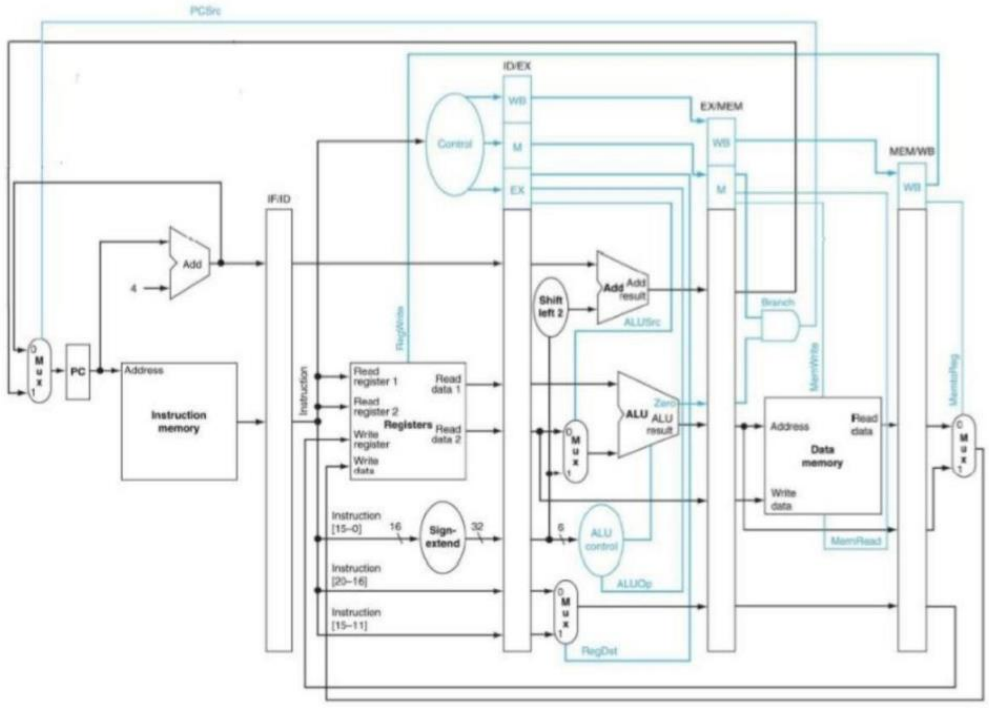
Computer Organization Lab4

Name:邱弘竣

ID:109550136

Architecture diagrams:



Hardware module analysis:

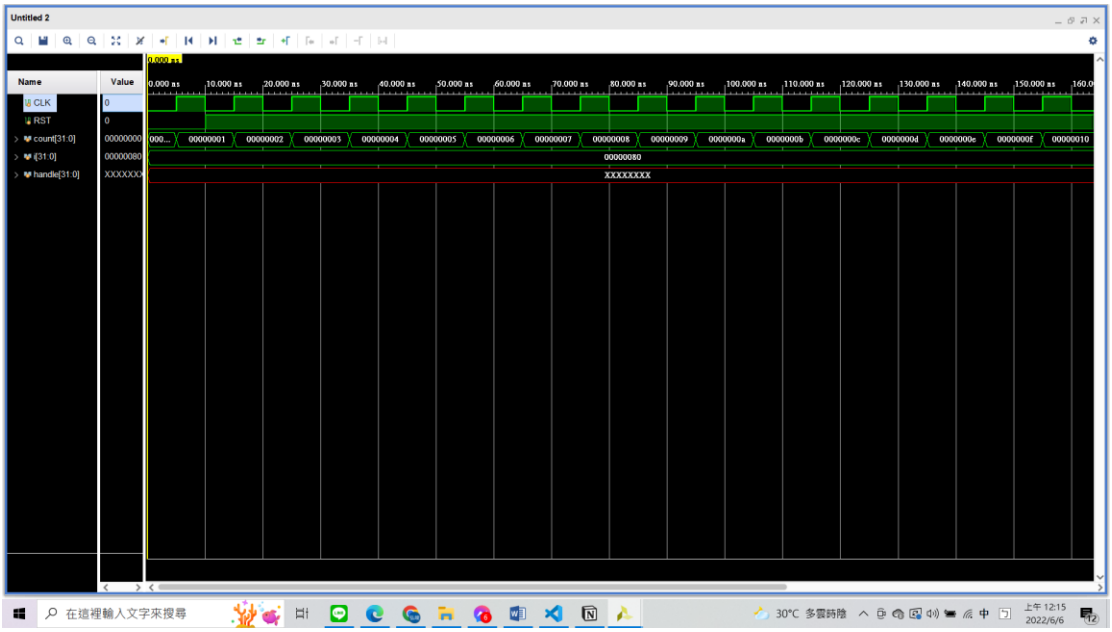
- Adder：we have two adder
  - one is for PC + 4, and one is for imm
- ALU
  - Do the different things based on different input
- ALU_Ctrl
  - control signal of ALU
- Decoder
  - output different control signals based on different operation and send them to different Pipe_Reg
- Instr_Memory
  - read instruction
- ProgramCounter

    - let us know the current PC
- Shift_Left_Two_32
    - shift left two bits and add zero to the empty bit
- Sign_Extend
    - Extend the leftmost bit
- MUX_2to1：we have four MUX_2to1
    - The first one is to check the value of Write Register
    - The second one is to check R-format or I format
    - The third one is to check the value of Write Data
    - The fourth one is to check whether to perform branch
- Pipe_CPU_1
    - combine all the units
- Pipe_Reg：we have four Pipe_Reg
    - IF / ID
    - ID / EX
    - EX / MEM
    - MEM / WB

