DATA PROCESSING DESIGN OF WIDEBAND CODE DIVISION MULTIPLE ACCESS (WCDMA)

Department of technology, University of Kalmar

Department of technology, Blekinge institute of technology

Cybertel Software Limited

BY
THUMATI VENKATA RAO
RATAKONDA RAMYA CHANDRA

MEE07:36

CONTENTS

1. Abstract 4

2. Introduction 5

2.1Communication satellites 5

2.2 Introduction to Mobile cellular telephony 5

2.3 Components of a cellular system 6

2.4 Evolution of cellular phones 6

3. System Analysis 8

3.1 Existing system 8

3.2 Proposed system 8

3.2.1 Multi user Detection 8

3.3 Objectives 9

3.4 Multiple-access techniques 9

3.4.1 Frequency division multiple access 10

3.4.2 Time division multiple access 11

3.4.3 Code division multiple access 12

3.4.4 Wideband Code division multiple access 14

3.5 The Main Difference between CDMA and WCDMA 15

3.6 Spread spectrum multiple access 15

3.7 Spread spectrum techniques 16

3.7.1 Frequency-hopping spread spectrum 16

3.7.2 Time-hopped spread spectrum 16

3.7.3 Direct sequence spread spectrum 16

4. Data Processing Design of WCDMA 18

4.1 Pseudo-noise sequence generator 20

4.1.1 Linear feedback shift register 20

4.2 Gold code generator 22

4.3 Spreader 25

4.4 RCPC codes 26

4.4.1 Convolution and Puncturing Process in RCPC 27

4.4.2 Search criterion for good RCPC codes 29

4.5 Modulator 30

4.6 Demodulator 31

4.7 RCPC decoding 32

5. Matched Filter Operation 34

5.1 Matched Filter 34

5.2 Receiver Pseudo-Noise Sequence Generator 35

5.2.1 Receiver Liner Feedback Shift Registers 35

5.3 De-Gold code Generator 37

5.4 De-Spreader 37

5.5 Final Data received 38

6. Conclusions 39

7. Appendix 40

7.1 C Program 40

7.2 MAT LAB Program 53

8. Bibliography 64

1. ABSTRACT

Wideband Direct-Sequence Code Division Multiple Access (W-CDMA) is the emerging protocol for the next generation (3G) wireless communication systems. W-CDMA has been designed to add features such as multimedia capabilities, high data rates and multi-rate services to the existing wireless communication framework. Several standards for third generation systems have been proposed and developed by different industrial committees in countries such as the U.S, Europe and Japan. All these standards have accepted CDMA in one form or another as the multiple access method for wireless communications requirements.

In this work, we study the implementation issues involved for one of the proposed Rate Compatible Punctured Convolution Coding (RCPC) to increase information rate in WCDMA, the detection algorithms for base-stations. It was found that these proposed algorithms for detection have different processing and precision requirements. In the rate compatible punctured convolution from each coded symbol some of the bits are punctured in order to achieve the higher information rate. At the same time decoding becomes quite easy even if some of the bits from the coded symbol are not available. We have written these algorithms for a single user at ideal conditions based on theory but not practically.

2. INTRODUCTION

2.1 Communication Satellites

The first communication satellites revolution is started in early 1945s by an article based on Communication satellites, " Extra-Terrestrial Relay" published by Arthur C. Clark in which he discussed astonishing secrets of covering whole Blue Planet by three satellites. Thus, how we started new era of communication and now reached to such extent that we communicate to every part world sitting at home, updating latest information, getting control over surroundings, not only this, it also made us to research still deeper into subject by classifying into various communication fields like mobile communications, networking communications, radar communication etc.

Basically a communication satellite has several transponders which receive some portion of spectrum and transmit it with another frequency and amplitude, to avoid interference with the incoming signal. The downward beams can be broad; covering a substantial fraction of the earth's surface, or narrow, covering an area only hundreds of kilometers in diameter [1].

In general the communication satellites have up to a dozen Transponders. Each transponder has a beam that covers some portion of the earth below it, ranging from a wide beam 10,000 km across to a Spot beam only to 250 km across [1]. Stations within the beam area send frames to the satellite on the transmission (uplink) frequency. The satellite then resends them on the downlink frequency. To keep the transponder from going into oscillation we are using different frequencies. Three classes of protocols are used on the multiple access (transmission) channels: FDMA, TDMA and CDMA. While resending there will be no channel allocation problem but the main problem is with the transmission channel, since the downlink channel has only a single sender (the satellite).

2.2 Introduction to Mobile Cellular Telephony

Cellular mobile radio telephony is an important innovation in the field of communication services. The main aim of cellular systems is to provide personal telephone services to the moving subscribers. Cellular systems have evolved from military radio design, but their civilian applications have become possible only due to the recent technological developments and requirements. In mobile telephony, it is necessary to change the operating frequency as one move from one transmitter's area to another. Traditionally military mobile radio, change of operating frequency is tedious involving manual retuning of circuits and readjustment of antennas .In modern cellular systems, operating frequencies can be changed many times a second under the control of microprocessors and pseudo-random number generators [2].

2.3 Components of a cellular system

- Mobile/Handset:

The main function of handset is voice transponder (mike/speaker) and radio "transceiver". It is portable and low power consumption device. The main issues involved in mobile or handset is quality of voice and battery life.

- Base Station:

The main function of a base station is cell management and radio transceiver. It is capital equipment with high cost. The main issues involved in this are multiple-access and maintaining links.

- Mobile Switching Center:

The main function of a mobile switching center is call management for a group of cells. It is also capital equipment with high cost [2].

2.4 Evolution of cellular phones

1G, 2G, 2.5G, 3G are the various generations in the history of cellular phones. Every generation developed with different features. Now-a-days the 3G network has very demand due to its additional features which is not available in 1G, 2G, and 2.5G.These systems are user-friendly.

1G

The first-generation were analog systems and circuit-switched. In This generation the voice links are poor, low capacity and security is also less. These systems are not under use now but in some areas 1G spectrum is being overtaken by 2G and 3G [3].

2G

Now 2G networks are in use around the world. Second-generation systems use digital encoding and include GSM (Global system for mobile communications), D-AMPS (Time

Division Multiple Access) and CDMA (Code Division Multiple Access). 2G network system supports voice with high bit rate and limited data communications. They offer additional services like data, fax and SMS. There are different levels of encryption by 2G protocols [3].

2.5G

2.5G systems are nothing but extension of 2G systems which can be provide additional features such as packet-switched connection (GPRS) and enhanced data rates (HSCSD, EDGE) [3].

3G

Third-generation systems support high data rates, which are measured in Mbits/second. 3G networks are introduced in Japan in 2001, in Europe and part of Asia/Pacific in 2002. 3G mainly supports the applications which are related to bandwidth-hungry applications such as full-motion video, video-conferencing and full Internet access. The 3rd generation wireless communication objective is to provide high speed wireless communication to support multi media, data and video in addition to voice [3].

3. SYSTEM ANALYSES

3.1 Existing System

One of the most important aspects that effects wireless communication system is the amount of interference present in communication channels, since interference is intrinsic in different wireless systems, as they are operated in multiple access systems. Generally, multiple access system contains multiple transmitter/receiver pairs communicate through the same communication channel using non-orthogonal multiplexing. This fabricated interference arises due to Non-systemic features of wireless systems, like sharing bandwidth with other dissimilar communication services.

CDMA Receivers treats this Multiple Access Interference (MAI) as noise. As the number of users increases, no of interferers also increased, which further increases, which leads increase in the power of communication channel, in turn MAI increases abruptly, leading to deterioration of the performance [5].

3.2 Proposed System

In order to overcome MAI, different modules are developed trying to eliminate MAI in different ways. The narrowband CDMA IS-95 standard, this system generally supports only voice and low-data rates up to 9.6 Kbps and implemented based on the single-user algorithms implemented by single user matched filter (SUMF) at the base-station receiver that ignore Multiple Access Interference (MAI) between different users. To improve performance at higher data rates systems requires the implementation of advanced multi-user algorithms for channel estimation and Multi User detection. These algorithms should overcome MAI by processing the signals of all users at the base-station receiver [5].

3.2.1 Multi User Detection

Multi-User Detection in CDMA is one of the significant concept which had been studied for several years. Now let us take brief introduction of some Multi-user algorithms, the first ever used algorithm is the concept of conventional filter or matched filter. The basic concept of this filter is, signal received from a particular user is correlated with a signature signal of that user, thus the signal is demodulated, and this process is done for every user using the same channel.

Different Advanced methodologies are proposed to overcome MAI, like Optimal Multi-user detection (Joint Optimal, Individual Optimal), Sphere decoding, variational free energy minimization, Minimum Mean Square Error (MMSE). This can also support variable data rates because CDMA is interference limited, the proposed system employs Optimal and MMSE Multi-user Detection to eliminate Multiple – Access Interference (MAI). Optimal Multi-user detection is employed when we have less number of users. Gradient projection algorithm is very much efficient than the optimal multi-user detection when the number of users is large.

3.3 Objectives

- To manifest orthogonal frequency division multiplexing which is other face of orthogonal CDMA that imparts quick signal processing at the receiver end, and has very efficient wireless link for high-speed applications.
- To design CDMA system, also to simulation of CDMA real time environment.
- To generate spreading and scrambling codes.
- Data Traffic Analysis
- To eliminate Multiple -Access Interference (MAI) by Multi User Detection

"The manifestation is to prove that spread-spectrum users can share a same frequency band without interfering carrier signal of one user with other users, thus, we can increase the efficiency within communication band used". In order to eliminate MAI in received bit vector different Multi-user Detection are implemented like optimal filters, Sphere decoding etc., MAI. Even Gradient projection algorithm was designed and deployed to resolve the problems with Multi-user Detection [6].

3.4 Multiple-Access Techniques

Multiple Access is basic in the field of satellite communication, since it is the best way by which one can cover entire geography and analyze nature of the satellite channel. The Multiple Access Technique has capability to interconnect large number of earth stations to connect their respective voice, data, messaging and TV links through a satellite. Multiple access means the users operating at same time, multiple can be supported. In other words a large number of users can share a common pool of radio channels and any user can gain access to any channel (each user always not assigned same channel).

In January 1998, WCDMA technology was decided to be as multiple access techniques for the third-generation mobile telephone system by European Telecommunication Standard Institute (ETSI).

Based on modern, layered network-protocol structure, similar to the protocol structure used in GSM networks, WCDMA became a services platform to 3G. This is particularly designed for high-speed data services and more particularly for internet-based packet-data, which is offering up to 2Mbps in stationary or office environments, and in wide area or mobile environments up to 384Kbps. The third generation radio standard system uses CDMA instead of TDMA to increase the data transmission rate in GSM system and also that 3G system will offer voice, data, motion-video and other multimedia capabilities. In the ITU's 3G specification WCDMA has became the Direct Sequence (DS) mode, which includes the 1X Multi-Carrier mode (1X MC) and 3X Multi-Carrier mode (3X MC). 1X MC (known as cdma2000) and 3X MC consist of 3G upgrade paths for carriers already using CDMA (cdmaOne) [7].

The geographic region is divided into cells by cellular systems. In this region the base station is communicated with the help of a mobile unit in each cell. The ultimate destiny in designing of the cellular systems is to be able to handle as many calls as possible in a given bandwidth with some reliability. There are many different ways to allow access to the channel.
They can be described as follows:

- Frequency division multiple-access (FDMA)
- Time division multiple-access (TDMA)
- Code division multiple-access (CDMA)
    1. Frequency-hop CDMA
    2. Direct-sequence CDMA
    3. Multi-carrier CDMA (FH or DS)
- Wideband code division multiple-access

3.4.1 Frequency Division Multiple Access (FDMA)

For cellular systems FDMA was the first step of the multiple-access technique. In this technique a user is assigned a pair of frequencies when placing or receiving a call. One frequency is used for downlink (base station to mobile) and one pair for uplink (mobile to base). This is called frequency division duplexing. That frequency pair is not used in the same cell or adjacent cells during the call. Even though the user may not be talking, the spectrum cannot be reassigned as long as a call is in place. In FDMA, each transmitter is assigned a distinct frequency channel that receivers can discriminate among them by tuning to the desired channel [8].

TDMA and CDMA are always used in combination with FDMA, i.e., a given frequency channel may be used for either TDMA or CDMA independently of signals on other frequency channels. (Ultra wide band is arguably an exception, as it uses essentially all of the usable radio spectrum in one location).

Advantages

- This technique has the advantage of simplicity and relies on the use of proven equipment.

Disadvantages

- Lack of flexibility in case of reconfiguration; to accommodate capacity variations it is necessary to change the frequency plan and this implies modification of transmitting frequencies, receiving frequencies and filter bandwidths of the earth stations.
- Due to the generation of intermediate products, a Capacity loss is detected when the number of accesses increases and the need to operate at a reduced satellite animating power (back-off).
- There is a need to control transmitting power of earth stations in such a way that, at the satellite input the carrier powers are the same which in turn helps in avoiding capture effect. In real time this control should be held and the attenuation caused by rain on the uplinks must be adapted [9].

3.4.2 Time Division Multiple Access (TDMA)

The Technology that evolved for shared medium usually for radio networks is Time Division Multiple Access (TDMA). In this the same frequency can be shared by several users by dividing into different time slots. The transmission of the users succession is rapid, one by one each assigned their own time slot. The same transmission medium can be used by multiple users (for example radio frequency) and at the same time only a part of the band width is used as per the requirement. In satellite systems, Local Area Networks (LAN´s), physical security systems and combat net radio systems TDMA is widely used.

In America TDMA is commonly refer to 2G-second generation mobile phone standard, mostly referred as DAMPS, that uses TDMA technique to timeshare the bandwidth of the carrier wave. TDMA can be described as a time division multiplexing with the specific point instead of having single transmitter connected to single receiver, there exist multiple transmitters.

