Cryptography Part V.c: Multiple Input Signature/Shift Registers (MISRs)

Cryptographic Hardware for Embedded Systems

ECE 3894

Fall 2019
Assoc. Prof. Vincent John Mooney III
Georgia Institute of Technology

©Georgia Institute of Technology, 2018-2019
Reading

• Please read Chapter 10 from *Digital Systems Testing and Testable Design* by Abramovici, Breuer and Friedman, 1990.
Linear Feedback Shift Register (Fig. 10.10,11)
Figure 10.10 Type 1 (external-XOR) LFSR

Figure 10.11 Type 2 (internal-XOR) LFSR
Notation

\[ Q_i = \begin{array}{cc} D & Q \\ Clk & \overline{Q} \end{array} \]
Linear Feedback Shift Register Example

![Diagram of a Linear Feedback Shift Register Example]
Initial State

• The input to the first FF is a bit value which is a function
• Each FF output defines a bit

• How many distinct values can four bits exhibit?
A Degenerate Case

• Suppose the initial state is all zeros, what happens each clock cycle?
A Question

• We must omit the initial state (IS) of all zeros from this design.
• We also must omit any state of all zeros – otherwise the state will never change again!
• So for this design we cannot choose $c_1$, $c_2$ and $c_3$ such that all $2^n = 2^4 = 16$ states are used; can we achieve $2^n - 1 = 15$?
Some Comments on the Notation

- The first bit input to the first FF is $a_0$ at time zero.
- The output of the first FF is the input to the first FF at the previous time step; if the current clock step is zero, let’s call the previous time step -1; hence the output of the first FF at time zero is $a_{-1}$.
Current State

• Clock cycle $m$

• For $n$ FFs, the sequence is $a_m, a_{m-1}, a_{m-2}, \ldots, a_{m-n}$
Addition Modulus 2

- Note that binary addition modulus 2 becomes XOR

\[ a_m = c_1 a_{m-1} + c_2 a_{m-2} + c_3 a_{m-3} = c_1 a_{m-1} \oplus c_2 a_{m-2} \oplus c_3 a_{m-3} \]

- Similarly, binary multiplication becomes AND
Galois Field

• Finite field with $p^n$ elements where $p$ is a prime number and $n$ is a positive integer
• Usual operations on integers, then mod $p$
• $GF(2)$

$$\text{Mod-2} \ (x^n + x^n = x^n - x^n = 0)$$
Binary Operations

Mod-2 \((x^n + x^n = x^n - x^n = 0)\)

**Addition/Subtraction**

\[
\begin{align*}
(x^5 + x^2 + 1) + (x^4 + x^2) &= x^5 + x^4 + 1 \\
x^5 &+ x^4 + x^2 + 1 \\
\hline
x^5 &+ x^4 + 1 \\
= x^5 + x^4 + 1
\end{align*}
\]

**Multiplication**

\[
\begin{align*}
(x^2 + x + 1) \times (x^2 + 1) &= x^2 + x + 1 \\
(x^2 + 1)^2 &+ x^2 \\
\hline
x^2 + x + 1 &+ x^2 \\
\hline
x^2 + x + 1
\end{align*}
\]

**Division**

\[
\begin{align*}
\frac{x^2 + x + 1}{x^2 + 1} &= \frac{x^2 + x + 1}{x^3 + x^2 + x + 1} \\
x^2 + 1 &\div \frac{x^4 + x^2}{x^3 + x + 1} \\
\hline
\frac{x^4 + x^2 + x}{x^2 + 1} &\div \frac{x^3 + x}{x^2 + 1} \\
\hline
0
\end{align*}
\]
Generating Function

• With no external input, we have an *autonomous LFSR*
• We can associate each $a_i$ with a distinct coefficient in a polynomial
• We use variable $x$ in the polynomial, but $x$ is never assigned any value
• In a way, $x$ keeps track of time where the power indicates the clock cycle (except for zero which indicates the input to the first FF)
• $G(x)$ is an infinite sequence
• $G(x) = a_0 + a_1x^1 + a_2x^2 + ... + a_mx^m + ... = \sum_{m=0}^{\infty} a_m x^m$
• Note that there is an initial sequence where $a_0$ works its way from the input to the output; this initial sequence is $n$ clock cycles long
Example 1

\[ c_1 \]
\[ c_2 \]
\[ c_3 = 1 \]

\[ Q_1 \]
\[ Q_2 \]
\[ Q_3 \]
Example 2

\[ Q_1 \rightarrow c_1 \rightarrow Q_2 \rightarrow c_2 \rightarrow Q_3 \]

\[ c_3 = 1 \]
Example 3
Some Mathematical Results

• \( a_m = \sum_{i=1}^{n} c_i a_{m-i} \)
  - E.g., if \( n = 3 \) then \( a_m = c_1 a_{m-1} + c_2 a_{m-2} + c_3 a_{m-3} \)

