# 挑战：奋战n(n<=3)小时，编台虚拟计算机VM！

## 需求：

1. 编写一代机

用面向对象的方法，设计一代虚拟计算机。包括：控制器(CU）,运算器(ALU),存储器（Memory），输入、输出部件。

要求：能够正确执行给定指令集的指令序列，输出正确结果。

高阶要求，能应对异常输入，可扩展指令集、设备等。

冯诺依曼计算机逻辑结构五部分：

控制器CU：根据指令指针(PC, Program Counter，保存当前运行的指令的地址，初值为0。假设指令从0号单元开始存放。在执行完一条指令后，PC默认指向下一条指令)。控制器指挥各个部件共同完成运行。它依次读取每条指令、译指、通知各个部件完成相应任务，直到读到HALT停机指令。

运算器ALU：能执行指定指令集的运算，有累加器(AC, accumulator)寄存器。

存储器Memory：一块连续的空间，可以存储数据和指令，可按初始化参数指定大小。

输入设备：可从标准控制台输入设备接收输入指令序列

输出设备：可从标准控制台输出设备输结果

**简单指令系统**:（假设每条指令由：两位操作码(opcode)整数+两位操作数(operand)整数）见下表：

表1 简单指令系统

加减法用补码的二进制操作

乘除法可以转换为补码的一位乘 这里直接用程序语言实现\*/

|  |  |  |
| --- | --- | --- |
| 指令 代码 | | 说明 |
| 输入/输出 | READ=10 | 从控制台读一个整数放入指定的内存单元中 IO指令存在，执行的时候操作系统内核态 中断 IO端口编址（特殊的“内存地址”）类似于访存指令，发中断，CPU访IO端口编址，总线进寄存器，CPU再写入内存，中断结束；还可以DMA，DMA控制器直接从IO端口地址读缓冲区进内存，完成后再发DMA中断给CPU |
| WRITE=11 | 将指定的内存单元的内容输出到控制台 CPU从内存读进寄存器，寄存器再总线进缓冲区 |
| 存/取指令 | LOAD=20 | 将一个内存指定单元中的整1数存入运算器中的累加器（或者某个寄存器） |
| STORE=21 | 将累加器（或者某个寄存器）中的内容存入内存的指定单元 |
| 算术指令 | ADD=30 | 将指定内存单元的内容与累加器中的内容相加，结果存入累加器 |
| SUB=31 | 从累加器中的内容减去指定内存单元中的内容，结果存入累加器 |
| MUL=32 | 将指定内存单元的内容与累加器中的内容相乘，结果存入累加器 |
| DIV=33 | 将累加器中的内容除以指定内存单元中的内容，结果存入累加器 |
| 控制/传输指令 | BR=40 | 将指令计数器设定为指定的内存地址 无条件跳转 |
| BRZ=41 | 当累加器内容为0时，指令计数器设定为指定的内存地址。 有条件跳转 |
| HALT=43 | 终止程序执行 |

表2一个机器语言程序样例

|  |  |  |
| --- | --- | --- |
| 内存单元 | 内容 | 动作 |
| 00 | 1007 | 从控制台读入一个整数放入07号单元 |
| 01 | 1008 | 从控制台读入一个整数放入08号单元 |
| 02 | 2007 | 将07号单元的内容存入运算器中的累加器 |
| 03 | 3308 | 将累加器中的内容除以08号单元的内容 |
| 04 | 2109 | 将累加器中的内容存入09号单元 |
| 05 | 1109 | 将09号单元的内容输出到控制台 |
| 06 | 4010 | 跳到内存10号单元 实际上要跳转了，但是实验不深，跳转往往意味着新的栈帧，新的函数，这里太复杂，不实现了 |
| 07 | 0000 | 空指令 |
| 08 | 0000 | 空指令 |
| 09 | 0000 | 空指令 |
| 10 | 4300 | 终止程序运行 |

**请自主设计出以下类**：（参照之前多文件组成项目的例子，最好设计分成几个头文件）

QD\_CU类：

QD\_ALU类：

QD\_Memory类：

QD\_In类

QD\_Out类

Computer\_1类：组合类 一开始记得把代码文本区硬编码进内存 初始化

其中的成员、函数自定。

**主调测试示例代码**：

int main(){

Computer\_1 mycomputer1("青芯CU0","青芯ALU0",512,"青入设备0","青出设备0"); //内存可存512条指令

mycomputer1.run();

return 0;

}

机器运行可接收控制台输入的指令序列.

测试1: 请尝试在控制台输入表2的机器指令序列，测试能否正确运行，输出正确结果。

测试2：测试输入非法数据，程序的应对。非法数据 比如除以0 或者是BR到一个不存在的地址（PC指针）读进07 08 号 读进的大小超过了寄存器的范围等

1. 继承开发二代机

在继承一代机的基础上，开发新的二代机Computer\_2（增加新的NPU和相应新指令,如: MCNN=34，输出”对内存单元所指对象进行卷积运算“）

生成二代机的对象，测试能否正确执行新指令。

如何实现NPU？如何从内存中取矩阵和卷积算子 还有就是怎么读矩阵 内存怎么存

NPU中的卷积操作，实现不了 就直接编码实现卷积运算 这里还需要有一个matrix内部类

LOAD STORE 都得有专门的针对M的方法 Matrix

READ WRITE 都得有专门针对M的方法 循环用单数寄存器吧。。。