Advantages

- The main advantage of TDMA is that the radio part of the mobile has its own timeslot to listen and broad caste. The mobile carries out the network measurements, surrounding transmitter's detection on different frequencies for the rest of the time.
- Only on a single carries the channel amplification is done at each instant, the band width is occupied by all the channels. There exist no intermediation products and the benefits of the carrier that is saturation of power in the channel. But existence of nonlinearity combined with the filtering effects on transmission and receiving, introduces degradation with respect to the ideal digital transmission.
- In the stations the control of transmitting power is not required.
- The transmission and receiving at all the stations on the same frequency with different or same origins and destinations of the burst simplifies tuning.

Disadvantages

- The disadvantage of TDMA systems is the creation of frequency interference that is connected directly to the length of the time slot. This is the irritating noise that is heard sometimes when a GSM phone is left next to a radio.
- Quality of the voice is poor [10].

3.4.3 Code Division Multiple Access (CDMA)

The use of spread spectrum in any form by multiple transmitters to send data to the same receiver at the same time on the same frequency channel with out harmful interference is called as CDMA (Code Division Multiple access). Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA) are the other multiple access techniques which are widely being used at present. In these three techniques among various signals receivers can be discriminated by using different codes, frequency channels and time slots respectively. At present CDMA is also used to refer as a family of specific implementations of CDMA pioneered by Qualcomm for use in digital cellular telephony that include IS-95 (cdma One) and IS-2000 (cdma2000). The two different uses of this term are:

Few frequently confusing facts of CDMA and WCDMA:

- CDMA (the multiplexing technique) is used as the principle of the WCDMA radio link.
- The WCDMA radio links is used in the global 3G standard, UMTS, and Japanese 3G standards and Vodafone.
- The CDMA family of standards (including cdmaOne and cdma2000) do not exist together with the standard family of WCDMA.

The receivers can discriminate the unwanted signals partially in all forms of CDMA by using spread spectrum process gain. Signals with different spreading codes or same spreading code but different timing offset appear with less wideband noise with the help of process gain, while signals with desired spreading code and timing are received.

In a random manner the TDMA or FDMA receivers can reject strong signals completely on other frequency channels or timeslots. But in the case of CDMA unwanted signals are rejected only partially. The desired signal can overcome, even if all the unwanted signals are much stronger than the required one. In a CDMA system the general requirement is at least near matching of different power levels of the signal at the receiver. A fast closed loop power control scheme is used by the base station to control each mobiles transmission power in CDMA cell. In CDMA scheme it is very important to minimize the ratio of signal interference and maximizing capacity of the channel. This is called as Forward error correction (FEC) coding [11].

Advantages:

- The main advantage of CDMA is availability of infinite CDMA codes over TDMA and FDMA.
- Ideally CDMA suites large number of transmitters each generating a comparatively small amount of traffic at irregular intervals, and avoids operating expense of continually allocating and de-allocating a limited number of orthogonal time slots or frequency channels to individual transmitters.

- CDMA transmitters simply work when data needs to be transmitted and rest of the time goes off air.

Disadvantages:

- Longer transmission delay due to low bit rate
- High interference due to high energy per bit
- Low total throughput

3.4.4 Wideband Code Division Multiple Access (WCDMA)

Initial developments of WCDMA (Wideband Code Division Multiple Access)/3G were done by NTT Docomo with the radio link of 3g network FOMA. WCDMA is a CDMA protocol. This supports very high-speed multimedia services (MMS) such as full-motion video, Internet access and video conferencing. WCDMA is developed based on radio access techniques by ETSI alpha group [4].

WCDMA Features:

- Full coverage and mobility for 144 kbps, preferably 384 kbps.
- Limited coverage and mobility for 2 Mbps.
- Compared with 2G it has High spectral efficiency.
- It has High flexibility to introduce new services.
- An early goal, now largely abandoned, was a global air-interface standard.
- Bandwidth of 5 MHz (or more, up to 20 MHz proposed).
- Chip rates from 3.84 million chips-per-second (up to 15 Mcps proposed).
- Support of multiple data rates & packet data.
- Coherent uplink (with uplink pilot channel).
- Base-station antenna beam forming.
- Advanced interference suppression techniques.

WCDMA Advantages:

- WCDMA supports high-speed data services.
- WCDMA is better suited for development in the very dense cities of Europe and Asia
- It supports two basic modes of duplex.

3.5 The Main Differences between CDMA and WCDMA

CDMA	WCDMA
1) It supports only data source 2) It is 2.5G technology 3) It Operates in MHz 4) Its Bit rate is in Kbps	1) It supports Multimedia applications 2) It is 3G technology 3) It Operates in GHz 4) Its Bit rate is in Mbps

Table1: Comparison between CDMA and WCDMA

3.6 Spread Spectrum Multiple Access

Basically spread spectrum technology was originally developed for military, anti-jam single user needs where the main aim was to transmission of data secretly, communication even in jamming. The main phenomenon behind the working of spread spectrum is spreading energy of narrow- band source signal (e.g., 10 kHz speech) over a wide bandwidth source signal (e.g., 1-10 MHz). The signals modulated through spread spectrum are broadband, noise like and resistant to multipath.

At present the major application of spread spectrum is multi user environments in 2G and 3G cellular communications. The same theoretical capacity and same duration of time exists for CDMA-based IS-95 and TDM-based GSM/IS-136 [in a given bandwidth (B Hz) and time duration (T sec) --- i.e., 2BT orthogonal carriers are possible].

Spread Spectrum is a (controlled) interference-limited system

- Carriers are chosen to be “random” waveforms with respect to each other
- A unique randomized code is assigned to each user/carrier, different and nearly orthogonal (i.e., low cross-correlation) to the other codes of unique time slot of TDMA and unique frequency of FDMA.
- To separate the users Correlation (CDMA) and frequency agile (Frequency Hopping Spread Spectrum ---FH/SS) receivers are used.
- Asynchronous transmission for the users can be done with respect to each other. In Code Division Multiple Access (CDMA) better performance can be achieved using especially Direct Sequence multiple access.
- Average, over a long time residual interference is added to correlation being rejected. (CDMA is said to be a noise-averaging system).

- The code is a pseudo-noise (PN) like, high bit-rate signal that is used to multiply the user information symbols.
- The capacity of a system is not subjected to a hard limit (like TDMA); increasing the number of users reduces the received signal-to-interference ratio and performance.

3.7 Spread Spectrum Techniques

The main aim of Code Division Multiple Access (CDMA) is each subscriber is assigned a unique code that can be used by the system to differentiate that user from all other users transmitting at the same instant over the same frequency band. There are several techniques that have been considered for mobile radio CDMA communications, including:

- Frequency-Hopping Spread Spectrum (FH/SS)
- Time-Hopped Spread Spectrum (TH/SS)
- Direct Sequence Spread Spectrum (DS/SS)

3.7.1 Frequency-Hopping Spread Spectrum

In a frequency-hopping system for duration of a specific time the signal frequency is constant, referred to as a time chip. The categorization of frequency-hopping can be described in two ways. They are fast-hop or slow-hop, as there will be a notable difference in their performance. Fast-hop system can be described as a system in which frequency-hopping takes place at a very high rate than the bit rate of the message. Similarly in a slow-hop system, slow frequency-hopping takes place at a lesser rate than the message bit rate [12].

3.7.2 Time-Hopped Spread Spectrum

The time of transmission is divided into intervals known as frames in a time-hopping system. Each frame is divided into M time slots. The modulation with the message will be done during each frame one and only one time slot. During the selected time slot, all the message bits collected in the previous frame are transmitted in a burst.

3.7.3 Direct Sequence Spread Spectrum

In a direct sequence system, a pseudo noise code digital stream multiplies the base band signal transmission. The direct sequence is an averaging type system where the reduction of interference takes place because the interference can be averaged over a large time interval. The avoidance systems are frequency hopping (FH) and time-hopping (TH) systems. Here, the occurrence of reduction in interference is due to the signal being made to avoid the interference for a large fraction of the time.

System	Advantages	Disadvantages
Direct Sequence	→Best noise and anti-jam performance. →Most difficult to detect (more private). →Best discrimination against multipath.	→Requires wideband channel. →Long acquisition time. →Near/far problem.
Frequency-Hopping	→Greatest amount of spreading. →Can be programmed to avoid portions of the spectrum. →Relatively short acquisition time. Less affected by near/far problem.	→Complex frequency synthesizer. →Not useful for range-rate measurement. →Error correction required.
Time-Hopping	→High bandwidth efficiency →Implementation simpler than frequency-hopping. →Useful when transmitter is average power limited, but not peak power limited. →Near/far problem avoided.	→Long acquisition time. →Error correction required

Table2: Comparison between CDMA spread spectrums

4. DATA PROCESSING DESIGN OF WCDMA

The Data Processing design in the transmitter and receiver will be discussed here. RCPC technique is being implemented for the coding and decoding purpose in the system. There are no channels for the transmission of the data so external noise models (e.g. AWGN...) are not being introduced into the system.

The coding has been implemented both in C and MATLAB. MATLAB was used for interactive plotting and GUI purpose. The obtained results from both C and MATLAB are shown in the examples.

BLOCK DIAGRAM OF WCDMA WITH RCPC

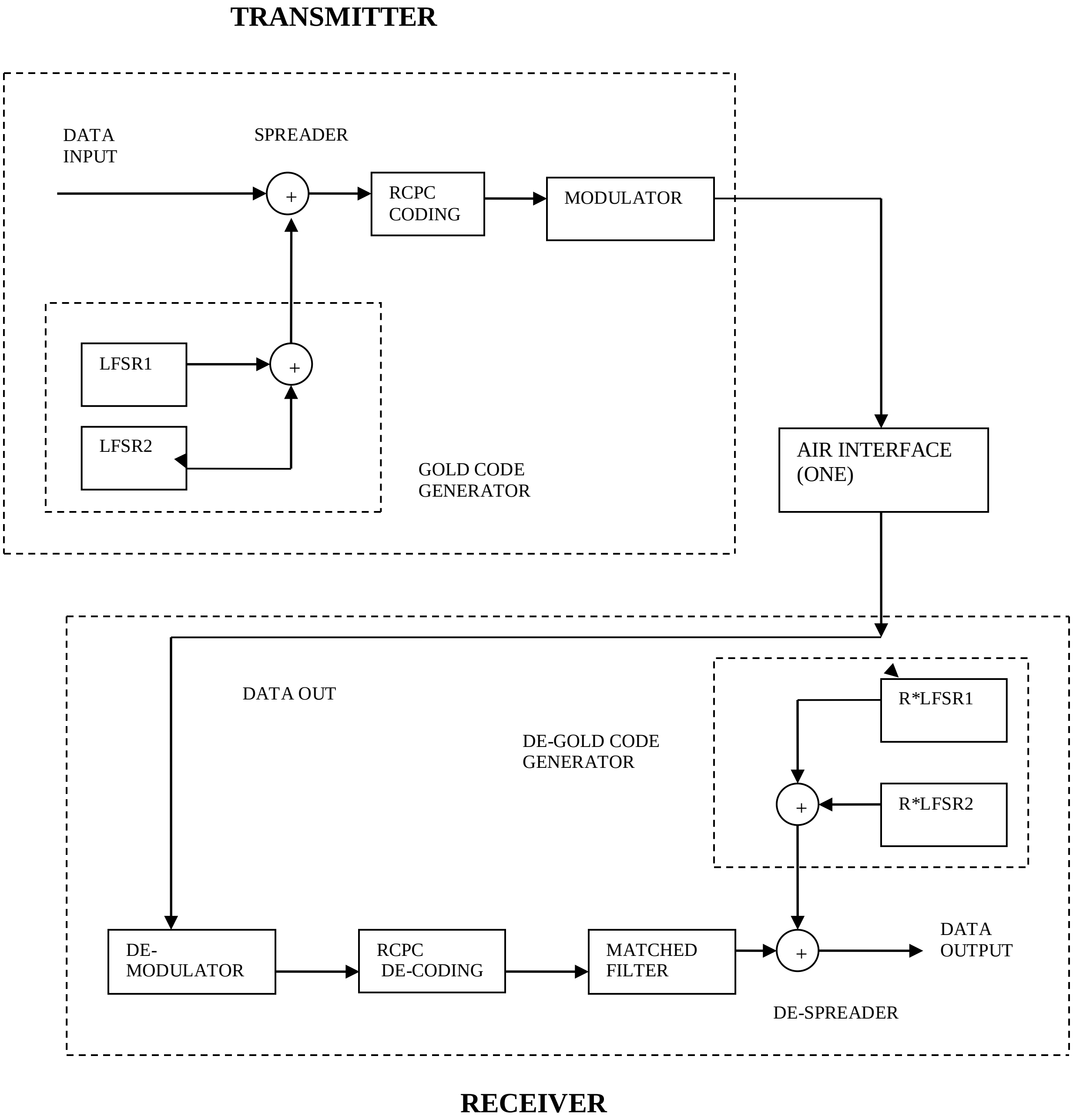


Figure1: Block diagram of WCDMA with RCPC

DESCRIPTION OF BLOCK DIAGRAM

4.1 Pseudo-Noise Sequence Generator

The PN Sequences can be generated by linear feedback shift registers.

4.1.1 Linear Feedback Shift Register

Linear feedback shift registers (LFSR) has (2^n-1) states, where 'n' is the number of registers in the LFSR. At each end of clock edge the content in the registers are shifted to right by one position. To give the feedback there is a predefined registers or taps in the register. They will give feedback to the left most register through XNOR or XOR gate. A value of all 1's is illegal in the case of XNOR feedback and a value of all 0's is illegal in the case of XOR feedback because in this state the counter would remain locked-up. Here we are creating pseudo noise using LFSRs to convert narrow band signal into wide band signal.

For example a 4-bit LFSR can sequence through (2^4-1) =15 states (the state 1111 is illegal state in XNOR feedback and the state 0000 is for XOR). On the other hand, a 4-bit binary up-counter would sequence through 2^4=16 states in which no illegal states. LFSR counters are very fast since they don't use carry signals [13].

