The Performance of Concatenated Codes in Optical CDMA LANs

By

Ms. Nuanpan Lertphaiboon

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in Telecommunications Science
Assumption University

October, 2003

The Performance of Concatenated Codes in Optical CDMA LANs

By

Ms. Nuanpan Lertphaiboon

Submitted in Partial Fulfillment of the
Requirement for the Degree of
Master of Science in
Telecommunication Science
Assumption University

October , 2003

The Faculty of Science and Technology

Master Thesis Approval

Thesis Title The Performance of Concatenated Codes in Optical CDMA LANs

By Ms. Nuanpan Lertphaiboon
Thesis Advisor Asst. Prof. Dr. Chanintorn J. Nukoon
Academic Year 1/2003

The Department of Telecommunications Science, Faculty of Science and Technology of Assumption University has approved this final report of the **twelve** credits course. **TS7000 Master Thesis**, submitted in partial fulfillment of the requirements for the degree of Master of Science in Telecommunications Science.

Approval Committee:

(Asst.Prof.Dr. Chanintorn J. Nukoon)
Advisor

(Dr. Jirapun Daengdej)
Committee Member

(Asst.Prof.Dr. Dobri Batovski)
Committee Member

(Asst.Prof.Dr. Surapong Auwatanamongkol)
Commission of Higher Education
University Affairs

Faculty Approval:

(Asst.Prof.Dr. Dobri Batovski)
Program Director

(Asst.Prof.Dr. Supavadee Nontakao)
Dean

September / 2003

ACKNOWLEDGEMENTS

I would like to express my gratitude to Asst. Prof. Dr. Chanintorn Jittawiriyanukoon and Dr. Pham Mahn Lam for their valuable advice, excellent suggestions, and constant encouragement throughout my thesis. Due to their support and help, this thesis was completed successfilly.

I also thank our committee members, Asst. Prof. Dobri Atanassov Batovski, PhD. and Dr. Chirapan Daengdej for their kind consideration and suggestions.

Finally, I would like to take this opportunity to express my sincere thanks to my family and my friends for their encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This work investigated the use of Concatenated sequences (Kronecker sequences) in optical fiber CDMA network. These sequences can overcome the loss and short of available sequences. I proposed another form of Kronecker sequences that are generated by balanced m-sequences and balanced gold sequences. In recent years, available kronecker sequences are constructed by Lampel codes and Gold codes. In this thesis, the programmable BER is measure of performance of kronecker sequences. The value of BER are compared between the generating balanced m-sequences as inner and balanced gold sequences as outer and the generating balanced gold sequenced as inner and balanced m-sequences as outer. MATLAB program is used for generating balanced m-sequences and balanced gold sequences. The calculation for auto-correlation and cross correlation of Kronecker sequences is designed to find the numerical result of BER values. The performance of Kronecker codes is evaluated to find the best combination.

Chapter I : Introduction

In a multiple access application, there are many users wishing to use the same channel. In a frequency-division multiple access (FDMA) system, each active user is allocated an individual frequency band for transmission, thus avoiding interference by way of using different frequencies. In a time-division multiple access (TDMA) system, each active user is allocated an individual time slot in each frame to transmit data. Therefore, interference is avoided. The FDMA and TDMA systems are used in conventional analog cellular phone system such as the AMPS (Advanced mobile phone sytem), while the TDMA is used in digital cellular phone systems such as GSM (General system mobile) systems. There is an inherent inefficiency in FDMA and TDMA systems that each user occupies a frequency band or a time slot for the whole duration of the calling. Normally, each party in a conversation speaks less than half of one time (there are periods of silence). Therefore, suring the silent periods, the frequency band or the time slot transmit no data, hence it is wasted. Schemes have been developed to make the silent periods available to other users, but the control of such schemes become complicated.

In a code-division multiple access system, which is also called spead spectrum multiple access (SSMA) system, all the active users share the same frequency band and transmit at the same time.Code Division Multiple Access provides concurrency with appropriate election of user codes. In CDMA users are not in frequency or time synchronism. They are assigned distinct codes with appropriate orthogonality properties and they use the entire frequency band at all times. CDMA techniques fall into four board categories: direct sequence (DS), frequency encoded (FE), time hopping (TH), and frequency hopping (FH). Direct sequence code division multiple-access (DS-CDMA) is the main multiple-access candidate for the third generation personal communication systems. The most important reason for this is the flexibility in supporting different services with different data rates and quality. In this technique, the receiver on the network is assigned a unique "address" sequence that is approximately orthogonal to the sequences assigned to all other receivers. Before the transmitter sent data bits to the target, data bits are replaced by the assigned sequence of the target receiver. To detect incoming data, the targeted receiver correlates with its own "address" sequence.

Recently, optical code-division multiple-access schemes attract much attention particularly in the field of fiber-optic networks, because it allows multiple users to access the network asynchronously and simultaneously. Fully asynchronous operation while providing a large number of address, able to support variable bit rate services as well as burst traffic and offering a natural increase in the security of transmission are a combination of advantages that are attractive from a network perspective. Optical CDMA is also attractive in other points-channel assignment is much easier with CDMA. Thus, optical CDMA is expected in ultra-high-speed LAN's, such as future ultra-high speed and real time computer combination.

1.1 Optical CDMA LANs

There has been considerable interest in applying CDMA techniques developed in the radio domain to optical fiber Local Area Networks (LANs). It can provide a flexible interconnection between a high number of active users in a high bit rate LAN. Practically, a LAN must support a large pool of subscribers, not all of who require access to the network at the same time.

The optical CDMA LANs are based on NxN passive star topology. The passive star topology allows the optical signals from simultaneous users that are transmitted signals on the same carrier frequency. The recived signals from transmitter are sent throgh N input ports to star topology and then they are transmitted to the receiver through N output ports.

Currently, there are many investigations about coherent (field modulated) and non-coherent (intensity modulated) schemes. Coherent optical CDMA systems offer better bit-error rate (BER) performance than their non-coherent optical CDMA counterpart and can accommodate a large number of network users. However, there are many serious problems that make it difficult to implement a coherent optical system. Moreover, the cost for this system is very expensive. Consequently, the non-coherent optical CDMA is more interesting than coherent counterparts.

The non-coherent of optical CDMA systems are based on intensity modulation at the transmitter direct (intensity) detection at the receiver. In direct detection, the photodetector gives the output current which is proportional to the average power of the received optical (modulated) signal. Because only the power level is detected the laser or LED transmitter is intensity modulated (IM). Amplitude shift keying (ASK) is performed on all the optical carrier frequencies at the same time by changing the drive current of the transmitter.

In general, the proposed non-coherent optical CDMA schemes can be separated in two groups. System uses low-weight sequences and high-weight sequences. The low-weight sequence group is based on unipolar-unipolar correlation. The technique for modulation is On-Off Keying (OOK) that is a sequence is transmitted for "1" bit and no signals for "0" bit. The low-weight or sparse codes can group into two groups: non-symmetric codes and symmetric codes. The encoder and decoder based on optical fiber delay lines are used for encoding and decoding signal in this system. The low-weight code system has the advantages of reduction of recombination loss and reducing the complexity of decoder. However, the disadvantage is the limitation of the number of available sequence with good correlation properties.

The other group of high-weight sequences use sequence-inversion keying (SIK) of intensity-modulated sequences. It relies on the unipolar-bipolar correlation at the receiver. The main problem of those systems are high loss a requiring optical switches at very high speed.

1.2 Motivation

It has been show in [11] that the use of concatenated codes (or Kronecker codes) for optical CDMA system can overcome the problem of high loss and the short of available sequences in optical CDMA using low-weight codes [6][7]. The use of Kronecker codes also helps to overcome problems of SIK systems described in [2]: High loss and that described in [3]: Requirement of very high-speed switches and synchronisation. However, in [11] only Lampel codes and Gold codes are used for constructing Kronecker codes.

In this work, I proposed to investigate the use of balanced m-sequence code and balanced Gold code for constructing Kronecker codes and uses the obtained Kronecker codes for optical CDMA network. I will design program (based on MATLAB) for generating the codes and programs for evaluating the Mean Power of Interference as well as program for calculating BER performance. I will find the best combination of inner and outer codes based on BER performance of the obtained codes.

Chapter II: Background and Survey of related work

2.1 Background

CDMA is a form of spread-spectrum, a family of digital communication techniques that have been used in military applications for many years. Spread spectrum is a means of transmission in which the signal occupies a bandwidth in excess of the minimum necessary to send the information [4] ; the band spread is accomplished by means of a code which is independent of the data, and a synchronized reception with the code at the receiver is used for despreading and subsequent data recovery. Some of these techniques are direct-sequence spread spectrum, frequency hopping, and time hopping. As a multiple access method, most interest has been given to direct sequence code division multiple access (DS-CDMA),where spreading is achieved by multiplication of the signal by a psueudo-random spreading sequence.

In direct-sequence spread spectrum, each bit of data is represented by a sequence of coded bits called chip. If code sequence has length L, then each data bit will be represented by L chips. Each sequence of L chips is transmitted in the same time duration as an original data bits; thus the effective transmission rate is increased by a factor of L. The increase in transmission rate has the effect of spreading the signal over a larger frequency band.

2.2 Code Sequences

The code sequences used in CDMA sytems are pseudo-noise (PN) or pseudo-random sequences, viz., sequences of zeroes and ones which resemble a random data pattern. They are generated in a deterministic way and have specific properties.

- Auto-correlation property: Auto-correlation is the measure of similarity between a signal and the time shifted version of itself . The auto-correlation of a code sequence can be obtained by looking at the number of agreements and disagreements. Auto-correlation between the sequence and a shifted version of the sequence is always –1. This property allows ease of synchronization.

<u>Example</u>

If a code sequence $C_1 = \{C_1(i)\} = 1\ 0\ 1\ 0\ 1\ 1\ 0 = C_1(6)$ of length N

$i = 0, 1, 2, \ldots\ldots\ldots N-1$

The Auto-correlation of C_1 is $A_1(s) = \sum_{i=0}^{N-1} C_1(i) * C_1(i \oplus s)$

$*$: Chip-by-chip Multiplication

$\oplus$: Modulo N addition

s : Time-shift can be 0, 1,2,.........,N-1

$A_1(0), A_1(1), A_1(2) \ldots\ldots\ldots, A_1(s)$

$A_1(s)$ is auto-correlation of C_1 at time-shift s

If s = 0 ;

$$A_1(0) = \sum_{i=0} C_1(i) * C_1(i \oplus 0) = \sum_{i=0}^{6} C_1(i) * C_1(i)$$

$$= C_1(0) * C_1(0) + C_1(1) * C_1(1) + \ldots\ldots + C_1(6) * C_1(6)$$

C_1 : 1 0 1 0 1 1 0

C_1 : 1 0 1 0 1 1 0

$A_1(0)$ = 1 0 1 0 1 1 0 = 4

If $s = 1$;

$$A_1(1) = \sum_{i=0}^{7-1} C_1(i) * C_1(i \oplus 1)$$

$$= C_1(0) * C_1(1) + C_1(1) * C_1(2) + \ldots\ldots + C_1(6) * C_1(1)$$

C_1 : 1 0 1 0 1 1 0

C_1 : 1 0 1 0 1 1 0

$A_1(0)$ = 0 0 0 0 1 0 0 = 1

For 1 code sequence, the values of auto-correlation are varied by time-shift.The value of auto-correlation at time shift (s) = 0 is called in-phase or peak auto-correlation. If time shift is inequality 0, the auto-correlation values are called out-of-phase or side lobe of auto-correlation.

The set of code is composed of unipolar (1,0) and bipolar(1,-1). Comparing between peak auto-correlation, an example is shown below.

For unipolar ;

C_1 : 1 0 1 1 0

C_1 : 1 0 1 1 0

$A(0)$ = 1 0 1 1 0 = 3

For bipolar ;

C_1 : 1 -1 1 1 -1

C_1 : 1 -1 1 1 -1

$A(0)$ = 1 1 1 1 1 = 5

From outputs of two codes, the peak auto-correlation of unipolar codes equalizes the weight of codes. Unlike the peak of auto-correlation of bipolar codes, it equalizes the length of the codes.

- Cross-correlation property: Cross-correlation is the measure of similarity between two code words.

Example

If $C_1 = \{C_1(i)\} = 1\ 0\ 1\ 0\ 1\ 1\ 0$; $C_2 = \{C_2(i)\} = 0\ 1\ 0\ 1\ 1\ 1\ 0$

are code sequence of length N ; i = 0, 1, 2,..........N-1

$$A_{1,2}(s) = \sum_{i=0}^{N-1} C_1(i) * C_2(i \oplus s)$$

For s = 0, 1,2,.........,N-1

C_1 : 1 0 1 0 1 1 0

C_2 : 0 1 0 1 1 1 0

$A_{1,2}(0) = 0\ 0\ 0\ 0\ 1\ 1\ 0 = 2$

2.3 Multi-access in Optical Fibers

In optical fibers, alternative techniques to access the huge bandwidth are being explored for high-speed networking applications. There are three multiple access approaches which are often considered to make the system bandwidth available to the individual user [5]: time-division multiple access (TDMA), wavelength-division multiple access (WDMA), and code-division multiple access (CDMA).

TDMA accommodates a large number of active users by interleaving bits from different sources into a period equal to that of the uncompressed bit. However, the performance of TDMA systems is ultimately limited by the time-serial nature of the technology. The time is divided into slots, and users take turns accessing the channel. In the WDMA approach, the available optical bandwidth is divided into distinct wavelength channels that are used concurrently by different users to achieve multiple access. The problem with using WDMA in LANs is that a significant amount of dynamic coordination between nodes is required. Furthermore, in order to utilize the fiber bandwidth most efficiently by WDMA, one needs to employ coherent optical transmission techniques along with heterodyne optical receivers, which are costlier than their non-coherent counterpart. Optical CDMA offers an interesting alternative for LANs because neither time management nor frequency management of all nodes is necessary. Optical CDMA operate without waiting-time delays, and does not suffer packet collisions. In CDMA, all users are allowed to access the entire bandwidth of the channel simultaneously. In order to distinguish individual transmissions in a CDMA system, each user is assigned a unique code, which is able to selectively receive the desired transmission. Only receiver, which is the correct code, may receive a given message.

2.4 Optical CDMA LANs

CDMA was originally investigated in the context of radio frequency (RF). In mid-1980's, researchers started to apply CDMA in optical domain [6-8]. In optical networks, the chips will be unipolar, consisting of +1s and 0s, not bipolar, having +1s and –1 s.

Optical CDMA has several advantages over optical TDMA, e.g., complete utilization of the entire time-frequency domain by each subscriber, flexibility in network design, and security against interception. Synchronous CDMA has an additional advantage over asynchronous CDMA, where the number of available code sequences is much higher in the former under a given throughput constraint. However, in a LAN environment where the traffic is usually burst, an efficient multiple-access protocol that allows users to access the network asynchronously at all times is important. Synchronous CDMA is suitable for very high-speed networks with real time requirements.

A typical fiber optic CDMA system is a passive N x N star coupler that is shown in figure 1. The encoded signal is sent to the N x N star coupler and broadcast to all nodes. In transmission signal, both input and output ports are equal. When transmitted port send signal N ports, the received port send N ports, too.

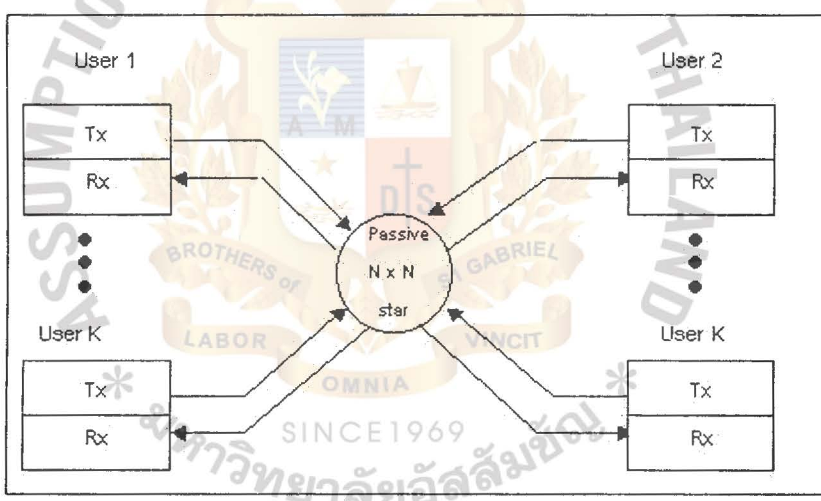


Figure 1 A fiber optic CDMA network using passive N x N star coupler

As usual, all nodes are connected together using a star coupler. In synchronous optical CDMA system must have 1 clock for all network. It is called Master Clock to distribute clock signal to all users of the network. Before optical transmitter sends data, the clock sent λ_1 to laser. Laser that is controlled by code sequences emits light to opto-electrical switch by λ_2. The switch at the transmitter sends encoded sequences to 1 x N optical power splitter. The pulse streams are splitted into N pulse through parallel optical delay lines and gate for detecting codes. After that codes are combined by N x 1 power combiner and sent to passive star coupler. At the receiver, There is wavelength division multiplexing (WDM) filter to generate wavelength of each source. Similar to the transmitter, N optical fiber delay lines are connected in parallel using a 1 x N optical power splitter and a N x 1 optical power combiner. The received signal is correlated to the destination address sequence. Sampling at the last time slot takes a sample and compares to the threshold. The sampling time is determined by the received clock signal. The resulting signal is converted into an electrical current and after amplification it is threshold detected.

Example

If threshold determines $\theta = 4$, the sample has to compare this value below.

Sample $\geq \theta$ then receive bit "1"

Sample $< \theta$ then receive bit "0"

For synchronous network, all transmission (T_x) can only be sent signal at the *starting of a bit times T_b, $2\ T_b$, $3\ T_b$, ... etc.*

$$\text{The time of 1 data bit } (T_b) = \frac{1}{\text{data bit rate(D)}} \text{ sec.}$$

Example

Generating code sequence C1: 0 1 1 0 1 0 1 0, code length is number of chips in code sequence (N) = 8. If data bit rate (D) = 1 Mbit/sec

Bit time (T_b) $= 1/D$

$= 1/1024^2$

$= 9.5 \times (10)^{-7}$ second

Chip rate (D_c) $= N \times D$

$= 8 \times 1$ Mbit/s

$= 8$ M chip/s

Chip time (T_c) $= 1 / D_c = 1/8(1024)^2$

$= 1.9 \times (10)^{-7}$ second

Optical CDMA can be implemented using either non-coherent or coherent detection techniques. In non-coherent detection, the receiver looks only at the amplitude of the received signal, not the phase. Example is direct detection. In coherent detection, the receiver looks at both amplitude and phase of the signal.

