中国科学技术大学计算机学院
《计算机系统概论》报告

实验题目：Lab 03

学生姓名：高楚晴

学生学号：PB18111688

完成日期：2019.12.5

实验目的

使用汇编语言实现求两个正数的最大公约数，两个数分别在R0和R1给出，将结果存放在R0中，执行过程中不改变R7的值。

设计思路

采用Stein算法作为基本框架，可表示为

$$gcd(2k_1+1,2k_2+1)=gcd(2k_1+1,2k_2-2k_1)(k_2>k_1)$$

$$gcd(2k_1+1,2k_2+1)=gcd(2k_1-2k_2,2k_2+1)(k_1>k_2)$$

$$gcd(2k_1,2k_2+1)=gcd(k_1,2k_2+1)$$

$$gcd(2k_1+1,2k_2)=gcd(2k_1+1,k_2)$$

$$gcd(2k_1,2k_2)=2*gcd(k_1,k_2)$$

其中k_1和k_2为自然数，且$gcd(x,y)$内参数均为正数，实际实现中由于lc3汇编语言没有移位操作，为减少执行代码条数，除二操作由查找表完成。其中乘二操作由一个寄存器记录该操作执行次数，最终统一完成。当最终R0与R1值相同时运算结束，将R0按照记录次数执行倍乘操作，最终R0值即为最小公倍数。

在初次编写时使用了每次进行上述递归操作中的一个，每次分别判断奇偶情况进行选择。在后续的尝试中发现在出现一奇一偶的情况下，将偶数除二直至其为奇数再返回继续执行会使代码执行数减少，因此采用这种方法。

其他基于原Stein的优化尝试均表现为性能劣于原Stein算法，具体在代码测试部分给出。

代码讲解

查找表存储形式

在主程序后内存部分将1~32767对应除二值进行存放，具体存储形式如下（因篇幅过长仅展示部分）。

```
OFFSET  .FILL   x3029       ;原数字与除二值偏移量
        .FILL   x1
        .BLKW   1
        .FILL   x2
        .BLKW   1
        .FILL   x3
        .BLKW   1
        .FILL   x4
        .BLKW   1
        .FILL   x5
        .BLKW   1
```

分类递归

```
        LD   R5,OFFSET          ;存放查找表对应偏移量
        LD   R4,ONE             ;将最终乘二次数初始值置1，便于最终递减后检测条件码
START   NOT  R6,R0              ;检测R0与R1是否相等，相等则退出
        ADD  R6,R6,#1
        ADD  R6,R1,R6
```

```
        BRz FINAL
        AND R3,R0,#1            ;检测R0的奇偶情况
        BRz EVEN0
        AND R3,R1,#1            ;R0为奇数时，检测R1奇偶情况
        BRnp    LOOP
LOOP2   ADD R1,R1,R5            ;R0奇,R1偶
        LDR R1,R1,#0
        AND R3,R1,#1
        BRz LOOP2               ;R1持续除2直至R1为奇数
        BR  START
LOOP    ADD R6,R6,#0            ;R1,R0均为奇数，作差
        BRp LOOP1               ;更相减损之后递归返回
        NOT R6,R6               ;R0>R1，以差值替代R0
        ADD R6,R6,#1
        ADD R0,R6,#0
        BR  START
LOOP1   ADD R1,R6,#0            ;R1>R0，以差值替代R1
        BR  START
EVEN0   ADD R0,R0,R5            ;even0表示R0为偶数的情况
        LDR R0,R0,#0
        AND R3,R1,#1
        BRz EVEN1
LOOP3   AND R3,R0,#1        ;R0偶数，R1奇数
        BRp START
        ADD R0,R0,R5        ;R0持续除2直至R0为奇数
        LDR R0,R0,#0
        BR  LOOP3
EVEN1   ADD R1,R1,R5        ;R0,R1均为偶数
        LDR R1,R1,#0
        ADD R4,R4,#1
        BR  START
FINAL   ADD R4,R4,#-1       ;最终结果进行R4对应次数的乘2操作
        BRz END
        ADD R0,R0,R0
        BR  FINAL
```

调试分析

正数算法

经调试代码编译正常通过。

由于使用查找表，空间复杂度较大，但空间占用情况与输入两正数值无关，对单次测试求$gcd(m,n)$,其实际空间复杂度是与m,n无关的常数，即为$O(1)$,但若引入参数 t 表示允许输入正数的最大值，则空间复杂度为$O(t)$。

该算法基于Stein算法，时间复杂度与欧几里得算法时间复杂度相同，为$O(log(m[R0]+m[R1]))$

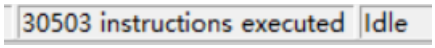
零和负数的情况

根据最大公约数的定义，最大公约数只讨论正整数范围内的情况，因此在输入不确定的情况下先检测R0和R1的条件码，若为两个正数则正常执行，若为两个负数则均取其相反数执行，其他情况将R0置为0，通过返回0表示输入不合法。

代码测试

通过网页工具生成随机数，修改代码进行测试，为方便存储，单次测试使用200个随机数按顺序分为100对正数计算其最大公约数，观察其指令数结果。

为比较各种改进算法的性能，仅取一组数的平均结果作为挑选算法的标准。根据随机手动赋值执行时原Stein算法的执行过程试图针对一些特例进行改进，共提出以下几种思路，对随机生成的一百组数的总指令数（包含内部测试LD测试数以及循环判断执行的指令数）分别如下，数据来源由图给出部分，为便于观察全部测试结果不由图片给出。



基础方法	奇偶连除算法[1]	一次更相减损[2]	多次更相减损[3]	循环更相减损[4]
34478	30503	48682	67249	84907

说明:

1：指出现一奇一偶时连除直至偶数变为奇数再返回

2：指对两数在最开始执行一次更相减损（针对两数极为接近的情况）

3：试图对差值较大的数进行改进，实际使用了两次，到此意识到事实上并不存在这种情况更优的情况 -> 除二操作更优

4：每次递归返回时附加一次更相减损，不针对特定情况，仅作以尝试

根据上述情况选择奇偶连除的算法，增大数据量进行多次测试，执行结果如下。

次数	1	2	3	4	5	6	平均值
指令数	30503	30199	30278	30192	29626	30124	30153

其中测试所用指令对每组数大约为10条，因此实际总平均约为290条指令。

实验总结

通过本次实验对Stein算法以及查找表的实现有了更深的理解，领会了空间换时间的基本思想，同时掌握了自行测试的基本方法。在代码编写过程中对求最大公约数的几种不同实现方法的优劣有了更直观且深刻的意识。此外通过学习欧几里得算法，对计算时间复杂度有了更好地掌握。

附录

PB18111688_高楚晴_Lab03.asm

PB18111688_Lab03.obj