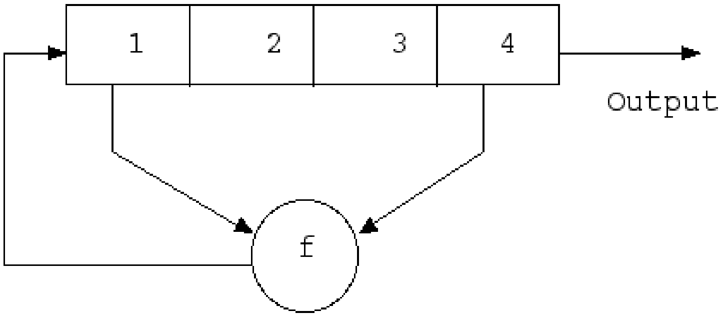


Figure2: 4-Bit LFSR, Tap Sequence; [4, 1]

One of the two main parts of an LFSR is the shift register and the other one is feedback function. The main function of the shift register is to shift its contents into adjacent positions with in the register or if it is the position on the end, out of the register. The position on the other end is left empty unless some new content is shifted into the register.

Uses of a shift register are

- Convert between parallel and serial data

- Delay a serial bit stream

An LFSR is also known as state machine. The bits used for the feedback function and the output of the feedback function together will decide the state of the LFSR. With each shift, the LFSR moves to a new state and if the contents of the register are all zeros then the LFSR will never change its state. For any given state there will be only one succeeding state or any given state can have only one preceding state.

The LFSR that we are using here is a five stage LFSR and hence the Maximal Sequence obtained is of 31-bits. We need two LFSR structures for the implementation of the Gold Code and the two structures chosen to be implemented are [1 3] and [1 2]. The figure 3&4 shows the coding result of PN sequence1 and PN sequence2.

Example: No. of bits in LFSR1 is 3 and sequence is 1 1 –1

Enter the no of taps 2

Tap locations are 1 and 3

Then PN sequence 1 is

By using 'c' and mat lab the result code is -1 1 1 -1 -1 -1 1

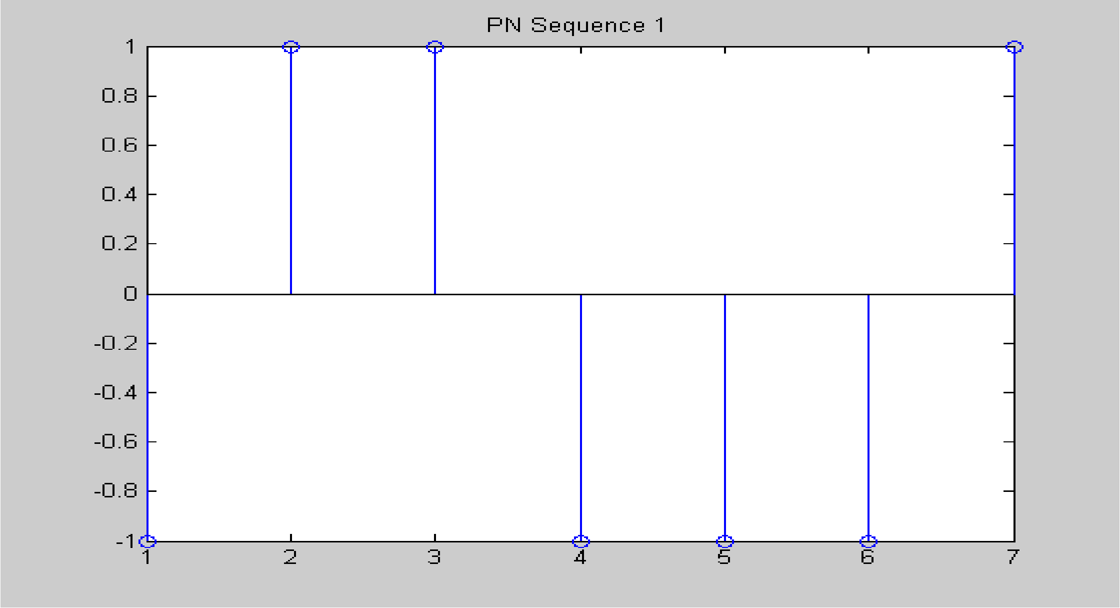


Figure3: Coding Result of PN sequence1

Example: No. of bits in LFSR 2 is 3 and sequence is 1 -1 1

Enter the no of taps 2

Tap locations are 1 and 2

Then PN sequence 2 is

By using 'c' and mat lab the result code is 1 -1 1 -1 -1 1 -1

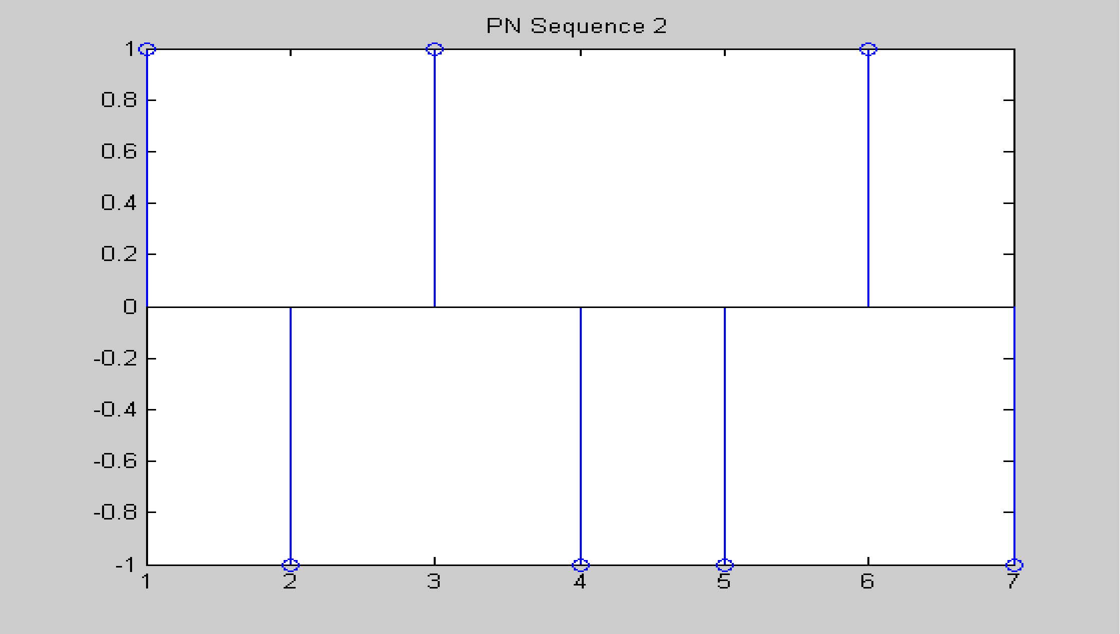


Figure4: Coding Result of PN sequence2

4.2 Gold Code Generator

The gold code generator is initially invented by "R. Gold" in 1967. According to him the set of small correlation PN codes can be created by modulo 2 additions of the results that are obtained from the two LFSRs. Then the result obtained is a set of codes with correlation properties and these codes are known as Preferred Pair Gold Codes. By multiplying the two PN sequences we will get Gold-Code as shown in the figure 5. The main advantage to use the gold code is that the cross correlation between the two codes is uniform and bounded [14].We can obtain preferred pair Gold-Code sequences by time-shifting the one of the PN sequence. These preferred pair Gold-Code sequences are checked for minimum cross correlation and the sequence with minimum cross correlation is used as the "key". In this project we are considering only the zero shift Gold-Code sequence, as it is generally used.

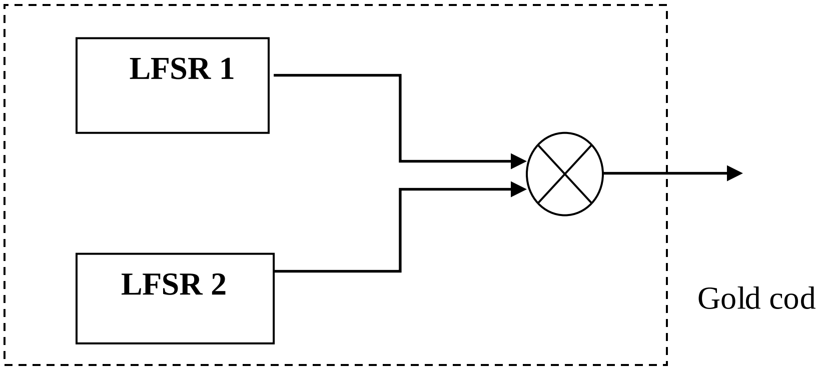


Figure5: Gold code generator

The logic involved in the simulation of Gold-Code generator is

- •The two PN Sequences are used as components in the program and are given common initial fill or “seed”.
- •We will declare two signals “pn1” and “pn2”, and we will obtain the outputs of the PN Sequence generators.
- •By multiplying these outputs we will get Gold-Code.

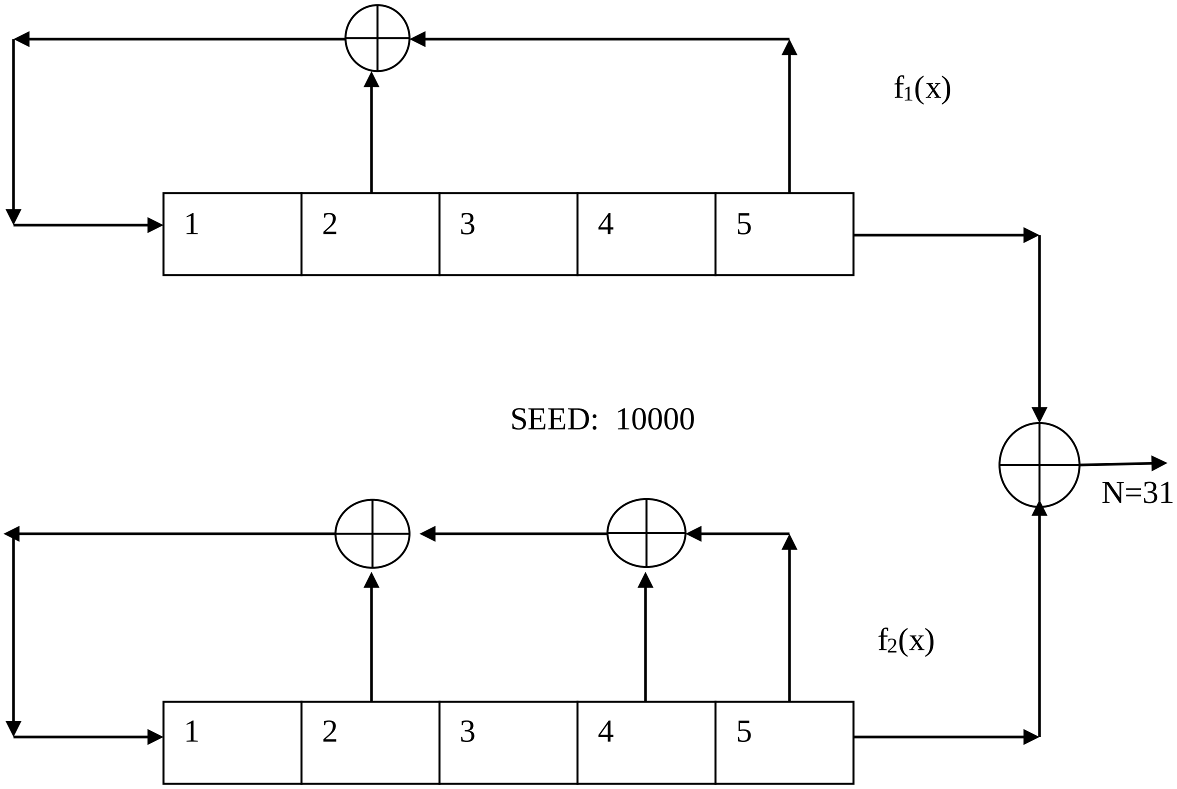


Figure6: Preferred Gold-Code Generator

For example consider two LFSRs with five states and the top locations are [5 2] and [5 4 2] respectively as shown in the above figure.

Here $f_1(x)$ is a 5-bit LFSR with a tap sequence [5, 2]

$f_2(x)$ is a 5-bit LFSR with a tap sequence [5, 4, 2]

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

seq1: $N=2^5-1=31$ bits

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

seq2: $N=2^5-1=31$ bits

After running the code results are shown in the figure 7 for the preferred gold code generator.

Example: PN sequence1 is -1 1 1 -1 -1 -1 1

PN sequence2 is 1 -1 1 -1 -1 1 -1

By using ‘c’ and mat lab the result code is -1 -1 1 1 1 -1 -1

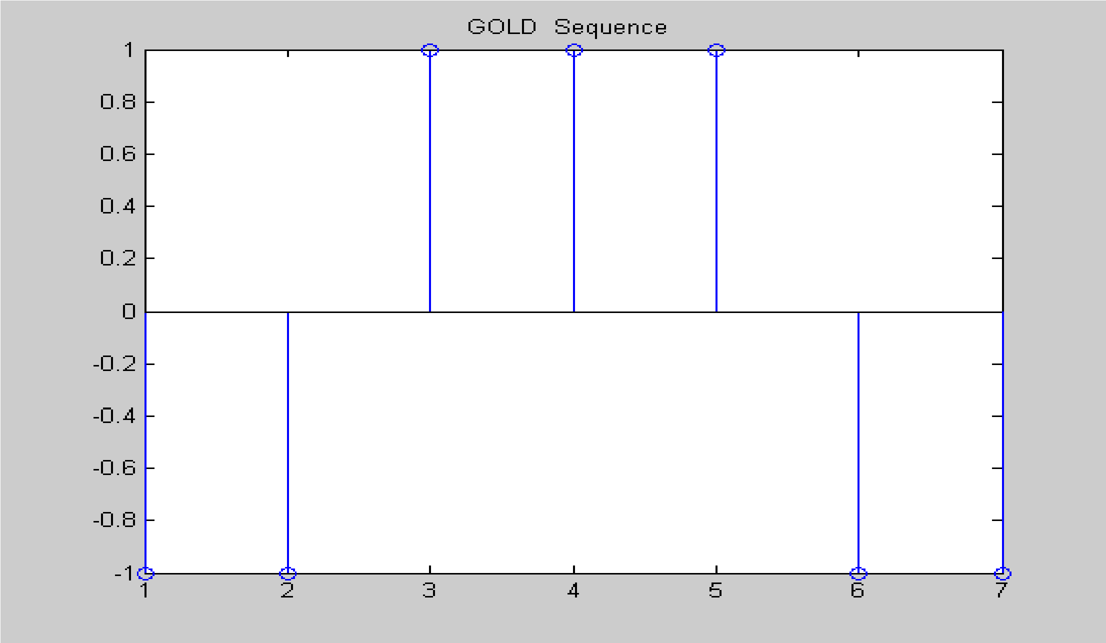


Figure7: Coding Result of Gold Code Generator.

