SEQUENCES FOR INTER NODE B SYNCHRONIZATION IN WCDMA TDD MODE

KHIT YOK-KHIONG
(B.Eng (Hons) NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2004

ACKNOWLEDGEMENTS

I would like to sincerely thank Professor Tjhung Tjeng Thiang for his continuing guidance, support and encouragement throughout this project. He gave several constructive suggestions for the project which help me a lot in improving my thesis. He has been in the area of communications for many years, hence his insight and broad knowledge in this area make every discussion with him valuable and beneficial. Indeed, he spends much effort and time on my project and thesis. I am truly grateful to have him as my supervisor.

I would also like to take this opportunity to sincerely thank Dr Wang Yuhong. Her working experience has enabled me to gain better understanding in my project. Her help in finalizing my thesis is greatly and deeply appreciated. I am truly blessed to have her assistance in my project.

TABLE OF CONTENTS

SUMMARY

In order to prevent inter-slot interference and enhance the system capacity in Wideband Code Division Multiple Access (WCDMA) Time Division Duplex (TDD) mode of communication, synchronization between the base stations is necessary. Although the most straightforward way of achieving synchronization is by using the global positioning system (GPS), 3rd Generation Partnership Project (3GPP) hopes to achieve a totally independent and self-reliant communication system. Moreover the cost of deploying the GPS is high. An alternative solution proposed by 3GPP uses Concatenated Extended Complementary (CEC) sequences with multiple code offsets to synchronize the base stations via air-interface since this would reduce the costs of base stations. However cross correlation functions between CEC sequences exhibits sharp spikes. Thus the synchronization burst of one cluster may interfere with the synchronization of neighboring cluster. We propose new synchronization sequences, formed by concatenating two *m*-sequences, with relatively better cross correlation properties than the proposed CEC sequences. The new sequences have comparable autocorrelation property as compared to the CEC sequences and better cross correlation properties. In this thesis, we will also show how the new concatenated sequences are generated and analyze their correlation properties.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 EVOLUTION OF MOBILE COMMUNICATION SYSTEMS

With the advance in Very Large Scale Integration (VSLI) and computer technology, we have witnessed in the past two decades a phenomenal growth and proliferations of mobile communication services. In the early stages, we saw the first generation (1G) mobile communication systems such as Analogue Mobile Phone Service (AMPS) and Total Access Communication System (TACS) [1]. These systems can only transmit analogue signals, and has a limited spectrum (800MHz to 900MHz) with 30kHz bandwidth for each voice channel [1]. However there is no room for spectral growth and this resulted in low calling capacity. Subsequently, second generation (2G) mobile communication systems such as the Global System for Mobile Communications (GSM) [2] and IS-95A based on Code Division Multiple Access (CDMA) technology have emerged and replaced the older analogue systems. These digital mobile communication systems provide broader coverage than 1G. The data rate for 2G systems ranges from 9.6kbps for GSM [2] to 14.4kbps for IS-95A CDMA [3]. However, due to the requirement for higher data rate, 2.5G mobile communication systems such as the General Packet Radio Service (GPRS) with a data rate of 117kbps [2] and Enhanced Data-rates for GSM Evolution (EDGE) [3], [4] with the data rate of 384kbps have emerged. Currently this 2.5G system is being installed in countries around the world, which have already adopted GSM-based technology. A

packet data service provides all the benefits for data services and, in addition, brings about the concept of paying-per-byte, where users are charged not for time but for the amount of information transferred. EDGE is also gaining in importance mostly because it promises to help older GSM systems remain competitive with the newer and faster networks. However as mobile communication system advances, it is expected to provide mobile multimedia capabilities allowing users to initiate and receive speech, video and digital data smoothly. The 2.5G communication system also allows greater mobility enabling users to provide and receive service anywhere in the world. In order to increase the data rate, 2G and 2.5G communication systems are gradually being replaced by 3G systems such as Wideband Code Division Multiple Access (WCDMA) and CDMA-2000 [5], [6]. 3G systems support a data rate of 384kbps for mobile communication and 2Mbps for quasi-stationary communications [7]. Using 3G, people can access all kinds of digital information, like music, photos, video and television with its higher data rate. Moreover, electronic payment is also made possible. A table summarizing the evolution of Wireless Systems is shown in Table 1.1.

	1G	2G	2.5G	3G
System	Analogue	Digital	Digital	Digital
Major Systems	AMPS, TACS	GSM, CDMA and other TDMA system	GPRS and EDGE	WCDMA and CDMA-2000
Application	Voice	Voice + little Circuit- switch Data	Similar to 2G + Low Rate Packet – Switch Data	Voice + Packet-switch Data
Speed		9.6kbps - 14.4kbps 9.6kbps for GSM 14.4kbps for IS-95A CDMA	117kbps for GPRS & 384kbps for EDGE	384kbps for pedestrian & 2Mbps for stationary
Properties	Incomplete coverage, no room for spectral growth and poor sound quality	More secure, data services available, broader coverage, allow more user, better sound quality	Similar to 2G but provide higher data rate	Multimedia data, positioning capability, connection to Internet, always connected

Table 1.1: Evolution of Wireless Communication

1.2 3rd GENERATION PARTNERSHIP PROJECT (3GPP)

As mentioned in the previous section, 3G communication system has been designed to provide high-speed data services and its development is based on the initiative of ITU (International Telecommunication Union) so-called IMT-2000 standards (International Mobile Telecommunication 2000). Based on the ITU definition, 3G systems must provide a minimum of 144 kbps packet-data service [3]. The first release of specifications for 3G mobile communication systems is based on 3rd Generation Partnership Project (3GPP).

The Universal Mobile Telecommunication System (UMTS) specified by the 3GPP and based upon WCDMA consists of both Frequency Division Duplex (FDD) and Time Division Duplex (TDD) transmission mode. In FDD, two different bands of frequencies are used for transmission and reception while in TDD, two different time slots are used instead. In 3GPP specification, the frequency bands that are used for both FDD and TDD are illustrated in Fig 1.1

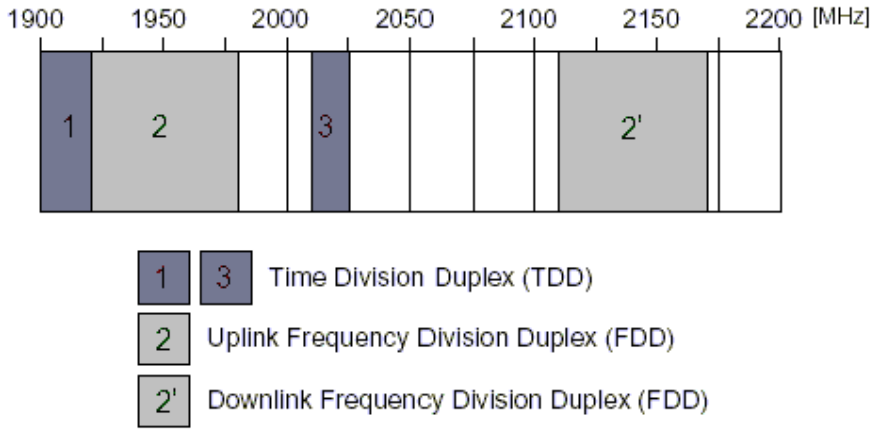


Fig 1.1: Frequency Band used for FDD and TDD

1.3 OVERVIEW OF THE WCDMA NETWORK

Fig 1.2 shows a block diagram of the WCDMA network.

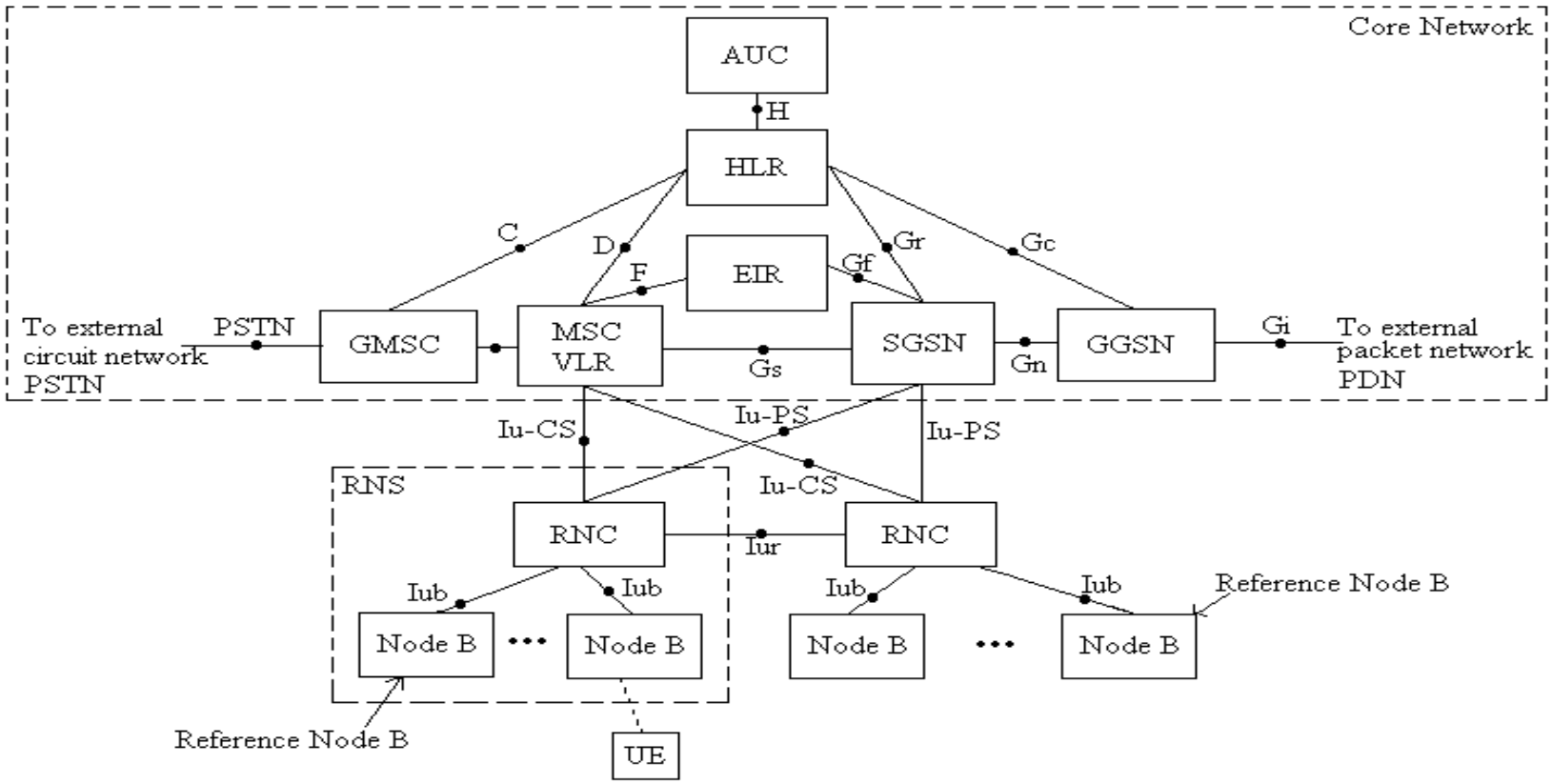


Fig 1.2: Architecture of the WCDMA network

The core network of the WCDMA is similar to the GSM network. Therefore GSM and WCMDA are compatible. The core network is divided into packet and circuit switched paths. The circuit switched traffic is handled by the MSC/VLR. At the end of the line is the GMSC. While packet traffic is handled by the SGSN and at the end of the line is the GGSN.

The Iu interface is the air interface between the RNC and the UE. The RNC uses physical channels to transmit information from the core network to the UE. The Iub interface is the messages between a Node B and the RNC. The Node B sends neighbouring cell measurements to the RNC through the Iub while the RNC transmits timing adjustment commands, and transmit and receive schedule to the Node Bs [12]. Such messages will be used in the synchronization of the Node Bs, which will be given in details in Chapter 2. The Iur interface is the interface between two RNCs, which is used in the handover operation for UE. As each RNS is synchronised individually to at least one reference clock (e.g. GPS), which is indicated by the reference Node B, no communication over Iur is necessary for cell synchronisation between RNSs [12].

In the thesis, we assume each Node B controls one cell. Therefore cell and Node B are used interchangeably. Moreover the UE is assumed to be WCDMA handsets.

1.4 LITERATURE REVIEW

1.4.1 REASONS FOR INTER NODE B SYNCHRONIZATION

In Universal Mobile Telecommunication System (UMTS) WCDMA TDD mode, inter-Node B synchronization is necessary for fully enhancing the system capacity of the base stations [12]. The three reasons why synchronization is required in enhancing the system capacity are as follows.

1. Inter-slot interference: Without frame synchronization there could be overlap between the uplink timeslot and the downlink timeslot, due to the potentially close interval between timeslots as shown in Fig 1.3 below.

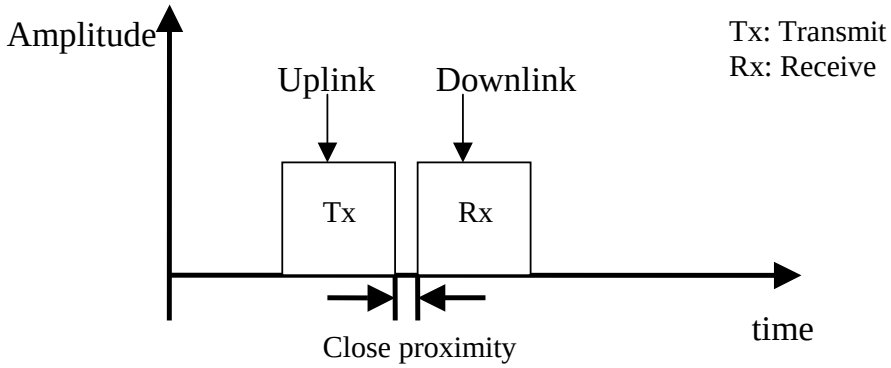


Fig 1.3: Time line for TDD

Power control is applied to UEs operating in WCDMA TDD mode to limit the interference within the system. In the event that power control is not used, all the UEs will transmit signals toward the Node B with the same power, without

taking into account the fading and the distance from the Node B. UEs that are closer to the Node B will cause significant interference to the UEs that are further from the Node B, as the power received by the Node B for UEs nearer to it will be larger than those of the UEs further away. This effect is known as the near-far effect. Since a UE will also affect the Node B of a neigbouring cell, power control is also essential to reduce inter-Node B interference. Thus synchronization is essential as it prevents interslot interference and intercell interference. The measurement of received signal to interference ratio (SIR) shall be carried out periodically at Node B to check on the transmission power of the UEs. This will avoid the use of too much power by the UEs.

