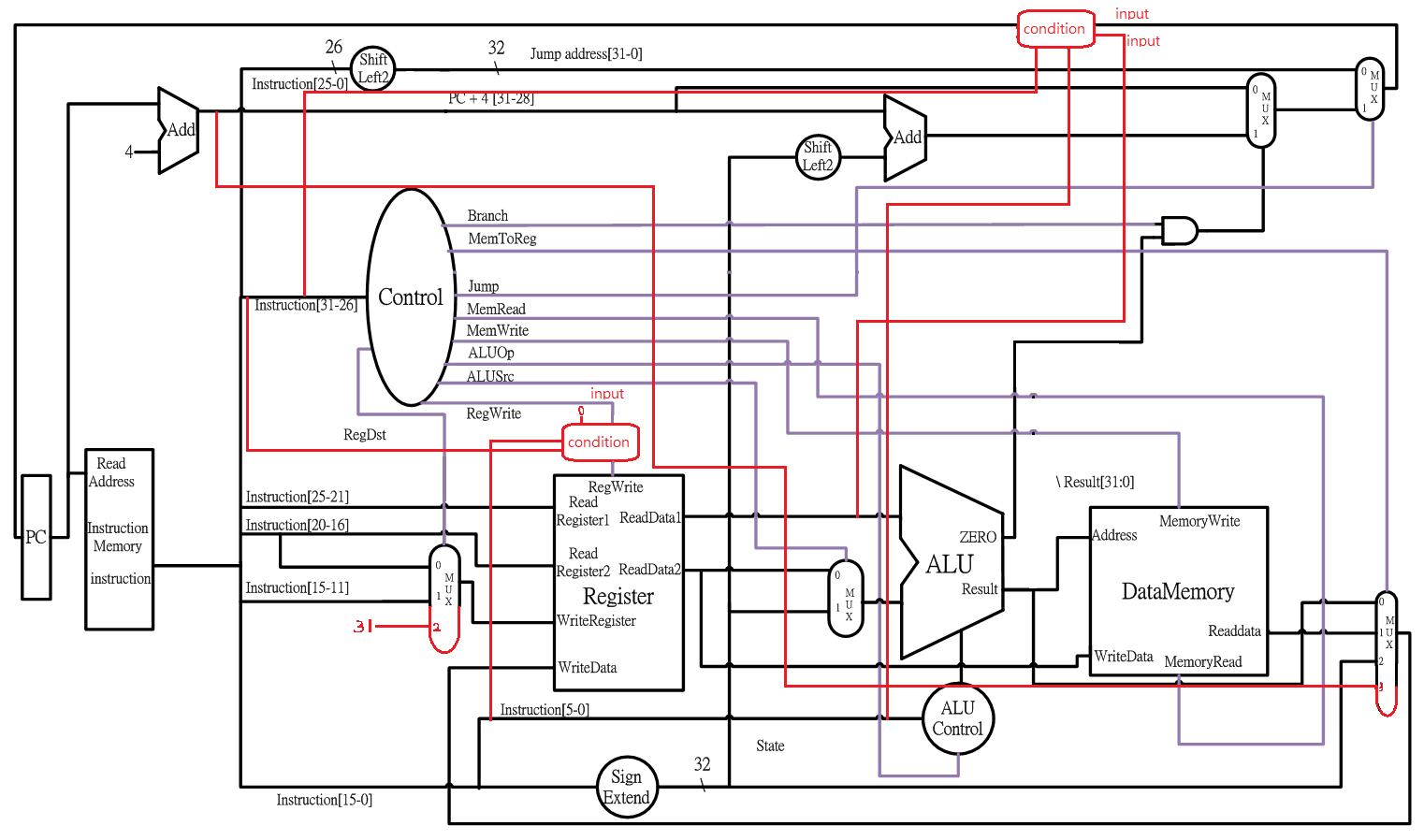
**Computer Organization**

**0516303 余采彧**

**0516208 黃郁恬**

**Architecture diagrams:**

**修改部分: 用紅色標示**

****

**Hardware module analysis:**

* **Simple\_Single\_CPU**

**按照上圖電路去接線**

* **Decoder:**

**R-format: 6'b000000，BEQ: 6'b000100**

**ADDI: 6'b001000，SLTI: 6'b001010**

**Lw: 6'b100011，Sw: 6'b101011**

**Jump: 6'b000010，Jal: 6'b000011**

**X: 表示don’t care**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **R-format** | **BEQ** | **ADDI** | **SLTI** | **Lw** | **Sw** | **Jump** | **Jal** |
| **RegWirte** | **1** | **0** | **1** | **1** | **1** | **0** | **0** | **1** |
| **ALU\_op\_o** | **010** | **001** | **000** | **011** | **000** | **000** | **X** | **X** |
| **ALUSrc\_o** | **0** | **0** | **1** | **1** | **1** | **1** | **X** | **X** |
| **RegDst\_o** | **01** | **X** | **00** | **00** | **00** | **00** | **X** | **10** |
| **Branch\_o** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** |
| **Jump\_o** | **1** | **1** | **1** | **1** | **1** | **1** | **0** | **0** |
| **MemRead\_o** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **1** |
| **MemWrite\_o** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** |
| **MemtoReg\_o** | **00** | **00** | **00** | **00** | **01** | **X** | **X** | **11** |

* **Shift\_Left\_Two\_32:**

**assign data\_o = data\_i<<2;**

**把輸入想左移兩個bit(<<)，賦值給輸出。**

* **Sign\_Extend:**

**data\_o = {{16{data\_i[15]}}, data\_i};**

**把輸入的前16個bit(0~15)複製給輸出的前16個bit，並把輸入的sign bit(也就是第15個bit: msb)複製給輸出的後16個bit。**

* **Adder:**

