

ICS Lab Report - lab2

SCGY Cao Gaoxiang PB20000061

December 9, 2021

Lab Name

Lending Your Name

Lab Purpose

Store the n -th number of a sequence into register R7.

Lab Content

In mathematics, the Fibonacci numbers, commonly denoted F_n , form a sequence, called the Fibonacci sequence, such that each number is the sum of the two preceding ones, starting from 0 and 1. That is $F(0) = 0$, $F(1) = 1$, $F(n) = F(n-1) + F(n-2)$.

As the Fibonacci sequence grows quite rapidly, we made a little difference with the origin Fibonacci sequence:

$$F(0) = 1$$

$$F(1) = 1$$

$$F(2) = 2$$

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

In the beginning, n is stored in R0. All other registers' initial values are 0.

Your tasks are:

- Store $F(n)$ in register R7.
- Divide your student number into four equal segments, labelling them with a , b , c and d . For example, TA's student ID is PB17000144, so $a = 17$, $b = 0$, $c = 14$, $d = 44$. Store value of $F(a)$, $F(b)$, $F(c)$ and $F(d)$ at the end of your code with ".FILL" pseudo command.

Lab Environment

Windows 11 Home Edition version 21H2, Visual Studio Code, LC3Tools v2.0.2.

Lab Procedure

This time, it is only acquired to provide l-version of the program. To help understand the content of this lab, also help to verify the correctness of the program, a C program is firstly written as below.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main(int argc, char const *argv[]) {
4     int n = 0;
5     if (argc == 1) scanf("%d", &n);
6     else sscanf(argv[1], "%d", &n);
7     int a0, a1 = 2, a2 = 1, a3 = 1;
8     if (n >= 3) for (int i = 3; i <= n; ++i) {
9         a0 = (a1 + 2 * a3) % 1024;
10        a3 = a2; a2 = a1; a1 = a0;
11    }
12    else if (n == 0) a0 = 1;
13    else if (n == 1) a0 = 1;
14    else if (n == 2) a0 = 2;
15    printf("%d\n", a0);
16    return 0;
17 }
```

However, the C program is only easy to read but not easy to translate to LC-3 Assembly Language. So after variable renaming and rearrange of statements, we get the code below which is easy to translate.

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 int main(int argc, char const *argv[]) {
4     int16_t R0 = 0;
5     if (argc == 1) scanf("%hd", &R0);
6     else sscanf(argv[1], "%hd", &R0);
7     int16_t R7, R2 = 2, R3 = 1, R4 = 1;
8     if (R0 == 0) R7 = 1;
9     else if (R0 == 1) R7 = 1;
10    else if (R0 == 2) R7 = 2;
11    else for (int16_t i = 3; i <= R0; ++i) {
12        R7 = (R2 + 2 * R4) % 1024;
13        R4 = R3; R3 = R2; R2 = R7;
14    }
15    printf("%hd\n", R7);
16    return 0;
17 }

```

Standing here, it seems that all we need to do is translating. But notice that decimal number 1023 cannot be stored using a 5-bit 2's complement number, we need to store it at another position of memory.

Another thing is to "lend my name". My student number is "PB20000061", which will be divided into $a = 20$, $b = 0$, $c = 0$, $d = 61$. After computing we know that $F(a) = 930$, $F(b) = 1$, $F(c) = 930$, $F(d) = 930$.

Finally we get the answer code below, whose length is 28, satisfying the standard of full marks.

```

1 .ORIG x3000
2 ADD R0, R0, #-1
3 BRz EXIT1
4 ADD R0, R0, #-1
5 BRz EXIT2
6 ADD R2, R2, #2
7 ADD R3, R3, #1
8 ADD R4, R4, #1
9 LD R5, MOD
10 LOOP ADD R0, R0, #-1
11 BRn OUTLOOP
12 ADD R6, R4, R4
13 ADD R7, R2, R6
14 AND R7, R7, R5
15 ADD R4, R3, #0
16 ADD R3, R2, #0
17 ADD R2, R7, #0
18 BRnzp LOOP
19 OUTLOOP BRnzp EXIT
20 EXIT1 ADD R7, R7, #1
21 BRnzp EXIT
22 EXIT2 ADD R7, R7, #2
23 BRnzp EXIT
24 MOD .FILL #1023
25 Fa .FILL #930
26 Fb .FILL #1
27 Fc .FILL #1
28 Fd .FILL #262
29 EXIT
30 .END

```

Correctness Verification

We use program written in LC-3 Assembly to calculate the answer of $F(n)$ and than compare the result with the one get from the C program above.

We have tested the integer n more than the following: 1, 2, 3, 20, 61, 16384. One of the cases in listed below as Figure 1.

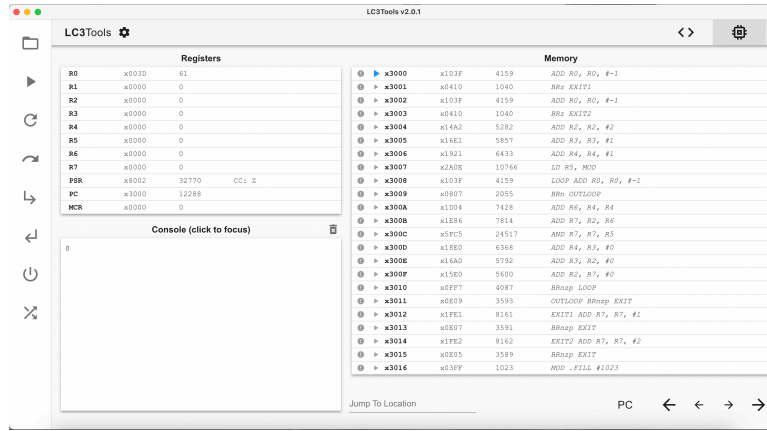
Note: Due to some reasons, tests were run on macOS Monterey with LC3Tools v2.0.1.

```

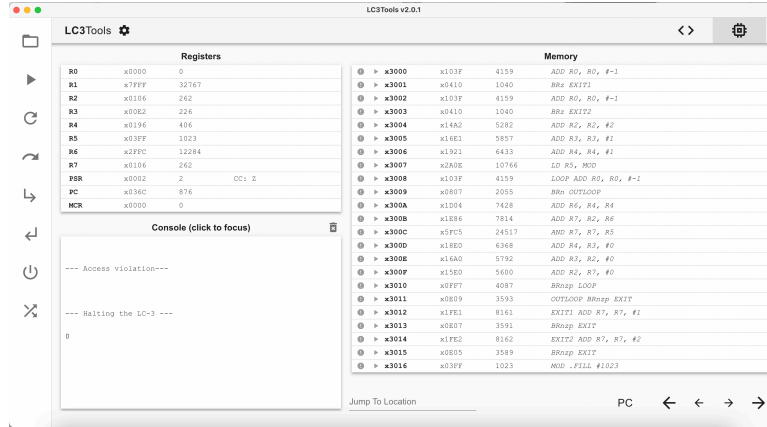
问题 输出 调试控制台 终端
+ LaTeX_Documents git:(main) $ /Users/bill/Documents/LaTeX_Documents/Reports/ICS_Reports/lab2'
+ lab2 git:(main) $ clang ans_origin.c -o ans_origin.out -std=c11
+ lab2 git:(main) $ ./ans_origin.out 61
262
+ lab2 git:(main) $ █

```

(a) The C Program



(b) Before Running the LC3 Program



(c) After Running the LC3 Program

Figure 1: One Example of Correctness Verification