However, there are many serious problems that make it difficult to implement a coherent optical system [9][10]. Modulation in this system require special lasers with separate sections for the continuous wave (CW) generation and modulation or using an external optical modulator after the laser transmitter. In fiber optic systems the fiber changes the polarization of the signal making it difficult to adjust the phase of local oscillator signal to have the same phase as the incoming optical carrier.

In this proposal, we concentrate in non-coherent optical CDMA system. This system can be separated into two groups. There are systems that rely on high-weight sequences and low-weight sequences. Both systems use parallel fiber delay-line type or tapped delay-line type. That is shown in figure 2.

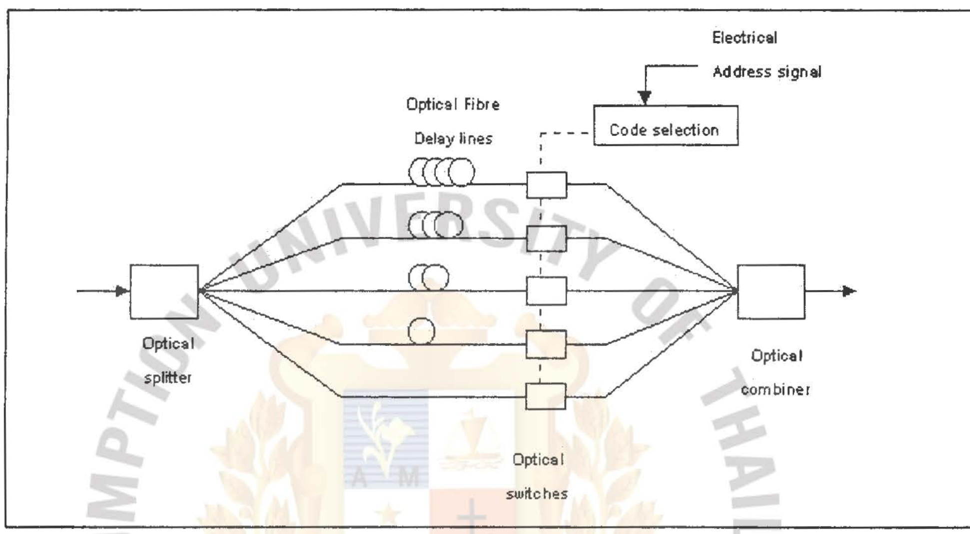


Figure 2 Optical fiber tapped delay-line encoder.

Parallel optical-delay line encoder is used in optical CDMA network because it can be encoded in high chip rate and generated any code sequence. However, If we increase chips, the number of delay line and size of coupler also increases. It causes addition of recombination loss. So, the signal will be reduced.

2.5 Non-coherent Optical CDMA Systems

Several architectures have been considered for the use of CDMA within an optical fiber, the most common being systems that use direct detection with 0/1.The data is modulated on the spreading sequence by OOK for non-coherent optical CDMA system for low-weight codes or Sequence Inversion Keying (SIK) for non-coherent optical CDMA system for high-weight codes. The figure 3 shows the principal transmitter of this system.

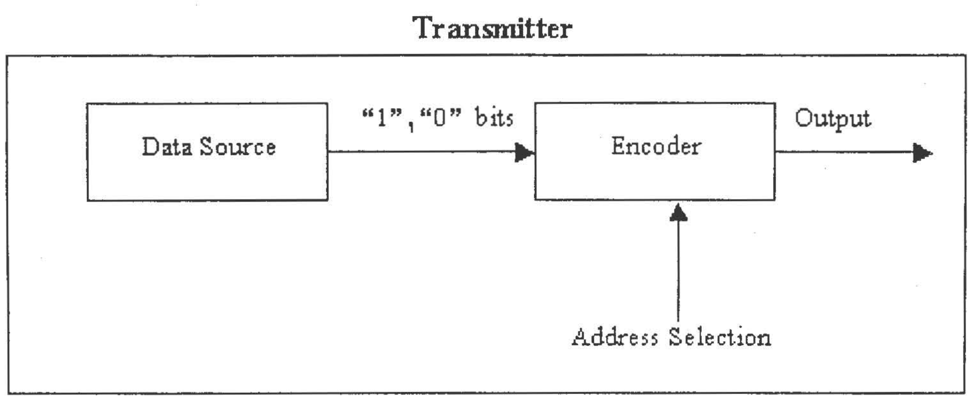


Figure 3 Transmitter of Optical CDMA.

<u>Example</u> of OOK for low-weight codes

CDMA is the process of letting all the users transmit on the same frequency at the same time but distinguishing between them by the code used. Each user is assigned 1 sequence used as its address

User 1 (C1) 0 1 1 1 0 1

User 2 (C2) 1 0 1 0 1 1

User 3 (C3) 0 0 1 1 1 1

The data bits are transmitted using a special form of on-off modulation. For a "0" bit, no signal is sent, while for a "1" bit an optical signal corresponding to the transmitter's code word is transmitted.

If user 1 wants to transmit data to user 2, user1 has to encode each data bit "1" by the address sequence of user 2. In figure 4, it illustrates encoding code sequence of transmission data from user 1 to user 2.

Bit "1" encoded by 1 0 1 0 1 1

Bit "0" encoded by 0 0 0 0 0 0

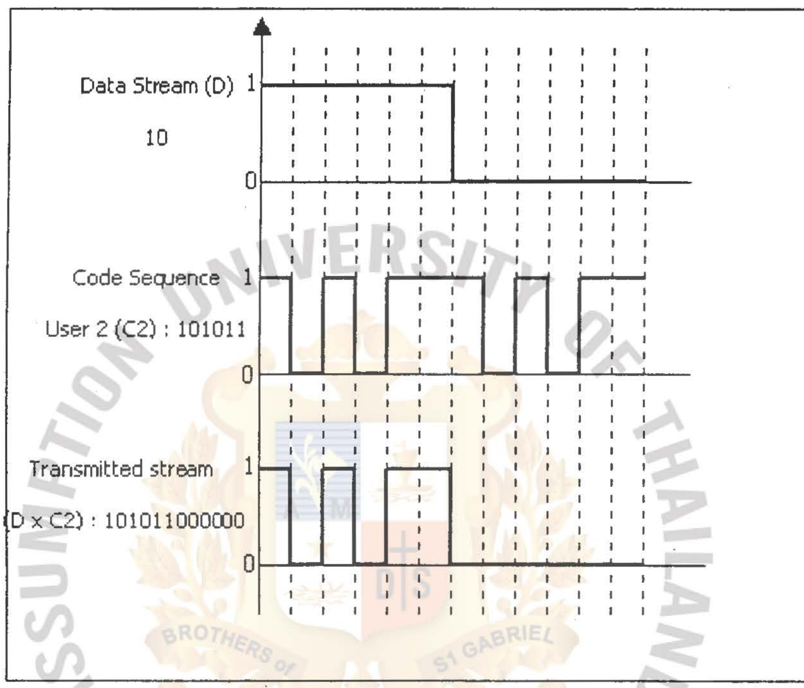


Figure 4 OOK Modulation technique.

For SIK modulation, the sequence is employed to transmit bit "1" while its complement is transmitted for bit "0"

Example of SIK for high-weight codes

If user 1 wants to transmit data to user 2, bit "1" encoded by address of user 2 and bit "0" encoded by complement of address of User 2. This modulation is shown in figure 5.

Bit "1" encoded by 1 0 1 0 1 1

Bit "0" encoded by 0 1 0 1 0 0

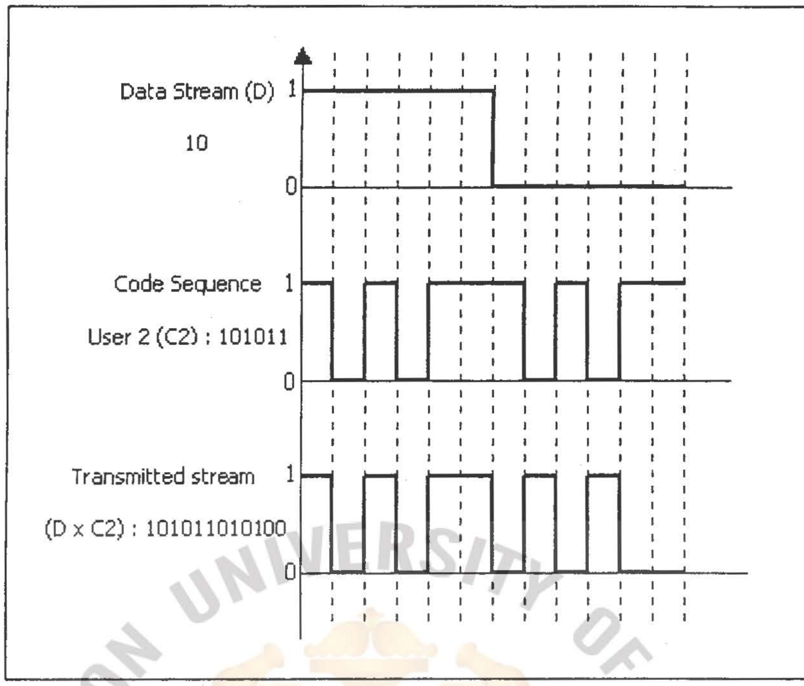


Figure 5 SIK Modulation technique.

2.6 Non-coherent All-optical CDMA Systems for High-Weight codes

Using high-weight sequences for non-coherent optical CDMA schemes, [6] relies on the unipolar-bipolar correlation at the receiver. Theses systems are based on a direct (DS) code-division multiple access (CDMA) technique using sequence inversion keying (SIK) of sequences. Figure 6 shows optical transmitter for optical CDMA systems using high-weight codes. The difference between encoder of SIK and encoder of OOK is that SIK has two encoder for code sequence data bit "1" and the complement of data bit "1". At the output of transmitter, there are two kinds of outputs that are output of "1" sequence and its complement sequence.

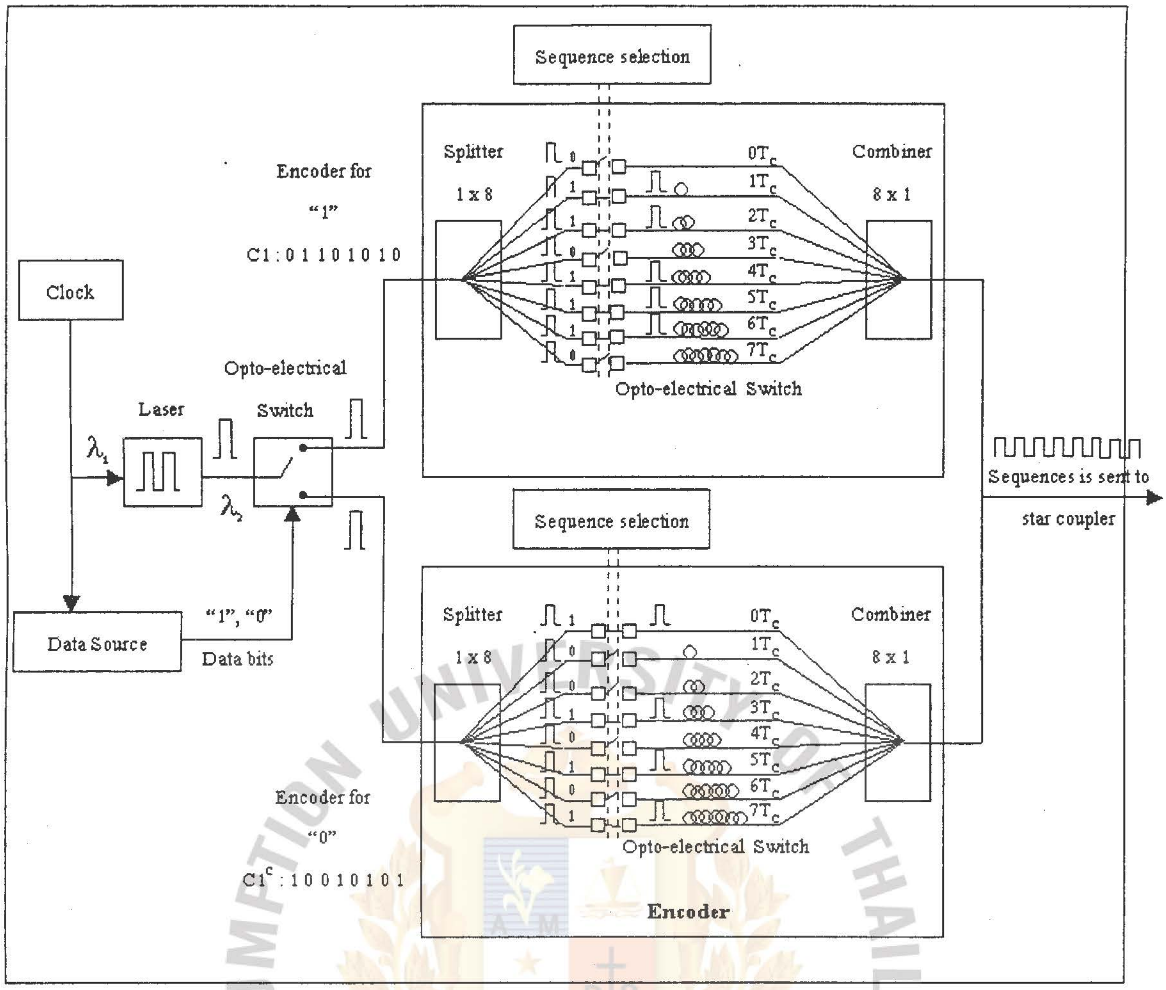


Figure 6 Optical transmitter for high-weight codes

2.7 Non-coherent Optical CDMA Systems for Concatenated codes

Optical CDMA systems for Kronecker codes, also called concatenated codes, which are product of two sequences of relatively large weight has been proposed in [11]. It is based on the principle of sequence-inversion keying (SIK) of component codes having different clock rates. This method is used for avoiding bottleneck of processing speed through all-optical encoding and decoding while simultaneously avoiding the problem of optical recombination loss by using a short inner sequence. The receiver consists of optical inner sequence and electronic outer sequence correlators. The first correlator corresponds to the inner sequence and is concatenated to a second correlator for the outer sequence. Both outer sequence and inner sequence are kept short enough to allow effective all-optical encoding and decoding.

It has been shown in [11] that Kronecker codes which have component sequeneces (inner and outer sequences) are balanced sequences that can be used in non-coherent optical CDMA systems. In [11] Kronecker sequences using Lampel sequences and Gold sequences were investigated. In this work, we will investigate optical CDMA systems using kronecker codes that are constructed using balanced m-sequence and balanced Gold sequences as components.

2.7.1 Linear Feedback Shift-Register (LFSR)

Maximal-length sequences (m-sequences) are, by definition, the largest codes that can be generated by a given shift register or a delay element of a given length. At each clock time the register shifts all contents to the right.

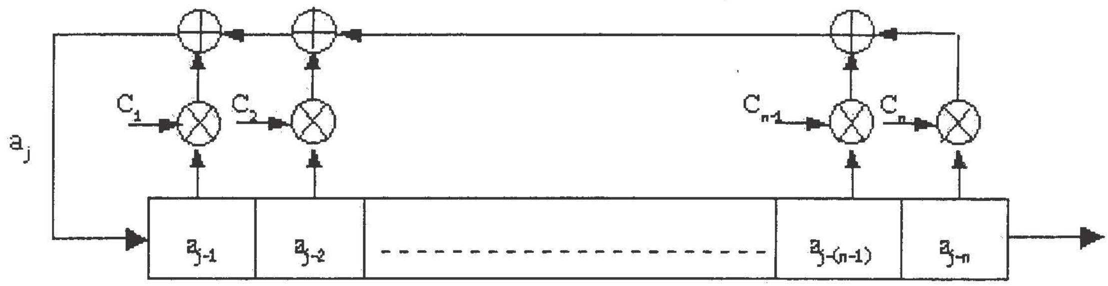


Figure 7 Linear Feedback Shift Register (LFSR)

The LFSR correspond to a characteristic polynomial which is shown by this equation (1)

Primitive Polynomial : $f(x) = C_r x^r + C_{r-1}x^{r-1} + \dots\dots + C_1 x + 1$ (1)

The number of stages of LFSR is equal to the maximum degree of primitive polynomial $f(x)$ that is illustrated by r. From this equation, the value of C_r is always 1 and the others C_1 , C_2 ,...., C_{r-1} can be either 0 or 1. For each primitive polynomial $f(x)$, it exists a reciprocal polynomial $f_R(x)$ which is represented by equation (2). The sequence generated by $f_R(x)$ is also an m- sequence of period $N = 2^r - 1$.

Reciprocal Polynomial : $f_R(x) = x^r f(\frac{1}{x})$ (2)

<u>Example</u>

The Primitive polynomial $f(x) = x^6 + x^4 + x^2 + 1$

$f(x) = 1x^6 + 0x^5 + 1x^4 + 0x^3 + 1x^2 + 0x + 1$

From this equation, the LFSR has 4 stages to generate m-sequence. The coefficient of equation are $C_6 = 1$, $C_5 = 0$, $C_4 = 1$, $C_3 = 0$, , $C_2 = 1$ and $C_1 = 0$.

Equation of reciprocal polynomial is

$$f(x) = x^6 + x^4 + x^2 + 1$$

$$x^6 f(\ 1\ ) = x^6\left[\left(\frac{1}{x}\right)^6 + \left(\frac{1}{x}\right)^4 + \left(\frac{1}{x}\right)^2 + 1\right]$$

$$f_R(x) = 1 + x^2 + x^4 + x^6$$

The Reciprocal Polynomial is $1 + x^2 + x^4 + x^6$. We can use this equation to generate new m-sequence by LFSR. The structure of LFSR is illustrated as below. The initial loading of primitive polynomial is 1 0 1 0 1 0.

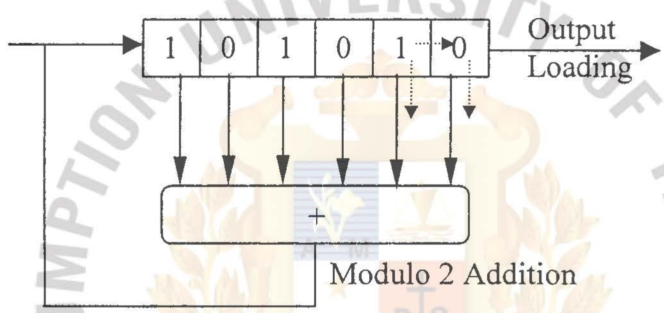


Figure 8 Diagram of a LFSR

2.7.2 Maximal-length sequence (m-sequence)

The process of generating m-sequence is represented in the following example.