4.3 Spreader

The simplest but critical component of the CDMA Transmitter is the Multiplier or Spreader. The spreading process is done in the system by the spreader.

Spreading Process means that each data bit is converted into 8 or 16 etc bits depending upon the length of the address-code. The input to the Multiplier or Spreader is the digital data and the address-code i.e. represented by the Gold code. The output from the spreader is the result of the data being superimposed (Exclusive-OR) on the address-code.

For example consider that the data is of 3-bits (110) and the address-code is of 8-bits (01010001). Then the result is obtained as below:

a) Superimposition (Exclusive-OR) of ‘1’ on “01010001” is “10101110” which is obtained by Ex-OR of ‘1’ with each bit of the address-code and the result forming the respective bit in the output code.

b) Superimposition (Exclusive-OR) of ‘1’ on “01010001” is “10101110” which is obtained by Ex-OR of ‘0’ with each bit of the address-code.

c) Superimposition (Exclusive-OR) of ‘0’ on “01010001” is “01010001” which is obtained by Ex-OR of ‘0’ with each bit of the address-code.

The final output contains 24-bits and is obtained by the concatenation of the above three individual outputs, which is given as

“10101110 0101110 01010001”

The DS-SS CDMA Transmitter in this project is designed for the 31-bit address-code and the input data of one byte or 8-bit for each clock pulse. This results in the resultant signal being 248-bit long. The address-code signal is also referred as the “Spreading Signal” as it acts as the base for the data signal to be spread. The figure 8 shows the coding result of spreader.

Example: The input message sequence is 1 -1

The gold code is -1 -1 1 1 1 -1 -1

Then spreader output is

By using ‘c’ and mat lab the result code is -1 -1 1 1 1 -1 -1 1 1 -1 -1 -1 1 1

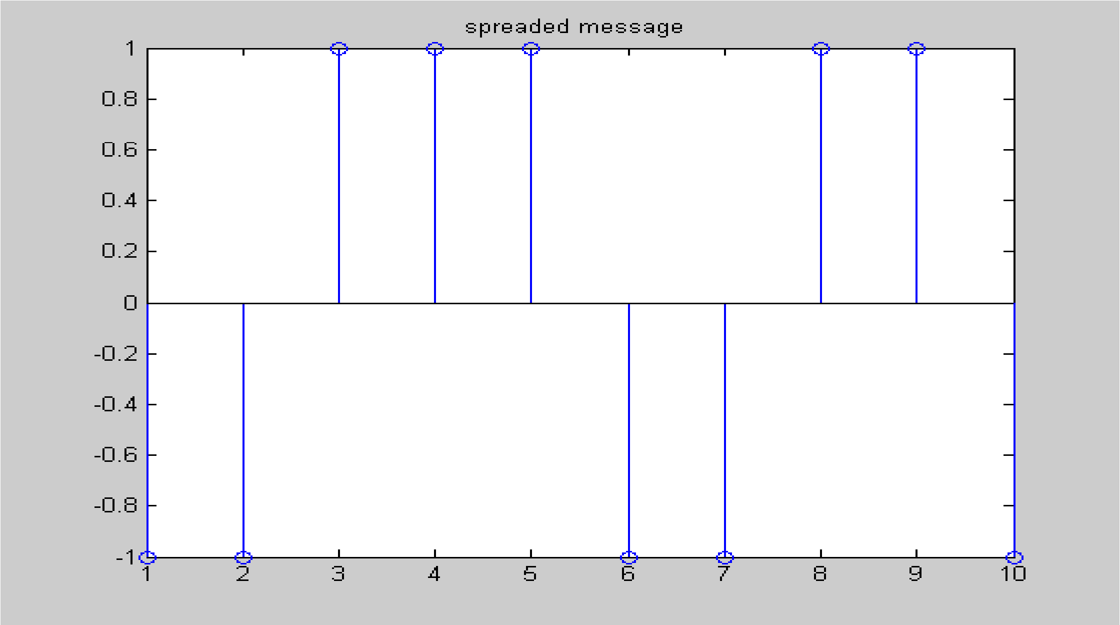


Figure8: Coding Result of Spreader

4.4 Rate-Compatible Punctured Convolutional (RCPC) Codes

Over past year's many new concepts had been introduced on channel coding and even wireless systems are differentiated by varying error prone channels, robust and flexible channel code schemas are introduced.

It is important to note on designing and implementing the channel code for a radio network, some important considerations like adapting flexible for varying channel capacity, possibility of dynamically changing the code rates during transmission plays a key role.

As we know there are two categories of channel code schema exists, Block code and Convolution codes. Using Block code there are major disadvantages especially on soft decision coding, which plays vital role for any bandwidth for efficient wireless communication system. In case of Convolutions codes the soft decision decoding comes naturally by using soft metric in viterbi decoder and more over use of Rate convolutions codes (RCC) makes it possible to the same decoder for all code rates within the family.

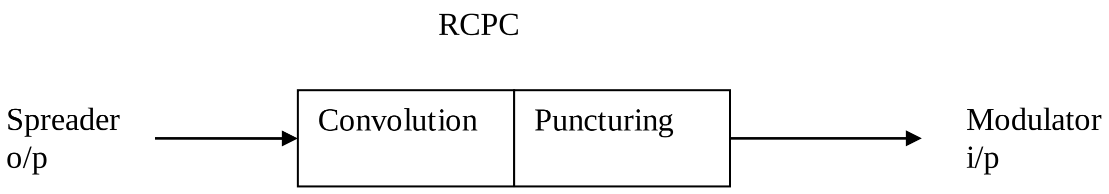
RCC are designed such that lower code rates used for the same code symbols as the higher codes symbols with some redundancy symbols, which can be obtained easily by repeating symbols. Repetition usually deteriorates the system performance, puncturing and nesting. One of the very flexible and power techniques of RCC code family are puncturing for higher code rates and nesting for very low rate and low complexity coding.

Rate-compatible punctured convolutional (RCPC) codes are obtained by puncturing a convolutional code of rate R=1/n and constraint length K, called the parent code. This RCPC codes are specified by its generator polynomials. The rate compatibility criterion decides how we have to puncture the code. The rate compatibility criterion means that the lower rate codes use same coded bits as higher rate codes plus one or more additional bit(s). The puncturing matrix of size n×p determines the number of bits that are going to be punctured. The number of code rates and rate resolution are determined by the number of columns or the period p of the puncturing matrix [15].

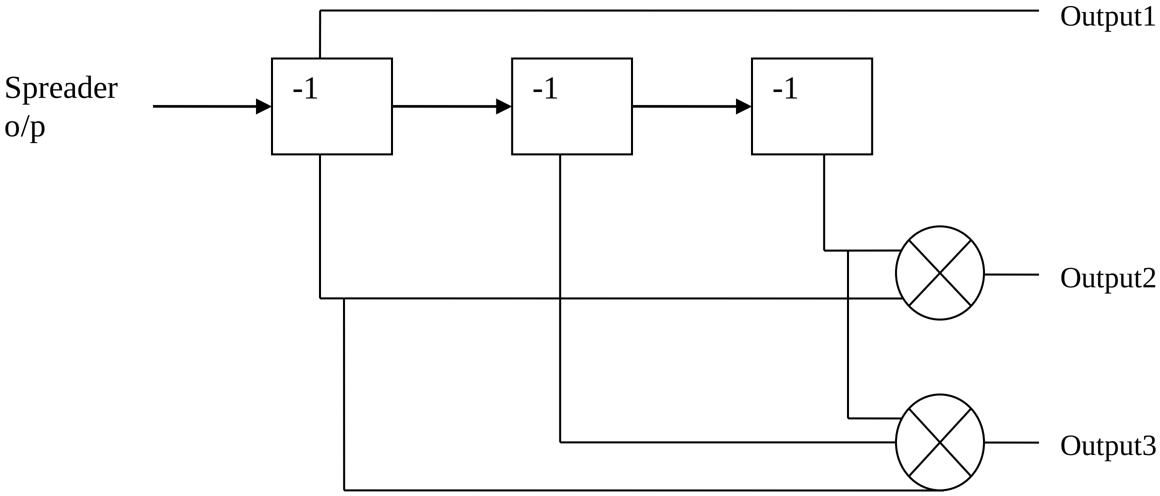
From a parent code of rate 1/n, we obtain a family of (n-1)p different codes with rates R= p/(np), p/(np-1) ……. p/(p+1). The problem with RCPC codes is the limited number of code rates. Now the codes with constraint lengths of 7 are available. But for the cellular system we need codes of constraint lengths of at least 9. So we are combining the RCPC codes and nested convolutional codes to get the desired codes. Nested codes are obtained by extending a code rate of 1/n to a rate of 1/(n+1).

The main objective of RCPC codes (which is suggested by Cybertel software limited Company) is high security because for every user we are allotting a unique puncturing matrix which cannot be accessed by other users. If the bit rate increases the cost increases but the user can bear this cost for high security reasons and for the system to be robust. The main drawback of RCPC codes is increase in bandwidth due to high bit rate.

4.4.1 Convolution and Puncturing process in RCPC



Block Diagram1



Block Diagram2

Figure9: Block diagram 1&2 shows the Convolution and Puncturing process in RCPC

In the convolution process the registers are assigned with a default value of -1. When the output from the spreader is given to the convolution diagram, the first bit from the output is processed through the registers and the values in the registers are shifted by one bit. That means the bit in the first register is shifted to second one and the bit in the second register is shifted to third register so on. Like that the process is continued for all the bits in the output of the spreader.

For example the output from the spreader is -101. Then the first bit from the output ‘1’ is assigned to the first register and the value in that register is shifted to the second one and so on. Then the values in the registers are 1-1-1. For one input bit we will get three output bits.

In the puncturing process the output that obtained from the convolution process is compared with the puncturing matrix. The puncturing matrix will be different for different users. The first bit from the convolution process is compared with the first bit from the puncturing matrix. If the first bit in the matrix is ‘1’we will take that bit and if it is ‘0’ we will cut that bit and compare the second bit from the convolution process with the second bit from the matrix. So in this way we will reduce the convolution output bits.

4.4.2 Search Criterion for Good RCPC Codes

There is no systematic method is available for constructing good rate RCPC codes. So our approach is based on that “good codes generate good codes”. The best known (3, 1) code of memory M is chosen as the mother code and is used to determine the puncturing matrices that will yield a family of best RCPC codes of same M[15]. A true high rate structure for RCPC codes is assumed and extensive computer searches were performed to evaluate the weight spectra of the codes. The criterion of goodness was the maximum free distance d_f and minimum total number of bit errors c_d produced by all incorrect paths with distance $d \leq d_f$ which diverge from the correct path. In general these codes are not as good as the best known high rate codes of the same rate and M [16].

The search procedure as follows

Step 1: Select a known rate 1/3 mother code of memory m and the puncturing period p. Set all elements of matrix A(δ=p) to 1, for δ=p-1 to 1

Step 2: Select an element a_{ij}, in turn, for $1\leq i \leq n$ and $1\leq j\leq p$ and set it to zero.

Step 3: Determine d_f and c_d for $d \geq d_f$ .Select code which gives maximum d_f and minimum c_d for d = d_f ,d_{f+1} , ….. . Discard catastrophic codes and codes of memory less than M.

Step 4: Repeat steps 2 to 3 with the rate-compatibility restriction, to find other puncturing matrices for the family of the entire RCPC codes. The figure 10 shows the result of RCPC code.

Example: The spreader output is -1 -1 1 1 1 -1 -1 1 1 -1 -1 -1 1 1
Then the output from the RCPC coding is

By using ‘c’ and mat lab the result code is -1 -1 -1 -1 -1 1 -1 1 -11
1 1 -1 -1 -1 -1 1 -1 1 -1-1
-1 -1 -1 -1 -1 1 1 -1 1 1 -1

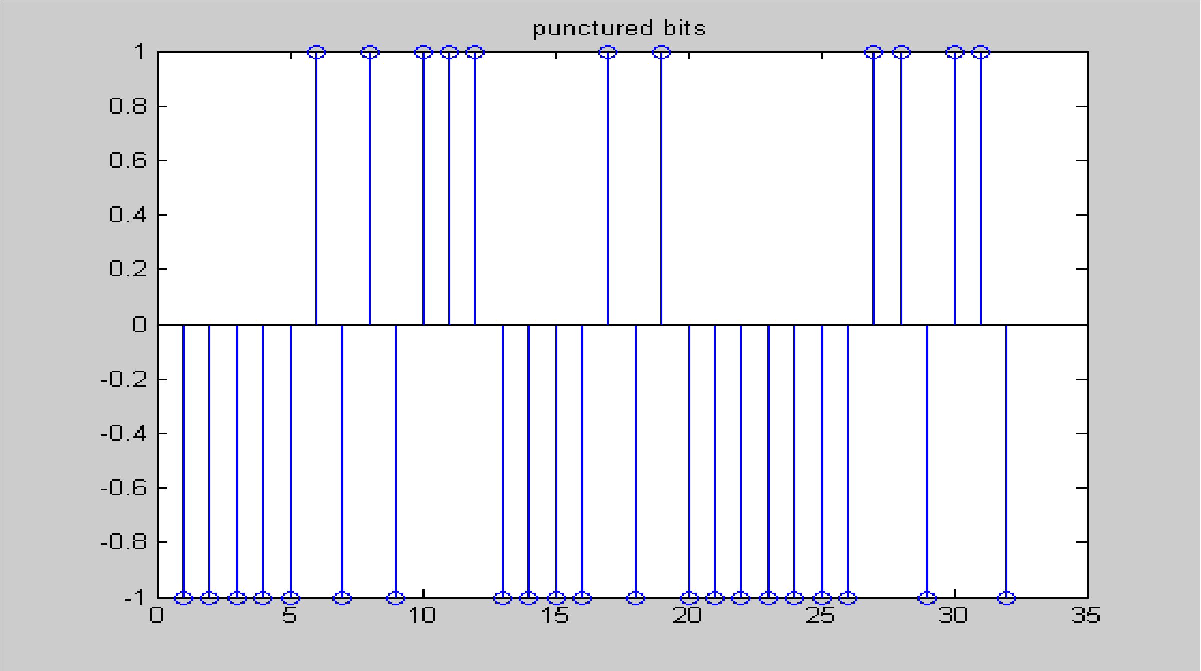


Figure10: Coding Result of RCPC code

4.5 Modulator

The modulation technique used is in this project is Binary Phase Shift Keying. The simulation of the modulator is done by the logic but not using any formula.

