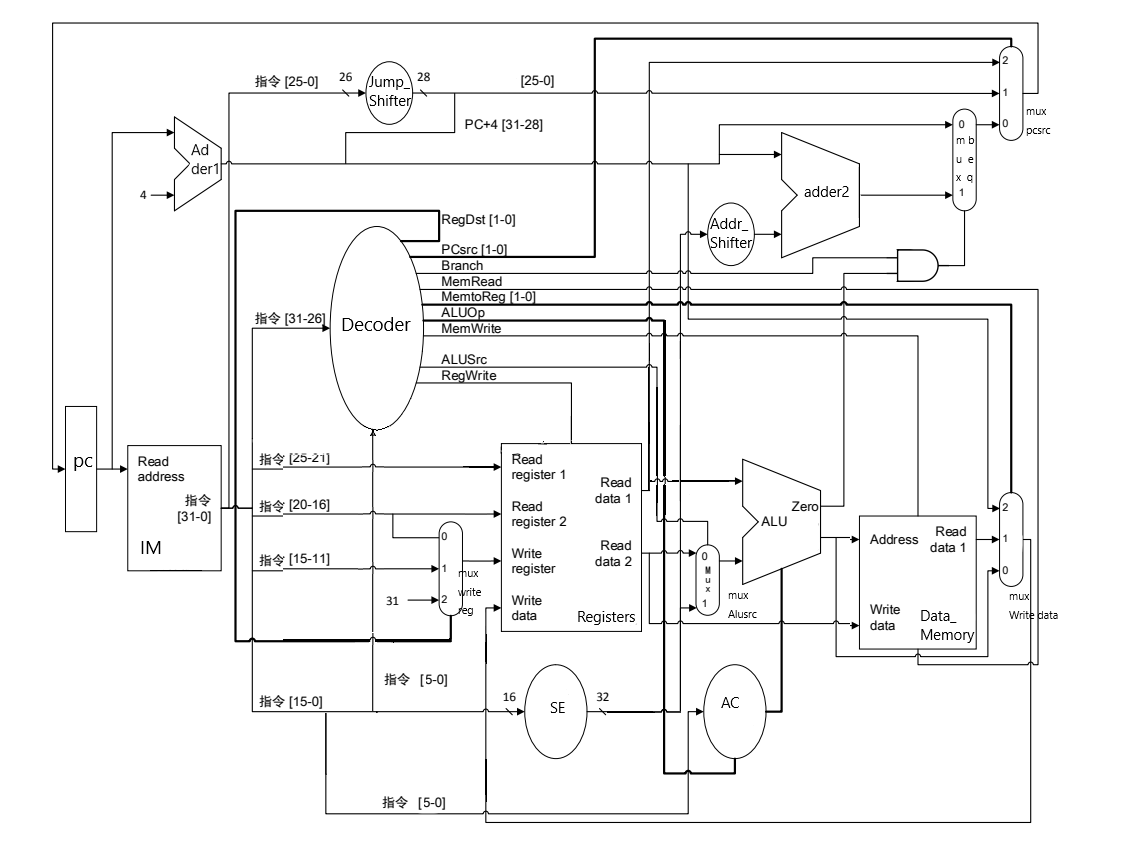
**Computer Organization**

**Architecture Diagram**

****

**Hardware module analysis(簡短解釋一下每個module的功能，以及在CPU裡扮演的角色):**

**ProgramCounter.v**

|  |
| --- |
| **一開始設定起始地址為零，並隨著clk更新出下一個的地址。藉由clk更新機制確保一個指令執行完下一個指令才會被讀取。** |

**Instr\_Memory.v:**

|  |
| --- |
| **根據PC給出的地址從txt(mem)中提取指令。** |

**Adder1**

|  |
| --- |
| **把現在地址+4，以算出下一個地址候補。** |

**Mux\_PC\_Source**

|  |
| --- |
| **根據PCsrc決定要用哪個地址。0為beq or pc+4；1為j、jal指令計算的地址值；2為jr指令計算的地址值。** |

**Reg\_File.v:**

|  |
| --- |
| **根據RF給出的指令提取出指定reg的數值，並隨著clk跟RegWrite的真偽決定是否更新reg的數值** |

**ALU\_Ctrl.v**

|  |
| --- |
| **根據decoder產生的ALUOp跟指令的function code決定要輸入給ALU的alu\_option。** |

**Mux\_Write\_Reg**

|  |
| --- |
| **根據Decoder產生的Regdst決定使用rt還是rd還是31($ra)當作write reg。不同指令的write reg在不同的位置。** |

