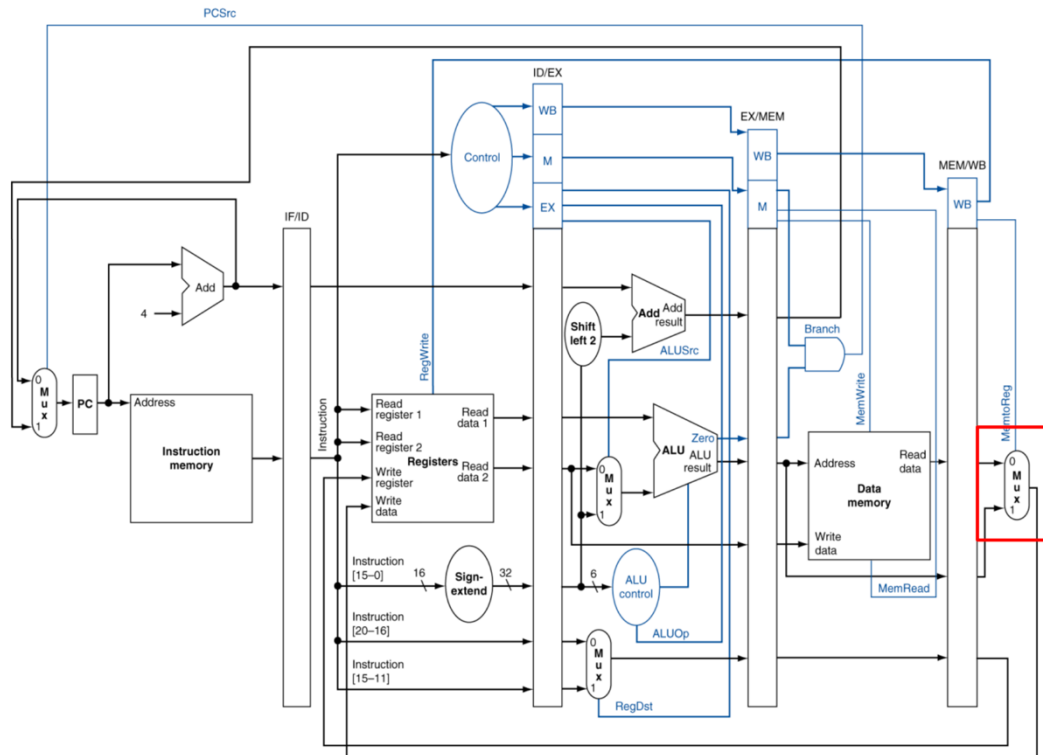


Computer Organization Lab4

ID: 110652011 Name: 郭佳翰

用iverilog測試。

Architecture diagrams:



更動：

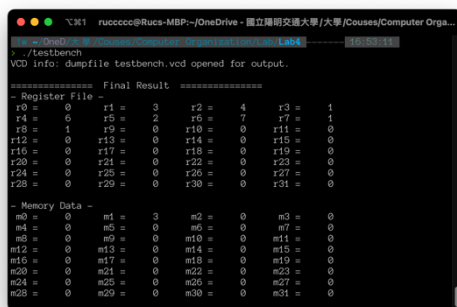
- Instruction 直接傳入 decoder, decoder 再傳各個參數給其他 module。
- 在 ID 階段加入了一個 Shift_left2 用來計算 jump 的地址，並將訊號一直傳至 MEM 階段。
- 延伸 Read_data1 到 MEM 階段，用來傳 jr 的地址。
- 延伸 pc+4 到 WB 階段，用來回傳 jal 的地址。

Hardware module analysis:

1. Decoder: 根據 Instruction 設定給其他 modulo 的參數。用巢狀 switch case 實做。
2. Adder: 相加二數。
3. Sign_Extend: 輸入一 16-bits 訊號，延展至 32-bits。用兩個 For 迴圈實做。
4. MUX_2to1: 輸入二訊號，select_i 決定輸出哪一個。用 if-else 實作。
5. MUX_4to1: 輸入四訊號，select_i 決定輸出哪一個。用 if-else 實作。
6. Shift_Left_Two_32: 左移兩格。用 operator<<實作。
7. 與 Lab3 的主要差在加入 pipeline，將 CPU 分成五個部分，每個部分同時運行。每個部分有各自的變數，用 pipe_reg 來傳需要的變數給下一部分。

Simulation results:

Testcase 1:



