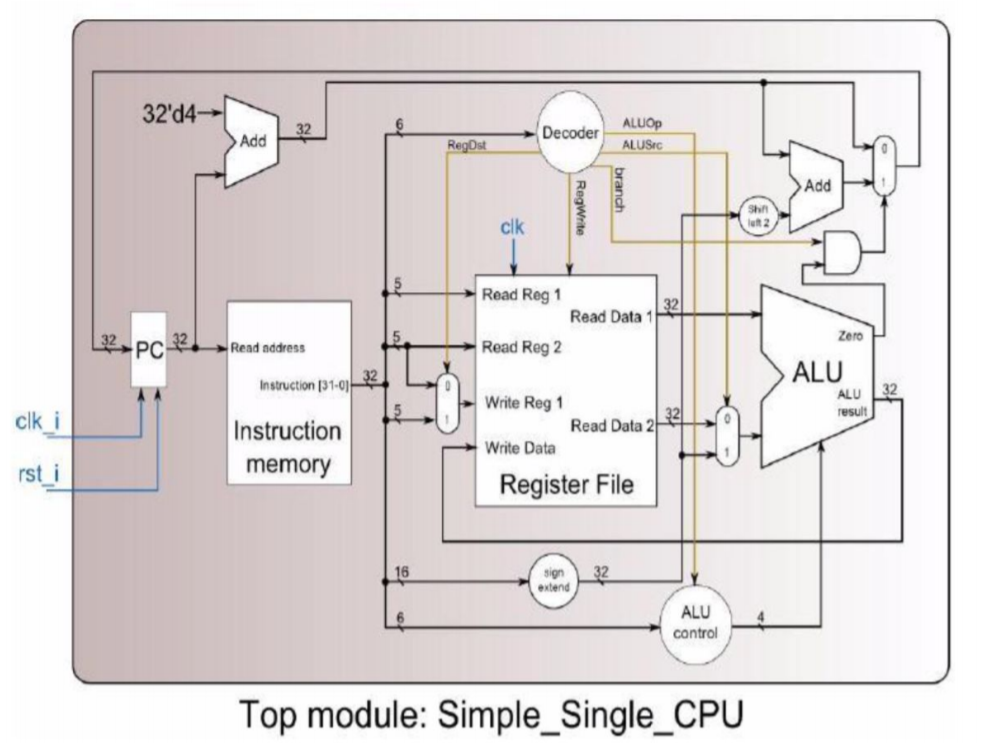
**Computer Organization**

**Architecture diagrams:**

****

**Hardware module analysis:** ModelSim

**Finished part:**

Adder.v : 實作加法就好

ALU.v : 實作ALU，可以直接參考PDF的Appendix

ALU\_Ctrl.v : 決定operation

Decorder.v : 將instruction轉變成實作用的code

MUX\_2to1.v : 套if…else or ... ? ... : ... 即可

Shift\_Left\_Two\_32.v : 將所有位數左移2即可

Sign\_Extend.v : 重複16次MSB 再concatenate input value

Simple\_Single\_CPU.v : 將所有小程式串起來，宣告及填完相應的參數 即可。

CO\_P2\_test\_data1.txt CO\_P2\_test\_data2.txt

r0= 0 r0= 0

r1= 10 r1= 1

r2= 4 r2= 0

r3= 0 r3= 0

r4= 0 r4= 0

r5= 6 r5= 0

r6= 0 r6= 0

r7= 0 r7= 14

r8= 0 r8= 0

r9= 0 r9= 15

r10= 0 r10= 0

r11= 0 r11= 0

r12= 0 r12= 0

**Problems you met and solutions:**

ModelSim常常當機，而且在路徑中不能出現中文，所以我只能把檔案移離桌面。display的內容一直無法出現，這樣導致很難debug，然後不停地重灌還是沒有用，只好在家裡的電腦上做。

一開始路徑弄錯導致一直無法simulate，後來才發現路徑問題會導致txt無法被讀取，也才發現會多個CO\_P2\_Result.txt。

Wire跟reg的差異還是不太清楚，還有光是理解一開始提供的檔案就花了很多時間，而Decorder.v的部分我也弄了很久，畢竟它是CPU的核心部分，解碼錯會導致它後面的程序完全不對，幸好後來好好試試跟想想後就弄好了。

最困難的是把所有的小程式併在一起，也就是Simple\_Single\_CPU.v，只要有一個沒弄好，輸出全部都是X，要花很多時間思考研究電路的連接，然後宣告跟填上相對應的參數，一開始參數的bits number弄錯害我一直沒過，而且有時候想錯參數就會擺錯位置，完全弄懂CPU後參數才放對，也才能看到輸出。

**Summary:**

這次只要把每個小部分完成，project就大致做完了，了解到CPU是由許多個小部分所組成。完成每一小份不難，但把他們合在一起就很有難度，整併往往是很棘手的，而且還要注意wire跟reg的差異、用法。先弄懂CPU的實作後，比較困難的只有接線填參數(Simple\_Single\_CPU.v)和解碼(Decorder.v)，還有要記得讀寫檔的路徑問題。

在Lab2中，我更加理解CPU，也更了解CPU的實作方法，還有對verilog的語法更熟悉了，不僅是一些語法還有一些module的概念、讀寫檔的實作，以及一些時序問題。或許不如C語言好理解，但能用軟體模擬硬體，這已經提供不小的便利性了。