ICS 2021　LAB02

实验报告

PB20111699　吴骏东

2021.12.07

一．实验内容

在实验 2 中，我们要将一个数列的第 n 项存储到寄存器 R7 中。

在数学中，斐波那契数通常记为 F(n)。由斐波那契数形成的序列称为斐波那契序列。序列中从 0 和 1 开始，之后的每个数都是前两个数的和。即

$$F(0)=0,\ F(1)=1,\ F(n)=F(n-1)+F(n-2)$$

——维基百科

由于斐波那契序列增长十分迅速，我们的数列与斐波那契序列略有不同。该数列的通项公式如下：

$$F(0)=1,\quad F(1)=1,\quad F(2)=2$$

$$F(n)=((n-1)+2*F(n-3)) \bmod 1024 \quad (1<=n<=16384)$$

初始状态：n 存储在寄存器 R0 中。所有其他寄存器的初始值都为 0。

你的任务是：

1. 在寄存器 R7 中存储 F(n)；

2. 将你的学生号码分成四个长度相等的部分，用 a、b、c 和 d 标记。例如，TA 的学生编号是 PB17000144，所以结果为 a=17，b=0，c=1，d=44。在代码结尾部分用 *.FILL* 命令存储 F(a)、F(b)、F(c)和 F(d)的值。

二．程序设计与分析

本实验的思路是十分自然的。利用三个寄存器存储 f(n-1)、f(n-2)、f(n-3)的值，将结果存入 R7 中。模 1024 的操作可以通过与 1023(0111 1111)按位与实

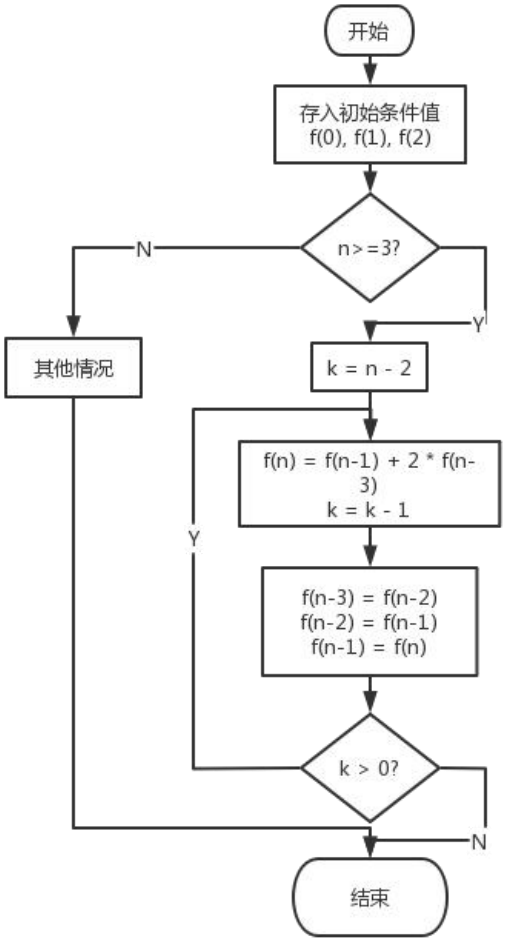

图 2.1

现。上述流程的流程图如图 2.1 所示。

注意到其中的其他情况部分。由于我们提前存入了 n=0~2 的值，所以在后续计算中对 n≦2 的情况应当单独进行处理。为此我们可以在 k=n-2 处进行判断：若此时 k 为 0，则 R7 存入 f(2) = 2；若此时 k 为负，则 R7 存入 f(1) = 1。这样我们就实现了对于 n > 0 的所有操作。

流程图中没有标出按位与的部分。这一步操作可以在 f(n) 计算后插入，即 R7 = R7 & 1023。1023 的产生方式有很多。由于本实验的要求比较宽松，故这里我选择直接循环计算得到 1024 再减 1。

最终的程序如下：其中 R0 存储目标 n 的值，R7 存储计算结果，R1~R3 用于存储前三项，R4~R5 用于计算 1023。**最终的代码行数为 27 行，符合要求。**

```
;programs start from now
[1] ADD     r7, r7, #1          ;set r7 to f(1)
[2] ADD     r5, r5, #1          ;2^(m)
[3] ADD     r4, r4, #10         ;m
[4] ADD     r0, r0, #-2
[5] BRp   #3
[6] BRn   #1
[7] ADD     r7, r7, #1          ;if n = 2 then r7 = 2
[8] BRnzp   #15

[9] ADD    r5, r5, r5
[10] ADD      r4, r4, #-1
[11] BRnp   #-3                 ;r5 is 1024
[12] ADD          r5, r5, #-1           ;r5 is 1023
```

```
[13] ADD        r1, r1, #1      ;f(n-3)
[14] ADD        r2, r2, #1      ;f(n-2)
[1]5 ADD        r3, r3, #2      ;f(n-1)

[16] ADD        r7, r1, r1      ;r7=2f(n-3)
[17] ADD        r7, r7, r3      ;r7=2f(n-3)+f(n-1)
[18] AND        r7, r7, r5

[19] ADD        r1, r2, #0
[20] ADD        r2, r3, #0
[21] ADD        r3, r7, #0

[22] ADD        r0, r0, #-1
[23] BRnp   #-8

[24] .FILL      #930
[25] .FILL      #246
[26] .FILL      #386
[27] .FILL      #854

;programs end from now
```

三. 小结

本实验中可以优化的点有很多。鉴于第三次实验就是代码优化，不妨就留给下一位同学吧。