System performance criteria in CDMA networks using gold codes

Yong H. Kim
New Jersey Institute of Technology

Follow this and additional works at: https://digitalcommons.njit.edu/theses
Part of the Electrical and Electronics Commons

Recommended Citation
https://digitalcommons.njit.edu/theses/1403
Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement.

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation.

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen.
The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.
ABSTRACT

Title of Thesis: System Performance Criteria in CDMA Networks Using Gold codes

Yong H. Kim, M. S. E. E., 1988

Thesis directed by: Dr. Joseph Frank

Associate Professor

Department of Electrical Engineering

First, we have presented the autocorrelation and crosscorrelation properties for periodic and aperiodic binary sequences. The generation of binary sequences using shift registers with feedback was reviewed. We have also included correlation properties for the Gold codes.

Next, we discussed Gold code generation for the balanced and unbalanced Gold codes.

Thirdly, we investigated the number of simultaneous users in a CDMA system using Gold codes for the worst case and the average case of mutual interference.

Finally, we simulated the probability of interference exceeding a threshold value, and the average crosscorrelation value caused by interference in a CDMA network which is using a Gold code. We compared probability and average crosscorrelation values simulated
with theoretical bounds calculated. Here the simulation programs are done in C computer language.
SYSTEM PERFORMANCE CRITERIA

IN

CDMA NETWORKS USING GOLD CODES

BY

YONG H. KIM

Thesis submitted to the faculty of the Graduate School of the New Jersey Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering 1988
APPROVAL SHEET

Title of Thesis: System Performance Criteria in CDMA Networks Using Gold codes

Name of Candidate: Yong H. Kim

Master of Science in Electrical Engineering, 1988

Thesis and Abstract Approved: ___________________  _____

Dr. Joseph Frank  Date

Associate Professor

Electrical Engineering

____________________  _____

Date

____________________  _____

Date
VITA

Name: Yong H. Kim

Address:

Degree and date to be conferred: M.S.E.E., 1988

Date of Birth:

Place of Birth:

Secondary Education: Taeryun High School, Taegu, Korea

Collegiate Institutions Attended:

<table>
<thead>
<tr>
<th>Institution</th>
<th>Dates</th>
<th>Degree</th>
<th>Date of Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Jersey Institute of Technology, Newark, New Jersey</td>
<td>1986 to</td>
<td>M.S.E.E.</td>
<td>May 1988</td>
</tr>
<tr>
<td>Kyung Pook National University, Taegu, Korea</td>
<td>1973 to 1978</td>
<td>B.S.E.E.</td>
<td>August 1978</td>
</tr>
</tbody>
</table>

Major: Electrical Engineering
ACKNOWLEDGEMENT

The author wishes to extend sincere appreciation and gratitude to professor Joseph Frank, whose valuable assistance and guidance has made this study a success.

I’m also grateful for the advices, support, and inspiration that my family and company, KTA, gave me during this period of study.
# Table of Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>General</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Correlation Property</td>
<td>4</td>
</tr>
<tr>
<td>2.1</td>
<td>Autocorrelation</td>
<td>5</td>
</tr>
<tr>
<td>2.1.1</td>
<td>Introduction</td>
<td>5</td>
</tr>
<tr>
<td>2.1.2</td>
<td>Periodic Autocorrelation</td>
<td>16</td>
</tr>
<tr>
<td>2.1.3</td>
<td>Aperiodic Autocorrelation</td>
<td>20</td>
</tr>
<tr>
<td>2.2</td>
<td>Crosscorrelation</td>
<td>27</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Introduction</td>
<td>27</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Periodic Crosscorrelation</td>
<td>33</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Aperiodic Crosscorrelation</td>
<td>37</td>
</tr>
<tr>
<td>2.3</td>
<td>Crosscorrelation Spectra in Gold codes</td>
<td>39</td>
</tr>
<tr>
<td>3</td>
<td>Gold Code Generation</td>
<td>44</td>
</tr>
<tr>
<td>3.1</td>
<td>Algorithm for Gold Code Selection</td>
<td>44</td>
</tr>
<tr>
<td>3.2</td>
<td>Generation of General Gold Codes</td>
<td>48</td>
</tr>
<tr>
<td>3.3</td>
<td>Generation of Balanced Gold Codes</td>
<td>54</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Relative Phase Requirement For Balanced Codes</td>
<td>56</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Initial Conditions for Balanced Gold Codes</td>
<td>59</td>
</tr>
<tr>
<td>3.4</td>
<td>Number of Gold codes</td>
<td>63</td>
</tr>
<tr>
<td>4</td>
<td>Number of Simultaneous Users in Gold Code</td>
<td>67</td>
</tr>
<tr>
<td>4.1</td>
<td>Worst Case</td>
<td>67</td>
</tr>
<tr>
<td>FIGURE</td>
<td>TITLE</td>
<td>PAGE</td>
</tr>
<tr>
<td>-------</td>
<td>---------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>2.1</td>
<td>M-sequence autocorrelation function</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Typical nonmaximal code autocorrelation function</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>Variable tap five-stage SRG</td>
<td>10</td>
</tr>
<tr>
<td>2.4</td>
<td>Autocorrelation for 21-chip maximal code[5,4]</td>
<td>15</td>
</tr>
<tr>
<td>2.5</td>
<td>Binary feedback shift register</td>
<td>19</td>
</tr>
<tr>
<td>2.6</td>
<td>Comparative autocorrelation and</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>crosscorrelation for 31-chip mirror</td>
<td></td>
</tr>
<tr>
<td></td>
<td>image m-sequences</td>
<td></td>
</tr>
<tr>
<td>2.7</td>
<td>Comparative autocorrelation and</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>crosscorrelation for 31 chip m-sequences</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(not images)</td>
<td></td>
</tr>
<tr>
<td>2.8</td>
<td>Crosscorrelation spectrum for N = 31, n = 5</td>
<td>42</td>
</tr>
<tr>
<td>3.1</td>
<td>Configuration of Gold code generator</td>
<td>49</td>
</tr>
<tr>
<td>3.2</td>
<td>Gold code generation</td>
<td>51</td>
</tr>
<tr>
<td>3.3</td>
<td>Balanced Gold code generation of length 31</td>
<td>61</td>
</tr>
<tr>
<td>3.4</td>
<td>Gold code pair generator for mode 2 return link</td>
<td>62</td>
</tr>
<tr>
<td>5.1</td>
<td>Flow diagram for the simulator in</td>
<td>78</td>
</tr>
</tbody>
</table>
CDMA networks

5.2 Network model for CDMA application  81
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Least energy sequences maximized for auto-optimality with periods from 31 to 255</td>
<td>24</td>
</tr>
<tr>
<td>2.2</td>
<td>Three-level crosscorrelation properties in Gold codes.</td>
<td>43</td>
</tr>
<tr>
<td>3.1</td>
<td>Performance of preferred pairs compared with worst case pairs.</td>
<td>45</td>
</tr>
<tr>
<td>3.2</td>
<td>Description of available tables of binary</td>
<td>47</td>
</tr>
<tr>
<td>3.3</td>
<td>Modulo-2-combined Gold code</td>
<td>51</td>
</tr>
<tr>
<td>3.4</td>
<td>Number of balanced and unbalanced codes for n odd</td>
<td>54</td>
</tr>
<tr>
<td>3.5</td>
<td>Relative phase shifts of the sequence a and b</td>
<td>58</td>
</tr>
<tr>
<td>3.6</td>
<td>Number of Gold codes</td>
<td>64</td>
</tr>
<tr>
<td>4.1</td>
<td>Number of simultaneous users(worst case)</td>
<td>67</td>
</tr>
<tr>
<td>4.2</td>
<td>Number of simultaneous users(average case)</td>
<td>70</td>
</tr>
<tr>
<td>5.1</td>
<td>The probability exceeding threshold value</td>
<td>76</td>
</tr>
<tr>
<td>5.2</td>
<td>Gold codes used in simulator</td>
<td>82</td>
</tr>
<tr>
<td>5.3</td>
<td>Simulation results(probability)</td>
<td>84</td>
</tr>
<tr>
<td>5.4</td>
<td>Simulation results(average value of crosscorrelation)</td>
<td>84</td>
</tr>
</tbody>
</table>
CHAPTER I

GENERAL

One of the applications of spread spectrum systems is to provide a means other than frequency division multiple access (FDMA) or time division multiple access (TDMA) of sharing the scarce channel resources. When channel resources are shared using spread spectrum techniques, a method known as code division multiple access (CDMA), all users are permitted to transmit simultaneously using the same band of frequencies. Therefore, code division multiple accessing does not require the time synchronization needed in TDMA nor the many filters required in FDMA. Users are each assigned a different spreading code so that they can be separated in the receiver despreading process. A goal of the spread spectrum designer for a multiple access system is to find a set of spreading codes or waveforms such that as many users as possible can use a band of frequencies with as little mutual interference as possible.

Here, the receiver despreading operation is a correlation operation with the spreading code of the desired transmitter. Ideally, a received signal that has been spread using a different spreading code will not be despread and will cause minimal interference in the
desired signal. The specific amount of interference from a user employing a different spreading code is related to the crosscorrelation between the two spreading codes and the power level of the two signals.

Unfortunately the ideal spreading code would be an infinite sequence of equally likely random binary digits. Hence, the use of an infinite random sequence implies infinite storage in both the transmitter and receiver. This is clearly not possible, so that the periodic pseudorandom codes (PN codes) are always employed.

Gold codes are specific PN codes and allow construction of families of $2^n - 1$ codes from pairs of $n$-stage shift registers in which all codes have well defined crosscorrelation characteristics. In other words, Gold codes are combinations of maximal-length(m) codes, which are by far the most widely used in multiple access systems.

The Gold codes introduced in this thesis were invented in 1964 at the Magnavox Corporation specifically for multiple-access applications of spread spectrum. The first spread spectrum modem to employ Gold codes was MX-170 and the first developmental models were demonstrated in 1964[13]. Relatively large sets of Gold codes exists which have well controlled crosscorrelation properties.
Spread spectrum communication techniques by using Gold codes have been recognized as a viable method to gain an advantage in interference environments. Many new military-oriented systems have been initiated and some civil systems have been attempted.
CHAPTER II
CORRELATION PROPERTY

This is the most fundamental theory when we think about Gold code. As we discussed before, CDMA that allows for the simultaneous operation of many signals at the same carrier frequency is a multiplexing technique in which many carriers, all at essentially the same frequency, send different PN type codes that have the property that the crosscorrelation is low between any pair of the codes. That is, the transmitted signals are of the form

\[ S_i(t) = \sqrt{2p} d_i(t)G_i(t)\sin(\omega_0 t + \Theta) \]

\[ i = 1, \ldots, N \]

where

- \( p \) is transmitted power
- \( d_i(t) \) is the \( i \)th data signal (±1)
- \( G_i(t) \) is the \( i \)th Gold code
- \( \omega_0 \) is the carrier radian frequency
- \( \Theta \) is the carrier phase

Further, it is required that

\[ \left| \int_{t}^{T} G_i(t)G_j(t-KT)dt \right| \ll \int_{t}^{T} G_j^2(t)dt \]

\[ i \neq j \text{ for arbitrary } k \]
so that the ith channel will not interfere with the jth channel for any time shift between them. This interference problem extends to acquisition, tracking, and data demodulation. The correlation properties of the code sequences used in spread spectrum communications depend on code type, length, chip rate, and even the chip-by-chip structure of the particular code being used. Both autocorrelation and crosscorrelation are of interest in communication system design.

2.1 Autocorrelation

2.1.1 Introduction

Autocorrelation, in general, is defined as the integral

\[ \Theta(r) = \int_{-\infty}^{\infty} g(t)g(t-r)dt, \]

which is a measure of the similarity between a signal and a phase-shifted replica of itself. An autocorrelation over all phase shift \((t-r)\) of the signal, where \(\Delta t\) is one-chip intervals.

Autocorrelation is of most interest in choosing code sequences that give the least probability of a false synchronization. In a communications system designed for maximum sensitivity it is no mean task to discriminate
between correlation peaks in a poorly chosen code. Therefore the designer should investigate the code he or she uses carefully, even if that code is one of the relatively safe m-sequences. Statements such as our extremely long 127-bit chip code sequence assures noisilike properties, which have been observed in the literature on spread spectrum systems, exhibit a lack of investigation.

Here, we introduce a term for the property of code sequences, pairs of sequences, or a sequence and an other signal that determines a receiver's ability to recognize the proper point of code synchronization. This property is called the index of discrimination (ID) and denotes the difference in correlation between the fully correlated (perfectly synchronized) code and the minor peak of autocorrelation or of crosscorrelations. A particular code will then have separate ID values for autocorrelation and crosscorrelation with noncoded signals. The higher the ID value, the better the code.

Code sequence autocorrelation is expressed as the number of agreements minus the number of disagreements when the code or codes are compared chip by chip. The following example shows autocorrelation for all shifts of a three stage shift register generator, generating a seven-chip maximal linear code:
\[ S = S_1S_2S_3, S_3 + S_2 \rightarrow S_1, L(S) \rightarrow R(S) \]

Reference sequence : 0111001

<table>
<thead>
<tr>
<th>Shift</th>
<th>Sequence</th>
<th>Agreements(A)</th>
<th>Disagreements(D)</th>
<th>A-D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0111001</td>
<td>7</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>1011100</td>
<td>3</td>
<td>4</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>0101110</td>
<td>3</td>
<td>4</td>
<td>-1</td>
</tr>
<tr>
<td>3</td>
<td>0010111</td>
<td>3</td>
<td>4</td>
<td>-1</td>
</tr>
<tr>
<td>4</td>
<td>1001011</td>
<td>3</td>
<td>4</td>
<td>-1</td>
</tr>
<tr>
<td>5</td>
<td>1100101</td>
<td>3</td>
<td>4</td>
<td>-1</td>
</tr>
<tr>
<td>6</td>
<td>1110010</td>
<td>3</td>
<td>4</td>
<td>-1</td>
</tr>
</tbody>
</table>

Here, the net correlation A-D is \(-1\) for all except zero-shift or synchronous condition and \(2^n - 1 = 7\) for the zero-shift condition. This is typical of all m-sequences. That is, the autocorrelation spectrum for an m-sequences is two valued

- \(N\) occurs 1 time
- \(-1\) occurs \(N-1\) times

In the region between the zero and plus or minus one chip shifts, correlation increases linearly so that the autocorrelation function for an m-sequence is triangular as shown in Figure 2.1.
This characteristic autocorrelation is used to great advantage in communication and ranging systems. Two communicators may operate simultaneously, for instance, if their codes are phase-shifted more than one chip. In a ranging system a range measurement is ensured of being accurate within one chip by using the correlation peak as the marker for measurement. This may be accomplished by setting the correlation detector in such a way that it recognizes the level associated with $\pm 1$-chip synchronization and does not recognize the lower level.

