

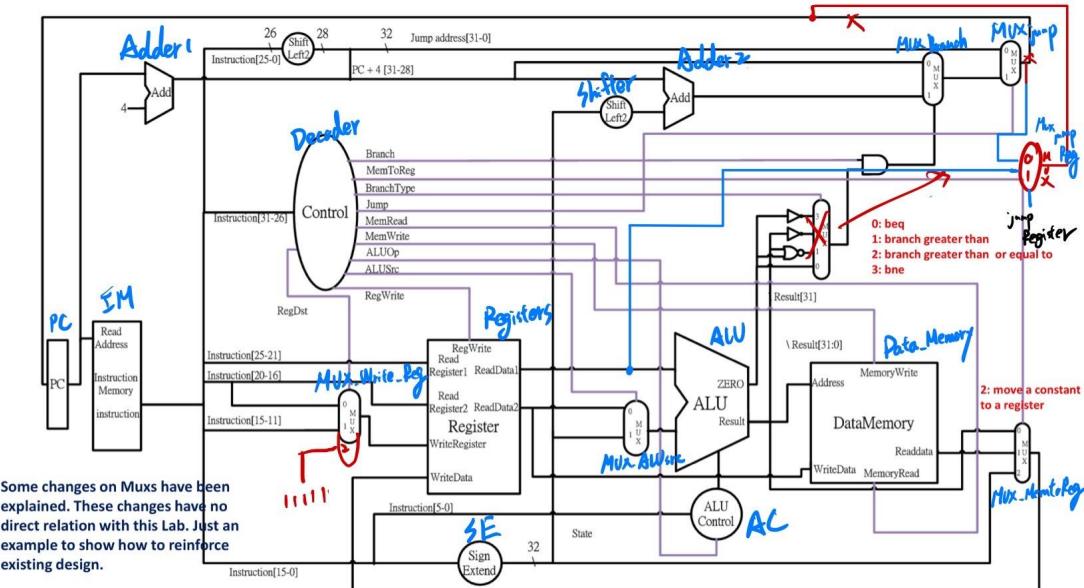
Computer Organization

Lab3

Name : 陳宥安

ID : 109550073

Architecture diagrams:



基本上我就是按照pdf上的電路圖小做修改後再照刻，每個部分在CPU中的名子也寫上去了，在下個部份會稍微敘述一下每個元件部份在幹嘛。

Hardware module analysis:

以下針對各個檔案(元件)做簡單說明：

Adder1, Adder2: 就是兩個加法器($a+b$)。

MUX2to1*4/ MUX3to1*2: 就是個選擇器。

ProgramCounter: 存答案的地方counter。

InstructionMemory: 讀取MIPS指令。

Data Memory: 暫時存依些資料的地方。

Decoder: 判斷MIPS是哪種format、判定要做何種instruction，並decode出一些訊號供續使用。

RegisterFile: 靠Decoder傳入判斷instruction是哪種format來決定ALU的輸入值。

SignExtend: 如果instruction為i-format要做extension。

ALU: 就是個ALU，跟上個Lab一樣。

ShiftLeft2: 就是個shifter，向右shift兩位。

Finished part:

1. CO_P3_test_data1.txt

```
PC = 128
Data Memory = 1, 2, 0, 0, 0, 0, 0, 0
Data Memory = 0, 0, 0, 0, 0, 0, 0, 0
Data Memory = 0, 0, 0, 0, 0, 0, 0, 0
Data Memory = 0, 0, 0, 0, 0, 0, 0, 0
Registers
R0 = 0, R1 = 1, R2 = 2, R3 = 3, R4 = 4, R5 = 5, R6 = 1, R7 = 2
R8 = 4, R9 = 2, R10 = 0, R11 = 0, R12 = 0, R13 = 0, R14 = 0, R15 = 0
R16 = 0, R17 = 0, R18 = 0, R19 = 0, R20 = 0, R21 = 0, R22 = 0, R23 = 0
R24 = 0, R25 = 0, R26 = 0, R27 = 0, R28 = 0, R29 = 128, R30 = 0, R31 = 0
$stop called at time : 705 ns : File "/home/iammrchen/Desktop/C03/C03.srscs/sim_1/imports/Desktop/testbench.v" Line 36
INFO: [USF-XSim-96] XSim completed. Design snapshot 'testbench_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:05 ; elapsed = 00:00:05 . Memory (MB): peak = 8271.137 ; gain = 0.000 ; free physical = 4787 ; free virtual = 18181
```

