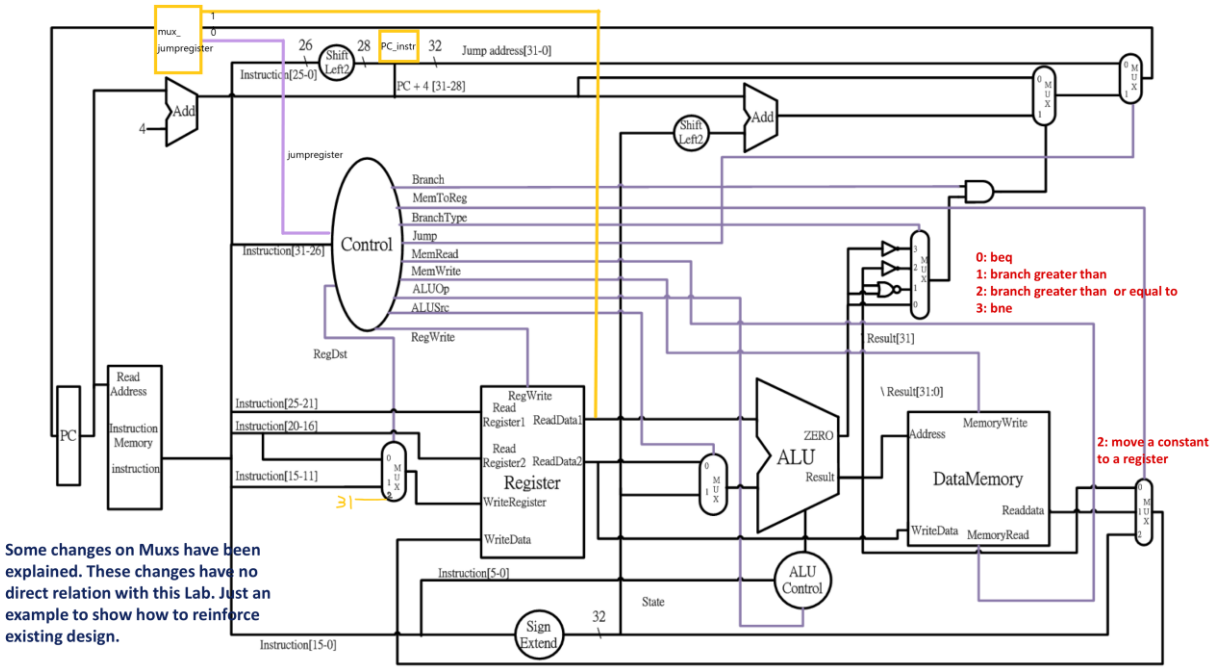
Computer Organization 0616086邱彥慈

Architecture



Hardware module analysis

Shift_Left_26_to_28	Jump address=instr[25:0]<<2
PC_instr	Jump address = {pc[31:28], instr[27:0]}
Mux_Jump	If (Jump) select 0 : jump address
Mux_A	If(branch and zero), select 1 (shift left 2 bits)
Mux_Jump_Register	If(Jump_register) jump to the contents of the first source register.
Mux_Mem_To_Reg	3 to 1 Mux. 2 connect to PC+4
Mux_Write_Reg	3 to 1 Mux. if(JAL), write the next instruction address to the 31th register

	Op	Function	MemRead	MemWrite	MemToReg	Jump	JumpRegister
lw	100011	-	1	0	1	1	0
sw	101011	-	0	1	0	1	0
J	000010	-	-	-	-	0	0
jal	000011	-	-	-	2	0	0
jr	000000	001000	-	-	-	-	1

Result:

Test 1:

```
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 =   0, R30 = 128, R31 =   0
```

Test 2:

```
Registers
R0 =   0, R1 =   0, R2 =   0, R3 =   0, R4 =   3, R5 =   0, R6 =   0, R7 =   0
R8 =   0, R9 =   1, R10 =   0, R11 =   0, R12 =   0, R13 =   0, R14 =   0, R15 =   0
R16 =   3, 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 =   0, R30 = 104, R31 =   0, R31 =  56
```

Summary:

不知道為什麼 r2 不等於 5（因為寫錯）但我已經身心俱疲覺得沒救了 QQ

這次多實作了 sw/lw 和 j/jal/jr，所以增加控制訊號 MemRead/Write/ToReg 跟 Jump/JumpRegister，尤其為了實現 j/jal/jr 增加許多 mux，令人十分心累