

Lab1 实验报告

一 . 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)

二 . 实验原理

1. 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
2. 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
3. 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

三 . 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。
3. 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。

四 . 实验结果

如下图为实验机器码：

```

1 0011 000 0000000000 ; 程序从X3000开始
2 1010 001 000010101 ; X3000: LDI R1,X3016: 将地址X3016的内容载入到R1
3 0101 010 010 1 00000 ; X3001: AND R2,R2,#0: R2清零, 用来存储1的个数
4 0101 011 011 1 00000 ; X3002: AND R3,R3,#0: R3清零, 用来存放R6+R7,判断前置循环是否完成
5 0101 110 110 1 00000 ; X3003: AND R6,R6,#0: R6清零, 用来存储判断前左移的次数
6 0101 111 111 1 00000 ; X3004: AND R7,R7,#0: R7清零, 用来存储已经左移多少次
7 1010 100 000010001 ; X3005: LDI R4,X3017: 将地址X3017的内容载入到R4
8 0000 010 000001101 ; X3006: BRZ: 若R4中的内容为0, 循环结束, 跳转X3013
9 0001 110 110 1 10001 ; X3007: ADD R6,R6,#-15: R6<-R6-15
10 0001 110 110 1 11111 ; X3008: ADD R6,R6,#-1: R6<-R6-1,此时R6=-16
11 0001 110 110 0 00100 ; X3009: ADD R6,R6,R4: R6中存入R6+R4, 即为判断前左移的次数
12 0000 110 000000001 ; X300A: BRN: 如果R6不为正数, 则跳转到X300C
13 0001 001 001 0 00001 ; X300B: ADD R1,R1,R1: R1->R1+R1=2R1, 实现左移
14 0001 111 111 1 00001 ; X300C: ADD R7,R7,#1: R7->R7+1
15 0001 011 111 0 00110 ; X300D: ADD R3,R7,R6: R3->R7+R6
16 0000 100 111111100 ; X300E: BRN: 如果R3为负数, 则跳转到X300B, 做循环, 直至为R3为0
17 0001 001 001 0 00001 ; X300F: ADD R1,R1,R1: R1->R1+R1=2R1, 实现左移(开始正式的判断)
18 0000 011 000000001 ; X3010: BRN: 若R1>=0, 则跳转到X3012
19 0001 010 010 1 00001 ; X3011: ADD R2,R2,#1: 若R1内的值为负, 则R2->R2+1
20 0001 100 100 1 11111 ; X3012: ADD R4,R4,#-1; R4->R4-1
21 0000 001 111111001 ; X3013: BRP: 若R4中的值为正数, 则跳转到X300E, 循环至R4减为0
22 0011 010 011101101 ; X3014: ST R2,X3102: 将R2中的值存入地址X3102
23 1111 0000 00100101 ; X3015: halt, 程序结束
24 0011 0001 0000 0000 ; X3016: 存放地址X3100
25 0011 0001 0000 0001 ; X3017: 存放地址X3101

```

使用助教提供的在线测评网站:

机器码评测

3 / 3 个通过测试用例

- 平均指令数: 96
- 通过 13:3, 指令数: 81, 输出: 2
- 通过 167:6, 指令数: 89, 输出: 4
- 通过 32767:15, 指令数: 118, 输出: 15

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。