ICS LABS REPORT

【设计思路】

①main.cpp：主t体代码框架如下：

```
virtual_machine_tp virtual_machine(gBeginningAddress,
 gInputFileName, gRegisterStatusFileName);

while(halt_flag) {
    // Single step
    halt_flag = virtual_machine.NextStep();
    ++time_flag;
}
```

从中即可得出设计思路：halt_flag对应的就是PC指针，对指令进行逐行模拟执行即可。

②simulator.cpp

函数作用分析:

```
NextStep()
//模拟逐行运行的步骤，返回当前PC值
//

virtual_machine_tp::VM_***（）
//模拟不同指令的运行结果

template <typename T, unsigned B>
inline T SignExtend(const T x){}
//位补全类模板函数，将B位的数x拓展为16位数

virtual_machine_tp(const int16_t address,
const std::string &memfile, const std::string &regfile)
//导入程序起始地址、寄存器文件及内存文件（代码存储块）

UpdateCondRegister(int regname)
//状态寄存器更新函数
```

③memory.cpp：模拟内存存取操作

④register.cpp：寄存器值的输出

【实验收获】

与LabA相同，对于本次大实验，本人仍然在项目框架构建上受益匪浅。参考LC3的构造，在模拟时将其分为寄存器、内存、处理器三大块分别进行功能复现，这样的设计思路大大简化了框架，同时也降低了出错

的可能。同时，进一步学会使用cmake进行多文件编译，更让我明白了处理大项目时批量编译工具的重要性。感谢助教，让我学到了如此之思想及技能。

还是那样，此次实验令我受益匪浅.