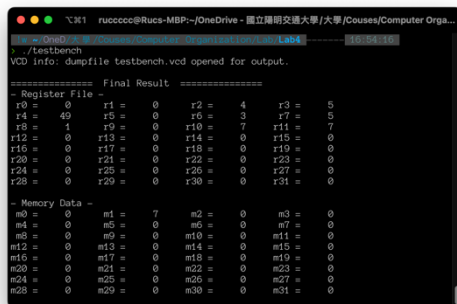
```
rcccccc@RuCs-MBP ~/OneDrive - 國立陽明交通大學 / Courses / Computer Orga...
./testbench
VCD info: dumpfile testbench.vcd opened for output.

===== Final Result =====
-- Register File --
r0 = 0    r1 = 3    r2 = 4    r3 = 1
r4 = 6    r5 = 2    r6 = 7    r7 = 1
r8 = 1    r9 = 0    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 = 0   r31 = 0

-- Memory Data --
m0 = 0    m1 = 3    m2 = 0    m3 = 0
m4 = 0    m5 = 0    m6 = 0    m7 = 0
m8 = 0    m9 = 0    m10 = 0   m11 = 0
m12 = 0   m13 = 0   m14 = 0   m15 = 0
m16 = 0   m17 = 0   m18 = 0   m19 = 0
m20 = 0   m21 = 0   m22 = 0   m23 = 0
m24 = 0   m25 = 0   m26 = 0   m27 = 0
m28 = 0   m29 = 0   m30 = 0   m31 = 0
```

結果符合預期。

Testcase 2:



```
~~~~~ Final Result ~~~~~
-- Register File --
r0 = 0    r1 = 0    r2 = 4    r3 = 5
r4 = 49   r5 = 0    r6 = 3    r7 = 5
r8 = 1    r9 = 0    r10 = 7   r11 = 7
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 = 0   r31 = 0

-- Memory Data --
m0 = 0    m1 = 7    m2 = 0    m3 = 0
m4 = 0    m5 = 0    m6 = 0    m7 = 0
m8 = 0    m9 = 0    m10 = 0   m11 = 0
m12 = 0   m13 = 0   m14 = 0   m15 = 0
m16 = 0   m17 = 0   m18 = 0   m19 = 0
m20 = 0   m21 = 0   m22 = 0   m23 = 0
m24 = 0   m25 = 0   m26 = 0   m27 = 0
m28 = 0   m29 = 0   m30 = 0   m31 = 0
```

結果符合預期

Testcase 3:

- 修改：

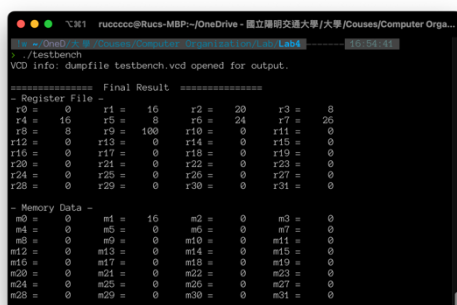
```
001000000000000010000000000010000 // i1
0010000000000000110000000000001000 // i3
0010000000000100100000000001100100 // i10
001000000001000100000000000000100 // i2
101011000000000010000000000000100 // i4
100011000000001000000000000000100 // i5
001000000010011100000000000001010 // i8
0000000000110000100110000000100000 // i7
000000000100000110010100000100010 // i6
000000000111000110100000000100100 // i9
```

- 說明：

I1/I2: 移動 I3, I10 到 I1, I2 之間。

I5/I6, I8/I9: 交錯 I5, I8, I7, I6, I9, I5/I6, I8/I9 之間都有間隔兩個指令。

運行結果：



```
~~~~~ Final Result ~~~~~
-- Register File --
r0 = 0    r1 = 16   r2 = 20   r3 = 8
r4 = 16   r5 = 8    r6 = 24   r7 = 25
r8 = 8    r9 = 100  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 = 0   r31 = 0

-- Memory Data --
m0 = 0    m1 = 16   m2 = 0    m3 = 0
m4 = 0    m5 = 0    m6 = 0    m7 = 0
m8 = 0    m9 = 0    m10 = 0   m11 = 0
m12 = 0   m13 = 0   m14 = 0   m15 = 0
m16 = 0   m17 = 0   m18 = 0   m19 = 0
m20 = 0   m21 = 0   m22 = 0   m23 = 0
m24 = 0   m25 = 0   m26 = 0   m27 = 0
m28 = 0   m29 = 0   m30 = 0   m31 = 0
```

結果符合預期。

Problems you met and solutions:

1. 花了蠻多時間在 pipereg 上，因為變數多所以有蠻多小錯誤的，少傳東西或沒對齊好傳錯等。

Summary:

有一張結構圖可以照著做會讓實作過程輕鬆蠻多的。