Institute for Computer Engineering (ZITI), Heidelberg University, Germany
email address: peter.fischer@ziti.uni-heidelberg.de

In this text I will show how the period of the sequence obtained in a LFSR with exclusive-or feedback can be calculated. In particular, the conditions to obtain the maximum possible period of $2^{\mathrm{n}} - 1$ for a register of n bit length will be clarified.

1 Introduction

When a digital shift register of n bit length is fed with an exclusive-or combination of the output signal and some of the internal bits, sequences with very interesting properties can be generated. One particularly interesting case is when the shift register cycles through all possible bit combinations, which gives a sequence of length $2^N - 1$ (one less than the total number of bit combinations, because the state with all zeros is stable). Such sequences have very interesting properties so that they are commonly used as noise generators, pseudo random number generators, counters or binary test pattern generators. In order to obtain such a *maximum length sequence*, the extra internal bits used in the exclusive-or must be at the correct positions. In this text, I want to show how the taps can be found for a given register length n. The intention is to really go through all the steps in the derivation and to make it understandable for a non mathematics expert. The derivation summarizes the approach presented in [1] and in the standard book on shift registers from Solomon W. Golomb [2]. The following text requires very little mathematical knowledge, just a bit of matrix algebra and no fear of sum notations, indices and so on. Many expressions could be written down in a much more general manner, but for simplicity, I restrict everything to the simplest case.
The derivation will start with a formalization of what is going on in the shift register. We will mix in an external input signal also (in addition to the exclusive-or of an arbitrary combination of shift register bits) to be able to excite the initially empty shift register at time $t = 0$. We will derive a general expression for the output state after an arbitrary number of clocks. We will then define the *z-transformation* and use it to extract periodicity information from the output sequence. At the end, we will find a criterion which must be met by the exclusive-or pattern to yield an output sequence of a given period.

2 Calculation of the output signal

Shift Register Description. We consider a shift register with n bit length. The state of the shift register flip flops is described by the n numbers $z_i(t) \in [0, 1]$ with $i = 1 \ldots \mathrm{n}$. The integer time argument $t = 0, 1, \ldots \infty$ tells us how many clocks have elapsed. z_1 is the output of the shift register, while z_{n} is the value of the first flip flop in the chain.

We denote the output also as $y(t) := z_1(t)$ (see fig. 1). For compact notation, we collect the $z_i(t)$ in a (column) vector

$$\vec{z}(t) := \begin{pmatrix} z_1(t) \\ z_2(t) \\ \vdots \\ z_n(t) \end{pmatrix}. \tag{1}$$

We start with an empty shift register at time $t = 0$:

$$\vec{z}(0) = \vec{0}. \tag{2}$$

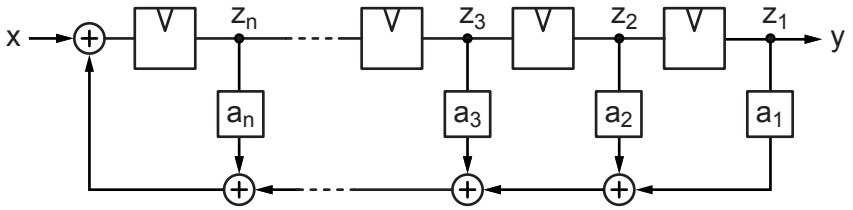

Figure 1: 'Fibonacci' type linear shift register with exclusive-or feedback and input signal

Feedback and Input Signal. Some internal signals of the register which we denote as *taps* and an additional input signal $x(t)$ are exor-ed together. This signal forms the input to the first flip flop (see fig. 1). For a general treatment, we multiply every possible tap z_i by a coefficient $a_i \in [0, 1]$ $(i = 1 \ldots \mathrm{n})$ and sum up all products plus the external input x. When we do the summing modulo 2 (i.e. 1+1 = 0), this sum is just an exclusive or. (Mathematicians would say that we are working on GF(2), the *Galois Field* with two elements.) It is clear that we always must use the tap at the end of the register (i.e. we must have $a_1 = 1$), as we would not really use the n bits otherwise.
For our purpose, we can restrict the input signal $x(t)$ to a single excitation at $t = 0$, i.e. we have a *delta pulse* at the input:

$$x(t) = x_\delta(t) := \delta_{t,0} = \begin{cases} 1 & \text{for} \quad t = 0 \\ 0 & \text{for} \quad t > 0 \end{cases}. \tag{3}$$