Example

$$\begin{array}{cccc} f(x) = x^4 & +\,0x^3 & +\,0x^2 & +\,x+1 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ C_4 = 1 & C_3 = 0 & C_2 = 0 & C_1 = 1 \end{array}$$

The degree of $f(x)$ = number of stage of LFSR

The degree of $f(x)$ is 4. So, The number of stage is 4 stages.

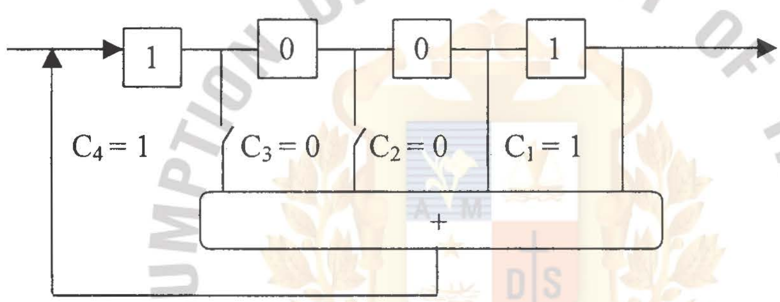
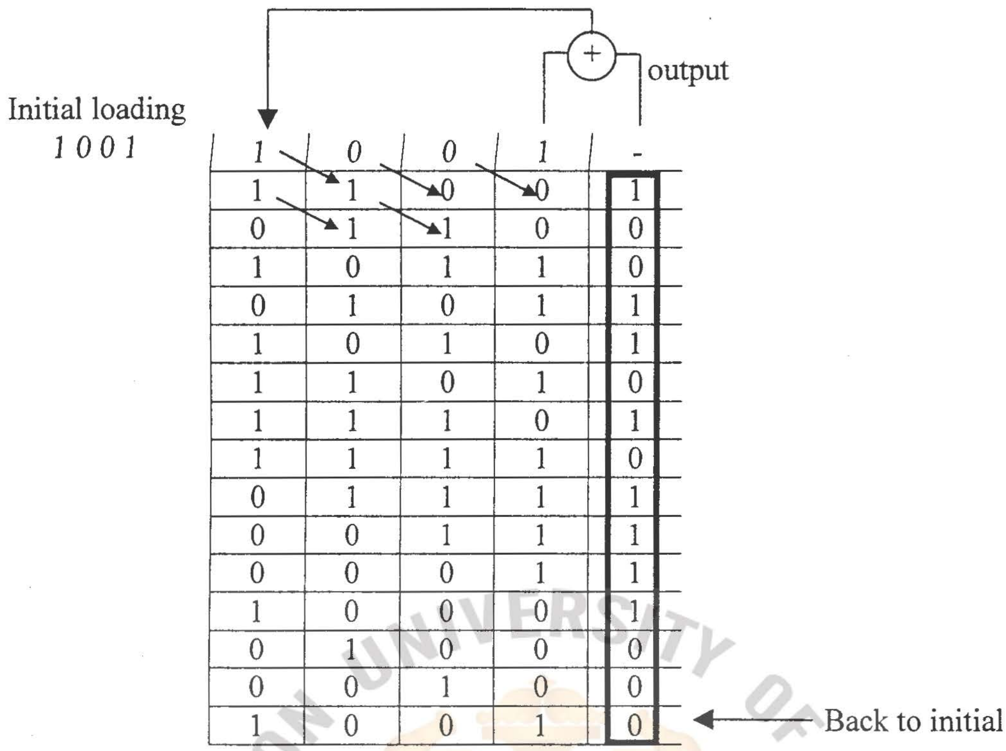


Figure 9 LFSR for polynomial $f(x) = x^4 + x + 1$

Initial loading should not be equal 0.



1	0	0	1	-
1	1	0	0	1
0	1	1	0	0
1	0	1	1	0
0	1	0	1	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0
0	1	1	1	1
0	0	1	1	1
0	0	0	1	1
1	0	0	0	1
0	1	0	0	0
0	0	1	0	0
1	0	0	1	0

At output of LFSR, we have sequence 0 0 0 1 1 1 1 0 1 0 1 1 0 0 1

Last chip First chip

This sequence is called m – sequence. The number of chip is $2^4 - 1 = 15$.

m-sequence length

N	Degree of Polynomial	N =
7	3	$2^3 - 1$
15	4	$2^4 - 1$
31	5	$2^5 - 1$
63	6	$2^6 - 1$
127	7	$2^7 - 1$
255	8	$2^8 - 1$
Etc.		

Reciprocal Polynomial can use to generate a new m-sequence

Replace x in $x^r f(x)$ by $\frac{1}{x}$

Example

$$f(x) = x^4 + x + 1$$

$$x^4 f\left(\frac{1}{x}\right) = x^4\left[\left(\frac{1}{x}\right)^4 + \frac{1}{x} + 1\right]$$

$$= 1 + x^3 + x^4$$

The Reciprocal polynomial is $f_r(x) = x^4 + x^3 + 1$

This can be used for generating for a new m-sequence.

Example

Find m-sequence generated by $f_1(x) = x^5 + x^2 + 1$

and by $f_2(x) = x^5 + x^4 + x^3 + x^2 + 1$

1) $f_1(x) = x^5 + x^2 + 1$

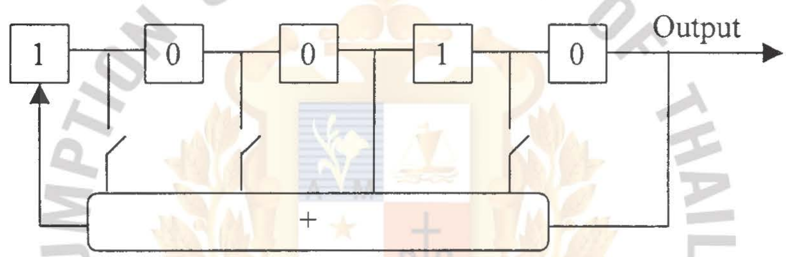
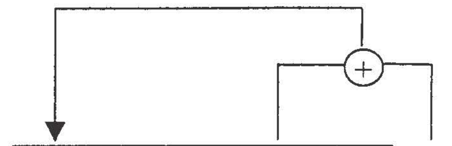


Figure 10 LFSR for $f_1(x) = x^5 + x^2 + 1$

Generating m-sequence by LFSR



1	0	0	1	0	
0	1	0	0	1	0
1	0	1	0	0	1
1	1	0	1	0	0
0	1	1	0	1	0
0	0	1	1	0	1
1	0	0	1	1	0
1	1	0	0	1	1
1	1	1	0	0	1
1	1	1	1	0	0
1	1	1	1	1	0
0	1	1	1	1	1
0	0	1	1	1	1
0	0	0	1	1	1
1	0	0	0	1	1
1	1	0	0	0	1
0	1	1	0	0	0
1	0	1	1	0	0
1	1	0	1	1	0
1	1	1	0	1	1
0	1	1	1	0	1
1	0	1	1	1	0
0	1	0	1	1	1
1	0	1	0	1	1
0	1	0	1	0	1
0	0	1	0	1	0
0	0	0	1	0	1
0	0	0	0	1	0
1	0	0	0	0	1
0	1	0	0	0	0
0	0	1	0	0	0
1	0	0	1	0	0

m_1-sequence is 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

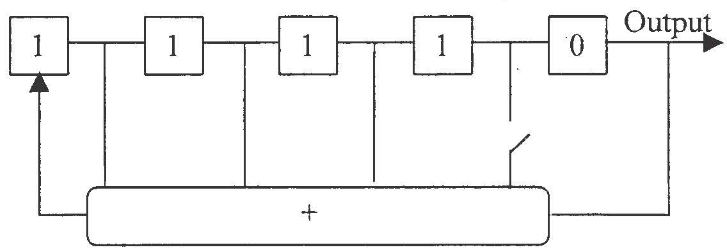


Figure 11 LFSR for $f_2(x) = x^5 + x^4 + x^3 + x^2 + 1$

1	1	1	1	0	
1	1	1	1	1	0
0	1	1	1	1	1
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	0	1	1
0	1	1	1	0	1
0	0	1	1	1	0
0	0	0	1	1	1
1	0	0	0	1	1
0	1	0	0	0	1
1	0	1	0	0	0
0	1	0	1	0	0
1	0	1	0	1	0
1	1	0	1	0	1
0	1	1	0	1	0
1	0	1	1	0	1
0	1	0	1	1	0
0	0	1	0	1	1
0	0	0	1	0	1
0	0	0	0	1	0
1	0	0	0	0	1
1	1	0	0	0	0
0	1	1	0	0	0
0	0	1	1	0	0
1	0	0	1	1	0
0	1	0	0	1	1
0	0	1	0	0	1
1	0	0	1	0	0
1	1	0	0	1	0
1	1	1	0	0	1
1	1	1	1	0	0

m_2-sequence is 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0

2.7.3 Gold Sequence

Gold sequences can be generated by 2 m-sequences of same length. The number of Gold sequence lenghts $N = 2^r + 1$. Those sequences include :

- m-sequence generated by $f_1(x)$ [Seq(a)]
- m-sequence generated by $f_2(x)$ [Seq(b)]

The other $2^r - 1$ Gold Sequences are given by

$$g_j\,(i) = a(i) \oplus T^j{}_b(i)\ ; \text{ for } 0 \le i \le N-1$$

Where $T^j{}_b(i)$ = the i^{th} element of the j^{th} cyclic shift of m- sequence {b}.

<u>For example</u> ; Generating gold sequence using 2 m-sequences m_1,m_2.

g_1 = m_1 = 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

g_2 = m_2 = 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0

$g_j\,(i) = a(i) \oplus T^j{}_b(i)$; for $0 \le i \le N-1$

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^0{}_2$; 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0

$g_3 = m_1 + m^0{}_2$ = 0 1 0 1 1 0 0 1 1 1 1 1 0 1 0 1 1 0 1 1 1 1 1 0 1 1 0 1 1 0 0

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^1{}_2$; 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0

$g_4 = m_1 + m^1{}_2$ = 1 0 0 0 1 1 0 1 1 0 0 0 0 0 1 0 0 1 1 1 0 1 1 1 0 1 0 1 1 1 0

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^2{}_2$; 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1

$g_5 = m_1 + m^2{}_2$ = 0 0 1 0 0 1 0 1 0 1 1 0 1 1 0 1 1 1 0 0 0 1 0 0 0 1 0 1 0 1 1

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

m^3_2 ; 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0

$g_6 = m_1 + m^3_2$ = 0 1 1 1 0 1 0 0 1 0 1 1 0 0 1 0 1 1 0 0 0 0 1 0 0 1 0 0 0 0 0

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

m^4_2 ; 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0

$g_7 = m_1 + m^4_2$ = 1 1 0 1 0 1 1 1 0 0 0 0 1 1 0 0 1 0 0 0 1 1 1 0 0 1 1 0 1 1 0

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

m^5_2 ; 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1

$g_8 = m_1 + m^5_2$ = 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 1 0 0 0 1 1 0 1 1

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

m^6_2 ; 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1

$g_9 = m_1 + m^6_2$ = 0 0 0 1 1 1 1 0 1 0 0 0 1 0 0 1 0 0 1 0 0 1 1 0 1 0 0 0 0 0 1

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

m^7_2 ; 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0

$g_{10} = m_1 + m^7_2$ = 0 0 0 0 0 0 1 1 0 1 1 1 1 0 1 1 0 1 0 0 0 1 1 1 1 1 1 0 1 0 0

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

m^8_2 ; 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 1 0 0 1 0 0 1 1 0 0

$g_{11} = m_1 + m^8_2$ = 0 0 1 1 1 0 0 0 1 0 0 1 1 1 0 0 1 0 0 0 0 1 0 1 0 0 1 1 1 1 0

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

m^9_2 ; 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0

$g_{12} = m_1 + m^9_2$ = 0 1 0 0 1 1 1 1 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^{10}{}_2$; 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0

$g_{13} = m_1 + m^{10}{}_2$ = 1 0 1 0 0 0 0 0 1 1 0 0 0 1 0 1 0 0 0 0 1 0 1 1 1 1 1 0 0 0 1 0

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^{11}{}_2$; 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1

$g_{14} = m_1 + m^{11}{}_2$ = 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 1 0 1 0 1 1 0 0 1 1

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^{12}{}_2$; 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0

$g_{15} = m_1 + m^{12}{}_2$ = 1 1 0 0 0 0 0 1 1 0 1 0 1 1 1 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^{13}{}_2$; 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1

$g_{16} = m_1 + m^{13}{}_2$ = 1 0 1 1 1 1 0 1 0 0 1 1 0 1 1 1 0 1 1 0 1 0 1 0 1 0 1 0 1 1 1

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^{14}{}_2$; 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1

$g_{17} = m_1 + m^{14}{}_2$ = 0 1 0 0 0 1 0 0 0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 0 0 1

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^{15}{}_2$; 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0

$g_{18} = m_1 + m^{15}{}_2$ = 1 0 1 1 0 1 1 0 0 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 0 0 1 0 0

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^{16}{}_2$; 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1

$g_{19} = m_1 + m^{16}_2$ = 0 1 0 1 0 0 1 0 1 0 1 0 0 1 0 0 0 1 1 0 0 0 0 1 1 1 1 1 1 1 1

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

m^{17}_2 ; 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0

$g_{20} = m_1 + m^{17}_2$ = 1 0 0 1 1 0 1 1 0 0 1 0 0 0 0 1 1 1 0 0 1 0 0 1 0 0 0 1 0 0 0

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

m^{18}_2 ; 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1

$g_{21} = m_1 + m^{18}_2$ = 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 1 0 0 1 1 0 0 0 1 1 0 0 1 1 1

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

m^{19}_2 ; 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0

$g_{22} = m_1 + m^{19}_2$ = 0 0 1 0 1 1 1 0 0 0 1 1 1 1 0 0 0 0 1 1 1 0 1 1 0 1 1 1 0 0 0

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

m^{20}_2 ; 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0

$g_{23} = m_1 + m^{20}_2$ = 0 1 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 1 1 1 1 0 0 0 0 0 0 1 1 0

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

m^{21}_2 ; 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0

$g_{24} = m_1 + m^{21}_2$ = 1 1 1 1 1 0 1 0 0 1 0 0 1 0 1 1 1 1 1 1 0 0 1 0 1 1 1 1 0 1 0

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

m^{22}_2 ; 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1

$g_{25} = m_1 + m^{22}_2$ = 1 1 0 0 1 0 1 0 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 0 0 0 0 0 1 1

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^{23}{}_2$; 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1

$g_{26} = m_1 + m^{23}{}_2$ = 1 0 1 0 1 0 1 1 1 0 0 1 0 1 0 0 1 1 0 1 0 1 0 0 1 1 1 0 0 0 1

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^{24}{}_2$; 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1

$g_{27} = m_1 + m^{24}{}_2$ = 0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 1 0 1 0 0 0 1 1 0 0 1 0 1 0 1

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^{25}{}_2$; 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0

$g_{28} = m_1 + m^{25}{}_2$ = 1 1 1 0 1 1 0 0 1 1 1 0 1 0 0 0 0 1 0 0 1 1 0 0 1 0 1 1 1 0 0

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^{26}{}_2$; 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1

$g_{29} = m_1 + m^{26}{}_2$ = 1 1 1 0 0 1 1 1 1 0 1 1 1 0 0 1 1 0 0 1 0 0 1 1 1 0 0 1 1 1 1

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^{27}{}_2$; 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1

$g_{30} = m_1 + m^{27}{}_2$ = 1 1 1 1 0 0 0 1 0 0 0 1 1 0 1 0 0 0 1 0 1 1 0 1 1 1 0 1 0 0 1

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^{28}{}_2$; 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1

$g_{31} = m_1 + m^{28}{}_2$ = 1 1 0 1 1 1 0 0 0 1 0 1 1 1 0 1 0 1 0 1 0 0 0 1 0 1 0 0 1 0 1

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^{29}{}_2$; 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1

$g_{32} = m_1 + m^{29}{}_2$ = 1 0 0 0 0 1 1 0 1 1 0 1 0 0 1 1 1 0 1 0 1 0 0 0 0 1 1 1 0 0 1

m_1 ; 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 0

$m^{30}{}_2$; 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1

$g_{33} = m_1 + m^{30}{}_2$ = 0 0 1 1 0 0 1 1 1 1 0 0 1 1 1 0 0 1 0 1 1 0 1 0 0 0 0 1 1 0 1

Both m-sequence and gold-sequence can be generated by LFSR. Those code sequences are unipolar (0,1). We can change unipolar to bipolar by replacing 1 for 0 and replacing -1 for 1.Balanced m-sequence can be obtained by replacing chip "1" of m-sequence by "1,-1" for chip "1" and "0" by "–1, 1".

For balance gold sequence, firstly, we must generate gold sequence from 2 m-sequence. Like m-sequence, gold sequence is changed to bipolar gold sequences. After that those bipolar gold sequence is changed to balance gold sequence.When we have balance m-sequence and balance gold sequence, we can generate kronecker sequence.

Chapter III Kronecker Codes for Optical Fiber CDMA Networks

Let $\{C_i(l)\}$ denoted aperiodic bipolar sequence of length N_1 such that for all integer l the sequence elements $\{C_i(l)\} \in \{-1,1\}$ and $\{D_i(j)\}$ denoted a periodic bipolar sequences of period N_2 whose sequence elements are also $\{-1,1\}$. Defined the periodic sequence $\{A_i(m)\}$ is kronecker sequence or concatenated sequence. The sequence $\{A_i(m)\}$ of length $N = N_1 N_2$ is concatenated sequence that composed of the inner sequence $\{C_i(l)\}$ and the outer sequence $\{D_i(j)\}$. Each outer chip multiplies each sequence of inner sequence. The number of sequence is equal the number of all balanced Gold sequence. An example of generating kronecker sequences using balanced m-sequences and balanced Gold sequences is shown in appendix A.

