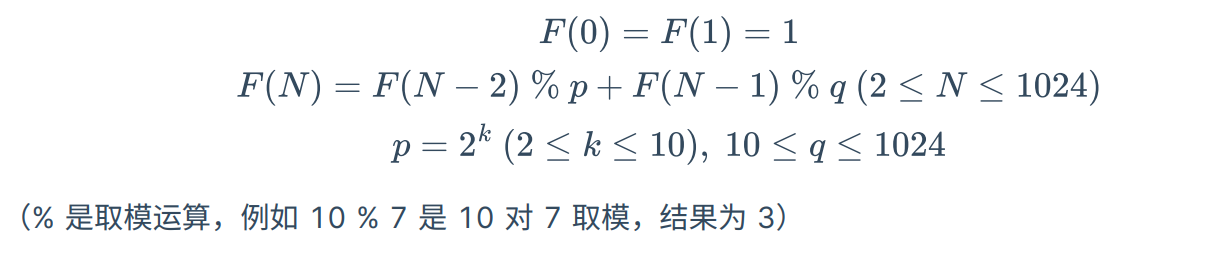
Lab2 实验报告

*PB21071416 张郑飞扬*

*2022.11.28*

一、实验目的（Purpose）

利用基于LC-3的汇编语言，设计一个计算变体斐波那契数列给定N的值F(N)的程序,该变体斐波那契数列的递推公式如下：

其中，p的存储地址为x3100，q为x3101，N为x3102。要求结果F(N)保存在x3103中。

另外，对汇编语言代码做如下要求：

1.代码需要以 .ORIG x3000开头

2.代码需要以 .END结尾

3.代码的最后一条指令应该为 TRAP X25 (HALT)

4.建议使用大写字母来表示关键字和Lable（如ADD，MO2）

5.逗号后面要加空格（如ADD R3, R2, #1）

二、实验原理（Principle）

1.思路：

由于该数列给出的是一个递推公式，可以考虑使用递归法来编写程序。然而本人能力有限，很难做到用汇编语言写递归。故本人采用递推的方式来编写程序，从F(2)开始递推，一直递推到F(N)，并输出到对应位置即可。

2.需要做的准备：

需要设计求模的子函数，本人设计了两个子函数，MODP和MODQ，用来求F(N-2)%p 和 F(N-1)%q。求模子函数的原理非常简单，即将原数不断循环减去模数，直到减为负数后，再加上一个模数，得到的就是所求值了。（如 9%2 = 9-2-2-2-2-2+2= 1）

需要一个计数器，作为判断何时停止递推的标准，显然，直接利用“N-1”作为计数器就可以，每运算一次递推式，N就减1。当N达到0时，就结束递推，输出值。

规定各寄存器在该程序中的作用：

R0：用作一个运算时的值保护寄存器，即每次递推运算之   
 前，把F(N-1)存在R0里面，到下一次运算时，R0里   
 的值要当做F(N-2)参与下一次运算。