2. Neighbouring cell monitoring: In TDD mode, certain measurements such as the signal strength from neighbouring cells have to be measured by the UE [12] for handover purposes. Without cell synchronisation, the UE would have to synchronise itself before being handovered as illustrated in Fig 1.4.

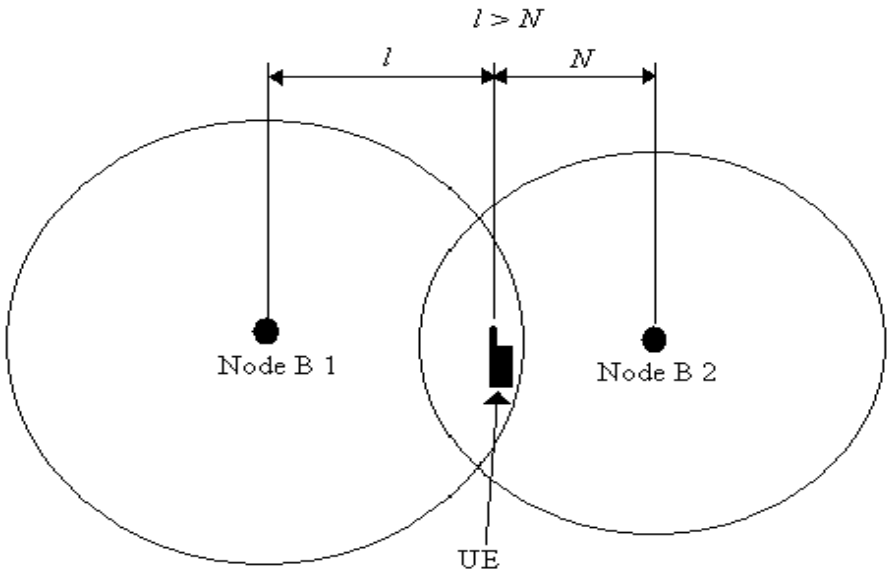


Fig 1.4: Neighboring cell monitored by the user equipment

Assume Node B 1 is serving the UE. The UE measures the strength from neighboring cells in order to check if the signal from neighboring cell is stronger than that from the current serving cell, so that the UE may handover to the neighboring cell, for better performance. The UE will be able to distinguish between the signals of Node B 1 and Node B 2 from the different scrambling codes used by each Node B. The different codes used by each Node B is given in detail in Chapter 2. If Node B 1 and Node B 2 are not synchronized, the UE will have to synchronize with Node B 2 again before being handovered by Node B 1.

3. Handover: In TDD mode “timing advance” is used to align the uplink radio signals from the UE to the Node B. After the Node B has received the reference timing from its RNC, the Node B is able to separate the timeline into frame numbers. The timing advance value is the time difference of the received uplink transmission by the UE in relation to its timeslot structure of the Node B that means in relation to the ideal case where an uplink transmission would have zero propagation delay [13]. This is only applied for large cells. Small cells do not need timing advance [12].

   To illustrate, consider Fig 1.5 where a Node B is serving 2 handphones, UE A and UE B. Assume that this is a large cell.

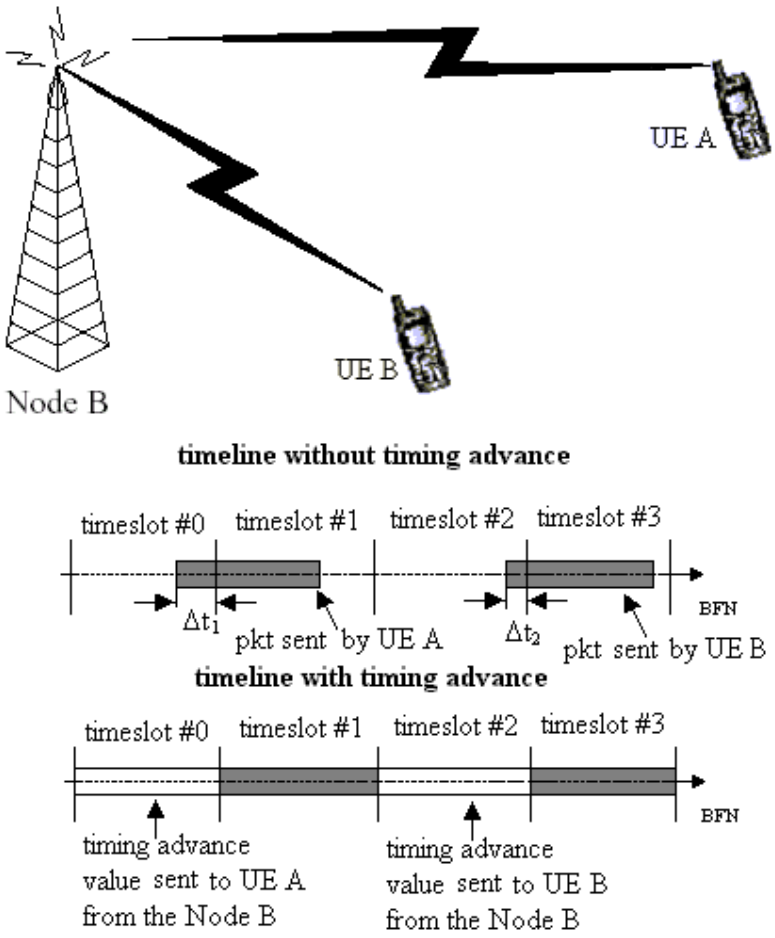


Fig 1.5: Propagation delays for uplink transmission with timing advance

UE A and UE B will initially search the available Node B serving them. The Node B will inform the RNC the UE within its range. The RNC calculates the required timing advance value of the UE and saves it in the RNC for later use when the UE requires a channel, that means when the UE establishes a call [13]. Frequently, the RNC also checks whether an up to date timing deviation measurement is available [13]. If no up to date timing deviation measurement is available, e.g. because of lack of uplink transmissions, the RNC has to trigger an uplink transmission from the UE [13].

Assume time slots with even numbers (timeslot #0, timeslot #2 and so on) are for downlink transmission from the Node B while the time slots with odd numbers (timeslot #1, timeslot #3 and so on) are for uplink transmission to the Node B. When a call is established between both UE A and UE B and the Node B, assuming Node B assigned timeslot #1 and timeslot #3 to UE A and UE B respectively, and if the propagation delays for the uplink signals from UE A and UE B to the Node B are Δt_1 and Δt_2, respectively, from the starting point of timeslot #1 and timeslot #3. Then the Node B will transmit the timing advance value (Δt_1 and Δt_2) for each UE via the downlink before each UE starts sending the uplink signal. See Fig 1.5.

If UE C is handovered from another Node B (Node B 2) to the Node B (Node B 1) serving both UE A and UE B, UE C will be provided with the information about the synchronization accuracy of this new cell so that it can apply the proper timing advance value in the new cell [13].

In cells with low synchronization accuracy, handover is done without timing advance, so that the maximum timing error during transmission would be twice the propagation delay, e.g 6 us or 24 chips for 1 km radius [12]. In cells which are considered to be medium synchronization accuracy, the timing advance value for handover is calculated by the RNC [12]. The correction value is subsequently calculated in the RNC and signalled to the UE before handover execution. The timing advance step size is 4 chips [12]. The maximum timing error during transmission would then be 2 μs or 8 chips for 1 km radius [12]. In cells with high synchronization accuracy, the same timing advance value in the UE will be used

after handover [12]. The maximum timing inaccuracy during transmission will be 400 ns.

Hence if the Node Bs are synchronized, resources in the Node B is saved as the timing advance values need not be calculated again.

1.4.2 CELL SYNCHRONIZATION BURST

It has been proposed by 3GPP that cell synchronization burst are used for inter-node B synchronization. This cell synchronization burst can be any orthogonal code that possesses good overall autocorrelation properties especially for small shifts around the main peak [12]. The autocorrelation of a code will determine how easily the code can be detected in a noisy environment. Moreover if the autocorrelation exhibits zero secondary peaks for a few chips next to the main peak, the possibility of a perfect autocorrelation window of adjustable size around the main autocorrelation peak can be used during simple correlation [12]. The purpose of this property is to prevent self-interference arising from mutipaths. This is given in greater details in Appendix A. Although the autocorrelation of a synchronization code is important, the cross correlation is also another important aspect of the code. This is to prevent interference to other Node Bs working within the same environment if the receiver of a Node B in an RNC locked onto the cross correlation peaks resulting from correlating with the synchronization code of its adjacent Node B, which is controlled by another RNC. Hence the transmission of an erroneous signal to the RNC even though there is no transmission of synchronization codes within that RNC. Hence not only is the autocorrelation important but also its cross correlation has to be taken into account.

Another restriction on the synchronization code is that according to the specification by 3GPP, each code should not have a length exceeding 2400 chips [14]. The structure of the synchronization burst is shown in Fig 1.6.

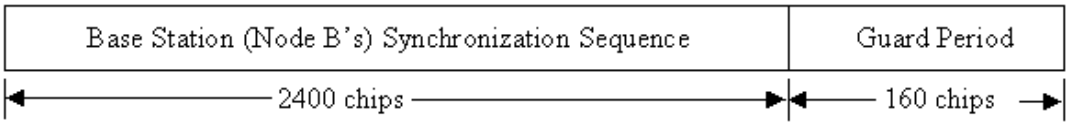


Fig 1.6: Synchronization Burst Structure

Thus a synchronization code used for this synchronization purpose must not only have good autocorrelation and cross correlation properties, it must not have a length exceeding 2400 chips.

It has been proposed by 3GPP that Concatenated Extended Complementary (CEC) Sequences with multiple offsets of nearly similar length [12], [15] are to be used by the radio network controller (RNC) to synchronize all the Node Bs under its control. In the Chapter 4 of this thesis, we show that although CEC sequences with multiple code offsets [12], [15] have many attractive properties, the cross correlation property of such sequences is very poor, as it exhibits sharp spikes.

1.5 CONTRIBUTION TO THIS THESIS

My contribution to the thesis involves doing a comprehensive literature review on the synchronization of Node Bs operating in WCDMA TDD mode and put together a detailed explanation in Chapter 2.

Another contribution to this thesis is the simulation of the aperiodic correlation properties of the CEC sequences with multiple code offsets, that have not been investigated previously.

The last contribution to this thesis is the development of a new type of sequence formed by concatenating two *m*-sequences. The aperiodic correlation properties of the concatenated *m*-sequences have been computed and shown. Moreover the cross correlation bound is also provided and the simulated and theoretical bounds are compared.

1.6 OVERVIEW OF THE THESIS

In the thesis, we focus on developing new codes for synchronization purposes. We consider using concatenated codes for this purposes. Another contribution to this thesis is the analysis of why CEC sequences with multiple code offsets have poor cross correlation properties.

The thesis is divided into 6 chapters and organized as follows. Chapter 1 gives a brief overview of the WCDMA network architecture. The motivation for the project is given. Moreover an outline of the thesis is given.

Chapter 2 gives a description on the major differences between the physical channel of WCDMA FDD and TDD mode. A detailed description of how Node Bs operating in the WCDMA TDD mode of communications are synchronized is also provided.

This Chapter will also provide details of how a WCDMA TDD mode handset or UE perform a cell search and acquire the synchronization of the cell it is in.

In Chapter 3, we provide details of some orthogonal codes such as Maximal Length (ML) sequences and Gold sequences, which will be used in Chapter 5 of the thesis. The properties of Golay complementary sequences, which are used to form the CEC sequences with multiple code offsets, are also provided.

The simulations for the cyclic autocorrelation and cross correlation of CEC sequences with multiple code offsets are given in Chapter 4. An analysis of why such sequences have poor cross correlation properties is provided.

In Chapter 5, a construction principle of a new form of concatenated sequences formed by concatenating ML sequences and Gold sequences is given. Simulations of aperiodic autocorrelation and cross correlation for the new concatenated sequence are also presented.

Finally, conclusions are drawn from the simulations done for both the aperiodic autocorrelation and cross correlation for both the CEC sequences with multiple code offsets and the new concatenated sequences, and suggestions for future work are given in Chapter 6.

CHAPTER 2

BACKGROUND

2.1 COMPARISON BETWEEN FDD AND TDD MODE OF WCDMA

In FDD, we differentiate the uplink and downlink by different frequency bands while in TDD, we differentiate the uplink and downlink by different timeslots. In both FDD and TDD, most of their physical channels typically consist of a three-layer structure of superframes, radio frames, and time slots [16], [17]. See Fig 2.1.

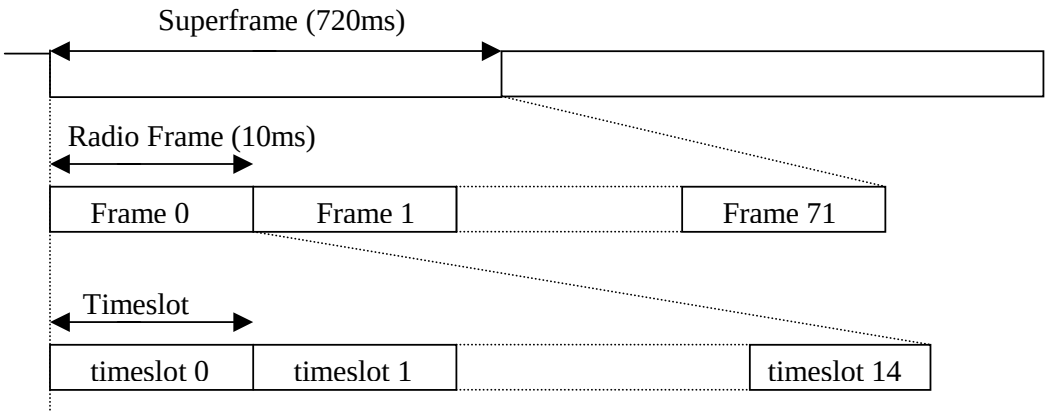


Fig. 2.1: FDD and TDD Physical Channel Signal Format

2.1.1 FDD

In FDD, depending on the symbol rate of the physical channel, the configuration of radio frames or time slots varies. A superframe has a duration of 720ms and consists of 72 radio frames. A radio frame is a processing unit which consists of 15 timeslots [16]. The number of bits per time slot depends on the physical channel [16].

In uplink transmission for FDD, there are two types of uplink dedicated physical channels, the uplink Dedicated Physical Data Channel (uplink DPDCH) and the uplink Dedicated Physical Control Channel (uplink DPCCH) [16]. The uplink DPDCH is used to carry data while the uplink DPCCH is used to carry control information, as shown in Fig. 2.2.