When codes other than m-sequences are used, autocorrelation properties may be markedly different from those of the m-sequences. Figure 2.2 illustrates a typical autocorrelation function for a nonmaximal code. The minor
correlation peaks are dependent on the actual code used and are used by partial correlations of the code with a phase-shifted replica of itself. When such minor correlation occur, a receiving system's ability to synchronize may be impaired because it must discriminate between the major(±1 chip) and minor correlation peaks, and the margin of discrimination is reduced.

Fig. 2.2 Typical nonmaximal code autocorrelation function

For purposes of illustration, let us consider the five-stage shift register generator (SRG) shown in Figure 2.3.

If feedback is taken from stages five and three, the code sequence output is:
The autocorrelation of this sequence is shown in Figure 2.1; its maximum value is $2^n - 1 = 31$ and its $\text{ID}_{\text{auto}} = 32$. This ID value is, as expected, typical of all linear maximal sequences (of which this is an example) for which $\text{ID}_{\text{auto}}$ is always equal to $2^n$.

Now if we modify the feedback to come from stages five and four, one possible output sequence is only 21 bits long:

$$\ldots 11110000100011001010\ldots (21 \text{ chip})$$
This is an example of a nonmaximal linear sequence that is less than $2^n - 1$ chip long. There are two other nonmaximal linear sequences available from this same feedback configuration whose lengths are seven and three chip:

...1001110... and ...101...

The initial start vector contained in the register determines which of the sequences is generated. For this region greater care is necessary when nonmaximal sequences are used, both to ensure that the initial start vector is correct (or at least is one of the allowable states) and that noise does not cause the register to go to a state outside the desired set. (In such a case the output code could suddenly change from one sequence to another.)

The three sets (not counting the images) of sequence generator states for the available nonmaximal sequences are
<table>
<thead>
<tr>
<th>Set 1</th>
<th>Set 2</th>
<th>Set 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q₁Q₂Q₃Q₄Q₅</td>
<td>Q₁Q₂Q₃Q₄Q₅</td>
<td>Q₁Q₂Q₃Q₄Q₅</td>
</tr>
<tr>
<td><strong>Initial condition</strong></td>
<td>1 1 1 1 1</td>
<td>1 1 0 0 1</td>
</tr>
<tr>
<td><strong>Next states</strong></td>
<td>0 1 1 1 1</td>
<td>1 1 1 0 0</td>
</tr>
<tr>
<td></td>
<td>0 0 1 1 1</td>
<td>0 1 1 1 0</td>
</tr>
<tr>
<td></td>
<td>0 0 0 1 1</td>
<td>1 0 1 1 1</td>
</tr>
<tr>
<td></td>
<td>0 0 0 0 1</td>
<td>0 1 0 1 1</td>
</tr>
<tr>
<td></td>
<td>1 0 0 0 0</td>
<td>0 0 1 0 1</td>
</tr>
<tr>
<td></td>
<td>0 1 0 0 0</td>
<td>1 0 0 1 0</td>
</tr>
<tr>
<td></td>
<td>0 0 1 0 0</td>
<td>1 1 0 0 1</td>
</tr>
<tr>
<td></td>
<td>0 0 0 1 0</td>
<td>(7-chip cycle)</td>
</tr>
<tr>
<td></td>
<td>1 0 0 0 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 1 0 0 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 1 1 0 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 0 1 1 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 0 0 1 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 1 0 0 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 0 1 0 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 1 0 1 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 1 1 0 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 1 1 1 0</td>
<td></td>
</tr>
<tr>
<td><strong>Starting point</strong></td>
<td>1 1 1 1 1</td>
<td></td>
</tr>
<tr>
<td>(21-chip cycle)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Here, a total set of only $31 = 2^5 - 1$ states exists in all of these nonmaximal; the same number that exists in a single maximal sequence. It is typical of linear sequence generators that for every feedback point that produces a subset of length $(2^n - 1) - k$ there are one or more other nonmaximal feedback connections whose subsets (in combination with the original set) have a total length $k$. Nonmaximal sequences often have high minor autocorrelation peaks. For this reason, the use of nonmaximal codes or even sectors of maximal codes for communication should be approached with caution.

Code sequences available from the five-stage generator of Figure 2.3 are the following:

<table>
<thead>
<tr>
<th>Feedback</th>
<th>Sequence</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>[5,3]</td>
<td>...111110001110101000100101100...</td>
<td>31</td>
</tr>
<tr>
<td>[5,2]</td>
<td>...111110011010010001010111011000...</td>
<td>31</td>
</tr>
<tr>
<td>[5,4,3,2]</td>
<td>...1111100100110000101101010001110...</td>
<td>31</td>
</tr>
<tr>
<td>[5,3,2,1]</td>
<td>...1111101110001010110100001100100...</td>
<td>31</td>
</tr>
<tr>
<td>[5,4,3,1]</td>
<td>...1111010001001010110000111001010...</td>
<td>31</td>
</tr>
<tr>
<td>[5,4,2,1]</td>
<td>...1111101100111000111010100100100...</td>
<td>31</td>
</tr>
<tr>
<td>[5,4]</td>
<td>...111110000100011001010...</td>
<td>21</td>
</tr>
<tr>
<td>[5,1]</td>
<td>...111110101001100010000...</td>
<td>21</td>
</tr>
</tbody>
</table>
Six of these sequences are maximal \((2^5 - 1)\) in length, whereas two are nonmaximal. Observation of pairs 
\([5, 4, 3, 2], [5, 3, 2, 1]\), \([5, 4, 3, 1], [5, 4, 2, 1]\), and 
\([5, 4], [5, 1]\) will show that they are paired inverse. None of these 31-chip codes is useful as the spectrum spreading element for a practical system because of their short length, but they are listed here to provide a model of the types of sequences that can be produced.
Fig. 2.4 Autocorrelation for 21-chip maximal code[5,4].
Let us examine these sequences for their autocorrelation properties; autocorrelation of all six maximal codes is the same (i.e., equal to -1 for all except the 0+1 chip shift). The zero shift produces an autocorrelation value of 31 for all of them.

Autocorrelation of one of the 21-chip nonmaximals is shown in Figure 2.4 (for sequence [5,4]), which is also typical for the other 21-chip sequence (although backward). The ID$_{auto}$ value for the 21-chip sequences is 20, which could cause a reduction of 37.5\% in a receiver's synchronization capability below that for the 31-chip maximal code. The detailed discussion for periodic and aperiodic autocorrelation is given in Section 2.1.2 and 2.1.3 respectively.

2.1.2 Periodic autocorrelation

We noticed that m-sequences are certain binary sequences of length $N = 2^n - 1$, where $n$ is the number of shift register stages.

A possible representation of the autocorrelation function is

$$\Theta(r) = A - D = N - 2D$$  \hspace{1cm} (2.1)
where $A$ is the number of places that the code $a_0, \ldots, a_{N-1}$ with period $N$ and the cyclic shift $a_r$ agree, and $D$ is the number of places where they disagree, so that $A + D = N$.

An $m$-sequence has the property that a period of the sequence contains $2^{n-1}$ 1s and $2^{n-1} - 1$ 0s because there are $2^{n-1}$ even numbers ending in 1, and $2^{n-1} - 1$ odd numbers in the same range with binary representation ending in 0.

The autocorrelation function for $m$-sequences is defined by

$$\theta(0) = N \quad r = 0$$

$$\theta(r) = -1 \quad 1 < r < N - 1 \quad (2.2)$$

where the periodic autocorrelation function $0(r)$ is defined by

$$\theta(r) = \sum_{l=0}^{N-1} a_l a_{l+r}$$

For synchronization purposes this periodic autocorrelation function is ideal. In fact it can be shown\[15\] that this is the best possible autocorrelation function of any binary sequence of length $2^{n-1}$ in the sense of minimizing $\max \theta(r), 0 < r < N$.

The final property justifying the name pseudorandom for $m$-sequences is the particular distribution of runs of 1s and 0s. A run is defined to be a maximal string of
consecutive identical symbols. In any m-sequence, one half of the runs have length 1, one quarter have length 2, one eighth have length 3 and so on, as long as these fractions give an integral number of runs. In each case the number of runs of 0s is equal to the number of runs of 1s.

All of these properties are the same as one would expect from a coin tossing sequence. However, one way in which an m-sequence is seen to be not truly random is that the properties hold for every sequence, whereas in a coin tossing sequence there would be some variations from sequence to sequence.

The m-sequences have a concise description by polynomials. In the general case the output of an m-stage binary feedback shift register will satisfy a recurrence relation of order $n$ (see Fig. 2.5):
The coefficients $C_r = 1$ if the stage $S_r$ is added to the feedback path and $C_r = 0$ otherwise. The highest term $S_{t+n}$ must always be present, and it is also assumed that $C_0 = 1$.

The solution to equation 2.3 is closely related to the roots of the characteristic polynomial taking the form

$$f(x) = x^n + C_{n-1}x^{n-1} + \ldots + C_1x + C_0 \quad (2.4)$$

The number of such polynomials describing the feedback required to generate $m$-sequences is the same as the number of irreducible and primitive polynomials of degree $m$. This is given by Euler's phi functions:
number of m-sequences of length $2^n - 1 = \frac{\phi(2^n - 1)}{n}$ (2.5)

Here, $\phi(2^n - 1)$ is an Euler number; the number of positive integers including 1 that are relatively prime to and less than $2^n - 1$. As an example of the use of this formula let us observe a five-stage register, for which $2^n - 1$ is 31. From equation 2.5 the five-stage register has maximal linear sequences available:

$$\frac{\phi(2^n - 1)}{n} = \frac{30}{5} = 6$$

The six m-sequences of period 31 will all have the ideal periodic autocorrelation described by equation 2.2, and the periodic autocorrelation function will not depend on the initial content of the shift register. This is expressed by saying that the periodic autocorrelation function of m-sequences are insensitive to the phase of the sequence.

2.1.3 Aperiodic autocorrelation

To complete the description of autocorrelation properties of m-sequences, calculation of the aperiodic function is also needed.
The aperiodic autocorrelation depend strongly on the phase of sequences. That is to say, sidelobes and possible multipath interference will depend on the initial starting condition of the shift registers producing the spreading codes. This gives a multitude of codes for any given sequence period $N$ corresponding to the $N$ possible choices of phase for each of the cyclically distinct $m$-sequences. The result is roughly $10^6$ possible codes of periods 31 to 4095 (there are 474 $m$-sequences of periods 31 to 4095), and there is a need for some criteria for sorting out the 'best' codes from this large selection. However, since no expression similar to equation 2.2 exists for the aperiodic parameters, one is left with an exhaustive search. Using a computer to find the phase for which $\hat{c}(1)$ is minimized will give, for several of the sequences, more than one possible phase. Hence such a first sieve does not generally give a unique set of optimal sequences.

A second condition that might be applied to reduce the probability of acquiring false synchronization, and which reduces the number of times. Both these sieves are concerned with peak aperiodic autocorrelation values. However, in a multiple access system the degree of mutual interference is very important. In particular, knowledge of the sidelobe energy of a sequence defined as
can be used to bound the value of the average other user interference between codes[5]. This parameter can be selected as the final sieve in sorting out a cyclic shift optimal to the sequence in question.

Applying these sieves in successive order gives a unique cyclic shift for all the m-sequences with periods from 31 to 4095.

The conditions satisfied for optimal sequences with least sidelobe energy can then formally be written as:

1. $\hat{\phi}_{k}(1)_{\text{max}} \leq \hat{\phi}_{k}(T^{s_1})$ for arbitrary $s$

2. When the number of elements of a finite set $A$ is denoted by the cardinal number $|A|$, the cardinality of the set \{$1; |\hat{\phi}_{k}(1)| = \hat{\phi}_{k\text{max}}, 0 < 1 \leq N-1 \}$ should be minimized.

3. $s_{k}$ is kept at a minimum.

Table 3.1 in the reference[7] gives the numerical results of the codes chosen from these criteria for all m-sequences with periods from 31 to 4095.