3.1 Programme for generating Kronecker sequence

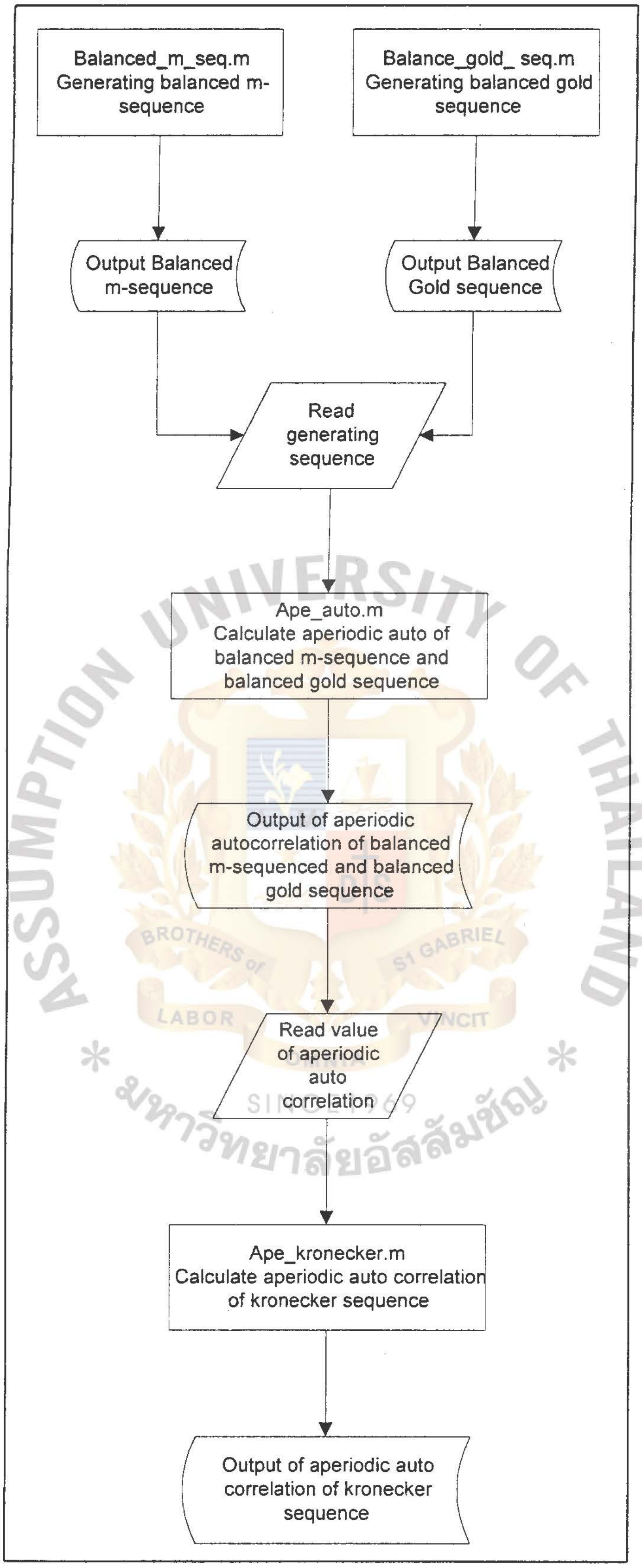


Figure 12 Flowchart of Calculating Aperiodic Auto Correlation for Kronecker sequences

3.1.1 Module for generating balanced m-sequence (balanced_m_seq.m)

To generate balanced m-sequence, the stages (r) of LFSR and coefficient of primitive polynomial are required to input in the program. The lenght of sequence

$N = (2^r - 1) \times 2$

Where : r =3

- Input Stage of LFSR for primitive polynomial r = 3
- Input value of coefficient

  Coefficient of LFSR = 1

  Coefficient of LFSR = 1

  Coefficient of LFSR = 0
- Input value of coefficient

  Initial loading of LFSR = 1

  Initial loading of LFSR = 1

  Initial loading of LFSR = 0

The ouput is kept in "Bm3" file

The lenght of sequences = $(2^3 - 1) \times 2 = 14$

Where : r =4

- Input Stage of LFSR for primitive polynomial r = 4
- Input value of coefficient

  Coefficient of LFSR = 1

  Coefficient of LFSR = 1

  Coefficient of LFSR = 1

  Coefficient of LFSR = 0

- Input value of coefficient

  Initial loading of LFSR = 1

  Initial loading of LFSR = 1

  Initial loading of LFSR = 1

  Initial loading of LFSR = 0

The ouput is kept in "Bm4" file

The lenght of sequences = $(2^4 - 1) \times 2 = 30$

Where : r =5

- Input Stage of LFSR for primitive polynomial r = 5

- Input value of coefficient

  Coefficient of LFSR = 1

  Coefficient of LFSR = 1

  Coefficient of LFSR = 1

  Coefficient of LFSR = 0

  Coefficient of LFSR = 1

- Input value of coefficient

  Coefficient of LFSR = 1

  Coefficient of LFSR = 1

  Coefficient of LFSR = 1

  Coefficient of LFSR = 0

  Coefficient of LFSR = 1

The ouput is kept in "Bm5" file

The lenght of sequences = $(2^5 - 1) \times 2 = 62$

3.1.2 Module for generating balanced gold sequence (balance_gold_seq.m)

The balanced gold sequences are generated by base on pair of m-sequences.The m-sequences are generated by primitive polynomial and reciprocal polynomial. The difference of gold sequence = $2^r + 1$ sequences. The lenght of balanced gold sequence = $(2^r - 1) \times 2$

Where : r =3

- Input Stage of LFSR for primitive polynomial r = 3
- Input value of coefficient

    Coefficient of LFSR = 1

    Coefficient of LFSR = 1

    Coefficient of LFSR = 0

- Input value of coefficient

    Initial loading of LFSR = 1

    Initial loading of LFSR = 1

    Initial loading of LFSR = 0

- Input Stage of LFSR for reciprocal of polynomial r = 3
- Input value of coefficient

    Coefficient of LFSR = 1

    Coefficient of LFSR = 0

    Coefficient of LFSR = 1

- Input value of coefficient

    Initial loading of LFSR = 1

    Initial loading of LFSR = 0

    Initial loading of LFSR = 1

The different of gold sequence = $2^3 + 1 =$ 9 sequences

The lenght of each sequence = $(2^3 - 1) \times 2 = 14$

The ouput is file name of balanced gold sequencec that are kept in files as below :

When r =4

- Input Stage of LFSR for primitive polynomial r = 4
- Input value of coefficient

  Coefficient of LFSR = 1

  Coefficient of LFSR = 1

  Coefficient of LFSR = 1

  Coefficient of LFSR = 0

- Input value of coefficient

  Initial loading of LFSR = 1

  Initial loading of LFSR = 1

  Initial loading of LFSR = 1

  Initial loading of LFSR = 0

- Input Stage of LFSR for reciprocal of polynomial r = 4
- Input value of coefficient

  Coefficient of LFSR = 1

  Coefficient of LFSR = 1

  Coefficient of LFSR = 1

  Coefficient of LFSR = 0

- Input value of coefficient

  Initial loading of LFSR = 1

  Initial loading of LFSR = 0

Initial loading of LFSR = 1

Initial loading of LFSR = 1

The difference of gold sequence = $2^4 + 1 =$ 17 sequences

The lenght of each sequences = $(2^4 - 1) \times 2 = 30$

The ouput is file name of balanced gold sequencec that are kept in files . The lists name is shown in index.

When r =5

- Input Stage of LFSR for primitive polynomial

  r = 5

- Input value of coefficient

  Coefficient of LFSR = 1

  Coefficient of LFSR = 1

  Coefficient of LFSR = 1

  Coefficient of LFSR = 0

  Coefficient of LFSR = 1

- Input value of coefficient

  Coefficient of LFSR = 1

  Coefficient of LFSR = 1

  Coefficient of LFSR = 1

  Coefficient of LFSR = 0

  Coefficient of LFSR = 1

- Input Stage of LFSR for reciprocal polynomial

  r = 5

- Input value of coefficient

Coefficient of LFSR = 1

Coefficient of LFSR = 1

Coefficient of LFSR = 0

Coefficient of LFSR = 1

Coefficient of LFSR = 1

- Input value of coefficient

Coefficient of LFSR = 1

Coefficient of LFSR = 1

Coefficient of LFSR = 0

Coefficient of LFSR = 1

Coefficient of LFSR = 1

The different of gold sequence = $2^5 + 1 =$ 33 sequences

The lenght of each sequences = $(2^5 - 1) \times 2 = 62$

The ouput is file name of belanced gold sequencec that are kept in files in Appendix B.

3.1.3 Module for calculating aperiodic auto correlation of balanced m-sequences and balanced gold sequences (ape_auto.m)

After generating all of balanced m-sequence and balanced gold sequences, each sequence has to calculate in aperiodic auto correlation forms. The output of them is used to calculate further in aperiodic auto correlation of kronecker sequences. All of files for aperiodic auto correlation display as follows.

a. Filename for Aperiodic Auto Correlation of balanced m-sequence

   r = 3 : Abm3

   r = 4 : Abm4

   r = 5 : Abm5

b. Aperiodic Auto Correlation of balanced gold sequence

3.1.4 Module for calculating aperiodic auto correlation of kronecker sequences (ape_kronecker.m)

The aperiodic auto correlation of kronecker sequence applies aperiodic auto correlation of balanced m-sequences and balanced gold sequence to calculate the value. Due to many of the balanced gold sequences, we select 9 different balanced gold sequence as outer and inner.

Chapter IV The Performance of Concatenated Codes in Optical CDMA LANs

4.1 BER Performance of Optical Fiber CDMA LANs Using Kronecker Codes

The calculation of the BER performance is based on the method that is shown in equation (3).

$$\mathrm{BER} = \frac{1}{2}\,\mathrm{erfc}\left(\frac{\Delta}{\sqrt{2}}\right) \qquad (3) \quad [11]$$

where Δ is given by

$$\Delta = \frac{N}{\sqrt{\sum_{k=2}^{k} \sigma^2{}_k}} = \frac{N}{\sqrt{(K-1)\sigma^2{}_k}} \qquad (4) \quad [11]$$

Where

N = the kronecker code length

K = the number simultaneous users

$\sigma^2{}_k$ = P_{MAI} the mean power of the multiple-access interference (MAI)

4.2 The Mean Power of Multiple-Access Interference

For a given correlator receiver configuration, the power of MAI caused by the k-th user at the receiver of the i-th user is a function of the cross-correlation parameters of the pair of sequences $\{A_i\ (m)\}$ and $\{A_k\ (m)\}$. Those cross-correlation parameters depend on each interference condition.

Let P_{MAI} denote the mean value of interference power. P_{MAI} is definded by :

$$P_{MAI} = \frac{1}{N_s} \sum_{1}^{N_S} \frac{1}{N} \sum_{m_k = 1-N}^{N-1} C^2{}_{a_k, a_i}(m_k) \qquad (5)$$

Where

N_s	=	the number of all pairs of sequence $\{A_i(m)\}$ and $\{A_k(m)\}$ of the set of Kronecker sequences
$C\ a_{k,}a_i\ (m_k)$	=	the aperiodic cross-correlation of Kronecker sequence $\{A_i(m)\}$ and $\{A_k(m)\}$ at the time shift m_k

The number of all pairs of sequence N_s is given by

$$N_s = \frac{S\,!}{(S-2)!\,2!} = \frac{S(S-1)}{2} \qquad (6)$$

Where S is the number of Kronecker sequence in the set.

4.3 Aperiodic cross-correlation of balanced kronecker sequences.

Let $\{A_i(m)\}$ and $\{A_k(m)\}$ denote two periodic bipolar Kronecker sequences of period N from the set of S sequences $\{\ A_k\ :\ 1\ \leq k \leq\ S\ \}$. The aperiodic cross-correlation function between $\{A_i(m)\}$ and $\{A_k(m)\}$ is definded by equation (7) where τ is the time shift which is an integer.

$$Ca_i,a_k(\tau) = \begin{cases} \sum\limits_{m=0}^{N-1-\tau} A_i(m)\,A_k(m+\tau) & 0\ \leq \tau \leq\ N \\ \sum\limits_{m=0}^{N-1+\tau} A_i(m-\tau)\,A_k(m) & -N\ \leq \tau \leq\ 0 \\ 0 & |\tau|\ \leq\ N \end{cases} \qquad (7) \quad [11]$$

It has been shown in [11] that for a noncoherent optical fiber DS-CDMA LANs using balanced Kronecker sequences of period N, the variance of the interference caused by the k-th user at the receiver of the i-th user depends on the aperiodic cross correlation of two Kronecker sequences $\{A_i(m)\}$ and $\{A_i(m)\}$.

We can find the sum of squares of aperiodic cross-correlation at all time shifts $m_k(1-N \leq m_k \leq N-1)$ for each pair of Kronecker sequence $\{A_i(m)\}$ and $\{A_i(m)\}$ by

$$\sum_{m_k=1-N}^{N-1} C^2_{a_k a_i}(m_k) = \sum_{m_k=1-N}^{N-1} C_{a_k}(m_k)\, C_{a_i}(m_k) \qquad (8)$$

where $C_{a_k}(m_k)$ and $C_{a_i}(m_k)$ are the aperiodic auto-correlation of kronecker sequences $\{A_i(m)\}$ and $\{A_i(m)\}$ at the time shift m_k.

$$P_{MAI} = \frac{1}{N_s}\sum_{1}^{N_s}\frac{1}{N}\sum_{m_k=1-N}^{N-1} C_{a_k}(m_k)\, C_{a_i}(m_k) \qquad (9)$$

4.4 Programme for calculating BER

Flowchart of Calculating BER by Programme MATLAB

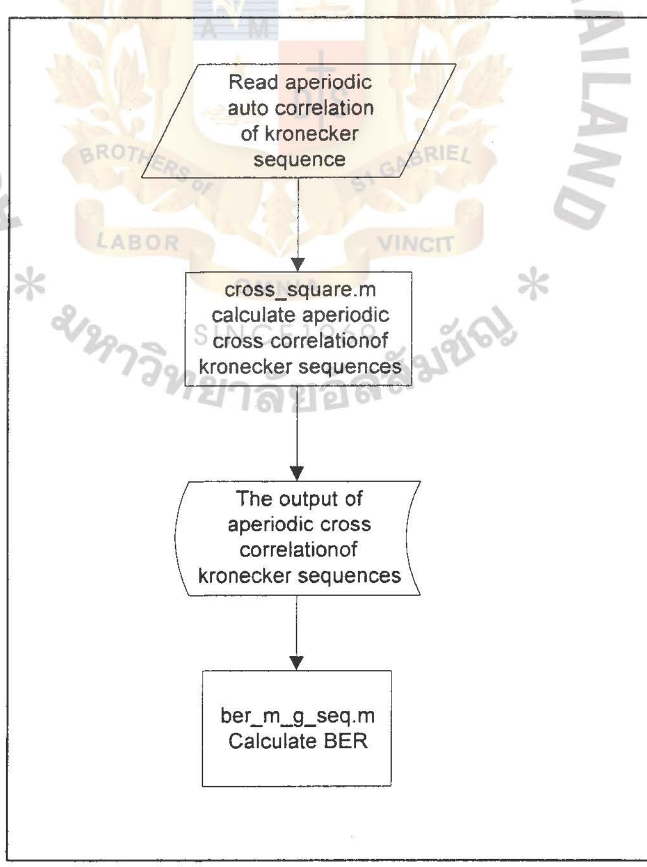


Figure 13 The Flowchart for calculating BER

4.4.1 Module for calculating The Mean Power of Multiple-Access Interference and Numerical result

$$P_{MAI} = \frac{1}{N_s} \sum_{1}^{N_s} \frac{1}{N} \sum_{m_k = 1-N}^{N-1} Ca_k(m_k)\, Ca_i(m_k)$$

To find the value of P_{MAI} , program is designed to calculate aperiodic cross correlation of Kronecker sequences.

The pairs of different sequences of Kronecker sequences have 36 sequences

$$N_s = \frac{S\,!}{(S-2)!\,2!} = \frac{S(S-1)}{2} = \frac{9\,(8)}{2} = 36$$

r = 3 Balanced gold sequence is inner

Balanced m-sequence is outer

Pairs of Sequence	Cak	Cai	Sum $C^2_{ak,ai}$	Sum $C^2_{ak,ai}$ / N
1	Kgm3_1	Kgm3_2	339388	1731.571
2	Kgm3_1	Kgm3_3	122332	624.143
3	Kgm3_1	Kgm3_4	88060	449.286
4	Kgm3_1	Kgm3_5	183260	935.000
5	Kgm3_1	Kgm3_6	110908	565.857
6	Kgm3_1	Kgm3_7	126140	643.571
7	Kgm3_1	Kgm3_8	339388	1731.571
8	Kgm3_1	Kgm3_9	228956	1168.143
9	Kgm3_2	Kgm3_3	198492	1012.714
10	Kgm3_2	Kgm3_4	187068	954.429
11	Kgm3_2	Kgm3_5	160412	818.429
12	Kgm3_2	Kgm3_6	251804	1284.714
13	Kgm3_2	Kgm3_7	84252	429.857
14	Kgm3_2	Kgm3_8	301308	1537.286
15	Kgm3_2	Kgm3_9	209916	1071.000
16	Kgm3_3	Kgm3_4	202300	1032.143

17	Kgm3_3	Kgm3_5	190876	973.857
18	Kgm3_3	Kgm3_6	168028	857.286
19	Kgm3_3	Kgm3_7	118524	604.714
20	Kgm3_3	Kgm3_8	198492	1012.714
21	Kgm3_3	Kgm3_9	156604	799.000
22	Kgm3_4	Kgm3_5	122332	624.143
23	Kgm3_4	Kgm3_6	426972	2178.429
24	Kgm3_4	Kgm3_7	179452	915.571
25	Kgm3_4	Kgm3_8	187068	954.429
26	Kgm3_4	Kgm3_9	236572	1207.000
27	Kgm3_5	Kgm3_6	69020	352.143
28	Kgm3_5	Kgm3_7	251804	1284.714
29	Kgm3_5	Kgm3_8	160412	818.429
30	Kgm3_5	Kgm3_9	168028	857.286
31	Kgm3_6	Kgm3_7	103292	527.000
32	Kgm3_6	Kgm3_8	251804	1284.714
33	Kgm3_6	Kgm3_9	270844	1381.857
34	Kgm3_7	Kgm3_8	84252	429.857
35	Kgm3_7	Kgm3_9	213724	1090.429
36	Kgm3_8	Kgm3_9	209916	1071.000
		Total	**6692084**	34143.29
		P_{MAI}=Total/Ns		948.42

