Back to Resources

LFSR Implementations
Feedback Tap Conventions
Maximal Length Sequences
M-Sequence Math
M-Sequence Properties
M-Sequence Taps

Linear Feedback Shift Registers

Implementation, M-Sequence Properties, Feedback Tables

Brought to you by New Wave Instruments, the leading manufacturer of PN Sequence & PRBS Generators for use in the development of data communication, wireless, and spread spectrum systems.

A linear feedback shift register (LFSR) is the heart of any digital system that relies on pseudorandom bit sequences (PRBS), with applications ranging from cryptography and bit-error-rate measurements, to wireless communication systems employing spread spectrum or CDMA techniques.

In this article we discuss the two implementations of LFSR generators, how to determine feedback taps for generating a maximal length sequence, and the properties of maximal length sequences (m-sequences). We also provide tables of m-sequence feedback taps.

LFSR Generator Implementations

Linear feedback shift registers can be implemented in two ways. The Fibonacci implementation consists of a simple shift register in which a modulo-2 sum of the binary-weighted taps is fed back to the input. (The modulo-2 sum of two 1-bit binary numbers yields 0 if the two numbers are identical, and 1 if the differ: 0+0=0, 0+1=1, 1+1=0.)

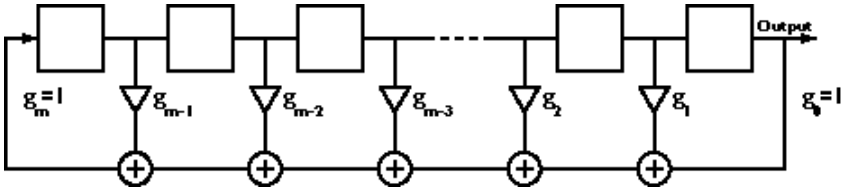


Figure 1. Fibonacci implementation of LFSR.

For any given tap, weight g_i is either 0, meaning "no connection," or 1, meaning it is fed back. Two exceptions are g_0 and g_m, which are always 1 and thus always connected. Note that g_m is not really a feedback connection, but rather is the input of the shift register. It is assigned a feedback weight for mathematical purposes, as is explained below.

The Galois implementation consists of a shift register, the content of which is modified at every step by a binary-weighted value of the output stage, again using modulo-2 math.

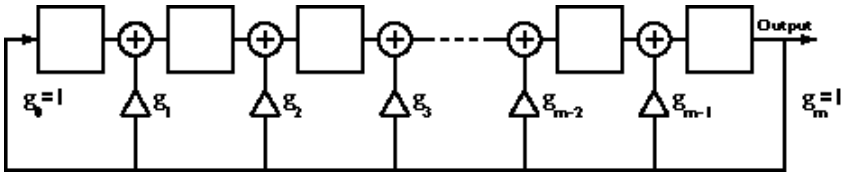


Figure 2. Galois implementation of LFSR.

Careful inspection reveals that the order of the Galois weights is opposite that of the Fibonacci weights.

initial vector, and this initial fill comprises the first m bits output from the ge... state ... generator must be adjusted appropriately to attain the equivalent output of its first m bits. (In mathematical literature, the initial state of either form is called the *seed*.)

When implemented in hardware, modulo-2 addition is performed using exclusive-OR (XOR) gates. The Galois form is generally faster than the Fibonacci in hardware due to the reduced number of logic gates in the feedback loop, thus making it the favored form.

It should be noted that, in some industries, the Fibonacci form LFSR is referred to as a *simple shift register generator* (SSRG), and the Galois form is referred to as a *multiple-return shift register generator* (MRSRG) or *modular shift register generator* (MSRG).

Conventions for Feedback Tap Specification

A given set of feedback connections can be expressed in a convenient and easy-to-use shorthand form, with the connection numbers being listed within a pair of square brackets. In doing so, connection g_0 (defined in Figures 1 and 2) is implied, and not listed, since it is always connected. Although g_m is also always connected, it is listed in order to convey the shift register size (i.e. the number of registers).

Specifically, a set of feedback connections, or taps, is denoted as

$$[f_1, f_2, f_3, ..., f_J]$$

where subscript J is the total number of feedback taps (not including g_0), $f_1 = m$ is the highest-order feedback tap (and the size of the LFSR), and f_j represent the remaining feedback taps. The value of each f_j is equal to the subscript of the corresponding connection g. Note that the tap numbers f_j are customarily arranged in descending order from left to right.