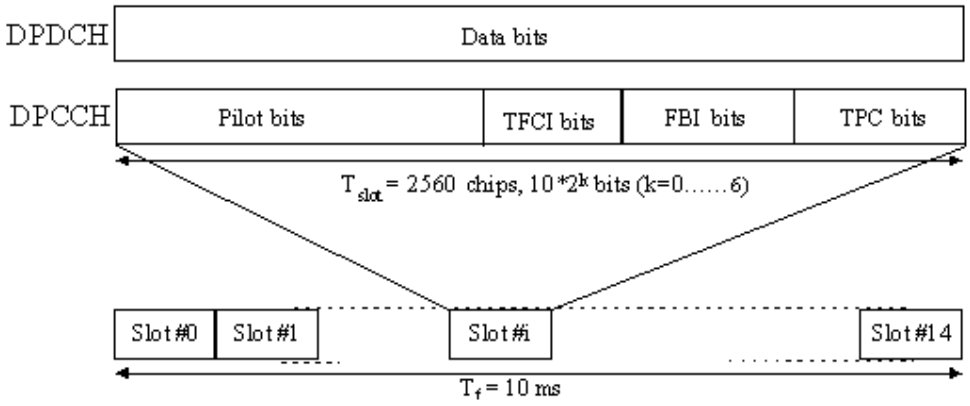


Fig 2.2: Uplink Dedicated Physical Channel

The DPCCH consists of pilot bits, transmit power-control (TPC) commands, feedback information (FBI), and an optional transport-format combination indicator (TFCI). The TFCI is used to inform the receiver about the instantaneous parameters of the different transport channels on the uplink DPDCH, and corresponds to the data

transmitted in the same frame. The parameter k in Fig. 2.2 determines the number of bits per uplink DPDCH/DPCCH slot. It is related to the spreading factor (SF) of the physical channel as $SF = 256/2^k$. The DPDCH spreading factor may range from 256 down to 4. The number of data bits (N_{data}) varies from 10 bits to 640 bits or $10 * 2^k$ (where k= 0……6) and it will be spread to have an eventual length of 2560 chips.

However there is only one type of Downlink Dedicated Physical Channel (downlink DPCH). The dedicated transport channel (DCH) is transmitted in time-multiplex with control information (pilot bits, TPC commands, and an optional TFCI). The downlink DPCH can thus be seen as a time multiplex of a downlink DPDCH and a downlink DPCCH, as compared to the uplink transmission [16], as shown in Fig. 2.3.

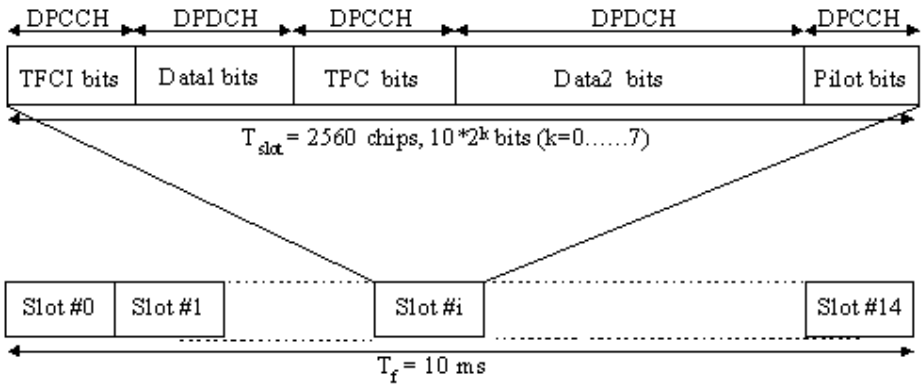


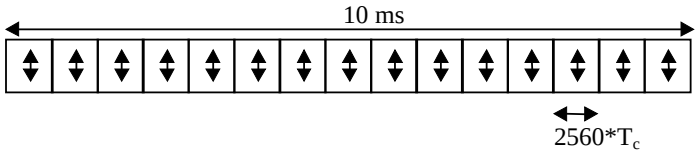
Fig 2.3: Downlink Dedicated Physical Channel

Similar to the uplink dedicated physical channel, the parameter k in Fig. 2.3 determines the total number of bits per downlink DPCH slot. It is related to the spreading factor SF of the physical channel as $SF = 512/2^k$. The spreading factor may thus range from 512 down to 4.

2.1.2 TDD

In TDD, all physical channels also take three-layer structure of superframes, radio frames, and time slots [17]. The duration of a radio frame is 10ms and the duration of a time slot is 2560 * T_C, where T_C is the chip duration. Every time slot have one guard period to prevent interference between neighbouring timeslots. The time slots are used in the sense of a Time Division Multiple Access (TDMA) component to separate different user signals in the time and the code domain.

Each 10 ms frame consists of 15 timeslots and each timeslots are allocated to either the uplink or the downlink [17]. See Fig 2.4.



.

Fig 2.4: TDD frame structure

As such, the TDD mode can adapt to different environments and deployment scenarios [17]. In any configuration at least one timeslot has to be allocated for the downlink and at least one time slot has to be allocated for the uplink [17]. Examples for multiple and single switching point configurations as well as for symmetric and asymmetric uplink/downlink allocations are shown in Fig 2.5 [17].

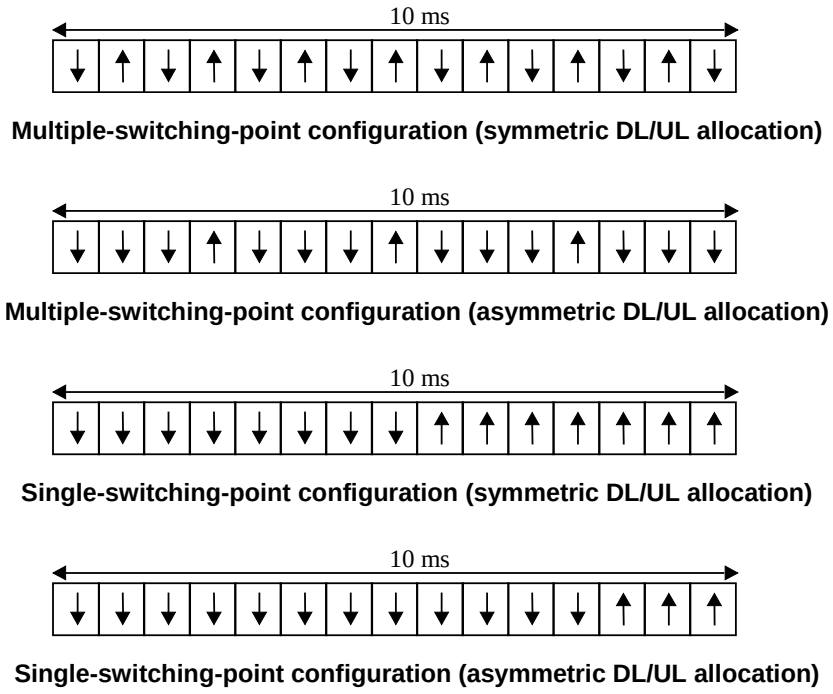


Fig 2.5: TDD frame structure examples

Each timeslot is also known as a burst and there are two types of burst namely burst type 1 and the burst type 2. Both consist of two data symbol fields, a midamble and a guard period. The bursts type 1 has a longer midamble of 512 chips than the burst type 2 with a midamble of 256 chips [17]. Fig 2.6a and Fig 2.6b shows the structure of burst type 1 and burst type 2 respectively [17].

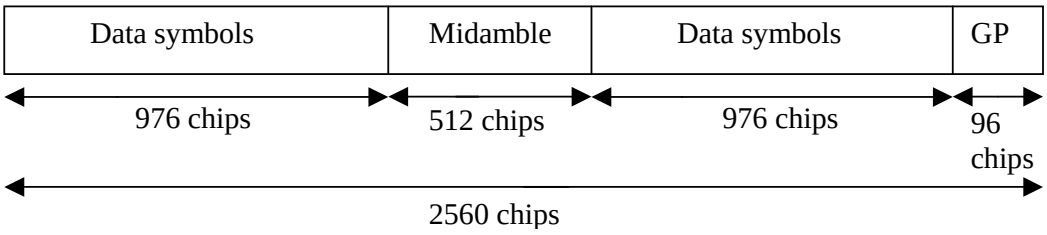


Fig 2.6a: Structure of Burst Type 1

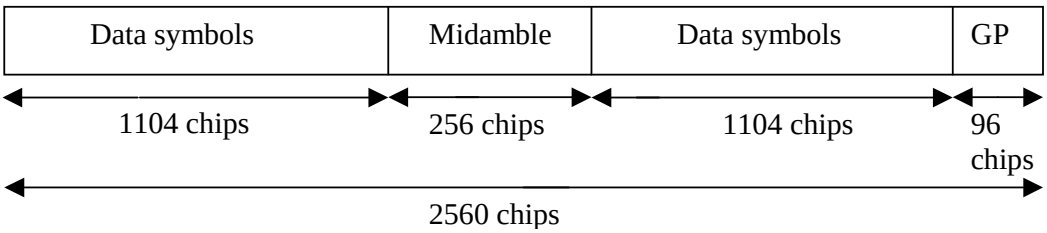


Fig 2.6b: Structure of Burst Type 2

Each data symbol in Fig 2.6a and Fig 2.6b are spread by a channelisation code. The resulting sequence is then scrambled by a sequence of length 16 [18].

2.1.3 ADVANTAGES & DISADVANTAGES OF FDD & TDD

FDD and TDD are the two most prevalent duplexing schemes used in wireless communications.

In FDD, one frequency channel is allocated for uplink and another frequency channel is allocated for downlink. This allows simultaneous transmission in both directions. Moreover, FDD systems utilize channel plans that comprise frequencies with equal bandwidth. Since each channel has a fixed bandwidth, the capacity of each channel is also fixed and equal to that of all other channels in the frequency band. This makes FDD ideal for symmetrical communication applications, such as voice communications [8]. However in any FDD channel plan, a guardband is always maintained to avoid self-interference [8], [9]. Since this guardband is unused, part of the spectrum is wasted [8], [9]. Moreover, since signal transmission and reception occur at the same time, FDD radio units require duplexers, along with Radio

Frequency shielding to separate the incoming and outgoing signals at the antenna in order to mitigate self-interference [8], [9]. This increases the complexity of the radio hardware, which in turn increases in costs.

Although voice transmission is probably symmetric in its data transmission, several other data applications, such as Internet connections or broadcast data (e.g., streaming video), sends more data from the server to the mobile device (i.e. downlink) [8], [10]. FDD can be used for such asymmetric traffic but in order to be spectrally efficient, the downstream and upstream channel bandwidth must be matched precisely to the asymmetry [8]. Since Internet data is bursty by nature and the symmetric is always changing, the channel bandwidth cannot be set in FDD, hence another form of duplexing is required.

TDD uses a single frequency with different time slots to transmit signals in both uplink and downlink. However, a guard time between uplink and downlink is required so that both the base station and mobile equipment can switch from transmit to receive mode [8], [9], [10]. Therefore TDD requires stricter time synchronization between the users than FDD, because of its time division nature; the base station cannot be allowed to transmit at the same time as the mobile stations. In TDD, as different number of timeslots can be used for uplink and downlink, TDD is more suitable for asymmetric traffic [8], [11]. Moreover, TDD terminals do not need duplexer and hence lesser hardware complexity than FDD terminals, which in turn brings down the costs of TDD equipment.

Another advantage of TDD over FDD is frequency planning of communication systems using these duplexing schemes. In many countries, block frequency allocations are made to service providers to provide maximum flexibility in deploying services. In many cases, these allocations are made without regard for the spacing required between transmit and receive frequencies for FDD equipment [8]. Thus, the frequencies allocated for broadband services can become fragmented and non-uniform when FDD rules are applied when traffic data increase [8]. Without careful planning, spectrum can be wasted [8]. On the other hand, TDD systems offer a tremendous advantage over FDD since they can be deployed with as little as one channel of available TDD spectrum. This eliminates the paired channel allocations required by FDD and allows operators to take full advantage of spectrum allocations that are contiguous, non-contiguous, or small [8]. A table summarizing the advantages and disadvantages between FDD and TDD is shown in Table 2.1.

Mode	Advantage	Disadvantage
FDD	Can transmit and receive at the same time due to different frequency being used.	A duplexer is required due to the different frequency used. Increase in hardware complexity and costs Does not allocate and utilize available bandwidth efficiently
TDD	Allocate and utilize available bandwidth efficiently as more time slots can be allocated to heavier traffic. More spectrum efficient No need for duplexers and thus reduce costs More Flexible in reconfiguring the allocated uplink and downlink data in response to customer needs	Require better time synchronization due to the closeness of time slots. This result in guard period which reduces bandwidth efficiency.

Table 2.1: Table comparing the advantages and disadvantages of FDD and TDD

Hence TDD is a more desirable duplexing technology that allows service providers to receive most from their investment in spectrum and telecom equipment, while meeting the needs of individual customers [8]. Therefore much research is done on TDD.

2.2 TDD INTER NODE B SYNCHRONIZATION

In Fig 1.2, we have shown a block diagram of the architecture of the WCDMA system. By Node B Synchronization, it is generally meant that a common timing reference among different Node Bs [13] within an RNS has been achieved. So how are Node Bs in the TDD mode of communication synchronized? The easiest and fastest way for a Node B to be synchronized is through the use of the Global Positioning System (GPS). However 3GPP hopes to achieve a totally independent and self-reliant system. In addition, GPS receivers such as the C/A-code (Coarse/Acquisition-code) pseudorange receivers [19] and the Precision code receivers [19] only works when they have good visibility with the space borne satellites, hence, the receiver will not operate accurately under densely urban and indoor environments. This will increase the deployment cost. Thus 3GPP proposed synchronizing the Node Bs by transmission of synchronization codes from these Node Bs through the air [13], [20].

Referring Fig 1.2, there is one Node B which is synchronized by an external timing reference (e.g. GPS) within each RNS. The RNC has the control of the whole algorithm for synchronization [20]. It will initialize, establish and maintain a connectivity plan which is used to establish where each Node B are [20]. The RNC shall collect measurements and compute adjustment. It may estimate the

synchronization accuracy between cells and signal the relevant information such as the timing advance values to the UEs for handover purposes [20].

For cell synchronization via air interface, there are mainly 3 different synchronization stages [12], [13], [20]. The first and so known “initial synchronization” stage is when a newly established TDD network is started to be synchronized [12], [13], [20]. It is assumed that none of the base stations are supporting any traffic during this stage. The next stage is the “steady state” phase denoting synchronization procedure during normal operation of the TDD network. The third stage is known as “late entrant cell” stage where new base stations achieve synchronization with the synchronized RNS.