Table 1 Sum of Aperiodic Cross Correlation of Kronecker Sequences (Balanced gold sequence as inner and Balanced m-sequence as outer) to find the Mean Power of Multiple Access when r = 3

r = 3 Balanced m-sequence is inner

Balanced gold sequence is outer

Pairs of Sequence	Cak	Cai	Sum $C^2_{ak,ai}$	Sum $C^2_{ak,ai}$ / N
1	Kmg3_1	Kmg3_2	306940	1566.020
2	Kmg3_1	Kmg3_3	103132	526.184
3	Kmg3_1	Kmg3_4	139004	709.204
4	Kmg3_1	Kmg3_5	150428	767.490
5	Kmg3_1	Kmg3_6	116156	592.633
6	Kmg3_1	Kmg3_7	209148	1067.082
7	Kmg3_1	Kmg3_8	306940	1566.020
8	Kmg3_1	Kmg3_9	199324	1016.959
9	Kmg3_2	Kmg3_3	196124	1000.633
10	Kmg3_2	Kmg3_4	212348	1083.408
11	Kmg3_2	Kmg3_5	171676	875.898
12	Kmg3_2	Kmg3_6	274268	1399.327
13	Kmg3_2	Kmg3_7	125980	642.755
14	Kmg3_2	Kmg3_8	282492	1441.286
15	Kmg3_2	Kmg3_9	187900	958.673
16	Kmg3_3	Kmg3_4	210748	1075.245
17	Kmg3_3	Kmg3_5	215772	1100.878
18	Kmg3_3	Kmg3_6	200924	1025.122
19	Kmg3_3	Kmg3_7	137404	701.041
20	Kmg3_3	Kmg3_8	196124	1000.633
21	Kmg3_3	Kmg3_9	212348	1083.408
22	Kmg3_4	Kmg3_5	101532	518.020
23	Kmg3_4	Kmg3_6	445404	2272.469
24	Kmg3_4	Kmg3_7	186300	950.510
25	Kmg3_4	Kmg3_8	212348	1083.408
26	Kmg3_4	Kmg3_9	274268	1399.327
27	Kmg3_5	Kmg3_6	78684	401.449
28	Kmg3_5	Kmg3_7	217372	1109.041

29	Kmg3_5	Kmg3_8	171676	875.898
30	Kmg3_5	Kmg3_9	148828	759.327
31	Kmg3_6	Kmg3_7	91708	467.898
32	Kmg3_6	Kmg3_8	274268	1399.327
33	Kmg3_6	Kmg3_9	238396	1216.306
34	Kmg3_7	Kmg3_8	125980	642.755
35	Kmg3_7	Kmg3_9	246620	1258.265
36	Kmg3_8	Kmg3_9	187900	958.673
276480		**Total**	**6968564**	35553.90
		P_{MAI}=Total/Ns		987.61

Table 2 Sum of Aperiodic Cross Correlation of Kronecker Sequences (Balanced m-sequence as inner and Balanced gold sequence as outer) to find the Mean Power of Multiple Access when r = 3

r = 4 Balanced gold sequence is inner

Balanced m-sequence is outer

Pairs of Sequence	Cak	Cai	Sum $C^2_{ak,ai}$	Sum $C^2_{ak,ai}$ / N
1	Kgm4_1	Kgm4_2	4345660	4828.51
2	Kgm4_1	Kgm4_3	4879950	5422.17
3	Kgm4_1	Kgm4_4	3493980	3882.20
4	Kgm4_1	Kgm4_5	3679580	4088.42
5	Kgm4_1	Kgm4_6	4879950	5422.17
6	Kgm4_1	Kgm4_7	3493980	3882.20
7	Kgm4_1	Kgm4_8	3679580	4088.42
8	Kgm4_1	Kgm4_9	4879950	5422.17
9	Kgm4_2	Kgm4_3	5678940	6309.93
10	Kgm4_2	Kgm4_4	4553340	5059.27
11	Kgm4_2	Kgm4_5	5721500	6357.22
12	Kgm4_2	Kgm4_6	5678940	6309.93

13	Kgm4_2	Kgm4_7	4553340	5059.27
14	Kgm4_2	Kgm4_8	5721500	6357.22
15	Kgm4_2	Kgm4_9	4876520	5418.36
16	Kgm4_3	Kgm4_4	3457890	3842.10
17	Kgm4_3	Kgm4_5	4578120	5086.80
18	Kgm4_3	Kgm4_6	4876520	5418.36
19	Kgm4_3	Kgm4_7	3457890	3842.10
20	Kgm4_3	Kgm4_8	4578120	5086.80
21	Kgm4_3	Kgm4_9	4876520	5418.36
22	Kgm4_4	Kgm4_5	3126780	3474.20
23	Kgm4_4	Kgm4_6	3457890	3842.10
24	Kgm4_4	Kgm4_7	5899780	6555.31
25	Kgm4_4	Kgm4_8	3126780	3474.20
26	Kgm4_4	Kgm4_9	4578120	5086.80
27	Kgm4_5	Kgm4_6	4578120	5086.80
28	Kgm4_5	Kgm4_7	3126780	3474.20
29	Kgm4_5	Kgm4_8	6097890	6775.43
30	Kgm4_5	Kgm4_9	4578120	5086.80
31	Kgm4_6	Kgm4_7	3457890	3842.10
32	Kgm4_6	Kgm4_8	4578120	5086.80
33	Kgm4_6	Kgm4_9	5881130	6534.59
34	Kgm4_7	Kgm4_8	3126780	3474.20
35	Kgm4_7	Kgm4_9	3457890	3842.10
36	Kgm4_8	Kgm4_9	5742140	6380.16
		Total	**155013840**	178617.76
		P_{MAI}=Total/Ns		4961.60

Table 3 Sum of Aperiodic Cross Correlation of Kronecker Sequences (Balanced gold sequence as inner and Balanced m-sequence as outer) to find the Mean Power of Multiple Access when r = 4

r = 4 Balanced m-sequence is inner

Balanced gold sequence is outer

Pairs of Sequence	Cak	Cai	Sum $C^2_{ak,ai}$	Sum $C^2_{ak,ai}$ / N
1	Kmg4_1	Kmg4_2	3054460	3393.84
2	Kmg4_1	Kmg4_3	3286300	3651.44
3	Kmg4_1	Kmg4_4	2492380	2769.31
4	Kmg4_1	Kmg4_5	2660060	2955.62
5	Kmg4_1	Kmg4_6	3286300	3651.44
6	Kmg4_1	Kmg4_7	2492380	2769.31
7	Kmg4_1	Kmg4_8	2660060	2955.62
8	Kmg4_1	Kmg4_9	3286300	3651.44
9	Kmg4_2	Kmg4_3	6538780	7265.31
10	Kmg4_2	Kmg4_4	7560220	8400.24
11	Kmg4_2	Kmg4_5	6543900	7271.00
12	Kmg4_2	Kmg4_6	6538780	7265.31
13	Kmg4_2	Kmg4_7	7560220	8400.24
14	Kmg4_2	Kmg4_8	6543900	7271.00
15	Kmg4_2	Kmg4_9	5546460	6162.73
16	Kmg4_3	Kmg4_4	4554940	5061.04
17	Kmg4_3	Kmg4_5	5196540	5773.93
18	Kmg4_3	Kmg4_6	5546460	6162.73
19	Kmg4_3	Kmg4_7	4554940	5061.04
20	Kmg4_3	Kmg4_8	5196540	5773.93
21	Kmg4_3	Kmg4_9	5546460	6162.73
22	Kmg4_4	Kmg4_5	4007420	4452.69
23	Kmg4_4	Kmg4_6	4554940	5061.04
24	Kmg4_4	Kmg4_7	7313020	8125.58
25	Kmg4_4	Kmg4_8	4007420	4452.69
26	Kmg4_4	Kmg4_9	5196540	5773.93
27	Kmg4_5	Kmg4_6	5196540	5773.93
28	Kmg4_5	Kmg4_7	4007420	4452.69

29	Kmg4_5	Kmg4_8	6306940	7007.71
30	Kmg4_5	Kmg4_9	5196540	5773.93
31	Kmg4_6	Kmg4_7	4554940	5061.04
32	Kmg4_6	Kmg4_8	5196540	5773.93
33	Kmg4_6	Kmg4_9	5546460	6162.73
34	Kmg4_7	Kmg4_8	4007420	4452.69
35	Kmg4_7	Kmg4_9	4554940	5061.04
36	Kmg4_8	Kmg4_9	5196540	5773.93
		Total	**170293460**	194988.89
		P_{MAI}=Total/Ns		5416.36

Table 4 Sum of Aperiodic Cross Correlation of Kronecker Sequences (Balanced m-sequence as inner and Balanced gold sequence as outer) to find the Mean Power of Multiple Access when r = 4

r = 5 Balanced gold sequence is inner

Balanced m-sequence is outer

Pairs of Sequence	Cak	Cai	Sum $C^2_{ak,ai}$	Sum $C^2_{ak,ai}$ / N
1	Kgm5_1	Kgm5_2	40232220	10466.238
2	Kgm5_1	Kgm5_3	35617500	9265.739
3	Kgm5_1	Kgm5_4	43442908	11301.485
4	Kgm5_1	Kgm5_5	43442908	11301.485
5	Kgm5_1	Kgm5_6	46944540	12212.419
6	Kgm5_1	Kgm5_7	38171356	9930.113
7	Kgm5_1	Kgm5_8	53237340	13849.464
8	Kgm5_1	Kgm5_9	47564636	12373.735
9	Kgm5_2	Kgm5_3	63515580	16523.304
10	Kgm5_2	Kgm5_4	34842748	9064.190
11	Kgm5_2	Kgm5_5	34842748	9064.190
12	Kgm5_2	Kgm5_6	45685980	11885.010
13	Kgm5_2	Kgm5_7	42366556	11021.477
14	Kgm5_2	Kgm5_8	33729660	8774.625

15	Kgm5_2	Kgm5_9	53647676	13956.211
16	Kgm5_3	Kgm5_4	37606364	9783.133
17	Kgm5_3	Kgm5_5	37606364	9783.133
18	Kgm5_3	Kgm5_6	45019964	11711.749
19	Kgm5_3	Kgm5_7	39402364	10250.355
20	Kgm5_3	Kgm5_8	22886428	5953.805
21	Kgm5_3	Kgm5_9	49927100	12988.319
22	Kgm5_4	Kgm5_5	68493244	17818.222
23	Kgm5_4	Kgm5_6	56492476	14696.274
24	Kgm5_4	Kgm5_7	46653596	12136.732
25	Kgm5_4	Kgm5_8	40916604	10644.278
26	Kgm5_4	Kgm5_9	45457852	11825.664
27	Kgm5_5	Kgm5_6	56492476	14696.274
28	Kgm5_5	Kgm5_7	46653596	12136.732
29	Kgm5_5	Kgm5_8	40916604	10644.278
30	Kgm5_5	Kgm5_9	45457852	11825.664
31	Kgm5_6	Kgm5_7	41536700	10805.593
32	Kgm5_6	Kgm5_8	42959100	11175.624
33	Kgm5_6	Kgm5_9	36046204	9377.264
34	Kgm5_7	Kgm5_8	27419964	7133.185
35	Kgm5_7	Kgm5_9	43233148	11246.917
36	Kgm5_8	Kgm5_9	68914236	17927.741
		Total	**1597376592**	397622.88
		P_{MAI}=Total/Ns		11045.08

Table 5 Sum of Aperiodic Cross Correlation of Kronecker Sequences (Balanced gold sequence as inner and Balanced m-sequence as outer) to find the Mean Power of Multiple Access when r = 5

r = 5 Balanced m-sequence is inner

Balanced gold sequence is outer

Pairs of Sequence	Cak	Cai	Sum $C^2_{ak,ai}$	Sum $C^2_{ak,ai}$ / N
1	Kmg5_1	Kmg5_2	50547740	13149.776
2	Kmg5_1	Kmg5_3	46971350	12219.394
3	Kmg5_1	Kmg5_4	47472220	12349.693
4	Kmg5_1	Kmg5_5	47472220	12349.693
5	Kmg5_1	Kmg5_6	54490972	14175.591
6	Kmg5_1	Kmg5_7	45823452	11920.773
7	Kmg5_1	Kmg5_8	60541852	15749.701
8	Kmg5_1	Kmg5_9	67494748	17558.467
9	Kmg5_2	Kmg5_3	68057404	17704.840
10	Kmg5_2	Kmg5_4	42359420	11019.620
11	Kmg5_2	Kmg5_5	42359420	11019.620
12	Kmg5_2	Kmg5_6	54715676	14234.047
13	Kmg5_2	Kmg5_7	48422492	12596.902
14	Kmg5_2	Kmg5_8	46088764	11989.793
15	Kmg5_2	Kmg5_9	62769980	16329.339
16	Kmg5_3	Kmg5_4	42848092	11146.746
17	Kmg5_3	Kmg5_5	42848092	11146.746
18	Kmg5_3	Kmg5_6	55670012	14482.313
19	Kmg5_3	Kmg5_7	47414012	12334.550
20	Kmg5_3	Kmg5_8	34808668	9055.325
21	Kmg5_3	Kmg5_9	56943804	14813.685
22	Kmg5_4	Kmg5_5	81751100	21267.196
23	Kmg5_4	Kmg5_6	65689852	17088.931
24	Kmg5_4	Kmg5_7	46549020	12109.527
25	Kmg5_4	Kmg5_8	47163580	12269.402
26	Kmg5_4	Kmg5_9	55652412	14477.735
27	Kmg5_5	Kmg5_6	65689852	17088.931
28	Kmg5_5	Kmg5_7	46549020	12109.527

29	Kmg5_5	Kmg5_8	47163580	12269.402
30	Kmg5_5	Kmg5_9	55652412	14477.735
31	Kmg5_6	Kmg5_7	47082364	12248.274
32	Kmg5_6	Kmg5_8	53026300	13794.563
33	Kmg5_6	Kmg5_9	47558844	12372.228
34	Kmg5_7	Kmg5_8	30286076	7878.792
35	Kmg5_7	Kmg5_9	49853308	12969.123
36	Kmg5_8	Kmg5_9	75712380	19696.249
		Total	**1881500490**	469767.98
		P_{MAI}=Total/Ns		13049.11

Table 6 Sum of Aperiodic Cross Correlation of Kronecker Sequences (Balanced m-sequence as inner and Balanced gold sequence as outer) to find the Mean Power of Multiple Access when r = 5

4.4.2 Numerical result of BER

$$BER = \frac{1}{2}\,\mathrm{erfc}\left(\frac{\Delta}{\sqrt{2}}\right)$$

where Δ is given by

$$\Delta = \frac{N}{\sqrt{\sum_{k=2}^{k}\sigma^2_k}} = \frac{N}{\sqrt{(K-1)\sigma^2_k}}$$

The BER of Kronecker sequences are based on balanced gold sequence as inner and balanced m-sequence as outer . We use nine kronecker sequences to calculate the result shown as below.

The lenght of sequence (N) = 196
The stage of LFSR (r) = 3
P_{MAI} = 948.42
K= number of simultaneous users

K	BER
2	9.80×10^{-11}
3	3.39×10^{-6}
4	1.19×10^{-4}
5	7.31×10^{-4}
6	2.20×10^{-3}
7	4.70×10^{-3}
8	8.10×10^{-3}
9	1.20×10^{-2}
10	1.60×10^{-2}
11	2.21×10^{-2}
12	2.75×10^{-2}
13	3.31×10^{-2}
14	3.88×10^{-2}
15	4.45×10^{-2}
16	5.02×10^{-2}
17	5.58×10^{-2}
18	6.13×10^{-2}
19	6.68×10^{-2}
20	7.21×10^{-2}

Table 7 BER of Kronecker Sequences (r=3)
(Balanced gold sequence as inner and Balanced m-sequence as outer)

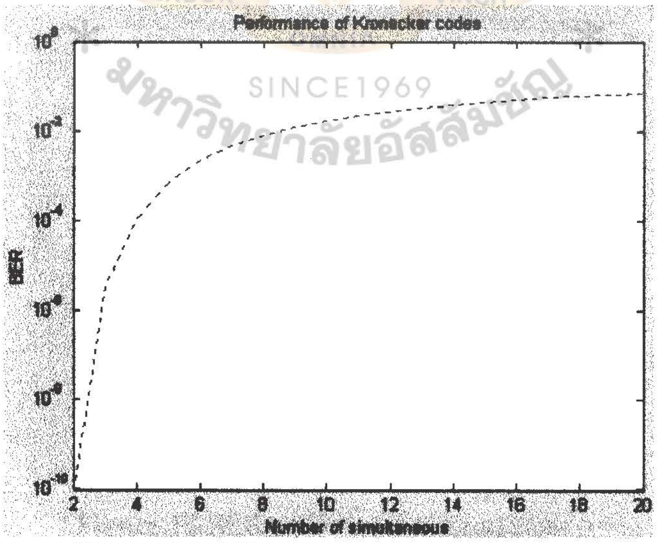


Figure 14 BER of Kronecker Sequences (r=3)
(Balanced gold sequence as inner and balanced m-sequence as outer)

The lenght of sequence (N) = 196
The stage of LFSR (r) = 3
P_{MAI} = 987.16
K= number of simultaneous users

K	BER
2	2.22×10^{-10}
3	5.14×10^{-6}
4	1.58×10^{-4}
5	9.07×10^{-4}
6	2.60×10^{-3}
7	5.40×10^{-3}
8	9.20×10^{-3}
9	1.37×10^{-2}
10	1.80×10^{-2}
11	2.40×10^{-2}
12	3.00×10^{-2}
13	3.59×10^{-2}
14	4.18×10^{-2}
15	4.77×10^{-2}
16	5.36×10^{-2}
17	5.94×10^{-2}
18	6.51×10^{-2}
19	7.07×10^{-2}
20	7.62×10^{-2}

Table 8 BER of Kronecker Sequences (r=3)
(Balanced m-sequence as inner and balanced gold sequence as outer)

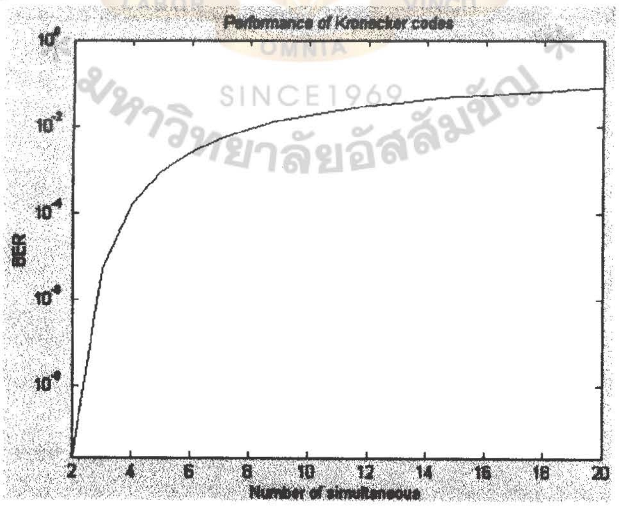


Figure 15 BER of Kronecker Sequences (r=3)
(Balanced m-sequence as inner and balanced gold sequence as outer)