For many applications the sidelobe energy parameter may be more important than the peak correlation parameters. A reversal of the priority on peak correlation parameters
and sidelobe energy, to give a set of phases which are auto-optimal among the set of phases, produces the numerical values in Table 2.1. The first column identifies the m-sequence in octal notation. The next column shows the initial or start position of the sequence when in auto-optimal phase. The third and fourth columns contain the same information as the first two for the reciprocal sequence. The fifth column gives the value of the sidelobe energy. The sixth column gives the peak aperiodic autocorrelation for the m-sequence pair, and the final includes the number of times the peak correlation value occurs. This table only covers sequence periods from 31 to 255. Here we can notice that there are a few sequences that are optimal irrespective of where the priority is put.
<table>
<thead>
<tr>
<th>Sequence</th>
<th>Loading Sequence</th>
<th>Loading</th>
<th>SE</th>
<th>Min</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) (n = 5, N = 31)</td>
<td>45 10011</td>
<td>51 10100</td>
<td>103</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>67 11000</td>
<td>73 01010</td>
<td>115</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>75 11110</td>
<td>57 10010</td>
<td>91</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>(b) (n = 6, N = 63)</td>
<td>103 000010</td>
<td>141 011111</td>
<td>427</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>133 100010</td>
<td>155 101100</td>
<td>492</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>147 110001</td>
<td>163 101011</td>
<td>351</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>(c) (n = 7, N = 127)</td>
<td>211 1100100</td>
<td>221 0100111</td>
<td>1915</td>
<td>21</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>217 0000101</td>
<td>361 1111111</td>
<td>2015</td>
<td>15</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>235 0110000</td>
<td>271 0010001</td>
<td>2107</td>
<td>21</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>247 0010111</td>
<td>345 0110001</td>
<td>2255</td>
<td>17</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>277 1001101</td>
<td>375 0000111</td>
<td>2167</td>
<td>21</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>357 1011110</td>
<td>367 0000101</td>
<td>2199</td>
<td>21</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>323 1111011</td>
<td>313 0001110</td>
<td>2055</td>
<td>21</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>203 0110111</td>
<td>301 1011011</td>
<td>2043</td>
<td>21</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>325 0101011</td>
<td>253 0100101</td>
<td>2191</td>
<td>27</td>
<td>2</td>
</tr>
<tr>
<td>(d) (n = 8, N = 255)</td>
<td>455 11011110</td>
<td>551 00000011</td>
<td>9463</td>
<td>29</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>453 11111111</td>
<td>651 01001000</td>
<td>9099</td>
<td>27</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>435 11000010</td>
<td>561 11111100</td>
<td>9059</td>
<td>27</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>537 10110100</td>
<td>765 00111100</td>
<td>8833</td>
<td>25</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>545 00110011</td>
<td>515 10110110</td>
<td>9383</td>
<td>29</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>543 01000110</td>
<td>615 10101001</td>
<td>9215</td>
<td>33</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>607 11100001</td>
<td>703 1011101</td>
<td>9223</td>
<td>29</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>717 01110011</td>
<td>747 00111100</td>
<td>8899</td>
<td>31</td>
<td>2</td>
</tr>
</tbody>
</table>


We noticed that the periodic autocorrelation for m-sequences is ideal. The aperiodic autocorrelation parameters are obtained by a computer search which does not reveal any ideal m-sequences, only best ones selected from
a large set. It would therefore be worth while to know how optimal these m-sequences are compared with a more general set of possible periodic spreading codes. Sarwate[6] showed that the aperiodic autocorrelation functions for a set of K complex valued sequences of period N are bounded by

\[ \hat{\theta}_{\text{max}}(1) \geq N[(K-1)/(NK-1)]^{1/2} \quad (2.6) \]

where

\[ \hat{\theta}_{\text{max}}(1) = \max |e^k(1), e^{kr(1)}| \]

For ranges of K from 2 to K = N and large N, this lower bound varies from \((1/2)N^{1/2}\) to \(N^{1/2}\). With \(N = 1023\) the value for \(\hat{\theta}_{\text{max}}(1)\) using the bound in equation 2.6 ranges from \(= 23\) to \(= 32\) when \(K = 2\) and \(K\) is large respectively.

Golay[8] derives a 'merit factor' for general binary sequences as

\[ F = \frac{N^2}{2S(a^k)} \quad (2.7) \]

and establishes a conjectured asymptotic value for \(F\) that will be valued for every long binary sequences. The bound on \(F\) is given by

\[ F_{\text{opt}} = \frac{12.3248}{(8\pi N)^{3/2N}} \]
From equation 2.7 it is clear that choosing m-sequences with a high merit factor is the same as selecting the sequences by the least sidelobe energy criteria, the result of which is shown in Table 2.1. For a random binary sequence the average F value becomes

\[ F_{\text{ave}} = \frac{N^2}{(N - 1)N} \approx 1 \]

for N values of interest. Thus optimal m-sequences perform considerably better with respect to the merit factor than true random sequences.
2.2 Crosscorrelation

2.2.1 Introduction

Crosscorrelation is of interest in several area such as CDMA systems (or any code addressed system) in which response of the receiver to any signal other than the proper addressing sequence is not allowable, and antijamming systems that may employ codes with extremely low crosscorrelation as well as unambiguous autocorrelation.

Crosscorrelation is the measure of similarity between two different code sequences. The only difference between autocorrelation and crosscorrelation is that in the general convolution integral for autocorrelation a different term is substituted

$$\theta_{\text{cross}} = \int_{-\infty}^{\infty} f(t)g(t - r)dt$$

Crosscorrelation for different code sequences can be tabulated by generating a comparison table and curve of agreements minus disagreements.

That is, it is expressed as the number of agreements minus the number of disagreements when the code or codes are compared chip by chip the same as for autocorrelation.
But even the linear maximal sequences are not immune to crosscorrelation problems, though they are, in general, the best available. It is also of some interest to note, even when the codes used exhibit excellent crosscorrelation properties when averaged over their entire length, that short-term crosscorrelations, which are quite effective in disrupting communications, can (and do) occur.

Here, we have restricted our consideration in this section to integration over a long period (-∞ to ∞). This is essentially (for our simple case) the same as integrating over the code length, for the codes spoken of here repeat at intervals of 2^n - 1 chips. We hasten to point out that integration (as in a synchronization detector) over a period less than that of the code used allows short-term correlations; that is, a short pattern occurring in two different codes or twice in the same code could appear as a legitimate code synchronization when the integration period does not significantly exceed the pattern period.
Fig. 2.6 Comparative autocorrelation and crosscorrelation for 31-chip mirror image m-sequences.
Figure 2.6 and 2.7 are given as illustrations of crosscorrelation and autocorrelation for maximal sequences. The autocorrelation for curve of the [5,3] code shows a zero-shift correlation value of 31. For the [5,3] and [5,2] codes cross-correlated, however, the peak value is 11. It gives an index of discrimination of 20 and is 37.5% less than the autocorrelation value. The [5,3] and [5,2] codes are images; that is, one is the same as the other, but generated in reverse order. Crosscorrelation of the [5,3] and [5,4,3,2] code is lower than that for the image codes, but is still such that the peak crosscorrelation value is seven, a value that occurs at 10 different shift positions.

The significant point is that these particular pairs of code sequences are not capable of operating in the same
Fig. 2.7 Comparative autocorrelation and crosscorrelation for 31 chip m-sequences (not images).
link if the transmitted power from either transmitter exceeds the other enough to raise the peak crosscorrelation to a value near peak autocorrelation. Of course, such short codes should not be used, but the comparison is reasonably representative of operation even with much longer sequences used in code division multiplexing or other multiple access applications.

Judge[9] has considered code division multiplexing by using quasi-orthogonal binary function (linear maximal sequences) and states that for two equal power signals multiplexed together signal to noise is

\[
\frac{S}{N(2)} = \frac{S}{(K_1^2 + T_0/T)^{1/2}}
\]

in each receiver. For b signals

\[
\frac{S}{N(b)} = \frac{S}{[b(K_1^2 + T_0/T)]^{1/2}}
\]

where

T = the crosscorrelation integration period,
T_0 = the code bit period,
K_1 = value of DC correlation.

Judge's result shows that some Mersenne prime sequences exhibit crosscorrelation values superior to others, sometimes even for nonprime sequences longer than prime sequences.
In systems using mark-space signalling, or in code division multiple access applications, the periodic and aperiodic crosscorrelation functions are of major importance. The detailed discussion for periodic and aperiodic crosscorrelation is given in next two section.

2.2.2 Periodic crosscorrelation

Since no analytical expression is known which can be used to calculate the crosscorrelation function between two particular m-sequences, the values must be computed by performing multiplication bit by bit and adding the result as we discussed before. This is a somewhat impractical solution, and it would at least be attractive if it would be shown statistically that the fluctuation in the crosscorrelation value for a pair of sequences was small for all relative phase shifts so that the mean could be used representatively. However, in Appendix D it is shown that the mean of the periodic crosscorrelation for all shifts is given by 1/N and the variance for all shifts is approximately given by 1 \((1/N)\).

For instance, for m-sequence pairs of period \(N = 1023\) the maximum crosscorrelation value may vary from 65 to 383 from pair to pair[1]. However, if the sequence pair with the lower crosscorrelation bounded by 65 is chosen there
is still a fluctuation from relative phase shift to phase shift of $1 \leq |\phi^{kr}| \leq 65$. Thus the mean is not very representative for the periodic crosscorrelation from relative phase shift to phase shift. In order to indicate what can be expected for the crosscorrelation a few lower bounds can be established. From the result of Gold[2] developed in Appendix C,

$$\sum_{l=0}^{N-1} [\phi^{kr}(l)]^2 = \sum_{l=0}^{N-1} \phi^k(l)\phi^r(l)$$

(2.8)

The left-hand side of equation 2.8 is upper bounded by $N_{\text{max}}(\phi^{kr})$. Furthermore, the right-hand side cannot be less than $N^2 - (N - 1)\text{max}(\phi^k)\text{max}(\phi^r)$. Therefore

$$N \text{max}(\phi^{kr}) > N^2 - (N - 1)\text{max}(\phi^k)\text{max}(\phi^r)$$

Since m-sequences are considered, the maximum value of the autocorrelation outside the main peak is $-1$, so that

$$\text{max}(\phi^{kr}) > (N - 1)^{1/2}$$

A much tighter bound due to Sidel’nikov[19] is valid for a set of $K$ sequences where $K > N$. Then the maximum crosscorrelation between any sequence pair taken from this set is lower bounded by

$$\text{max} (\phi^{kr}) > (2N - 2)^{1/2}$$

(2.9)

A designer’s approach to the avoidance of large mutual crosscorrelation has been to examine the factors of the period of the sequences. If it was found that the sequence period had small factors, the chance of large values of
crosscorrelation between some pairs was considered high. It is however a rather drastic approach to use this guide and totally exclude all sequences with periods having small factors does however a general sieve to remove only those particular sequence pairs which are likely to exhibit large crosscorrelation. It can be shown that it is the combination of sequence period factors and the decimation property of m-sequences which makes possible large crosscorrelation values. Let \( q \) denote a positive integer, and consider the sequence \( a^r \) formed by taking every \( q \)th element of the m-sequence \( a^k \). The sequence \( a^k \) is said to be a decimation by \( q \) of \( a^k \). If \( \gcd(N, q) = 1 \), where "gcd" denotes the greatest common divisor, the decimation is called proper and the sequence \( a^r \) of period \( N \) is another m-sequence. In this way it is possible to construct all m-sequences of a particular period by proper decimations of one m-sequence.

The crosscorrelation between two sequences \( a^k \) and \( a^r \) can now be defined as

\[
\phi^{kr}(i) = \sum_{l=0}^{N-1} a^k_l a^r_{l+i} = \sum_{l=0}^{N-1} a^k_l a^k_{q(l+i)}
\]

This means that the crosscorrelation between \( a^k \) and \( a^r \) for any \( i \) can be obtained through the process of multiplying the digits \( a_l \) and \( a_{ql} \), \( l = 1, 2, \ldots, N \), and summing over the period where \( \gcd(N, q) = 1 \).
To obtain the mutual crosscorrelation for the different relative phases \( i \) between the \( m \)-sequence pair, it is necessary to carry out the decimation on all the cyclic permutations. However, the decimations by any particular \( q = q_1 \) on the cyclic permutations \( T^s a^k, s = 0, 1, \ldots, N-1 \) will in general not lead to all possible relative phase positions between \( a^k \) and \( a^r \). The complete result is obtained through further decimations of the form \( q_2 = q_1^{2^h} \mod N \) for positive integers \( h \). The number \( c \) of times \( T^s \) can be used on the original sequence \( a^k \), after which decimation by \( q_1 \) results in a new phase for \( a^r \), is given by

\[
 cq \mod(yN) = c
\]

(2.10)

where \( y \) is any positive integer. Equation 2.10 can also be written as

\[
\frac{c}{y} = \frac{N}{q-1}
\]

(2.11)

The number of times \( T^s \) can be used on the original sequence \( a^k \), after which decimation by \( q_1 \) does not result in a new phase for \( a^r \), is a direct measure of how often the sequence digits \( a_l = a_{q_1 l} \). The number of times \( a_l = a_{q_1 l} \) equals \( N/c = |D| \), where \(| \cdot | \) denotes the number of elements in the set \( D \), so that

\[
\sum_{l=D} a^k_{q_1 l} a^k_{q_1 l} = +|D|
\]

36
both of which give +1 as the result. For cases where |D| is large, the crosscorrelation will have a large positive bias, which is sufficient for one to expect the correlation to exhibit large peak values.

Gold[4] describes in his paper an analytical technique which tells how to select m-sequences with a specified upper bound on the crosscorrelation function.

2.2.3 Aperiodic crosscorrelation

Although there are no analytical expressions which give the values of $\hat{\theta}_{kr}$ for a set of m-sequences, several bounds can be applied. The result in equation 2.6 for a set of K sequences can also be used to lower bound the aperiodic crosscorrelation since $\theta_{max}$ is the maximum of $\hat{\theta}_k$ and $\hat{\theta}_{kr}$. In fact equation 2.6 is derived from an expression relating the maximum values of the aperiodic autocorrelation and aperiodic crosscorrelation parameter is shown in Appendix E. thus

$$\frac{(\hat{\theta}_{kr})^2}{N} + \frac{N - 1}{N(K-1)} \left[ \frac{(\hat{e}_k)^2}{N} \right] > 1 \quad (2.12)$$

This seems to verify the common observation that when autocorrelation parameters are good the crosscorrelation parameters are not very good. Any of the sequences which satisfy the bound from Gold's theorem are upper bounded by
As for the aperiodic autocorrelation, the aperiodic crosscorrelation is sensitive to phase shifts. It is, however, a formidable task to find optimal phases for multiple access applications which require a large number of sequences. Pursley and Roefs[21] have calculated peak crosscorrelation parameters for some short m-sequences and Gold sequences.

Notice that although reciprocal sequences have periodic crosscorrelations very close to the bound from Gold’s theorem and thus are attractive from a periodic viewpoint, the interference due to other users depending on \( c_k(l)c_r(l) \) will under certain conditions be large for such sequences. The reason is that when \( k \) and \( r \) are reciprocal sequences and are placed in phase positions so as to obtain optimal sequences with least sidelobe energy, these phase positions \( (T^{a_k}) \) and \( (T^{a_r}) \) are unique and such that \( \alpha + \beta = N + m \). Under this condition \( c_k(l) = c_r(l) \) and the product becomes positive in the summation \( c_k(l)c_r(l) \).

