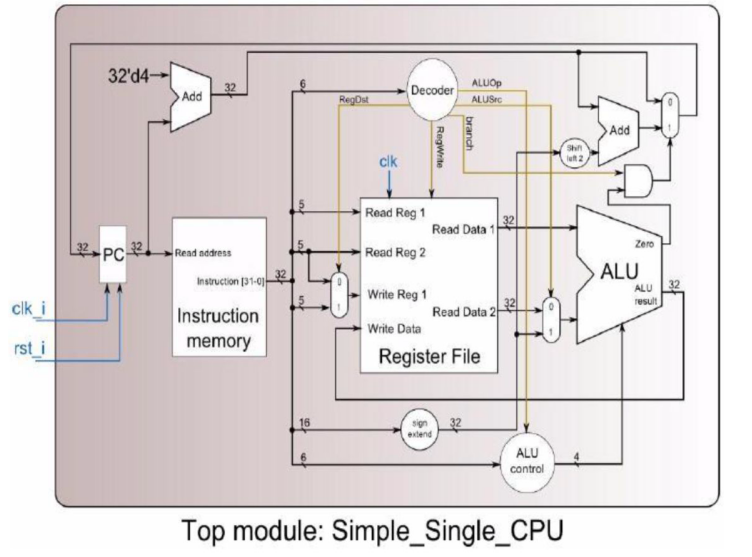
**Computer Organization**

0310020 黃百川

**Architecture diagrams:**



**Hardware module analysis:**

Designed Decode form

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | op | funct | RegDst | RegWrite | branch | ALUSrc | ALUOP[2] | ALUOP[1] | | ALUOP[0] |
| add | 000000 | 100000 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | |
| sub | 000000 | 100010 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | |
| and | 000000 | 100100 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | |
| or | 000000 | 100101 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | |
| slt | 000000 | 101010 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | |
| addi | 001000 | 000000 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | |
| slti | 001010 | 000000 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | |
| beq | 000100 | 000000 | x | 0 | 1 | 0 | 1 | 0 | 1 | |

Rformat = ~op[5] & ~op[4] & ~op[3] & ~op[2] & ~op[1] & ~op[0]

Beq = ~op[5] & ~op[4] & ~op[3] & op[2] & ~op[1] & ~op[0]

Slti = ~op[5] & ~op[4] & op[3] & ~op[2] & op[1] & ~op[0]

Immediate = ~op[5]& ~op[4] & op[3] // designed for Immediate type

RegDst = Rformat

ALUSrc = Immediate

RegWrite = Immediate | Rformat

Branch = Beq

ALUOP[2] = Immediate | Beq

ALUOP[1] = Slti | Rformat

ALUOP[0] = Slti | Beq

主要設計的部分是綠色部分，這樣的設計是希望能在符合要求的情況下，盡量減少更改原先投影片上的設計。

**Finished part:**

1. Mux.v

2. ALU.v

3. Adder.v

4. Sign\_Extend

5. Shift\_left2

6. ALU\_ctrl

7. Decoder

8. Simple\_single\_cpu

**Problems you met and solutions:**

這次Lab主要是根據不同的指令要求來重新設計CPU的Decoder跟ALU\_Ctrl的部分。由於一開始沒注意到指令的要求不同，所以根據課堂上的下去設計，後來利用Modelsim內建的wave來偵錯。

**Summary:**

這次Lab有簡化Adder跟ALU的要求，不用針對每個bit下去做運算，但沒有overflow的偵測。