The lenght of sequence (N) = 900
The stage of LFSR (r) = 4
P_{MAI} = 4961.6
K= number of simultaneous users

K	BER
2	1.1×10^{-37}
3	8.22×10^{-20}
4	8.10×10^{-14}
5	8.37×10^{-11}
6	5.51×10^{-9}
7	9.13×10^{-8}
8	6.85×10^{-7}
9	3.13×10^{-6}
10	1.03×10^{-5}
11	2.67×10^{-5}
12	5.85×10^{-5}
13	1.13×10^{-4}
14	1.97×10^{-4}
15	3.19×10^{-4}
16	4.85×10^{-4}
17	7.01×10^{-4}
18	9.71×10^{-4}
19	1.30×10^{-3}
20	1.70×10^{-3}

Table 9 BER of Kronecker Sequences (r=4)
(Balanced gold sequence as inner and balanced m-sequence as outer)

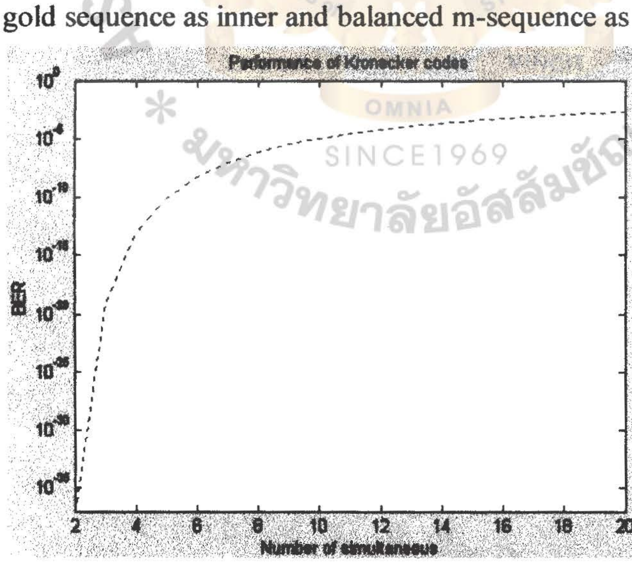


Figure 16 BER of Kronecker Sequences (r=4)
(Balanced gold sequence as inner and balanced m-sequence as outer)

The lenght of sequence (N) = 900
The stage of LFSR (r) = 4
P_{MAI} = 5416.36
K= number of simultaneous users

K	BER
2	1.01×10^{-34}
3	2.64×10^{-18}
4	8.30×10^{-13}
5	4.84×10^{-10}
6	2.26×10^{-8}
7	2.98×10^{-7}
8	1.89×10^{-6}
9	7.68×10^{-6}
10	2.29×10^{-5}
11	5.51×10^{-5}
12	1.13×10^{-4}
13	2.08×10^{-4}
14	3.47×10^{-4}
15	5.41×10^{-4}
16	7.96×10^{-4}
17	1.10×10^{-3}
18	1.50×10^{-3}
19	2.00×10^{-3}
20	2.50×10^{-3}

Table 10 BER of Kronecker Sequences (r=4)
(Balanced m-sequence as inner and balanced gold sequence as outer)

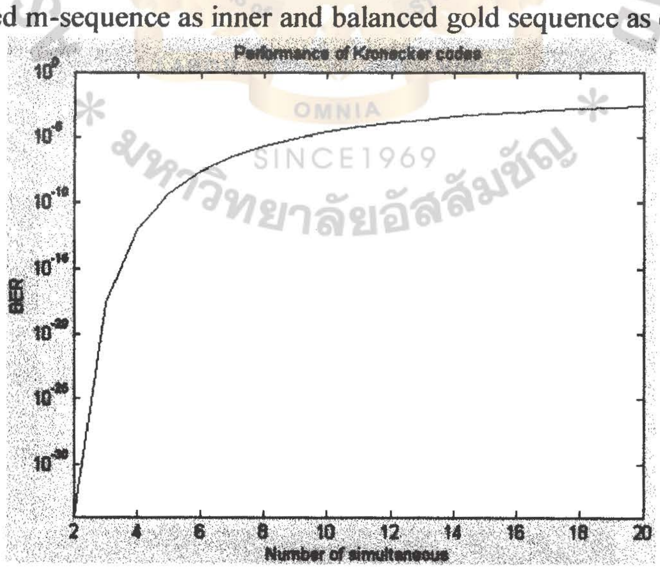


Figure 17 BER of Kronecker Sequences (r=4)
(Balanced m-sequence as inner and balanced gold sequence as outer)

The lenght of sequence (N) = 3844
The stage of LFSR (r) = 5
P_{MAI} = 11045.08
K= number of simultaneous users

K	BER
2	3.41×10^{-293}
3	6.62×10^{-148}
4	2.76×10^{-97}
5	5.15×10^{-75}
6	1.93×10^{-60}
7	1.02×10^{-50}
8	9.07×10^{-44}
9	1.49×10^{-38}
10	1.71×10^{-34}
11	3.05×10^{-31}
12	1.39×10^{-28}
13	2.32×10^{-26}
14	1.75×10^{-24}
15	7.18×10^{-23}
16	1.79×10^{-21}
17	3.01×10^{-20}
18	3.62×10^{-19}
19	3.32×10^{-18}
20	2.41×10^{-17}

Table 11 BER of Kronecker Sequences (r=5)
(Balanced gold sequence as inner and balanced m-sequence as outer)

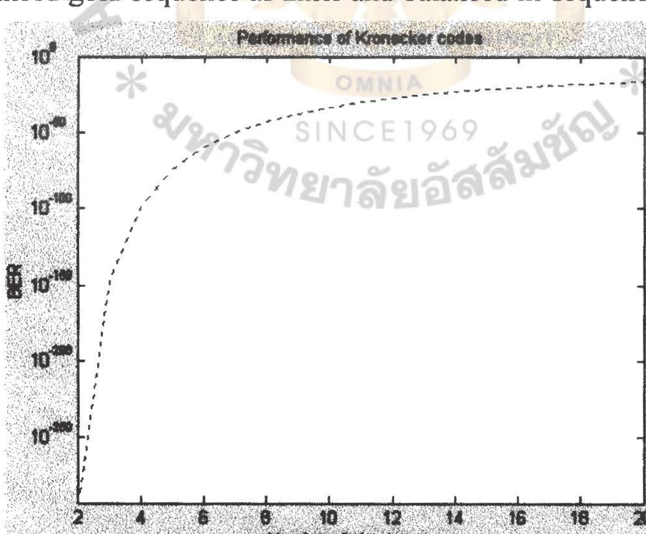


Figure 18 BER of Kronecker Sequences (r=5)
(Balanced gold sequence as inner and balanced m-sequence as outer)

The lenght of sequence (N) = 3844
The stage of LFSR (r) = 5
P_{MAI} = 13049.11
K= number of simultaneous users

K	BER
2	1.53×10^{-248}
3	1.90×10^{-125}
4	2.23×10^{-84}
5	7.96×10^{-64}
6	1.75×10^{-51}
7	3.01×10^{-43}
8	2.33×10^{-37}
9	6.11×10^{-33}
10	1.69×10^{-29}
11	9.58×10^{-27}
12	1.73×10^{-24}
13	1.31×10^{-22}
14	5.15×10^{-21}
15	1.19×10^{-19}
16	1.83×10^{-18}
17	2.00×10^{-17}
18	1.66×10^{-16}
19	1.08×10^{-15}
20	5.82×10^{-15}

Table 12 BER of Kronecker Sequences (r=5)
(Balanced m-sequence as inner and balanced gold sequence as outer)

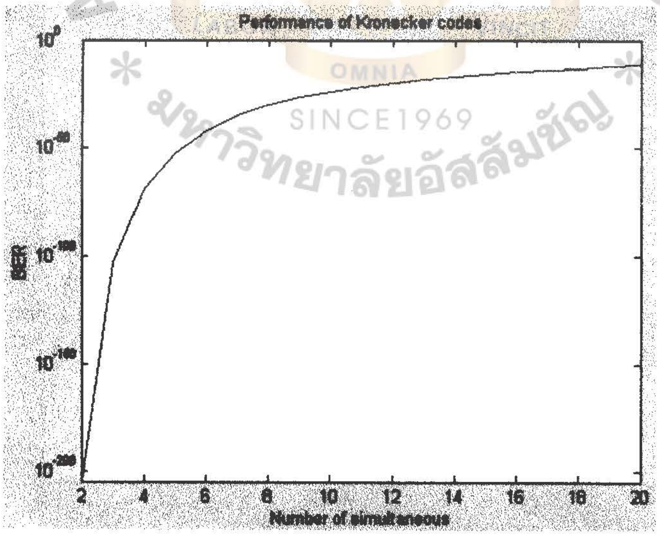


Figure 19 BER of Kronecker Sequences (r=5)
(Balanced m-sequence as inner and balanced gold sequence as outer)

4.5 Comparison performance between combination of Kronecker sequences

- The Kronecker sequences are constructed by

a. balanced gold sequences as inner and balanced m-sequences as outer (- -)

b. balanced m-sequences as inner and balanced gold sequences as outer (-)

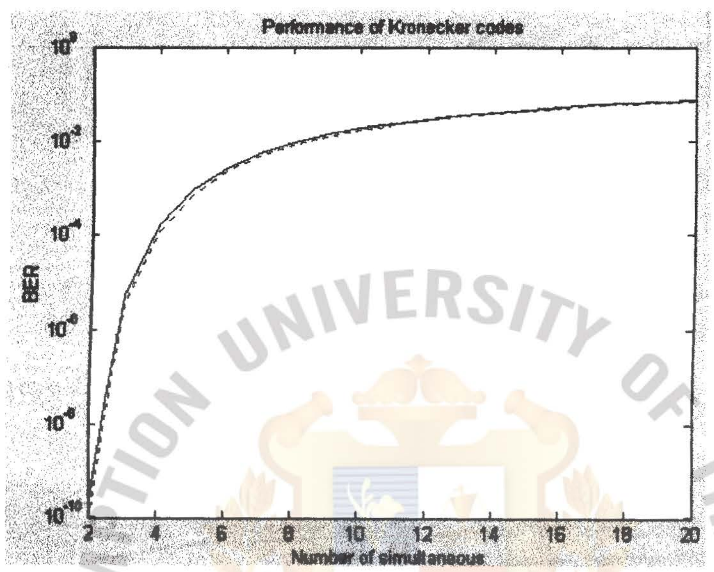


Figure 20 Comparison BER of Kronecker Sequences (r=3)

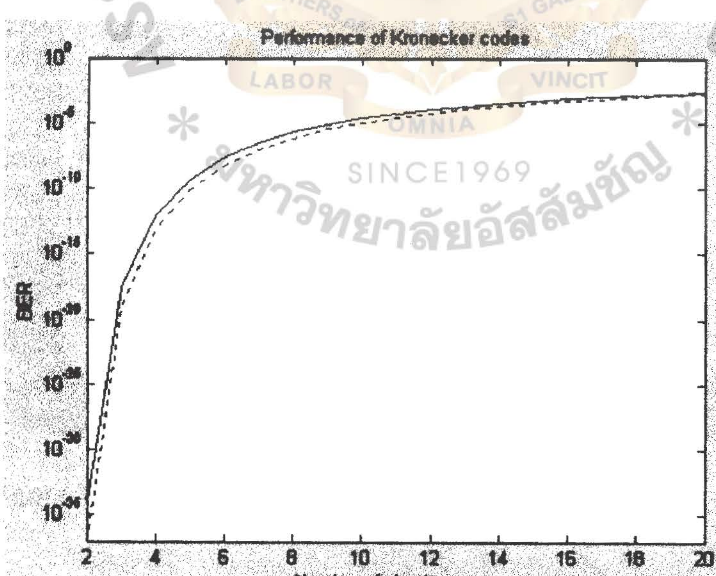


Figure 21 Comparison BER of Kronecker Sequences (r=4)

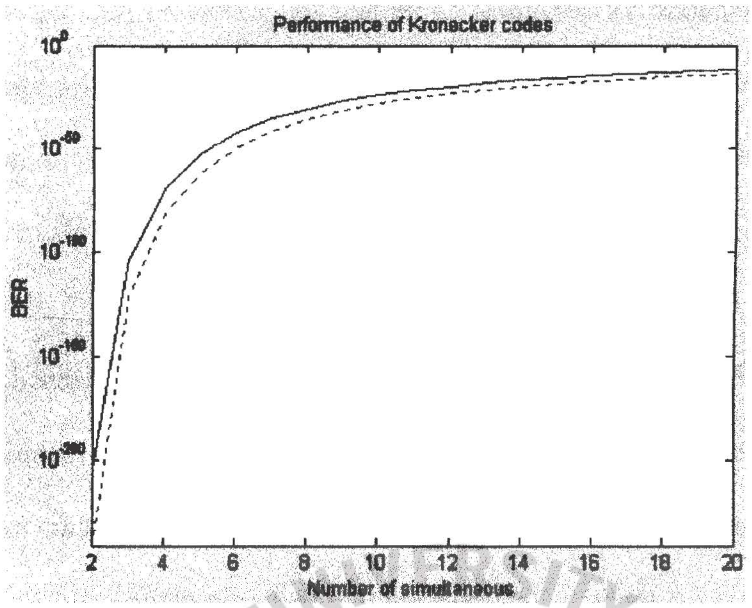


Figure 22 Comparison BER of Kronecker Sequences (r=5)

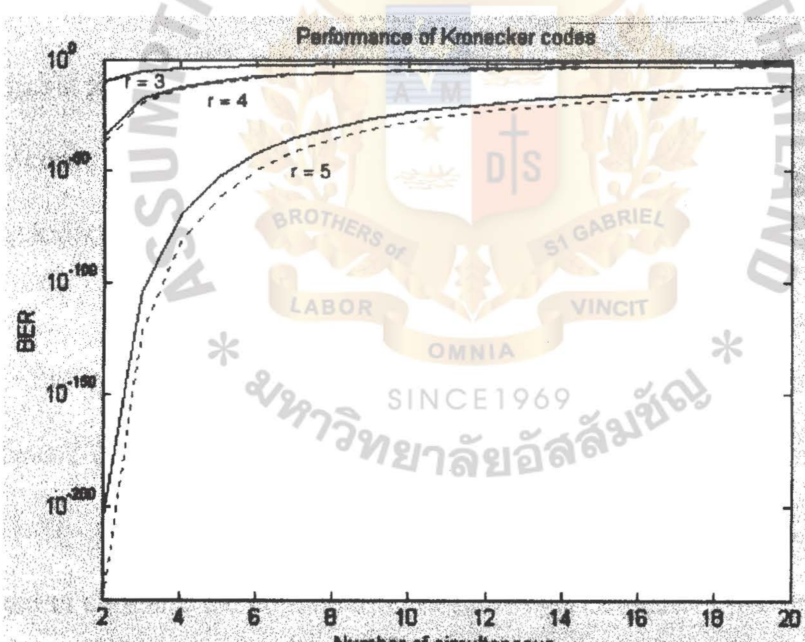


Figure 23 Comparison BER of Kronecker Sequences (r=3, 4, 5)

Chapter IV Conclusion

There are many CDMA techniques that are investigated for optical LANs. In order to accommodate a large number of users on CDMA networks, long sequence requiring large transmission bandwidths are needed. For this reason, incoherent optical CDMA over single mode optical fibers hold out the promise of very long sequences.

The first scheme for low-weight sequences use OOK to modulate sequence. But it has the limitation of number of available sequences with good correlation properties. After researching for new techniques, the SIK can solve this problem. However, it still have a problem about high loss requiring optical switches at very high speed.

The concatenated codes (Kronecker codes) can help this problem about high loss and short available sequence in very high-speed switch and synchronization. Recently, the kronecker codes have only been constructed from Lampel codes and Gold codes. The generating kronecker sequences are complicated.

Consequently, we proposed another method for generating kronecker sequences by using balanced m-sequence and balanced gold sequences. The performance is based on MATLAB program to evaluate. To find the best combination, this work investigated to compare the combination of kronecker sequences that are generated by balanced gold sequence as inner and balanced m-sequences as outer. Another combination is balanced m-sequence as inner and balanced gold sequences as outer.

From the numerical result, the outputs are generated from the 9 kroneckers sequences. Due to the limitation of computer unit processing, the computer is Pentium II 450 that can generate the sequences codes in maximum for r = 3, 4, 5. To calculate the mean power multiple access, the pairs of difference in kronecker sequences are 36 sequences. The stage of LFSR is assigned for 3, 4 and 5 to generate kronecker sequences. The comparing of each stage for combination shows the BER values to evaluate the best combination. The result of BER for kronecker sequences that are constructed from balanced m-sequences as inner and balanced gold sequence as outer is higher than the combination of balanced gold sequence as inner and balanced m-sequences as outer. When the kronecker sequences have more stages, the BER graph will show the increased gap between the BER values of two combination sequences. The simultaneous users can gain more access to the system when the length of sequence is increased.

Therefore, the kroneckers that are combined from balanced gold sequence as inner and balanced m-sequences as outer should be used for generating the inner and outer sequences.

References

[1] PARHAM M.J. et al.(1992) "Code division multiple-access techniques for use in optical-ffibre local-area networks", Electron & Commun Engineering Journal, August 1992, pp. 203-212.

[2] O'FARRELL T. and LOCHMANN S. (1994) "Performance analysis of an optical correlator receiver for SIK DS-CDMA communication systems",Elect Lett,1994, Vol. 30, No.1, pp.63-65.

[3] O'FARRELL T. and LOCHMANN S.I (1995) "Switch correlator receiver architecture for optical CDMA networks with bipolar capacity", Elect Lett,1994, Vol.30, No.1, pp.63-65.

[4] Raymond L. Pickhol, Donald L. Schilling and Laurence B. Milstein, " Theory of Spread-Spectrum Communications- A tutorial", IEEE Trans. Commun., Vol. Com30, No.5May 1982.

[5] Andrew Stok and Edward H. Sargent,"lighting the Local Area : Optical Code Division Multiple Access and Quality of Service Provisioning", IEEE Network November/December 2000 pp.42-46.

[6] P.R. Prucnal, M.A. Santoro, and T.R. Fan, "Spread spectrum Fiber-Optic Local Area Network Using Optical Processing", IEEE J. Lightwave Tech., Vol. LT-4,No. 5, May 1986, pp.547-54.

[7] J.A. Salehi, "Code division Multiple-Access Techniques in Optical Fiber Networks – Part I: Fundamental Principles",IEEE Trans.Commun.,Vol 37, No.8, Aug. 1989, pp.824-33

[8] J.A. Salehi and C.A. Brackkett, "Code division Multiple-Access Techniques in Optical Fiber Networks – Part II: Systems Performance Analysis",IEEE Trans. Commun., Vol 37, No.8, Aug. 1989, pp.834-42.