**assign sum\_o = src1\_i + src2\_i;**

**把輸入的兩個運算元做相加。**

* **Mux\_2to1:**

**assign data\_o= (select\_i ? data1\_i : data0\_i) ;**

**根據輸入的select\_i判斷:**

**🡪select\_i 為1，輸出就等於輸入的data1\_i**

**🡪select\_i 為0，輸出就等於輸入的data0\_i**

**(MUX\_3to1, MUX\_4to1, Chooser也是相同概念)**

**(Chooser: 圖中綠色框框處)**

* **ALU\_Ctrl:**

**case(ALUOp\_i)**

**0: ALUCtrl\_o=4'd2;**

**1: ALUCtrl\_o=4'd6;**

**2: begin**

**ALUCtrl\_o[0]= funct\_i[0] || funct\_i[3];**

**ALUCtrl\_o[1]= (!funct\_i[2]);**

**ALUCtrl\_o[2]= funct\_i[1];**

**ALUCtrl\_o[3]= 1'b0;**

**end**

**3: ALUCtrl\_o=4'd7;**

**endcase**

**根據Decoder的ALU\_op\_o:**

**ALU\_op\_o為0時:**

**ADDI，ALUCtrl\_o設為2，也就是add(0010)。**

**ALU\_op\_o為1時:**

**BEQ，ALUCtrl\_o設為6，也就是sub(0110)，因為beq就是要做相減的動作，看結果是否為0來判斷兩運算元是否相等。**

**ALU\_op\_o為2時:**

**R-format，ALUCtrl\_o進階透過function判斷(畫真值表)。**

**ALU\_op\_o為3時:**

**SLTI，ALUCtrl\_o設為7，也就是做slt(0111)。**

* **ALU:**

**同上次作業，只是參數的地方稍微調整成這次lab的格式。**

**(因此還必須引入alu\_top)**

**Problems you met and solutions:**

1. **看不太懂Spec的描述，導致不曉得怎麼做**
   * **Sol: 重複看好幾次，加上參考上課投影片**
2. **讀檔案讀不進來**
   * **Sol: 按照之前助教提供在討論區的解決辦法(一步一步按著做)，且修改成絕對路徑**

****

1. **Jump出來的結果有誤**
   * **Sol: 解碼給的input檔案，trace位址**

**Summary:**

**心得**

**真是一次比一次刺激啊!**

**Division of work:**

**0516303:**

**Simple\_Single\_CPU**

**Adder**

**MUX\_3to1**

**MUX\_4to1**

**0516208:**

**Chooser**

**Sign\_extend**

**Shift\_Left\_Two\_32**

**Decoder**

**Report**

* **ALU, alu\_top: 之前Lab1的 (0516303, 0516208)**