If however some of the sieves used to find optimal sequences with least sidelobe energy are removed, more than one possible phase exists and it is possible to obtain \( \alpha + \beta = N + m \). In this case reciprocal sequences will not behave differently from any other sequence pair.
2.3 Crosscorrelation spectra in Gold codes

The Gold code sequences are of great utility when crosscorrelation is a prime consideration. Their real advantages lies in that for every code in a set of \(2^n - 1\) codes, each of length \(2^n - 1\), crosscorrelation values are well defined, and a system can be designed to operate within this definition.

The Gold code sets to be defined shortly have a crosscorrelation spectrum which is three-valued.

Consider an \(m\)-sequence that is represented by a binary vector \(b\) of length \(N\), and a second sequence \(b'\) obtained by sampling every \(q\)th symbol of \(b\). The decimation of an \(m\)-sequence may or may not yield another \(m\)-sequence. When the decimation does yield an \(m\)-sequence, the decimation is said to be a proper decimation. The table of irreducible polynomials in reference[17] can be used to determine whether a particular decimation of a particular \(m\)-sequence is proper. One entry from this table is:

\[
\begin{array}{cccccccc}
\text{DEGREE} & 6 & 1 & 103F & 3 & 127B & 5 & 147H & 7 & 111A \\
9 & 015 & 11 & 155E & 21 & 007 \\
\end{array}
\]
Here, each octal number represents a polynomial and those numbers followed by an E, F, G, or H are primitive polynomials which generate m-sequences. Let \( b \) denote the m-sequence generated by 103 in the table. The decimal number \( q \) preceding the octal entry indicates that the sequence generated by that polynomial is the \( q \)th decimation of the sequence generated by the first entry in the table. Thus \( b' = b[3] \) is generated by the polynomial 127, which is not primitive, so that \( b' \) is not an m-sequence and this decimation is not proper. It has also been proven (Sarwate and Pursley) that \( b' = b[q] \) has period \( N \) if and only if \( \gcd(N,q) = 1 \). Since \( N = 2^6 - 1 = 63 \) for the degree 6 polynomials, \( \gcd(63,3) = 3 \) and the period of \( b' \) does not equal \( N \). Sarwarte and Pursley have also shown that proper decimation by odd integers \( q \) will give all of the m-sequences of period \( N \). Thus any pair of m-sequences having the same period \( N \) can be related by \( b' = b[q] \) for some \( q \).

The crosscorrelation spectrum of pairs of m-sequences can be three-valued, four-valued, or possibly many-valued, where those three values are \( -t(n), -1, [t(n) - 2] \)

where

\[
t(n) = \begin{cases} 
1 + 2^{(n+1)/2} & \text{for } n \text{ odd} \\
1 + 2^{(n+2)/2} & \text{for } n \text{ even}
\end{cases}
\]
where the code period $N = 2^n - 1$, are called preferred pairs of $m$-sequences. Finding preferred pairs of $m$-sequences is necessary in defining sets of Gold codes. The following conditions are sufficient to define a preferred pair $b$ and $b'$ of $m$-sequences:

1. $n \not\equiv 0 \mod 4$; that is, $n$ is odd or $n = 2 \mod 4$
2. $b' = b[q]$ where $q$ is odd and either
   \[ q = 2k + 1 \]
   or \[ q = 2^{2k} - 2^k + 1 \]
3. $\gcd(n, k) = 1$ for $n$ odd
   \[ 2 \text{ for } n = 2 \mod 4 \]

Here, we find a preferred pair of $m$-sequences having a period 31 units and evaluate their crosscorrelation spectrum for the period, $N = 31$, degree, $n = 5$. The referenced (Peterson and Weldon) table of irreducible polynomials contains the following entry:

```
DEGREE 5 1 45 E 3 75G 5 67H.
```

Arbitrarily choose $b$ as the $m$-sequence generated by the primitive polynomial 45. The decimation $b' = b[3]$ is proper, so that the pair $(b, b[3])$ is a candidate pair. The first condition is satisfied since $n = 1 \mod 4$. The second condition is satisfied also since $q$ is odd and $q = 2^k + 1$ for $k = 1$. Finally, $\gcd(5, 1) = 1$, so that all three conditions are satisfied and a preferred pair has been found. The $m$-sequences $b$ and $b[3]$ are
Fig. 2.8 Crosscorrelation spectrum for $N = 31$, $n = 5$

A straightforward but tedious manual calculation of the crosscorrelation will show in Figure 2.8 that for any phase shift the crosscorrelation takes on one of the three values $-9$, $-1$, or $7$.

Let $b(D)$ and $b'(D)$ represent a preferred pair of $m$-sequences having period $N = 2^n - 1$. The family of codes defined by, \( \{b(D), b'(D), b(D) + b'(D), b(D) + Db'(D), b(D) + D^2b'(D), \ldots, b(D) + D^{N-1}b'(D)\} \) is called the set of Gold codes for this preferred pair of $m$-sequences. In this definition, the notation $D^j b'(D)$ represents a phase shift of the $m$-sequence $b'(D)$ by $j$ units.
The result that Gold codes have three-level crosscorrelation values with crosscorrelation and relative (approximate) frequencies of occurrence is shown in Table 2.2

<table>
<thead>
<tr>
<th>SRG stages</th>
<th>Code length</th>
<th>Value of crosscorrelation</th>
<th>Frequency of occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td>n odd</td>
<td>$N = 2^n - 1$</td>
<td>-1</td>
<td>= .50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$-(2^{(n+1)/2} + 1)$</td>
<td>= .25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$(2^{(n+1)/2} - 1)$</td>
<td>= .25</td>
</tr>
<tr>
<td>n even</td>
<td>$N = 2^n - 1$</td>
<td>-1</td>
<td>= .75</td>
</tr>
<tr>
<td>not divisible by 4</td>
<td>$-(2^{(n+2)/2} + 1)$</td>
<td>= .125</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$(2^{(n+2)/2} - 1)$</td>
<td>= .125</td>
</tr>
</tbody>
</table>

Table 2.2 Three-level crosscorrelation properties in Gold codes.
CHAPTER III
GOLD CODE GENERATION

3.1 Algorithm for Gold code selection

Any code can be represented by a polynomial, where the binary codes are represented by a polynomial of the form

\[ 1 + AX + BX^2 + CX^3 + \ldots + ZX^n \]

Here, each coefficient (A, B, ..., Z) is either 0 or 1, each term of the polynomial (except for the first, 1) corresponds to a stage of a binary shift register, and there are n stages in the register. That is, each term in the polynomial containing an X corresponds one-to-one with a stage in a binary shift register.

The feedback connections in the code generator are defined by the terms in the polynomial whose coefficient is 1.

For instance, a code generator whose characteristic polynomial is \( 1 + 1X + 1X^2 + 1X^3 + 1X^7 \) would have seven stages with feedback taken from its first, second, third, and seventh stages. Here, we can express this code as

\[ [7, 3, 2, 1] = 1 + X + X^2 + X^3 + X^7 \]

Linear maximal codes, in which we have major interest, have characteristic polynomials that are primitive. That is, the primitive or nonfactorable polynomials each define
a different linear maximal code. Fortunately, the tables of polynomials mentioned in Appendix F both define primitive polynomials and provide information that allows proper selection of pairs of codes for use in generating Gold codes.

<table>
<thead>
<tr>
<th>Degree</th>
<th>Period</th>
<th>Worst case</th>
<th>Preferred pair</th>
<th>Difference (DB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>31</td>
<td>11</td>
<td>9</td>
<td>1.7</td>
</tr>
<tr>
<td>6</td>
<td>63</td>
<td>23</td>
<td>15</td>
<td>3.7</td>
</tr>
<tr>
<td>7</td>
<td>127</td>
<td>41</td>
<td>17</td>
<td>7.6</td>
</tr>
<tr>
<td>8</td>
<td>255</td>
<td>95</td>
<td>31</td>
<td>9.7</td>
</tr>
<tr>
<td>9</td>
<td>511</td>
<td>113</td>
<td>33</td>
<td>10.7</td>
</tr>
<tr>
<td>10</td>
<td>1023</td>
<td>383</td>
<td>63</td>
<td>15.7</td>
</tr>
<tr>
<td>11</td>
<td>2047</td>
<td>287</td>
<td>65</td>
<td>12.9</td>
</tr>
<tr>
<td>12</td>
<td>4095</td>
<td>1407</td>
<td>127</td>
<td>20.9</td>
</tr>
<tr>
<td>13</td>
<td>8191</td>
<td>703</td>
<td>127</td>
<td>14.9</td>
</tr>
</tbody>
</table>

Table 3.1 Performance of preferred pairs compared with worst case pairs.

Before going further, let us make it clear that the codes need be properly chosen. Arbitrary selection of code pairs from the tables can result in very poor correlation
performance, as is demonstrated by the results shown in Table 3.1.

The preferred pairs of codes, as selected by the Gold-derived algorithm, always give undesired correlation that is bounded at $2^{(n+1)/2} + 1$ (n odd) and $2^{(n+2)/2} - 1$ (n even), however.

The Gold-derived algorithm for selection of preferred pairs requires the use of code tables that list the polynomial roots (as do Peterson’s tables[17]). The algorithm is used as follows:

1. Select a polynomial of the proper degree from the table (an n-stage shift register requires an nth degree polynomial).

2. Read the number (k) in the polynomial roots column associated with the polynomial selected.

3. If the code generator has an odd number of stages, then calculate $2^k + 1$. If the number of stages is even, calculate $2^{(k+2)/2} + 1$.

4. The number calculated in step 3 is the polynomial root of a second code that completes a preferred pair.

Use of any polynomial (code) with the polynomial root calculated in step 4 will produce Gold codes when combined with the original code that has properly bounded
correlation with all members of the set. As an example, for 19-stage codes, suppose we select a code whose polynomial is 2000047, which converts to 01000000000000100111 or 1 + x + x^2 + x^5 + x^{19}. The polynomial has 1 as its polynomial roots. With the algorithm from the foregoing definition (step 4), we calculate the second polynomial root required as 2^k + 1 = 2^1 + 1 = 3.

<table>
<thead>
<tr>
<th>Irreducible Number of polynomials maximals</th>
<th>Degree</th>
<th>Polynomial roots</th>
<th>Correlation function of sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marsh (1957)</td>
<td>Yes</td>
<td>All</td>
<td>19</td>
</tr>
<tr>
<td>Peterson (1961)</td>
<td>Yes</td>
<td>All</td>
<td>16</td>
</tr>
<tr>
<td>Watson (1961)</td>
<td>No</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>Gold (1964)</td>
<td>No</td>
<td>All</td>
<td>13</td>
</tr>
<tr>
<td>Bradford (1965)</td>
<td>No</td>
<td>Partial</td>
<td>58</td>
</tr>
</tbody>
</table>

Table 3.2 Description of available tables of binary polynomials.

A second polynomial having polynomial root = 3 is 2020471 or 1 + x^3 + x^4 + x^5 + x^8 + x^{13} + x^{19}. This pair of codes would, when combined, produce Gold codes, every one of
which would have crosscorrelation bounded at \(2^{(n+1)/2} + 1 = 1025\), which is \(20 \log_2 19/1025 = 54\text{dB}\) below the peak of autocorrelation. A listing of some readily available tables of primitive polynomials is given in table 3.2. Note that of those given, however, only two list the polynomial roots.

### 3.2 Generation of general Gold codes

Gold code sequence generators are useful because of the large number of codes they supply, although they require only one pair of feedback tap sets. A bonus awarded on the basis of the use of these codes is that only a few sets of feedback connection are required for each simple shift register generator (SSRG) while retaining the capacity to generate a large number of codes. The single-tap SSRG is the fastest configuration possible. Thus the Gold code sequences are potentially available at rates equal to the capacity of the fastest SSRG.

The Gold codes are generated by modulo-2 addition of a pair of maximal linear sequences as shown in Figure 3.1. The code sequences are added chip-by-chip by synchronous clocking. Here, code 1 and code 2 are the same length. Thus the two code generators maintain the same phase
relationship, and the codes generated are the same length as the two base codes, which are added together but are nonmaximal. A specific example is shown in Figure 3.2.

Fig. 3.1 Configuration of Gold code generator

The shift and add property of maximal sequences tells us that any maximal sequence added to a phase-shifted replica of itself (any integral number of bits) produces a different phase shift as an output. Here the same
operation is performed, with the new sequence having the same length as those being added, and nonmaximal. Furthermore, every change in phase position between the two generators causes a new sequence to be generated. To see this advantage, consider the following example.

Given a five-stage sequence generator, we choose a set of feedback taps, [5,3] and [5,4,2,1] from the reference table in Appendix F. Here, there are only six feedback sets available for the five-stage register and half are images of the other half. If more than six 31-chip codes are needed, we cannot get them from our five-stage register.

Therefore, we use two five-stage sequence generators connected in the Gold code sequence generator configuration, as shown in Figure 3.2. Table 3.3 also shows the modulo-2-combined Gold codes produced by combining the two output maximal codes with different initial offsets; that is, the two code generators are started with initial conditions offset by various amounts to give different output codes. The all-ones vector is set into both registers as an initial condition. In addition one, three, and five-chip shifts (from all-ones vector) are also shown in initial conditions:
Fig. 3.2 Gold code generation.

<table>
<thead>
<tr>
<th></th>
<th>[5,4,2,1]</th>
<th>[5,3]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11111011001110001110101001010110</td>
<td>1111100011101110101000100101100</td>
</tr>
<tr>
<td>Zero-shift</td>
<td>00000011111001010001010000001110</td>
<td>000010101000010111011101110111</td>
</tr>
<tr>
<td>combination</td>
<td></td>
<td></td>
</tr>
<tr>
<td>One-chip-shift</td>
<td>000111011101001001111011101000100101</td>
<td>0011110111010010011110001000101</td>
</tr>
<tr>
<td>combination</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Three-chip-shift</td>
<td>11100000100100001000001000101111101</td>
<td></td>
</tr>
<tr>
<td>combination</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Five-chip-shift</td>
<td>11110000100100000100001000101111101</td>
<td></td>
</tr>
<tr>
<td>combination</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3 Modulo-2-combined Gold code
Any shift in initial conditions from 0 to 30 chips can be used (A 31-chip shift is the same as the zero shift). Thus, from this Gold sequence generator, 33 maximal-length codes are available. Extending this demonstration, we can show that any two-register Gold code generator of length n can generate $2^n - 1$ m-sequences (length $2^n - 1$) plus the two maximal base sequences. A multiple-register Gold code generator can generate $(2^n - 1)^r$ nonmaximal sequences of length $2^n - 1$ plus r m-sequences of the same length where r is the number of registers and n is register length.

