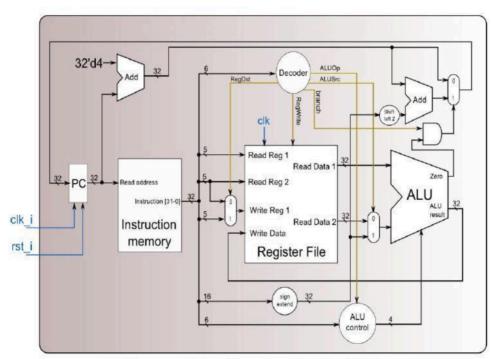
# **Computer Organization**

0516013 吳泓寬

## **Architecture diagrams:**



Top module: Simple\_Single\_CPU

(取自 CO\_Lab\_2.pdf)

## Hardware module analysis:

運作流程十分簡單,一開始時會把 instruction 全部放入 instruction memory 中,然 後取出 PC 的值去 instruction memory 拿指令,接下來再由 decoder 去做第一層的解析算出 ALU\_op 和一些作為 MUX 的選擇的值(ALUSrc / RegWrite / RegDst / Branch ),而 R-format(ALU\_op=3'b000)則須由 funct 的部分進一步求出 ALU control(如下表)

instruction	ALU_op	funct	ALU_control	ALUSrc	RegWrite	RegDst	Branch
ADD		32	0010				
SUB		34	0110				
AND	0	36	0000	0	1	1	0
OR		37	0001				
SLT		42	0111				
ADDI	8	Х	0010	1	1	0	0
SLTI	10	X	0111	1	1	0	0
BEQ	4	X	0110	0	0	0	1

解析完後便可以由 ALU 計算出結果,基本上就是按照那些 control 的值(由 MUX 去選擇)去執行。

而 PC 的部分基本上就是會有個 mux 根據 branch & zero(代表 beq 的 rs 跟 rt 相等) 是否成立,來看是要走 branch 還是 pc+4,然後選好 pc\_next 後就等待下次的 pos edge 再將 pc next 賦值至 pc now,然後重複執行至全部指令執行完。

#### **Finished part:**

1. Adder 直接拿 ALU module (control=4'b0010)來實作 2. ALU control 由 ALU op 和 funct 來決定 ALU control 3. alu top/ALU Lab1 的部分 4. Decoder 由 instruction 的 op entry 去決定 ALU op 由 select 來決定是取前者還是後者 5. MUX 2to1 6. Shift Left Two 以 data-flow 方式[<<2]完成 7. Sign Extend 將前 16bits 接上 sign-bit 8. Simple Single CPU 依據 architecture diagram 將全部 module 組在一起

#### **Problems you met and solutions:**

Compile 完後執行出現 \$readmemb(CO\_P2\_test\_data1.txt): Not enough words in the file for the requested range [0:31] 以為是出錯,結果發現是因為 instruction memory 是 32\*32bits 然後測資的指令沒有到 32 個,所以才出現這個 warning。

### **Summary:**

經過這次作業對 CPU 又有了更近一步的認識,基本的架構也掌握到了,比較有疑問的地方可能是 ALU op 為什麼要 3bits,也許是之後作業會出現其他指令吧。

#### **Results:**

<CO\_P2\_test\_data1.txt>

####