Chapter 9
Pseudo-Random Binary Sequences and Data Scramblers

Contents

Chapter 9
Pseudo-Random Binary Sequences and Data Scramblers

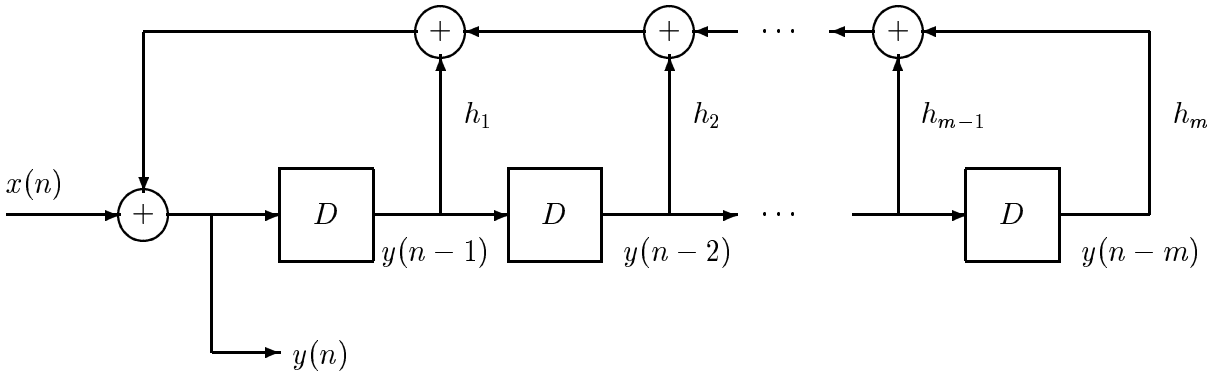
- You should complete this experiment in one week. It is like a homework assignment in a regular course and does not require using the DSP boards.
- You can use the C compiler on the PC's in the lab and get help from your TA during the lab period and/or use a C compiler on any computer you prefer.

GOALS:

To generate pseudo-random binary sequences that can be used for:

1. testing digital communication systems
2. scrambling input data to break up long strings of 0's or 1's

Linear Feedback Shift Register Sequence Generator



$$y(n) = x(n) + \sum_{k=1}^{m} h_k y(n-k)$$

- All constants and variables can only have the values 0 or 1.
- All additions are performed "modulo 2" (exclusive-or).

$$0+0=0, \quad 0+1=1+0=1, \quad 1+1=0$$

- The State Vector

$$\begin{aligned} \mathbf{s}(n) &= [y(n-1), y(n-2), \cdots, y(n-m)] \\ &= [s_1(n), s_2(n), \cdots, s_m(n)] \end{aligned}$$

The Huffman Transform

(Like z-transform with $D = z^{-1}$)

Let $x(n)$ be a binary sequence. Then

$$X(D) = \sum_{n=0}^{\infty} x(n)\, D^n$$

The Connection Polynomial

$$h(D) = 1 + \sum_{k=1}^{m} h_k D^k$$

Output Transform with Zero Initial State

$$y(n) + \sum_{k=1}^{m} h_k y(n-k) = x(n)$$

So

$$Y(D)h(D) = X(D) \quad \text{or} \quad Y(D) = X(D)/h(D)$$

Properties of Ideal Binary Random Sequences

- 0 and 1 are equally likely. That is,
  $P\{x(n) = 1\} = P\{x(n) = 0\} = 1/2$
- The bits are independent. That is,
  $P\{x(n) = i \cap x(m) = j\} =$
  $P\{x(n) = i\}\, P\{x(m) = j\}$ for $n \neq m$
- Therefore, they are uncorrelated.
  $\mathrm{cov}\{x(n), x(m)\} = 0$ for $n \neq m$

Pseudo-Noise (PN) or Maximal Length Sequences

Let $x(n) = 0$ for all n. Then the shift register generator output satisfies the homogeneous equation

$$y(n) + \sum_{k=1}^{m} h_k y(n-k) = 0$$

The Trivial Solution

If the initial state is $\mathbf{s}(n) = 0$,
then $y(n) = 0$ for all n.

Non-Trivial Solutions

- It can be shown that any non-zero initial state results in a non-trivial solution which never becomes identically zero.
- The state vector can have at most $N = 2^m - 1$ possible non-zero values so it must repeat at some time. Then the output and state sequence will repeat.
- So the shift register will generate a periodic output sequence with a period at most $2^m - 1$. Sequences with period N are called **PN** or **maximal length** sequences.

Properties of Non-Trivial Solutions

The properties of the non-trivial solutions are determined by the connection polynomial $h(D)$.

Connection Polynomial for Maximal Length Sequences

Primitive polynomials must be used for $h(D)$ to obtain maximal length.

