第二次实验实验报告

实现思路

递推公式为：F(n) = (F(n-1) + 2 * F(n-3)) mod 1024 (1 <= n <= 16384)

所以需要做三件事：

- 存储 F(n-1)，F(n-2)，F(n-3)
- 实现取模运算
- 特殊情况处理：如果 n 小于 3，那么不会进行递推计算，直接输出相应的值

对应的实现思路为：

- 用 R7 存储 F(n)，R1 存储 F(n-3)，R1 存储 F(n-2)，R1 存储 F(n-1)。每次进行完一次递推求值，则更新这四个寄存器的值
- 取模：判断 R7-1024，若结果非负，那么 R7 = R7-1024，重复执行这个操作直到 R7 的值小于 1024
- 特判 n 小于 3 的情况，使用跳转指令，直接更新 R7，然后停止程序

代码

代码思路：

> 置 R1 = 1, R2 = 1, R3 = 2
>
> 判断 n < 3，若是，则直接更改 R7 的值，并停止程序
>
> 若 n ≥3，进入循环：
>
> R7 = R1 + R1 + R3
>
> 若 R7 > 1024，R7 = R7 - 1024，重复执行这一步骤直到 R7 < 1024
>
> 置 R1 = R2, R2 = R3, R3 = R7，注意先后顺序
>
> R0 = R0 - 1
>
> 若 R0 > 0，则重复执行该循环

之后计算我的学号的四部分 20，06，13，43 分别对应的斐波那契值，存在代码的末尾

核心代码为：

```
.ORIG x3000
    ADD R1,R1,#1
    ADD R2,R2,#1
    ADD     R3,R3,#2
    ADD R0,R0,#0
    BRz NZO
    ADD     R0,R0,#-1
    BRz NZO
    ADD R0,R0,#-1
    BRz NTWO
LOOP    ADD     R7,R1,R1
    ADD R7,R7,R3
    LD  R5,N1024
```

```
    ADD R4,R7,R5
    BRn NOR7
MOD ADD     R7,R7,R5
    ADD R4,R7,R5
    BRp MOD
NOR7    ADD R1,R2,#0
    ADD R2,R3,#0
    ADD R3,R7,#0
    ADD R0,R0,#-1
    BRp LOOP
    TRAP    x25
NZO ADD R7,R1,#0
    TRAP    x25
NTWO    ADD R7,R3,#0
    TRAP    x25
N1024   .FILL   #-1024
Fa  .FILL   #930
Fb  .FILL   #18
Fc  .FILL   #710
Fd  .FILL   #22
.END
```

代码分析

- 核心代码共 32 行
- 代码正确性检验：用以下 C++ 代码检验

```cpp
#include <iostream>
using namespace std;
int f[17000];
int n;
int main()
{
    f[0] = 1;
    f[1] = 1;
    f[2] = 2;
    for (int i = 3; i <= 16384; i++)   f[i] = (f[i - 1] + 2 * f[i - 3]) % 1024;
    while (1) {
        cout << "Input n: ";
        cin >> n;
        cout <<"F[n] = " << f[n] << endl;
    }
    return 0;
}
```

代码优化

- 在 n < 3 的特殊情况处理时，因为 F(0) = 1，F(1) = 1，所以这两种情况可以合并处理，可以减少两行代码：

  在之前分开处理 n = 0 与 n = 1 情况的代码为：

```
    ADD R0,R0,#0
    BRz NZERO
    ADD     R0,R0,#-1
    BRz NONE

NZERO   ADD R7,R1,#0
    TRAP    x25
NONE    ADD R7,R2,#0
    TRAP    x25
```

合并之后的代码为：

```
    ADD R0,R0,#0
    BRz NZO
    ADD     R0,R0,#-1
    BRz NZO

    NZO ADD R7,R1,#0
    TRAP    x25
```

可以节省两行代码

总结

- 通过与第一次实验做对比，我可以感受到汇编语言比机器码更适合编程，无论是在代码书写、代码阅读以及 Debug 方面
- 仍需特别注意一些操作的实现，在最初实现取模运算时，我的思路是如果 R7 大于 1024，则将 R7 - 1024，但是忽略了 R7 - 1024 仍可能大于 1024 这种情况，导致出现问题