

instruction	ALU_op	funct	ALU_control	ALUSrc	RegWrite	RegDst	Branch
ADD	0	32	0010	0	1	1	0
SUB		34	0110				
AND		36	0000				
OR		37	0001				
SLT		42	0111				
ADDI	8	x	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 |
| 5. MUX_2to1 | 由 select 來決定是取前者還是後者 |
| 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 又有了更進一步的認識，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>

```
maxam@maxam ~/Desktop/CO_lab2_2/code: iverilog -o test *.v
maxam@maxam ~/Desktop/CO_lab2_2/code: vvp test
WARNING: Instr_Memory.v:34: $readmemb(CO_P2_test_data2.txt): Not enough words in the file for the requested range [0:31].
** VVP Stop(0) **
** Flushing output streams.
** Current simulation time is 260000 ticks.
> finish
** Continue **
maxam@maxam ~/Desktop/CO_lab2_2/code: cat CO_P2_Result.txt
r0= 0
r1= 1
r2= 0
r3= 0
r4= 0
r5= 0
r6= 0
r7= 14
r8= 0
r9= 15
r10= 0
r11= 0
r12= 0
```

<CO_P2_test_data2.txt>

```
maxam@maxam ~/Desktop/CO_lab2_2/code $ iverilog -o test *.v
maxam@maxam ~/Desktop/CO_lab2_2/code $ vvp test
WARNING: Instr_Memory.v:34: $readmemb(CO_P2_test_data1.txt): Not enough words in the file for the requested range [0:31].
** VVP Stop(0) **
** Flushing output streams.
** Current simulation time is 260000 ticks.
> finish
** Continue **
maxam@maxam ~/Desktop/CO_lab2_2/code $ cat CO_P2_Result.txt
r0=      0
r1=     10
r2=       4
r3=       0
r4=       0
r5=       6
r6=       0
r7=       0
r8=       0
r9=       0
r10=      0
r11=      0
r12=      0
```