Before describing inter Node B synchronization in WCDMA TDD mode, some parameters are defined first.

2.2.1 IMPORTANT PARAMETERS

BFN Node B Frame Number counter. This is the Node B common frame number counter. Range: 0 to 4095 frames.

RFN RNC Frame Number counter. This is the RNC node common frame number counter. Range: 0 to 4095 frames.

SFN Cell System Frame Number counter. In TDD, SFN is locked to the BFN (i.e. SFN mod 256 = BFN mod 256). Range: 0 to 4095 frames.

2.2.2 INITIAL SYNCHRONIZATION

This stage covers 3 phases namely the Preliminary, Frequency Acquisition and Initial synchronization phase [12], [13], [20].

A. Preliminary Phase

A time diagram of this preliminary phase is shown in Fig 2.7.

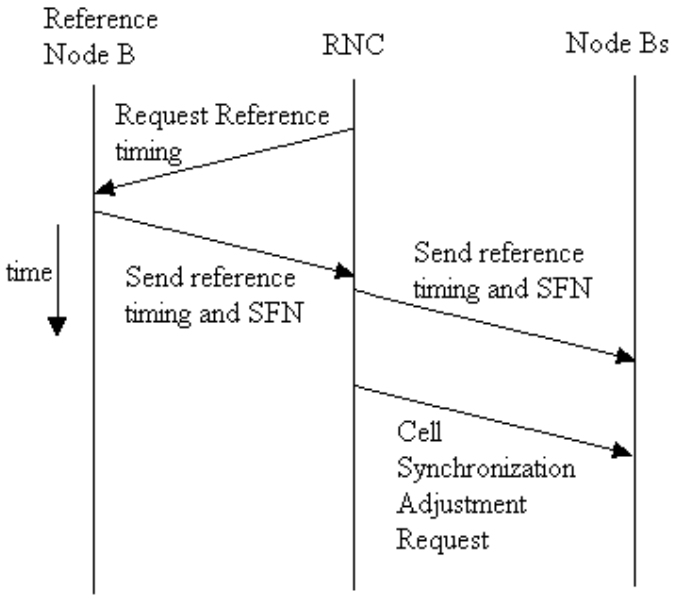


Fig 2.7: Time diagram for the preliminary phase

In the RNS, there should be one Node B or cell, which is synchronized by an external timing reference (e.g. GPS). The cell with the reference timing will initialize their Cell System Frame Number (SFN) counter so that the frame with SFN = 0 starts on January 6, 1980 at 00:00:00 Greenwich Meridian Time (GMT) [13]. Using the GMT from the GPS, this Node B will adjust its timing to suit its region. For example, the actual timing in Singapore is GMT + 08:00:00 while in Brisbane and Sydney, it would be GMT + 10:00:00. In this way, this reference Node B will have the absolute

time of the region and since SFN = 0 starts on January 6, 1980 at 00:00:00, it will be able to set the current SFN for the current timing and date.

This Node B will add the reference clock availability within the RESOURCE STATUS INDICATION message that is sent to the RNC, which enables the RNC to know which Node B has the reference timing [13], [20]. During the start of the synchronization, the RNC will retrieve the reference timing signal from the cell with the external timing reference via the Iub. The reference cell will also indicate the SFN to the RNC. The SFN ranges for from 0 to 4095 frames [13]. However there is a delay when the RNC retrieve the reference timing. Thus the RNC finds the round trip delay as follows. The RNC will send a downlink Node Synchronization control frame to the Node Bs containing the parameter T1, which contains the RNC specific frame number (RFN) that indicates the time when RNC sends the downlink Node Synchronization control frame. See Fig 2.8.

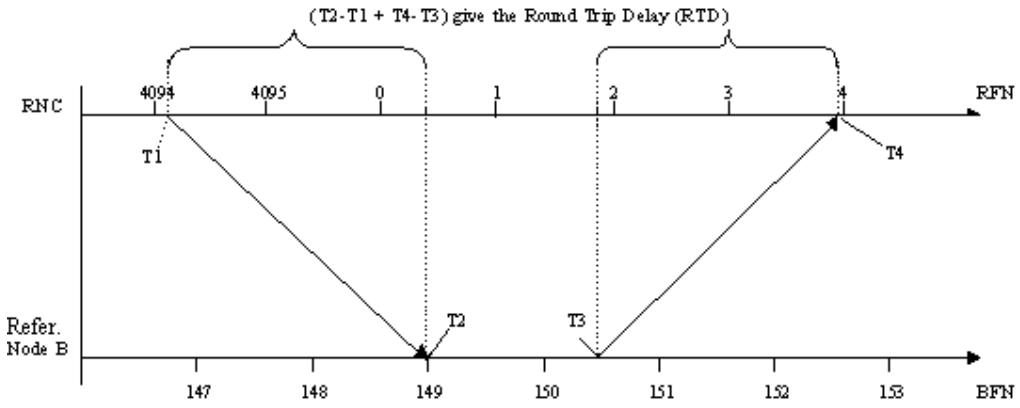


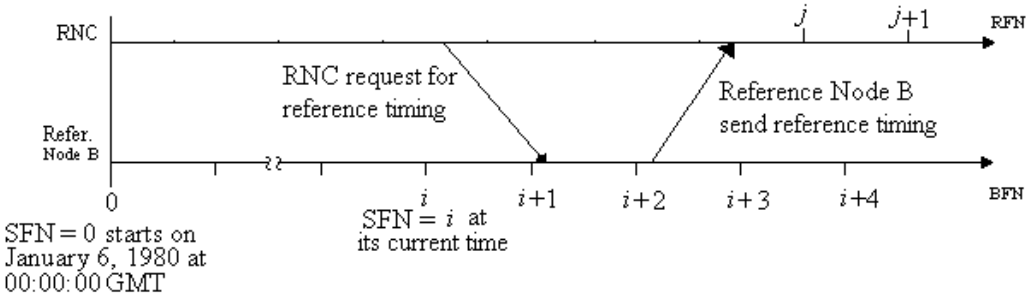
Fig 2.8: RNC and Node B round trip delay

Upon reception of a downlink Synchronization control frame, the Node B shall respond with uplink Synchronization Control Frame, indicating the parameter T2 and

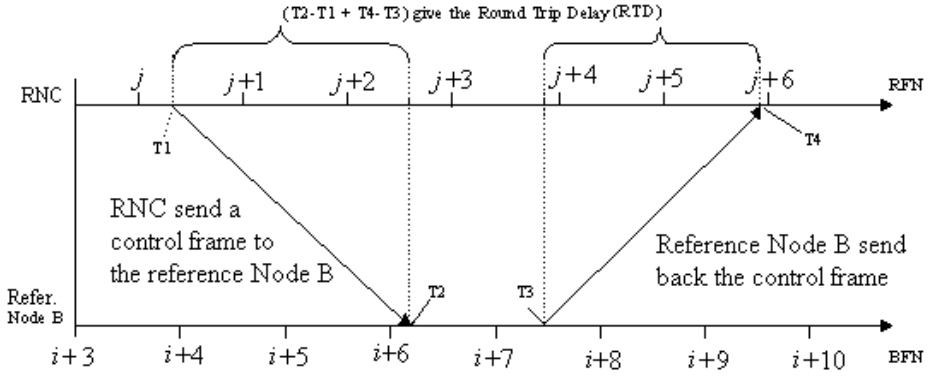
T3, as well as T1 which was indicated in the initiating downlink Node Synchronization control frame [13]. T2 contains the Node B specific frame number (BFN) that indicates the time when Node B receives the correspondent downlink Node Synchronization control frame while T3 contains the BFN that indicates the time when Node B sends the uplink Node Synchronization control frame [13]. Then with the parameter T4 which contains the RFN that indicates the time when RNC receives the uplink Node Synchronization control frame, the RNC is able to find the delay from it to the reference Node B by computing T2 – T1 + T4 – T3. Knowing the delay, the RNC is able to make the necessary adjustment to its absolute time. See Fig 2.9 for the overall process of this part of the initial synchronization phase. It is from Step 1 to Step 3.

Now the RNC proceeds by sending the timing and the SFN of all the remaining Node Bs in the RNS, instructing them to adjust their clocks. The RNC then sends a CELL SYNCHRONISATION ADJUSTMENT REQUEST message to all the cells or Node Bs for SFN update, apart from the one containing the reference clock [14], [19]. The cell shall adjust their SFN and frame timing accordingly. After the Node Bs within the RNS obtain the reference timing, the Node Bs begin to set their own reference timing such as when the first timeslot starts. See Fig 2.10. Fig 2.10 shows the timeline between an RNC and two Node Bs under its control after both Node B obtain the reference timing from the RNC. Each timeslot has a duration of 2560 * T_C, where T_C is the chip duration.

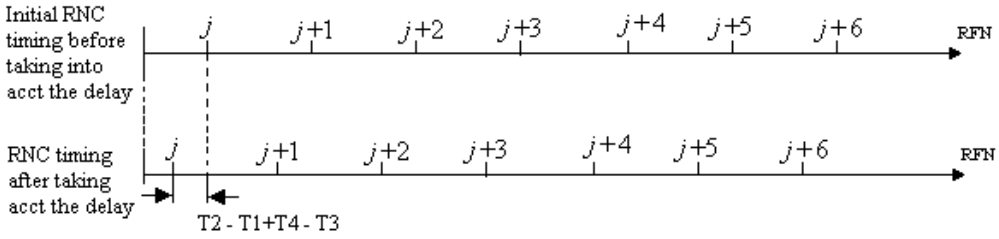
Step 1: RNC retrieve reference timing from reference Node B



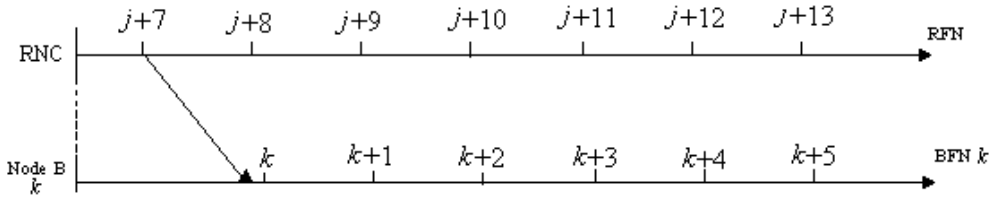
Step 2: After the RNC receives the absolute timing and adjust its RFN and time
The RNC finds the round trip delay from the RNC to the reference Node B



Step 3: After the RNC obtain the round trip delay, the RNC shifts its timing backward by (T2 - T1 + T4 - T3) amount to take into account the delay involved when the RNC obtain the reference timing



Step 4: **After the RNC adjust its reference timing obtained from the reference Node B, the RNC continues to send the reference timing to the other Node Bs under its control by the Iub**



Step 5: **The RNC finds the round trip delay between itself and the individual NodeBs under its control as in step 2 and update each individual Node B according to their round trip delay**

Fig 2.9: Timeline showing how the RNC obtain the reference timing

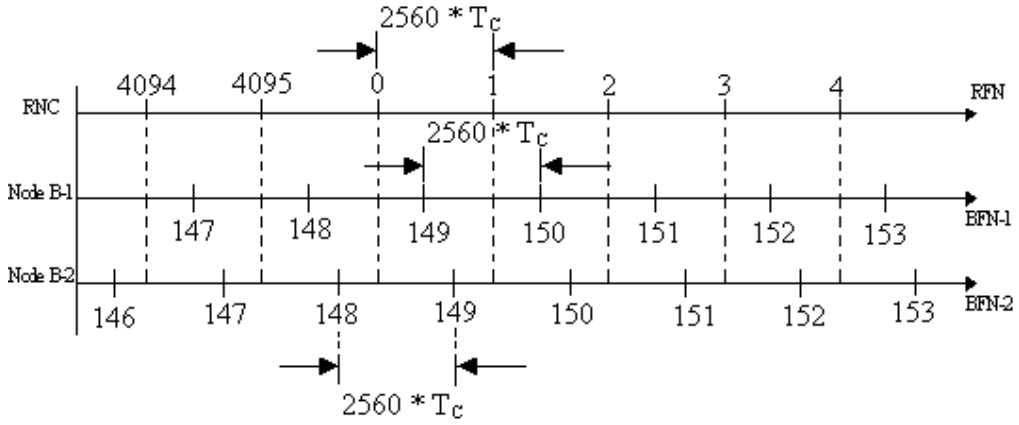


Fig 2.10: Timeline showing the RNC and two Node Bs after obtaining the reference time

Note that the RFN, the BFN for Node B1 and the BFN for Node B 2 are not phase aligned. This is to reduce the transmission delay between the RNC and the Node Bs. However the timing difference between the RNC and the Node Bs must be known [13].

B. Carrier Frequency Acquisition Phase

The time diagram for this phase is shown in Fig 2.11.

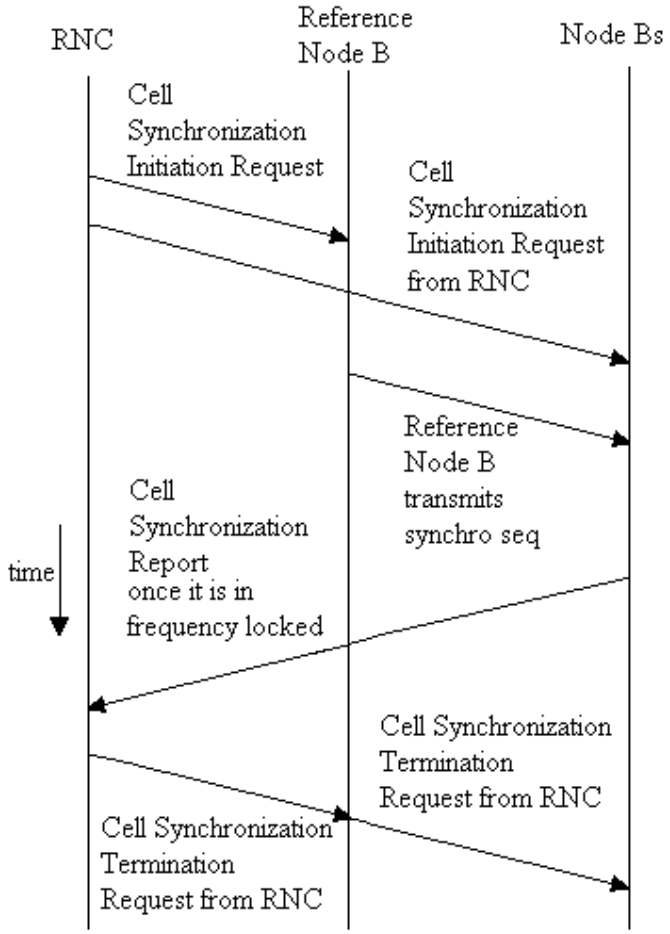


Fig 2.11: Time diagram for the Frequency Acquisition Phase

After obtaining the reference timing, the Node Bs must frequency lock onto the carrier frequency of the WCDMA system. This phase is used to bring cells within an RNS area to within frequency limits prior to the initial synchronization phase. In order to do this the Node B with the reference timing will transmit cell synchronization burst continuously in every time slot where possible [13], [20]. The RNC will

transmit CELL SYNCHRONIZATION INITIATION REQUEST message, which contains the synchronization sequence to transmit and the power of transmission.