In addition to their advantage in generating large numbers of codes, the Gold codes may be chosen so that over a set of codes available from a given generator the crosscorrelation between the codes is uniform and bounded. Thus the Gold codes are attractive for applications in which a number of code-division-multiplexed signals are to be used. The same guarantee of bounded crosscorrelation is impossible for m-sequences of the same length.

Gold[22] has presented a method for choosing the linear maximal codes used as components to Gold sequences that gives a set of sequences, each of whose members has crosscorrelation, and autocorrelation side lobes, bounded by $|\theta(r)| \leq 2^{(n+1)/2} + 1$ for n odd, and by $|\theta(r)| \leq 2^{(n+2)/2} - 1$ for n even.
An equivalent result is given by Anderson[32] for the Gold codes; that is, Anderson's expression for the correlation bound is

$$|\Theta(r)|_G \leq \left[\left(\sqrt{2} \sqrt{1+1/L} + 1/\sqrt{2}\right)/\sqrt{L}\right]^{1/2}$$

where L is correlation value.

It is apparent from this expression that as $L \rightarrow \infty$, $|\Theta(r)| \rightarrow 2/L$. Convergence is sufficiently rapid that for any code sequence length of interest $|\Theta(r)| = \sqrt{2/L}$.

Here one expression gives crosscorrelation in chips, whereas the other gives a percentage of maximum correlation. By normalizing maximum correlation to one, $2(n+1)/2 + 1 \approx \sqrt{2L/ L} = \sqrt{2}$ for large L.

Anderson also states that the crosscorrelation function for maximal sequences is bounded by

$$|\Theta(r)| \leq \left\{1 + 1/L - 1/L^2\right\}^{1/2}$$

Now, as $L \rightarrow \infty$, $|\Theta(r)| \rightarrow 1/\sqrt{L}$. For a given value of L the Gold codes exhibit crosscorrelation that is $(\sqrt{2}/L)/(1/\sqrt{L}) = \sqrt{2}$ greater than m-sequences of the same length.
3.3 Generation of balanced Gold codes

Gold has shown that Gold codes can be broken into three classes of balance. A balanced code is one in which the number of "zeros" differs from the number of ones by one. The other two classes have an excess and deficiency of "ones". For $n$ odd Gold has shown that the number of "ones" and the number of codes with that number of "ones" is as shown in Table 3.4.

<table>
<thead>
<tr>
<th>Set</th>
<th>Number of &quot;Ones&quot;</th>
<th>Number of codes with this number of &quot;ones&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2^{n-1}$</td>
<td>$2^{n-1} + 1$</td>
</tr>
<tr>
<td>2</td>
<td>$2(n-1) + 2(n-1)/2$</td>
<td>$2(n-2) - 2(n-3)/2$</td>
</tr>
<tr>
<td>3</td>
<td>$2(n-1) - 2(n-1)/2$</td>
<td>$2(n-2) + 2(n-3)/2$</td>
</tr>
</tbody>
</table>

Table 3.4 Number of balanced and unbalanced codes for $n$ odd.

Here, in the first set there are $2^{n-1}$ "ones" and therefore $2^{n-1} - 1$ "zeros" and therefore set 1 is balanced. Sets 2 and 3 are not balanced. Because balanced codes have more desirable spectral characteristics, we show, following Gold how to generate balanced codes; that is, we generate Gold codes of the first set. We do this by selecting the proper relative phase of the two original m-sequences.
First we must determine the characteristic phase. Every \textit{m}-sequence has a characteristic phase. One important property is that if a maximal PN sequence, in its characteristic phase, is sampled at every other symbol, the same sequence results. Let \(f(x)\) be the \(n\)th degree characteristic. Any phase of the \(m\)-sequence can be represented by the ratio \(g(x)/f(x)\), where \(g(x)\) is the number of the generating function and is of degree less than \(n\). As we have seen, long division of these polynomials results in a formal binary power series whose binary coefficients are the symbols of the sequence generated by the shift register. The formula for the polynomial \(g(x)\) that results in the characteristic phase for the \(m\)-sequence has been shown by Gold to be given by
\[
\begin{align*}
g(x) &= \frac{d[xf(x)]}{dx} \quad f(x) \text{ odd degree} \\
g(x) &= f(x) + \frac{d[xf(x)]}{dx} \quad f(x) \text{ even degree}
\end{align*}
\]
Differentiation is carried out in the usual way with coefficients interpreted, mod 2.

As an example, consider the characteristic polynomial \(f(x) = 1 + x + x^3\).

We compute
\[
\begin{align*}
g(x) &= d(x + x^2 + x^4)/dx = 1 \mod 2
\end{align*}
\]
Therefore the characteristic phase is given by
\[
G(x) = 1/(1 + x + x^3)
\]

55
By long division we have

\[
\begin{array}{c|cccccc}
& 1 & + & x & + & x^2 & + & x^4 & + & x^7 & + & \ldots \\
\hline
1 + x + x^3 & \overline{1} \\
& 1 & + & x & + & x^3 & & & & & & \\
\hline
& x & + & x^3 & & & & & & & & \\
& & x & + & x^2 & + & x^4 & & & & & \\
\hline
& & x^2 & + & x^3 & + & x^4 & & & & & \\
& & & x^2 & + & x^3 & + & x^5 & & & & \\
\hline
& & & x^4 & + & x^5 & & & & & & \\
& & & & x^4 & + & x^5 & + & x^7 & & & \\
\hline
& & & & & & & x^7 & + & x^8 & + & x^{10}
\end{array}
\]

We conclude that the initial conditions must have been 111 to yield the successive "ones". The sequence and the sequence represented by odd numbered symbols is given by

- sequence 11101001110100...
- sampled sequence 1 1 1 0 1 0 0...

3.3.1 Relative phase requirement for balanced codes

We shall now describe the relative phase in which the preferred pair of maximal PN sequences must be added in order to result in a balanced member of the family.
Let \( a \) and \( b \) be the preferred pair of m-sequences in their characteristic phase. When \( x \) is of odd degree it is clear that the generator polynomial is of the form

\[
G(x) = \frac{1 + c(x)}{1 + d(x)}
\]

where the degree of \( d(x) \) is \( n \) and the degree of \( c(x) \) is not greater than \( n - 1 \). By long division it is clear that the quotient will be of the form \( 1 + x + \ldots \) so that the initial symbol of the characteristic sequence will be a "one".

Any relative phase shifts of the sequence \( a \) and \( b \) (in their characteristic phase) that are obtained by shifting the sequence \( b \) until its initial "one" corresponds to a "zero" in the sequence \( a \) will result in a balanced Gold code when the two sequences are added together mod 2.

Let us consider an example. Let \( n = 3 \) and \( f(x) = x^3 + x + 1 \). Now \( s \) is a root of \( f(x) \) and \( s^3 \) is a root of the other part of the preferred pair. Since the polynomial for \( s^3 \) is not listed in reference [Peterson and Weldon], try the reciprocal polynomial, that is, \( g(x) = x^3 + x^2 + 1 \). We find that \( g(s^3) = 0 \). Hence the two sequences are

\[
a = \frac{1}{1 + x + x^3} = 1 + x + x^2 + \ldots
\]

\[= 1110100 \text{ (initial condition is 111)}\]

\[
b = \frac{1 + x^2}{1 + x^2 + x^3} = 1 + 0x + 0x^2 + \ldots
\]

\[= 1001011 \text{ (initial condition is 100)}\]
If the sequence $b$ is shifted cyclically three, five, or six positions to the right, then the initial "one" in sequence $b$ will be under a "zero" in sequence $a$. Addition of the two sequences in all cases leads to a balanced Gold code.

If, however the code is shifted by any other phase (that is, zero, one, two, and four shift), an unbalanced code is produced. These results are summarized in Table 3.5.

\[
\begin{align*}
\text{a} & \quad 1110100 \\
\text{b} & \quad 1001011
\end{align*}
\]

<table>
<thead>
<tr>
<th>Shift(b)</th>
<th>Sequence</th>
<th>a + b</th>
<th>&quot;Ones&quot; - &quot;Zeros&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1001011</td>
<td>0111111</td>
<td>5 (unbalanced)</td>
</tr>
<tr>
<td>1</td>
<td>1100101</td>
<td>0010001</td>
<td>-3 (unbalanced)</td>
</tr>
<tr>
<td>2</td>
<td>1110010</td>
<td>0000110</td>
<td>-3 (unbalanced)</td>
</tr>
<tr>
<td>3</td>
<td>0111001</td>
<td>1001101</td>
<td>1 (balanced)</td>
</tr>
<tr>
<td>4</td>
<td>1011100</td>
<td>0101000</td>
<td>1 (balanced)</td>
</tr>
<tr>
<td>5</td>
<td>0101110</td>
<td>1011010</td>
<td>1 (balanced)</td>
</tr>
<tr>
<td>6</td>
<td>0010111</td>
<td>1100011</td>
<td>1 (balanced)</td>
</tr>
</tbody>
</table>

Table 3.5 Relative phase shifts of the sequence a and b
3.3.2 Initial conditions for balanced Gold codes

We have seen how to proceed to produce balanced Gold codes. Generator for the balanced Gold codes is the same as given in Figure 3.1. The initial conditions for SRG 2 are those initial conditions, of the m-sequence that determine the characteristic phase of the maximal PN sequence generated by shift register 2. These initial conditions are determined such that the numerator of the generating functions is determined by

\[ g(x) = \frac{d}{dx}[xf(x)] \quad n \text{ odd} \]
\[ g(x) = f(x) + \frac{d}{dx}[xf(x)] \quad n = 0 \mod 4 \]

as before. Then the initial conditions are obtained by long division of \( g(x)/f(x) \) to provide the first \( n \) coefficients, which are, in fact, the \( n \) initial conditions.

The initial conditions for the SRG 1 are only subject to the constraint that the first stage (the one on the right) contain a zero.

As an example of the above technique, we shall construct a balanced Gold code of period \( 31 = 2^5 - 1 \). A preferred pair is found in Appendix C[6].

\[
\begin{array}{c|ccc}
1 & 45 & f(x) = 1 + x^2 + x^5 \\
5 & 67 & y(x) = 1 + x + x^2 + x^4 + x^5 \\
\end{array}
\]
since \(2^{(n-1)/2} + 1 = 2^2 + 1 = 5\). Here, polynomials are given in an octal representation. 45 converted to binary digit would give 100101 which specified in terms of variable is \(1 + x^2 + x^5\) since the binary digits are the coefficients of the polynomial. The characteristic sequence generated by the shift register corresponding to polynomial 45 is represented by the ratio

\[
g(x)/(1 + x^2 + x^5)
\]

where,

\[
g(x) = \frac{d}{dx}(x + x^3 + x^6) = 1 + x^2
\]

The initial conditions required for this register are found from the quotient,

\[
(1 + x^2)/(1 + x^2 + x^5) = 1 + x^5 + ...
\]

The initial conditions for register B become

\[
[0 0 0 0 1]
\]

The only constraint on the initial conditions of register A is that the entry in the first stage be zero. Hence our Gold code encoder is shown in Figure 3.3.
An example of the Tracking and Data Relay Satellite System (TDRSS, which is a NASA program) staggered quadriphase signals is shown in Figure 3.4 based on the above stated extension of Figure 3.3.
Fig. 3.4 Gold code pair generator for mode 2 return link.
3.4 Number of Gold codes

Table 3.6 gives the number of Gold codes calculated for m-sequences with periods from 3 to 65,535. The first column identifies the number of shift register stages. The next column shows period of code. The third and fourth columns contain maximal connected sets and total possible preferred pairs respectively[3]. Here, maximal connected set means the largest possible connected set of a preferred pair. The fifth and sixth columns give number of linear m-sequences and number of available all m-sequences separately[14]. Notice that the sixth column includes number of mirror image m-sequences. The final column gives the number of all possible Gold codes including mirror image Gold codes.
<table>
<thead>
<tr>
<th>n</th>
<th>N</th>
<th>M</th>
<th>T</th>
<th>M-sequences</th>
<th>Gold codes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Linear</td>
<td>Total</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>31</td>
<td>3</td>
<td>12</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>63</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>127</td>
<td>6</td>
<td>60</td>
<td>9</td>
<td>18</td>
</tr>
<tr>
<td>8</td>
<td>255</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>511</td>
<td>2</td>
<td>4</td>
<td>10</td>
<td>48</td>
</tr>
<tr>
<td>10</td>
<td>1,023</td>
<td>3</td>
<td>12</td>
<td>10</td>
<td>60</td>
</tr>
<tr>
<td>11</td>
<td>2,047</td>
<td>4</td>
<td>24</td>
<td>10</td>
<td>176</td>
</tr>
<tr>
<td>12</td>
<td>4,095</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>144</td>
</tr>
<tr>
<td>13</td>
<td>8,191</td>
<td>4</td>
<td>24</td>
<td>10</td>
<td>630</td>
</tr>
<tr>
<td>14</td>
<td>16,383</td>
<td>3</td>
<td>12</td>
<td>14</td>
<td>756</td>
</tr>
<tr>
<td>15</td>
<td>32,767</td>
<td>2</td>
<td>4</td>
<td>21</td>
<td>1,800</td>
</tr>
<tr>
<td>16</td>
<td>65,535</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>2,048</td>
</tr>
</tbody>
</table>

Table 3.6 Number of Gold codes

64
Example: For \( n = 9 \), we can get \( N = 2^n - 1 = 2^9 - 1 = 511 \). The referenced table [3] gives 2 as the maximal connected set. Here if we include image cases, the total number of maximal connected sets are 4. Hence the total number of combinations for Gold code generation are:

\[
\binom{4}{2} = 6.
\]

But the pair between certain Gold sequence and its image sequence does not generate Gold code sequence. Since we have to deduct these cases, the total possible preferred pairs are 6 - 2 = 4. Each preferred pair gives rise to a set of Gold sequences, and thus there are 4 different sets of Gold sequences, each of period 511. Each set contains 513 sequences, and the total number of Gold codes is:

\[
4 \times 513 = 2,052
\]

The referenced table [14] gives 10 and 48 as the number for linear m-sequences and all m-sequences respectively.

