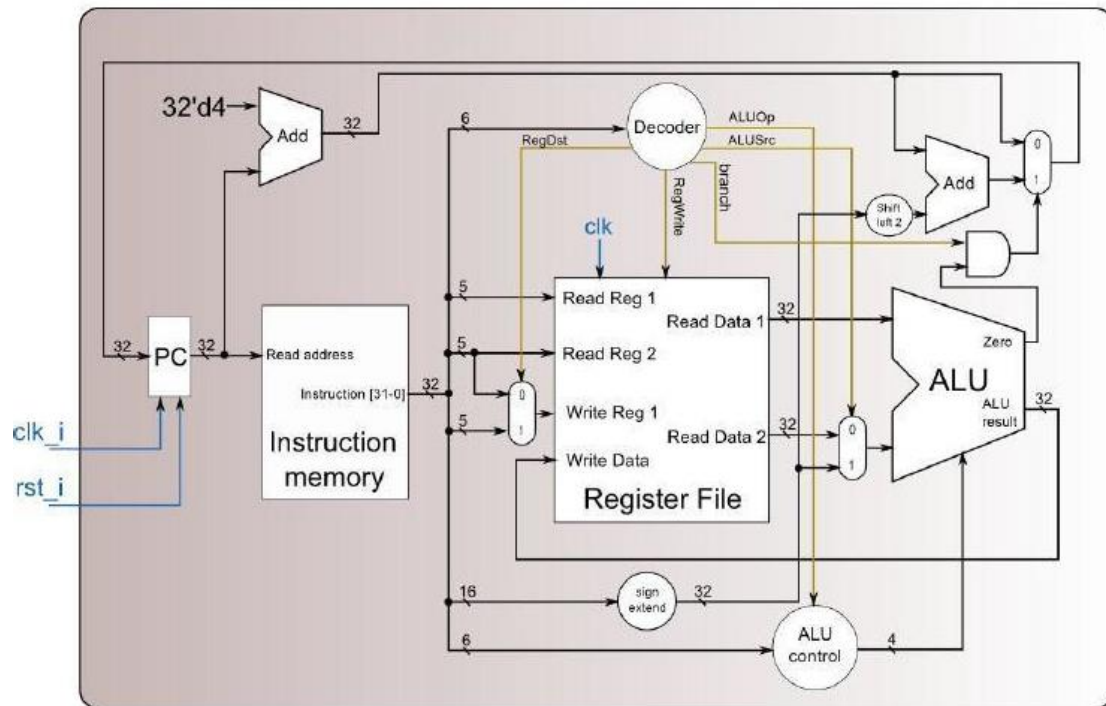


# Computer Organization

## Architecture diagrams:

### - Single Cycle CPU



## Hardware module analysis:

直接照著題目 pdf 里的 diagram 設計

- Program Counter 每回合通過 左上的 Adder + 4
- Instruction memory 根據 R-format, I-format 去決定從哪個 Register file 的位置讀資料
- Decoder 去用來判斷該 instruction 是否需要 RegDst, RegWrite, branch, ALUOp, ALUSrc
- 如果是 I-format, 會透過 sign extender 把 16 bit extend 成 32 bit
- Branch instruction 的結果是由 ALU 的 zero output 來決定
- 右上的 adder 是用來決定 branch target 的

## Finished Part:

### -Testcase1

CO\_P2\_Result.txt - Notepad

File Edit Format View Help

```
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

CO\_P2\_Result.txt - Notepad

File Edit Format View Help

```
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:

一開始沒發現 testcase2 有 beq，花了蠻多時間在研究 r6 的值為什麼會是 0

## Summary:

通過此次 lab 了解到 1 個簡單的 CPU 所需要的 component 及其讀 instruction 的方式