General Output Signal. For the output $y(t)$ and the state bits $z_i(t)$ we obviously have

$$\begin{aligned} z_i(t+1) &= z_{i+1}(t) \quad \text{for} \quad i = 1 \ldots \mathrm{n}-1 \\ z_\mathrm{n}(t+1) &= x(t) + \sum_{i=1}^{\mathrm{n}} a_i \cdot z_i(t) \\ y(t) &= z_1(t). \end{aligned}$$

The sum is calculated modulo 2, of course. This can be written very nicely using a matrix notation as

$$\begin{aligned} \vec{z}(t+1) &= \mathbf{A}\,\vec{z}(t) + \mathbf{B}\,x(t) && (4)\\ y(t) &= \mathbf{C}\,\vec{z}(t). && (5) \end{aligned}$$

$\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ have dimensions of $\mathrm{n} \times \mathrm{n}, 1 \times \mathrm{n}$ and $\mathrm{n} \times 1$, respectively, and are defined as

$$\mathbf{A} := \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ a_1 & a_2 & a_3 & \cdots & a_{\mathrm{n}} \end{pmatrix}, \mathbf{B} := \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \tag{6}$$

and

$$\mathbf{C} := (1\,0\,0\,\ldots\,0)\,. \tag{7}$$

Output Signal after Time t. We now use (4) to calculate the shift register state for successive time steps starting at $t = 0$ with the initial condition (2) and the properties of the input signal (3):

$$\begin{aligned} \vec{z}(1) &= \mathbf{A}\,\vec{z}(0) + \mathbf{B}\,x(0) = \mathbf{B} \\ \vec{z}(2) &= \mathbf{A}\,\vec{z}(1) + \mathbf{B}\,x(1) = \mathbf{A}\,\mathbf{B} \\ \vec{z}(3) &= \mathbf{A}\,\vec{z}(2) + \mathbf{B}\,x(2) = \mathbf{A}^2\mathbf{B} \\ &\vdots \\ \vec{z}(t) &= \mathbf{A}^{t-1}\,\mathbf{B}. && (8) \end{aligned}$$

According to (5), the output is therefore

$$y(t) = \mathbf{C}\,\mathbf{A}^{t-1}\,\mathbf{B}. \tag{9}$$

3 The z-Transformation

For the next steps, we will need the *z-transform* $Z[f]$ of an infinite time sequence of function values given by $\{f(t)\} = f(0), f(1), \ldots$. It is defined as

$$Z[f(t)] := \sum_{j=0}^{\infty} f(j)\,s^j. \tag{10}$$

The z-transform has a few interesting properties which we will derive next.

z-Transform of a Delta Pulse. The z-transform of a delta pulse at $t = 0$ is constant:

$$Z[x_\delta(t)] = \sum_{j=0}^{\infty} \delta_{t,0}\,s^j = 1. \tag{11}$$

z-Transform of a Time Shifted Signal. When the input sequence is shifted by one time step, the z-transform is basically just divided by s:

$$\begin{aligned} Z[f(t+1)] &= \sum_{j=0}^{\infty} f(j+1)\, s^j = \sum_{j=1}^{\infty} f(j)\, s^{j-1} \\ &= \frac{Z[f(t)] - f(0)}{s}. \end{aligned} \tag{12}$$

z-Transform of a Periodic Signal. Let us assume that the sequence $f(t)$ has a period of P starting at time $t = \tau$, i.e. that the sequence looks like this:

$$\begin{aligned} \{f(t)\} = \; & r_0, r_1, \ldots, r_{\tau-1}, && \text{('preamble')} \\ & r_\tau, r_{\tau+1}, \ldots, r_{\tau+P-1}, && \text{(first period)} \\ & r_{\tau+P}, r_{\tau+1+P}, \ldots && \text{(next periods)} \end{aligned} \tag{13}$$

with $r_{\tau+P} = r_\tau$, $r_{\tau+1+P} = r_{\tau+1}$ and so on. The z-transform of this periodic signal is given by

