

# RISCV实现的GCD算法

这里使用的模拟器是<http://venus.cs61c.org/>

这里的实现思路与LC-3实现的思路大致相同：

运用辗转相除法，其中实现 $a \bmod b$ 的过程采用下面的方法：

a一次减b，用R0来储存前一个相减的值，用R2来储存相减之后的值，如果相减之后的值小于0，则回退，并且比较ab大小，如果 $a < b$ ，则交换ab重复此操作，如果等于0则结束并且判定b为最大公约数。

具体实现代码如下：

```
addi ra,ra,-50
    andi a2,a2,0
    andi a3,a3,0
    addi a4,a4,0
    andi a5,a5,0
    andi a6,a6,0
    addi a2,a0,0
JUDGE1:
bgtz a0,JUDGERS
beqz a0,ZERO
bltz a0,REVER0

JUDGERS:
addi a1,a1,0
bgtz a1,GCD
beqz a1,ZERO
bltz a1,REVER1

ZERO:
andi a0,a0,0
addi ra,ra,-50
ret

REVER0:
not a0,a0
andi a0,a0,1
jal JUDGERS

REVER1:
not a1,a1
addi a1,a1,1

GCD:
addi a5,a5,1
addi a0,a2,0
not a3,a1
addi a3,a3,1
add a2,a0,a3
```

```

    beqz a2,THEEND
    bgtz a2,GCD
    addi a6,a6,1
    addi a2,a0,0
    addi a4,a0,0
    addi a0,a1,0
    addi a1,a4,0
    addi a2,a0,0
    jal GCD

    THEEND:
    addi a0,a2,0
    addi a0,a1,0
    andi ra,ra,0
    addi ra,ra,-50
    ret

```

代码分析：

【part1】

```

    addi ra,ra,-50
    andi a2,a2,0
    andi a3,a3,0
    addi a4,a4,0
    andi a5,a5,0
    andi a6,a6,0
    addi a2,a0,0

```

初始化一些需要用到的寄存器，其中R5,R6用于计算经过了多少循环，便于分析性能与用到的代码量。需要计算gcd的两个输入值存入R0和R1中。

【part2】

```

JUDGE1:
    bgtz a0,JUDGERS
    beqz a0,ZERO
    bltz a0,REVER0

    JUDGERS:
    addi a1,a1,0
    bgtz a1,GCD
    beqz a1,ZERO
    bltz a1,REVER1

    ZERO:
    andi a0,a0,0
    addi ra,ra,-50
    ret

    REVER0:
    not a0,a0
    andi a0,a0,1

```

```

addi a2,a0,0
jal JUDGERS

REVER1:
not a1,a1
addi a1,a1,1

```

特殊情况处理，对于输入为0或者是负数的情况进行讨论，如果输入为负数，则将其取相反数后进行gcd运算，如果输入为0，则直接输出0.

```

GCD:
addi a5,a5,1
addi a0,a2,0
not a3,a1
addi a3,a3,1
add a2,a0,a3
beqz a2,THEEND
bgtz a2,GCD

```

将R0的值减去R1的值，将所得到的值存入R2中，如果得到的值大于0，则将R2的值给R0并且继续这样的操作，否则后退，并且交换R0和R1中的值。如果为0，则结束循环，将R1的值赋给R0作为输出。

```

addi a6,a6,1
addi a2,a0,0
addi a4,a0,0
addi a0,a1,0
addi a1,a4,0
addi a2,a0,0
jal GCD

```

交换R0和R1的值。

```

THEEND:
addi a0,a2,0
addi a0,a1,0
andi ra,ra,0
addi ra,ra,-50
ret

```

#### 【测试用例1】

a0=34, a1=56

a0	2
(x10)	

#### 【测试用例2】

a0=-10 a1=25

测试结果：

a0  
(x10) 5

### 【测试用例3】

a0=34,a1=0

测试结果：

---

a0  
(x10) 0

### 【遇到的问题】

在这里发现ret指令并不能使程序停止，而是跳转到ra中的值中,所以开始导致代码出现死循环，一个有效的解决方法是在ret指令之前给ra赋值为一个代码块之外的地址，使得程序可以停下。