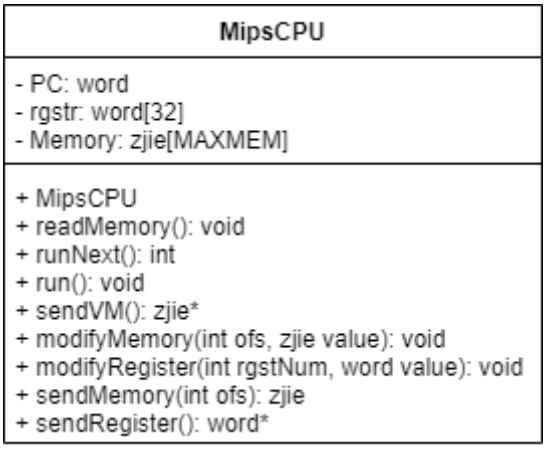
孙恺元 3180102085

# CPU设计模拟

为了模拟CPU，我们需要以下功能

- 可以模拟内存和寄存器
- 可以将文件内容读入内存中，并当作指令运行
- 可以模拟显存，根据指令要求输出
- 可以向图形界面提供接口

因此，我们设计CPU的类结构如下：




变量和方法的实现逻辑如下。

## 成员变量

### PC

PC是当前指令位置的标记，CPU运行时，会运行内存的PC位置的指令。

### rgstr

rgstr是寄存器数组，按照编号存储了寄存器的值，均为32位无符号整数。

### Memory

整个CPU的核心部分，是一个4096zjie大小的数组。在读入指令时，MipsCPU会打开文件，将文件内容按二进制存储到Memory从0开始的地方；内存的第1024zjie处开始的800个zjie是显存，CPU每次运行完一条指令，都会更新显示模拟显存中的内容。

## run

run函数是整个类的核心方法，我们通过run来运行一条指令。指令的内容由PC决定，我们从内存的PC处读取32位，作为要执行的指令IR，并且给PC自增2（因为32位是两个zjie）。

在指令处理中我们可能需要从IR中析取许多数据，为了方便，我们不做判断，一次性全部析取，分别是指令的

- op：31~26位
- rs：25~21位
- rt：20~16位
- rd：15~11位

- sft：11~6位
- fun：5~0位
- dat：15~0位
- adr：25~0位，然后再自乘4

程序结束的标志是指令为全1。

根据析取得到的各个量，可以确定指令的类型和具体数据，然后操作对应的寄存器即可。

每一次run后，都会更新显存。显存的显示方式是将800个zjie按照20行40列的方式输出，每一个zjie看作一个char类型，即舍去了高8位。

## syscall

syscall指令是最特殊的指令，因此我们特别提及。

syscall指令根据寄存器v0（编号为2）中的值判断功能

| v0 | 功能 |
| --- | --- |
| 1 | a0作为整数输出 |
| 2 | a0作为浮点数输出 |
| 4 | a0作为字符串的首地址，输出字符串 |
| 5 | 输入整数到v0 |
| 6 | 输入浮点数到v0 |
| 8 | 输入字符串，v0存首地址 |
| 10 | 退出 |
| 11 | a0作为char输出 |
| 12 | 输入char到a0 |