$$\begin{aligned} Z[f(t)] &= r_0 + r_1 s + \ldots + r_{\tau-1} s^{\tau-1} + \\ &\quad r_\tau s^\tau + \ldots + r_{\tau+P-1} s^{\tau+P-1} + \\ &\quad r_{\tau+P} s^{\tau+P} + r_{\tau+2P-1} s^{\tau+2P-1} + \ldots \\ &= \sum_{i=0}^{\tau-1} r_i s^i + s^\tau (1 + s^P + \ldots) \times (r_\tau + r_{\tau+1} s + \ldots + r_{\tau+P-1} s^{P-1}) \\ &= \sum_{i=0}^{\tau-1} r_i s^i + \frac{s^\tau}{1 - s^P} \sum_{i=0}^{P-1} r_{\tau+i} s^i. \end{aligned} \tag{14}$$

We have simplified the expression by replacing the infinite geometric sum $(1 + s^P + \ldots)$ by $(1 - s^P)^{-1}$.

4 z-Transform of the Output Sequence

We now use the z-transform to study the output signal $y(t)$ as given by (9). We will then be able to analyze the periodicity of $y(t)$. We start by applying the z-transform to (4) using the delta pulse input (3):

$$\begin{aligned} Z[\vec{z}(t+1)] &= \mathbf{A}\, Z[\vec{z}(t)] + \mathbf{B}\, Z[x_\delta(t)] \\ s^{-1} Z[\vec{z}(t)] &= \mathbf{A}\, Z[\vec{z}(t)] + \mathbf{B} \\ (\mathbf{I} - s\mathbf{A})\, Z[\vec{z}(t)] &= s\, \mathbf{B} \\ Z[\vec{z}(t)] &= s\, (\mathbf{I} - s\mathbf{A})^{-1} \mathbf{B} \end{aligned} \tag{15}$$

In the first step, we have used (11) and (12). From (5) we then get

$$\begin{aligned} Z[y(t)] &= \mathbf{C}\, Z[\vec{z}(t)] \\ &= s\, \mathbf{C}\, (\mathbf{I} - s\mathbf{A})^{-1} \mathbf{B} \end{aligned} \tag{16}$$

This result could have been directly obtained, in principle, by transforming (9). It looks a bit complicated, but because $\mathbf{B}$ and $\mathbf{C}$ both contain just one non-zero element (see (6) and (7)), we in fact only need the upper right element of the inverse matrix in the middle. A general expression for the inverse of a matrix $\mathbf{M}$ with elements $M_{i,j}$ is $(\mathbf{M}^{-1})_{i,j} = |\mathbf{M}^{\dagger}_{j,i}|/|\mathbf{M}|$ where $|\ldots|$ is the determinant and $|\mathbf{M}^{\dagger}_{i,j}|$ is the adjoint $(\mathrm{n}-1)\times(\mathrm{n}-1)$ matrix. It is obtained by crossing out the i-th column and the j-th row in $\mathbf{M}$, multiplied by an extra sign factor $(-1)^{i+j}$. The matrix $\mathbf{M} := \mathbf{I} - s\mathbf{A}$ in the middle looks explicitly like this:

$$\mathbf{M} = \begin{pmatrix} 1 & -s & 0 & \cdots & 0 \\ 0 & 1 & -s & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & -s \\ -a_1 s & -a_2 s & -a_3 s & \cdots & 1 - a_{\mathrm{n}} s \end{pmatrix}. \tag{17}$$

For the calculation of the required $(\mathbf{M}^{-1})_{\mathrm{n},1}$, we need $|\mathbf{M}^{\dagger}_{1,\mathrm{n}}|$ which we obtain by crossing out the leftmost column and the lowermost row in $\mathbf{M}$. The remaining matrix has $(\mathrm{n}-1)$ times $-s$ on the diagonal and only zeros in the upper right part, so that its determinant is $(-s)^{\mathrm{n}-1}$. Together with the $(-1)^{\mathrm{n}+1}$ sign factor, this gives just $s^{\mathrm{n}-1}$. One method to calculate the determinant of $\mathbf{M}$ is as follows: go through all elements $M_{i,j}$ in a fixed row j and sum up $(-1)^{i+j} \cdot M_{i,j} \cdot |\mathbf{M}^{\dagger}_{i,j}|$. Applying this to the bottom row, we get a sign factor $(-1)^{\mathrm{n}+i}$ multiplied by $-a_i s$ and the determinant of the matrix found by crossing out the lowermost row and the i-th column, which turns out to be $(-s)^{\mathrm{n}-1}$. Each of these terms gives a contribution of $-a_i s^{\mathrm{n}-i+1}$, except from the last column, where we have just $(1 - a_{\mathrm{n}} s)$. All together, we have