The logic used in this is, take two consiqutive bits and compare them if they are same keep them as it is, if they are not same interchange them.

We can initiate the BPSK Output MSB with either “1” or “0”, we have chosen to initiate the MSB with “0”. To obtain the value of MSB-1 bit of the BPSK output, we check the MSB and MSB-1 bits of the Multiplier Output. The figure 11 shows the result of the modulator.

Example: Data after spreading 1 –1 1 –1 –1 –1 1 –1 1 -1 1 1 1 –1

Data after modulation –1 1 –1 1 –1 –1 –1 1 –1 1 1 1 –1 1

The output from the RCPC coding is -1 -1 -1 -1 -1 1 -1 1 -11

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

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

Then the output from the Modulator is

By using ‘c’ and mat lab the result code is -1 -1 -1 -1 1 -1 1 -1 1 -1

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

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

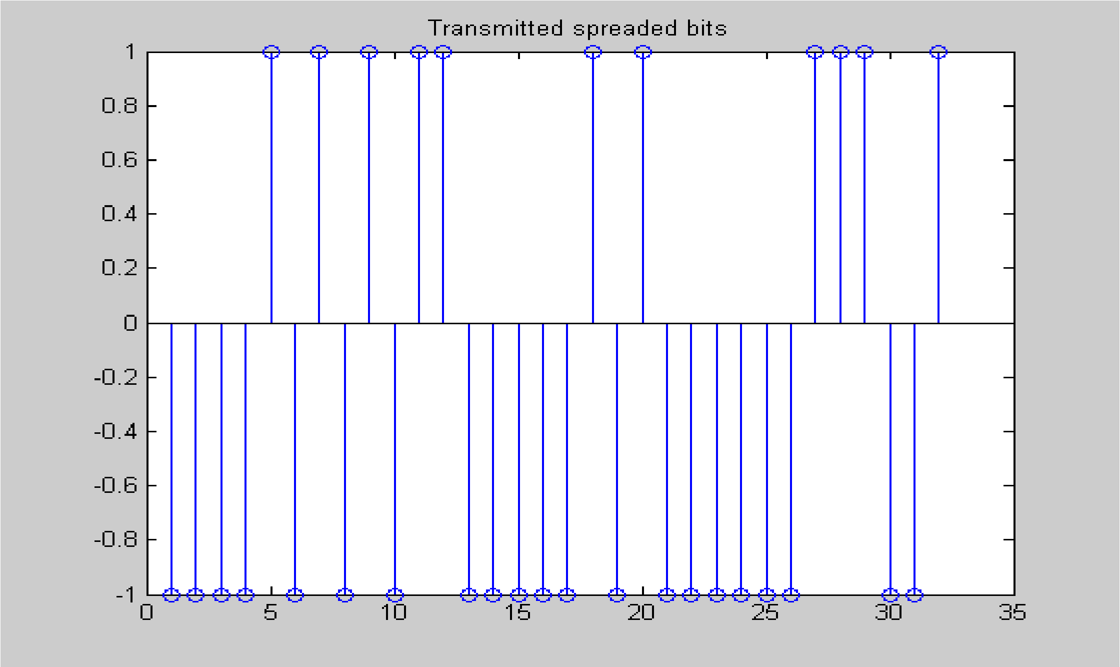


Figure11: Coding Result of Modulator Transmitted spreaded bits

4.6 Demodulator

It works on the same principle as that of Modulator. The modulation technique used is Binary Phase Shift Keying. The simulation has been done the logic explained below and not using the formula.

Take to consecutive bits and compare them if they are same keep them as it is, if they are not same interchange them. The figure12 shows the coding result of demodulator.

Example: Received data –1 1 –1 1 –1 –1 –1 1 –1 1 1 1 –1 1

Data after demodulation 1 –1 1 –1 –1 –1 1 –1 1 -1 1 1 1 –1

The output from the Modulator is -1 -1 -1 -1 1 -1 1 -1 1 -1

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

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

Then the output from the De-modulator is

By using 'c' and mat lab the result code is -1 -1 -1 -1 -1 1 -1 1 -1 1

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

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

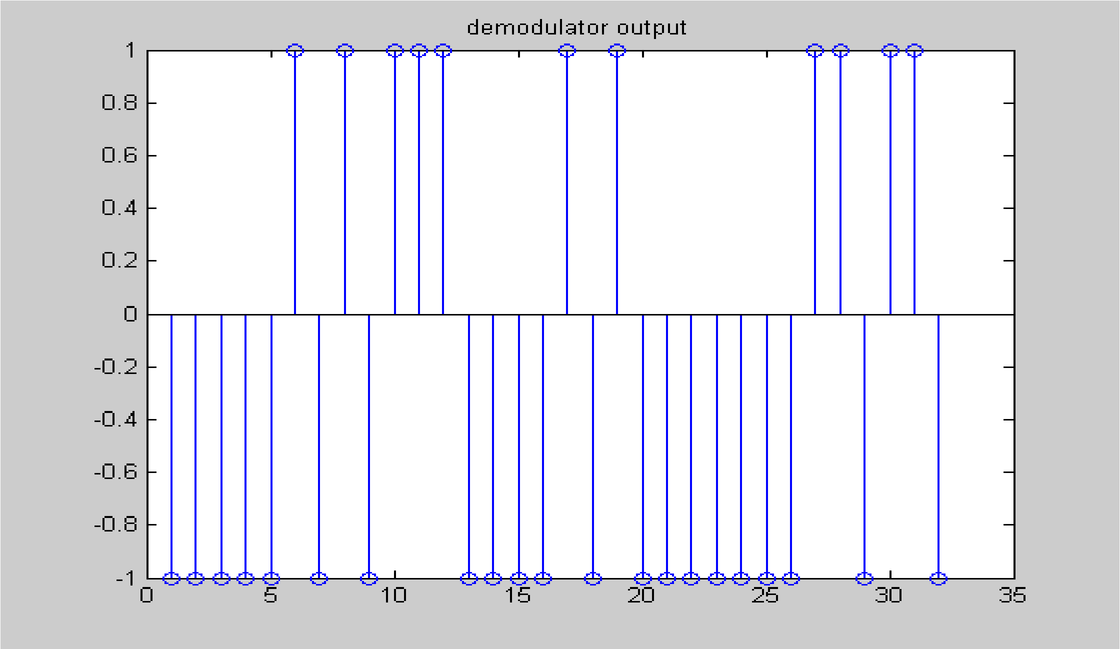


Figure12: Coding Result of Demodulator Output

4.7 RCPC Decoding

The output obtained from the De-modulator is a convoluted one i.e. done by the RCPC coding, so to get the output at receiver end we need RCPC decoding after the De-modulator. The RCPC decoding works same as that of RCPC coding. The figure13 shows the coding result of RCPC Decoding.

Example: The output from the De-modulator is -1 -1 -1 -1 -1 1 -1 1 -1 1

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

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

Then the deconvoluted output is

By using 'c' and mat lab the result code is -1 -1 1 1 1 -1 -1 1 1 -1 -1 -1 1 1

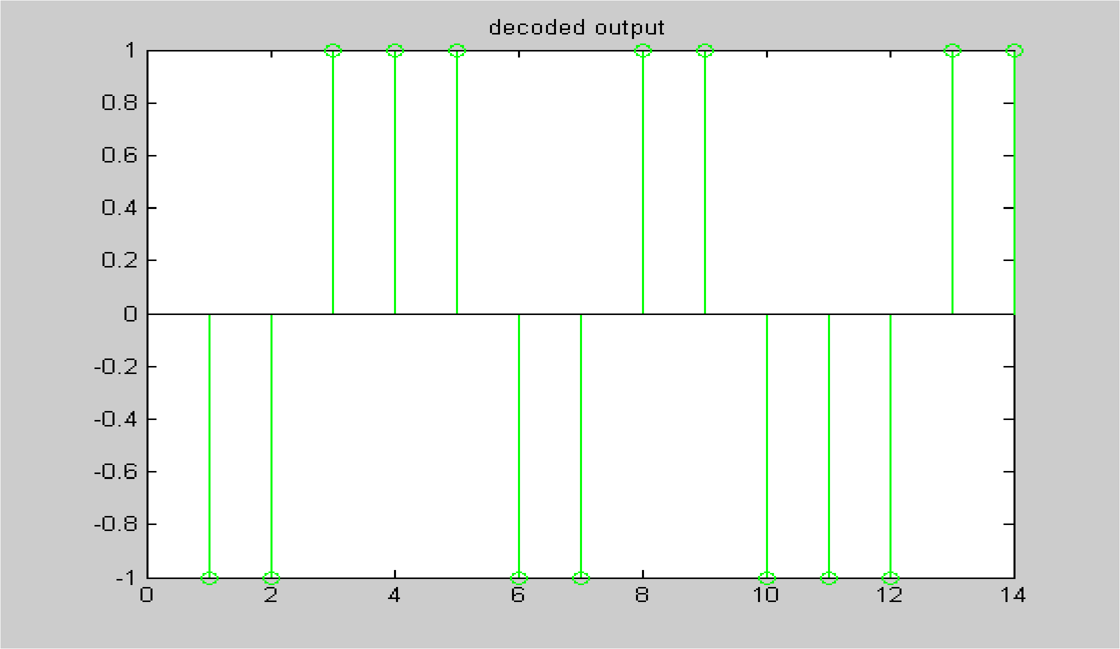


Figure13: Coding Result of RCPC Decoding

5. MATCHED FILTER OPERATION

5.1 Matched Filter

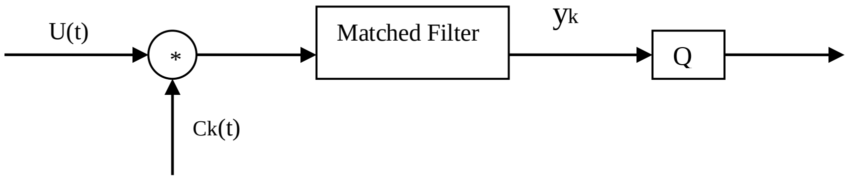


Figure14: Conventional Code Generator (Matched Filter)

Matched filter, which is also know as Conventional Detector is used to reduce the noise signal content in the original signal. The operating principle of a matched filter is that we will send a known signal to the signal that contains noise in that. Then the known signal, or template, correlates with an unknown signal to detect the presence of noise in that. The matched filter will maximize the signal to noise ratio (SNR) in the presence of additive stochastic noise. It is especially for single user designs, as it doesn’t consider any other users in the communication channel, therefore its not so robust to asynchronism, fading channels or PN sequences with cross correlation [18]. The figure 5 shows a model of matched filter or conventional code detector. Assuming a bit synchronous system with no channel dynamics or power variations, the output of the matched filter is defined as

$$y_k = \frac{1}{T}\int u(t)c_k(t)\, dt$$

$$y_k = P_{ik} + MAI_k + w(t)$$

Where P_{ik} is cross correlation density of c_k

MAI is multiple access interference and

W(t) is filtered AWGN

The Matched Filter Operation generated by De-gold Code Generator and Despreader will takes place. The figure15 shows the coding output of Matched Filter.

Example: The deconvoluted output is -1 -1 1 1 1 -1 -1 1 1 -1 -1 -1 1 1

Then the output from the Matched filter is

By using 'c' and mat lab the result code is -1 -1 1 1 1 -1 -1 1 1 -1 -1 -1 1 1

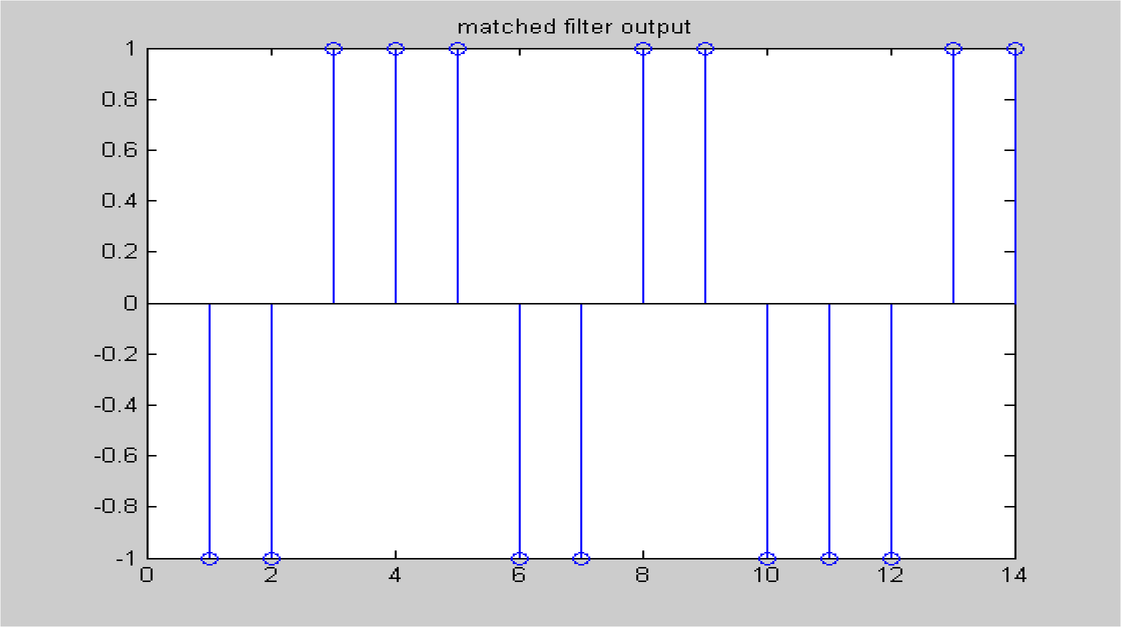


Figure15: Coding Result of Matched Filter

5.2 Receiver Pseudo-Noise Sequence Generator

The PN Sequences can be generated by linear feedback shift registers.

