**Lab1实验报告**

1. **实验目的**

用LC-3机器码实现求指定二进制数的低地址位指定位数中1的个数

（两个数据均人为输入）

1. **实验原理**
2. 数据处理：利用LDI、ST等指令从指定地址载入数据或向指定地址存入数据
3. 左移：巧妙使用ADD指令将R1(存有数据A的寄存器)中的数据\*2，使R1中的二进制数每一位左移一位，方便对指定位数作判断
4. 判断1或0的方法：在我们需要的位置移到16个位置的最左端时，若这个数是1，则此时R1中存储的数是一个负数；此时，可以利用BR指令，进行判断并在后续步骤计数
5. **实验步骤**

核心的原理和步骤已经在“实验原理”这一部分阐述，下面说说此次实验中思路变化以及实际写机器码和调试的时候遇到的一些问题和改进方法：

1. 最开始我的思路并不是通过左移来对指定的位数作判断，而是想实现右移操作。当时认为右移更一目了然，程序也更会简化。与左移不同的是，右移需要先对数据的奇偶性作判断，（奇数加一，偶数不变）再将数据除以2实现。在实际操作中，因为左移能省去自己构造除法操作、判断奇偶性的麻烦，所以我最后选择了左移。
2. 关于如何正确使用数据B，我的思路是分为两个部分：

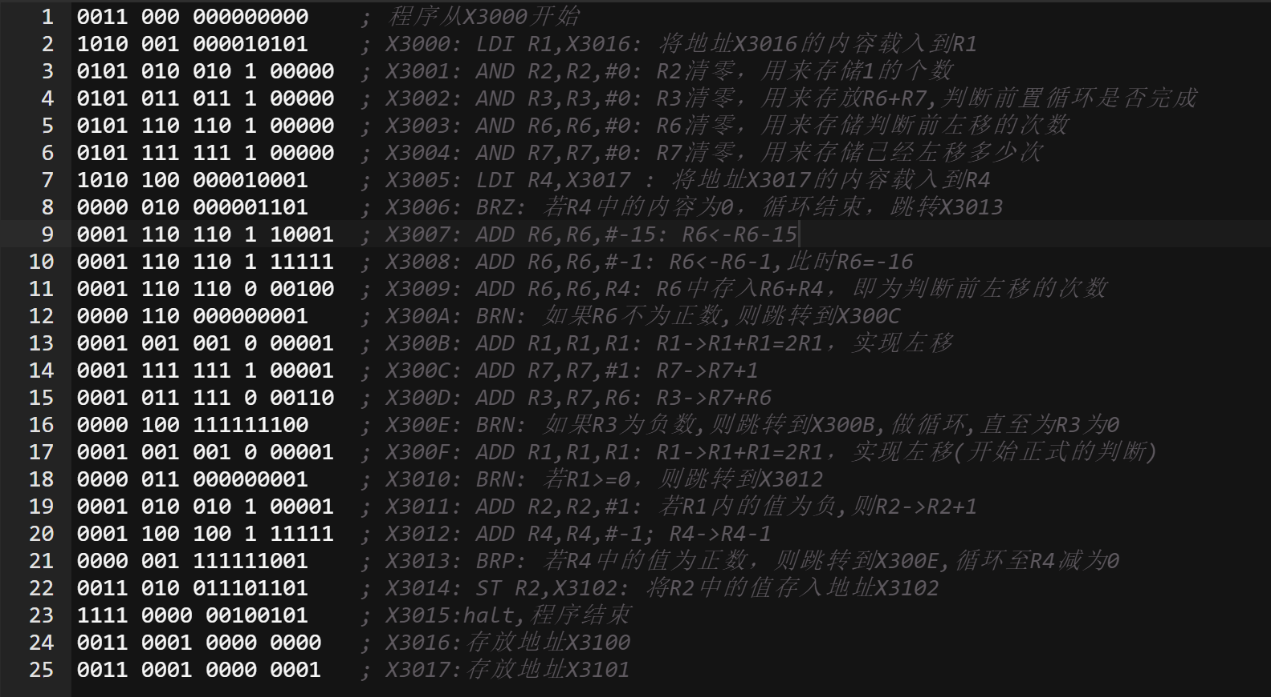
先将R1内的数据左移（16-B）次，再开始“左移+判断+计数”相结合地作B次操作。所以整个实验一共用到了5次BR(判断)跳转指令。在实现过程中也走了一些弯路，比如跳转条件是大于0还是大于等于0没有想清楚，或者密集的跳转指令相互冲突，写成了死循环。

1. 寄存器的使用：

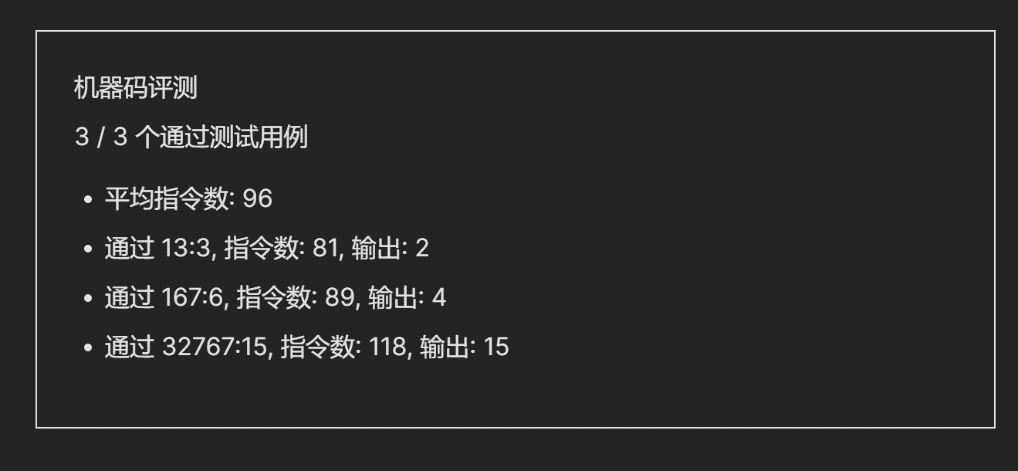
在使用寄存器方面，最初写程序时我把每个功能都各用一个寄存器，后来发现因为我的滥用，将要面临8个寄存器不够用的问题。观察发现，有些指令操作后（例如ADD指令），涉及到的源寄存器及其中的数据在后面不再需要，可以用这个寄存器作为目的寄存器，解决问题的同时还减少了对其他寄存器的清零操作或者调用，使程序更简便。

1. **实验结果**

如下图为实验机器码：



使用助教提供的在线测评网站：



在测试中，扩大测试样例范围，测试了A=1;A=32767;B=0;B=6等数据，在这些测试中用例都通过了。