

乒乓序列

郑悟强 PB22051082

2023.11.18

1 实验目的

本实验需要计算一个 PingPong 数列，PingPong 数列的计算规则如下：

给定函数列 $f(n) = \{(v_n, d_n) \mid v_n \in \mathbb{Z}, d_n \in \{+, -\}, n \geq 1\}$

$f(1) = (3, +)$

$v_{n+1} = 2v_n d_n$

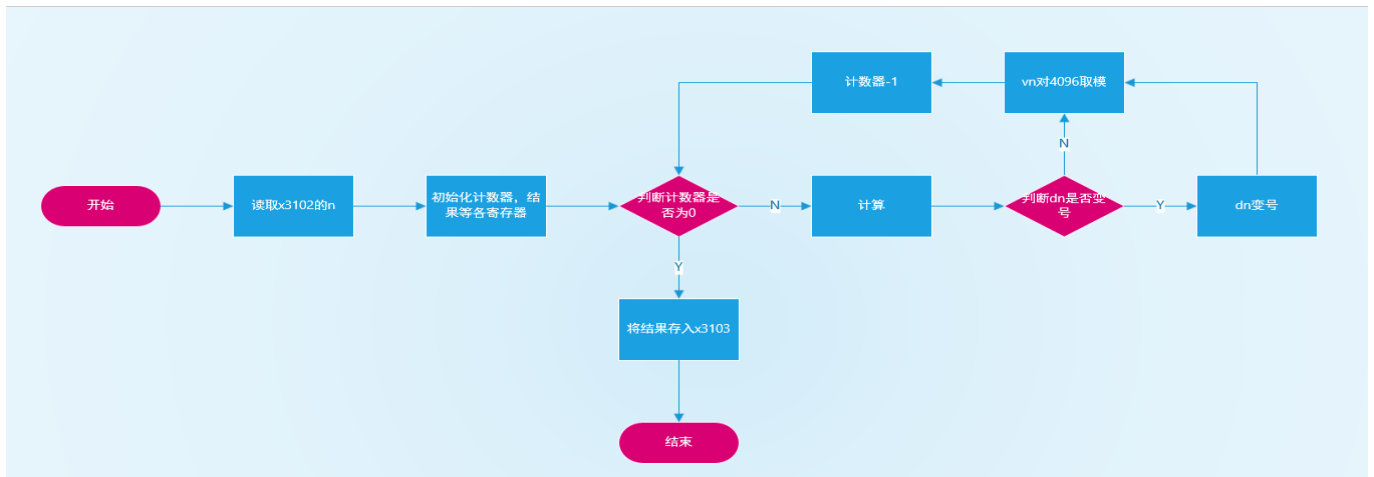
在计算 v_{n+1} 后，判断若 v_{n+1} 为 8 的倍数或 v_{n+1} 十进制下末位是 8，那么 d_{n+1} 变号，否则 $d_{n+1} = d_n$