5.2.1 Receiver Linear Feedback Shift Registers

Here the Receiver LFSR works as same as LFSR´s procedure. The figure 16&17 shows the coding result of Receiver PN sequence1 and PN sequence2.

Example: No. of bits in LFSR1 is 3 and sequence is 1 1 –1

Enter the no of taps 2

Tap locations are 1 and 3

Then PN sequence 1 is

By using 'c' and mat lab the result code is -1 1 1 -1 -1 -1 1

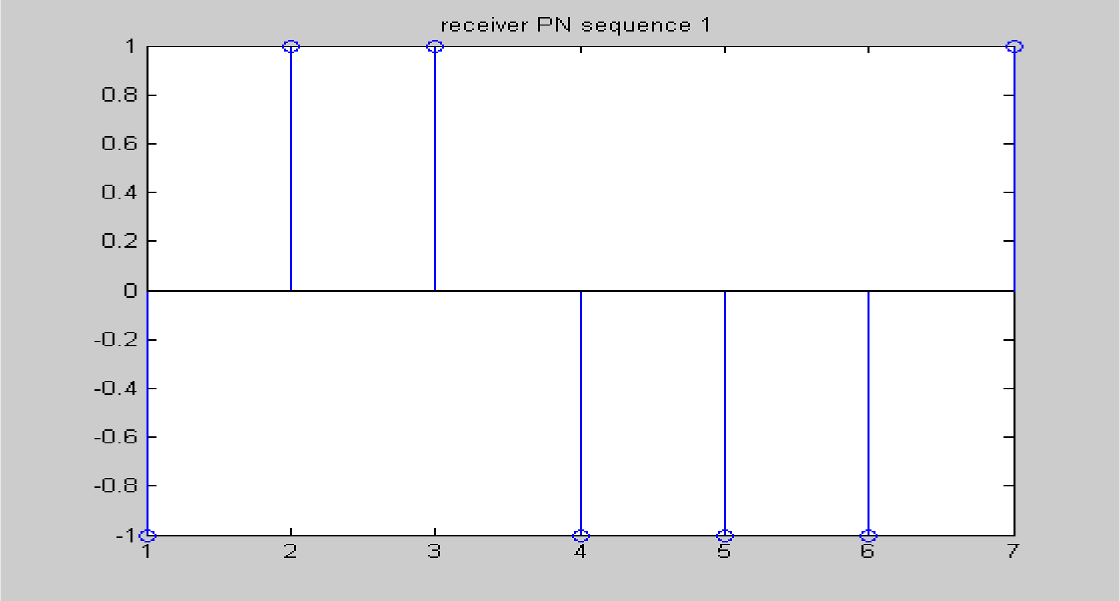


Figure16: Coding Result of Receiver PN sequence1

Example: No. of bits in LFSR 2 is 3 and sequence is 1 -1 1

Enter the no of taps 2

Tap locations are 1 and 2

Then PN sequence 2 is

By using 'c' and mat lab the result code is 1 -1 1 -1 -1 1 -1

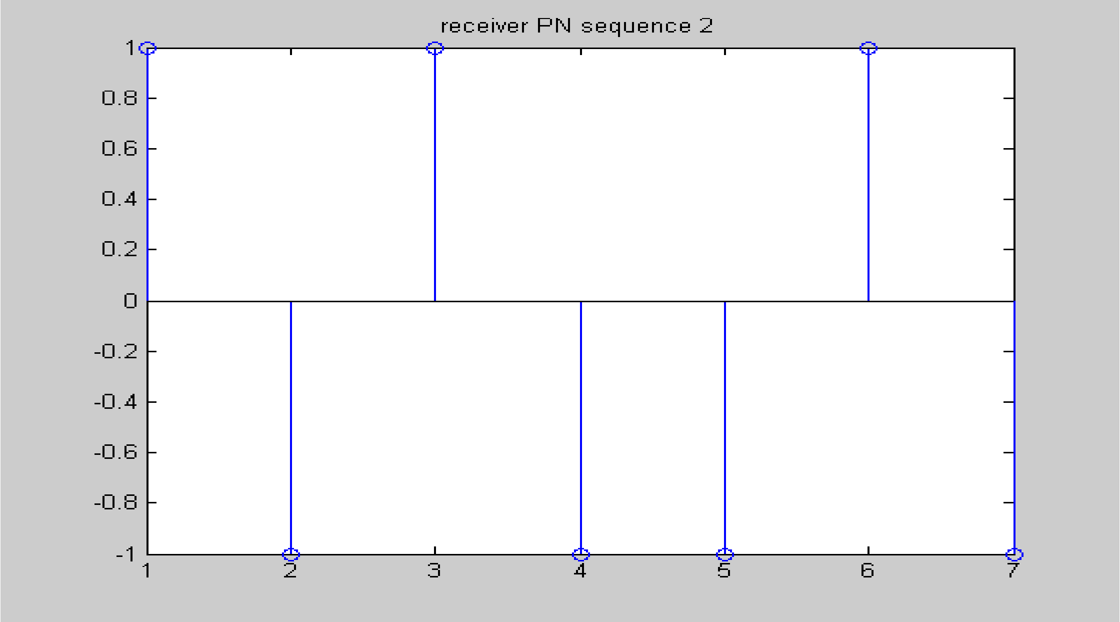


Figure17: Coding Result of Receiver PN sequence2

5.3 De-gold code generator

The output from the Matched filter is de-spreaded by using De-gold code generator into the output at the receiver end. Finally we will get the desired output at the receiver end without any loss of information at the high rate of transmission. The figure 18 shows the coding result of De-Gold code generator.

Example: PN sequence1 is -1 1 1 -1 -1 -1 1

PN sequence2 is 1 -1 1 -1 -1 1 -1

By using 'c' and mat lab the result code is -1 –1 1 1 1 -1 –1

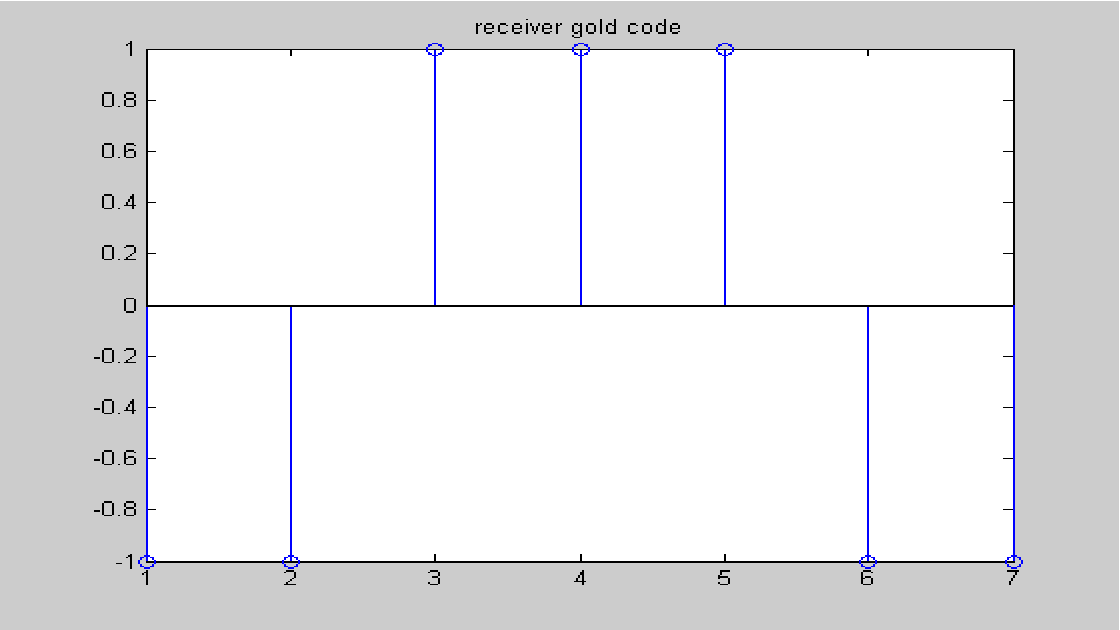


Figure18: Coding Result of Receiver Gold code generator

5.4 De-Spreader

In order to get message bit from spreaded received bits, we are using two techniques.

1. Compare received data with gold code if they are same, received data bit is 1, if they are opposite received data is –1.

2. Multiply received bits with gold code bits and count number of ones and minus ones, and received message bit is decided depending up on the majority of ones or minus ones which ever may be higher.

5.5 Final Data Received

Finally the data is received with out any loss of information at high rate of transmission which input given while data transferring.

Example: The received message bits are 1 –1

6. CONCLUSIONS

There are a large variety of solutions to the problem of multiple access to a repeater by a group of network stations. For continuous traffic, FDMA, TDMA & CDMA access techniques are the most appropriate. Nowadays WCDMA has more advanced techniques which involves with telephone traffic, television transmission and videoconferencing.

Despite its low throughput, CDMA may be preferred to FDMA & TDMA, which loses its efficiency and requires relatively costly earth station equipment respectively. The current system has more user-friendly nature since its features can be easily accessed.

In this project we preferred RCPC because we can transfer the data securely and rapidly with out any loss of information.

The drawback of our project is if we increase the frequency the band width is also increased. So we did our project with less bandwidth.

We conclude that this project can be further extended so that the resultant is much sophisticated than this.

7. APPENDIX

7.1 C Program

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
/*declareations 4 Tx part*/

int lfsr1[50],pns1[50],lfsr2[50],pns2[50],goldcode[50],i,j,k,temp;
int spreader[100],msg[50],l;
int reg1=1,reg2=1,reg3=1,output[500];
int pmatrix[25]={0,1,1,1,1,1,0,1,1,0,1,1,0,1,1,1,1,1,0,1,1,0,1,1,0};
int finaloutput[500],tmp[300],z=0;
int tpseq[10],n,m,len,count;
int len_finaloutput,modulatedoutput[400];

/*declarations 4 Rx part*/

int len_modulatedoutput,demodulatedoutput[400];
int prev_state[4]={1,1,1,1},current_state[4],q,p,out,tempout[50],v=1;
int matchedoutput[100],d,w,g;
int receivedmsg[50],flag1=1,flag2=1,b=1,var;

//clrscr();

/*program for lfsr1 starts here*/

printf("enter the value of no. of bits for generating noise in lfsr1\n");
scanf("%d",&n);
printf("enter the n bit sequence in 1 & -1\n");
for(i=1;i<=n;i++)
scanf("%d",&lfsr1[i]);
printf("enter the no of taps\n ");
scanf("%d",&m);
printf("Enter the tap locations between 1&n\n");
for(i=1;i<=m;i++)
scanf("%d",&tpseq[i]);
len=(pow(2,n)-1);
for(j=1;j<=len;j++)
{
 pns1[j]=lfsr1[n];
 temp=1;
 for(k=1;k<=m;k++)
  {
    temp=temp*lfsr1[tpseq[k]];
   }

 for(i=n;i>=2;i--)
 {
  lfsr1[i]=lfsr1[i-1];
  }
  lfsr1[1]=temp;
 }
 printf("\n Pnseq is\n");
 for(j=1;j<=len;j++)
 printf("%d\t",pns1[j]);
```

/*program for lfsr2 starts here*/

```
printf(" \nenter the value of no. of bits for generating noise in lfsr2\n");
scanf("%d",&n);
printf("enter the n bit sequence in 1 & -1\n");
for(i=1;i<=n;i++)
scanf("%d",&lfsr2[i]);
printf("enter the no of taps\n ");
scanf("%d",&m);
printf("Enter the tap locations between 1&n\n");
for(i=1;i<=m;i++)
scanf("%d",&tpseq[i]);
len=(pow(2,n)-1);
for(j=1;j<=len;j++)
{
 pns2[j]=lfsr2[n];
 temp=1;
 for(k=1;k<=m;k++)
  {
    temp=temp*lfsr2[tpseq[k]];
   }

 for(i=n;i>=2;i--)
 {
  lfsr2[i]=lfsr2[i-1];
  }
  lfsr2[1]=temp;
 }
 printf("\n Pnseq is\n");
 for(j=1;j<=len;j++)
 printf("%d\t",pns2[j]);
```

/*prog 4 gldcode*/

```
for(i=1;i<=len;i++)
{
 goldcode[i]=pns1[i]*pns2[i];
}
printf("\n goldcode is\n");
for(i=1;i<=len;i++)
printf("%d\t",goldcode[i]);
```

/*prog 2 generate spreader output*/

```
printf("\nenter the no. of message bit\n");
scanf("%d",&l);
printf("\nenter the l bit message sequence in 1 and -1\n");
for(i=1;i<=l;i++)
scanf("%d",&msg[i]);
for(j=0;j<l;j++)
{
  for(i=1;i<=len;i++)
  {
    spreader[i+(j*len)]=msg[j+1]*goldcode[i];
  }
}
printf("spreader output is\n");
for(i=1;i<=(l*len);i++)
printf("%d\t",spreader[i]);
```

/*program 4 Rate Compatible Puncturing Convolution coding*/

