

# 計算機結構 HW3

姓名：莊育權

學號：B03901142

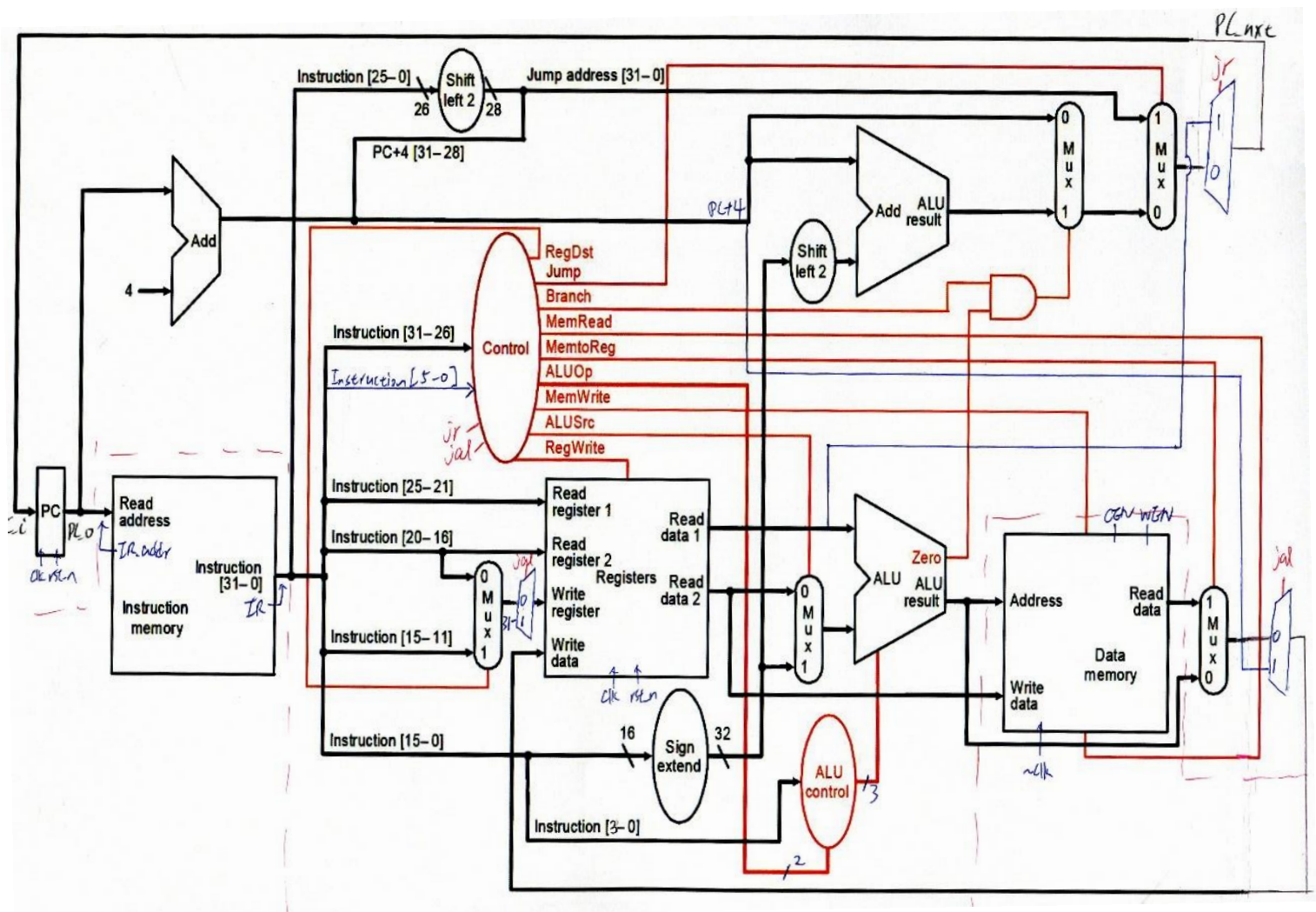
系級：電機三

## I、作業報告

### 一、SingleCycle\_MIPS

#### 1. 設計架構

##### (1) CPU



## (2) Control Signals

Input/ Output	Signal name	R-form	lw	sw	beq	jump	jal	jr
Inputs	Op5	0	1	1	0	0	0	0
	Op4	0	0	0	0	0	0	0
	Op3	0	0	1	0	0	0	0
	Op2	0	0	0	1	0	0	0
	Op1	0	1	1	0	1	1	0
	Op0	0	1	1	0	0	1	0
Outputs	RegDst	1	0	X	X	X	X	X
	ALUSrc	0	1	1	0	X	X	X
	MemToReg	0	1	X	X	X	X	X
	RegWrite	1	1	0	0	0	1	0
	MemRead	0	1	0	0	0	0	0
	MemWrite	0	0	1	0	0	0	0
	Branch	0	0	0	1	X	X	X
	ALUOp1	1	0	0	0	X	X	X
	ALUOp0	0	0	0	1	X	X	X
	Jump	0	0	0	0	1	1	X
	jal	0	0	0	0	0	1	X
	jr	0	0	0	0	0	0	1

## (3) ALU Control Signals

ALUOp1	ALUOp0	Function field	ALU control input
0	0	XXXX	010
0	0	XXXX	010
0	1	XXXX	110
1	0	0000	010
1	0	0010	110
1	0	0100	000
1	0	0101	001
1	0	1010	111

## 二、 模擬結果

### 1. Timing

Set cycle = 6 (ns)

clock CLK (rise edge)	6.00	6.00
clock network delay (ideal)	0.50	6.50
clock uncertainty	-0.10	6.40
output external delay	-4.00	2.40
data required time		2.40
-----		
data required time		2.40
data arrival time		-2.39
-----		
slack (MET)		0.01

Simulated time = 5.51 (ns)

### 2. Area

```
*****
Report : area
Design : SingleCycle_MIPS
Version: G-2012.06
Date   : Mon Apr 17 15:07:14 2017
*****

Library(s) Used:

    typical (File: /home/raid7_2/course/cvstd/CBDK_IC_Contest/CIC

Number of ports:          172
Number of nets:           530
Number of cells:          35
