

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 R0,R0,#0 ;clear R0
AND R1,R1,#0 ;clear R1
ADD R0,R0,#10 ;load a number in R0
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 R0 and R1
ADD R0,R2,#0 ;else R0=R0-R1
BRNZP GCD ;call the gcd function until R0=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

Registers

R0	x0005	5
R1	x0005	5
R2	x0000	0
R3	x0000	0
R4	x0000	0
R5	x0000	0
R6	x2FFE	12286
R7	x0000	0
PSR	x0002	2
PC	x025D	605
MCR	x0000	0

Console (click to focus)

Memory

025D	x025	57381	TRAP_HALT LEA R0,TRAP_HALT_MSG
025E	x022	61474	PUTS
025F	xA1B0	41392	LDI R0,OS_MCR
0260	x23B1	9137	LD R1,MASK_HI
0261	x5001	20481	AND R0,R0,R1
0262	xB1AD	45485	STI R0,OS_MCR
0263	x0FF9	4089	BRnzp TRAP_HALT
0264	xE073	57459	BAD_TRAP LEA R0,BAD_TRAP_MSG
0265	x022	61474	PUTS
0266	x025	61477	HALT
0267	xE038	57400	EX_PRIV LEA R0,EX_PRIV_MSG
0268	x022	61474	PUTS
0269	x025	61477	HALT
026A	xE052	57426	EX_ILL LEA R0,EX_ILL_MSG
026B	x022	61474	PUTS

R0=3,R1=7

Registers

R0	x0001	1
R1	x0001	1
R2	x0000	0
R3	x0000	0
R4	x0000	0
R5	x0000	0
R6	x0000	0
R7	x0000	0
PSR	x8002	32770
PC	x300D	12301
MCR	x0000	0

Console (click to focus)

Memory

3000	x5020	20512	AND R0,R0,#0
3001	x5260	21088	AND R1,R1,#0
3002	x1023	4131	ADD R0,R0,#3
3003	x1267	4711	ADD R1,R1,#7
3004	x54A0	21664	GCD: AND R2,R2,#0
3005	x1481	5249	ADD R2,R2,R1
3006	x94BF	38079	NOT R2,R2
3007	x1402	5122	ADD R2,R0,R2
3008	x14A1	5281	ADD R2,R2,#1
3009	x0403	1027	BRZ STOP
300A	x0803	2051	BRN SWAP
300B	x10A0	4256	ADD R0,R2,#0
300C	x0FF7	4087	BRNZP GCD
300D	x025	61477	STOP: HALT
300E	x1420	5152	SWAP: ADD R2,R0,#0
300F	x1060	4192	ADD R0,R1,#0
3010	x1023	4131	ADD R1,R2,#0