**Decoder.V**

|  |
| --- |
| **ID Stage。根據instruction的opcode決定RegWrite、ALU\_OP、ALUSrc、RegDst、Branch。他們分別會影響是否更新reg、ALU\_oprion的數值、ALU的input2、write reg的值、是否跳轉到另一個地址。** |

**Sign\_extend.v**

|  |
| --- |
| **根據輸入的數值的正負，決定補到32位時是要補1還是0。避免16bit數值在進入32bit alu之前因為擴充位元而正負號改變。** |

**ALU.v**

|  |
| --- |
| **根據輸入的的alu\_option，決定如何處裡輸入的兩個數值。輸出結果跟是否為0出來。** |

**Mux\_ALUSrc.v:**

|  |
| --- |
| **根據輸入的的alu\_src，決定輸入ALU的數值。因為I、Rtype指令需要處裡的數值分別來自reg或是sign\_extend。** |

**Addr\_shifter**

|  |
| --- |
| **因為beq讀取出的地址需要\*4得到真正的位移數值，所以左移2。** |

**Adder2**

|  |
| --- |
| **EXE stage。把現有地址+4跟跳轉數值相加，決定候補地址2。** |

**Jump\_shifter**

|  |
| --- |
| **因為J、JAL的地址[27:0]需要\*4得到真正的絕對地址，所以左移2。** |

**Data\_Memory**

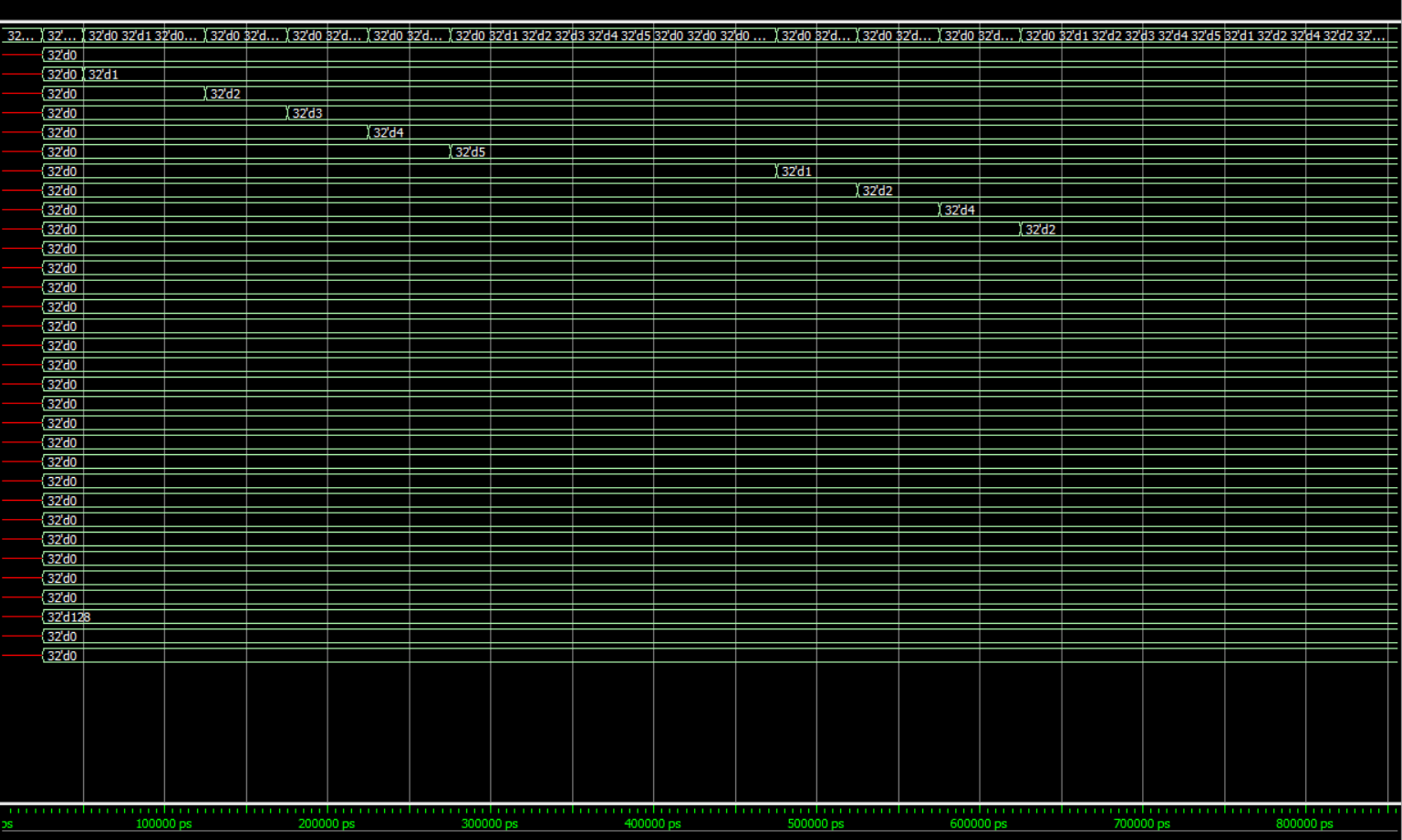
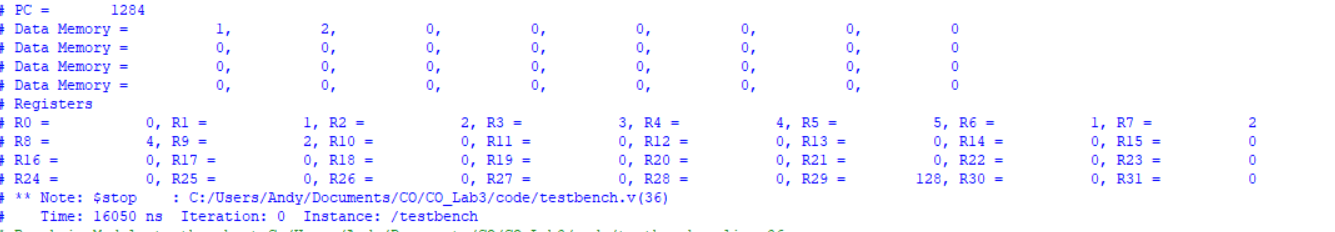
|  |
| --- |
| **根據decoder的memread及memwrite值，決定要從記憶體讀出or寫入從write\_data進來的資料。** |