```
for(i=1,j=0;i<=(l*len)*3,j<=((l*len*3)-1);i++,j++)
{
 reg3=reg2;
 reg2=reg1;
 reg1=spreader[i];
 output[i+(j*2)]=reg1;
 output[i+(j*2)+1]=reg1*reg3;
 output[i+(j*2)+2]=reg1*reg2*reg3;
}
 printf("\n the output is\n");
 for(i=1;i<=(l*len*3);i++)
 printf("%d\t",output[i]);


/*program 4 puncturing matrix*/

j=0;
count=0;
for(i=1;i<=(l*len*3);i++)
{
  if(i<=24)
  {
  loop:
  if(pmatrix[i-(24*j)]==1)
   {
   finaloutput[i-z]=output[i];
   count++;
   }
  else
    {
    tmp[i]=output[i];
    z=z+1;
    }
   }
  else if( i>24 && i<=(24*(j+1)))
    {
                goto loop;
    }
    else if(j<(j+1))
     {
      j++;
      goto loop;
     }
}


  printf("\nthe final output is\n");
  for(i=1;i<=count;i++)
  printf("%d\t",finaloutput[i]);
```

/*program to obtain output from modulator*/

```
len_finaloutput=count%2;
if (len_finaloutput==0)
 {
```

```
 for(i=1;i<=count;i=i+2)
  {
   if(finaloutput[i]==finaloutput[i+1])
    {
         modulatedoutput[i]=finaloutput[i];
         modulatedoutput[i+1]=finaloutput[i+1];
    }
    else
         {
          modulatedoutput[i]=finaloutput[i+1];
          modulatedoutput[i+1]=finaloutput[i];
         }
   }
 }
else
 {
    count=count-1;
    for(i=1;i<=count;i=i+2)
  {
   if(finaloutput[i]==finaloutput[i+1])
    {
         modulatedoutput[i]=finaloutput[i];
         modulatedoutput[i+1]=finaloutput[i+1];
    }
    else
         {
          modulatedoutput[i]=finaloutput[i+1];
          modulatedoutput[i+1]=finaloutput[i];
         }
   }
    modulatedoutput[count+1]=finaloutput[count+1];
    count=count+1;
 }

 printf("\n the modulated output is\n");
 for(i=1;i<=count;i++)
 {
  printf("%d\t",modulatedoutput[i]);
 }
```

/*END OF TRANSMISSION PART*/
/* START OF RECEPTION PART*/

/*program for demodulation of recieved data*/

```
len_modulatedoutput=count%2;
if (len_modulatedoutput==0)
 {
   for(i=1;i<=count;i=i+2)
    {
      if(modulatedoutput[i]==modulatedoutput[i+1])
       {
            demodulatedoutput[i]=modulatedoutput[i];
            demodulatedoutput[i+1]=modulatedoutput[i+1];
       }
       else
             {
              demodulatedoutput[i]=modulatedoutput[i+1];
              demodulatedoutput[i+1]=modulatedoutput[i];
             }
     }
   }
  else
   {
       count=count-1;
       for(i=1;i<=count;i=i+2)
    {
      if(modulatedoutput[i]==modulatedoutput[i+1])
       {
             demodulatedoutput[i]=modulatedoutput[i];
             demodulatedoutput[i+1]=modulatedoutput[i+1];
       }
       else
             {
              demodulatedoutput[i]=modulatedoutput[i+1];
              demodulatedoutput[i+1]=modulatedoutput[i];
             }
     }
       demodulatedoutput[count+1]=modulatedoutput[count+1];
       count=count+1;
   }
      printf("\n the demodulated output is\n");
   for(i=1;i<=count;i++)
   {
    printf("%d\t",demodulatedoutput[i]);
   }
```

/*program 2 get deconvoluted output*/

```
 p=0;
  q=0;
  out=1;
  j=1;
  loop1:
 for(i=1;i<=24;i=i+3)
   {
            if(((i+q)-p)<=count)
            {
   // printf("prev1=%d prev2=%d prev3=%d\n",prev_state[1],prev_state[2],prev_state[3]);
     if(pmatrix[i]==1&&pmatrix[i+1]==1&&pmatrix[i+2]==1)
           {
            current_state[1]=demodulatedoutput[(i+q)-p];
            current_state[2]=demodulatedoutput[((i+q)-p)+1];
            current_state[3]=demodulatedoutput[((i+q)-p)+2];



       //   printf("curr1=%d curr2=%d
curr3=%d\n",current_state[1],current_state[2],current_state[3] );
            if(prev_state[1]==1&&prev_state[2]==1&&prev_state[3]==1)
              {
               if(current_state[1]==-1&&current_state[2]==-1&&current_state[3]==-1)
                 {
                 tempout[j]=-1;
                 out++;
                 }

               else if(current_state[1]==1&&current_state[2]==1&&current_state[3]==1)
                        {
                        tempout[j]=1;
                         out++;
                        }
               }
                else
                 {
                 //   p++;
                      goto loop2;
                 }
                // printf("tempout[%d]=%d\n",j,out);
               prev_state[1]=current_state[1];
               prev_state[2]=current_state[2];
               prev_state[3]=current_state[3];
             }
```

```
else   if(pmatrix[i]==1&&pmatrix[i+1]==1&&pmatrix[i+2]==0)
  {
   current_state[1]=demodulatedoutput[(i+q)-p];
   current_state[2]=demodulatedoutput[((i+q)-p)+1];
   ++p;
   loop2:
   if(prev_state[1]==-1&&prev_state[2]==-1)
         {
          if(current_state[1]==-1&&current_state[2]==-1)
            {
             tempout[j]=-1;
             out++;
            }
           else if(current_state[1]==1&&current_state[2]==1)
             {
             tempout[j]=1;
             out++;
             }
          }
        if(prev_state[1]==1&&prev_state[2]==1)
          {
           if(current_state[1]==1&&current_state[2]==-1)
             {
             tempout[j]=1;
             out++;
             }
            else if(current_state[1]==-1&&current_state[2]==1)
             {
             tempout[j]=-1;
             out++;
             }
             else if(current_state[1]==-1&&current_state[2]==-1)
             {
             tempout[j]=-1;
             out++;

             }
             else if(current_state[1]==1&&current_state[2]==1)
                  {
                  tempout[j]=1;
                  out++;
                  }
           }
          if(prev_state[1]==-1&&prev_state[2]==-1)
```

```
{
 if(current_state[1]==1&&current_state[2]==-1)
    {
    tempout[j]=1;
    out++;
    }
  else if(current_state[1]==-1&&current_state[2]==1)
     {
     tempout[j]=-1;
     out++;
     }
}
if(prev_state[1]==1&&prev_state[2]==-1)
 {
  if(current_state[1]==1&&current_state[2]==1)
    {
    tempout[j]=1;
    out++;
    }
    else if(current_state[1]==-1&&current_state[2]==-1)
    {
    tempout[j]=-1;
    out++;
    }
 }
 if(prev_state[1]==-1&&prev_state[2]==1)
 {
  if(current_state[1]==1&&current_state[2]==1)
    {
    tempout[j]=1;

    out++;
    }
  else if(current_state[1]==-1&&current_state[2]==-1)
    {
    tempout[j]=-1;
    out++;
    }

  }
 if(prev_state[1]==1&&prev_state[2]==-1)
 {
  if(current_state[1]==1&&current_state[2]==-1)
    {
```

```
                    tempout[j]=1;
                    out++;
                    }
                    else if(current_state[1]==-1&&current_state[2]==1)
                     {
                     tempout[j]=-1;
                     out++;
                     }
                 }
                if(prev_state[1]==-1&&prev_state[2]==1)
                 {
                    if(current_state[1]==1&&current_state[2]==-1)
                      {
                      tempout[j]=1;
                      out++;
                      }
                    else if(current_state[1]==-1&&current_state[2]==1)
                     {
                     tempout[j]=-1;
                     out++;
                     }
                    }
                    prev_state[1]=current_state[1];
                    prev_state[2]=current_state[2];

                    prev_state[3]=0;
               }
               j++;
               }
              }
              q=24*v;
              v++;
              while(q<=count)
              goto loop1;
          printf("the deconvoluted decoder output is\n");
          for(i=1;i<=(out-1);i++)
           {
            printf("%d\t",tempout[i]);
           }
```

/*PROGRAM FOR MATCHED FILTER OPERATION*/
/*program for lfsr1 starts here*/

```
printf("\n enter the value of no. of bits for generating noise in lfsr1\n");
  scanf("%d",&n);
```

```
 printf("enter the n bit sequence in 1 & -1\n");
  for(i=1;i<=n;i++)
   scanf("%d",&lfsr1[i]);
 printf("enter the no of taps\n ");
 scanf("%d",&m);
 printf("Enter the tap locations between 1&n\n");
  for(i=1;i<=m;i++)
   scanf("%d",&tpseq[i]);
 len=(pow(2,n)-1);
 for(j=1;j<=len;j++)
 {
   pns1[j]=lfsr1[n];
   temp=1;
 for(k=1;k<=m;k++)
  {

   temp=temp*lfsr1[tpseq[k]];
  }

 for(i=n;i>=2;i--)
 {
  lfsr1[i]=lfsr1[i-1];
 }
  lfsr1[1]=temp;
}
 printf("\n Pnseq is\n");
 for(j=1;j<=len;j++)
 printf("%d\t",pns1[j]);

/*program for lfsr2 starts here*/

printf(" \nenter the value of no. of bits for generating noise in lfsr2\n");
scanf("%d",&n);
printf("enter the n bit sequence in 1 & -1\n");
for(i=1;i<=n;i++)
 scanf("%d",&lfsr2[i]);
printf("enter the no of taps\n ");
scanf("%d",&m);
printf("Enter the tap locations between 1&n\n");
for(i=1;i<=m;i++)
 scanf("%d",&tpseq[i]);
len=(pow(2,n)-1);
for(j=1;j<=len;j++)
 {
  pns2[j]=lfsr2[n];
```

```
   temp=1;
  for(k=1;k<=m;k++)
   {
    temp=temp*lfsr2[tpseq[k]];
   }

  for(i=n;i>=2;i--)
   {
    lfsr2[i]=lfsr2[i-1];
   }
    lfsr2[1]=temp;
  }
   printf("\n Pnseq is\n");
   for(j=1;j<=len;j++)
    printf("%d\t",pns2[j]);
```

/*prog 4 gldcode*/

```
 for(i=1;i<=len;i++)
 {
  goldcode[i]=pns1[i]*pns2[i];
 }
 printf("\n goldcode is\n");
 for(i=1;i<=len;i++)
printf("%d\t",goldcode[i]);
d=0;
w=1;
g=1;
loop4:
for(i=1;i<=len;i++)
{
 if(tempout[i+d]==goldcode[i]||tempout[i+d]==-goldcode[i])
  {
    matchedoutput[i+d]=tempout[i+d];
    w++;
   }
 }
 if(w>len && w<=(out-1))
  {
   d=len*g;
   g++;
   goto loop4;
  }
```

```
printf("\n the matched filter output is\n");
 for(i=1;i<=(out-1);i++)
  printf("%d\t",matchedoutput[i]);
 /*final program 2 recieve the original message bits(despreading operation)*/

 var=0;
 for(i=1;i<=(out-1);i=i+len)
   {
    flag1=1;
    flag2=1;
     for(j=1;j<=len;j++)
      {
            if(matchedoutput[j+var]==goldcode[j])
             {
              flag1++;
             }
             if(matchedoutput[j+var]==-goldcode[j])
             {
              flag2++;
             }
            }
            var=len+var;

       if(flag1>flag2)
            {
             receivedmsg[b]=1;
             b++;
            }
       else if(flag1<flag2)
            {
             receivedmsg[b]=-1;
             b++;
            }
      }
     printf("\n the received message bits are\n");
     for(i=1;i<b;i++)
      printf("%d\t",receivedmsg[i]);
     getch();
   }
```

7.2 MAT LAB Coding

```
/*Program for lfsr1
%clear all
%TL=[];
L=input('enter  the intial state of pns1 gen');
taplr=input('enter tap loc of pns1');
len=length(L);
notap=length(taplr);
for pn=1:1:2^len-1
   pn1(pn)=L(len);
   %fed=L(5)*L(2);
   %fed=L(1)*L(3)*L(4);
   %TL=[TL;L];
   fed=1;
   for t=1:1:notap
      fed=fed*L(taplr(t));
   end
   for m=len:-1:2
      L(m)=L(m-1);
   end
   L(1)=fed;

end
figure;
stem(pn1);
title('PN Sequence 1');

/*Program for lfsr2
%clear all
%TL=[];
L=input('enter  the intial state of pns2 gen');
taplr=input('enter a int no. tap loc of pns2');
len=length(L);
notap=length(taplr);
for pn=1:1:2^len-1
   pn2(pn)=L(len);
   %fed=L(5)*L(2);
   %fed=L(1)*L(2)*L(3);
   %TL=[TL;L];
   fed=1;
   for t=1:1:notap
      fed=fed*L(taplr(t));
   end
   for m=len:-1:2
```

```
        L(m)=L(m-1);
    end
    L(1)=fed;
end
figure;
stem(pn2)
title('PN Sequence 2');

/*Program for gold code generator
for i=1:1:(2^len)-1
    code(i)=pn1(i)*pn2(i);
end
figure;
stem(code);
title('GOLD  Sequence ');
/*Program for spreader
%clear all
%pns1;
%pns2;
%gold;
data=input('enter the message');
slen=2^len-1;
dlen=length(data);
for dt=1:1:dlen
    for d=1:1:slen
        spr((slen*(dt-1))+d)=data(dt)*code(d);
        %k=k+1;
    end
end
figure;
stem(spr);
axis([1 1 -1 1])
title('spreaded message ');