[9] Leslie A. Rusch and H. Vincent Poor "Effect of Laser Phase Drifte on Coherent Optical CDMA" IEEE J. on Selected Area in Communications, special issue on High Speed Point-to-Point Optical Communications Systems.

[10] Joseph E. Geusic, William B. Bridges "Coherent Optical Sources for Communications" IEEE J. Vol. 58,No.10, October 1970, pp.1419 - 1439.

[11] Pham Manh Lam, "Study of a Novel Hybrid Optical-Electronic Code Division Multiple Access Local Area Network Based on Kronecker Sequences", Ph.D. Dissertation, Telecommunication Program, Asian Institute of Technology, Thailand, 1998.

Appendix A The generating codes for Kronecker sequences

Example

$\{C_i(l)\}$; balanced m-sequence {-1 1 -1 1 1 1 -1} as inner sequences

$\{D_i(j)\}$; balanced m-sequence { 1 -1 -1 1 -1 1 1} as outer sequences

$\{A_i(m)\}$; kronecker sequence = { inner codes * 1, inner codes * -1, inner codes * -1,
inner codes * 1, inner codes * -1, inner codes * 1,
inner codes * 1}

={ (-1 1 -1 1 1 1 -1) * 1, (-1 1 -1 1 1 1 -1) * -1,
(-1 1 -1 1 1 1 -1) * -1, (-1 1 -1 1 1 1 -1) * 1,
(-1 1 -1 1 1 1 -1) * -1, (-1 1 -1 1 1 1 -1) * 1,
(-1 1 -1 1 1 1 -1) * 1}

={-1 1 -1 1 1 1 -1, 1 -1 1 -1 -1 -1 1, 1 -1 1 -1 -1 -1 1,
-1 1 -1 1 1 1 -1, 1 -1 1 -1 -1 -1 1,-1 1 -1 1 1 1 -1,
-1 1 -1 1 1 1 -1}

Example of generating kronecker sequence from polynomial

Find m-sequence generated by ; Primitive Polynomial $f_1(x) = x^3 + x^2 + 1$

Another polynomial is Reciprocal Polynomial for new m-sequence

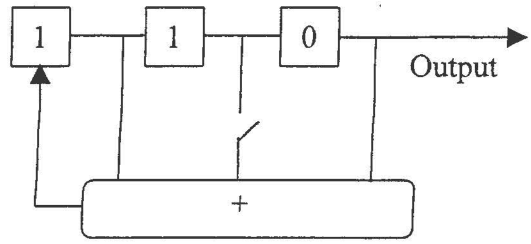
$$f(x) = x^3 + x^2 + 1$$

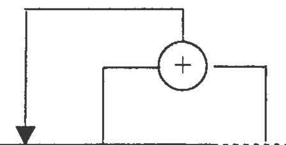
$$x^3 f\left(\frac{1}{x}\right) = x^3 \left[\left(\frac{1}{x}\right)^3 + \left(\frac{1}{x}\right)^2 + 1\right]$$

$$= 1 + x + x^3$$

The Reciprocal polynomial is $f_2(x) = x^3 + x + 1$

1) Using LFSR to generate $f_1(x) = x^3 + x^2 + 1$ for m_1-sequence

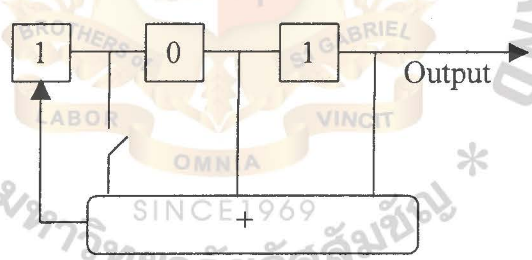


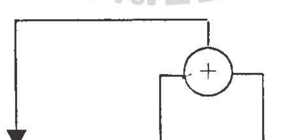


1	1	0	
1	1	1	0
0	1	1	1
1	0	1	1
0	1	0	1
0	0	1	0
1	0	0	1
1	1	0	0

m_1-sequence is 0 1 0 1 1 1 0

2) Using LFSR to generate $f_2(x) = x^3 + x + 1$ for m_2-sequence





1	0	1	
1	1	0	1
1	1	1	0
0	1	1	1
0	0	1	1
1	0	0	1
0	1	0	0
1	0	1	0

m_2-sequence is 0 0 1 1 1 0 1

Gold Sequence:

We can using two m-sequence m_1 and m_2, we generate gold sequences.

$g_j(i) = a(i) \oplus T^j{}_b(i)$; for $0 \le i \le N-1$

m_1 ; 0 1 0 1 1 1 0

$m^0{}_2$; 0 0 1 1 1 0 1

$g_1 = m_1 + m^0{}_2$ = 0 1 1 0 0 1 1

m_1 ; 0 1 0 1 1 1 0

$m^1{}_2$; 0 1 1 1 0 1 0

$g_2 = m_1 + m^1{}_2$ = 0 0 1 0 1 0 0

m_1 ; 0 1 0 1 1 1 0

$m^2{}_2$; 1 1 1 0 1 0 0

$g_3 = m_1 + m^2{}_2$ = 1 0 1 1 0 1 0

m_1 ; 0 1 0 1 1 1 0

$m^3{}_2$; 1 1 0 1 0 0 1

$g_4 = m_1 + m^3{}_2$ = 1 0 0 0 1 1 1

m_1 ; 0 1 0 1 1 1 0

$m^4{}_2$; 1 0 1 0 0 1 1

$g_5 = m_1 + m^4{}_2$ = 1 1 1 1 1 0 1

m_1 ; 0 1 0 1 1 1 0

$m^5{}_2$; 0 1 0 0 1 1 1

$g_6 = m_1 + m^5{}_2$ = 0 0 0 1 0 0 1

m_1 ; 0 1 0 1 1 1 0

m^5_2 ; 1 0 0 1 1 1 0

$g_7 = m_1 + m^5_2$ = 1 1 0 0 0 0 0

$g_8 = m_1$ = 0 1 0 1 1 1 0

$g_9 = m_2$ = 0 0 1 1 1 0 1

Bipolar m-sequence and bipolar Gold sequence

After we finished to generate m-sequence and gold sequence, these codes are unipolar (0,1). For changing to bipolar sequences, we can replace 0 by 1 and 1 by -1. Firstly, we change 7 chips of m-sequence to bipolar. There is m_1-sequence (0 1 0 1 1 1 0) that is changed to bipolar m-sequence 1 -1 1 -1 -1 -1 1. Then we change all of gold sequences to bipolar gold sequence.

g_1 = 0 1 1 0 0 1 1 ⟶ 1 -1 -1 1 1 -1 -1

g_2 = 0 0 1 0 1 0 0 ⟶ 1 1 -1 1 -1 1 1

g_3 = 1 0 1 1 0 1 0 ⟶ -1 1 -1 -1 1 -1 1

g_4 = 1 0 0 0 1 1 1 ⟶ -1 1 1 1 -1 -1 -1

g_5 = 1 1 1 1 1 0 1 ⟶ -1 -1 -1 -1 -1 1 -1

g_6 = 0 0 0 1 0 0 1 ⟶ 1 1 1 -1 1 1 -1

g_6 = 0 0 0 1 0 0 1 ⟶ 1 1 1 -1 1 1 -1

g_7 = 1 1 0 0 0 0 0 ⟶ -1 -1 1 1 1 1 1

g_8 = 0 1 0 1 1 1 0 ⟶ 1 -1 1 -1 -1 -1 1

g_9 = 0 0 1 1 1 0 1 ⟶ 1 1 -1 -1 -1 1 -1

From a bipolar m-sequence and bipolar gold sequences, we can get balanced m-sequence and balanced gold sequence form replacing "1" by [1,-1] and "-1" by [-1,1] such that bipolar m-sequence is 1 -1 1 -1 -1 -1 1 is changed to balanced m-sequence 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1.

Bipolar gold sequence		Balanced gold sequence
g_1 = 1 -1 -1 1 1 -1 -1	→	1 -1 -1 1 -1 1 1 -1 1 -1 -1 1 -1 1
g_2 = 1 1 -1 1 -1 1 1	→	1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1
g_3 = -1 1 -1 -1 1 -1 1	→	-1 1 1 -1 -1 1 -1 1 1 -1 -1 1 1 -1
g_4 = -1 1 1 1 -1 -1 -1	→	-1 1 1 -1 1 -1 1 -1 -1 1 -1 1 -1 1
g_5 = -1 -1 -1 -1 -1 1 -1	→	-1 1 -1 1 -1 1 -1 1 -1 1 1 -1 -1 1
g_6 = 1 1 1 -1 1 1 -1	→	1 -1 1 -1 1 -1 -1 1 1 -1 1 -1 -1 1
g_6 = 1 1 1 -1 1 1 -1	→	1 -1 1 -1 1 -1 -1 1 1 -1 1 -1 -1 1
g_7 = -1 -1 1 1 1 1 1	→	-1 1 -1 1 1 -1 1 -1 1 -1 1 -1 1 -1
g_8 = 1 -1 1 -1 -1 -1 1	→	1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1
g_9 = 1 1 -1 -1 -1 1 -1	→	1 -1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1

Kronecker sequence

Inner Sequence = balanced m-sequence

Outer sequence = balanced gold sequence

Balanced m-sequence = 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1

Balanced gold sequence g_1 = 1 -1 -1 1 -1 1 1 -1 1 -1 -1 1 -1 1

Kronecker Sequence k_1 = [(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x -1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x -1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x -1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x -1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x -1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x -1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x -1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1]

k_1 = 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1

Kronecker Sequence k_2 = [(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x -1]

k_2 = 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

Kronecker Sequence k_3 = [(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x -1]

k_3 = -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

Kronecker Sequence k_4 = [(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1]

k_4 = -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1
-1 1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

Kronecker Sequence k_5 = [(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1]

k_5 = -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1

Kronecker Sequence k_6 = [(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1]

k_6 = 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1

Kronecker Sequence k_7 = [(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x -1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x -1]

k_7 = -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

Kronecker Sequence k_8 = [(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x −1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x −1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x −1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x −1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x −1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x −1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x -1]

k_8 = 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 −1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 −1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 −1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 −1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

Kronecker Sequence k_9 = [(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x –1

(1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1) x 1]

k_9 = 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 –1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1

1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1

-1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1

Appendix B The List Name of files that are generated

No.	Balanced gold sequence
1	Bg3_1
2	Bg3_2
3	Bg3_3
4	Bg3_4
5	Bg3_5
6	Bg3_6
7	Bg3_7
8	Bg3_8
9	Bg3_9

Table 13 Filename of Balanced Gold sequences when the stage of LFSR (r) = 3

No.	Balanced gold sequence
1	Bg4_1
2	Bg4_2
3	Bg4_3
4	Bg4_4
5	Bg4_5
6	Bg4_6
7	Bg4_7
8	Bg4_8
9	Bg4_9
10	Bg4_10
11	Bg4_11
12	Bg4_12
13	Bg4_13

14	Bg4_14
15	Bg4_15
16	Bg4_16
17	Bg4_17

Table 14 Filename of Balanced Gold sequences when the stage of LFSR (r) = 4

No.	Balanced gold sequence
1	Bg5_1
2	Bg5_2
3	Bg5_3
4	Bg5_4
5	Bg5_5
6	Bg5_6
7	Bg5_7
8	Bg5_8
9	Bg5_9
10	Bg5_10
11	Bg5_11
12	Bg5_12
13	Bg5_13
14	Bg5_14
15	Bg5_15
16	Bg5_16
17	Bg5_17
18	Bg5_18
19	Bg5_19
20	Bg5_20

21	Bg5_21
22	Bg5_22
23	Bg5_23
24	Bg5_24
25	Bg5_25
26	Bg5_26
27	Bg5_27
28	Bg5_28
29	Bg5_29
30	Bg5_30
31	Bg5_31
32	Bg5_32
33	Bg5_33

Table 15 Filename of Balanced Gold sequences when the stage of LFSR (r) = 5

No.	Aperiodic Auto Correlation Balanced gold sequence
1	ABg3_1
2	ABg3_2
3	ABg3_3
4	ABg3_4
5	ABg3_5
6	ABg3_6
7	ABg3_7
8	ABg3_8
9	ABg3_9

Table 16 Filename of Aperiodic Auto Correlation of Balanced Gold sequences when the stage of LFSR (r) = 3

No.	Balanced gold sequence
1	ABg4_1
2	ABg4_2
3	ABg4_3
4	ABg4_4
5	ABg4_5
6	ABg4_6
7	ABg4_7
8	ABg4_8
9	ABg4_9
10	ABg4_10
11	ABg4_11
12	ABg4_12
13	ABg4_13
14	ABg4_14
15	ABg4_15
16	ABg4_16
17	ABg4_17

Table 17 Filename of Aperiodic Auto Correlation of Balanced Gold sequences when the stage of LFSR (r) = 4

No.	Balanced gold sequence
1	ABg5_1
2	ABg5_2
3	ABg5_3
4	ABg5_4
5	ABg5_5
6	ABg5_6
7	ABg5_7

8	ABg5_8
9	ABg5_9
10	ABg5_10
11	ABg5_11
12	ABg5_12
13	ABg5_13
14	ABg5_14
15	ABg5_15
16	ABg5_16
17	ABg5_17
18	ABg5_18
19	ABg5_19
20	ABg5_20
21	ABg5_21
22	ABg5_22
23	ABg5_23
24	ABg5_24
25	ABg5_25
26	ABg5_26
27	ABg5_27
28	ABg5_28
29	ABg5_29
30	ABg5_30
31	ABg5_31
32	ABg5_32
33	ABg5_33

Table 18 Filename of Aperiodic Auto Correlation of Balanced Gold sequences when the stage of LFSR (r) = 5

Kronecker sequence r=3			
Balanced gold sequences as inner			
Balanced m-sequences as outer			
No.	**Inner**	**Outer**	**Kronecker**
1	ABg3_1	ABm3	Kgm3_1
2	ABg3_2	ABm3	Kgm3_2
3	ABg3_3	ABm3	Kgm3_3
4	ABg3_4	ABm3	Kgm3_4
5	ABg3_5	ABm3	Kgm3_5
6	ABg3_6	ABm3	Kgm3_6
7	ABg3_7	ABm3	Kgm3_7
8	ABg3_8	ABm3	Kgm3_8
9	ABg3_9	ABm3	Kgm3_9

Table 19 Filename of Aperiodic Auto Correlation of Kronecker sequences (balanced gold sequence as inner and balanced m-sequenced as outter) when the stage of LFSR (r) = 3

Kronecker sequence r=3			
Balanced m-sequences as inner			
Balanced gold sequences as outer			
No.	**Inner**	**Outer**	**Kronecker**
1	ABm3	ABg3_1	Kmg3_1
2	ABm3	ABg3_2	Kmg3_2
3	ABm3	ABg3_3	Kmg3_3
4	ABm3	ABg3_4	Kmg3_4
5	ABm3	ABg3_5	Kmg3_5
6	ABm3	ABg3_6	Kmg3_6
7	ABm3	ABg3_7	Kmg3_7
8	ABm3	ABg3_8	Kmg3_8
9	ABm3	ABg3_9	Kmg3_9

Table 20 Filename of Aperiodic Auto Correlation of Kronecker sequences (balanced m-sequence as inner and balanced gold sequenced as outter) when the stage of LFSR (r) = 3

Kronecker sequence r=4			
Balanced gold sequences as inner			
Balanced m-sequences as outer			
No.	**Inner**	**Outer**	**Kronecker**
1	ABg4_1	ABm4	Kgm4_1
2	ABg4_2	ABm4	Kgm4_2
3	ABg4_3	ABm4	Kgm4_3
4	ABg4_4	ABm4	Kgm4_4
5	ABg4_5	ABm4	Kgm4_5
6	ABg4_6	ABm4	Kgm4_6
7	ABg4_7	ABm4	Kgm4_7
8	ABg4_8	ABm4	Kgm4_8
9	ABg4_9	ABm4	Kgm4_9

Table 21 Filename of Aperiodic Auto Correlation of Kronecker sequences (balanced gold sequence as inner and balanced m-sequenced as outter) when the stage of LFSR (r) = 4

Kronecker sequence r=4			
Balanced m-sequences as inner			
Balanced gold sequences as outer			
No.	**Inner**	**Outer**	**Kronecker**
1	ABm4	ABg4_1	Kmg4_1
2	ABm4	ABg4_2	Kmg4_2
3	ABm4	ABg4_3	Kmg4_3
4	ABm4	ABg4_4	Kmg4_4
5	ABm4	ABg4_5	Kmg4_5
6	ABm4	ABg4_6	Kmg4_6
7	ABm4	ABg4_7	Kmg4_7
8	ABm4	ABg4_8	Kmg4_8
9	ABm4	ABg4_9	Kmg4_9

Table 22 Filename of Aperiodic Auto Correlation of Kronecker sequences (balanced m-sequence as inner and balanced gold sequenced as outter) when the stage of LFSR (r) = 4

The lenght of kronecker sequence

= The lenght of balanced gold sequences(N_1) x

The lenght of balanced m-sequences (N_2)

= $(N_1) \times (N_2) = 30 \times 30 = 900$

Kronecker sequence r=5			
Balanced gold sequences as inner			
Balanced m-sequences as outer			
No.	**Inner**	**Outer**	**Kronecker**
1	ABg5_1	ABm5	Kgm5_1
2	ABg5_2	ABm5	Kgm5_2
3	ABg5_3	ABm5	Kgm5_3
4	ABg5_4	ABm5	Kgm5_4
5	ABg5_5	ABm5	Kgm5_5
6	ABg5_6	ABm5	Kgm5_6
7	ABg5_7	ABm5	Kgm5_7
8	ABg5_8	ABm5	Kgm5_8
9	ABg5_9	ABm5	Kgm5_9

Table 23 Filename of Aperiodic Auto Correlation of Kronecker sequences (balanced gold sequence as inner and balanced m-sequenced as outter) when the stage of LFSR (r) = 5

Kronecker sequence r=5			
Balanced m-sequences as inner			
Balanced gold sequences as outer			
No.	**Inner**	**Outer**	**Kronecker**
1	Abm5	ABg5_1	Kmg5_1
2	Abm5	ABg5_2	Kmg5_2
3	Abm5	ABg5_3	Kmg5_3
4	Abm5	ABg5_4	Kmg5_4
5	Abm5	ABg5_5	Kmg5_5
6	Abm5	ABg5_6	Kmg5_6
7	Abm5	ABg5_7	Kmg5_7
8	Abm5	ABg5_8	Kmg5_8
9	Abm5	ABg5_9	Kmg5_9

