lab2:Fibonacci sequence

郑子涵 PB20000248

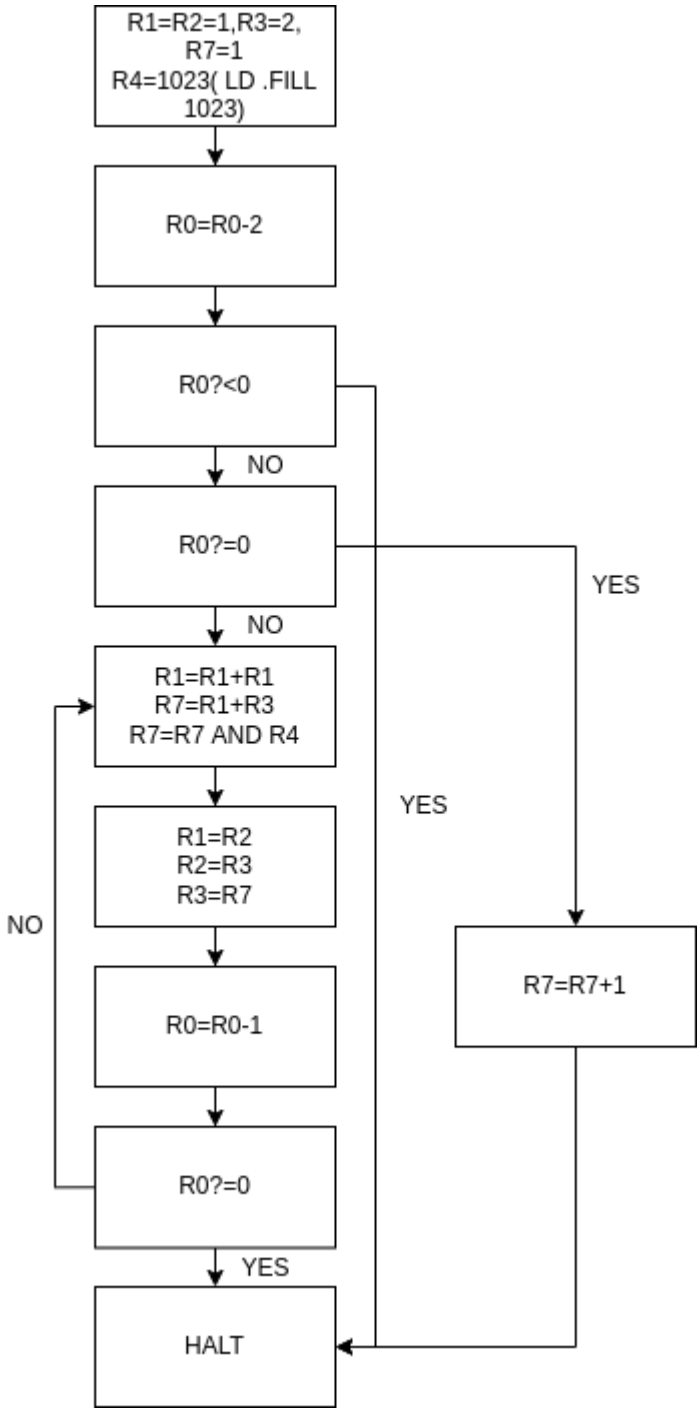
实验要求

- 把一个数列的第n位存储到**R7**寄存器.
- 数列满足如下条件： F(0) = 1 F(1) = 1 F(2) = 2 ..... F(n) = (F(n-1) + 2 * F(n-3)) mod 1024 (1 <= n <= 16384)
- 最开始**n储存在R0，其余寄存器均为0**.
- 把学号每两位分开为**a，b，c，d**四部分，如PB20000248，有a=20，b=0，c=2，d=48。用"**.FILL**"伪代码储存 F(a), F(b), F(c)，F(d) 的值在代码的结尾。

设计实现

设计思路

按照数列的定义来计算F(n)的值，把F(0)，F(1)，F(2) 分开来讨论，其余的按定义依次计算出每个值，取余操作通过"AND 1023"来实现，即保留小于1024的部分，一直到n为止。再通过运行程序，算出F(a), F(b), F(c)，F(d) 的值后，用“.FILL”伪代码填入程序中 流程图如下：