Initially, all cells are considered not in frequency locked and will listen for the transmission. The RNC will send a CELL SYNCHRONIZATION INITIATION REQUEST message to the rest of the Node Bs except the one with the reference timing to set the parameters such as the synchronization sequence being used. A cell will then try to lock onto the frequency of the signal by using an external circuit such as the phase locked loop (PLL).

Once a cell completes this frequency acquisition, fulfilling the frequency stability requirement of ±0.05 parts per millions (ppm) observed over a period of one timeslot [21], the cell will signal to the RNC a CELL SYNCHRONIZATION REPORT message [13], [20].

This frequency acquisition phase will continue until the RNC has received completion of frequency acquisition signals from all the cells. After this is done, the RNC will send a CELL SYNCHRONIZATION TERMINATION REQUEST message to each cell to stop all transmissions [13].

C. INITIAL SYNCHRONIZATION PHASE

This final phase of the “Initial Synchronization” stage is used to bring cells in the RNS area into fine synchronization before network startup [13]. Up till this phase, the cells are not supporting any traffic yet. This stage is to bring the Node B to start from

the same timeslot. See Fig 2.12 where the timeline of the RNC and two Node Bs before and after this initial synchronization.

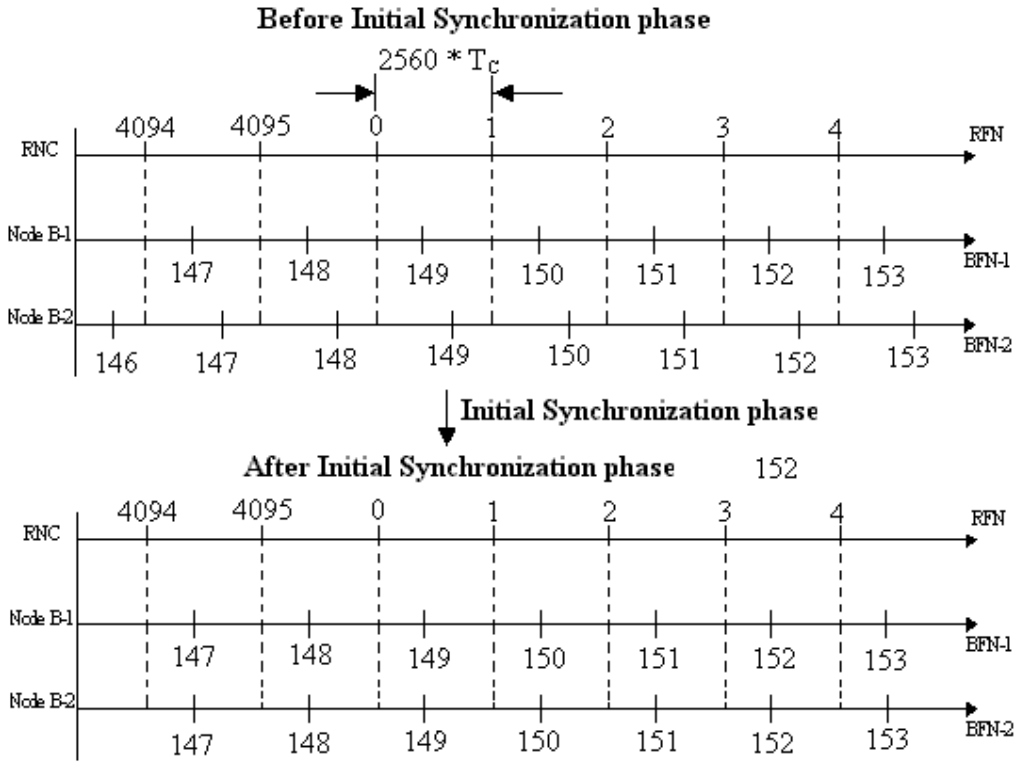


Fig 2.12: Timeline showing the timing before and after initial synchronization phase

The time diagram for the initial synchronization phase is shown in Fig 2.13.

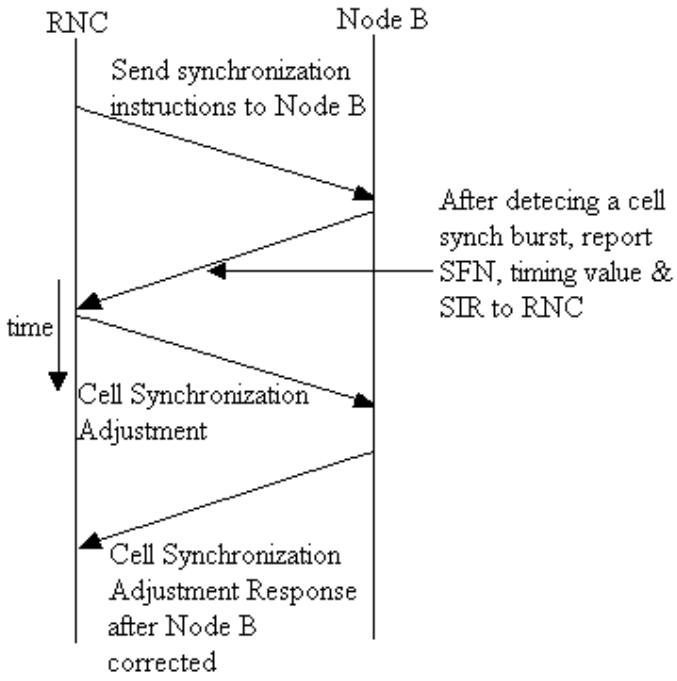


Fig 2.13: Time Diagram for the Initial synchronization phase

In this phase, all the Node Bs will be instructed to transmit the cell synchronization burst one after another. The RNC will send a CELL SYNCHRONIZATION BURST INSTRUCTION message to all the Node B within the RNS. The message will indicate the start of the SFN, the timeslot for the transmission, the cell synchronization burst code and the transmission power [13], [20]. As mentioned earlier, the cell burst code takes up one time slot.

After receiving the message, each individual Node B will only transmit at the particular SFN given and for the rest of the period, the Node B will “listen” from other Node B within its own RNS. So only one Node B transmits while the rest “listen”. This goes on until all the Node B within the RNS has transmitted at least once.

When a Node B detects a cell synchronization burst from another Node B, it will take note of the SFN where the cell synchronization burst is received, the time of arrival of the cell synchronization burst timing and the cell synchronization burst signal to interference ratio (SIR) [13], [20]. Since the cell knows the transmission power of the synchronization burst, it is able to calculate the SIR. The cell will then transmit the information to the RNC via the Iub.

The RNC will establish a “connectivity matrix” for each Node B [12]. For instance if an RNC has 3 Node Bs namely α, β and γ and α is instructed to transmit first followed by β and γ respectively, a “connectivity matrix” for α, β and γ that is established by the RNC will be as follows. See Fig 2.14

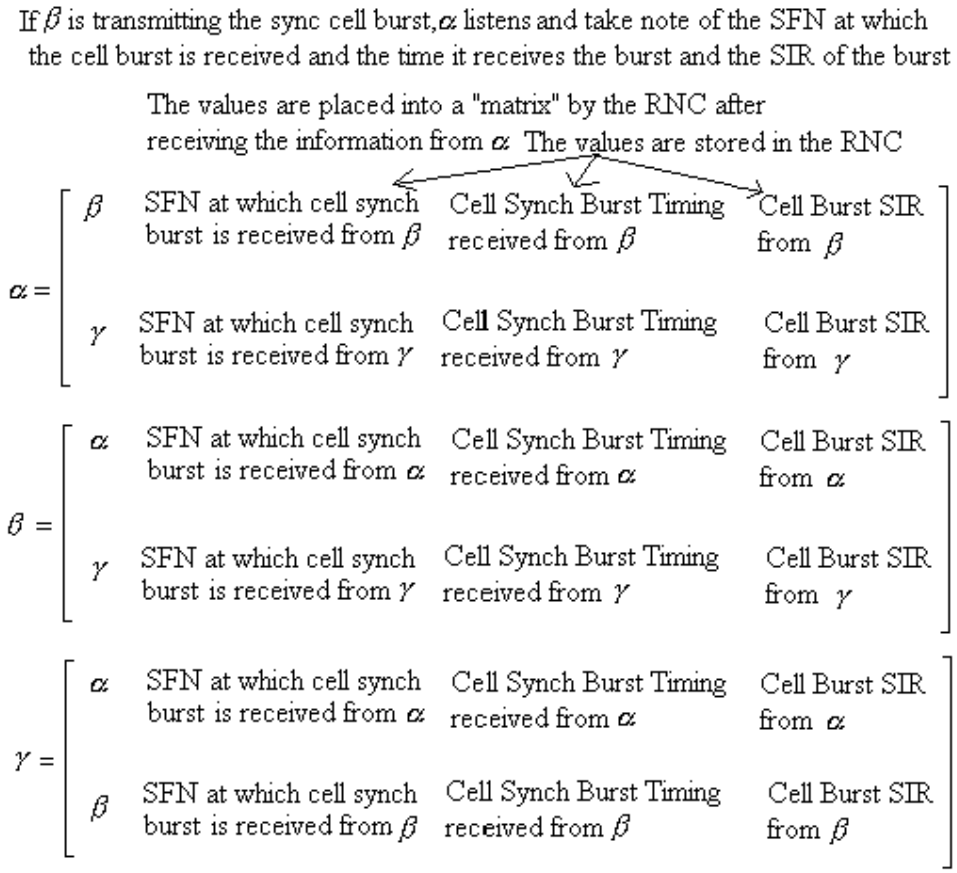


Fig 2.14: “Connectivity Matrix” for Initial Synchronization phase

In Fig 2.14, α will receive the cell synchronization burst from β and γ. Upon the reception of the cell synchronization burst from β, α will take note of which SFN did it receive the cell synchronization burst and the time it receive the burst. Moreover α will also take note of the SIR of the burst. Subsequently, α will send the information to its RNC by the Iub. Similar procedures will apply to both β and γ. With the information, the RNC is able to come out with the matrix in Fig 2.14. The RNC is able to establish this matrix because it knows the schedule of the transmission of each individual Node B. That means the RNC knows which Node B is supposed to send and which are supposed to listen. Using the cell burst SIR, the RNC is also able to plan a re-use frequency pattern [12]. This is performed as follows [12].

1 A matrix of minimal connectivity is computed where pairs of cells are labelled as "minimal neighbours" if either of the SIR exceeds a threshold
2 The set of cells is divided into partitions where cells which are in the same partition are sufficiently separated and can therefore be allowed to send the same cell synchronization burst at the same time. For this purpose each partition must satisfy the requirement that no pairs of cells within that partition are minimally connected.

With the measurements obtained and with the knowledge of the time of sending the cell synchronization burst, the RNC is able to make the necessary correction such as the timing and the transmission power of the Node B. To illustrate the calculation made at the RNC, we look at the connectivity matrix for α from β. Upon receiving the time of arrival of the synchronization code from α, the RNC will know whether the synchronization code arrives at α late as the RNC knows the actual timing when β

should start sending. This allows the RNC to know if there is any timing delay in the transmission time of β. The RNC will subsequently calculate the timing difference between the time the synchronization code arrives and the ideal time the code should arrive and inform β via the Iub to make the necessary correction in its timing.

Hence after the “connectivity matrix” is established, the RNC will be able to calculate a set of timing updates if there is a timing error in its Node Bs. The adjustment is known as the SYNCHRONIZATION ADJUSTMENT [20]. For instance if the RNC detects an error at the SFN used by a Node B through the knowledge of SFN received by the other Node Bs, the RNC will do a Frame Adjustment value by informing the Node B the required correction of the SFN.

If the timing of the received cell synchronization burst is incorrect, the RNC will instruct the transmitting Node B the required time adjustment to correct the transmission time [20]. The received power could also be corrected if the received SIR is low indicating insufficient power is used during the transmission [20].

After the RNC has made the necessary correction, the RNC will transmit a CELL SYNCHRONIZATION ADJUSTMENT message to the affected Node Bs. Upon reception of a CELL SYNCHRONIZATION ADJUSTMENT message, the Node B will adjust its timing accordingly [13], [20]. The timing adjustment must be completed before the CELL SYNCHRONIZATION ADJUSTMENT RESPONSE message is sent back to the RNC, indicating the acknowledgement that the correction has been made [13], [20].

This phase is repeated several times (typically 10 times) to bring the network into tight synchronization [20].

2.2.3 STEADY-STATE PHASE

The time diagram for the steady state phase is given in Fig 2.15.

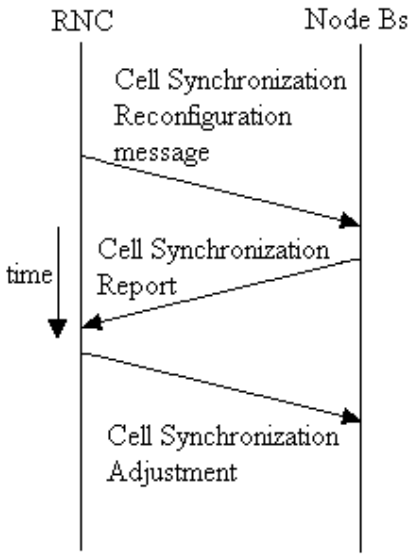


Fig 2.15: Time Diagram for Steady-State Phase

With the start of the steady-state phase, traffic is supported in a cell [13], [20]. The steady state phase starts with each Node B having a cell burst plan, which defines the synchronization schedule [20]. This plan indicates when to transmit a cell synchronization burst and when the individual cell synchronization bursts from the neighbouring cells will be measured. This schedule is important as the cells are now supporting traffic unlike the “Initial Synchronization” stage, so that not all the resources are taken up. The cell burst plan is transmitted to all the Node Bs by sending a CELL SYNCHRONIZATION RECONFIGURATION REQUEST message by the RNC [13], [20].

