实验报告

PB18061443 江昊霖

2022 年 11 月 28 日

实验目的

实现变形斐波那契数列

$$F(0) = F(1) = 1$$

$$F(N) = F(N-2)\%p + F(N-1)\%q, \quad 2 \leqslant N \leqslant 1024$$

$$p = 2^k, \quad 2 \leqslant k \leqslant 10$$

$$10 \leqslant q \leqslant 1024.$$

p 存储在`x3100`, q 存储在`x3101`, N 存储在`x3102`.
将 $F(N)$ 的结果放到`x3103`

实验原理

将此次任务写写成如下伪代码:

实验步骤

取余操作

要想获得 R1%R2, 将 R1 减去 R2 直到结果为负数，再将 R1 加上 R2 即可得到.

减法

因 LC-3 中只有加法，需将被减数和取反加 1 后的减数相加，方能得到结果。

Algorithm 1: variant of the Fibonacci sequence

Data: p, q, N

Result: f

$n1 \leftarrow 1$;

$n2 \leftarrow 1$;

$N \leftarrow N - 1$;

while $N > 0$ **do**

  $t1 = n1\%p$;

  $t2 = n2\%q$;

  $f = t1 + t2$;

  $n1 \leftarrow n2$;

  $n2 \leftarrow f$;

  $N \leftarrow N - 1$;

end

代码

读取 p, q, N

```
0 LD R0, x0ff ;R0 <- p
1 LD R1, x0ff ;R1 <- q
2 LD R2, x0ff ;R2 <- N
```

因为 p 是 2 的 k 次方, 所以对于一个二进制数 B, $B\%p = B$ `ADD` $(p-1)$

```
3 ADD R0, R0, #-1 ; p = p-1
```

```
4 AND R3, R3, #0 ; R3 <- 0
5 ADD R3, R3, #1 ; f(n-2) = 1
6 ADD R4, R3, #0 ; f(n-1) = f(n-2) = 1
```

变形斐波那契数列的 $N-1$ 次递归

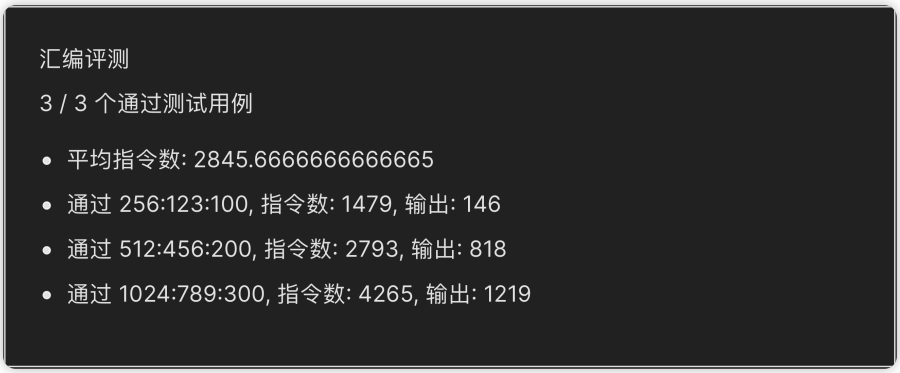
```
7 ADD R2, R2, #-1 ; N -= 1
8 BRnz #11 ; x3014
9 AND R5, R3, R0 ; temp1 = f(n-2)%p
10 ADD R6, R4, #0 ; temp2 = f(n-1)
11 NOT R7, R1 ; -p
12 ADD R7, R7, #1
13 ADD R6, R6, R7 ; temp2 - q
14 BRzp #-2 ; x300D
```

```
ADD R6, R6, R1 ; temp2 = temp2 + q
ADD R7, R5, R6 ; R7 = f(n) = t1 + t2
ADD R3, R4, #0 ; f(n-2) = f(n-1)
ADD R4, R7, #0 ; f(n-1) = f(n)
BRnzp #-13 ; x3007
```

储存结果到x3013

```
ST R7, xEE ; x3103
```

实验结果



实验思考

每次循环都要计算 $-q$, 若有多余的寄存器, 就可只计算一次, 可减少 $2(N-2)$ 次执行。

减少了不必要的跳转指令

将原本必定跳转的指令BRnzp, 并对代码结构稍作修改

```
.ORIG x3000

LD R0, x0ff ;R0 <- p
LD R1, x0ff ;R1 <- q
LD R2, x0ff ;R2 <- N
ADD R0, R0, #-1 ; p = p-1
ADD R2, R2, #-1 ; N = N - 1
```

```
8
9  AND R3, R3, #0 ; R3 <- 0
10 ADD R3, R3, #1 ; f(n-2) = 1
11 ADD R4, R3, #0 ; f(n-1) = f(n-2) = 1
12
13 AND R5, R3, R0 ; temp1 = f(n-2)%p
14
15 ; temp2 = f(n-1)%q
16 ADD R6, R4, #0 ; temp2 = f(n-1)
17
18 ; -q
19 NOT R7, R1
20 ADD R7, R7, #1
21
22 ADD R6, R6, R7 ; temp2 - q
23 BRzp #-2 ; x300C
24 ADD R6, R6, R1 ; temp2 = temp2 + q
25
26 ADD R7, R5, R6 ; R7 = f(n) = t1 + t2
27
28 ADD R3, R4, #0 ; f(n-2) = f(n-1) = 1
29 ADD R4, R7, #0 ; f(n-1) = f(n)
30
31 ADD R2, R2, #-1
32 BRp #-12 ; x3008
33 ST R7, xEE ; x3103
34 TRAP x25
35
36 .END
```

运行指令数减少了 N 次。

汇编评测

3 / 3 个通过测试用例

- 平均指令数: 2645.6666666666665
- 通过 256:123:100, 指令数: 1379, 输出: 146
- 通过 512:456:200, 指令数: 2593, 输出: 818
- 通过 1024:789:300, 指令数: 3965, 输出: 1219