

计算机系统概论 A 实验报告



实验题目：_____实验二_____

学生姓名：_____牛庆源_____

学生学号：_____PB21111733_____

完成日期：_____2022. 12. 5_____

【实验题目】

给定 p , q , N , 分别存储在 $x3100$, $x3101$ 和 $x3102$ 中, 汇编语言得到斐波那契数 $F(N)$, 存储在 $x3103$ 中。

【实验目的】

1. 掌握汇编语言各种指令的写法, 包括操作指令 ADD 和 AND, 地址搬移指令 LD 和 ST 以及控制指令。
2. 认识 PC 相对寻址模式, 可以使用控制指令写条件或者构成循环语句。

【实验原理】

1. $R0$ 存 p (位置为 $x3100$), $R1$ 存 q (位置为 $x3101$), $R2$ 存 N (位置为 $x3102$)。
2. $R3$ 存 $F(N-1)$, $R4$ 存 $F(N-2)$ 。
3. $R5$ 用于临时存储 $F(N-1)$, $F(N-2)$ 以及每一步新得到的 $F(N)$ 。
4. $R6$ 存储 $F(N-1) \% q$, $R7$ 存储 $F(N-2) \% p$ 。
5. 先进行预处理, 得到 p 和 q 的补码, 并按要求存储变量。 $R3$, $R4$ 分别初始化为 $F(1)$ 和 $F(0)$, 代表 $F(N-1)$ 和 $F(N-2)$, 此时 N 处理成为 $N-2$ 。
6. 操作得到 $F(N-1) \% q$ 和 $F(N-2) \% p$ 。方法是循环递减, 对 $F(N-1)$ 递减 q 直到小于 0, 最后再加 q 得到结果。对 $F(N-2)$ 递减 p 直到小于 0, 最后再加 p 得到结果。
7. 将 $F(N-1)$ 赋值给 $F(N-2)$, $F(N)$ 赋值给 $F(N-1)$, 循环进行第 6 步的操作直到 N 小于 0 结束。

8. 最后将 $F(N)$ 存在指定地址。

【实验步骤】

1. 初始地址。
2. 初始化三个寄存器到指定地址。
3. 计算得到 p 和 q 的补码。
4. 寄存器 $R3$ 和 $R4$ 存储 $F(N-1)$ 和 $F(N-2)$ ，并初始化为 1。
5. N 自减 2（已经得到了两个）。
6. 采用临时变量 $R5$ 保存 $R3$ 和 $R4$ 的数据，然后循环递减得到 $F(N-1) \% q$ 和 $F(N-2) \% p$ ，分别赋值给 $R6$ 和 $R7$ 。
7. $R6, R7$ 相加得到 $F(N)$ 存放在临时变量 $R5$ 。
8. $N-1$ ， F 对应的 N 也相应变化。
9. 判断 N 是否小于 0，否则重回第 6 步。
10. 若小于 0 则结束，将 $R5$ 的数据存入对应的地址中。

【实验结果】

1. 遇到的最多的 bug 是控制指令跳转的位置不对（没有熟练掌握），写好之后也深入理解了。其次是最后 N 判断一开始写的是 BRp ，判断条件提早了一个循环，也都予以修改。
2. 对于循环效率的提高，浓缩循环体中的语句条数。减少不必要的循环。（目前硬减...，想不到有别的方法减少了）

1. .asm 文件截图如下:

```
1      .ORIG x3000
2      LD    R0, #255      ;将p存入R0
3      LD    R1, #255      ;将q存入R1
4      LD    R2, #255      ;将N存入R2
5      NOT   R0, R0
6      ADD   R0, R0, #1     ;-p的补码
7      NOT   R1, R1
8      ADD   R1, R1, #1     ;-q的补码
9      ADD   R3, R3, #1     ;R3为F(N-1), 初始化为1
10     ADD   R4, R4, #1     ;R4为F(N-2), 初始化为1
11     ADD   R2, R2, #-2    ;N变为N-2(已经赋值了F(1)和F(2))
12 STAR0 ADD   R5, R3, #0
13 STAR1 ADD   R6, R5, #0   ;R6中存F(N-1) % q
14     ADD   R5, R5, R1
15     BRn   STAR2
16     ADD   R6, R5, #0
17     BRzp  STAR1
18 STAR2 ADD   R5, R4, #0   ;R7中存F(N-2) % p
19     ADD   R7, R5, #0
20 STAR3 ADD   R5, R5, R0
21     BRn   STAR4
22     ADD   R7, R5, #0
23     BRzp  STAR3
24 STAR4 ADD   R5, R6, R7   ;R5记录计算得到的F(N)
25     ADD   R4, R3, #0     ;F(N-1)赋给F(N-2)
26     ADD   R3, R5, #0     ;F(N)赋给F(N-1)
27     ADD   R2, R2, #-1    ;循环次数递减
28     BRzp  STAR0
29     ST    R5, #231
30     HALT
31     .END
```

2. 运行测试样例的结果如下：

lc3 评测

单样例最大指令数

1000000

选择评测实验

☐ lab1 ☒ lab2 ☐ lab3 ☐ lab4 ☐ 自定义

测试样例，样例之间以逗号分割

256:123:100, 512:456:200, 1024:789:300

代码文本

```
ADD R7, R5, #0
BRzp STAR3
STAR4 ADD R5, R6, R7 ;R5记录计算得到的F(N)
ADD R4, R3, #0 ;F(N-1)赋给F(N-2)
ADD R3, R5, #0 ;F(N)赋给F(N-1)
ADD R2, R2, #-1 ;循环次数递减
BRzp STAR0
ST R5, #231
HALT
.END
```

调试模式



评测

汇编评测

3 / 3 个通过测试用例

- 平均指令数: 3494.3333333333335
- 通过 256:123:100, 指令数: 1841, 输出: 146
- 通过 512:456:200, 指令数: 3384, 输出: 818
- 通过 1024:789:300, 指令数: 5258, 输出: 1219