Notice that we have to choose the Gold code from one preferred pair, when we design a system with CDMA. Because Gold code sequences generated from the different preferred pairs sometimes have a higher crosscorrelation value than the crosscorrelation value for codes generated from the same preferred pair.
Example: Let us consider crosscorrelation values between the Gold code \( a \) generated from preferred pair [5,2] and [5,4,3,2] and the Gold code \( b \) generated from preferred pair [5,2] and [5,4,2,1].

\[
\begin{align*}
a & \quad 1101111110110010100000000111 \\
b & \quad 0010000001100111011110011001111
\end{align*}
\]

A computer calculation shows that the crosscorrelation takes on one of the four values -9, -1, +7, +15. From this we can see that Gold code pair \( a \) and \( b \) does not satisfy the Gold code definition, namely that the crosscorrelation be 3 valued.
CHAPTER IV
NUMBER OF SIMULTANEOUS USERS IN GOLD CODE

4.1 Worst case

For this condition all users are assumed to have the peak magnitude of crosscorrelation that is \(\{2^{(n+1)/2} + 1\}\) for \(n\) odd and \(\{2^{(n+2)/2} + 1\}\) for \(n\) even. The result is generalized as a function of \(n\), which is the number of shift register stages, and tabulated in Table 4.1 for both \(n\) odd and \(n\) even (not divisible by 4).

<table>
<thead>
<tr>
<th>(n)</th>
<th>Code length (N)</th>
<th>Magnitude of crosscorrel.</th>
<th>Number of simultaneous users as a function of n</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>2(^n) - 1</td>
<td>2(^{(n+1)/2}) + 1</td>
<td>(2^{(n-1)/2} - 1) worst case</td>
</tr>
<tr>
<td>odd</td>
<td></td>
<td></td>
<td>(2^{(n+1)/2} - 1)</td>
</tr>
<tr>
<td>(2)</td>
<td></td>
<td></td>
<td>(2^{(n-1)/2})</td>
</tr>
<tr>
<td>even( not divisible by 4)</td>
<td>2(^n) - 1</td>
<td>2(^{(n+2)/2}) + 1</td>
<td>(2^{(n-2)/2} - 1) worst case</td>
</tr>
<tr>
<td>visible</td>
<td></td>
<td></td>
<td>(2^{(n+2)/2} - 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(2^{(n-2)/2})</td>
</tr>
</tbody>
</table>

Table 4.1 Number of simultaneous users. (worst case)
It is noted that number of simultaneous users with given level of interference are found by taking the worst case condition. For the worst case condition, number of simultaneous users with given level of interference is

$$\text{Integer value of } \frac{\text{peak magnitude of autocorrelation}}{\text{peak magnitude of the worst case crosscorrelation}}$$

For our problem, under consideration, that is with five shift register stages we have,

- Peak of autocorrelation = \(N = 2^n - 1 = 31\)
- The worst case crosscorrelation peak = 9

Hence, number of simultaneous users with given level of interference is

$$\text{Integer value of } \left[\frac{31}{9}\right] = 3$$

Now as seen from above, that we have peak of autocorrelation equals \(N = 2^n - 1\) and for the worst case condition the number of simultaneous users are \(\frac{2^{(n-1)/2} - 1}{2}\). Then the interference by these users with the worst case crosscorrelation peak of \(\frac{2^{(n+1)/2} + 1}{2}\) will be \(\frac{2^{(n-1)/2} - 1}{2} \cdot \frac{2^{(n+1)/2} + 1}{2}\). It can be seen here that, even with these number of simultaneous users, the margin between the autocorrelation peak and the peak value of interference will be very small, so it becomes very difficult to acquire the signal.
Example: Consider the case of \( n = 5 \) as an example of the computation. As we saw above, we can have 3 simultaneous users with the worst case crosscorrelation peak of 9. Hence, the total interference = \( 9 \times 3 = 27 \). That means that 27 units out of 31 units will be used by interferers and it will be difficult to acquire the signal. Therefore, for a practical case, fewer than 3 users can operate simultaneously to avoid the difficulty of acquisition.

4.2 Average case

Average case condition means considering the average value of crosscorrelation magnitudes taking into account the frequency of occurrence of these magnitudes. The general result for both \( n \) odd and \( n \) even (not divisible by 4) is tabulated in Table 4.2. The first column identifies the number of shift register stages. The next column shows the magnitude of crosscorrelation function. The third column contains the frequency of occurrence of crosscorrelation magnitude. The fourth column includes the average value for the magnitude of crosscorrelation function. The final column gives the number of simultaneous users.
Now the number of simultaneous users are found by taking average case condition as shown below. For average case, the number of simultaneous users with given level of interference is

\[
|\Theta_{xy}| = 0.5 \left(2^{(n+1)/2} + 1\right) \quad 2^{(n+1)/2 - 2}
\]

odd \quad \frac{2^{(n+1)}}{2} + 1 = 0.25
\quad \frac{2^{(n+1)}}{2} - 1 = 0.25

even(1) \quad \frac{2^{(n+1)}}{2} + 1 = 0.75 \quad \frac{2^{(n+2)}}{2} \quad 2^{(n+2)/2 - 3}
\quad \frac{2^{(n+2)}}{2} - 1 = 0.125

Table 4.2 Number of simultaneous users. (average case)

Now the number of simultaneous users are found by taking average case condition as shown below. For average case, the number of simultaneous users with given level of interference is

\[
\text{Integer value of} \quad \frac{\text{peak value of autocorrelation}}{\text{average value of crosscorrelation magnitudes}}
\]

For the problem with five shift register stages which we have been considering, \( n = 5 \) and the code length = autocorrelation peak = 31. And from Table 4.2 the average value of crosscorrelation magnitude is

70
Then the number of simultaneous users is

\[
\text{Integer value of } (31/4.5) = 6
\]

Also using the formula shown in Table 4.2, the number of simultaneous users is

\[
2^{(n+1)/2} - 2 = 2^{(5+1)/2} - 2 = 3 - 2 = 6
\]

Now the average margin between the peak autocorrelation and the crosscorrelation with the interference would be

\[
\{.5(2^{(n+1)/2} + 1)\}(2^{(n+1)/2} - 2) \quad \text{for } n \text{ odd}
\]

and \[
[.75 + .25(2^{(n+1)/2})][2^{(n+1)/2} - 3] \quad \text{for } n \text{ even not divisible by 4}
\]

Therefore, for the problem we have been considering, the average margin would be only

\[
(31 - (4.5)(6)) = 4
\]
5.1 Analysis of the CDMA networks

The most commonly used quantity in CDMA systems is that of "process gain", although it must be pointed out that what is usually intended is not process gain but "jamming margin". The processing gain achieved using large spreading chip/data bit ratios could efficiently be utilized by transmitting many signals simultaneously on the same carrier frequency and applying code division multiplexing. Successful use of spread spectrum CDMA techniques requires the construction of spreading codes giving rise to a minimum of interlink interference. However, situations arise where the effects of interlink interference are amplified owing to operational considerations. Consider a network operated in a master-slave configuration. Let slave station $M_1$ transmit the desired signal $S$ to the master, which is $d_1$ km away, and let another slave station $M_2$ at a distance $d_2$ km from the master transmit a signal $I$ which is interference to the $M_1$-master link. Let $C(dB)$ be the signal/interference ratio ($S/I$) required at the terminals of the master station's receiving input to produce the desired output $S/I$. The requirement is that
With transmitting powers of $P_{M1}$ and $P_{M2}$ and path losses of $L_1$ and $L_2$ respectively, equation 5.1 after rearranging yields

\[ (P_{M1} - L_1) - (P_{M2} - L_2) \geq C \quad (5.2a) \]

\[ L_2 \geq C + (P_{M2} - P_{M2}) + L_1 \quad (5.2b) \]

Equation 5.2b is an explicit formula for the near-far problem and will put a restriction on where slave station $M_2$ can operate if the original system specification is to be met. If the transmitter powers remain constant while the value of $L_2$ is reduced, the inequality in equation 5.2a may be violated for any practical code design. The value chosen for the factor $C$ will however depend on how well the codes can be designed. In CDMA networks all members communicate on the same frequency. Usually there is also a common data rate $R_d = 1/T_d$ and a common spreading code chip rate $R_c = 1/T_c$. In burst transmission networks no overall timing reference to enable chip synchronization is usually achievable. In portable systems a probe signal from a possible time master would not be a sufficiently accurate time reference, since one would lack compensation for the different delays in the various transmission paths. Thus it is unlikely that the spreading codes would be chip synchronized.
The receiver will attempt to extract the individual spreading codes from a composite of many during the matched filter processing. Crosscorrelation functions play key roles in calculating system performance for such situations. The wideband input signal consists of the wanted signal as well as the interfering signals, each spread by their own code. The spreading code modulating the wanted signal will match the receiver filter and the correlation peak will be sampled at a rate equal to the data rate. If the unwanted signals were totally uncorrelated with the wanted signal, then they would produce no correlation peaks at the filter output. However, the effect of crosscorrelation between the local sequence and the sequences of the unwanted signals appears as crosscorrelation peaks at the output of the filter.
5.2 Simulation

5.2.1 Theoretical bound

For the comparison with the simulated results, we calculate the probability, that the crosscorrelation with interfering users would exceed the threshold value 15 as a function of the number simultaneous communication stations, for the system in our example.

Table 5.1 shows the probability for 2 to 6 simultaneous users. The first column identifies the number of simultaneous users. The next column shows the number of ways to exceed the threshold value of 15. The third column contains the number of possible combinations. The fourth column gives probability of exceeding the threshold value. The final column includes total probability of exceeding the threshold value.
<table>
<thead>
<tr>
<th>SU</th>
<th>$\theta(r) &gt; 15$</th>
<th>Number</th>
<th>Probability</th>
<th>Total Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>9 9</td>
<td>1</td>
<td>(.25)$^2$</td>
<td>.1875</td>
</tr>
<tr>
<td></td>
<td>9 7</td>
<td>2</td>
<td>2(.25)$^2$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>9 9 9</td>
<td>1</td>
<td>(.25)$^3$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 9 7</td>
<td>3</td>
<td>3(.25)$^3$</td>
<td>.40625</td>
</tr>
<tr>
<td></td>
<td>9 7 7</td>
<td>3</td>
<td>3(.25)$^3$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 9 1</td>
<td>3</td>
<td>3(.25)$^2$(.5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 7 1</td>
<td>6</td>
<td>6(.25)$^2$(.5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 7 7</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>9 9 9 9</td>
<td>1</td>
<td>(.25)$^4$</td>
<td>.6875</td>
</tr>
<tr>
<td></td>
<td>9 9 9 7</td>
<td>4</td>
<td>4(.25)$^4$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 9 7 7</td>
<td>6</td>
<td>6(.25)$^4$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 7 7 7</td>
<td>4</td>
<td>4(.25)$^4$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 7 7 7</td>
<td>1</td>
<td>(.25)$^4$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 9 9 1</td>
<td>4</td>
<td>4(.25)$^3$(.5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 9 1 1</td>
<td>6</td>
<td>6(.25)$^2$(.5)$^2$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 9 7 1</td>
<td>12</td>
<td>12(.25)$^3$(.5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 7 7 1</td>
<td>12</td>
<td>12(.25)$^3$(.5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 7 7 1</td>
<td>4</td>
<td>4(.25)$^3$(.5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 7 1 1</td>
<td>6</td>
<td>6(.25)$^2$(.5)$^2$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 7 1 1</td>
<td>12</td>
<td>12(.25)$^2$(.5)$^2$</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>9 9 9 9 9</td>
<td>1</td>
<td>(.25)$^5$</td>
<td>.8125</td>
</tr>
<tr>
<td></td>
<td>9 9 9 9 7</td>
<td>5</td>
<td>5(.25)$^5$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 9 9 7 7</td>
<td>10</td>
<td>10(.25)$^5$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 9 7 7 7</td>
<td>10</td>
<td>10(.25)$^5$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 7 7 7 7</td>
<td>5</td>
<td>5(.25)$^5$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 9 9 9 1</td>
<td>5</td>
<td>5(.25)$^4$(.5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 9 9 1 1</td>
<td>10</td>
<td>10(.25)$^3$(.5)$^2$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 9 1 1 1</td>
<td>10</td>
<td>10(.25)$^2$(.5)$^3$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 9 9 7 1</td>
<td>20</td>
<td>20(.25)$^4$(.5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 9 7 7 1</td>
<td>30</td>
<td>30(.25)$^4$(.5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 7 7 7 1</td>
<td>20</td>
<td>20(.25)$^4$(.5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 7 7 7 1</td>
<td>5</td>
<td>5(.25)$^5$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 9 7 1 1</td>
<td>30</td>
<td>30(.25)$^3$(.5)$^2$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 7 7 1 1</td>
<td>30</td>
<td>30(.25)$^3$(.5)$^2$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 7 7 1 1</td>
<td>10</td>
<td>10(.25)$^3$ (.5)$^2$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 7 1 1 1</td>
<td>10</td>
<td>10(.25)$^2$ (.5)$^3$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 7 1 1 1</td>
<td>20</td>
<td>20(.25)$^2$ (.5)$^3$</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1 The probability exceeding threshold value 15

76
Table 5.1 (Continued)

5.2.2 Flow diagram

A simplified diagram for the calculation of crosscorrelation values in CDMA networks is shown in Figure 5.1

77
Fig. 5.1 Flow diagram for the simulator in CDMA networks.
Fig. 5.1 (Continue)
For the purpose of computer simulation, let us consider a network configuration such as that shown in Figure 5.2 where a number of stations at some time are in simultaneous pair wise communication and are causing mutual interference to each other.