Table 24 Filename of Aperiodic Auto Correlation of Kronecker sequences (balanced m-sequence as inner and balanced gold sequenced as outter) when the stage of LFSR (r) = 5

The lenght of kronecker sequence

= The lenght of balanced gold sequences(N_1) x

The lenght of balanced m-sequences (N_2)

= (N_1) x (N_2) = 62 x 62 = 3,844

Appendix C Programme to generate balanced m-sequence

```
%File name = balanced_m_sequence.m
%Generate balanced bipolar m-sequences of length N=2^r - 1,
%For primitive polynomial

%Save the sequences into a file

clear;
pack;

r=input ('Value of r for primitive polynomial :  ');%input the stage to generate m-
sequence
N = 2 ^ r - 1;%lenght of m-sequence

disp ('Sequence length=');
disp (N);%display value of lenght for m-sequence

%Generator primitive polynomial
C =[];%define array to keep coefficient or primitive polynomial

disp ('Input coefficient of primitive polynomial, C(1) TO C(r)')

for I = 1:r %input the coefficient until r stages
  cr=input ('Coefficient C(r)=  ');%keep value in each coefficient
  C = [C cr];%C = matrix of coefficients C(r)
end

disp (C);%display the value of coefficient

%Initial loading of LFSR
disp ('Initial loading of LFSR')
A=[];%define array for initial loading to LFSR
for I = 1:r%input the initial value until r stages
  i1=input ('Initial=  ');
  A = [A i1];%A = matrix of initial state of LFSR
end
disp (A);%display the initial loading value to LFSR

%M-SEQ GENERATION

u =[];  % array for unipolar sequences
for J = 1:N % the process will finish until N loop or equal lenght of sequence
  Fb = 0;
  out=A(r);    %Output chip
  u=[u out];   %Store the output chip in unipolar sequence

  %Calculate the Linear Feedback Shift Register
  for I = 1:r-1 % Calculate the value for the first shift on the left in the sequence
```

```
    Fb=Fb+A(I)*C(r-I);
  end
  Fb=Fb+out;%combine value of each chip in sequence
  Fb=mod(Fb,2);    %Take modulo 2 of Fb to have feedback value

  %Right Shift the content of LFSR 1 position = Right shift matrix A 1 position

  for I = r:-1:2 % changing position of array from r to 2 loop by 1 step
    A(I) = A(I-1);
  end
  A(1)=Fb; %define first array is equal the output after taking modulo 2
end

disp('Unipolar m-sequence for primitive polynomial');
disp(u);%display the unipolar m-sequence

%Change the unipolar to bipolar

b=[];%array for bipolar of m-sequence
for J = 1:N % change the value until the lenght of m-sequence
   chip(J) = u(J);
   % chip = 0 replaced by 1
   % chip = 1 replaced by -1
   if chip(J) == 0
      b=[b 1];
   else
      b=[b -1];
   end

end
disp ('Display array bipolar m-sequence');
disp (b); % display bipolar m-sequence

% change bipolar m-sequence to balanced m-sequence

bal=[];
for J = 1:N % replace value until all of chip in the m-sequence

      chip(J) = b(J);
      %if chip = 1 replaced by [1,-1]
      %if chip = -1 replaced by [-1,1]
      if chip(J) == 1
         bal=[bal 1 -1];
      else
         bal=[bal -1 1];
      end

end
disp ('Display array balance m-sequence');
```

```
disp (bal); % display balanced m-sequence

% save balance m-sequence into file

fnw=input ('Input the file name for copying the balance m-sequence :   ', 's');

fid=fopen(fnw,'wb');

fwrite (fid,bal,'integer*2');


% read file of balance m-sequence to check value

fnr1=input ('Input the file name for reading the balance m-sequence: ', 's');

fid1=fopen(fnr1);

balance_m=fread (fid1,[1,inf],'integer*2');

disp (balance_m);%display balanced m-sequence that is saved into file

st=fclose('all');
```

Appendix D Programme to generate balanced gold sequence

```
%File name = balanced_m_sequence.m
%Generate balanced bipolar m-sequences of length N=2^r - 1,
%For primitive polynomial

%Save the sequences into a file

clear;
pack;

r=input ('Value of r for primitive polynomial :  ');%input the stage to generate m-
sequence
N = 2 ^ r - 1;%lenght of m-sequence

disp ('Sequence length=');
disp (N);%display value of lenght for m-sequence

%Generator primitive polynomial
C =[];%define array to keep coefficient or primitive polynomial

disp ('Input coefficient of primitive polynomial, C(1) TO C(r)')

for I = 1:r %input the coefficient until r stages
  cr=input ('Coefficient C(r)=  ');%keep value in each coefficient
  C = [C cr];%C = matrix of coefficients C(r)
end

disp (C);%display the value of coefficient

%Initial loading of LFSR
disp ('Initial loading of LFSR')
A=[];%define array for initial loading to LFSR
for I = 1:r%input the initial value until r stages
  i1=input ('Initial=  ');
  A = [A i1];%A = matrix of initial state of LFSR
end
disp (A);%display the initial loading value to LFSR

%M-SEQ GENERATION

u =[];  % array for unipolar sequences
for J = 1:N % the process will finish until N loop or equal lenght of sequence
  Fb = 0;
  out=A(r);    %Output chip
  u=[u out];   %Store the output chip in unipolar sequence

  %Calculate the Linear Feedback Shift Register
```

```
  for I = 1:r-1 % Calculate the value for the first shift on the left in the sequence
    Fb=Fb+A(I)*C(r-I);
  end
  Fb=Fb+out;%combine value of each chip in sequence
  Fb=mod(Fb,2);    %Take modulo 2 of Fb to have feedback value

  %Right Shift the content of LFSR 1 position = Right shift matrix A 1 position

  for I = r:-1:2 % changing position of array from r to 2 loop by 1 step
    A(I) = A(I-1);
  end
  A(1)=Fb;  %define first array is equal the output after taking modulo 2
end

disp('Unipolar m-sequence for primitive polynomial');
disp(u);%display the unipolar m-sequence

%Change the unipolar to bipolar

b=[];%array for bipolar of m-sequence
for J = 1:N % change the value until the lenght of m-sequence
   chip(J) = u(J);
   % chip = 0 replaced by 1
   % chip = 1 replaced by -1
   if  chip(J) == 0
      b=[b 1];
   else
      b=[b -1];
   end

end
disp ('Display array bipolar m-sequence');
disp (b); % display bipolar m-sequence

% change bipolar m-sequence to balanced m-sequence

bal=[];
for J = 1:N % replace value until all of chip in the m-sequence

     chip(J) = b(J);
     %if chip = 1 replaced by [1,-1]
     %if chip = -1 replaced by [-1,1]
     if  chip(J) == 1
        bal=[bal 1 -1];
     else
        bal=[bal -1 1];
     end

end
```

```
disp ('Display array balance m-sequence');
disp (bal); % display balanced m-sequence

% save balance m-sequence into file

fnw=input ('Input the file name for copying the balance m-sequence :   ', 's');

fid=fopen(fnw,'wb');

fwrite (fid,bal,'integer*2');


% read file of balance m-sequence to check value

fnr1=input ('Input the file name for reading the balance m-sequence: ', 's');

fid1=fopen(fnr1);

balance_m=fread (fid1,[1,inf],'integer*2');

disp (balance_m);%display balanced m-sequence that is saved into file

st=fclose('all');
```

Appendix E Programme to calculate aperiodic of balanced gold sequence and balanced m-sequence

```
%File name = balanced_m_sequence.m
%Generate balanced bipolar m-sequences of length N=2^r - 1,
%For primitive polynomial

%Save the sequences into a file

clear;
pack;

r=input ('Value of r for primitive polynomial :  ');%input the stage to generate m-
sequence
N = 2 ^ r - 1;%lenght of m-sequence

disp ('Sequence length=');
disp (N);%display value of lenght for m-sequence

%Generator primitive polynomial
C =[];%define array to keep coefficient or primitive polynomial

disp ('Input coefficient of primitive polynomial, C(1) TO C(r)')

for I = 1:r %input the coefficient until r stages
  cr=input ('Coefficient C(r)=  ');%keep value in each coefficient
  C = [C cr];%C = matrix of coefficients C(r)
end

disp (C);%display the value of coefficient

%Initial loading of LFSR
disp ('Initial loading of LFSR')
A=[];%define array for initial loading to LFSR
for I = 1:r%input the initial value until r stages
  i1=input ('Initial=  ');
  A = [A i1];%A = matrix of initial state of LFSR
end
disp (A);%display the initial loading value to LFSR

%M-SEQ GENERATION

u =[];  % array for unipolar sequences
for J = 1:N % the process will finish until N loop or equal lenght of sequence
  Fb = 0;
  out=A(r);   %Output chip
  u=[u out];  %Store the output chip in unipolar sequence
```

```
  %Calculate the Linear Feedback Shift Register
  for I = 1:r-1 % Calculate the value for the first shift on the left in the sequence
    Fb=Fb+A(I)*C(r-I);
  end
  Fb=Fb+out;%combine value of each chip in sequence
  Fb=mod(Fb,2);    %Take modulo 2 of Fb to have feedback value

  %Right Shift the content of LFSR 1 position = Right shift matrix A 1 position

  for I = r:-1:2 % changing position of array from r to 2 loop by 1 step
    A(I) = A(I-1);
  end
  A(1)=Fb;  %define first array is equal the output after taking modulo 2
end

disp('Unipolar m-sequence for primitive polynomial');
disp(u);%display the unipolar m-sequence

%Change the unipolar to bipolar

b=[];%array for bipolar of m-sequence
for J = 1:N % change the value until the lenght of m-sequence
   chip(J) = u(J);
   % chip = 0 replaced by 1
   % chip = 1 replaced by -1
   if  chip(J) == 0
      b=[b 1];
   else
      b=[b -1];
   end

end
disp ('Display array bipolar m-sequence');
disp (b); % display bipolar m-sequence

% change bipolar m-sequence to balanced m-sequence

bal=[];
for J = 1:N % replace value until all of chip in the m-sequence

      chip(J) = b(J);
      %if chip = 1 replaced by [1,-1]
      %if chip = -1 replaced by [-1,1]
      if  chip(J) == 1
         bal=[bal 1 -1];
      else
         bal=[bal -1 1];
      end
```

```
end
disp ('Display array balance m-sequence');
disp (bal); % display balanced m-sequence

% save balance m-sequence into file

fnw=input ('Input the file name for copying the balance m-sequence :   ', 's');

fid=fopen(fnw,'wb');

fwrite (fid,bal,'integer*2');


% read file of balance m-sequence to check value

fnr1=input ('Input the file name for reading the balance m-sequence: ', 's');

fid1=fopen(fnr1);

balance_m=fread (fid1,[1,inf],'integer*2');

disp (balance_m);%display balanced m-sequence that is saved into file

st=fclose('all');
```

Appendix F Programme to calculate aperiodic of kronecker Sequences

```
% Generate Aperiodic Auto-Correlation for Kronecker sequence

% get length of outter sequence  N2
fnr1=input ('File name of Aperiodic Auto-correlation for OUTER balance bipolar m-sequence :  ', 's');
fid1=fopen(fnr1);
outer=fread (fid1,[1,inf],'integer*2');
disp (outer);%diplay aperiodic of balanced sequence that is chose for outer
N2= size (outer,2); %determine number of chips of balance m-sequences
disp('Length of outter sequence (d) N2');
disp (N2);% display lenght of sequence for outer sequence


% get length of inner sequence  N1
fnr2=input ('File name of Aperiodic Auto-correlation for INNER balance bipolar m-sequence :  ', 's');
fid2=fopen(fnr2);
inner=fread (fid2,[1,inf],'integer*2');
disp (inner);
N1= size (inner,2); %determine number of chips of balance m-sequences
disp ('Length of inner sequence (c) N1 =');
disp(N1);%display lenght of sequence for inner sequence

% calculate length of N( time shift)Aperiodic auto correlation of Kronecker seq

N = N1 * N2;% fomular to calculate lenght of Kronecker sequence

disp( ' Length of all time shift for Kronecker sequence =');

disp (N);%dispaly lenght of Kronecker sequence


% Calculate Aperiodic Auto-correlation for Kronecker seq
% Example : t=11, N1=4 ;
%         Take T1 = mod (t,N1)+1 = mod (11,4)+1 = 3 + 1 = 4
%         Take Tj = floor(t/N1)+1 = floor(11/4)+1 = 2 + 1 = 3


ape_kro=[]; % keep Aperiodic Auto of Kronecker seq

for t = 0: N-1 %Calculate aperiodic auto correlation at each time chip

  Tl = mod (t, N1)+1;% modulo N1
  x1=inner(Tl);% Receive the array for inner
  Tj = floor (t/N1)+1;% result for round toward minus infinity
  x2=outer(Tj);%Receive the array for outer
```

```
  Cd = Tj + 1; %Define value for camparing each chip
  %Compare the value
  if Cd > N2 %if Cd is longer than the lenght of outer sequence
    x3=0;
  else
     x3=outer(Cd);
  end

  Cc= N1-Tl+2;%Find the output for comparing with the lenght of inner sequence

  if Cc>N1
    x4=0;
  else
    x4=inner(Cc);
  end
   Ca = x1*x2 + x3*x4;% formular to calculate Aperiodic Auto of Kronecker seq
   ape_kro = [ ape_kro Ca]; % Keep the value for array of aperiodic kronecker
sequence

end

disp('Aperiodic Auto-Correlation of kronecker sequence');
disp(ape_kro);%display value for aperiodic kronecker sequence


% save the result into file

fnw=input ('Input the file name for copying the Aperiodic Autocorrelation of
Kronecker seq :  ', 's');

fid=fopen(fnw,'wb');

fwrite (fid,ape_kro,'integer*2');

st=fclose('all');


% read file to check the value after saving file in the storage

fnr1=input ('Input the file name for reading the Aperiodic Autocorrelation of
Kronecker seq:  ', 's');

fid1=fopen(fnr1);

aperiodic=fread (fid1,[1,inf],'integer*2');

disp (aperiodic);
```

Appendix G Programme to calculate aperiodic cross correlation of kronecker sequences

```
%File name = cross_sqaure.m
%Sum of squares of aperiodic cross-correlation of one pair of sequence
%sum squares Cak,ai at the time shift mk = Cak (mk) * Cai (mk)

Cak = [];%Array for the aperiodic auto-correlation of kronecker sequences ak at the time shift mk
Cai = [];%Array for the aperiodic auto-correlation of kronecker sequences ai at the time shift mk

%Read file file from the storage for Cak
fnr1=input ('Input the file name for Cak :  ', 's');
fid1=fopen(fnr1);
Cak=fread (fid1,[1,inf],'integer*2');
%disp (Cak);

%Read file file from the storage for Cai
fnr2=input ('Input the file name for Cai:  ', 's');
fid2=fopen(fnr2);
Cai=fread (fid2,[1,inf],'integer*2');
%disp (Cai);

%Sq=[];
N = size (Cai,2);%Define size for Cai
b = 0 ; % set b = 0 in order to used in summation
a= Cak(1) * Cai(1)
%Calculate square of aperiodic cross-correlation
for x = 2 : N % summation loop
   c = Cak(x)*Cai(x);
   b = b + c;
end
sum= b*2 % result
%disp (sum);
d=sum+a % combination value after multiply each chip
fprintf('\n');
disp('Square of Cross-Correlation : ');
disp(d);
%Sq= [Sq d];
%disp(Sq);
```

Appendix H Programme to calculate BER and show graph for output in BER values

```
%File Name = ber_m_g_seq.m
%= = = = = = = = = = Calculating BER= = = = = = = = = = = = = = = = = %
%Comparing BER of balanced m-sequence and balanced gold sequence
%There are two line in the graph that show output of BER
%balanced gold sequence as inner and balanced m-sequence as outer
%balanced m-sequence as inner and balanced gold sequence as outer

%This value is shown for stage of LFSR (r) = 3

%The output for mean power of  Multiple Access Interference of kronecker
sequence ;
%Balanced gold sequence as inner
%Balanced m-sequence as outer
N1=196;  %sequence lenght = lenght of balanced m-sequence*lenght of
balanced gold-sequence
Sum_PMAI1 = 948.425% Value of sigma square
BER1=[]; % define Bit Error Rate is array

%The output for mean power of  Multiple Access Interference of kronecker
sequence ;
%Balanced m-sequence as inner
%Balanced gold sequence as outer
N2=196;  %sequence lenght = lenght of balanced m-sequence*lenght of
balanced gold-sequence
Sum_PMAI2 = 987.608;% Value of sigma square
BER2=[]; % define Bit Error Rate is array


for K = 2 : 20  %value of simultanous users : 2 - 20 users
  % Calculating BER : balanced gold sequence as inner and balanced m-
sequence as outer
  Value1 =(K-1)*Sum_PMAI1;
  delta1 = N1/sqrt(Value1);
  disp ('K=')
  disp (K)
  B1 = 0.5*erfc(delta1/sqrt(2))    %Equation for calculating BER
  BER1 = [BER1 B1];

  % Calculating BER : balanced m-sequence as inner and balanced gold
sequence as outer
  Value2 =(K-1)*Sum_PMAI2;
  delta2 = N2/sqrt(Value2);
  disp ('K=')
  disp (K)
  B2 = 0.5*erfc(delta2/sqrt(2))    %Equation for calculating BER
```

```
    BER2 = [BER2 B2];

  end


  disp('The values of BER : balanced gold sequence is inner,balanced m-
sequence is outer');
  %display the value of BER from
  %balanced gold sequence is inner sequence, balanced m-sequence is outer
sequence
  disp(BER1);

  disp('The values of BER : balanced m-sequence is inner,balanced gold
sequence is outer');
  %display the value of BER from
  %balanced m-sequence is inner sequence, balanced gold sequence is outer
sequence
  disp(BER2);

  K=2:1:20; % define K=2 until 20 : increase by 1 for value

  %Graph for BER balanced gold sequence is inner sequence, balanced m-
sequence is outer sequence
  %This graph is red line
  semilogy(K,BER1,'r-') % Plot with y-axis scaled logarithmically (basic)
  hold on;

  %Graph for BER balanced gold sequence is inner sequence, balanced m-
sequence is outer sequence
  %This graph is blue line
  semilogy(K,BER2,'b-') % Plot with y-axis scaled logarithmically (basic)

  axis([2 20 0 1]); % axis([xmin xmax ymin ymax])

  xlabel('Number of simultaneous'),ylabel('BER'); %  define label for x-axis and
y-axis

  title('Performance of Kronecker codes'); % define title for graph
```