# 一、实验目的与要求

（1）分析和理解实验指定的问题。

（2）利用LC-3的机器代码设计实现相关程序。

（3）通过LC-3仿真器调试和运行相关程序并得到正确的结果。

# 二、实验内容与方法

# 利用LC-3的机器代码计算一个16位的字中有多少位是‘1’。

# 程序从x3000开始。

# 需计算的字存储在x3100。

# 计算的结果存储在x3101。

# 三、实验步骤与过程

（依照实验内容，逐条撰写实验过程与实验所得结果：包括程序总体设计，核心数据结构及算法流程，调试过程。请附上核心代码，及注意格式排版的美观。实验提交时，以上为评分依据，请不删除本行）

**程序总体设计及算法流程**

问题是计算一个16位的字中有多少位是‘1’，初步思考一下，解决这个问题需要进行计数，判断是不是‘1’，以及一个16次的循环。做出流程图如图1所示。

用R0作为计数器，R1控制循环的次数，R2存储需要判断的字。

先用AND指令让R0和R1赋值为0，接着需要让R1的值为16，因为ADD指令的立即数寻址模式imm5只能表示-16到15，所以不能直接把16赋值给R1，我的方法是先把8赋值给R1，即用ADD指令让R0=R0+8，之后用ADD指令让R0=R0+R0，即可让R0的值为16。后用LD指令把内存地址为x3100的内容读进R2。主要问题即计算字中有多少个是‘1’，我采取的方法是通过判断R2的值的正负来确定，如果是负数，那么第一位就是‘1’，之后使其乘2，即让其各位左移一位，再次重复判断第一位，直到累计判断了16次。