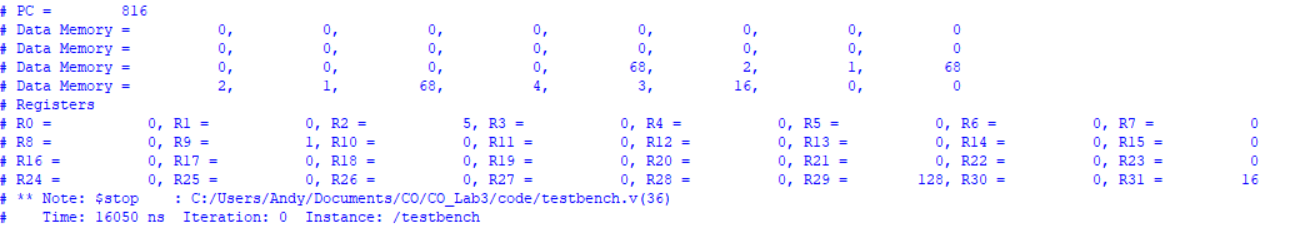
**Mux\_beq**

|  |
| --- |
| **根據branch值跟ALU 的zero決定進入mux\_pcsrc的0號地址為beq or pc +4。** |

**RESULT:**

**Test1**

****

**Test2**

**Problems you met and solutions:**

**Problem1:檔案讀不到**

**Solution:改成絕對地址**

**Promblem2: beq指令壞掉**

**Solution: 發現是ALU中的zero\_o打成zero\_0**

**Promblem3: jr指令壞掉**

**Solution: 發現是wire線接錯，把{pc+4[31:28],jump\_shifter}的值放到了mux\_PCsrc的date1\_i、date2\_i中，所以無法順利跳轉**

**Summary:**

**逐漸習慣硬體語言。**

**只犯了一點小錯誤，例如:**

**接錯一次wire**

**Debug速度大幅提升，清楚知道出了甚麼問題要看哪個wave。**