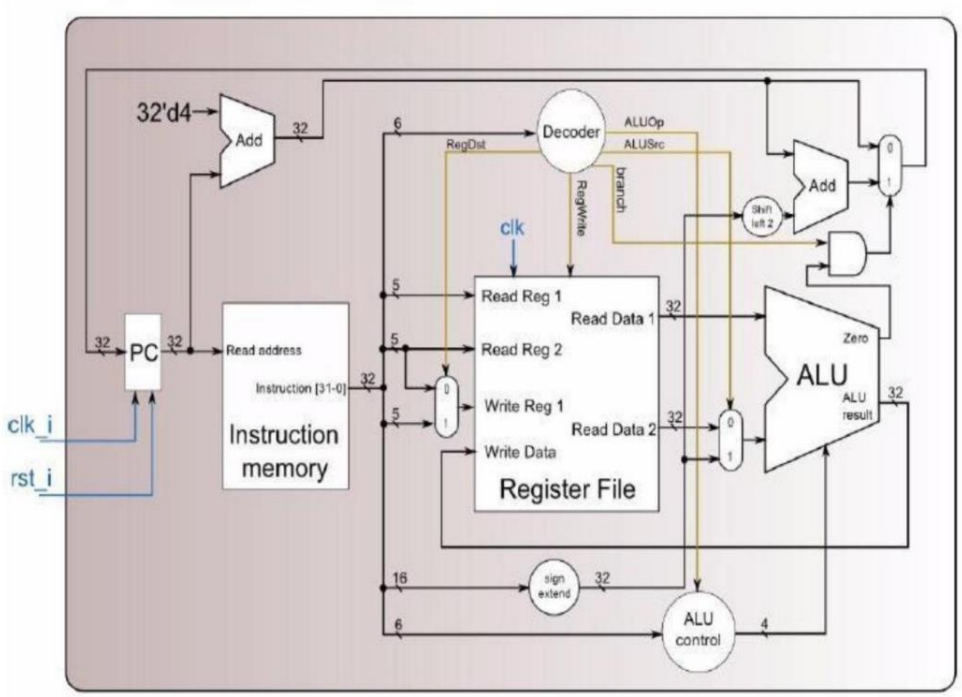
Computer Organization Lab2

Name: 陳星宇

ID: 109550060

Architecture diagrams:



Hardware module analysis:

(explain how the design work and its pros and cons)

MIPS instructions will be processed in the instruction memory then parses its opcode in the decoder and further decide each parameter: RegOut is for which register is the destination register; AluOp (combined with function field) is to tell which operation should be given in the ALU control; ALUSrc is to tell the writing data is from I-format data field or the data read from the register and RegWrite is for whether the instruction involves register writing. Register File will store every register and its value. ALU is for processing instruction sets. The other module is to tell whether there is a branch instruction to jump back or forward.

Finished part:

(show the screenshot of the simulation result and waveform, and

explain it)

Testcase1:

```
# run 1000ns
          2
r0=           0
r1=          10
r2=           4
r3=           0
r4=           0
r5=           6
r6=           0
r7=           0
r8=           0
r9=           0
r10=          0
r11=          0
r12=          0
```

Testcase2:

```
# run 1000ns
          2
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
```

Problems you met and solutions:

A big problem that I've met is that the ALU_Ctrl is written wrong when dealing with 3 bit opcode and function control so that all of my R-format MIPS instructions are unable to run correctly. The variable in the always block should contain funct_i and opcode.

Summary:

This lab makes me implement and how the single-cycled CPU work and I know that I can import sub-module into the waveform by adding it from the scope panel while I need to debug my code.