Number of combinational cells: 19
Number of sequential cells:  0
Number of macros:         0
Number of buf/inv:        16
Number of references:      24

Combinational area:      50151.379271
Noncombinational area:   34056.632446
Net Interconnect area:   undefined (No wire load specified)

Total cell area:         84208.011717
Total area:              undefined
```

Total cell area = 84208 ( $\mu\text{m}^2$ )

### 3. AT 值

Area \* Time = 84208 \* 5.51 = 463986.08

### 三、問題與討論

#### 1. IR\_addr 與 IR 是 unknown

一開始寫完 RTL 時，去做測試，發覺 IR\_addr 都是 unknown，而且看了波形圖，IR 也是 unknown，檢查電路也是對的，事後才發現是自己忘記給 IR\_addr 一個初始值=0，給初始值，這樣 CPU 才知道一開始要做甚麼指令。

#### 2. 測試 timing

在做測試 testbench 的時候，去調整 testbench 的 time cycle 一開始從 10 開始往下調，調到 7 的時候發覺不行，但之後繼續往下調發覺居然可以。最後謝謝助教，原來打 7 的時候應該打成 7.0，如果直接打成 7，half cycle 會把 3.5 直接 truncate 成 3，就會造成測試不過。

#### 3. 合成電路

在做合成的時候，因為大家都是用相同的電路圖，合成出來的 area 我猜應該會是差不多，結果最後我自己合成出來發現自己都比別人還多。事後才知道，原來，RTL 裡面如果寫很多 submodule 的話，合成器是使用 module 當成 block 去做最佳化合成，而不是全部電路一起去做最佳化合成。因此才發現原來合成 tool 其實有提供一個功能是可以去 ungroup 所有 submodule，讓所有的一起去做最佳化設計。

### 四、心得

這算是第二次寫 verilog，語法也比上次來的更加熟悉。這次都是按照老師講義上面 CPU 電路設計圖慢慢做，其中只需要再加幾個東西是給 jump, jal 和 jr 使用就好，在寫的過程中並沒有遇到甚麼太大的問題。但也藉由這一次的練習，讓我更熟悉 single cycle CPU 的運作，之前聽老師講，其實都沒有真的很懂，但當你要親手實作的時候，必須自己再去鑽研到要讓自己真的了解 CPU 是怎麼運作，這樣也才寫得出來，也因為這個作業，讓我在第一次小考幾乎不用去複習這部分！

而這次比較特別的是，是需要把電路真的合成出來，自己比較幸運的是合成出來的電路可以直接通過 testbench。可能是因為這次比較不需要做甚麼特別的事情，並竟是 single cycle 而已。也藉由這次作業讓我實地跑過一次合成，了解其流程是如何進行。