Note that page numbering is turned on in this template.

Chapter Contents

1.1. Review of the Z Transform

1.2. A Polynomial Divider
Figure 1: a polynomial divider circuit,Symbols, Analysis of Figure 1
Registers: hold the next k symbols of O(z)
Example: The Scrambler

1.3 Interpretation of Polynomial Division
Finite Division, The Hidden Remainder
Impulse Response

1.4 The standard form of the polynomial divider.
Figure 2: The standard polynomial divider circuit, Analysis of Figure 2
Registers: hold the current dividend

1.5 Comparison to Long Division of Polynomials
Figure 3: long division
Figure 4 : register contents of Figure 2 after each clock.
Figure 5 : vectorized action of the divider of Figure 2.
Application of Figure 2 to CRC :
Figure 6 : explanation of Figure 1

1.6. A Polynomial Multiplier
Figure 7: A polynomial multiplier circuit, Analysis of Figure 7
Interpretation of Polynomial Multiplication
Impulse Response
Registers: hold the most recent k input symbols of I(z).
Example: The Descrambler

1.7 The standard form of the polynomial multiplier.
Figure 8: The standard polynomial multiplier circuit, Analysis of Figure 8
Registers: hold the current partial product
Figure 9: Comparison of Figure 7 with Long Multiplication
Figure 10: Comparison of Figure 8 with Long Multiplication
Figure 11: Vectorized view of Figure 8 Long Multiplication

1.8 Polynomial Processors in the Time Domain
The Multiplier of Figure 7
The Divider of Figure 1
Convolution Theorem Approach, Summary box

1.9 Simultaneous Polynomial Multiply and Divide

Figure 12: A simultaneous polynomial multiplier and divider.

Analysis of Figure 12

Application: A CRC checksum calculator.

Appendix 1.1. Processing polynomials vs. processing integers.

****

Chapter Contents (partial)

2.1. The Maximum Period of a Shift Register Generator

Definitions: state vector period, output period

Fact 2: In a shift register generator with k registers each of which can store Q symbols, neither the state vector period nor the output period can exceed $Q^k - 1$.

2.2 The State Vector Sequence of a Shift Register Generator

The Figure 2 Iteration Equation

A note on Symbols

Galois Field Review

Fact 3: Any monic h(x) of degree k that is irreducible over GF(p) is the minimum polynomial of the element {x} of GF(q=p^k).

The Figure 2 Galois Iterator

Fact 5: The state vector of a (Figure 2) shift register generator steps periodically through some subset N of the non-zero elements of GF(p^k), provided it is initialized in some non-zero state. The state vector period N must integrally divide $p^k - 1$, so $N \leq p^k - 1$. The integer N is the order of {x}.

Fact 6: If $p^k - 1$ happens to be a prime number, the state vector period is $p^k - 1$.

Fact 8: If h(x) is a *primitive polynomial* of GF(q=p^k), then the state vector period is is p^k-1.

An example : GF(q=2^9)

The non-terminating Remainder in polynomial division.

Fact 10: The *state vector period* of a Figure 2 shift register generator is the same as the *period* of h(x) .

Fact 11: The period of any particular register of the Figure 2 circuit must integrally divide the state vector period N.

Corollary: The *output period* of a Figure 2 shift register generator must be a divisor of the state vector period N.

2.3 The Output Sequence of a Shift Register Generator

Fact 1: There are exactly p^k solutions to the difference equation (2.3.2).

Fact 5: All p^k solutions of the difference equation (2.3.2) are of the form $\{o_i\}$ = {c,c,c,c ...}.

An example : GF(16)

Fact 8: If h(x) of degree k is a *primitive polynomial* of GF(p^k), then *all* p^k- 1 non-zero solutions of the difference equation (2.3.2) have period n = p^k - 1.
Fact 9: If h(x) of degree k is a *primitive polynomial* of GF(p^k), then there is essentially only *one* characteristic non-zero output pattern of the shift register generator, its period is p^k - 1. **Fact 10**: For a Figure 1 shift register generator, the *state vector period* is the same as the *output period*.

(contents continues on next page)

2.4. Characteristic Sequences (general p)

Fact 1: Each possible state vector occurs exactly once somewhere in a characteristic sequence.

Fact 4: In a characteristic sequence , every non-zero symbol appears exactly (p^{k-1}) times, and the 0 symbol appears exactly (p^{k-1} - 1) times.

Fact 5: If we line up our characteristic sequence with any delayed version of itself and compare over one period, we will find that the two sequences differ in exactly $(p-1)p^{k-1}$ places.

Binary Characteristic Sequences (p=2).

Facts 1-5: Restatements of previous facts for p = 2.

Fact 6: A string of (k-1) 1's appears exactly once in a binary characteristic sequence.

The product of two shifted binary characteristic sequences.