初始代码

核心代码**23**行

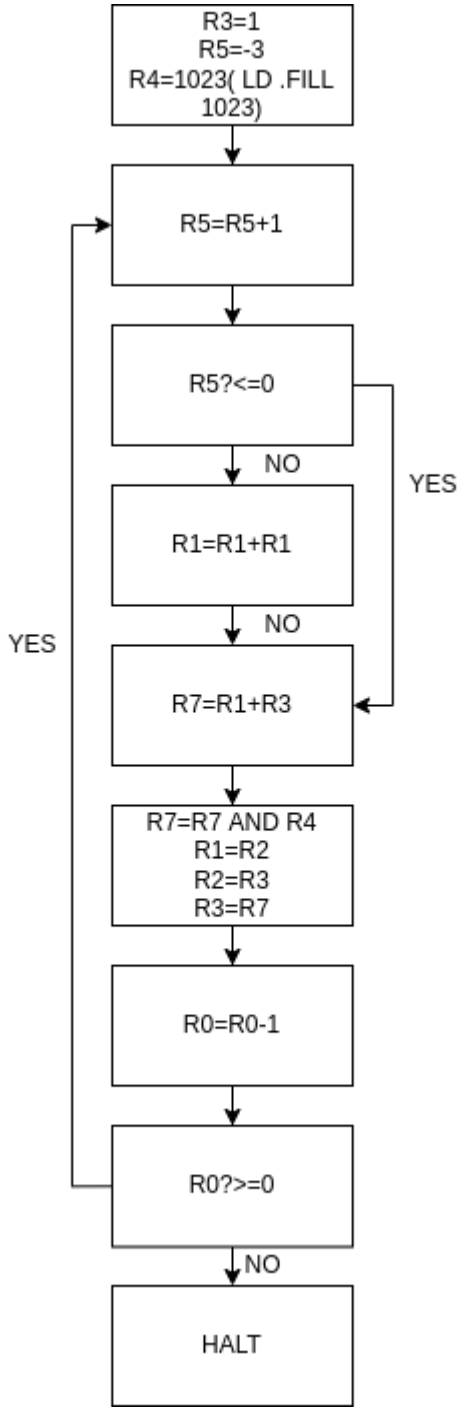
```
.ORIG   x3000
    AND R1, R1, #0
    AND R2, R2, #0
    AND R3, R3, #0
    AND R4, R4, #0
    AND R7, R7, #0
    ADD R1, R1, #1
    ADD R2, R2, #1
    ADD R3, R3, #2
    ADD R7, R7, #1
```

```
      LD  R4, NUM
      ADD R0, R0, #-2
      BRn END1
      BRz POSE1
      POSE3 ADD   R1, R1, R1
      ADD R7, R1, R3
      AND R7, R7, R4
      ADD R1, R2, #0
      ADD R2, R3, #0
      ADD R3, R7, #0
      ADD R0, R0, #-1
      BRp POSE3
      BRnzp   END1
      POSE1 ADD   R7, R7, #1
      END1 HALT
      NUM .FILL #1023
      a .FILL #930;F(a),a=20
      b .FILL #1;F(b),b=0
      c .FILL #2;f(c),c=2
      d .FILL #898;F(d),d=48
   .END
```

优化思路

刚开始想通过设置F(-3),F(-2),F(-1)来把F(0)，F(1)，F(2)的情况包括到 一般的情况中，但发现会出现浮点数（F(-1)=1/2,F(-2)=0,F(-3)=1/4），用LC3难以实现。但可以通过改变前三次的递推规则来使结果成立。 具体如

下： F(-3)=0 F(-2)=0 F(-1)=1 F(n) = (F(n-1) + F(n-3)) mod 1024 (0 <= n <= 2) 如此便可使这几种情况有所合并 流程图如下：



优化代码

核心代码**18**行

```
.ORIG   x3000
    AND R1, R1, #0
    AND R2, R2, #0
    AND R3, R3, #0
    AND R4, R4, #0
    AND R7, R7, #0
    ADD R3, R3, #1
    AND R5, R5, #0
    ADD R5, R5, #-3
```

