

实验二

- 实验题目：lending your name
- 姓名：倪睿阳
- 学号：PB20111624

设计思路

关于斐波那契数列的题目在C语言中已经见过许多了，由于递归的复杂度太高，一般采用递推的方式**一步一步迭代**下去。

观察题目中的式子，其中乘以2可以使用add自身来实现，mod 1024可以考虑与 **x03ff** 做一个与的操作。

- 初始化部分(已经略去了初始化为0)：

```
LD R6, COMP
LEA R1, LABEL1
...
COMP .FILL x03ff
```

其中LABEL1在下方：

```
LABEL1 .FILL #2
LABEL2 .FILL #1
LABEL3 .FILL #1
```

- 检测是否R0小于2，是就跳转

```
ADD R0, R0, #-2
BRnz JUMP
...
JUMP ADD R3, R0, #2
      ADD R1, R1, R2 //move the point
```

- 对“指针”进行微调，并将一个被加数放入寄存器中

```
LOOP ADD R2, R2, #1 //a loop starts
      LEA R1, LABEL1 //get the address of array[0]
      ADD R2, R2, #-3 //keep R2 less than 3
      BRz IGNORE
      ADD R2, R2, #3
      ADD R1, R1, R2 //move the point
```

```
LDR R5, R1, #-1 //R3 mod 3 is 1 or 2
BRnzp STEP
IGNORE LDR R5, R1, #2 //R3 mod 3 is 0
```

- 对另一个需要被加的数进行处理并放入开辟的内存中

```
STEP LDR R4, R1, #0 //R4 is the number to be changed
      ADD R4, R4, R4
      ADD R4, R4, R5
      AND R4, R4, R6 //mod 1024
      STR R4, R1, #0 //save the number
      BRnzp NEXT
```

- 循环末尾对R0进行判断(是否小于等于0了)

```
NEXT LDR R7, R1, #0
      ADD R0, R0, #-1 //stop when R0 is 0
      BRp LOOP
```

改进方法

起初的设计思路是将F(0), F(1), F(2)直接赋值给R1, R2, R3然后在其中操作, 最后赋值给R7, 但是其中对于模3的操作占了太多指令行数, 最终考虑直接在三块内存中进行操作。

P.S. LC-3tools中使用 ; 作为注释的, 但Markdown中没有这种语言, 使用了 // 当做注释, 原代码没这么搞。

原代码 :

```
.ORIG x3000
LD R0 NUM
AND R2, R2, #0
AND R4, R4, #0
AND R5, R5, #0

LD R6, COMP
LEA R1, LABEL1

ADD R0, R0, #-2 ;test whether R0 is less than 3 or not
BRnz JUMP ;less than 3

LOOP ADD R2, R2, #1 ;a loop starts
      LEA R1, LABEL1 ;get the address of array[0]
      ADD R2, R2, #-3 ;keep R2 less than 3
      BRz IGNORE
      ADD R2, R2, #3
      ADD R1, R1, R2 ;move the point
      LDR R5, R1, #-1 ;R3 mod 3 is 1 or 2
```

```
BRnzp STEP
IGNORE LDR R5, R1, #2 ;R3 mod 3 is 0

STEP LDR R4, R1, #0 ;R4 is the number to be changed
      ADD R4, R4, R4
      ADD R4, R4, R5
      AND R4, R4, R6 ;mod 1024
      STR R4, R1, #0 ;save the number
      BRnzp NEXT

JUMP ADD R3, R0, #2 ;R2 is borrowed here
      ADD R1, R1, R2 ;move the point

NEXT LDR R7, R1, #0
      ADD R0, R0, #-1 ;stop when R0 is 0
      BRp LOOP

HALT

LABEL1 .FILL #2
LABEL2 .FILL #1
LABEL3 .FILL #1
COMP .FILL x03FF
RESA .FILL #930
RESB .FILL #246
RESC .FILL #386
RESD .FILL #706

NUM .FILL #30
.END
```