• \( G(x) = \sum_{m=0}^{\infty} a_m x^m = \sum_{i=1}^{n} c_i x^i G(x) + \sum_{i=1}^{n} c_i x^i (a_{-i} x^{-i} + \ldots + a_{-1} x^{-1}) \)

\[ \Rightarrow G(x) = \frac{\sum_{i=1}^{n} c_i x^i (a_{-i} x^{-i} + \ldots + a_{-1} x^{-1})}{1 + \sum_{i=1}^{n} c_i x^i} \], which for \( a_{-i} = 0 \) except \( a_{-n} = 1 \)

\[ \Rightarrow G(x) = \frac{c_n x^n (a_{-n} x^{-n})}{1 + \sum_{i=1}^{n} c_i x^i} = \frac{1}{1 + \sum_{i=1}^{n} c_i x^i} \] since \( c_n = 1 \) always for an n-bit LFSR

• The denominator is referred to as the characteristic polynomial of the sequence:

\[ P(x) = 1 + c_1 x + c_2 x^2 + \ldots + c_n x^n \]
Some Mathematical Results (Continued)

• recall $G(x) = \sum_{m=0}^{\infty} a_m x^m = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m + \ldots$

• and if we choose initial state (IS) $a_{-1} = 0, a_{-2} = 0, \ldots, a_{-(n-1)} = 0, a_{-n} = 1$

$\Rightarrow G(x) = \frac{1}{\sum_{i=1}^{n} c_i x^i} = \frac{1}{1+c_1 x+c_2 x^2+\ldots+c_n x^n} = \frac{1}{P(x)}$ and note $G(x)$ periodic (say, $p$)

$\Rightarrow \frac{1}{P(x)} = (a_0 + a_1 x + a_2 x^2 + \ldots + a_{p-1} x^{p-1} + x^p(a_0 + a_1 x + a_2 x^2 + \ldots + a_{p-1} x^{p-1}) + x^{2p}(a_0 + a_1 x + a_2 x^2 + \ldots + a_{p-1} x^{p-1}) + \ldots$

$\Rightarrow \frac{1}{P(x)} = (a_0 + a_1 x + a_2 x^2 + \ldots + a_{p-1} x^{p-1})(1 + x^p + x^{2p} + \ldots) = \frac{a_0 + a_1 x + a_2 x^2 + \ldots + a_{p-1} x^{p-1}}{1-x^p}$
Comments

• There can also be derived what is called the *reciprocal polynomial* $P^*(x) = x^nP(1/x) = c_n + c_{n-1}x + c_{n-2}x^2 + ... + c_1x^{n-1} + x^n$

• As a result, this LFSR can be described by two types of polynomials
Figure 10.12  Reciprocal characteristic polynomials
Example 1
Example 2

\[ c_1 \]

\[ Q_1 \]

\[ c_2 \]

\[ Q_2 \]

\[ c_3 = 1 \]

\[ Q_3 \]
Example 3

\[ c_3 = 1 \]
Periodicity

• Maximum length of period $p$ for an LFSR with $n$ FFs is $2^n - 1$

• Theorem 10.6: Given an LFSR with initial state $a_{-1} = 0$, $a_{-2} = 0$, ..., $a_{-(n-1)} = 0$, $a_{-n} = 1$, the LFSR sequence $\{a_m\}$ is periodic with the smallest integer $k$ for which $P(x)$ divides $(1-x^k)$

• Note that “divides” in the theorem means there is no remainder

• recall $\frac{1}{P(x)} = \frac{a_0 + a_1x + \cdots + a_{p-1}x^{p-1}}{1-x^p}$, this implies $\frac{1-x^p}{P(x)} = a_0 + a_1x + \cdots + a_{p-1}x^{p-1}$

