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

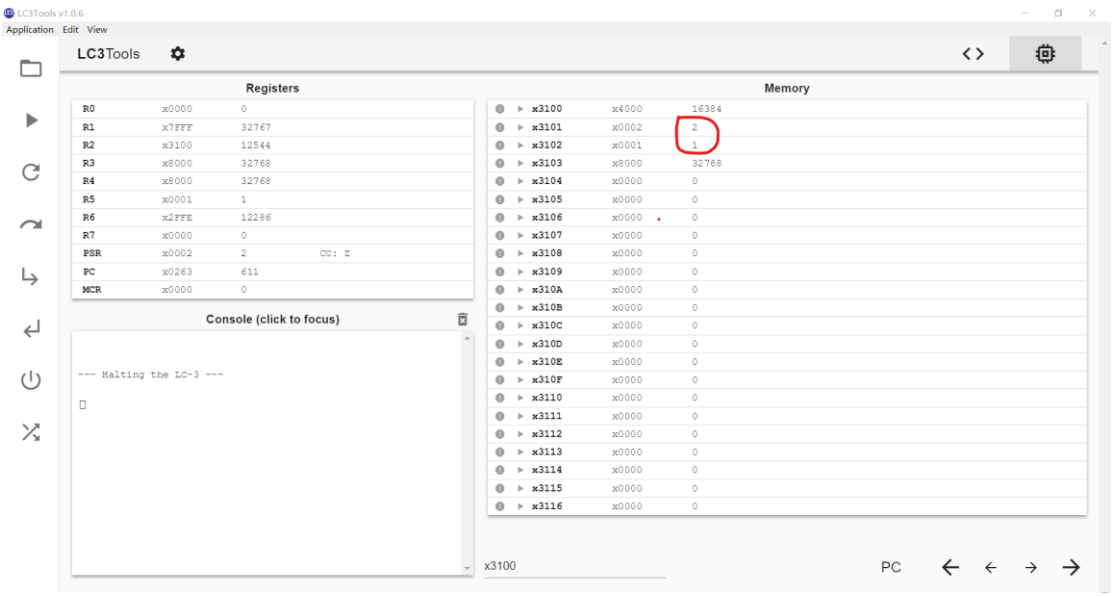
三、

Self-test：

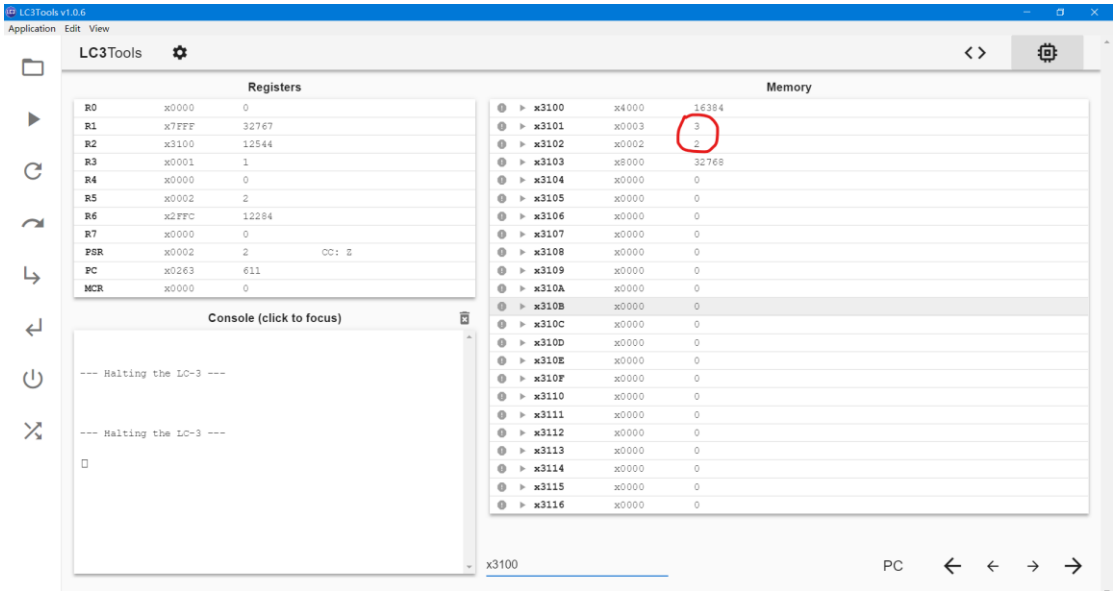
第一次调试令 rotate=0，验证初始化程序是可行的

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）


LC3Tools v1.0.6
Application Edit View
LC3Tools
Registers
R0 x0000 0
R1 x7FFF 32767
R2 x3100 12544
R3 x0001 1
R4 x0000 0
R5 x0002 2
R6 x2FFC 12284
R7 x0000 0
PSR x0002 2 CC: Z
PC x0263 611
MCR x0000 0
Console (click to focus)
--- Halting the LC-3 ---
--- Halting the LC-3 ---
Memory
x3100 x4000 16384
x3101 x0003 3
x3102 x0002 2
x3103 x8000 32768
x3104 x0000 0
x3105 x0000 0
x3106 x0000 0
x3107 x0000 0
x3108 x0000 0
x3109 x0000 0
x310A x0000 0
x310B x0000 0
x310C x0000 0
x310D x0000 0
x310E x0000 0
x310F x0000 0
x3110 x0000 0
x3111 x0000 0
x3112 x0000 0
x3113 x0000 0
x3114 x0000 0
x3115 x0000 0
x3116 x0000 0
x3100
PC

（Rotate=3）