Now the whole SFN period (0 to 4095 frames) is subdivided into several ‘synchronization cycles’ of equal length and in each synchronization cycle, there are N_{slot} synchronization slots [20]. Therefore a burst plan is required by each Node B to instruct it when to detect the synchronization codes from other Node Bs, without taking up too much resources for this phase, which continues indefinitely [13], [20].
In the cell burst plan, each Node B gets timing plan which defines when they should start transmitting cell synchronization bursts and when cell synchronization bursts should be received. Upon reception of the cell synchronization burst by a Node B, it take note of the timing it receives the cell synchronization burst and the cell synchronization burst SIR [13], [20]. Subsequently, the Node B conveys the information to the RNC by means of a CELL SYNCHRONIZATION REPORT message.

Upon receiving the message from the Node Bs, the RNC will calculate the difference of the reception time of the respective cell synchronization burst from the expected "ideal" time, i.e. the burst would be received when the cells would be perfectly synchronized [19]. If the received power is insufficient, the RNC will also change the transmission power of the particular Node B. After doing the necessary changes, the RNC adjusts the timing of the cell by means of the CELL SYNCHRONIZATION ADJUSTMENT message [15], [20]. The timing and power adjustments will be completed by the next synchronization slot [20].

2.2.4 LATE ENTRANT CELLS

A time diagram of the “late entrant cell” stage is shown in Fig 2.16.

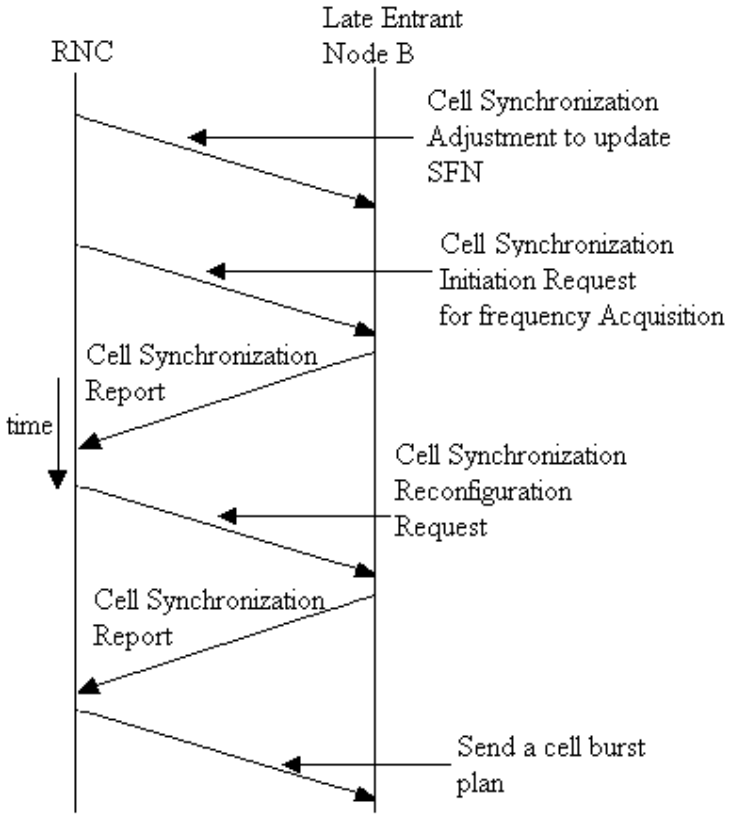


Fig 2.16: Time Diagram of the Late Entrant cell

This stage is used for introducing unsynchronized Node Bs into a synchronized RNS. Late entrant Node B (new Node Bs being added without reference clock) or Node Bs recovering from unavailability initially, will be roughly synchronized first [13], [20]. The RNC sends a CELL SYNCHRONISATION ADJUSTMENT message to the late-entrant Node B for SFN update only [13], [20].

Frequency acquisition of the late entrant Node B is started when the RNC instructs the late entrant Node B first to “listen” to the regular schedule of cell synchronization bursts of the surrounding cells [20]. Using the cell burst plan, the RNC will signal the transmission schedule of the surrounding Node Bs to the late entrant Node B within the CELL SYNCHRONISATION INITIATION REQUEST message. The late entrant

Node B will detect the synchronization cell burst from the synchronization slot, N_{slot}, and try to lock to the frequency of the signal by using PLL. Once late entrant Node B achieved a frequency stability of ±0.05 ppm [21], it sends a report to the RNC using the CELL SYNCHRONIZATION REPORT message [13], [20].

Subsequently, the RNC instructs the surrounding cells of the late entrant Node B to transmit the cell synchronization burst by sending a CELL SYNCHRONIZATION RECONFIGURATION REQUEST message to them [13], [20]. The late entrant cell will detect the cell synchronization burst and send a CELL SYNCHRONIZATION REPORT message to the RNC, informing the timing of the received cell synchronization burst from the respective Node Bs and the cell bursts SIR [13], [20]. Knowing the locations of all the Node B that transmitted, the RNC is able to compute the timing adjustment for the late entrant Node B. This is sent to the late entrant Node B via the Iub [13], [20]. After this is done, the new entrant cell will receive a cell burst plan and is placed in the "steady state" phase, where all the Node Bs are synchronized continuously [13], [20].

2.3 CELL SEARCH IN TDD

Now with the Node Bs synchronized, how a User Equipment (UE) such as a mobile knows which Node B is serving it and the timeslot of that Node B? 3GPP gives a cell search procedure for TDD. There are altogether 3 steps in a TDD cell search [22].

Step1: Primary Synchronization Code Acquisition

During the first step of the cell search procedure, the UE uses the synchronization channel (SCH) primary synchronization code to find a cell [22]. This is typically done with a single matched filter (or any similar device) matched to the primary synchronization code which is common to all cells [22]. A cell can be found by detecting peaks in the matched filter output [22].

Step 2: Code Group Identification and Slot Synchronization

During the second step of the cell search procedure, the UE uses the SCH's secondary synchronization codes to identify 1 out of 32 code groups for the cell found in the first step [22]. This is typically done by correlating the received signal with the secondary synchronization codes at the detected peak positions of the first step. The primary synchronization code provides the phase reference for coherent detection of the secondary synchronization codes [22]. The code group can then uniquely be identified by detection of the maximum correlation values. Each code group indicates a different t_{offset} parameter and 4 specific cell parameters. Each of the cell parameters is associated with one particular downlink scrambling code and one particular long and short basic midamble code. See Table 2.2 for the cell parameters [18].

CELL PARA-METER	Code Group	Associated Codes			Associated t_{offset}
		Scrambling Code	Long Basic Midamble Code	Short Basic Midamble Code	
0	Group 1	Code 0	m_{PL0}	m_{SL0}	t_0
1		Code 1	m_{PL1}	m_{SL1}	
2		Code 2	m_{PL2}	m_{SL2}	
3		Code 3	m_{PL3}	m_{SL3}	
4	Group 2	Code 4	m_{PL4}	m_{SL4}	t_1
5		Code 5	m_{PL5}	m_{SL5}	
6		Code 6	m_{PL6}	m_{SL6}	
7		Code 7	m_{PL7}	m_{SL7}	
. . . .					
124	Group 32	Code 124	m_{PL124}	m_{SL124}	t_{31}
125		Code 125	m_{PL125}	m_{SL125}	
126		Code 126	m_{PL126}	m_{SL126}	
127		Code 127	m_{PL127}	m_{SL127}	

Table 2.2: Table showing the cell parameters

When the UE has determined the code group, it can unambiguously derive the slot timing of the found cell from the detected peak position in the first step and the t_{offset} parameter of the found code group in the second step [22].

Note that the modulation of the secondary synchronization codes also indicates the position of the SCH slot within a 2 frames period, e.g. a frame with even or odd SFN [22]. Additionally, in the case of SCH slot configuration following case 2, the SCH slot position within one frame, e.g. first or last SCH slot, can be derived from the modulation of the secondary synchronization codes [22].

Step 3: Downlink scrambling code, basic midamble code identification and frame synchronization

During the third and last step of the cell search procedure, the UE determines the exact downlink scrambling code, basic midamble code and frame timing used by the

found cell. The long basic midamble code can be identified by correlation over the Primary Common Control Physical Channel (P-CCPCH) with the 4 possible long basic midamble codes of the code group found in the second step [22]. A P-CCPCH always uses the midamble derived from the long basic midamble code and always uses a fixed and pre-assigned channelisation code.

When the long basic midamble code has been identified, downlink scrambling code and cell parameter are also known. The UE can read system and cell specific Broadcast channel (BCH) information and acquire frame synchronization.

From the SCH will allow the UE to know at which timeslot it can find the PCCPCH containing the BCH. With knowledge of where the BCH can be found, the UE can read the system and cell specific BCH information from it and acquire frame synchronization [22].

For handover, once the SIR drops below a threshold level, the UE will inform the serving Node B and the Node B will inform the RNC who in turn give the UE a list of cells which the UE shall monitor in its idle timeslots [22]. The RNC knows which Node Bs are beside the current Node B from the “connectivity matrix”. At the beginning, the UE shall find the synchronization code of the cell using the synchronization channel as in Step 1 of the cell search. Similarly, a new cell can be found by detecting peaks in the matched filter output. The strongest signal obtained will be the cell to be handovered to. Subsequently, Step 2 and Step 3 of the initial cell search is performed to find the downlink scrambling code, basic midamble code identification and frame synchronization of the new cell [22]. The whole process is

controlled by the RNC. After finding the cell, the RNC will inform the UE to do the handover.

CHAPTER 3

ORTHOGONAL CODES

In this chapter, several pseudo-orthogonal codes commonly used for synchronization in spread spectrum CDMA system will be described. In Section 3.1, we shall review the Golay complementary sequence from which the CEC sequences proposed by 3GPP for inter base station synchronization in WCDMA TDD mode are constructed. In section 3.2, we shall describe the *m*-sequence or maximal length (ML) sequence from which a new concatenated *m*-sequence proposed in this thesis as an alternative to the CEC sequence are built. While in section 3.3, we shall review another type of linear shift register sequence known as Gold codes, which we shall use as a comparison bench mark against the other synchronization sequences.

3.1 GOLAY COMPLEMENTARY SEQUENCES

Complementary sequences have found major applications in the field of radar and biotechnology. A set of complementary series is defined as a pair of equally long, finite sequences of two kinds of elements which have the property that the number of pairs of like elements with any one given separation in one series is equal to the number of pairs of unlike elements with the same given separation in the other series [23].

For instance two sequences A = 00010010 and B = 00011101 are complementary. This is illustrated in the Fig 3.1. Sequence A has three like adjacent pairs and sequence B has three unlike adjacent pairs. Moreover, sequence A has three like alternate elements and sequence B has three unlike alternate elements. This continues for all possible separation. In the illustration, sequence A is used to search for the like pairs while sequence B is used to search for the corresponding number of unlike pairs.

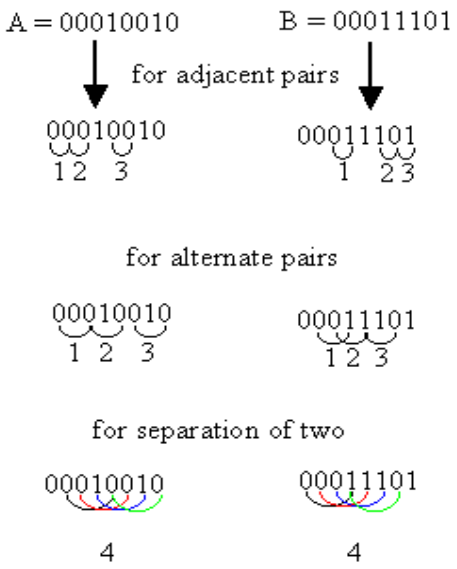


Fig 3.1: Illustration of complementary sequences

Longer complementary Golay sequences can be constructed from the shorter complementary Golay sequences. Consider two complementary sequences A and B of length N each and their elements are $a_1\ a_2 \ldots\ldots\ldots a_N$ and $b_1\ b_2 \ldots\ldots\ldots b_N$ respectively. Other sets of complementary sequences, T_1, T_2, U_1 and U_2, each of length $2N$ can be constructed using either (3.1) or (3.2), as follows [23].

$$\begin{aligned} T_1 &= AB = a_1 a_2 \ldots\ldots\ldots a_N b_1 b_2 \ldots\ldots\ldots b_N \\ T_2 &= AB' = a_1 a_2 \ldots\ldots\ldots a_N b_1' b_2' \ldots\ldots\ldots b_N' \end{aligned} \tag{3.1}$$

$$U_1 = a_1 b_1 a_2 b_2 \ldots\ldots\ldots\ldots a_N b_N$$
$$U_2 = a_1 b_1^{'} a_2 b_2^{'} \ldots\ldots\ldots\ldots a_N b_N^{'} \tag{3.2}$$

where $a^{'}$ and $b^{'}$ are the inverse elements of both a and b respectively.

To illustrate further, consider the two sequences A = 00010010 and B = 00011101 which were presented earlier. Both sequences have 8 elements each. Using (3.1), two new complementary sequences of 16 elements $T_1 = AB$ = 0001001000011101 and $T_2 = AB'$ = 0001001011100010 can be formed. This can be continued to form longer complementary sequences. If (3.2) is used, another new set of complementary sequences with 16 elements each, U_1 = 0000001101011001 and U_2 = 0101011000001100 can be formed. Similarly, this can be continued to form longer complementary sequences.

A generalization of the recursive algorithm for (3.1) is given as follows [25]

$$\begin{aligned} a_0(i) &= \delta(i) \\ b_0(i) &= \delta(i) \\ a_n(i) &= a_{n-1}(i) + W_n b_{n-1}(i - 2^{P_n}) \\ b_n(i) &= a_{n-1}(i) - W_n b_{n-1}(i - 2^{P_n}) \end{aligned} \tag{3.3}$$

where a_n and b_n are two complementary sequences, each of length $N = 2^n$, in the nth iteration, $\delta(i)$ is the Kronecker delta function, i = 0,1,…, N-1, n = 0,1,…, N-1 and W_n is a number of unit magnitude [25]. P_n is any permutation of numbers from 0 to N-1 [25]. Examples on how (3.3) is being used is shown in Appendix B.

The basic property of complementary sequences may be expressed also in autocorrelative terms. Let two complementary sequences both of length N, with 0 and

1 being mapped onto +1 and –1 respectively, and let their respective autocorrelative sequence be defined by [23]

$$c_j = \sum_{i=1}^{N-j} a_i a_{i+j}$$
$$d_j = \sum_{i=1}^{N-j} b_i b_{i+j} \qquad (3.4)$$

where $j = 1, 2, 3, \ldots, N$

The expressions of c_j and d_j in (3.4) are definition of "aperiodic" autocorrelation sequence of Golay complementary series, and they posses the perfect property such that

$$c_j + d_j = 2N \qquad for\ j = 0$$
$$c_j + d_j = 0 \qquad for\ j \neq 0 \qquad (3.5)$$

To illustrate the above defined autocorrelative property, the two sequences *A* and *B*, which were presented earlier is used to extend to two longer complementary sequences *A'* and *B'* of length $N = 512$ using (3.1), and their autocorrelative property are presented in Fig 3.2, Fig 3.3 and Fig 3.4. In both Fig 3.2 and Fig 3.3, the individual autocorrelation of two sequences are simulated while in Fig 3.4, the sum of the autocorrelation of the two complementary series is given, and as can be seen, they sum perfectly to zero.