• Defn.: If the sequence generated by an LFSR with $n$ FFs has period $2^n - 1$, then it is called a maximum-length sequence

• Defn.: the characteristic polynomial associated with a maximum-length sequence is called a primitive polynomial
More Comments

• We will not cover how to generate primitive polynomials
• However, the sequences they generate have the following properties:
  • Property 1: given a sequence of m bits, the number of ones differs from the number of zeros by at most one
  • Property 2: given a sequence of m bits, the number of runs of ones equals the number of runs of zeros
  • Property 3: given a sequence of m bits, one half of the runs have length one, one fourth have length two, one eighth have length three, and so forth, as long as the fractions result in integral numbers of runs
\[ P(x) = 1 + \sum_{i=1}^{n} c_i x^i \]  
Initial state: \( I(x) = 0 \); Final state: \( R(x) \)

\[ P^*(x) = 1 + \sum_{i=1}^{n} c_i x^{n-i} \]

\[ \frac{G(x)}{P^*(x)} = Q(x) + \frac{R(x)}{P^*(x)} \]

or

\[ G(x) = Q(x) P^*(x) + R(x) \]

Figure 10.15  A type 2 LFSR used as a signature analyzer
Signature Analyzers

• LFSRs can be used to compress test bit data (stuck-at fault testing)
• Consider an input of \( m \) bits: there are \( 2^m \) possible inputs from the testing
• However, the \( n \)-bit LFSR can produce a periodic output with \( 2^n \) possible outputs (note the all-zero state is now allowed due to the presence of input)
• It has been shown that the number of bitstreams of length \( m \) that produce the same output of length \( n \) is \( 2^m/2^n \)
• Thus \( 2^m/2^n - 1 = 2^{m-n} - 1 \) erroneous bitstreams exist that produce the same signature (i.e., there is a test failure but the \( n \)-bit output is identical to a passing value)
• Since there are \( 2^m - 1 \) erroneous bitstreams possible, the proportion of erroneous bitstreams exist that produce the same signature is \( (2^{m-n} - 1)/(2^m - 1) \approx 2^{-n} \)
• For \( n = 16 \), \( 100(1 - 2^{-n}) = 99.9984 \) percent
Characteristic Polynomial

\[ X \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow Z \]

\[ P^*(x) = 1 + x^2 + x^4 + x^5 \]

Input sequence: 1 1 1 0 1 0 1 (8 bits)

\[ G(x) = x^7 + x^6 + x^5 + x^4 + x^2 + 1 \]

\[ \sum_{i=1}^{n} c_i x^i a_m(t) \]
Input sequence: $11110101$ (8 bits)
$G(x) = x^7 + x^6 + x^5 + x^4 + x^2 + 1$

\[ P^*(x) = 1 + x^2 + x^4 + x^5 \]

<table>
<thead>
<tr>
<th>Time</th>
<th>Input stream</th>
<th>Register contents</th>
<th>Output stream</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1 0 1 0 1 1 1 1</td>
<td>0 0 0 0 0 0</td>
<td>Initial state</td>
</tr>
<tr>
<td>1</td>
<td>1 0 1 0 1 1 1</td>
<td>1 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>5</td>
<td>1 0 1 0 1 1 1</td>
<td>1 1 1 1 1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1 0 1 0 1 1 1</td>
<td>0 0 0 1 0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1 0 1 0 1 1 1</td>
<td>0 0 0 1 0</td>
<td>0 1</td>
</tr>
<tr>
<td>8</td>
<td>Remainder</td>
<td>0 0 1 0 1</td>
<td>1 0 1</td>
</tr>
</tbody>
</table>

\[ R(x) = x^2 + x^4 \]
\[ Q(x) = 1 + x^2 \]
\[ P^*(x): x^5 + x^4 + x^2 + 1 \]
\[ \times Q(x): x^2 + 1 \]
\[ \frac{x^7 + x^6 + x^4 + x^2 + x^5 + x^4 + x^2 + 1}{x^7 + x^6 + x^5 + 1} \]
\[ = x^7 + x^6 + x^5 + 1 \]

Input sequence: 1 1 1 0 1 0 1 (8 bits)
\[ G(x) = x^7 + x^6 + x^5 + x^4 + x^2 + 1 \]
Figure 10.17  Multiple-input signature register
Linear Feedback Shift Register Example

\[ Q_i \quad Q_i \quad Q_i \]