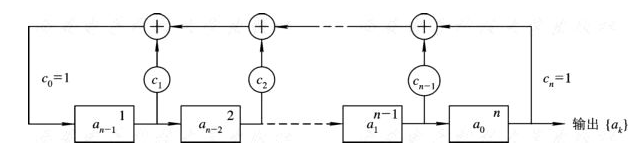
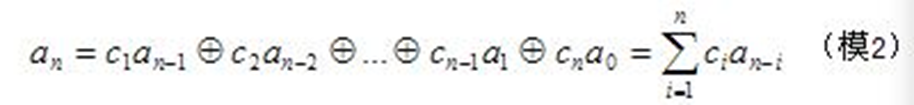
M序列原理介绍

# M序列的递推方程:





### 反馈系数：本原多项式

这里是：**C1-Cn**，称为反馈系数，经证明，只有反馈系数符合**本原多项式**，才可以产生M序列。本原多项式中必须含有零次幂项，即对应C0=1。

### 初始状态：

初始状态为a0-an-1，共n个数。一般情况下，**初始状态为[0 0 0 ... 1]**。

# 本原多项式（Primitive Polynomial）

## 函数primpoly(7,'all')

这个命令的目的是得到7阶M序列的**所有本原多项式**，如果想得到其他阶的m序列，自行修改数字7即可。

**得到的结果为**

**Primitive polynomial(s) 本原多项式**：

输入命令：

primpoly(7,'all')

结果为：

Primitive polynomial(s)=  
D^7+D^1+1  
D^7+D^3+1  
D^7+D^3+D^2+D^1+1  
D^7+D^4+1  
D^7+D^4+D^3+D^2+1  
D^7+D^5+D^2+D^1+1  
D^7+D^5+D^3+D^1+1  
D^7+D^5+D^4+D^3+1  
D^7+D^5+D^4+D^3+D^2+D^1+1  
D^7+D^6+1  
D^7+D^6+D^3+D^1+1  
D^7+D^6+D^4+D^1+1  
D^7+D^6+D^4+D^2+1  
D^7+D^6+D^5+D^2+1  
D^7+D^6+D^5+D^3+D^2+D^1+1  
D^7+D^6+D^5+D^4+1  
D^7+D^6+D^5+D^4+D^2+D^1+1  
D^7+D^6+D^5+D^4+D^3+D^2+1

**输出一个ans，对ans转置（十进制）：**

ans ‘=

131;137;143;145;157;167;171;185;191;193;203;211;213;229;239;241;247;253

上面的数字是十进制，转化为**八进制**为：

203;211;217;221;235;247;253;271;277;301;313;323;325;345;357;361;367;375;

上面的数字是十进制，转化为二进制为：  
ans =

10000011

10001001

10001111

10010001

10011101

10100111

10101011

10111001

10111111

11000001

11001011

11010011

11010101

11100101

11101111

11110001

11110111

11111101

## 输入mseq([1 0 0 0 0 0 1])

便可得到**一个127位的m序列**，整理之后为

1000000111111101010100110011101110100101100011011110110101101100100100011100001011111001010111001101000100111100010100001100000

## 解释说明：

**1 0 0 0   0 0 1**这个输入参数的物理意义为**各寄存器的初始状态**，**其数学意义是所选用的本原多项式D^7+D^1+1中D^1到D^7的各个系数**。读者可以试试自行选用其他本原多项式，得到不同的m序列。

例如如果选用**D^7+D^3+1**，则只需调用

mseq([0 0 1 0 0 0 1])，得到的m序列为

1000000100100110100111101110000111111100011101100010100101111101010100001011011110011100101011001100000110110101110100011001000

# M序列的性质

## M序列具有很好的自相关性；

# M序列的matlab代码实现

**function [M\_Seq] = My\_M\_Seq(primitive,seed,N)**

%根据原本多项式生成M序列

%参数介绍：

% 1. primitive：本原多项式系数,按照幂升序排列,即[C1,C2,...,Cn]，C0=1缺省

% 2. seed为初始状态，按照An-1，An-2,...,A0递减顺序

% 3. N：所需要的M序列的长度，M序列的循环周期为2^n-1,n为M序列级数

%说明： 输入参数可以为1个，就是原本多项式系数；也可以为2个，原本多项式和初始状态

n = length(primitive);%M序列的级数

if nargin == 1 % 若输入参数为1个，则默认初始状态为[0 0 0... 1]

seed = [zeros(1,n-1),1];

N = 2^n-1;%M序列的最小周期

else if nargin == 2% 若输入参数为2个，则默认获取长度为N=2^n-1

N = 2^n-1;

end

end

M\_Seq = zeros(1,N);%初始化M序列为0

for i = 1:N

M\_Seq(i) = seed(n);%seed数组向右移动，M1-Mn都是初始状态

An = mod(sum(primitive.\*seed),2);%获取新的m值

seed = [An,seed(1:n-1)];%新的状态

end

end

下载简化版mseq.m文件,内容为:

(经过验证，两个程序结果相同)

function[mseq]=m\_sequence(fbconnection)

n=length(fbconnection);   
N=2^n-1;    
register=[zeros(1,n-1) 1];  %移位寄存器的初始状态   
mseq(1)=register(n);        %m序列的第一个输出码元   
for i=2:N

newregister(1)=mod(sum(fbconnection.\*register),2);

for j=2:n

newregister(j)=register(j-1);       
end;

register=newregister;       
mseq(i)=register(n);

end

end