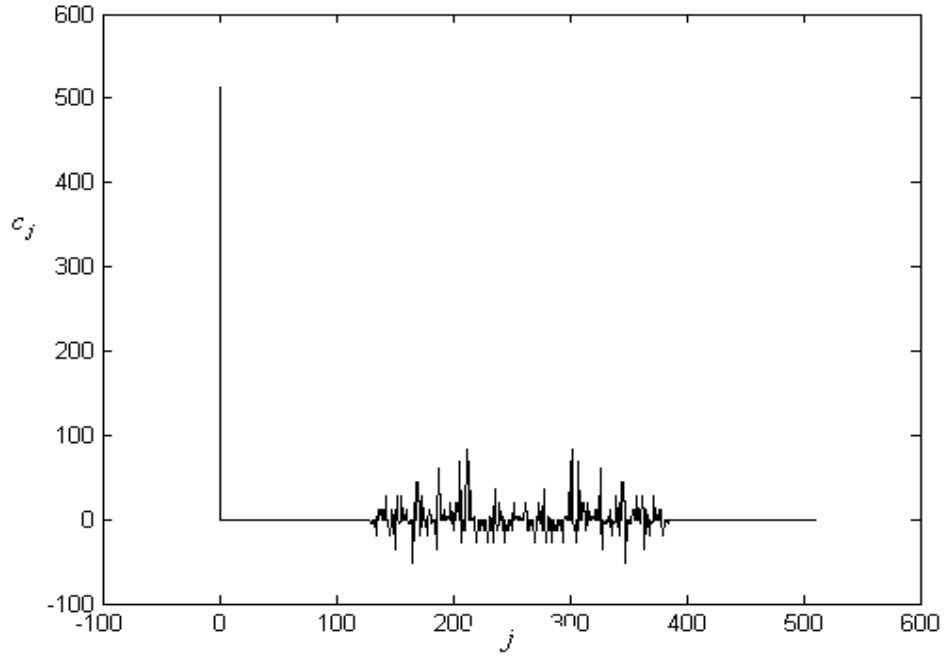


Fig 3.2: Individual aperiodic autocorrelation of complementary sequence *A'*

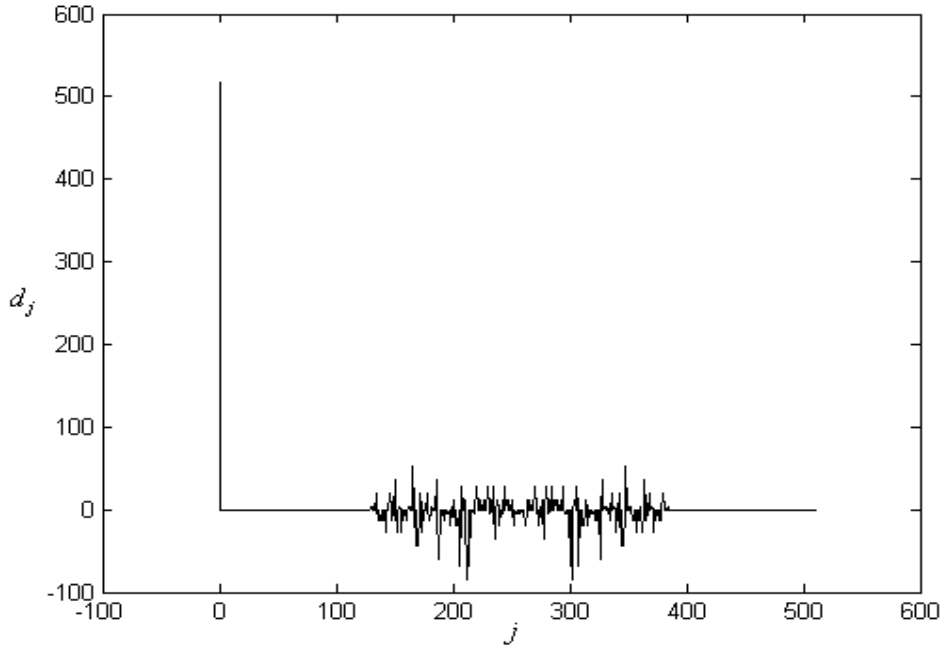


Fig 3.3: Individual aperiodic autocorrelation of complementary sequence *B'*

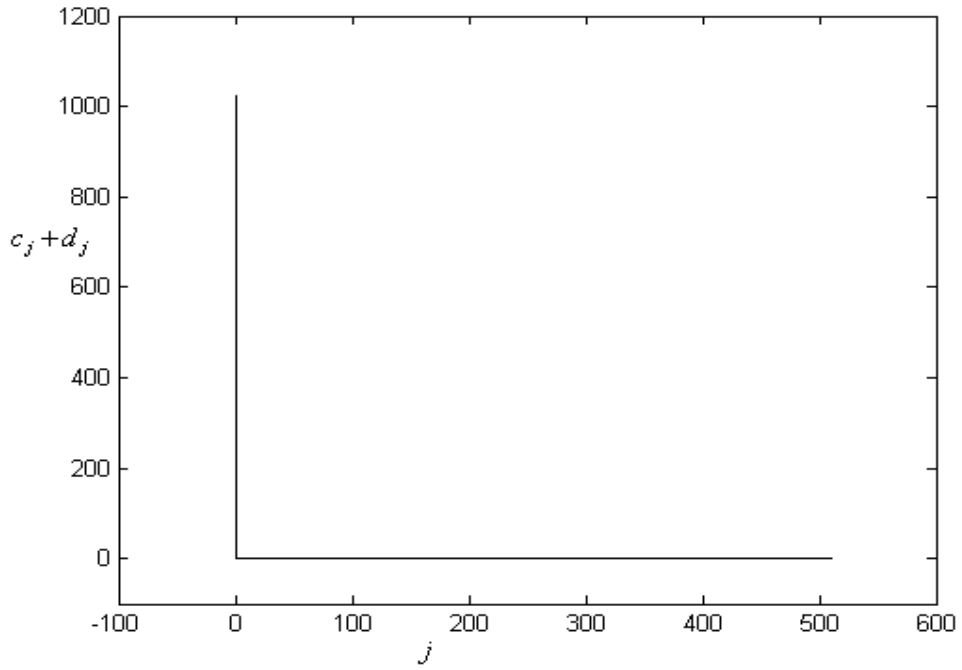


Fig 3.4: Sum of aperiodic autocorrelation of the two complementary sequences *A'* and *B'*

Another property of the Golay complementary sequence is that elements of the sequences can be altered without affecting their complementary properties [23]. Such operations are listed as follows

1. Interchanging the sequences
2. Reversing the first sequence
3. Reversing the second sequence
4. Altering the first sequence by taking the inverse of each element in the first series
5. Altering the second sequence by taking the inverse of each element in the second sequence.
6. Inversing the elements of even order of each sequence

However, each operation can only take place one at a time.

To illustrate, we consider the two sequences A = 00010010 and B = 00011101. Using each operation, the resulting pair of complementary sequences are as follows

1. 00011101 and 00010010
2. 01001000 and 00011101
3. 00010010 and 10111000
4. 11101101 and 00011101
5. 00010010 and 11100010
6. 01000111 and 01001000

Such Golay complementary sequences are often used in communications and radar [24] because the number of different Golay sequences increases rapidly with the length of the sequence. The number of different sequence of length *N*, is given by [24]

$$N!\ 2^{N-1} \tag{3.6}$$

3.2 *M*-SEQUENCES

A Pseudo-Noise (PN) sequence is a periodic sequence of 0's and 1's with certain noise-like autocorrelation properties. A maximal length (ML) or *m*-sequence is a specific class of PN sequence often used in spread spectrum communications. A comprehensive discussion of ML sequence requires the use of a finite field mathematics which we shall first review in the following.

A set of integers {0,1, …, *M*-1}, where *M* is prime, with mod-*M* addition and multiplication is known as a field. A field with p^m elements, with *m* being an integer, is called an extension field and a field having *M* elements is called a Galois field, GF(*M*). In the generation of *m*-sequences, we are interested in GF(2) and its extension field GF(2^m). GF(2^m) is a field consisting of polynomials, *g(D)*, of degree at most *m*-1 with coefficients in GF(2) or

$$g(D) = g_0 + g_1 D + g_2 D^2 + ... + g_{m-1} D^{m-1} \qquad with\ g_i \in \{0,1\} \tag{3.7}$$

To illustrate the two binary operations of addition and multiplication in a field, we assume two polynomials, such as *f*(*D*) and *g*(*D*),

$$\begin{aligned} g(D) &= g_0 + g_1 D + g_2 D^2 + ... + g_{m-1} D^{m-1} \qquad with\ g_i \in \{0,1\} \\ f(D) &= f_0 + f_1 D + f_2 D^2 + ... + f_{m-1} D^{m-1} \qquad with\ f_i \in \{0,1\} \end{aligned} \tag{3.8}$$

The GF(2^m) addition of *f*(*D*) and *g*(*D*), results in *h(D)* which is given by

$$h(D) = h_0 + h_1 D + h_2 D^2 + ... + h_{m-1} D^{m-1} \qquad with\ h_i \in \{0,1\} \tag{3.9}$$

where $h_i = (g_i + f_i)$ mod-2 where $i = 0,\ldots, m-1$.

Multiplication of two elements of GF(2^m) is defined by using special polynomials of primitive polynomials, *p(D)* [26]. A polynomial is primitive if it has the following properties.

1 It is irreducible, i.e. it cannot be factorized into products of polynomials of lower degrees.

2 The smallest degree for which *p*(*D*) divides D^n-1 is $n = 2^m$-1.

The product of two elements of GF(2^m)is defined as the remainder when the normal polynomial is divided by the primitive polynomial, *p(D)*, chosen to define the multiplication [26] or

$$f(D)g(D) \equiv \text{Rem}\left\{\frac{f(D)g(D)}{p(D)}\right\} \tag{3.10}$$

To illustrate, assume GF(2^3) and the primitive polynomial $p(D) = D^3 + D + 1$. Then the sequence of non-zero elements of GF(2^3) becomes

Elements	Polynomial Representation
D^0	1
D^1	D
D^2	D^2
D^3	$D + 1$
D^4	$D^2 + D$
D^5	$D^2 + D + 1$
D^6	$D^2 + 1$
D^7	1
D^8	D
D^9	D^2

Table 3.1: Table showing the corresponding polynomial representation in the field GF(2^3)

We observe from Table 3.1 that the sequence is periodic after D^7 and there are 7 distinct elements in GF(2^3), which are $\{D^0, D^1, D^2, D^3, D^4, D^5, D^6\}$.

Each element will appear only once and only once as we progress through the sequence because suppose D^i appears as well as D^j where $i < j$ and $i, j < 2^m$-2, then

$$\text{Rem}\left\{\frac{D^{j-i}}{p(D)}\right\}=1 \Rightarrow D^{j-i}=q(D)\,p(D)+1$$

where $q(D)$ *is the quotient* (3.11)

$$\Rightarrow D^{j-i}+1=q(D)\,p(D)$$

$$\Rightarrow p(D) \text{ divides } D^{j-i}+1$$

However $p(D)$ is primitive, which means that the smallest n for which $p(D)$ divides is $n = 2^m\text{-}1$ but $j\text{-}\ i < 2^m\text{-}1$, which is a contradiction. Thus the elements will appear only once.

Elements of GF(2^m) can be realized by a dividing circuit, made up of shift registers. Assuming *p(D)* is the primitive polynomial, the elements GF(2^m), *b(D)*, can be realized by Fig 3.5

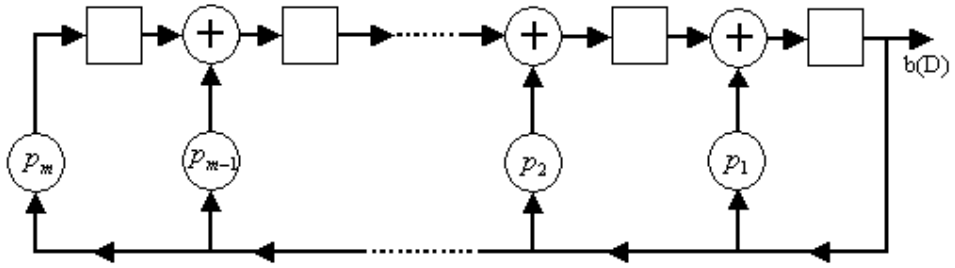


Fig 3.5: Dividing circuit used to realized elements in GF(2^m)

For a extension field of GF(2^3) with $p(D) = D^3 + D + 1$, the circuit can be realized in Fig 3.6

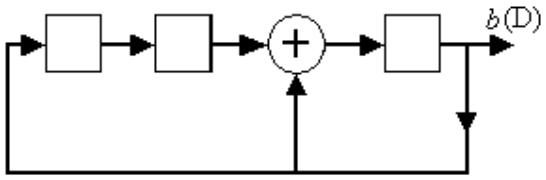


Fig 3.6: Circuit to generate the elements of GF(2^3) with $p(D) = D^3 + D + 1$

If the feedback connection of the generator of the dividing circuit is a primitive polynomial, then the output *b*(D) will be an *m*-sequence. If and only if the feedback connection is a primitive polynomial, the state of the shift register will pass through all the non-zero 'states'. To illustrate, assume the initial state of Fig 3.6 is 001, then the shift register will pass through all the non-zero 'states' as follows

State of the shift register	Output *b*(D)
001	1
101	1
111	1
110	0
011	1
100	0
010	0

Table 3.2: States of the shift register with $p(D) = D^3 + D + 1$ as the feedback connection

The *m*-sequence generated is periodic and has a period of

$$N = 2^r - 1 \qquad (3.12)$$

where *r* is the number of shift register used in the Galois generator .

M-sequences are deterministic signal but appear to have the statistical properties of sampled white noise. However, to an unauthorized listener, these sequences appear to be a random signal. The four unique properties of an *m*-sequence are listed as follows [26], [27].

