

## 实验二-RISCV实现

### 实验目的

- 利用RISCV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);
    }
}
```

这里函数调用由于参数个数和存储空间有限，我们将参数直接放在寄存器中，使用跳转实现函数调用过程的模拟

### 汇编程序设计

加上注释的汇编代码如下

```
addi x1,x0,10 ;load the first parameter into register x1
addi x2,x0,15 ;the second parameter into register x2
loop: beq x2,x1,END ;if x2=x1,then return
blt x2,x1,swap #if x2<x1,then x1=x1-x2;else x2>x1
sub x2,x2,x1 ;x2=x2-x1
j loop ;funccall
END: ret
swap: sub x1,x1,x2 ;x1=x1-x2
j loop ;funccall
```

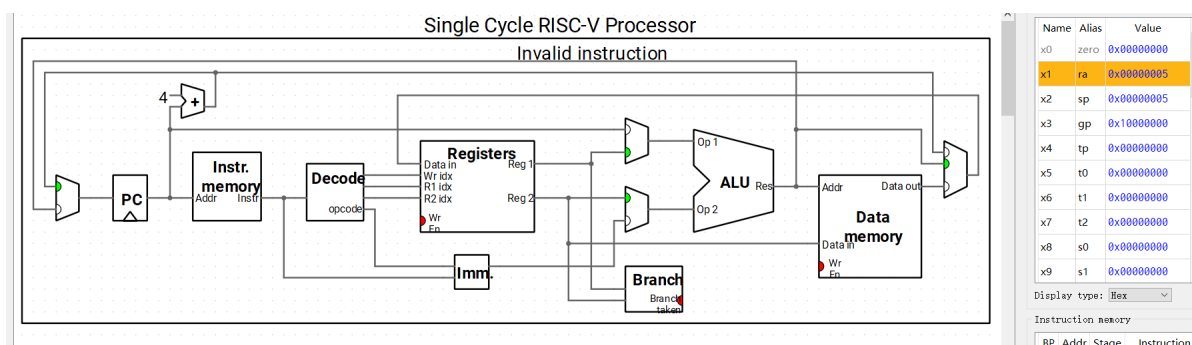
这里直接向x1，x2加载值，这是gcd的参数

内存中，指令的布局如下

0:	00a00093	addi x1 x0 10
4:	00f00113	addi x2 x0 15
00000008 <loop>:		
8:	00110863	beq x2 x1 16 <END>
c:	00114863	blt x2 x1 16 <swap>
10:	40110133	sub x2 x2 x1
14:	ff5ff06f	jal x0 0x8 <loop>
00000018 <END>:		
18:	00008067	jalr x0 x1 0
0000001c <swap>:		
1c:	402080b3	sub x1 x1 x2
20:	fe9ff06f	jal x0 0x8 <loop>

## 测试与验证

x1=10 x2=15



x1=7 x2=3

