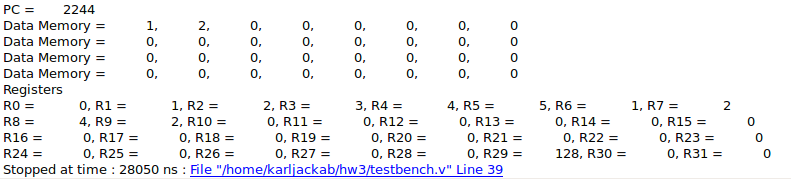
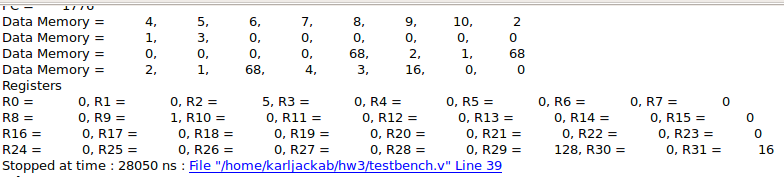
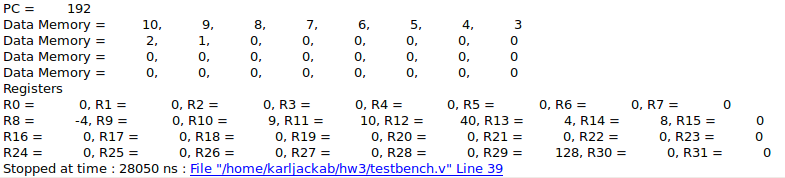
**Computer Organization**

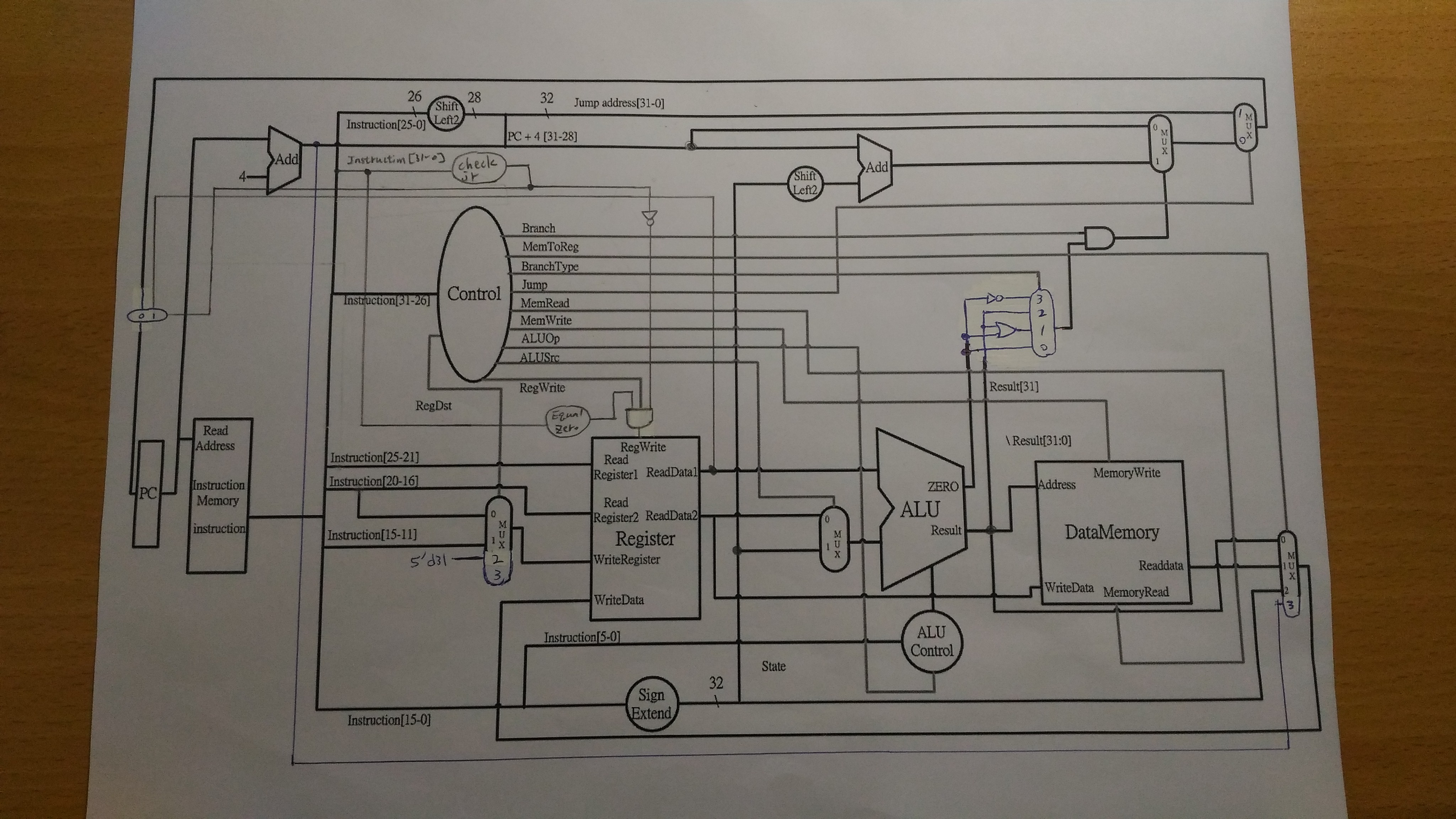
**HDL simulator:**

***Basic Instruction***

***Advanced set 1***

***Advanced set 2***

**Architecture diagram:**

****

**Detailed description of the implementation:**

一開始比較費力的就是加了一些module和Decoder接出來的wire，做出來之後Basic instruction就填上去應該在的地方就好了。

Advance 1的重點有兩個：

一個是jal要儲存地址，因此需要把add 4完後的PC接到最後的MUX，讓MemToReg控制MUX來選擇寫入的資料，還要在Register的WriteRegister前面加一個4 to 1的MUX，讓port 2接 31 這個數字，代表stack point，port 3就先放著不用。

另一個是jr因為是R type，但要做的事跟平常的R type實在差非常多，因此我再接了一個check jr的 module來控制跳轉與存儲之類的動作。

最麻煩的就是Advance 2了：

接好ble,bnez,bltz並不難，只要想好brach type那邊的MUX要怎麼做就好，比較麻煩的是要自己寫machine code，讓我好好的複習了各個指令的格式，還花好久時間de一個因為verilog時序關係而出現的bug。

**Problems encountered and solutions:**

Basic instruction沒有太大的困難，Advance 1稍微想了一下也很快就做出來了

花最久時間的是Advance 2

寫machine code查了網路資源和跟同學討論才確定好正確版本，很麻煩的一個點就是發生錯誤都要再檢查到底是machine code錯還是verilog code錯。

另外就是一個時序問題的bug，反覆確認machine code都沒錯時才認命的去把一些可能會錯的wire print出來，才找到問題，花了非常多時間。

**Lesson learnt (if any):**

一、各個指令的格式

二、實際一行一行trace指令執行後，register與memory的資料變化

三、verilog時序的問題