最后用ST指令把R0的值存进地址为x3101的内存单元。

****

图1 算法流程图

**核心数据结构**

核心变量有三个，R0作为计数器，负责计数字中为1的个数，R1控制循环的次数，从16开始自减，R2存储需要判断的字，每一次循环之后都做一次与自己相加的操作。

程序一开始，先让R0和R1赋值为0，即与0相与，然后让R1赋值为8，之后R1做一次与自己相加的操作使其值变成16，之后把内存地址为x3100的数据存进R2。

在主循环中，通过判断R2是否是负数来计数，如果是负数，那么让R0自增，之后让R2与自己相加，然后让R1自减。

最后把R0的值存进内存地址为x3101的内存单元。

**实验及调试过程**

首先在LC-3Editor中打出机器码，并编译成二进制程序，如图2所示。

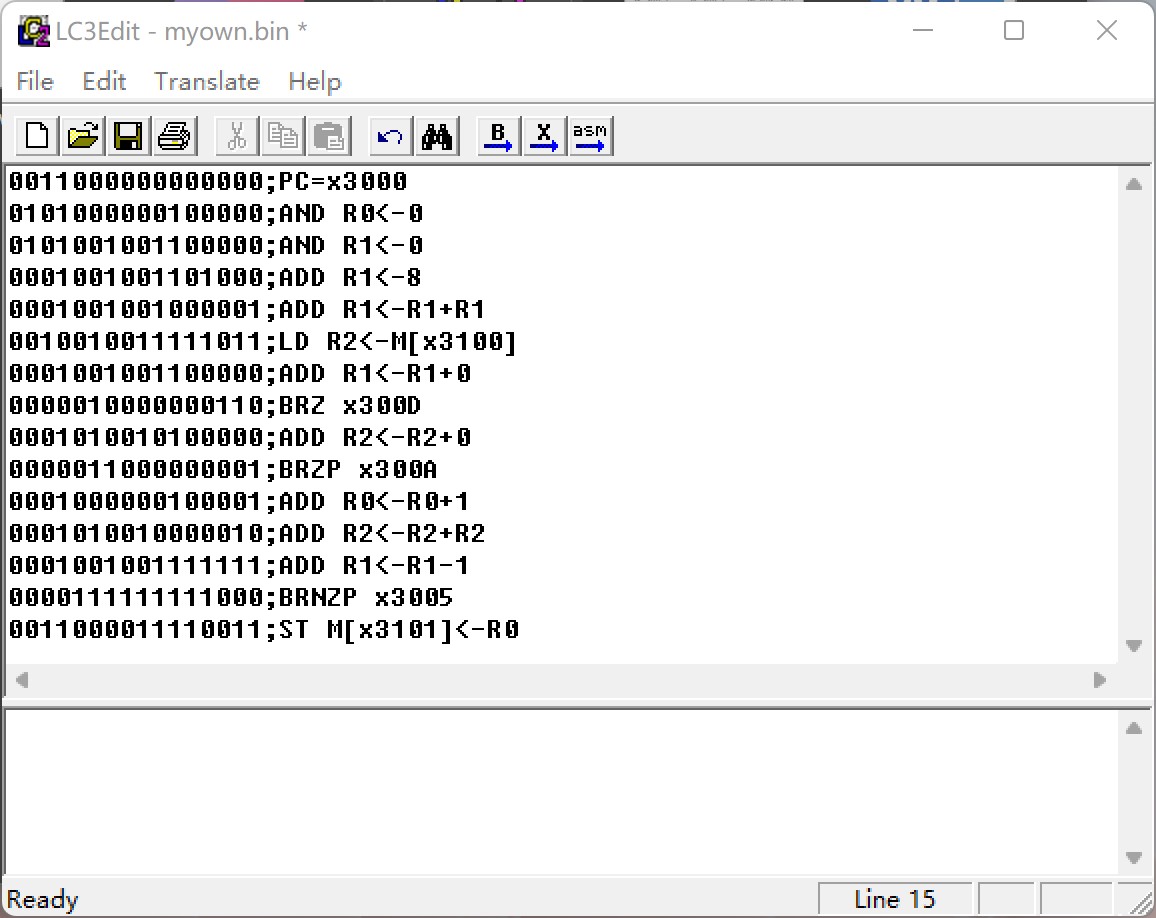


图2

然后在LC-3Simulate中加载程序，并在x300E处设置断点，如图3所示。

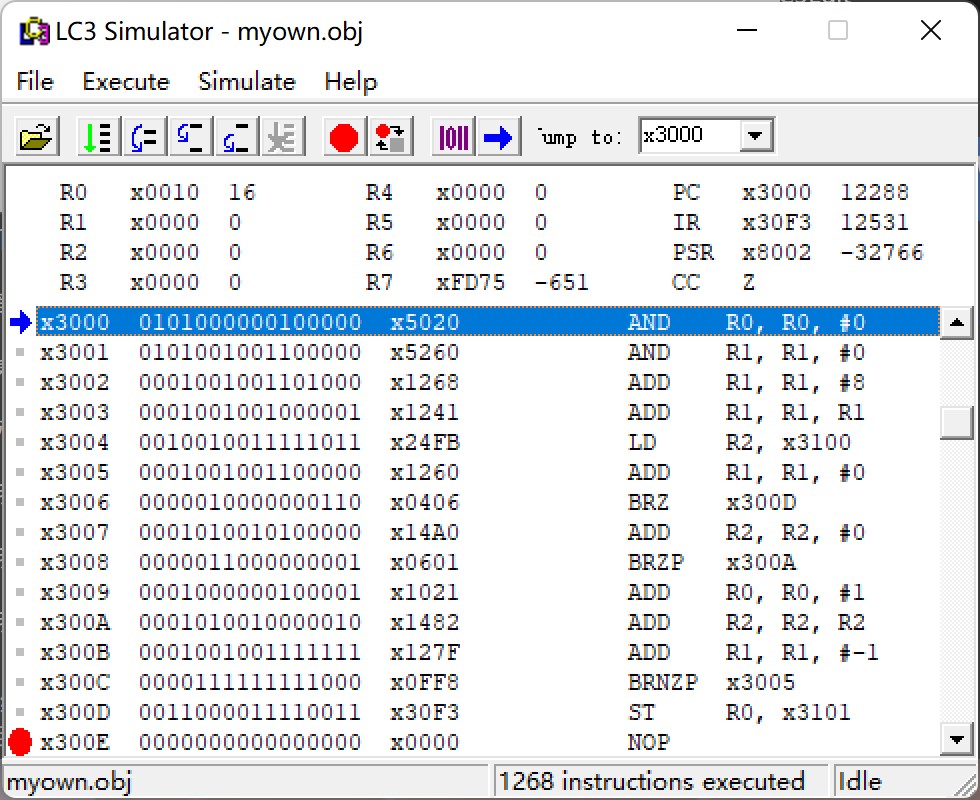


图3

第一次运行测试，将x3100地址的数据设置为x0001，点击运行，如图4所示。

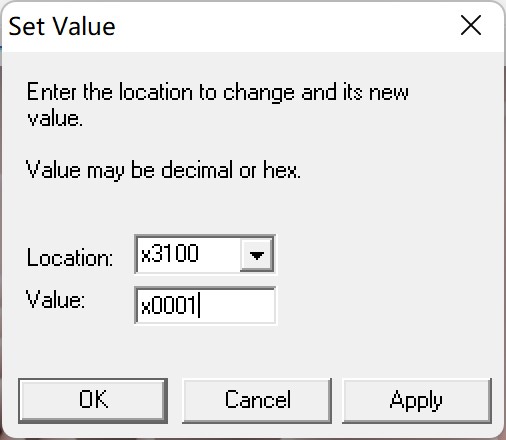


图4

然后发现地址x3101的数据为0，这显然不符合预期结果。

于是开始调试，逐步执行代码。