Finished part：

CO_P4_test_1.txt

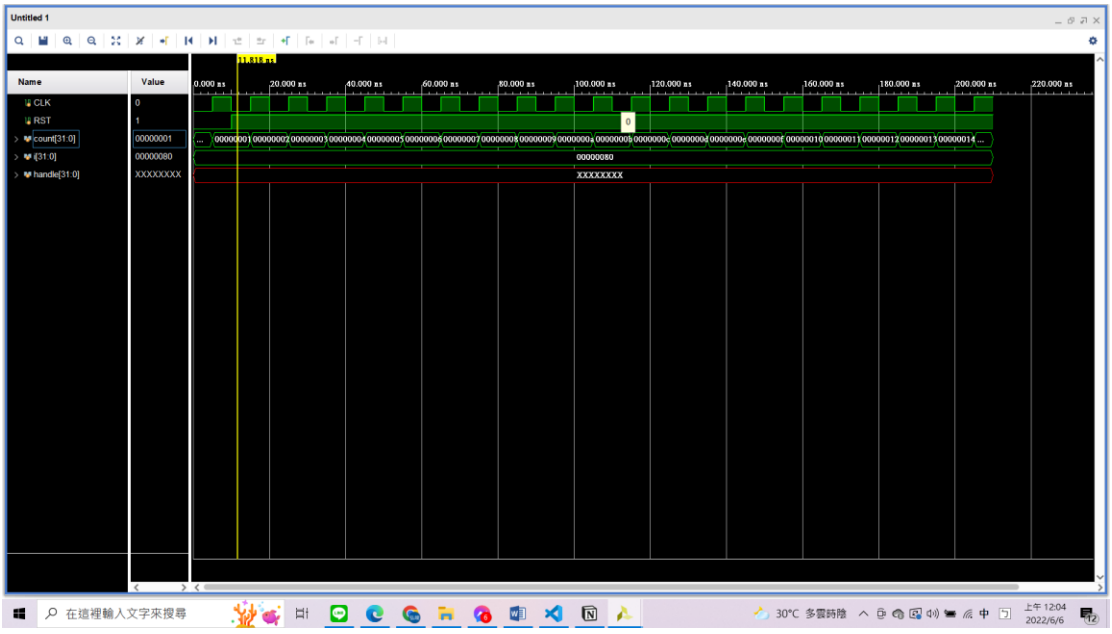
```
Register=============================================================

r0=          0, r1=          3, r2=          4, r3=          1, r4=          6, r5=          2, r6=          7, r7=          1

r8=          1, r9=          0, r10=          3, 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=============================================================

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

r16=          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
```



- In our waveform, Pipe_Reg will output some value every time when clock goes from zero to one so that we can perform out instructions in different state.
- In the testbench, count will plus one every time when clock goes from zero to one.
- The instruction below is the meaning of CO_P4_test_1.txt. Trace it from the begining to the end and we can get the value in the memory and the registers.
    - addi $1,$0,3; // a = 3
    - addi $2,$0,4; // b = 4
    - addi $3,$0,1; // c = 1
    - sw $1,4($0); // A[1] = 3
    - add $4,$1,$1; // $4 = 2*a*
    - *or $6,$1,$2; // e = a | b*
    - *and $7,$1,$3; // f = a & c*
    - *sub $5,$4,$2; // d = 2*a - b
    - slt $8,$1,$2; // g = a < b
    - beq $1,$2,begin
    - lw $10,4($0); // i = A[1]

CO_P4_test_2.txt

```
Register=============================================================

r0=         0, r1=         16, r2=         20, r3=         8, r4=         16, r5=         8, r6=         24, r7=         26

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=============================================================

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

r16=         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
```



- Same as data1. However, after solving the problem of hazard, we can get the real answer.
- Trace it from the begining to the end and we can get the value in the memory and the registers. The machine code is below.

Problems you met and solutions:

- One of the parts that stopped my footsteps is that I don’ t know how to input a lot of different wires into a port. It is intuitive to add extra ports to Pipe_Reg. However, I don’ t think it’ s a good idea. If that is the only solution then I

don’ t think TA will design our template like that way. As a result, I searched the information on the Internet and finally found a better way.

- After connecting all the wires, I still got the wrong answer. I felt confused about this circumstance. As a result, I checked my answer in each step and looked for the function outputting the wrong value. Then I found that I forgot to add “multiplication” in ALU.

Bonus (optional):

- for I1 / I2
    - We just need to switch the position of the two instructions. It means that we perform I2 first and then I1.
- for I5 / I6, I8 / I9
    - can be solved by reorder instructions
    - modified machine code：

00100000000000010000000000010000
00100000000000110000000000001000
00000000000000000000000000000000
00100000001000100000000000000100
10101100000000010000000000000100
10001100000001000000000000000100
00100000001001110000000000001010
00000000011000010011000000100000
00000000100000110010100000100010
00000000111000110100000000100100
00100000000010010000000001100100

Summary:

After this lab, I think I understand more about pipeline. During the class, I felt confused of the rectangle, Pipe_Reg. It is a strange register because I didn’t know what to do in that register. Is seems like I input something into it and the register will output the same thing without doing anything else. It sounded a little bit strang. During this lab, I realize that what I understood before is right. It is a “register”, not a gate. It’s just a place to store our data not to perform something in it. I feel great after solving the problem deep in my heart.