- Primitive polynomials of all degrees exist.
- A necessary but not sufficient condition for a polynomial to be primitive is that it be *irreducible*. A polynomial with binary coefficients is said to be irreducible over the field of binary numbers if it cannot be factored into the product of polynomials with binary coefficients and degrees at least 1. (An irreducible polynomial is not necessarily primitive.)
- Irreducible polynomials over the binary field must have an odd number of 1 coefficients. Otherwise $h(1) = 0$ and $D + 1$ is a factor.

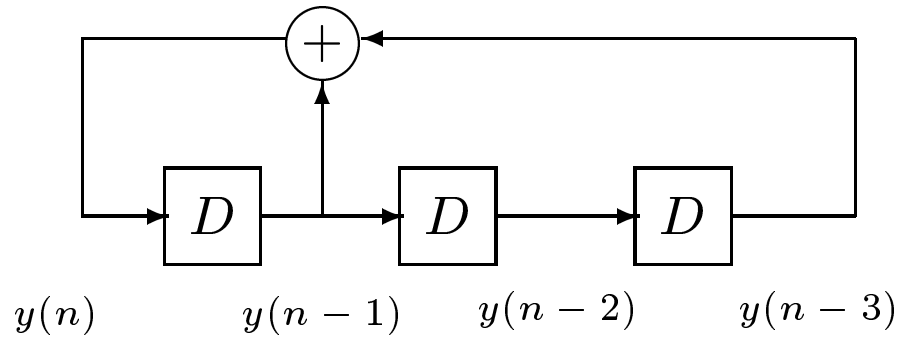
Sequence Period When $h(D)$ is Irreducible

- The period with an irreducible connection polynomial $h(D)$ of degree m is the smallest nonzero integer L such that $D^L - 1$ is divisible by $h(D)$ using modulo 2 arithmetic for the coefficients. L is called the *exponent* of $h(D)$.
- It can be shown that $h(D)$ always must divide $D^{2^m-1} - 1$. L may be smaller than $N = 2^m - 1$ but must divide it.
- $h(D)$ is primitive when $L = 2^m - 1$.

EXAMPLE

Let $h(D) = D^3 + D + 1$. Notice that $h(1) = h(0) = 1$, so $D + 1$ and D are not factors. Also $h(D)$ cannot be the product of a 1st and 2nd degree factor because it has no 1st degree factors. Therefore, $h(D)$ is irreducible.

EXAMPLE (cont.)



$$y(n) = y(n-1) + y(n-3) \mod 2$$

n	$y(n)$	$y(n-1)$	$y(n-2)$	$y(n-3)$
0	1	1	0	0
1	1	1	1	0
2	0	1	1	1
3	1	0	1	1
4	0	1	0	1
5	0	0	1	0
6	1	0	0	1
7	1	1	0	0

$N = 2^3 - 1 = 7$, so $h(D)$ is primitive.

Properties of PN Sequences

1. **Frequency of 1's and 0's**
   Each period contains 2^{m-1} ones and $2^{m-1}-1$ zeros.

2. **Frequency of Runs of 1's and 0's**
   A run of k 1's is defined to be a string starting with a 0, followed by k 1's, and ending with a 0. Runs of 0's are defined similarly.

   - In one period of a maximal length sequence, there is one run of m 1's and no run of $m-1$ 1's. For $1 \leq k \leq m-2$, there are 2^{m-k-2} runs of k 1's.
   - There is no run of m 0's, one run of $m-1$ 0's, and 2^{m-k-2} runs of k 0's for $1 \leq k \leq m-2$.

3. **Autocorrelation Function**
   In discussing correlation properties, it will be convenient to transform sequences of 0's and 1's into sequences of $+1$'s and -1's.

Properties of PN Sequences (cont.)

Let $y(n)$ be a sequence with period N that can have the value 0 or 1. The transformed sequence is

$$\check{y}(n) = \begin{cases} +1 & \text{if } y(n) = 0 \\ -1 & \text{if } y(n) = 1 \end{cases}$$

The periodic autocorrelation function is

$$R(n) = \frac{1}{N} \sum_{k=0}^{N-1} \check{y}(k)\check{y}(n+k)$$

where the sum is performed using ordinary addition.

For maximal length sequences with period $N = 2^m - 1$,

$$R(n) = \begin{cases} -\frac{1}{N} & \text{for } n \text{ not a multiple of } N \\ 1 & \text{for } n \text{ a multiple of } N \end{cases}$$

Periods with Reducible Connection Polynomials

The situation is more complicated when the connection polynomial is reducible, that is, can be factored. Suppose

$$h(D) = \prod_{k=1}^{L} f_k(D)$$

where each factor is irreducible and has exponent N_k. In this case, the shift register will generate sequences with different periods depending on its initial state. Each period will be the product of a subset of the exponents.