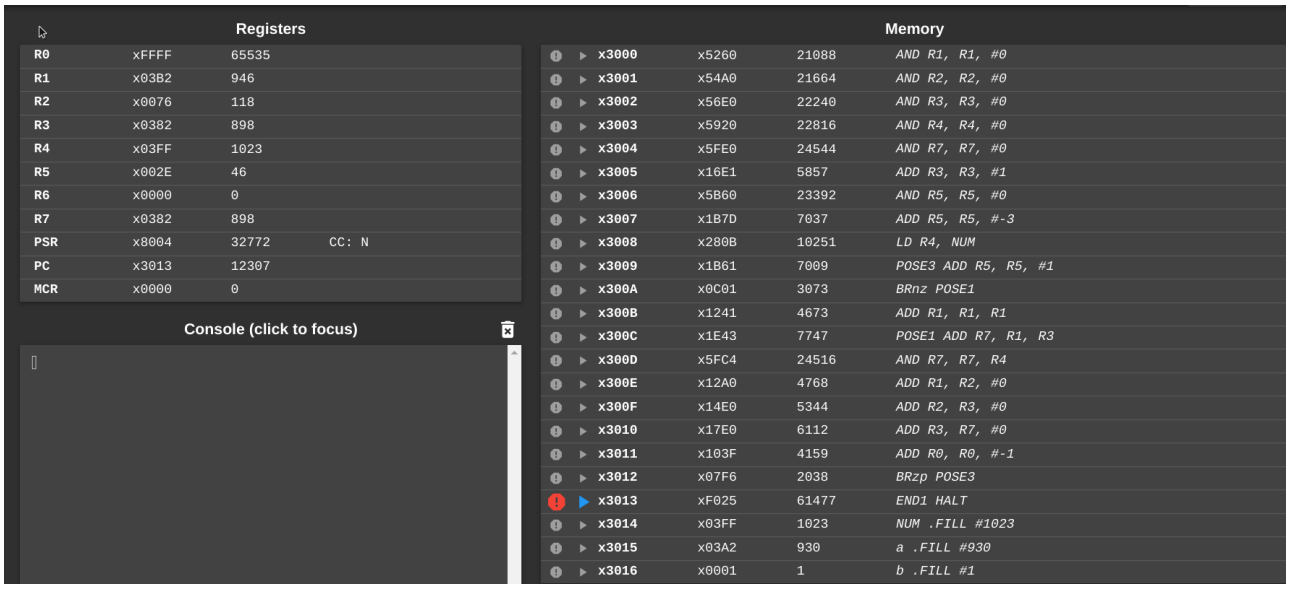
```
    LD  R4, NUM
    POSE3 ADD   R5, R5, #1
    BRnz POSE1
    ADD R1, R1, R1
    POSE1 ADD   R7, R1, R3
    AND R7, R7, R4
    ADD R1, R2, #0
    ADD R2, R3, #0
    ADD R3, R7, #0
    ADD R0, R0, #-1
    BRzp POSE3
    END1 HALT
    NUM .FILL #1023
    a .FILL #930;F(a),a=20
    b .FILL #1;F(b),b=0
    c .FILL #2;f(c),c=2
    d .FILL #898;F(d),d=48
.END
```

实验总结

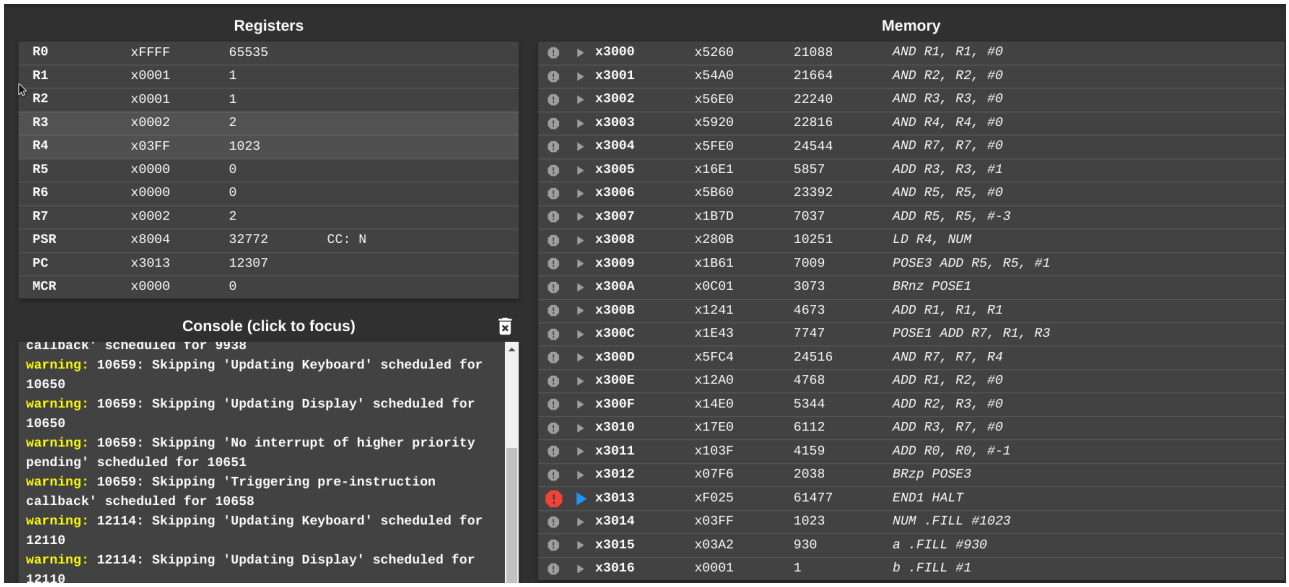
最开始是按分类讨论的思想来写，但为了优化代码行数，试图把这几种情况合并在一起，但是需要稍微改变规则才能完成，但总归还是用递推的思想来写的。

样例测试

- 把样例运行的结果和编写的C语言运行结果相比较即可，若相同则成功。
- F(48)=898



- F(2)=2



对于测试结果，所测试的都符合要求，过程也符合设计。