2. CO_P3_test_data2.txt

```
PC = 128
Data Memory = 0, 0, 0, 0, 0, 0, 0, 0
Data Memory = 0, 0, 0, 0, 68, 2, 1, 68
Data Memory = 2, 1, 68, 4, 3, 16, 0, 0
Registers
R0 = 0, R1 = 0, R2 = 5, R3 = 0, R4 = 0, R5 = 0, R6 = 0, R7 = 0
R8 = 0, R9 = 1, R10 = 0, R11 = 0, R12 = 0, R13 = 0, R14 = 0, R15 = 0
R16 = 0, R17 = 0, R18 = 0, R19 = 0, R20 = 0, R21 = 0, R22 = 0, R23 = 0
R24 = 0, R25 = 0, R26 = 0, R27 = 0, R28 = 0, R29 = 128, R30 = 0, R31 = 16
$stop called at time : 705 ns : File "/home/iammrchen/Desktop/C03/C03.srscs/sim_1/imports/Desktop/testbench.v" Line 36
INFO: [USF-XSim-96] XSim completed. Design snapshot 'testbench_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
```

如圖，輸出正常。

Problems you met and solutions:

我遇到最大的問題是當Counter一到128時下一個cycle就會變成x，但其他答案那些的都是對的，真的完全不知道怎麼下手，只能胡亂改code，改了快兩個小時才發現MUX2to1中選擇的部份一定要用case來做，而且不能always@(*)，但我還是不大清楚為神麼，初步猜測是test的訊號中有雜訊造成MUX選擇錯誤。另外，最後在Simple_Cycle_CPU接線時比上次更辛苦（線有夠多）。

Summary:

這次的兩組MIPS指令如下：

```
addi r1,r0,1          r1=1
addi r2,r0,2          r2=2
addi r3,r0,3          r3=3
addi r4,r0,4          r4=4
addi r5,r0,5          r5=5
jump j
---若jump對，以下兩個addi將不會執行
addi r1,r0,31         r1=31
addi r2,r0,32         r2=32
---
j:
sw   r1,0(r0)        m0=1
sw   r2,4(r0)        m1=2|
lw   r6,0(r0)        r6=1
lw   r7,0(r4)        r7=2
add  r8,r1,r3        r8=4
lw   r9,4(r0)        r9=2
```

```

add r0,r0,r0          //f(0),改變r4的值代表 f(r4) ,若設太大可能要把data memory設大一些
addi a0,zero,4
addi t1,zero,1
jal fib               //JAL:當fib function結束後PC會跳到j final
j final

fib:
addi sp,sp,-12         //stack pointer -12
sw ra,0(sp)            //以下三道sw將reg存入memory中
sw s0,4(sp)
sw s1,8(sp)
add s0,a0,zero
beq s0,zero,rel        //判斷是否f(0)
beq s0,t1,rel          //判斷是否f(1)
addi a0,s0,-1
jal fib
add s1,zero,v0
addi a0,s0,-2
jal fib
add v0,v0,s1

exitfib:
lw ra,0(sp)
lw s0,4(sp)
lw s1,8(sp)
addi sp,sp,12
jr ra                  //function call結束

rel:
addi v0,zero,1
j exitfib

final:
nop

```

第1組主要是為了測試jump、lw、sw的功能，由我們的output結果可知，r1和r2最後沒有變成31和32，所以我們verilog code是正常的。第二組是透過費式數列來測試第1組再加上jal的功能，最後r2是5代表jal的verilog是正常的，完美結束。

另外，希望助教下次在設計此lab時，提供的跟前面lab一樣的檔案中的變數名稱不要改變，要不然上一次lab就完成的東西這次卻不能無縫使用，有些尷尬.....另外希望自己不要再遇到一堆奇怪的bug，或許是因為沒有貓貓的關係(?)