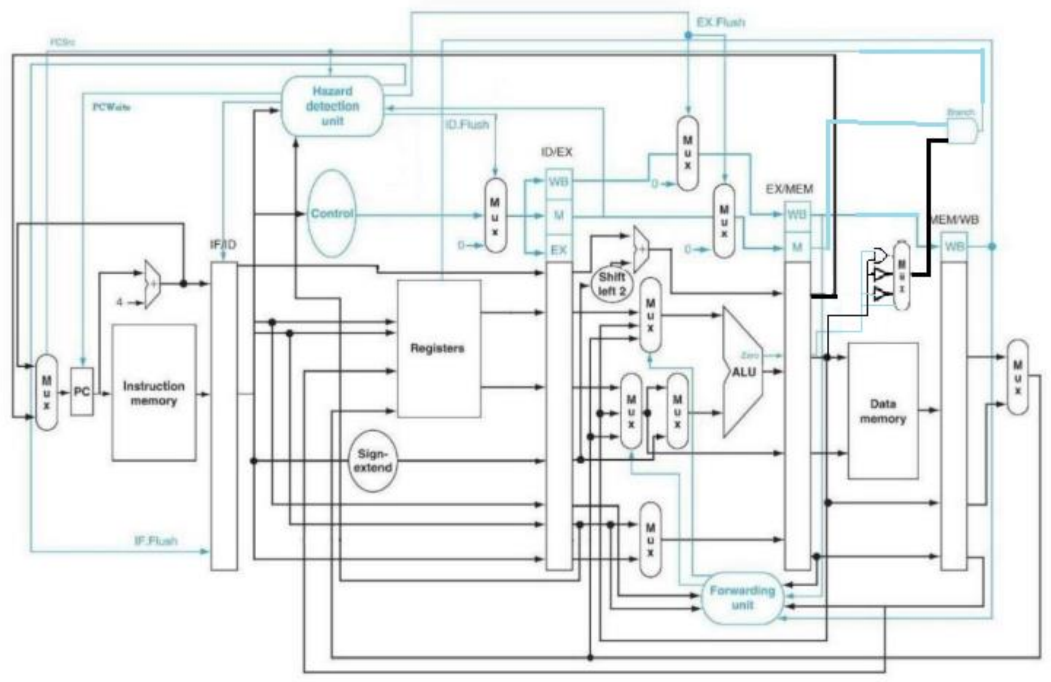
Computer Organization Lab5

Name:邱弘竣

ID:109550136

Architecture diagrams:



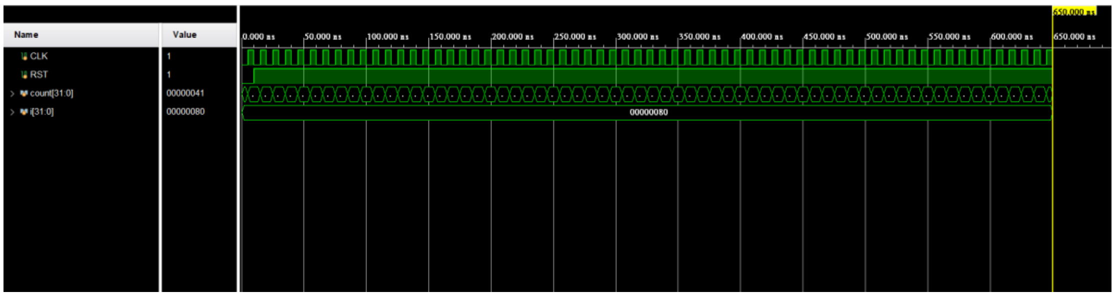
Hardware module analysis:

- 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

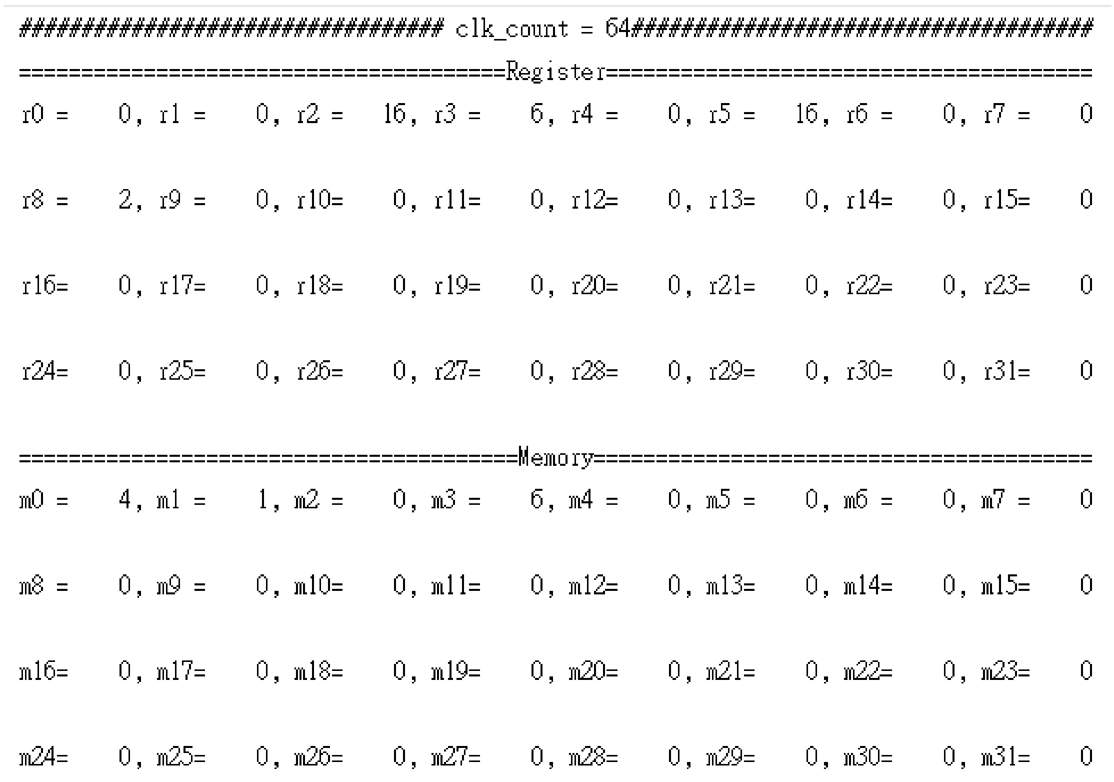
    - read instruction
- ProgramCounter
    - let us know the current PC
- Shift_Left_Two_32
    - shift left two bits and add zero to the empty bit
- Sign_Extend
    - Extend the leftmost bit
- MUX_2to1：we have four MUX_2to1
    - The first one is to check the value of Write Register
    - The second one 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
- MUX_3to1：we have two MUX_3to1
    - The first one is to check ForwardA
    - The second one is to check ForwardB
- MUX_4to1：we have one MUX_4to1
    - check the type of the branch
- Pipe_CPU_1
    - combine all the units
- Pipe_Reg：we have four Pipe_Reg
    - IF / ID
    - ID / EX
    - EX / MEM
    - MEM / WB
- Hazard_detection
    - check load-use hazard
- forwarding_unit
    - If data hazard happens, we have to perform “forward”.

Finished part:

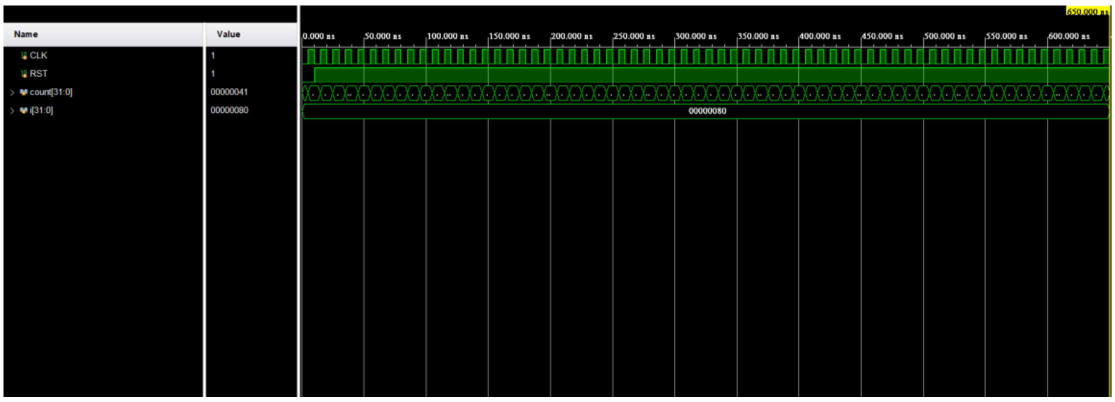
```
################################## clk_count = 17#####################################
=======================================Register=======================================
r0 =     0, r1 =    16, r2 =   256, r3 =     8, r4 =    16, r5 =     8, r6 =    24, r7 =    26

r8 =     8, 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=     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

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
```



```
Name            Value
CLK             1
RST             1
> count[31:0]   00000041
> i[31:0]       00000080
```



```
################################## clk_count = 64#####################################
=======================================Register=======================================
r0 =    0, r1 =    0, r2 =   16, r3 =    6, r4 =    0, r5 =   16, r6 =    0, r7 =    0

r8 =    2, 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========================================
m0 =    4, m1 =    1, m2 =    0, m3 =    6, 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
```



```
Name          Value
CLK           1
RST           1
count[31:0]   00000041
i[31:0]       00000080
```

The result is the same as the answer in the spec.

Problems you met and solutions:

I wasn't sure how to detect hazards at first using our diagram. As a result, I tried to find the technique in the PPT and looked up some material online. Then I realized that the diagram needs to be revised to match our requirements. I spent some time pondering how to complete our lab, and I eventually finished it.

Summary:

This lab is easier to complete than the previous ones. The most pressing issue is dealing with the problem of detecting hazards. The PPT in class explains the concept explicitly in this section, so all we have to do is understand it and design Hazard_Detection.v. We completed this lab after resolving a few bugs.