实验目的：

读取位于 x3102 地址的 n，并计算 $f(n)$ ，结果存储在 x3103。

2 程序设计

2.1 总体思路

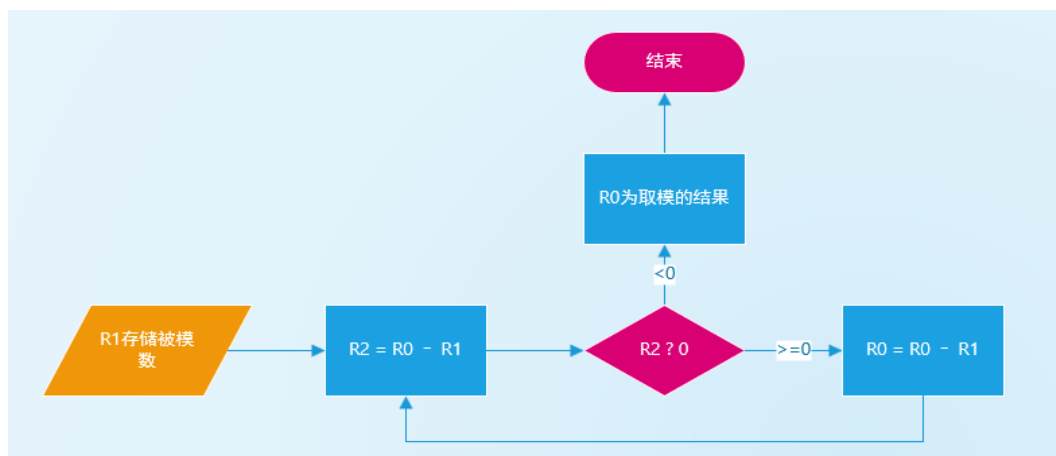


2.2 核心操作：取模运算

几个主要步骤均需要取模运算：

1. 判断是否变号时：判断是否 $v_{n+1} \bmod 10 = 1$ 。2. 计算后还需将 $v_{n+1} \bmod 4096$ 。

解决方法：记录被取模的数，不断进行减对应的数，判断得到的值的正负，若为正就继续减，若为 0 说明模为 0，若小于 0 说明被减的数即为模。流程图如下：



原理：通过反复将 R3 减 10，若大于 0 就继续重复减 10，直到小于等于 0，若刚好等于 0，说明是 10 的倍数，若小于 0，说明不是 10 的倍数。

2.3 完整代码

```

1 ;Initialization
2     .ORIG    x3000
3     LDI     R0, NLOCA ;R0读取x3102存储N的值
4     AND     R1, R1, #0 ;R1置0，用于存储v_n
5     ADD     R1, R1, #3 ;v_n初始值为3
6     AND     R2, R2, #0 ;R2置0用于表示d_n
7     ADD     R2, R2, #2 ;d_n初始为正
8 ;
9 ;Circulation
10 ;
11 CALCU    ADD     R0, R0, #-1 ;计数器减1
12          BRnz    OUTPUT ;计数器为0跳到结束
13          ADD     R1, R1, R1 ;计算v_(n+1)
14          ADD     R1, R1, R2
15 ;
16 ;Determine whether to change the signal of d_n
17 ;
18 DETERCH  ADD     R3, R1, #0 ;R3临时存储R1(v_n+1)用于判断是否用将d_n变号
19 EIGHT    ADD     R3, R3, #-8 ;反复减8用于得到是否为8的倍数
20          BRp     EIGHT ;为正的话继续减8
21          BRz     CHANGE ;为0说明刚好为8的倍数
22          ADD     R3, R1, #-8 ;继续判断是否为10的倍数
23 TEN      ADD     R3, R3, #-10 ;反复减10
24          BRp     TEN ;为正继续减10
  
```

```
25      BRz      CHANGE      ;为0说明刚好为10的倍数
26      BRnzp    SMOD        ;都不满足，说明不给d_n改变，继续下一步给fn取模
27 CHANGE ADD     R2, R2, #0
28      BRp      POSI        ;d_n若为正，就改成负的
29      BRn      NEG         ;d_n为负，就改成正的
30 POSI  ADD     R2, R2, #-4
31      BRnzp    SMOD
32 NEG   ADD     R2, R2, #4
33      BRnzp    SMOD
34      ;
35      ;FN = FN mod 4096
36      ;
37 SMOD  LD      R4, MLOCA    ;R4为4096
38      NOT     R5, R4
39      ADD     R5, R5, #1    ;R5为-4096
40 MOD   ADD     R6, R1, R5   ;R6 = v_(n+1) - 4096
41      BRnz    CALCU        ;继续下一次循环
42      ADD     R1, R1, R5   ;v_(n+1) = v_(n+1) - 4096
43      BRnzp    MOD
44      ;Output the answer
45      ;
46 OUTPUT STI     R1, FLOCA
47      TRAP    x25
48
49 MLOCA .FILL    #4096
50 NLOCA .FILL    x3102
51 FLOCA .FILL    x3103
52      .END
```

3 过程中遇到的错误

汇编语言 label 存储的地址的读取存储问题：

关于在最后用 NLOCA 和 FLOCA 位置，用.FILL 存储地址值如何调用出现问题，起初使用 LD 和 ST 进行调用，但调用得到的结果并不正确，求助助教得知应该使用 LDI 和 STI。

原因：.FILL 的作用是在这一地址分配 16bits 地址存储一个数值，这个地址的内容是 x3102 地址，读取时应该调用 x3102 这个地址里的内容，即 LDI 操作。而直接使用 LD，会读取当前 PC+x3102 的值的地址的内容，明显不是所需要的。

4 调试结果

使用助教提供的自测网站。

汇编评测

4 / 4 个通过测试用例

- 平均指令数: 29.75
- 通过 1, 指令数: 8, 输出: 3
- 通过 2, 指令数: 20, 输出: 8
- 通过 3, 指令数: 35, 输出: 14
- 通过 4, 指令数: 56, 输出: 26

5 效率的改进

本代码的时间效率为：考虑到计算 $f(n)$ 需循环 n 次，同时每一次循环中需要判断是否为 $10k+8$ 或 $8k$ ，就要反复循环-10 或-8，时间效率为 $O(f(n))=O(2^n)$ 。所以最终的时间效率为 $O(2^n n)$ 。

改进效率的想法：考虑到满足变号条件的数，最后四位只有特定情况，可以通过判断最后四位是否是特定数值的方法来实现判断。

想法的缺陷：

1. 这只是必要不充分条件，也有满足后四位为这几种但不满足条件的。
2. 由于判断后四位需先累加 12 次，当 n 较小 (小于 10) 时，效率并不高，并且当 n 很大时， $f(n)$ 为定值，所以效率提升不大。

6 实验体会与收获

通过本实验我熟悉了 LC-3 汇编的写法，并且了解了汇编语言的技巧与方便之处。同时也学会了使用 LC-3 汇编实现判断整除的方法。