# Lab2\_利用LC-3实现gcd(最大公约数算法)

#### 实验目的

- 利用LC-3实现特定的算法
- 利用跳转指令和寄存器传值实现函数调用

### 算法设计

采用递归算法, 计算最大公约数

C源代码如下

```
int gcd(int x,int y){
    if(x==0||y==0){
        return 0;
    }
    if(x==y){
        return x;
    }
    else if(x>y){
        return gcd(x-y,y);
    }
    else{
        return gcd(y,x);
    }
}
```

## 汇编程序设计

加上注释的汇编源码如下

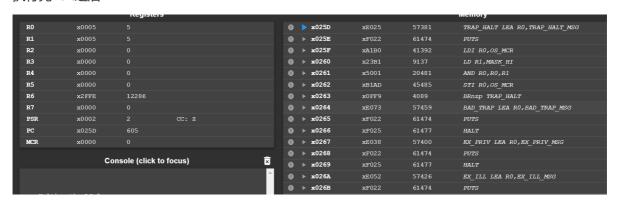
```
.ORIG x3000
AND RO,RO,#0 ;clear RO
AND R1,R1,#0 ;clear R1
ADD RO, RO, #10 ; load a number in RO
ADD R1,R1,#15 ; load a number in R1
GCD: AND R2,R2,#0 ;clear R2
ADD R2,R2,R1; R2=R1
NOT R2, R2 ; R2=! R2
ADD R2, R0, R2
ADD R2, R2, \#1; R2=R0-R1
BRZ STOP ;R0=R1
BRN SWAP ; change the value store in RO and R1
ADD R0,R2,\#0 ;else R0=R0-R1
BRNZP GCD ; call the gcd function until RO=R1
STOP: HALT
SWAP: ADD R2,R0,\#0
ADD R0,R1,#0
ADD R1,R2,#0 ;swap the value between R0 and R1
BRNZP GCD
.end
```

解释一下,第7行到第10行,实际上再计算R0-R1,并将结果存到R2中对于R2的各种情况,11,12行做了判断第14,19行无条件跳转,相当于递归调用注意调用之间需要将参数保存在R0,R1当中

#### 检验

R0=10,R1=15

#### 执行完halt之后



#### R0=3.R1=7