Fact 7: Exactly 2^{k-2} ones appear in (one period's worth of) the product of two shifted binary characteristic sequences. If the sequences are unshifted, or are shifted by a multiple of the period, then exactly 2^{k-1} ones appear in the product.

Probability of strings of 1's and 0's.

Fact 8: In a characteristic sequence of reasonably large k, the distribution of string lengths is very close to the theoretical white sequence distribution, up to a length where strings are no longer allowed to exist. Up to this length, the relative probability of a string of exactly n identical symbols is given by P(n) ♠ $1/2^n$. For k = 9, the error in this estimate is < 0.5%.

2.5 Autocorrelation, Probability, and Statistics for a sequence in GF(2)

A few Basics of Probability and Statistics Applied to GF(2)

Fact 1: The first and second order statistics are completely determined by the autocorrelation function.

Fact 2: For a statistically independent sequence, all higher order statistics are determined by the first order statistics

Definition: A *white* sequence over GF(2) is one which is statistically independent and has $p(a_n) = 1/2$.

Statistics for Modified GF(2): Coefficients are (+1, -1).

Plots of autocorrelation functions for white sequences.

2.6 The Spectral Power Density of a Shift Register Generator

Case 1 Analysis: Singled Ended Output

Fact 1: Comparison of Figure 2.6.3 with Figure 2.5.1 shows that the characteristic sequence has an autocorrelation function that differs only slightly from that of a white sequence. For k = 9, P = 511 so (1 + 1/P) ♠ 1.002. Thus, the first and second order statistics of a characteristic sequence are very close to those of a a white sequence.

Case 2 Analysis: Differential Output

Fact 2: Apart from a possible DC line, the spectrum of the characteristic sequence output by a shift register generator is entirely continuous, and it is the envelope of the pulse shape used.

Appendix 2.1: Proof of Fact 4 of Section 2.3

Appendix 2.2: A List of Primitive Polynomials over GF(2)

****

Chapter Contents

3.1 Review of earlier Chapters

3.2 Matrix Solution of Figure 2.

The A matrix.

3.3 Matrix Solution of Figure 1.

The B matrix.

3.4 Shift Register Generators Revisited

Fact 1: The entire state vector of a primitive polynomial shift register generator satisfies the difference equation (2.3.2). Thus, each component of the state vector satisfies (2.3.2) as well, and each component is the state of a particular register.

Fact 2: The state of each individual register in a primitive polynomial shift register generator cycles through the characteristic sequence.

3.5 The Output of a Scrambler

Fact 2: In the presence of an input stream $\{i_n\}$, and with initial state vector $\mathbf{q}_0 = \mathbf{0}$, the state of any register of a primitive polynomial scrambler equals the "dot product" of a characteristic sequence multiplied by the (time reversed) input sequence.

A Lemma:

Review of Leeper 1973.

3.6 The Spectral Density of a Scrambler Output

" Barring anomalous input sequences, the spectral density of the output of an operating scrambler is essentially the same as the spectral density of a random white NRZ signal."

Interpretation of the Spectrum;

Power Distribution by Hump

3.7 The NRZI Mini-Scrambler.

Fact 1: The output of the above mini-scrambler changes state each time a one is encountered in the input data stream, and it holds its state each time a 0 is encountered in the input stream.

Definition: An *NRZI line code* is what emerges from the mini-scrambler when you input a NRZ linecode.

Scrambler Summary!

3.8. Matrix analysis of the polynomial multiplier (de-scrambler) of Figure 7 .

3.9 Matrix analysis of the standard polynomial multiplier of Figure 8 .

3.10. Proof that a Descrambler really descrambles the output of a Scrambler.

Figure 3.10.1: Proposed scrambler structure for serial digital communication .
Proof in the Z Domain;
Proof in the Time Domain;
Conclusions

(contents continued on next page)

3.11 The Kill Sequence Problem and Strings of Zeros

Fact 1: It is possible that a k-stage scrambler will output a string of N+k zeros if a string of N zeros exists in the input stream; the probability is 1/P where $P = 2^k - 1$.

Example 1: Suppose in D1 video.

Example 2: Ancillary data

Fact 2: Assume a scrambler is in some state **a**. Consider the k-symbol pattern that next shifts in, call this pattern **i**. After this pattern shifts in, the scrambler is in state **b**. We claim that, given any patterns **a** and **b**, there exists a unique pattern **i** which causes **a** to go to **b**.

Corollary 2: If pattern **b** is **0**, there exists a unique pattern **i** which takes **a** to **0**.

Definition: Such a pattern is called a *kill vector*.

Explicit formula for vector **i**.

Example: k=9

Discussion of the zeros problem.

Fact 3: If a k-stage scrambler has an even number of feedback taps, then if the input stream is set to all 1's, the scrambler behaves exactly as if the input stream were all 0's and there were an inverter on the final output.

Some final notes:

Appendix 3.1: Proof of Eq. (3.10.15)