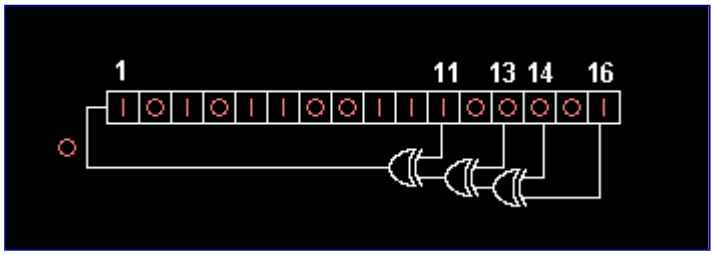
线性反馈移位寄存器（英语：Linear feedback shift register，LFSR）是指给定前一状态的输出，将该输出的线性函数再用作输入的移位寄存器。异或运算是最常见的单比特线性函数：对寄存器的某些位进行异或操作后作为输入，再对寄存器中的各比特进行整体移位。

赋给寄存器的初始值叫做“种子”，因为线性反馈移位寄存器的运算是确定性的，所以，由寄存器所生成的数据流完全决定于寄存器当时或者之前的状态。而且，由于寄存器的状态是有限的，它最终肯定会是一个重复的循环。然而，通过本原多项式，线性反馈移位寄存器可以生成看起来是随机的且循环周期非常长的序列。

线性反馈移位寄存器的应用包括生成伪随机数，伪随机噪声序列，快速数字计数器，还有扰频器。线性反馈移位寄存器在硬件和软件方面的应用都非常得普遍。

循环冗余校验中用于快速校验传输错误的数学原理，就与线性反馈移位寄存器密切相关。

Fibonacci LFSRs



一个 16-位 Fibonacci LFSR. 图中白色数字为抽头，与表中本原多项式相对应，则寄存器的循环周期为最大，65535（不包括全零状态）。图中的状态为 0xACE1 (十六进制) 下一个状态是 0x5670.

影响下一个状态的比特位叫做抽头。图中，抽头序列为[16,14,13,11]。LFSR 最右端的比特为输出比特。抽头依次与输出比特进行异或运算，然后反馈回最左端的位。最右端位置所生成的序列被称为输出流。

- 影响 LFSR 下一个状态的比特位叫做抽头（图中白色数字）
- 最大长度的 LFSR 生成一个 M 序列（例如，只有与有一定抽序列的 LFSR 才能通过所有 $2^n - 1$ 个内部状态，不包括全零状态），除非它本身为全零，亦即状态永不改变
- 作为基于异或运算的 LFSR 的替换，LFSR 也可以给予同或运算。与使用异或门的 LFSR 全零状态下为无效状态相应的，使用同或门的 LFSR 在全“1”状态下也是无效的。

有 LFSR 或者基于同或门的 LFSR 生成的序列可以被认为是通格雷码或者自然二进制码同样有效的二进制序列。

在 LFSR 中，抽头的设定可以用有限域算数中模 2 的多项式来表示。这就意味着，多项式中的所有系数必须是“1”或者“0”。这个多项式被称作回授多项式或特征多项式。例如图中的抽头为在第 16，14，13，11 个比特，则相应的特征多项式为：

$$x^{16} + x^{14} + x^{13} + x^{11} + 1.$$

多项式中常数“1”并不代表某一个抽头，它所指的是一个比特位的输入（例如 x^0,等效为 1 ）。多项式中的指数代表从左至右的抽头位。第一个和最后一个比特一般相应的是输入和输出位。

当且仅当相应的回授多项式是本原多项式时，LFSR 才能达到最大长度。这表示一下条件是必须的：

- 抽头的数量必须为偶数。
- 抽头之间不能成对出现，必须是互质的。

生成最长 LFSRs 的本原多项式表可通过下部的链接找到。 这类型 LFSR 也被成为**标准**，**多对一**或**外部异或门**的 LFSR。下一节将会介绍 Galois 型的 LFSR。

Fibonacci LFSR 也叫 many to one,好几个 flip-flop 的输出经过 xor，给第一位作为输入。它一般用来加密数据，因为它产生的序列在有限范围内是随机的。
以下是 C 的源文件，用来查看所有的输出：

```
#include <stdint.h>
#include <stdio.h>

int main()
{
  printf("This program is used to generate fibonacci with polynomial=x3+x2+1");
  uint16_t start=7;
  uint16_t lfsr=start;
  unsigned bit;
  unsigned period=0;

  do{
    bit=((lfsr>>2)^(lfsr>>1)) & 1;
    lfsr=lfsr<<1;
    lfsr=lfsr & 6;
    lfsr=lfsr & bit;
    period =period +1;
    printf("%d:  %d\n",period,lfsr);
  }while(lfsr!=start);

  return 0;
}
```