A set of feedback taps specified in this format is called a *feedback tap set*, *feedback set*, or *feedback pattern*. As an example, the [8, 4, 3, 2] feedback set would signify an eight-stage shift register with feedback connections at taps g_8, g_4, g_3, g_2, and, as always, at g_0.

A related convention is that an LFSR with m shift register stages is said to be an Rm LFSR. For example, an R8 generator is one with eight stages. An alternative to this convention is PNm, or PN8 in this example. (PN is an acronym for pseudonoise, which is a term used in some industries for maximal length pseudorandom sequences, which are discussed below.)

A Competing Convention

The reader might be aware that, in other literature or in some circles, it is noted that the tap order must to be reversed when switching between the Fibonacci and Galois forms of LFSR. The reason this isn't the case here in this article is because the tap numbers in the Galois implementation in Figure 2 are reversed from those in Fibonacci implementation in Figure 1. This conveniently takes the reversal into account, and so any given feedback set will produce the same sequence in either implementation.

However, if the tap numbers aren't reversed, the feedback sets for the Fibonacci and Galois forms must be made distinguishable from each other. For sake of completeness, we will now describe the feedback set convention for that literature or mindset. (None of the following applies this article.)

A set of feedback taps for a Galois generator is denoted as

$$[f_1, f_2, f_3, ..., f_J]_g$$

where subscript J is the total number of feedback taps (not including input g_0), $f_1 = m$ is the highest-order feedback tap (and the size of the LFSR), and f_j are the remaining feedback taps. The value of each f_j is equal to the subscript of the corresponding connection g. The g subscript on the right bracket signifies the Galois LFSR form.

The set of feedback taps for the equivalent Fibonacci generator is denoted as

$$[f_1, m\text{-}f_2, m\text{-}f_3, ..., m\text{-}f_J]_f$$

where the f subscript on the right bracket signifies the Fibonacci LFSR form. Note that subtracting the feedback tap numbers from m is equivalent to reversing the order of the feedback taps.

Maximal Length Sequences

LFSR generators produce what are called *linear recursive sequences* (LRS) because all operations are linear. Generally speaking, the length of the sequence before repetition occurs depends upon two factors, the feedback taps and the initial state. An LFSR of any given size m (number of registers) is capable of producing every possible state during the period $N=2^m-1$ shifts, but will do so only if proper feedback taps have been chosen. For example, such an an eight stage LFSR will contain every possible combination of ones and zeros after 255 shifts. Such a sequence is called a *maximal length sequence*, *maximal sequence*, or less commonly, *maximum length sequence*. It is often abbreviated as *m-sequence*. In certain industries m-sequences are referred to as a *pseudonoise* (PN) or *pseudorandom* sequences, due to their optimal noise-like characteristics. (Informally, even non-maximal sequences are often called pseudonoise or pseudorandom sequences.)

Technically speaking, maximal length generators can actually produce two sequences. The first--the trivial one--has a length of one, and occurs when the initial state of the generator is set to all zeros. (The generator simply remains in the zero state indefinitely.) The other one--the useful one--has a length of 2^m-1. Together, these two sequences account for all 2^m states of an m-bit state register.

When the feedback taps of an LFSR are non-maximal, the length of the generated sequence depends upon the initial state of the LFSR. A non-maximal generator is capable of producing two or more unique sequences (plus the trivial all-zeros one), with the initial state determining which is produced. Each of these sequences is referred to as a *state space* of the generator. Together, every non-maximal sequence the generator can produce accounts for all 2^m states of an m-bit state register.

Properties of non-maximal sequences are generally inferior to those of maximal sequences. So the use of non-maximal sequences in real systems is usually avoided in favor of their maximal-length counterparts.

Galois Field Mathematics and M-Sequences

Finite (Galois) field mathematics are used to derive m-sequence feedback taps. Any LFSR can be represented as a polynomial of variable X, referred to as the *generator polynomial*:

$$G(X) = g_m X^m + g_{m-1} X^{m-1} + g_{m-2} X^{m-2} + \ldots + g_2 X^2 + g_1 X + g_0$$

Equation 1. Generalized generator polynomial.