$$\begin{aligned} Z[y(t)] &= \frac{s^{\mathrm{n}}}{1 - a_{\mathrm{n}} s - a_{\mathrm{n}-1} s^2 - \ldots - a_1 s^{\mathrm{n}}} \\ &= \frac{s^{\mathrm{n}}}{g(s)}. \end{aligned} \tag{18}$$

g(s) is the *characteristic function* of the shift register. It only depends on the position of the taps.

Requiring a Period in $y(t)$. As we start with zeros in the shift register at time $t = 0$, we will see zeros at the output during $\mathrm{n}-1$ clock cycles. At $t = \mathrm{n}$, the injected one will appear at the output. We therefore have $r_{\mathrm{n}} = 1$ and $r_i = 0$ for $i = 0 \ldots \mathrm{n}-1$ in (14). Note that any periodic sequence *must* start exactly at t_{n}: it is not possible that there is a further 'preamble' after $t = \mathrm{n}$ because every state has a well defined predecessor state (which we can calculate) for this type of register! We want the following sequence to have a period P and denote the output signals after r_{n} by α_i, i.e. $r_{\mathrm{n}+i} = \alpha_i$. We can now use (14) with $\tau = \mathrm{n}$ to describe $y(t)$:

$$Z[y(t)] = \frac{s^{\mathrm{n}}}{1 - s^P} \sum_{i=0}^{P-1} \alpha_i s^i = s^{\mathrm{n}} \frac{h(s)}{1 - s^P}. \tag{19}$$

Comparing (18) and (19), which both describe the same output sequence, gives us the

condition which must be fulfilled for a period P:

$$h(s)\,g(s) = s^P + 1. \tag{20}$$

We have replaced '$-$' by '$+$' on the right hand side, which we can do on $GF(2)$ where '$- = +$'. We will later do the same in the polynomials $h(s)$ and $g(s)$.

5 Conclusions

From (20) We conclude the following:

1. The shift register can only have a period P if $g(s) = 1 + a_{\mathrm{n}}s + a_{\mathrm{n}-1}s^2 + \ldots + a_1 s^{\mathrm{n}}$ divides $1 + s^P$.

2. A register of period P_1, has, of course, also periods $k \cdot P_1$ for all possible k. To make sure that the period P is the smallest period, $g(s)$ may not divide any $1 + s^p$ with $p < P$. When we want to check that, we do not have to verify all possible p, just the prime terms of P.

3. When $g(s)$ is *not* prime over the Galois Field, it can be written as $g(s) = \alpha(s) \cdot \beta(s)$.

6 Examples for $\mathrm{n} = 4$

Maximum Length. When using only the second last tap, i.e. with $g_a(s) = s^4 + s^3 + 1$, the sequence has the maximum period $P = 2^{\mathrm{n}} - 1 = 15$:

1. $\frac{s^{15}+1}{s^4+s^3+1} = s^{11} + s^{10} + s^9 + s^8 + s^6 + s^4 + s^3 + 1.$

2. $\frac{s^5+1}{s^4+s^3+1} = s + 1 + \frac{s^3+s}{s^4+s^3+1}$,
   $\frac{s^3+1}{s^4+s^3+1} < 1$

3. ???

Shorter Period, $g(x)$ not prime. We now use only the third last tap, i.e $g_b(s) = s^4 + s^2 + 1$. We want to find out which period this gives, i.e. we must find $h(s)$ and P which fulfill

$$h(s)\,(s^4 + s^2 + 1) = s^P + 1.$$

$h(s)$ must contain a 1 term, i.e. $h(s) = h'(s) + 1$, so that we get

$$\begin{aligned}
(h'(s) + 1)\,(s^4 + s^2 + 1) &= s^P + 1 \\
h'(s)\,(s^4 + s^2 + 1) &= s^P + 1 + s^4 + s^2 + 1 \\
h'(s)\,(s^4 + s^2 + 1) &= s^P + s^4 + s^2.
\end{aligned}$$