执行过程中发现R0、R1和R2值变化符合预测，但在执行BR指令时，跳转的指令不对，发现是跳转地址计算错误，并且忽略了地址为16进制数，于是进行修改。

再次测试，得出正确结果，x3101的值为x0001，如图5所示。

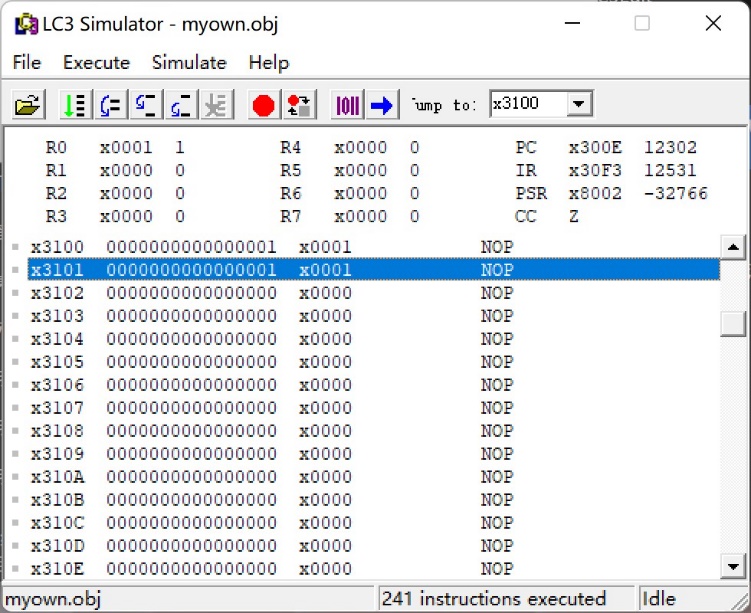


图5

再让x3100的值为xFFFF，进行测试，同样得出了正确结果，如图6所示。

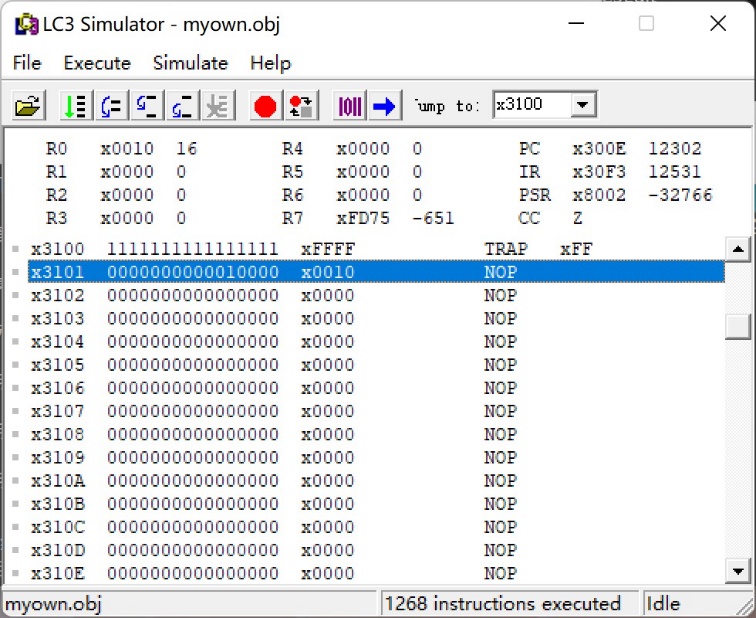


图6

# 四、实验结论或体会

（撰写实验收获及思考）

这次的实验主要是用机器码编程，由于之前学过C、C++以及Python这些高级编程语言，所以分析问题，找出解决问题的算法难度不大，但使用高级语言编程是在一个很高的层次进行解决问题了，转到机器语言编程时，意味必须从底层内存出发，从核心数据的存储出发解决问题，更加强调以计算机思维去看待问题和解决问题。

在这次实验中，我学会了利用LC-3的机器代码设计实现相关程序，通过LC-3仿真器调试和运行相关程序并得到正确的结果。通过这次机器码实验，我更加了解了LC-3仿真器的调试和运行功能，对计算机底层数据存储以及底层运行有了更深的认识，希望在将来能够更多的学习高级编程语言在计算机底层的实现机制。

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。