R1：存储-p的值（不存q而存-q是为了方便ADD运算时  
 直接减去q）

R2：存储-q的值（不存q的道理同上）

R3：存储N-1（用作计数器，当N=0，结束递推运算）

R4：在每次运算开始前存储F(N-2)的值（该值从R0处读  
 取），每次取模运算结束后存储F(N-2)%p的值。

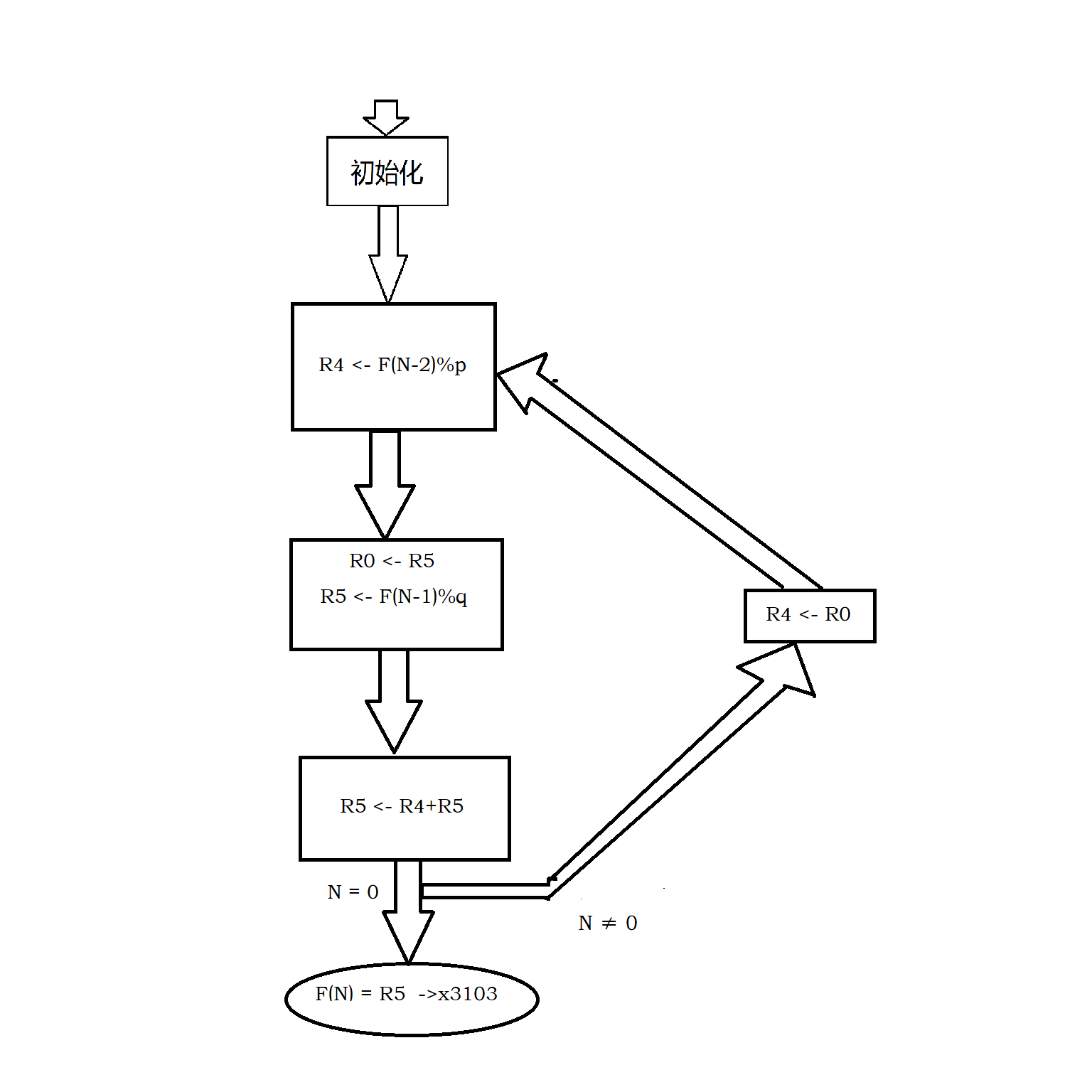
R5：在每次运算开始前存储F(N-1)的值（并且把该值赋给   
 R0），每次取模运算结束后存储的F(N-1)%q的值。  
 并且，每次加法运算结束后存储F(N)的值。

3.流程图

该流程图中，R4 <- F(N-2)%p 即为调用子函数MODP的  
 过程，R5 <- F(N-1)%q 即为调用子函数MODQ的过程。

初始化内容为：Clear All Registers； R1<- -q;

R2 <- -p; R3 <- N-1; R4<- F(0); R5<- F(1)



三、实验过程（Procedure）

在该实验中，本人主要遇到了以下Bug：

1.LD（ST）与LDI（STI）弄混，导致读取和存储环节弄错。  
 这个需要仔细辨析一下，比如有：

RESULT .FILL x3103 （假设这行代码位于x3012）

想要把R5存到地址x3103中，应该使用STI R5, RESULT

而不能用ST R5, RESULT 因为FILL的功能是占用下一个  
 地址并填充初始值，在此例中填充的即是x3103，那么我们  
 需要存储到的位置是x3103，而不是x3012，所以应该用  
 STI，而不是ST。

2.出现丢值，死循环等问题

在一开始，本人由于对寄存器分配不当，导致出现了丢值、死  
 循环等问题，在经过一步步运行程序，分析错误原因后，改正   
 并完善了这些问题，对必要值进行了保护。(比如R0用于保护  
 F(n-1))

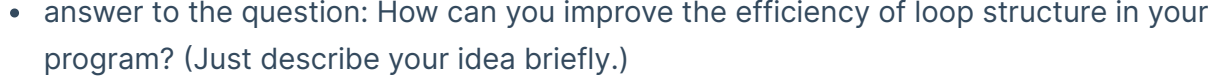
四、测试结果

测试数据及结果：(p q N均为手动录入)

|  |  |  |  |
| --- | --- | --- | --- |
| p (x3100) | q (x3103) | N (x3102) | F(N) (x3103) |
| 2 | 10 | 2 | 2 |
| 2 | 10 | 3 | 3 |
| 2 | 10 | 4 | 3 |
| 2 | 10 | 5 | 4 |
| 256 | 123 | 100 | 146 |
| 512 | 456 | 200 | 818 |
| 1024 | 789 | 300 | 129 |

以上结果均与该数列的对应值吻合。

五、问题回答



为了优化循环效率，有必要减少循环中对内存的操作次数，转而尽量多地使用寄存器来操作数据。在编写本程序是，本人在初始化中将所有待操作数据录入寄存器，整个循环运行过程中仅对寄存器中的数据进行操作，而不访问内存，待循环结束后再把结果录入内存指定位置。这样可以一定程度提高循环效率。