Now $h'(s)$ must obviously contain a s^2 term, i.e. $h'(s) = h''(s) + s^2$:

$$\begin{aligned}(h''(s) + s^2)\,(s^4 + s^2 + 1) &= s^P + s^4 + s^2 \\ h''(s)\,(s^4 + s^2 + 1) &= s^P + s^6.\end{aligned}$$

Obviously, we can solve this with $P = 6$, which is the period we are looking for. The shift register states after the injection of the starting '1' are:
1000
0100
1010
0101
0010
0001
1000 (period)

Shorter Period, $g(x)$ prime. The polynomial $g_c(s) = s^4 + s^3 + s^2 + s + 1$ is prime. We can prove that by trying to divide by all polynomials up to degree 2 (the square root of the degree of $g_c(s)$). There are not so many. For degree 1, we only have $p_{11} = s$, for degree 2 there are only $p_{21} = s^2$ and $p_{22} = s^2 + s + 1$ (because $s^2 + 1 = (s + 1)^2$ and $s^2 + s = (s + 1)s$).

7 Fibonacci and Galois Type Registers

So far, we have treated the type of shift registers shown in fig. 1, where the output, the taps and, possibly, an input are all XOR-ed together and fed back to the input. This topology is often called a 'Fibonacci' type shift register. In a practical application, it has the drawback that the many daisy chained XOR gates can introduce quite some delay so that the maximum clocking frequency of the register is limited. Another possible topology, the 'Galois' type shown in fig. 2 avoids this drawback because there is at most one XOR gate in front of every flipflop.

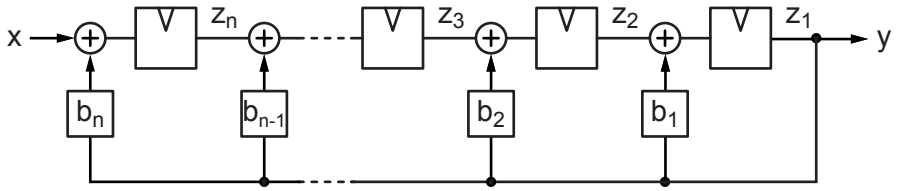

Figure 2: 'Galois' type linear shift register

Can we apply what we have learned to this type of register? Let's write down again the recursion which allows us to calculate $\vec{z}(t + 1)$ from $\vec{z}(t)$:

$$z_i(t + 1) \quad = \quad z_{i+1}(t) + b_i \cdot z_1(t) \quad \text{for} \quad i = 1 \ldots \mathrm{n} - 1$$

$$\begin{aligned} z_{\mathrm{n}}(t+1) &= x(t) + b_{\mathrm{n}} \cdot z_1(t) \\ y(t) &= z_1(t). \end{aligned}$$

In Matrix notation, this is

$$\vec{z}(t+1) = \begin{pmatrix} b_1 & 1 & 0 & \cdots & 0 \\ b_2 & 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ b_{\mathrm{n}-1} & 0 & 0 & \cdots & 1 \\ b_{\mathrm{n}} & 0 & 0 & \cdots & 0 \end{pmatrix} \vec{z}(t) + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} x(t)$$

This looks very similar to (6), just that now, the coefficients with small indices are close to the diagonal. If we do the same as before with this new matrix, it turn out that the characteristic polynomial is the same as before, if we use b_1 for a_{n} and so on! The period of the Galois Register is therefore the same as for the Fibonacci register, if we *reverse the tap order*. check!

8 More stuff to come...

- Shift register type 2 with self start
- sequences of non maximum length
- properties of the maximum sequence (autocorrelation function...)

References

[1] M. Gössel, Angewandte Automatentheorie I und II, Wissenschaftliche Taschenbücher Band 116 und 117, Akademie Verlag, Berlin, 1972, ISBN 3528061170

[2] W. S. Golomb, Shift Register Sequences, Aegean Park Press, 1982, ISBN 0894120484

[3] Efficient Shift Registers, LFSR Counters and Long Pseudo-Random Sequence Generators, Xilinx Application Note XAPP 052, July 7,1996 (Version 1.1), http://www.xilinx.com/bvdocs/appnotes/xapp052.pdf