5.2.3 Computer simulation
In this network configuration, we neglect the near-far problem, the sidelobe energy problem, and noise. Specific Gold code assignments are randomly generated from one preferred pair. Table 5.2 shows Gold code sequences used in this simulation. Here the codes expressed in bold character are our local Gold code sequences.
<table>
<thead>
<tr>
<th>Preferred pair [5,2][5,4,3,2]</th>
<th>Balanced Gold code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1101111110110010001000000111</td>
</tr>
<tr>
<td></td>
<td>10110101100100111101100110110</td>
</tr>
<tr>
<td></td>
<td>0110000111111110101001010100</td>
</tr>
<tr>
<td></td>
<td>00111011011100000101110011110</td>
</tr>
<tr>
<td></td>
<td>11010100111001101011011001001</td>
</tr>
<tr>
<td></td>
<td>010011001011100111011001110100</td>
</tr>
<tr>
<td></td>
<td>001011011101011001111110001000110</td>
</tr>
<tr>
<td></td>
<td>0011010101010111101111111110011</td>
</tr>
<tr>
<td></td>
<td>01101101010101111011110101111</td>
</tr>
<tr>
<td></td>
<td>110010010001001001101100100001</td>
</tr>
<tr>
<td></td>
<td>10011000111011100001000011001010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Preferred pair [5,2][5,4,2,1]</th>
<th>Balanced Gold code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>001000000110011100011110011001111</td>
</tr>
<tr>
<td></td>
<td>010010100010010101111111101111</td>
</tr>
<tr>
<td></td>
<td>10011110101010101100110001111</td>
</tr>
<tr>
<td></td>
<td>1100001110111001101110011001111</td>
</tr>
<tr>
<td></td>
<td>01111111100000001110101101011</td>
</tr>
<tr>
<td></td>
<td>11110011001101010101010110001110</td>
</tr>
<tr>
<td></td>
<td>101000011101110000101101110111100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Preferred pair [5,4,3,2][5,4,2,1]</th>
<th>Balanced Gold code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>001000000110111001110110011001111</td>
</tr>
<tr>
<td></td>
<td>01100101000010111001111111001111</td>
</tr>
<tr>
<td></td>
<td>11000000101011011111000011110101</td>
</tr>
<tr>
<td></td>
<td>100010011110010000111110111100</td>
</tr>
<tr>
<td></td>
<td>001101001100110010101101010101</td>
</tr>
<tr>
<td></td>
<td>1111101000111100000011110111001</td>
</tr>
<tr>
<td></td>
<td>1001001101101010100011011111011</td>
</tr>
<tr>
<td></td>
<td>0000001000010001110111101100111</td>
</tr>
<tr>
<td></td>
<td>0001111110000001100110011001101</td>
</tr>
<tr>
<td></td>
<td>0001010111010001000111000000110</td>
</tr>
<tr>
<td></td>
<td>01001011010111000001000100001</td>
</tr>
<tr>
<td></td>
<td>100111100011111101111111011001</td>
</tr>
<tr>
<td></td>
<td>001101101010100001001100010010</td>
</tr>
<tr>
<td></td>
<td>000110001001100110011001100010</td>
</tr>
</tbody>
</table>

Table 5.2 Gold codes used in simulator
The stations are using the Gold code of period 31 and a configuration can be obtained by specifying the number of stations. For such a situation maximum the autocorrelation value is 31.

First, we calculate crosscorrelation value between the desired code and accumulated interferenced code for each network configuration. Here, we examine 961 crosscorrelation values for 2 simultaneous communication stations and 29,791 crosscorrelation values for more than 2 simultaneous communication stations.

Next, we calculate the probability that the crosscorrelation is greater than the threshold value of 15 for 6 network configurations. Here the threshold value is defined such that for value less than this value, the link operates normally; whereas for value greater than this threshold, the link fails. The procedure has been to study each configuration for a value of crosscorrelation, thus providing a statistical estimate of the network failure. Notice that we examine the probability for balanced and unbalanced Gold codes of each preferred pair such as that shown in Table 5.3.
<table>
<thead>
<tr>
<th>Simultaneous number of users</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preferred pair [52,5432]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balanced</td>
<td>0.0208</td>
<td>0.0584</td>
<td>0.1503</td>
<td>0.1968</td>
<td>0.2645</td>
</tr>
<tr>
<td>Unbalanced</td>
<td>0.0104</td>
<td>0.0692</td>
<td>0.1951</td>
<td>0.1595</td>
<td>0.3370</td>
</tr>
<tr>
<td>Preferred pair [52,5421]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balanced</td>
<td>0.0219</td>
<td>0.0906</td>
<td>0.1572</td>
<td>0.1306</td>
<td>0.1726</td>
</tr>
<tr>
<td>Unbalanced</td>
<td>0.0229</td>
<td>0.0887</td>
<td>0.1271</td>
<td>0.1110</td>
<td>0.2091</td>
</tr>
<tr>
<td>Preferred pair [5432,5421]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balanced</td>
<td>0.0312</td>
<td>0.0890</td>
<td>0.1446</td>
<td>0.1930</td>
<td>0.2293</td>
</tr>
<tr>
<td>Unbalanced</td>
<td>0.0062</td>
<td>0.0799</td>
<td>0.1308</td>
<td>0.2080</td>
<td>0.2855</td>
</tr>
</tbody>
</table>

Table 5.3 Simulation results (probability).

<table>
<thead>
<tr>
<th>Simultaneous number of users</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preferred pair [52,5432]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balanced</td>
<td>5.9</td>
<td>6.6</td>
<td>8.3</td>
<td>9.1</td>
<td>10.3</td>
</tr>
<tr>
<td>Unbalanced</td>
<td>5.9</td>
<td>6.8</td>
<td>9.3</td>
<td>8.6</td>
<td>12.3</td>
</tr>
<tr>
<td>Preferred pair [52,5421]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balanced</td>
<td>6.2</td>
<td>7.2</td>
<td>8.2</td>
<td>7.9</td>
<td>8.6</td>
</tr>
<tr>
<td>Unbalanced</td>
<td>6.4</td>
<td>7.0</td>
<td>8.0</td>
<td>7.8</td>
<td>8.6</td>
</tr>
<tr>
<td>Preferred pair [5432,5421]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balanced</td>
<td>6.4</td>
<td>7.3</td>
<td>8.2</td>
<td>8.7</td>
<td>9.1</td>
</tr>
<tr>
<td>Unbalanced</td>
<td>6.4</td>
<td>7.4</td>
<td>7.6</td>
<td>8.8</td>
<td>9.7</td>
</tr>
</tbody>
</table>

Table 5.4 Simulation results (ave. value of crosscorrel.).
Finally, we calculated average value of the crosscorrelations shown in Table 5.4. Figure 5.3, 5.4, 5.5, and 5.6 give theoretical bound compared with simulation results.
Fig. 5.3 Num. of Simul. Users vs Probability

Unbalanced Gold code

Probability exceeding threshold value 1.5

Number of Simultaneous Users

Symbols:
- [52, 5432]
- + [52, 5421]
- * [5432, 5421]
- △ The.
Fig. 5.4 Num. of Simul. Users vs Probability

Balanced Gold code

Prob. exceeding threshold value 1.5

Number of Simultaneous Users

- [52,5432]
- [52,5421]
- [5432,5421]

△ The.
Fig. 5.5 Num. of Users vs Ave. X—corr. Value

Unbalanced Gold code

Number of simultaneous users

\[ [52,5432] \quad + \quad [52,5421] \quad \diamond \quad [5432,5421] \quad \Delta \quad \text{The.} \]
Fig. 5.6 Num. of Users vs Ave. X–correl. Value

Balanced Gold code

Average X-correlation values

Number of simultaneous users

- [52,5432]
- [52,5421]
- [5432,5421]
- The.
5.3 Discussion of the results of the simulations

The simulation was performed for comparison between the theoretical bound and simulated results in each CDMA network configuration. The operation of a CDMA network configuration using a Gold code generated from the given preferred pair was simulated. The balanced Gold code and unbalanced Gold code given by each preferred pair were investigated for possible variations in the performance results.

When the performance results of the simulation in CDMA networks are compared with the theoretical bound, the simulation results were found to deviate by a factor of as much as 3 from the theoretical bound such as shown in Figure 5.3 to 5.6 because the theoretical bound is based on the sum of the magnitudes of the crosscorrelation, while the actual result depends on the magnitude of sum of the crosscorrelation.

Example: Let us consider the following simple codes. Here C code is our local code.

A = -1 -1 1
B = 1 -1 1
C = 1 1 1

The crosscorrelation magnitudes between code A and C and between code B and C are always 1. Hence average
crosscorrelation value by theoretical calculation in CDMA is always 2. But actual calculation gives the crosscorrelation magnitudes of 0, 0, and 3. Therefore, average crosscorrelation value is 1.

We checked simulation results by using sampled different Gold code from the same preferred pair. However, it is found that the results are not much different. In those figures, we can notice that probability and average code length are different for every preferred pair of given simultaneous users. This means that we can decrease interference if we use optimal code in given simultaneous users.

This simulation results are the part of calculation for given Gold code which is greater than 4 simultaneous users. We also used the approximations in calculating the theoretical bound. Through the supercomputer, whole calculation is done by using exact probabilities for all possible Gold code which remains for future study. An analysis of the statistical behavior of the noise corrupted Gold code and the analysis of CDMA networks considering the near-far problem is also recommended for future research in this area.
CHAPTER VI
CONCLUSION

We started our investigation with the correlation property of binary sequences. We have presented the autocorrelation and crosscorrelation properties for periodic and aperiodic sequences. In addition, we have included the correlation properties of the Gold codes.

We then discussed Gold code generation for the balanced and unbalanced Gold codes.

Thirdly, we investigated the number of simultaneous users in a CDMA system using Gold codes for the worst case and the average case of mutual interference.

Finally, we compared simulated results with the theoretical bound. There we found that the simulation results deviate by a large factor from the theoretical bound because of the approximations made in calculating the theoretical bound. We recommend the analysis of the statistical behavior of the noise corrupted Gold code and the analysis of CDMA networks considering the near-far problem for future research in this area.
This program calculate crosscorrelation value, its average value, probability of crosscorrelation value exceeding the threshold value from the Gold codes used in CDMA networks. The Gold codes used in this program is given by Gold code generation program in Appendix B.

The program given here is good for all period of Gold codes, but it can read 4 codes. By adding input array and loop, we can expand for the increased codes. But it takes a long time to calculate more than $N^4$ of crosscorrelation values for even $N = 31$. This simulation program is done in C computer language.
```c
#include <stdio.h>
#define max 500
int A[max], B[max], C[max], D[max], E[max], F[max],
     X[max], Y[max], Z[max], W[max],
h,i,j,n,m,temp,templ,Acode,Bcode,Ccode,Dcode,sign,
done,k;
float ave,sum4,sum3,sum2,sum1,sum,nb,prob,ave,prob;
main()
{
    scanf("%d",&Acode);
    n = 1;
    while (Acode != 20)
    {
        A[n] = Acode;
        X[n] = Acode;
        scanf("%d",&Acode);
        n++;
    }
    sign = 0;
    for (i=1; i<=n-1; i++)
    {
        if (abs(A[i])<=1)
        {
            if (A[i]==0)
                A[i] = -1;
            else
                sign = 1;
        }
        else
            sign = 1;
    }
    if (sign)
    {
        for (i=1; i<=n-1; i++)
            A[i] = X[i];
    }
    n =1;
    scanf("%d",&Bcode);
    while (Bcode != 20)
    {
        B[n] = Bcode;
        Y[n] = Bcode;
        scanf("%d",&Bcode);
        n++;
    }
    sign = 0;
    for (i=1; i<=n-1; i++)
    {
        if (abs(B[i])<=1)
```
{ if (B[i]==0) 
    B[i] = -1;
} else 
    sign = 1;
}
if (sign)
{
    for (i=1; i<=n-1; i++)
        B[i] = Y[i];
}
n = 1;
scanf("%d", &Ccode);
while (Ccode != 20)
{
    C[n] = Ccode;
    Z[n] = Ccode;
    scanf("%d", &Ccode);
    n++;
} sign = 0;
for (i=1; i<=n-1; i++)
{
    if (abs(C[i])<=1)
    {
        if (C[i] == 0)
            C[i] = -1;
    } else 
        sign = 1;
} if (sign)
{
    for (i=1; i<=n-1; i++)
        C[i] = Z[i];
}
n = 1;
scanf ("%d", &Dcode);
while (Dcode != 20)
{
    D[n] = Dcode;
    W[n] = Dcode;
    scanf("%d", &Dcode);
    n++;
} sign = 0;
for (i=1; i<=n-1; i++)
{ }
if (abs(D[i]) <= 1)
{
    if (D[i] == 0)
        D[i] = -1;
} else
    sign = 1;
if (sign)
{
    for (i=1; i <= n-1; i++)
        D[i] = W[i];
} m = 1; n = n - 1; sum4 = 0.0;
nb = 0.0;
while (m <= n)
{
    h = 1; sum3 = 0.0;
    while (h <= n)
    {
        sum2 = 0.0;
        for (i=1; i <= n; i++)
        {
            E[i] = A[i] + B[i] + C[i];
        }
        for (j=1; j <= n; j++)
        {
            sum = 0.0;
            for (i=1; i <= n; i++)
            {
                F[i] = D[i] * E[i];
                sum = sum + F[i];
            }
            if (j % 10 != 0)
                printf("%.1f	", sum);
            else
            {
                printf("%.1f	", sum);
                printf("\n");
            }
            if (sum < 0)
            {
                sum1 = -sum;
                sum2 = sum2 + sum1;
            } else { sum1 = sum;
                sum2 = sum2 + sum1;
            }
        }
    }
}
if (sum1 >= 16)
96
++nb;
templ = E[1];
for (k=1; k<=n-1; k++)
{
    E[k] = E[k+1];
}
E[n] = templ;

sum3 = sum2 + sum3;
ave = sum2/n;
printf(" Ave = %5.1f\n",ave);
printf("\n");
temp = C[1];
for (j=1; j<=n-1; j++)
{
    C[j] = C[j+1];
}
C[n] = temp;
h++;
printf(" sum2 = %6.1f\n",sum2);
printf(" sum3 = %6.1f\n",sum3);
}
sum4 = sum3 + sum4;
ave = sum3/(n*n);
prob = nb/(n*n);
printf(" Average = %5.1f\n",ave);
printf(" Number = %5.0f\n", nb);
printf(" prob = %8.4f\n", prob);
temp = B[1];
for (i=1; i<=n-1; i++)
{
    B[i] = B[i+1];
}
B[n] = temp;
m++;