/*Program for rcpc coding
cco1=[];
s1=spr;
%s1=tra;
%s1=[1 1 1 -1 -1 -1 1 -1 1];
reg11=[1];
reg21=[1];
reg31=[1];

for i=1:1:length(s1)
    reg31=reg21;
    reg21=reg11;
    reg11=s1(i);

    mod11=reg11;
    mod21=reg11*reg31;
    mod31=reg11*reg21*reg31;

    cco1=[cco1; mod11 mod21 mod31];
end


%puncturing matrix for 3/4
%
P=[1 1 1 1 1 1 1 1;
   1 1 1 1 1 1 1 1;
   1 0 0 0 1 0 0 0];
txpcco=[];
[nr nc] = size(cco1);

for j = 1:8:nr

    if j+7 > nr
        tp=nr-j;
    else
        tp=7;
    end

    blk=cco1(j:j+tp,:);
    blk=blk';

    for p1=1:1:tp+1
        for mo=1:1:nc
            if P(mo,p1) == 1
                txpcco=[txpcco blk(mo,p1)];
            end
        end
    end
end


figure;
stem(txpcco);
title('punctured bits');

/*Program for modulator
```

```
modi=txpcco;
lemodi=length(modi);
if ( mod(lemodi,2) ~= 0 )
   tmod(lemodi)=modi(lemodi);
   lemodi=lemodi-1;
end
for tb=1:2:lemodi%length(data)*k
   if modi(tb)==modi(tb+1)
      tmod(tb)=modi(tb);
      tmod(tb+1)=modi(tb+1);
   else
      tmod(tb)=modi(tb+1);
      tmod(tb+1)=modi(tb);
   end
end

figure;
stem(tmod);
title('Transmitted spreaded bits');
%tra=tmod;

/*Or program for transformation
clear all;

pns1;
pns2;
gold;

spread;

cod1;
punc;

modu;
% data=[1    -1    -1     1     1    -1] ;

save test;

/*Program for demodulator
demodi=tmod;
ledemodi=length(demodi);

if ( mod(ledemodi,2) ~= 0 )
   rdemod(ledemodi)=demodi(ledemodi);
   ledemodi=ledemodi-1;
```

```
end

for tb=1:2:ledemodi%length(data)*k

   if demodi(tb) == demodi(tb+1)
      rdemod(tb) = demodi(tb);
      rdemod(tb+1) = demodi(tb+1);
   else
      rdemod(tb)=demodi(tb+1);
      rdemod(tb+1)=demodi(tb);
   end

end

figure
stem(rdemod)
title('demodulator output');
%save test11
/*Program for rlfsr1
%clear all
TL=[];
L=input('enter  the intial state of pns1 gen of the receiver');
taplr=input('enter a int no. tap loc of pns1 of the receiver');
len=length(L);
notap=length(taplr);
for pn=1:1:2^len-1
   rpn1(pn)=L(len);
   %fed=L(5)*L(2);
   %fed=L(1)*L(3)*L(4);
   TL=[TL;L];
   fed=1;
   for t=1:1:notap
      fed=fed*L(taplr(t));
   end

   for m=len:-1:2
      L(m)=L(m-1);
   end
   L(1)=fed;

end
figure;
stem(rpn1);
title('receiver PN sequence 1');
```

/*Program for lfsr2

```
%clear all
TL=[];
L=input('enter  the intial state of pns2 gen of the receiver');
taplr=input('enter a int no. tap loc of pns2 of the receiver');
len=length(L);
notap=length(taplr);
for pn=1:1:2^len-1
   rpn2(pn)=L(len);
   %fed=L(5)*L(2);
   %fed=L(1)*L(2)*L(3);
   TL=[TL;L];
   fed=1;
   for t=1:1:notap
      fed=fed*L(taplr(t));
   end
   for m=len:-1:2
      L(m)=L(m-1);
   end
   L(1)=fed;

end
figure;
stem(rpn2);
title('receiver PN sequence 2');
```

/*Program for degold generator

```
for i=1:1:(2^len)-1
   rcode(i)=rpn1(i)*rpn2(i);
end
figure;
stem(rcode);
title('receiver gold code');
```

/*Program for rcpc decoding

```
%pns1;
%pns2;
%gold;
%spread;
%cod1;
%punc;
%modu;
prev1=[1 1 1];   % used as prev state in order to decode the symbol
rece1=[];        % decoded message
```

```
deconc1=[];      % deconvolutio output
k=1;      % counts numbe rof bits received
%lon=0;
%[rows co]=size(cco1); for i=1:1:rows rpcco=rdemod;

repcco=rdemod;
%repcco=txpcco;
%rle=length(repcco);
%nit=floor(rle/8);
%dnit=mod(rle,8);

%dncod=rpcco(rle-dnit+1:rle); repcco=repcco(1:rle-dnit);
rele=length(repcco);

%for KK=1:k:rele

 %k=1;
while k <= rele

for i=1:1:8

    if k > rele
        break
    end

    curr1=[];
    L=0;

    for j=1:1:3
        if P(j,i) == 1
            L=L+1;
        end
    end

    %curr1=[repcco(KK+k-1:KK-1+k+L-1)];
    curr1=[repcco(k:k+L-1)];
    k=k+L;

    if length(curr1) == 2
        prev1=[prev1(1:2)];
    end
    if length(prev1) == 2
        curr1=[curr1(1:2)];
    end
    %if length(rece1)==11
 %   dddd=0;
 %end
%curr1=cco1(i,:); if length(rece1) == 38
 %   lon==1000;
 %end

 if length(curr1) == 3 & length(prev1) == 3

if prev1 == [ 1 1 1 ]
   if curr1 == [ 1 1 1 ]
      deconc1 = [ deconc1 1 ];
   elseif curr1 == [ -1 -1 -1 ]
      deconc1 = [ deconc1 -1 ];
   end
elseif prev1 == [ -1 -1 -1 ]
   if curr1 == [ 1 1 -1 ]
      deconc1 = [ deconc1 1 ];
   elseif curr1 == [ -1 -1 1 ]
      deconc1 = [ deconc1 -1 ];
   end
elseif prev1 == [ 1 1 -1 ]
   if curr1 == [ 1 -1 -1 ]
      deconc1 = [ deconc1 1 ];
   elseif curr1 == [ -1 1 1 ]
      deconc1 = [ deconc1 -1 ];
   end
elseif prev1 == [ -1 -1 1 ]
   if curr1 == [ 1 -1 1 ]
      deconc1 = [ deconc1 1 ];
   elseif curr1 == [ -1 1 -1 ]
      deconc1 = [ deconc1 -1 ];
   end
elseif prev1 == [ 1 -1 -1 ]
   if curr1 == [ 1 1 1 ]
      deconc1 = [ deconc1 1 ];
   elseif curr1 == [ -1 -1 -1 ]
      deconc1 = [ deconc1 -1 ];
   end
elseif prev1 == [ -1 1 1 ]
   if curr1 == [ 1 1 -1 ]
      deconc1 = [ deconc1 1 ];
   elseif curr1 == [ -1 -1 1 ]
      deconc1 = [ deconc1 -1 ];
   end
elseif prev1 == [ 1 -1 1 ]
```

```
      if curr1 == [ 1 -1 -1 ]
         deconc1 = [ deconc1 1 ];
      elseif curr1 == [ -1 1 1 ]
         deconc1 = [ deconc1 -1 ];
      end
   elseif prev1 == [ -1 1 -1 ]
      if curr1 == [ 1 -1 1 ]
         deconc1 = [ deconc1 1 ];
      elseif curr1 == [ -1 1 -1 ]
         deconc1 = [ deconc1 -1 ];
      end
   end

else
   if prev1 == [ 1 1 ]
      if curr1 == [ 1 1 ]
         deconc1 = [ deconc1 1 ];
      elseif curr1 == [ -1 -1 ]
         deconc1 = [ deconc1 -1 ];
      elseif curr1 == [ 1 -1  ]
         deconc1 = [ deconc1 1 ];
      elseif curr1 == [ -1 1 ]
         deconc1 = [ deconc1 -1 ];
      end
   elseif prev1 == [ -1 -1 ]
      if curr1 == [ 1 1 ]
         deconc1 = [ deconc1 1 ];
      elseif curr1 == [ -1 -1 ]
         deconc1 = [ deconc1 -1 ];
      elseif curr1 == [ 1 -1 ]
         deconc1 = [ deconc1 1 ];
      elseif curr1 == [ -1 1 ]
         deconc1 = [ deconc1 -1 ];
      end
   elseif prev1 == [ 1 -1 ]
      if curr1 == [ 1 1 ]
         deconc1 = [ deconc1 1 ];
      elseif curr1 == [ -1 -1 ]
         deconc1 = [ deconc1 -1 ];
      elseif curr1 == [ 1 -1 ]
         deconc1 = [ deconc1 1 ];
      elseif curr1 == [ -1 1 ]
         deconc1 = [ deconc1 -1 ];
      end
   elseif prev1 == [ -1 1 ]
```

```
        if curr1 == [ 1 1 ]
           deconc1 = [ deconc1 1 ];
        elseif curr1 == [ -1 -1 ]
           deconc1 = [ deconc1 -1 ];
        elseif curr1 == [ 1 -1 ]
           deconc1 = [ deconc1 1 ];
        elseif curr1 == [ -1 1  ]
           deconc1 = [ deconc1 -1 ];
        end
    end

end
prev1=curr1;

end
%for loop ends
end
%while loop ends

rece1=deconc1;
figure;
stem(rece1,'g');
title('decoded output');
/*Program for matched filter
for i=1:slen:dlen*slen
    a=rece1(i:i+slen-1);
    if a==rcode | a==-rcode
        match(i:i+slen-1)=a;
    end
end

figure;
stem(match,'b');
title('matched filter output ');

/*Program for despreader
k=1;
for i=1:slen:dlen*slen;

    a=match(i:i+slen-1);
    if a==rcode
        dout(k)=1;
    else
        dout(k)=-1;
    end
    k=k+1;
    %for j=1:1:slen
     %   de(j)=a(j)*rcode(j);
     % end
%    ones=0;
%    mones=0;
%    for j=1:1:slen
 %       if de(j) == 1
  %           ones=ones+1;
  %     elseif de(j) == -1
  %           mones=mones+1;
  %     end
  %  end
  %
   %if ones > mones
    %   dataout(k)=1;
    %else
    %    dataout(k)=-1;
    %end
    %k=k+1;
end
figure
stem(dout,'b')
title('data received ');

/*or program for receiver
demodu;

rpns1;
rpns2;
rgold;

dcod12;

matchf;
despread;
```

8. BIBLIOGRAPHY

1. ‘‘Computer networks’’ by Andrew S. Tanenbaum Fourth Edition, Publisher: Prentice hall, Publish date: August 09, 2002.

2. http://en.wikipedia.org/wiki/Mobile_phone
3. About the different generations and technologies-

http://www.nawras.com.om/glossary_a-e.shtml

http://www.itu.int/osg/spu/ni/3G/technology/index.html

4. ‘‘System-Defined Design and Performance of RFIC's for WCDMA

Mobile Radio Terminals’’ by L. Maurer1, C. Lanschützer2, A. Springer3, R. Weigel4 - http://www.iis.ee.ethz.ch/nwp/lemon/pub/Maurer_E_I_2001.pdf

http://en.wikipedia.org/wiki/W-CDMA
http://www.devx.com/wireless/Door/11329

5. http://whitepapers.zdnet.co.uk/0,1000000651,260151429p,00.htm

6. http://www.sss-mag.com/ss.html

7. http://www.mobiledia.com/glossary/271.html

8. http://www.wtec.org/loyola/wireless/02_04.htm

9. ‘‘Satellite Communications System’’ by Michele Bousquet, Gerard Maral, Published 2002 John Wiley ans Sons, ISBN 0471496545.

10. http://www.mobilealmanac.com/glossary/8_g-TDMA.html

11. http://www.funsms.net/cdma.htm

12. ‘‘Broadband Power line communications Networks’’ by Halid Hrasnica, Abdelfatteh, 2005 ISBN**:** 9780470857410

13. http://homepage.mac.com/afj/lfsr.html

14. http://www.xilinx.com/bvdocs/appnotes/xapp217.pdf

15. http://www.ist-phoenix.org/documents/D2.2a_v1.0_final.pdf

16. http://www.elec.mq.edu.au/shared/publications/abstracts/CameraReady_Manuscript_4_WITSP06.pdf

‘‘Rate-compatible punctured convolution codes (RCPC codes) and their applications’’ by Hagenauer. J and DFVLR in April 1988(vol 36)

17. New Rate-Compatible Punctured Convolutional Codes for Viterbi Decoding

L. H. C. Lee and J. Sodha -http://www.elec.mq.edu.au/shared/publications/abstracts/Camera-Ready_Manuscript_4_WITSP06.pdf

18. http://en.wikipedia.org/wiki/Matched_filter

http://bard.ece.cornell.edu/downloads/sim_tools/spanc_me/report/node11.html

19. ‘‘Digital Communications (4th edition) ’’ by John G.Proakis, published on August 2000 McGraw-Hill Science Engineering.

20. Some notes on ‘‘Rate-compatible punctured turbo codes (RCPTC) design’’ by Babich, F. Motorsi and G. Vatta, F. in may 2004(vol 52) & ‘‘Rate compatible punctured turbo coding for W-CDMA’’ by Hasung Kim Struber, G.L. in Dec 2000, ISBN: 0-7803-5893-7.

21. ‘‘Turbo-coded ARQ schemes for DS-CDMA data networks over fading and shadowing channels: throughput, delay, and energy efficiency ’’ by Tingfang Ji stark, W.E. published on august 2000(vol 18), ISSN: 0733-8716.

22. ‘‘Mobile Communications ’’ by Jochen Schiller, published on 1999 by Addison-Wesley Longman Publishing Co., Inc, ISBN: 0-201-39836-2.

23. For C Language ‘‘Let us C ’’ by Yashavant Kanetkar, published on 2002 by BPB publications, ISBN: 8176566217 &

24. ‘‘MATLAB- Signal Processing Toolbox User’s Guide ’’ by The Math works, Inc, published on 1996 & ‘‘Getting Started with MATLAB 7: A Quick Introduction for Scientists and Engineers ’’ by Rudra Pratap, published on 2006, ISBN: 0-19-517937-4.