Self Synchronizing Data Scramblers

Long strings of 1's and 0's must be randomized before being applied to the transmitter in a modem or else the carrier recovery loop, symbol clock tracker, and adaptive equalizer in the remote receiver will not work properly.

The system in Slide 9-2 is a *self synchronizing* scrambler with input $x(n)$ and output $y(n)$. A primitive connection polynomial is usually used.

The Scrambler Input/Output Equation

$$y(n) = x(n) + \sum_{k=1}^{m} h_k y(n-k)$$

In terms of the transform notation

$$Y(D) = \frac{X(D)}{h(D)}$$

The Descrambler

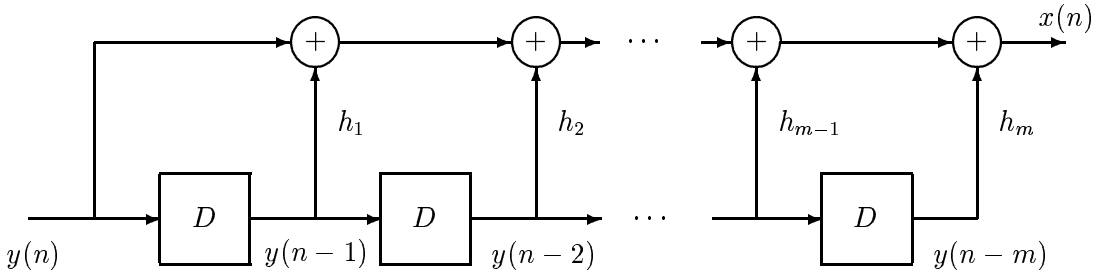
The descrambling equation is

$$x(n) = y(n) + \sum_{k=1}^{m} h_k y(n-k)$$

or in transform notation

$$X(D) = Y(D)h(D)$$

The descrambler is just an FIR filter with $m+1$ taps that uses modulo 2 arithmetic. A block diagram of the descrambler is shown in the following figure.



If $y(n)$ is corrupted by channel errors, the scrambler output will also contain errors. If $h(D)$ has K nonzero coefficients, a single isolated error in $y(n)$ will cause K output errors as it propagates by the nonzero coefficients. Therefore, a connection polynomial with the least number of nonzero coefficients should be chosen.

Exercises for a Primitive Connection Polynomial

$h(D) = 1 + D^2 + D^5$ is a primitive polynomial. Perform the following exercises:

1. Let $x(n) = 0$ for all n and the initial state not be **0**. Calculate the theoretical period N of the output sequence from the shift register generator.

2. Write a C program to implement the scrambler including an input $x(n)$ which is set to 0. Store the shift register state as 5 consecutive bits in a single integer variable. Update the state by or-ing each new output into the appropriate bit in this integer variable and then shifting it with one of the C shift operators.

3. Set the state of your shift register to a nonzero value. Generate and record enough outputs to verify your calculation of the period.

4. Count the number of 1's and 0's in one period of your sequence and check that the results agree with the theory.

5. Count the number of runs of 1's and 0's of each possible length in one period of your sequence and make a table showing the results. Make sure they agree with the theoretical values.

6. Compute the scaled periodic autocorrelation function $NR(n)$ for $n = 0, 1, \ldots, N-1$ from your sequence and check that it agrees with the theoretical result.

7. Now write a C program to implement the descrambler. Again, store the descrambler shift register as a string of 5 consecutive bits in a single C integer variable. Let the input to the scrambler be $x(n) = 1$. Generate a scrambled sequence, put it through the descrambler, and check that the descrambler output is all 1's. (Notice that the initial states of the scrambler and descrambler do not have to be identical if an initial burst of errors is acceptable as the descrambler shift register fills up with received bits.)

Exercises for an Irreducible Non-Primitive $h(D)$

Let $h(D) = 1 + D + D^2 + D^3 + D^4$. This $h(D)$ is irreducible but not primitive.

1. Find the period N for this sequence generator.

2. The four stage shift register can have $2^4 - 1 = 15$ nonzero values. Let $y(n)$ be the sequence generated by a particular nonzero state. Consider $y(n)$ and the sequences obtained by delaying $y(n)$ by $1, 2, \ldots, N-1$ samples to be an *equivalence class* of N sequences. It can be shown that each member of the equivalence class corresponds to a unique shift register initial state. Thus, there must be $(2^4 - 1)/N$ equivalence classes. Find one member of each equivalence class and its corresponding initial shift register state.

Exercises for a Reducible $h(D)$

$$h(D) = (1 + D + D^2)(1 + D + D^3) = 1 + D^4 + D^5$$

1. Verify that the product is correct.
2. Both factors are irreducible. Find the exponents for the factors.
3. Find initial states that result in sequences with periods 3, 7, and 21. Record the states and corresponding sequences.