printf(" Sum4 = %10.0f\n", sum4);
ave = sum4/(n*n*n);
prob = nb/(n*n*n);
printf(" Average = %5.1f\n", ave);
printf(" Number = %7.0f\n", nb);
printf(" prob = %8.4f\n", prob);
}
This program generates Gold codes from the m-sequences for preferred pair. The program given here is good for all period of Gold codes if we input the Gold code sequences given from the preferred pair. This program is done in C computer language.
```c
#include<stdio.h>
define max 500
int A[max], B[max],
    C[max],ans,i,j,n,m,temp,Acod,Bcod,done,sum,nl;
main()
{
    done = 0;
    scanf("%d", &Acod);
    n = 1;
    while (Acod != 20)
    {
        A[n] = Acod;
        scanf("%d", &Acod);
        n++;
    }
    m = 1; n = 1;
    scanf("%d", &Bcod);
    while (Bcod != 20)
    {
        B[n] = Bcod;
        scanf("%d", &Bcod);
        n++;
    }
    n = n - 1;
    while (m <= n)
    {
        nl = 0;
        for (i=1; i<=n; i++)
        {
            C[i] = A[i]^B[i];
            if (C[i]==1)
                ++nl;
            printf("%2d", C[i]);
        }
        printf(" 20\n");
        printf(" Num. 1 = %d", nl);
        printf("\n");
        temp = B[1];
        for (j=1; j<=n-1; j++)
        {
            B[j] = B[j+1];
        }
        B[n] = temp;
        m++;
    }
}
```
In order to relate the crosscorrelation of sequences to their autocorrelation, one might apply the following theorem due to Gold[2]:

\[ \sum_{l=0}^{N-1} [\alpha_k r(l)]^2 = \sum_{l=0}^{N-1} \left( \sum_{l=0}^{N-1} \alpha_k(i+1) \alpha_r(i) \right)^2 \]

from the definition of the crosscorrelation \( \alpha_k r(l) \). The right-hand side can then be developed to give

\[
= \sum_{l=0}^{N-1} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \alpha_k(i+1) \alpha_r(i) \alpha_k(j+1) \alpha_r(j) \\
= \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \alpha_r(i) \alpha_r(j) \sum_{l=0}^{N-1} \alpha_k(i+1) \alpha_k(j+1) \\
= \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \alpha_r(i) \alpha_r(j) \alpha_k(j-1) \\
= \sum_{i=0}^{N-1} \sum_{l=0}^{N-1} \alpha_r(i) \alpha_r(l+i) \alpha_k(l) \\
= \sum_{l=0}^{N-1} [\alpha_r(l)][\alpha_k(l)] \\
\]

From this it is possible to develop the result[3]
\[ \sum_{l=1-N}^{N-1} [c_{kr}(1)]^2 = \sum_{l=1-N}^{N-1} [c_k(1)][c_r(1)] = N^2 + 2 \sum_{l=1}^{N-1} [c_k(1)c_r(1)] \]

(A.2)

for sequences where \(c_k(0) = c_r(0) = N\).

Then
\[
[c_{kr}(1-N)]^2 + [c_{kr}(1-N+1)]^2 + [c_{kr}(1)]^2 + [c_{kr}(1+1)]^2
\]
\[= 2 \sum_{l=1-N}^{N-1} [c_{kr}(1)]^2 \]
\[= 2N^2 + 4 \sum_{l=0}^{N-1} [c_k(1)][c_r(1)] \]

(A.3)
CROSSCORRELATION PARAMETERS FOR MAXIMAL LENGTH SEQUENCES

Let \( a^k \) be a maximal length sequence with period \( N = 2^m - 1 \) and \( a^r \) another \( m \)-sequence with the same period. Both sequences thus have an autocorrelation given by

\[
\varnothing^k(1) = 1 \quad l = 0 \mod m
\]
\[
\varnothing^r(1) = 1 \quad l = 0 \mod m
\]
\[
\varnothing^k(1) = -1/N \quad l = 0 \mod m
\]
\[
\varnothing^r(1) = -1/N \quad l = 0 \mod m
\]

The crosscorrelation between \( a^k \) and \( a^r \) is given by

\[
\varnothing_{kr}(1) = \frac{1}{N} \sum_{s=1}^{N} a^k_s a^r_{s+1}
\]

The mean \( \varnothing_{kr} \) of the crosscorrelation for all shifts is given by

\[
\overline{\varnothing_{kr}} = \frac{1}{N} \sum_{l=1}^{N} \varnothing_{kr}(1) = \frac{1}{N^2} \sum_{l=1}^{N} \sum_{s=1}^{N} a^k_s a^r_{s+1} = \frac{(-1)(-1)}{N^2} = \frac{1}{N^2}
\]

The variance \( 6^2 \) of the crosscorrelation for all shifts is given by

\[
6^2 = \frac{1}{N} \sum_{l=1}^{N} [\varnothing_{kr}(1)]^2 - (\varnothing_{kr})^2
\]
This is because $\Theta^r(1)$ has period $N$ and one peak during the period

$$\theta^2 = \frac{1}{N^2} \frac{N - (-1)}{N^2} \sum_{s=1}^{N} a_k s a_k t + \frac{1}{N^2} \sum_{s=1}^{N} \sum_{t=1}^{N} a_k s a_t - \frac{1}{N^4}$$

Summing with respect to $s$:

$$\theta^2 = \frac{1}{N} \frac{N - (-1)}{N} + \frac{1}{N^2} \frac{(-1)}{N} - \frac{1}{N^4} = \frac{N + 1}{N^2} - \frac{1}{N^3} - \frac{1}{N^4}$$

For $N >> 1$,

$$\theta^2 = \frac{1}{N}$$
APPENDIX E

BOUNDS ON APERIODIC AUTOCORRELATION AND CROSSCORRELATION OF SEQUENCES

From the theorem of Gold developed in Appendix C, it is also possible to show that the odd correlation parameters are related through

\[ \sum_{k=0}^{N-1} [\hat{\theta}^kr(1)]^2 = \sum_{l=0}^{N-1} [\hat{\theta}^k(l)][\hat{\theta}^r(l)] \quad (E.1) \]

Applying this to all members of a set \( A \) consisting of \( K \) sequences of period \( N \) where the inphase autocorrelation \( \theta^k(0) = N \) for all \( k \notin A \), one obtains

\[
\sum_{k \in A} \sum_{r \in A} \sum_{l=0}^{N-1} [\hat{\theta}^kr(1)]^2 = \sum_{k \in A} \sum_{r \in A} \sum_{l=0}^{N-1} [\hat{\theta}^k(1)]^2 + \sum_{k \notin A} \sum_{l=0}^{N-1} [\hat{\theta}^k(l)]^2
\]

Thus from (E.1):

\[
\sum_{k \in A} \sum_{r \notin A} \sum_{l=0}^{N-1} [\hat{\theta}^kr(1)]^2 + \sum_{k \in A} \sum_{l=0}^{N-1} [\hat{\theta}^k(1)]^2 = \sum_{k \notin A} \sum_{r \in A} \sum_{l=1}^{N-1} [\hat{\theta}^k(1)][\hat{\theta}^r(l)]
\]

\[
= \sum_{l=0}^{N-1} [\sum_{k \in A} \hat{\theta}^k(l)]^2
\]

\[
= K^2N^2 + \sum_{l=1}^{N-1} [\sum_{k \in A} \hat{\theta}^k(l)]^2
\]

The left side of this equation is upper bound by
\[ K(K - 1)N\hat{\sigma}_r^{kr(1)} + KN^2 + K(N-1)\hat{\sigma}_r^{k\theta(1)}^2 \quad 0 \leq \theta \leq N-1 \]

The right-hand side is lower bounded by \(K^2N^2\):

\[ K(K - 1)N\hat{\sigma}_r^{kr(1)} + KN^2 + K(N - 1)\hat{\sigma}_r^{k\theta(1)}^2 > K^2N^2 \]

\[ [\hat{\sigma}_r^{kr(1)}]^2K(K - 1)N + [\hat{\sigma}_r^{k\theta(1)}]^2K(N - 1) > N^2K(K - 1) \]

\[ \frac{[\hat{\sigma}_r^{kr(1)}]^2}{N} + \frac{[\hat{\sigma}_r^{k\theta(1)}]^2}{N} \quad \frac{N - 1}{N(K - 1)} > 1 \]
## APPENDIX F

### FEEDBACK CONNECTIONS FOR LINEAR M-SEQUENCES

<table>
<thead>
<tr>
<th>Number of Stages</th>
<th>Code Length</th>
<th>Maximal Taps</th>
</tr>
</thead>
<tbody>
<tr>
<td>2&lt;sup&gt;a&lt;/sup&gt;</td>
<td>3</td>
<td>[2, 1]</td>
</tr>
<tr>
<td>3&lt;sup&gt;a&lt;/sup&gt;</td>
<td>7</td>
<td>[3, 1]</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>[4, 1]</td>
</tr>
<tr>
<td>5&lt;sup&gt;a&lt;/sup&gt;</td>
<td>31</td>
<td>[5, 2], [5, 4, 3, 2], [5, 4, 2, 1]</td>
</tr>
<tr>
<td>6</td>
<td>63</td>
<td>[6, 1], [6, 5, 2, 1], [6, 5, 3, 2]</td>
</tr>
<tr>
<td>7&lt;sup&gt;a&lt;/sup&gt;</td>
<td>127</td>
<td>[7, 1], [7, 3], [7, 3, 2, 1], [7, 4, 3, 2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[7, 6, 4, 2], [7, 6, 3, 1], [7, 6, 5, 2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[7, 6, 5, 4, 2, 1], [7, 5, 4, 3, 2, 1]</td>
</tr>
<tr>
<td>8</td>
<td>255</td>
<td>[8, 4, 3, 2], [8, 6, 5, 3], [8, 6, 5, 2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[8, 5, 3, 1], [8, 6, 5, 1], [8, 7, 6, 1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[8, 7, 6, 5, 2, 1], [8, 6, 4, 3, 2, 1]</td>
</tr>
<tr>
<td>9</td>
<td>511</td>
<td>[9, 4], [9, 6, 4, 3], [9, 8, 5, 4], [9, 8, 4, 1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[9, 5, 3, 2], [9, 8, 6, 5], [9, 8, 7, 2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[9, 6, 5, 4, 2, 1], [9, 7, 6, 4, 3, 1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[9, 8, 7, 6, 5, 3]</td>
</tr>
<tr>
<td>10</td>
<td>1023</td>
<td>[10, 3], [10, 8, 3, 2], [10, 4, 3, 1], [10, 8, 5, 1]</td>
</tr>
<tr>
<td>11</td>
<td>2047</td>
<td>[11, 1], [11, 8, 5, 2], [11, 7, 3, 2], [11, 5, 3, 5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[11, 10, 3, 2], [11, 6, 5, 1], [11, 5, 3, 1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[11, 9, 4, 1], [11, 8, 6, 2], [11, 9, 8, 3]</td>
</tr>
<tr>
<td>12</td>
<td>4095</td>
<td>[12, 6, 4, 1], [12, 9, 3, 2], [12, 11, 10, 5, 2, 1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[12, 11, 6, 4, 2, 1], [12, 11, 9, 7, 6, 5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[12, 11, 9, 5, 3, 1], [12, 11, 9, 8, 7, 4]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[12, 11, 9, 7, 6, 5], [12, 9, 8, 3, 2, 1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[12, 10, 9, 8, 6, 2]</td>
</tr>
<tr>
<td>13&lt;sup&gt;a&lt;/sup&gt;</td>
<td>8191</td>
<td>[13, 4, 3, 1], [13, 10, 9, 7, 5, 4]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[13, 11, 8, 7, 4, 1], [13, 12, 8, 7, 6, 5]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[13, 9, 8, 7, 5, 1], [13, 12, 6, 5, 4, 3]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[13, 12, 11, 9, 5, 3], [13, 12, 11, 5, 2, 1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[13, 12, 9, 8, 4, 2], [13, 8, 7, 4, 3, 2]</td>
</tr>
<tr>
<td>14</td>
<td>16,383</td>
<td>[14, 12, 2, 1], [14, 13, 4, 2], [14, 13, 11, 9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[14, 10, 6, 1], [14, 11, 6, 1], [14, 12, 11, 1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[14, 6, 4, 2], [14, 11, 9, 6, 5, 2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[14, 13, 6, 5, 3, 1], [14, 13, 12, 8, 4, 1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[14, 8, 7, 6, 4, 2], [14, 10, 6, 5, 4, 1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[14, 13, 12, 7, 6, 3], [14, 13, 11, 10, 8, 3]</td>
</tr>
</tbody>
</table>
15 32,767

[15,13,10,9][15,13,10,1][15,14,9,2]
[15,1][15,9,4,1][15,12,3,1][15,10,5,4]
[15,10,5,4,3,2][15,11,7,6,2,1]
[15,7,6,3,2,1][15,10,9,8,5,3]
[15,12,5,4,3,2][15,10,9,7,5,3]
[15,13,12,10][15,13,10,2][15,12,9,1]
[15,14,12,2][15,13,9,6][15,7,4,1]
[15,4][15,13,7,4]

Mersenne prime length generator.
REFERENCES


8. M. J. E. Golay, "The Merit Factor of Long Low
Autocorrelation binary sequences", IEEE Trans., 1982,
9. W. J. Judge, "Multiplexing Using Quasiorthogonal
10. J. K. Holmes, Coherent Spread-Spectrum Systems, John
12. J. E. McDermott, Radio-Electronics, v58, April 87,
p55(4)
13. R. C. Dixon, Spread Spectrum Techniques, IEEE Press,
15. S. W. Colomb, (ed.), Digital Communications with Space
Spectrum Communications, Computer Science Press, Inc.,
1985(I, II, III).
17. W. W. Peterson, and E. J. Weldon, Jr., Error-
Correcting Codes-Second Edition, Cambridge, MA.,
18. C. E. Cook, F. W. Ellersick, L. B. Milstein, and D.L.