The coefficients g_i represent the tap weights, as defined in Figures 1 and 2, and are 1 for taps that are connected (fed back), and 0 otherwise. The order of the polynomial, m, represents the number of LFSR stages. Rules of linear algebra apply to the polynomial, but all mathematical operations are performed in modulo-2:

Modulo-2 addition:

0 + 0 = 0
0 + 1 = 1
1 + 1 = 0

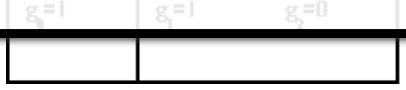
Modulo-2 multiplication:

0 * 0 = 0
0 * 1 = 0
1 * 1 = 1

As an example of polynomial representation, the generator polynomial

$$G(X) = X^3 + X^1 + 1$$

represents an LFSR with feedback taps 3 and 1, denoted as [3, 1]:



The constant 1 in the generator polynomial represents the input connection of the shift register, g_0.

Now, here is the key to determining m-sequence feedback taps: The generator polynomial of Equation 1 is said to be *primitive* if it cannot be factored (i.e. it is prime), and if it is a factor of (i.e. can evenly divide) X^N+1, where $N = 2^m-1$ (the length of the m-sequence). It can be shown that an LFSR represented by a primitive polynomial will produce a maximal length sequence.

Consider again the example of the [3, 1] LFSR just given. We wish to know if this generator will produce an m-sequence. First we note that m = 3 and $N=2^3-1=7$. It can be shown that its polynomial, X^3+X^1+1 , cannot be factored, and it can be shown that its polynomial is a factor of X^7+1. Thus, we conclude that this LFSR will indeed produce a maximal length sequence.

In this example, we went through the process of determining whether or not the given set of feedback taps would result in a maximal length sequence. Normally, however, we are required to do just the opposite. That is, we are normally asked to find all sets of feedback taps that will produce m-sequences for a given register size.

For example, we may be asked to find all sets of maximal-length feedback taps for an LFSR with m=3 registers. We do this as follows: The length of the m-sequences will be $N=2^3-1=7$. We know that the solution lies in all the primitive factors of polynomial X^7+1. We use modulo-2 linear algebra (probably with the aid of a computer algorithm) to find the prime factors to be

$$X^7+1 = (X+1)(X^3+X+1)(X^3+X^2+1)$$

The primitive polynomials are those factors whose order is the same as the register size, m = 3. Of the three prime factors we see here, the last two meet this criterion. Thus we see that there are exactly two sets of m-sequence feedback taps, [3, 1] and [3, 2].

It is interesting to note that, given any shift register size, there will always be an even number of m-sequence feedback sets. More specifically, given any one of its m-sequence feedback sets,

$$[f_1, f_2, f_3, ..., f_J]$$

there will be a companion set described as

$$[f_1, m\text{-}f_2, m\text{-}f_3, ..., m\text{-}f_J]$$

whose sequence will be the mirror image of the original set's sequence. Note that subtracting feedback tap numbers from m for the companion set is equivalent to reversing the order of those taps. Thus we conclude that, for any given feedback set that produces an m-sequence, the mirror image of the feedback set will produce the mirror image of the m-sequence. And, of course, the resulting sequence will also be an m-sequence since all possible states are exhausted. An astute reader may have noticed in the last example that the two derived sets of m-sequence feedback taps, [3, 1] and [3, 2], are in fact mirror images of each other.

Tables of m-sequence feedback taps are presented below for the reader's convenience.

If the reader wishes to determine m-sequence feedback sets not available here, the method described in this section can be used to do so. Howerver, writing a software that performs modulo-2 polynomial factorization using, for example, the famous Berlekamp algorithm will typically not be a trivial task for a non-mathematician. Fortunately there are off-the-shelf solutions available. Maplesoft sells a product called Maple which includes a function called Berlekamp. This will return all primitive polynomial factors of any given polynomial. If you own MatLab or some other popular numerical computing environment, you might be able to find and download a free copy of a factorization script written by another user.

M-Sequence Properties

Properties of m-sequences include the following:

1. An m-bit register produces an m-sequence of period 2^m-1.

3. The modulo-2 sum of an m-sequence and another phase (i.e. time-delayed version) of the same sequence yields yet a third phase of the sequence.

3a. (A corollary of 3.) Each stage of an m-sequence generator runs through some phase of the sequence. (While this is obvious with a Fibonacci LFSR, it may not be with a Galois LFSR.)

