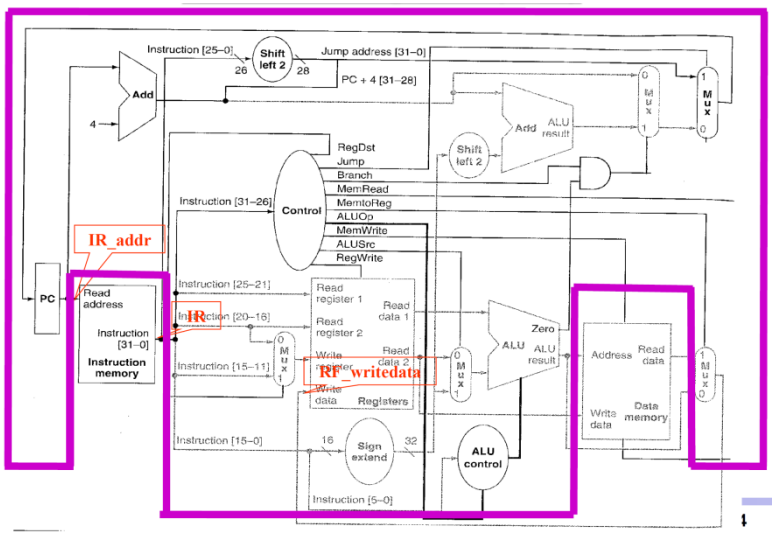
Single MIPS Report

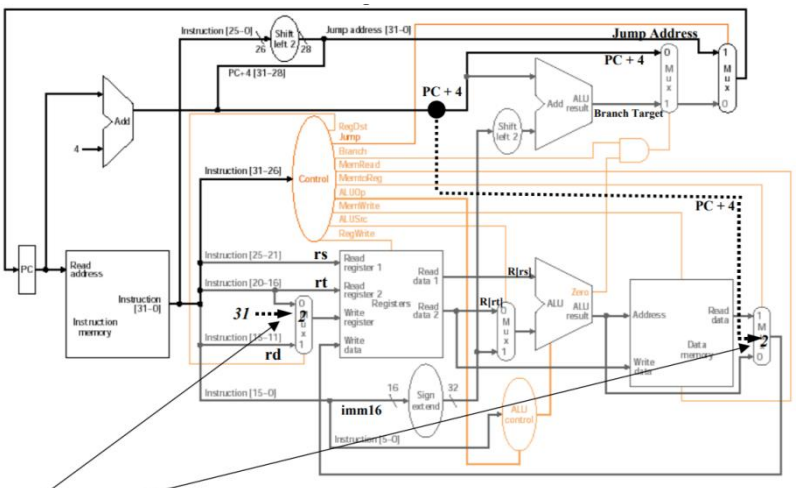
姓名：鐘民憲　　學號：B06901017　　系級：電機二

一、硬體設計

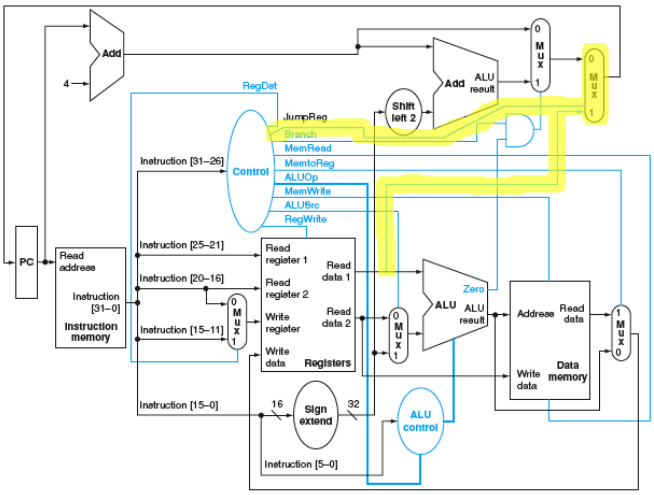


硬體基本上就是按照此圖設計，不過由於作業要求 support 的 instructions 包含 R-type(add, sub, and, or, slt)、lw, sw、beg 以及 j, jal, jr，而此設計並沒有辦法執行 jal 與 jr，故做了以下兩個修改：

jal：(control signal also need to consider this condition)



jr：(control signal also need to consider this condition)



與圖片稍有不同，我設計的方式是將 MUX_Jr 與 MUX_Jump 合併在一起，變成一個 3-to-1 的 MUX，然後 control signal “Jump”也改成 2-bits

二、程式設計

Verilog 程式的部分我分成兩個 module 去寫，一個是 SingleCycle_MIPS，基本上囊括了所有東西，另一個是 register_file，原本的寫法是直接將 register_file 另成變數 reg [31:0] register [31:0](reg 的陣列)，但是 simulation 發現一直會出現 error: virtual memory limit exceeded，於是就將它獨立出來寫成一個 block，雖然基本上是相同的東西，但是這樣就不會出現 error 了。

此外，我一開始在寫的時候將所有的變數都令為 reg 並把所有 combination circuit 都寫在 always @(*)裡面，想說這樣 code 看起來比較一致，感覺也沒什麼問題，但是我發現這樣做程式執行會出錯，在

看 nWave 時明明就是直接連接的兩個 reg，理論上在同個時刻顯現出來的數值要相等，但是結果卻會有所不同，所以我將許多變數改成了 wire，只要不牽涉到 if…else 或 case，都一律改成 wire，這樣就能保證數值會確實傳遞。

三、執行結果

```
[b06017@cad29 ~/Single_MIPS]$ ncverilog  MIPS_tb.v MIPS.v HSs18n_128x32.v +access+r
ncverilog: 15.20-s039: (c) Copyright 1995-2017 Cadence Design Systems, Inc.
Loading snapshot worklib.SingleCycle_tb:v ..................... Done
*Verdi* Loading libsscore_ius152.so
ncsim> source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc
ncsim> run
Your lw instruction is correct!
Your lw instruction is correct!
Your add instruction is correct!
Your sub instruction is correct!
Your and instruction is correct!
Your beq instruction is correct!
Your or instruction is correct!
Your slt instruction is correct!
Your sw instruction is correct!
Your jump instruction is correct!
Your jal instruction is correct!
Your jr instruction is correct!
Your beq instruction is correct!

--------------------------------------------------------

  Congratulations!! Your design has passed all the test!!

--------------------------------------------------------

Simulation complete via $finish(1) at time 185 NS + 0
./MIPS_tb.v:226                $finish;
ncsim> exit
```