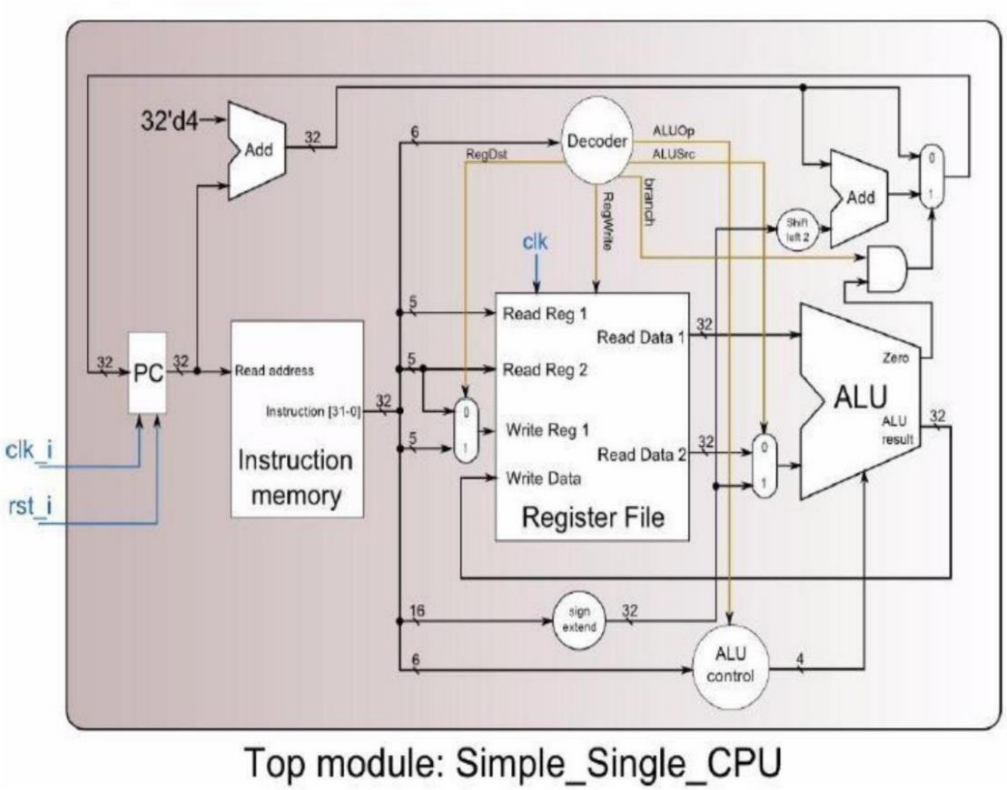
Computer Organization Lab2

110550085房天越

Architecture diagrams:



Top module: Simple_Single_CPU

Hardware module analysis:

This CPU is implemented by the given module above, it can accept the commands consisting: R-format, addi, beq, and slti.

The Simple_Singl_CPU.v combines the following seven parts:

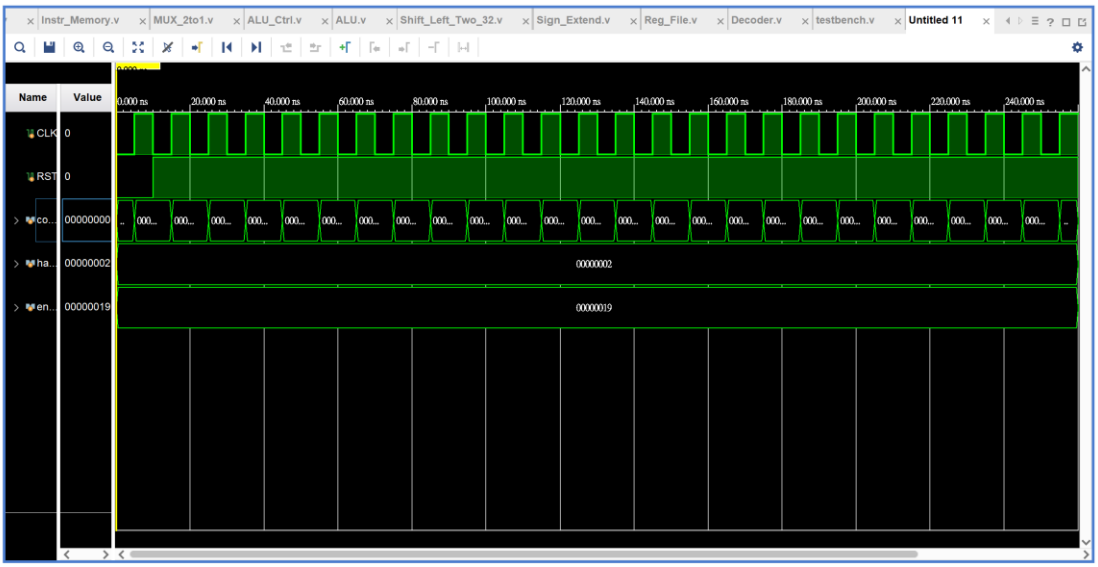
1. Adder.v:
   It returns the sum of the two given numbers.
2. ALU_Ctrl.v:
   Input funct_i and ALUOp_i, output ALUCtrl_i to control the ALU to do the functions listed above.
3. ALU.v:
   Do the function for the two numbers with the code given by ALU_Ctrl.v. If the result is zero, output the zero signal.
4. Decoder.v:
   Given the first six bits of instruction (opcode), output the

signals to control other modules.

5. MUX_2to1.v:
   Given the input 0 or 1, output the corresponding number.
6. Shift_Left_Two_32.v:
   Shift the given number left for two bits.
7. Sign_Extend.v:
   Convert the given number to 32 bits by filling the front 16 bits with the sign bit in the given number.

Finished part:

The waveform:



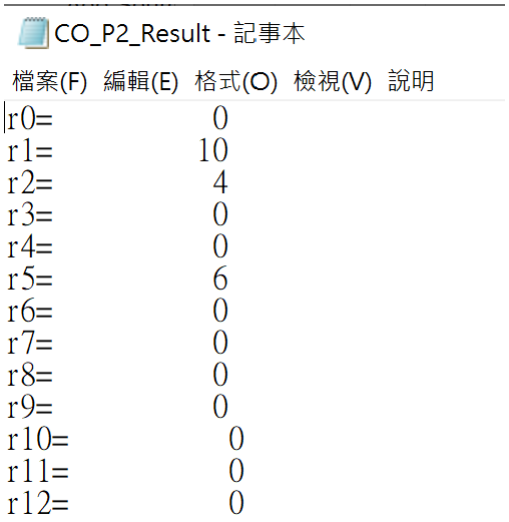
The result for the first test data:

CO_P2_Result - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V)

```
r0=          0
r1=          1
r2=          0
r3=          0
r4=          0
r5=          0
r6=          0
r7=         14
r8=          0
r9=         15
r10=         0
r11=         0
r12=         0
```

The result for the second test data:



These results are the same as the expected results given in the spec.

Problems you met and solutions:

First, for the decoder part, I didn't know what kind of signals I should generate for the outputs, like ALUop, ALUSrc, branch, etc. After reading the lecture file again, I finally understood what I should do.
Also, while this CPU is relatively simple, I still had a hard time to debug. At first my results are all zero, after a long time of debugging, I finally found that that is because I accidently set a 3-bit signal to only 1 bit. After remedying this problem, the results finally became correct.

Summary:

After this Lab, I have a better comprehension of what CPU does and also understand better about what each part of the instruction code does and how to design modules to deal with them, to conclude, I learned a lot in this lab.