

实验二

一、汇编码

```
.ORIG X3000
AND R0 R0 #0;初始化 0
AND R1 R1 #0; 用 R1 来代表符号位, 1 为正,-1 为负
LD R6 X00FF ; X3003+X00EB=X3100 R6=N,计数器
LD R5 X0FC ; -4096->R3 X3100
LD R4 X0FC ; -4096->R4
ADD R6 R6 #-1;计数器先-1 方便后续判断
ADD R0 R0 #3; R0 代表 fn,f1 初始化为 3
ADD R1 R1 #1; R1 初始化为 1, 正数
ADD R6 R6 #0;
BRz #14;如果 N==0, 则得到最终结果, 可以直接存储->36
AND R7 R0 #7; 判断是否整除,结果为零则可以整除
BRz #19 ;是 0, 可以整除, 调到反转符号->33
ADD R3 R0 #0; R3==R0 一直减 10 直到为负数
ADD R3 R3 #-10;一直减 10 直到为负数
BRzp #-2;
ADD R3 R3 #10; 得到 R0 的个位数
AND R7 R3 #8;判断是否为 8
BRp #13; 是 8 则转到翻转符号 goto 33
ADD R0 R0 R0 ;fn=2fn
ADD R1 R1 #0; 判断符号位是啥
BRn #2 ;负号则 goto 25 行
ADD R0 R0 #2; fn=2fn+2
BR #1;
ADD R0 R0 #-2;fn=2fn-2
ADD R3 R0 #0; 这里开始判断是否越界
ADD R3 R3 R5;
BRp #-2;
ADD R0 R3 R4;
ADD R6 R6 #-1;N=N-1;
BRz #4 ;if N==0,程序结束, 得到结果存储 goto36
BR #21 ;否则继续循环, goto->12
NOT R1 R1;
ADD R1 R1 #1;这里两步完成反转;
BR #-16 ;无条件跳转到 20 行
ST R0 X0E0;
TRAP X25;
.END
```

二、实验结果

①	▶ x3100	x0000	61440
②	▶ x3101	x1000	4096
③	▶ x3102	x0065	101
④	▶ x3103	x0FFE	4094
⑤	▶ x3104	x0000	0
⑥	▶ x3105	x0000	0
⑦	▶ x3106	x0000	0

①	▶ x3100	x0000	61440
②	▶ x3101	x1000	4096
③	▶ x3102	x0050	80
④	▶ x3103	x0FFE	4094
⑤	▶ x3104	x0000	0
⑥	▶ x3105	x0000	0
⑦	▶ x3106	x0000	0
⑧	▶ x3107	x0000	0

①	▶ x3100	x0000	61440
②	▶ x3101	x1000	4096
③	▶ x3102	x0008	8
④	▶ x3103	x018A	394
⑤	▶ x3104	x0000	0
⑥	▶ x3105	x0000	0
⑦	▶ x3106	x0000	0
⑧	▶ x3107	x0000	0
⑨	▶ x3108	x0000	0

①	▶ x3100	x0000	61440
②	▶ x3101	x1000	4096
③	▶ x3102	x0014	20
④	▶ x3103	x07FE	2046

结果分析可知，最终只要 n 够大， $f(n)$ 的值都会是 4094

三、代码思路分析

1. $f(n)$ 能否整除 8 的判断：
利用二进制码，只要后三位都是零即可，又 $f(n)>0$ ，可以用 $f(n)$ 与 0111 取与来判断；
2. $f(n)$ 尾数是否为 8 的判断：
这里我用了一个比较低效率的办法，每一次把 $f(n)$ 减 10，直到 $f(n)<0$ ，然后加 10 来得到 $f(n)$ 的个位数字；
3. $f(n)$ 模 4096 的运算：
与上一个类似的判断方法，每一次都把 $f(n)$ 减 4096，直到 $f(n)\leq 0$ ，然后加 4096 来完成一个取模运算
4. 通过设置一个计数器来判断运算次数
5. 符号 flag，1 代表+，-1 代表-；每次运算之后对这个 flag 进行维护

四、实验总结与收获

1. 通过使用汇编码来完成一个数据处理，学习使用 lc3 的一些基本操作指令集和，提高了对课程内容的掌握程度
2. 熟悉了 LC3 的编程环境，学会使用监视每一步 pc 指令的移动来进行 bug 发现和解决