10. Shift_Left_two_32: 對 Sign_Extend 後的 32-bit 左移 2 位達到 x4 的效果。
11. Sign_Extend: 把 Instr[15-0] 做 sign extension 到 32-bit。
12. Single_Simple_CPU: 把上述的 module 依照目標連接起來。

- Waveform:



- Questions:

1. What is the difference between "input [15:0] input_0" and "input [0:15] input_0" inside the module?

input [15:0] input_0 的 index 是降冪排列，依序由 input_0[15] 排列至

input_0[0]，而 input [0:15] input_0 則相反，是升冪排列。

2. What is the meaning of "always" block in Verilog?

"always" block 表示當()內的訊號改變時，會執行 block 內部的指令。

3. What are the advantages and disadvantages of port connection by order and port connection by name in Verilog?

by order：優點是可以打比較少字，缺點是如果 module port 很多或是做過修改，不容易閱讀且比較難找出新增或移除的參數。

by name：優點是清楚知道參數對應的值是什麼，缺點是要打比較多的字。

- Contribution:

我負責寫 Decoder 還有測試 testbench 與 debug，我的組員敘溶負責寫其餘 module，並一起完成 ALU、連接各 module。

- Discussions:

一開始時有點無從下手，尤其是有些指令沒有看過，仔細了解過後可以發現只要分清楚是 r-type 還是 i-type、要 branch 或是 load 之後，就比較知道怎麼下手了。測試時在 sra 和 srav 的 r6 結果卡比較久。後來發現會被辨識成 unsigned 的表示，所以就改成放進陣列，就成功表示出 shift right arithmetic 了。這次寫起來因為不用像 lab1 用 1bit ALU 慢慢組合，寫起來比較不會有憤怒的感覺。