**实验1.原型机vspm1.0**

【实验目的】

1. 了解冯诺伊曼体系结构；
2. 理解指令集结构及其作用；
3. 理解计算机的运行过程，就是指令的执行过程，并初步掌握调试方法。

【实验准备】

1. 阅读教材，掌握冯诺伊曼体系的相关内容；
2. 学习课程《最小系统与原型机I》。

【实验步骤】

1. 进入终端，使用cd vspm1.0进入目录；

使用.**/vspm a-inst.txt**来运行原型机1.0的模拟器，其中a-inst.txt为代码文件，

6

in R1 #输入a到R1

movi 1 #设置R0为1

add R2,R1 #R2存放累加值

sub R1,R0 #R1的值即a减去1,此时会设置Ｇ值

movd #将当前PC值保存在R3中

movi -3 #存放-3到R0中,跳转到第二行

add R3,R0 #R3减去3，注意此时不能用SUB指令，会影响G值

jg #如果R1的值还大于1，则跳到第2行去执行

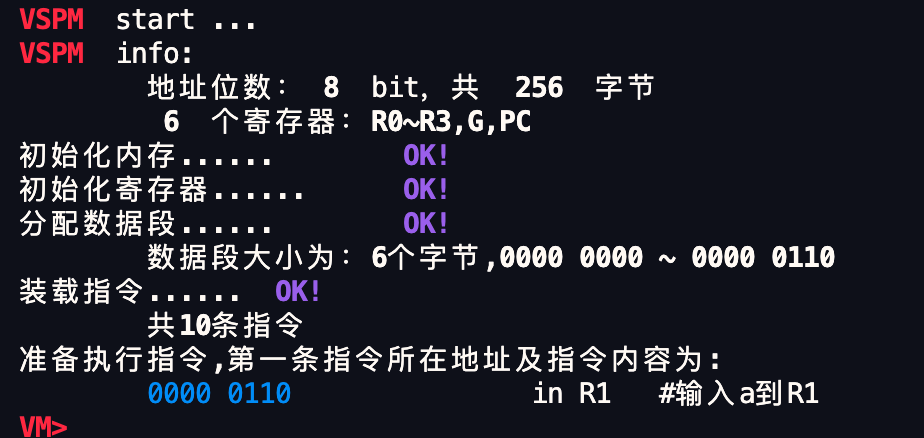
out R2 #如果R1的值此时小于等于1，则准备输出

halt #停机

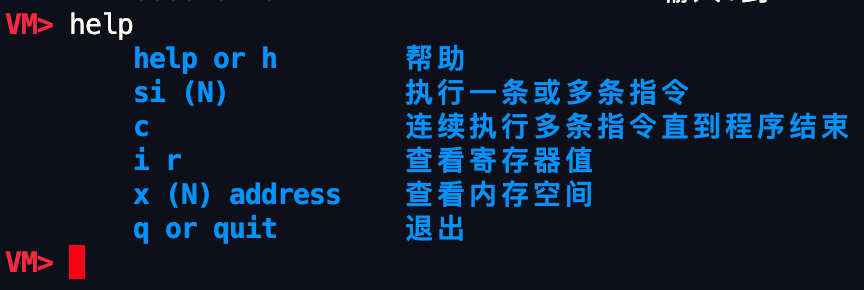
第一行的6，表示分配6个字节的数据段；

后面的均为原型机指令，每一行指令代表的意义及整体执行结果在《最小系统与原型机I》中已经进行了详细说明。

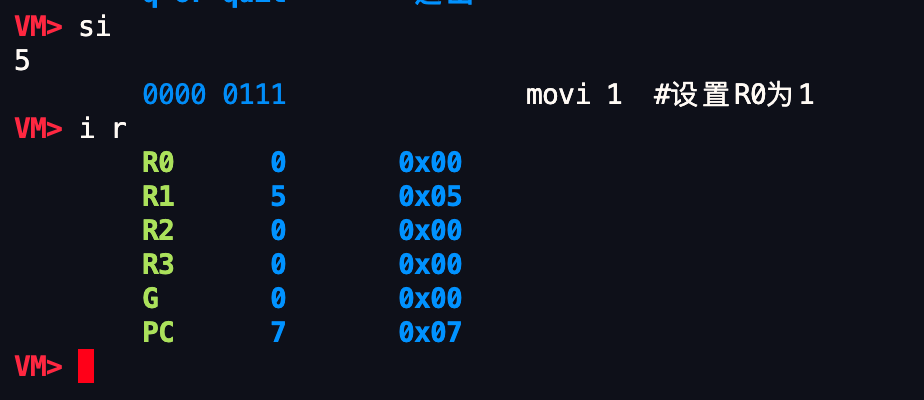
运行后界面如下图所示。



1. 在运行后，提示将要执行的指令地址及内容，在本例中，提示要执行位于内存**00000110**处的指令“in R1”，即等待输入一个整数值。此时输入**si**则表示执行此指令，同时也可以输入其他的指令，使用help可以查看此模拟器支持的命令：
2. 此时输入**i r**，可以查看各个寄存器的值，而输入**x 6 0000**则表示查看从**0000**开始的连续6个内存地址值，结果如图所示。



1. 输入**si**，则表示运行一条指令，例如此时运行的指令是“in R1”，表示等待输入，输入一个值5，在输入完成后将此数值存至**R0**寄存器，运行完成后，再运行**i r**指令，就可以看到输入的值5确实是已经存在**R0**寄存器中，每个寄存器的值都用十进制和十六进制表示，如下图所示。



后续程序的执行过程可参照视频进行操作。

1. 程序执行完毕后，可以使用q退出。

## 三、练习思考

### 1.练习内容

1. 按照上述的实验步骤，完成相关操作；
2. 在目录下还有**b-inst.txt**和**c-inst.txt**，请运行并调试，并对这些代码所做的工作进行解释；
3. 完成实验报告。

### 2.思考问题

（1）如果基于这些指令实现两个整数的乘法与除法？

（2）vspm1.0的指令集是否完备？如果是，那么如何证明（提示：搜索并阅读“可计算性理论”）？如果不是，那么要增加哪些指令？

我认为vspm1.0的指令集是不完备的。

（3）如果一台计算机只支持加法、减法操作，那么能否计算三角函数，对数函数？（提示：搜索并阅读“泰勒级数展开”等内容）

三角函数和对数函数可以通过泰勒级数用无限项连加式来表示，通过舍去后面的无限小项，留下只有四则运算的式子，近似得到函数值。而乘、除运算可以通过加、减操作实现。总的来说，如果一台计算机只支持加法、减法操作，可以计算三角函数，对数函数。

（4）对于某个需要完成的功能，如果既可以通过硬件上增加电路来实现，也可以通过其他已有指令的组合来实现，那么如何判断哪一种比较合适？（提示：搜索并阅读RISC与CISC）。

