**实验报告**

PB19030861 王湘峰

一、

思路：首先考虑将后n-2位数字前移两位，在此之前先思考前移一位的情况。在十进制中，一个数乘以十，则所有位数前移一格，如果这时把第一位数字记录下来加到最后，即可完成一次“冒泡”。相应的，二进制逢二进一，所以一个二进制数加上自己（即乘以二）就会相应的往前移一位。只要记录第一位数字并把它加到最后即可完成一次“冒泡”。再重复该操作一次即可将两位数字“冒泡”。首先可以把数据“乘以二”并存放到一个寄存器1中，然后使用10000000000与原始数据作AND运算，将结果放到寄存器2中。然后根据这个数据是否为0再对寄存器1进行加一/无操作。

二、

（具体的源码以及注释已经放在了program.bin中）

内存中应预存的数据：

X3000:任意16 bits的二进制数

X3001：rotate amount

X3003：“x8000”(即1000 0000 0000 0000)

X3002:待写入

各寄存器的作用：

R1：取x3003中的值，用于后续进行AND运算

R2：储存原始数据的地址，即x3000

R3：取原始数据的值或上一次经过冒泡的值

R4：储存AND运算的结果

R5：用于储存冒泡过后的值以及作为写入内存的源寄存器

R6：储存rotate amount的值，作为counter

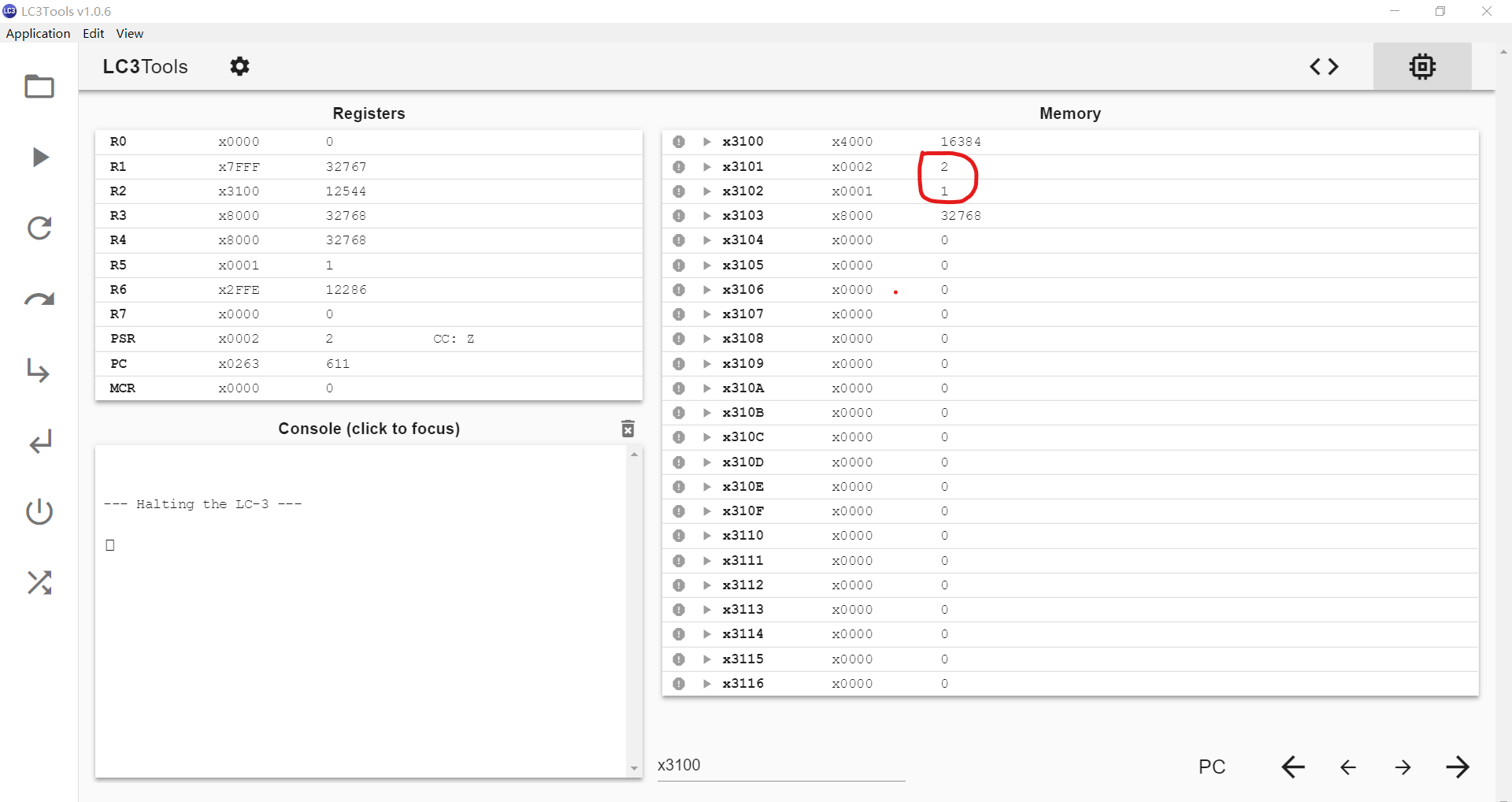
三、

Self-test：

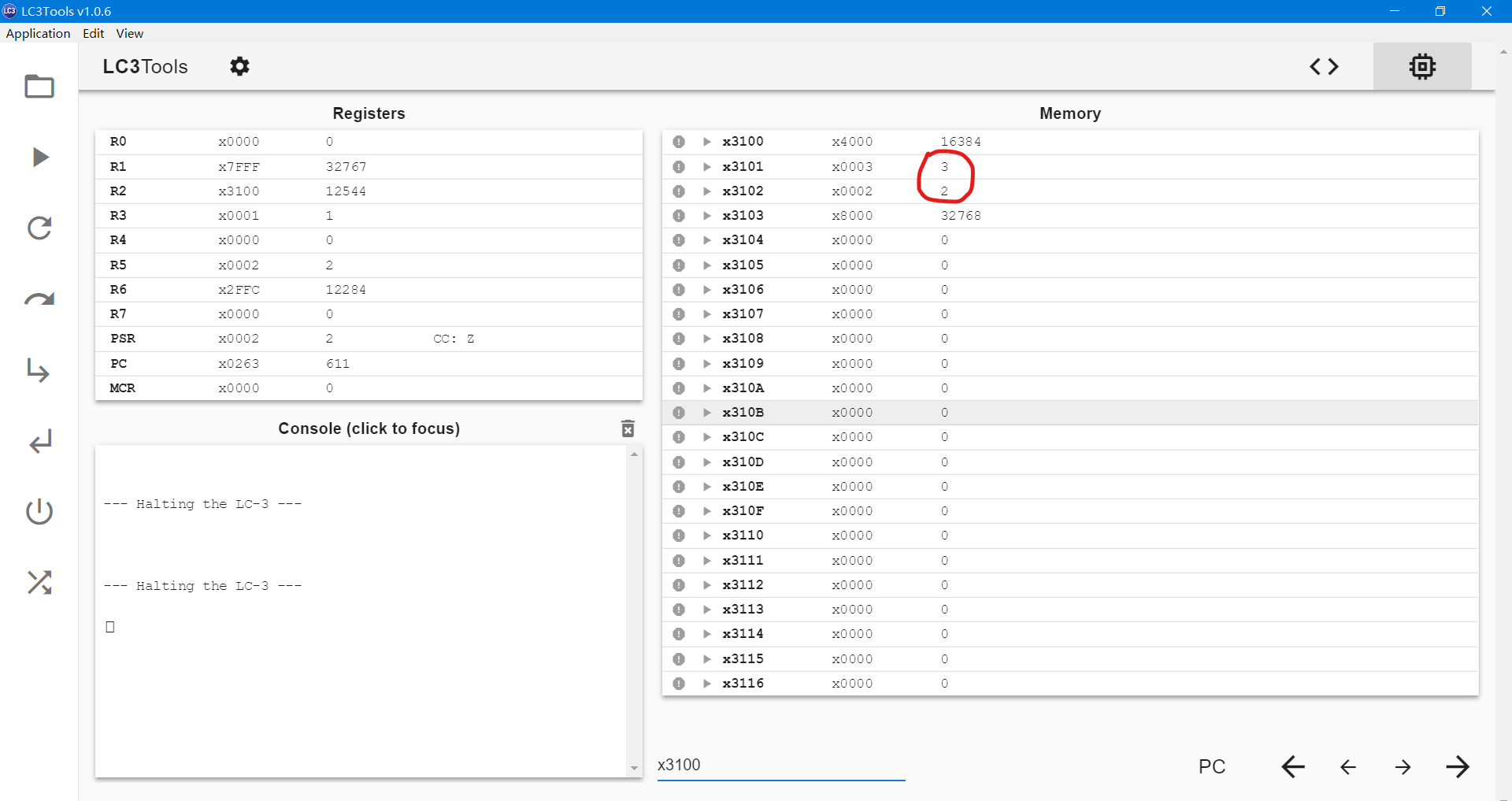
第一次调试令rotate=0，验证初始化程序是可行的

第二次调试令rotate=1，验证冒泡部分的程序也是可行的

第三次调试令rotate=3，验证算法的普适性，即只需要改变rotate的值即可实现前n个值的冒泡。



（Rotate=2）



（Rotate=3）