4. A sliding window of length m, passed along an m-sequence for 2^m-1 positions, will span every possible m-bit number, except all zeros, once and only once. That is, every state of an m-bit state register will be encountered, with the exception of all zeros.

5. Define a run of length r to be a sequence of r consecutive identical numbers, bracketed by non-equal numbers. Then in any m-sequence there are:

1 run of ones of length m.
1 run of zeros of length m-1.
1 run of ones and 1 run of zeros, each of length m-2.
2 runs of ones and 2 runs of zeros, each of length m-3.
4 runs of ones and 4 runs of zeros, each of length m-4.
.
.
.
2^{m-3} runs of ones and 2^{m-3} runs of zeros, each of length 1.

6. If an m-sequence is mapped to an analog time-varying waveform, by mapping each binary zero to -1 and each binary one to +1, then the autocorrelation function for the resulting waveform will be unity for zero delay, and $-1/(2^m-1)$ for any delay greater that one bit, either positive or negative in time. The shape of the autocorrelation function between -1 bit and +1 bit will be triangular, centered around time 0. That is, the function will rise linearly from time = -(one-bit) to time 0, and then decline linearly from time 0 to time = +(one-bit).

Other interesting facts regarding m-sequences and feedback sets that produce them include the following:

1. If the order of the feedback taps (as defined in Figures 1 and 2) is reversed, the resulting sequence will be the time reversal of the original sequence, and will also be an m-sequence.

2. The feedback set for any given m-sequence consists of an even number of taps, never odd (as defined in Figures 1 and 2, including output g_m but not including input g_0).

Tables of M-Sequence Feedback Taps

The following tables contain m-sequence feedback sets for LFSR sizes R3 through R32. The tables for R3 through R24 contain all maximal feedback sets (except for mirror image sequences). The tables for R25 through R32 contain maximal feedback sets for 2, 4, and 6 taps only. The last table contains a sampling of "dense feedback sets" (those with taps for at least half the stages) for R25 through R32 registers.

The table files are in text format. However, the files for R21 through R24 have been zipped for faster downloading.

Reversing the tap order for any given feedback set will result in another valid m-sequence feedback set. These reversed sets are not listed in the tables. The sequence produced with the reversed taps will be the mirror image of the sequence produced with the original taps. If the original feedback set is [m, A, B, C], the reversed feedback set is described by [m, m-C, m-B, m-A], where m is the number of LFSR stages.

3 Stages	11 Stages	19 Stages	27 Stages
4 Stages	12 Stages	20 Stages	28 Stages
5 Stages	13 Stages	21 Stages	29 Stages
6 Stages	14 Stages	22 Stages	30 Stages
7 Stages	15 Stages	23 Stages	31 Stages
8 Stages	16 Stages	24 Stages	32 Stages
9 Stages	17 Stages	25 Stages	25-32 Dense
10 Stages	18 Stages	26 Stages	

Related Search Terms

For the benefit of those searching the Web, following are search terms related to this page:

Spread Spectrum Topics:

Auto-Correlation, Balance Property, Chip Rate, Code Division Multiple Access (CDMA), Direct Sequence Spread Spectrum (DSSS), Finite Field Arithmetic, Frequency Hopping Spread Spectrum (FHSS), Galois Field Arithmetic, Generation, Gold Code, LFSR Tutorial, Linear Feedback Shift Register (LFSR), Linear Feedback Shift Registers (LFSR), M-Sequence Tutorial, M-Sequences, Maximal Length Sequence, Maximal Length Sequences, Maximum Length, ML, Orthogonal, Orthogonality, Periodic, PN Code, PN Codes, PRBS, Pseudonoise, Pseudo-Noise, Pseudorandom, Pseudo-Random, Run-Length, Shift-and-Add Property, Spreading Codes, Spread Spectrum Modulation, Synchronize, Synchronizer, Synchronization

RF & Microwave Topics:

Bit Rate, Cross-Correlation, Data Rate, Digital, Signal, Signals

Resources:

Block Diagram, Circuit Diagram, Circuit Diagrams, Circuits, Design, How to, PDF, OEM Product, OEM Products, System, Technique, Technology, Table, Test Equipment, Tester, Theory, Tutorial, Tutorials, What is

Misspellings:

Crosscorrelation

British Equivalents:

Synchronise, Synchroniser, Synchronisation

 Page revised on 04/05/10.