1. In an *m*-sequence of period, *N*, there are (*N*+1)/2 *ones* and (*N*-1)/2 *zeroes*. This is because the initial stage cannot be the all *zeroes* stage and the shift registers will subsequently pass through all the non-zero binary "states" of *N* again. For example, for the above example where $N = 15$, the shift register will pass through all the possible 15 non-zero binary stages (from 0001 to 1111). This property is known as the *balance property* [26].

2. In a period of *m*-sequence with period $N = 2^r-1$, there are 2^{r-2} runs of *ones* alternating with 2^{r-2} runs of *zeroes*. Half the runs of each kind have length 1, one-fourth of the runs of each type have length 2. Half the runs of each kind have length 1, one-fourth of the runs of each type have length 2, and in general $1/2^k$ of the runs of each type (2^{r-k-2} runs of each type) have length *k*, for $1 \leq k \leq r-2$. In addition, there is a single run of *r-1 zeroes*, and a single run of *r ones*. For example, in the above example, there are 4 runs of *zeroes* and *ones* each. For the 4 runs of *zeroes*, half of the runs are of length 1 (2 single *zeroes*), 1 (one-fourth of 4 runs of *zeroes*) run of length 2 and there is also a single run of 3 *zeroes*. Likewise for the runs of *ones*. This is known as the *run property* [27].

3. Another property is the *cycle-and-add property*, which states that if an *m*-sequence is modulo-2 added term-by term to a cyclic shift of itself, the result is another cyclic shift of itself. This is due to the "closure" property of non-zero 2^r-1 distinct r-tuples of initial load [26].

4. An *m*-sequence has two level of correlation. When an *m*-sequence, *a(i)*, is compared with any cyclic shift of itself, the sequence has 2^{r-1}-1 "agreements" and 2^{r-1} "disagreements". If we consider the sequence of +1's and –1's instead of 1and 0, the autocorrelation function $K_1(t)$ is given by

$$K_1(t) = \theta(a,a) = \begin{cases} N & for\ \ m.N\ \ where\ m = 1,2,..... \\ -1 & otherwise \end{cases} \qquad (3.13)$$

The autocorrelation will be similar to the diagram shown in Fig 3.7.

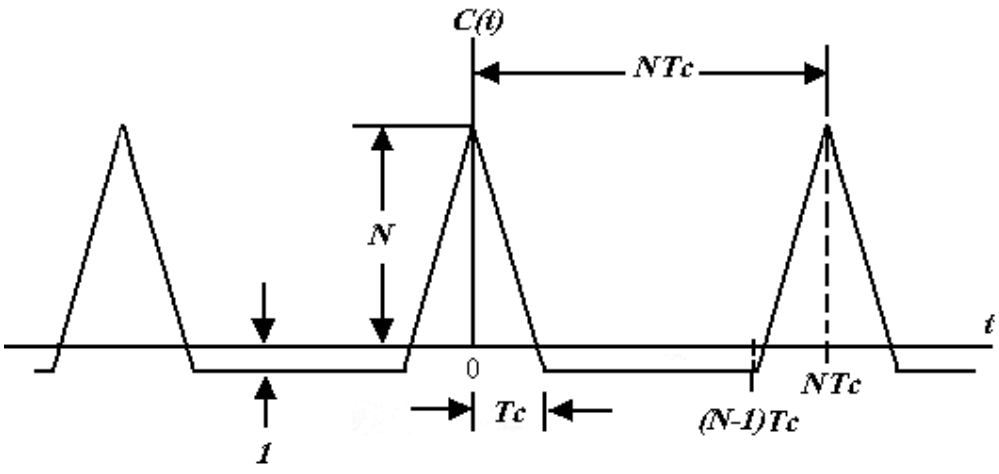


Fig 3.7: Cyclic autocorrelation function for an *m*-sequence with chip duration *Tc*

A mathematical proof for the autocorrelation function is given as follows [27]

Proof

Consider the members of the *m*-sequence, *a(i)*, are $a_1, a_2, \ldots, a_N$, where N is the length or period of the sequence and $a_i \varepsilon GF(2)$. Consider $\eta(a_i)$ as the isomorphism of {0,1} onto {1,-1} respectively, then the autocorrelation, $K_1(t)$, for $t = mN$ for $m = 1,2,\ldots$ is expected to be N. For $t \neq mN$,

$$K_1(t) = \sum_{i=1}^{N} \eta(a_i)\eta(a_{i+t})$$
$$= \sum_{i=1}^{N} \eta(a_i + a_{i+t})$$

From *property 3*, we know that the mod-2 additional of two *m*-sequences yield another *m*-sequence. And from *property 1*, we know that there is one more ones than zeroes in an *m*-sequence. Thus $K_1(t) = -1$ for $t \neq mN$.

Although *m*-sequences exhibit a two-valued autocorrelation, the cross correlation between any preferred pair of different *m*-sequences takes on three values [26], [28]. Such preferred pair of *m*-sequences can be used to form Gold sequences which will be discussed in the next section. The cross correlation between two *m*-sequences of similar length takes on three values, *–t(r), -1* or *t(r)-2*, where *r* is the number of flip-flops used to generate the *m*-sequence and *t(r)* is given by [28]

$$t(r) = \begin{cases} 1 + 2^{\frac{(r+1)}{2}} & \text{for } r \text{ odd} \\ 1 + 2^{\frac{(r+2)}{2}} & \text{for } r \text{ even} \end{cases} \qquad (3.14)$$

However the autocorrelation and cross correlation of an *m*-sequence will take on 2 and 3 values respectively if the *m*-sequence is periodic. If it is aperiodic, both (3.13) and (3.14) will not hold.

For the synchronization of Node B in WCDMA TDD mode, the synchronization code is a burst. Thus we consider the aperiodic autocorrelation and cross correlation of the *m*-sequence. Consider two *m*-sequences, *x* and *y*, both with length *N*, the autocorrelation, Am_j, and cross correlation, Cm_j, is given by

$$\begin{aligned} Am_j &= \sum_{i=1}^{N-j} x_i x_{i+j} \\ Cm_j &= \sum_{i=1}^{N-j} x_i y_{i+j} \end{aligned} \tag{3.15}$$

where $j = 1, 2, \ldots, N$.

Fig 3.8 and Fig 3.9 show the aperiodic autocorrelation and cross correlation of an *m*-sequence formed by the primitive polynomial given in octal form [4005]. In Fig 3.8, for $j \neq 0$, maximum of Am_j is 46 at $j = 1605$. In Fig 3.9, we show the cross correlation between this *m*-sequence and another *m*-sequence formed by the primitive polynomial given in octal form [4215]. Both *m*-sequences have a length of 2047.

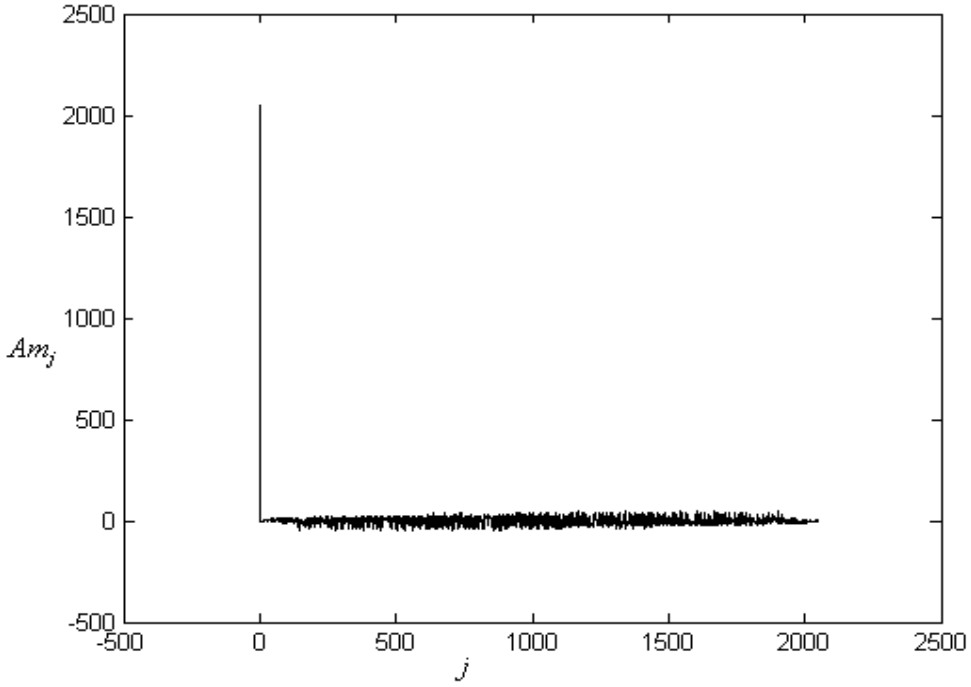


Fig 3.8: Aperiodic autocorrelation of the *m*-sequence formed by the octal [4005]

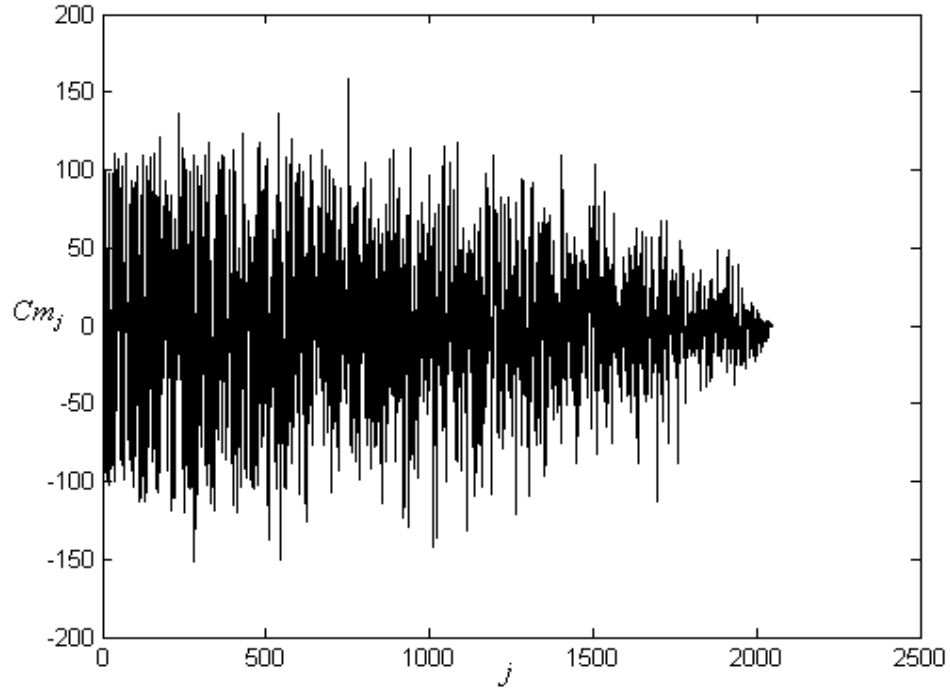


Fig 3.9: Aperiodic cross correlation between *m*-sequence formed by [4005] and [4125]

The number of different *m*-sequences for a given length is determined by the number of different generating primitive polynomials and is given by [24]

$$\frac{\varphi(2^r - 1)}{r} \tag{3.16}$$

where r is the number of shift registers in the Galois generator and $\varphi(x)$ is the Euler function. The mathematical formula of the Euler function is presented in Appendix C. Comparing (3.16) with (3.6), as both N and r increase, the number of available complementary sequences increases much more than the number of available *m*-sequences [24]. Hence complementary sequences are sometimes preferred over *m*-sequences.

Spread spectrum communication using *m*-sequences are often used to protect data transmissions from intentional jamming or to preclude unintentional listening by other parties, However, when the jammer receives a noise-free copy of the transmitted signal, $p(D)$ and the initial state of the generator can be determined quite easily. Therefore *m*-sequences are a poor choice for higher security applications. Thus Gold codes are often considered.

3.3 GOLD SEQUENCE

Consider an *m*-sequence represented by *b(x)* of period *N*. If *b(x)* is sampled at every *q*th symbol, another sequence *b'(x)* can be formed. *b'(x)* is known as a decimation of *b(x)* and is commonly denoted by $b'(x) = b[q]$. The new sequence *b'(x)* is known as a Gold code. The decimations of an *m*-sequence may or may not yield another *m*-

sequence. The decimation that yields another *m*-sequence is called a 'proper' decimation. For a 'proper' decimation, a common rule for decimation is given by [26]

$$\gcd(N, q) = 1 \tag{3.18}$$

where gcd is the greatest common divisor

For 'proper' decimation, q is an odd integer given by $q = 2^k + 1$ or $q = 2^{2k} - 2^k + 1$ and the sequence *b(x)* and *b'(x)* are known as a preferred pair of *m*-sequence. Then the family of codes formed by

$$\begin{Bmatrix} b(D), b'(D), b(D) + b'(d), b(D) + Db'(D)..... \\ .....b(D) + D^{N-2}b'(D), b(D) + D^{N-1}b'(D) \end{Bmatrix} \tag{3.19}$$

is called the set of Gold codes for *b(D)* and *b'(D)* [26]. Evidently, there are *N+2* sequences in a Gold code set. With the exception of *b(D)* and *b'(D)*, the set of Gold sequence are not *m*-sequences and their autocorrelation functions are not two-valued but has the same three values as the cross correlation of *m*-sequences [26]. The autocorrelation of these Gold code, $K_2(t)$, formed by two *m*-sequences, has the following property

$$K_2(t) = \begin{cases} N & c = m \cdot N \ \textit{where}\ m = 1, 2,.... \\ \phi & \textit{otherwise} \end{cases} \tag{3.20}$$

where the value of ϕ takes on three values *–t(r), -1 or t(r)-2*, and *t(r)* is similar to that given in (3.14). Since the cross correlation of such spread spectrum sequences are bounded, it will minimize interlink interferences.

Similarly, (3.20) holds if the Gold sequence is periodic. Once again, if we plot both the aperiodic autocorrelation and cross correlation of the Gold sequences, it will be different. Consider two Gold sequences, p and q, both with length N, the autocorrelation, Ag_j, and cross correlation, Cg_j, is given by

$$\begin{aligned} Ag_j &= \sum_{i=1}^{N-j} p_i p_{i+j} \\ Cg_j &= \sum_{i=1}^{N-j} p_i q_{i+j} \end{aligned} \tag{3.21}$$

where $j = 1, 2, \ldots, N$.

Fig 3.10 and Fig 3.11 shows the aperiodic autocorrelation and cross correlation of a Gold sequence formed by the primitive polynomials given in octal form [4005] and [4125]. In Fig 3.10, for $j \neq 0$, maximum of Ag_j is 88 at $j = 583$. In Fig 3.11, we show the cross correlation of this gold sequence with another Gold sequence formed by the primitive polynomials given in octal form [4445] and [4005]. Both Gold sequences have a length of 2047.

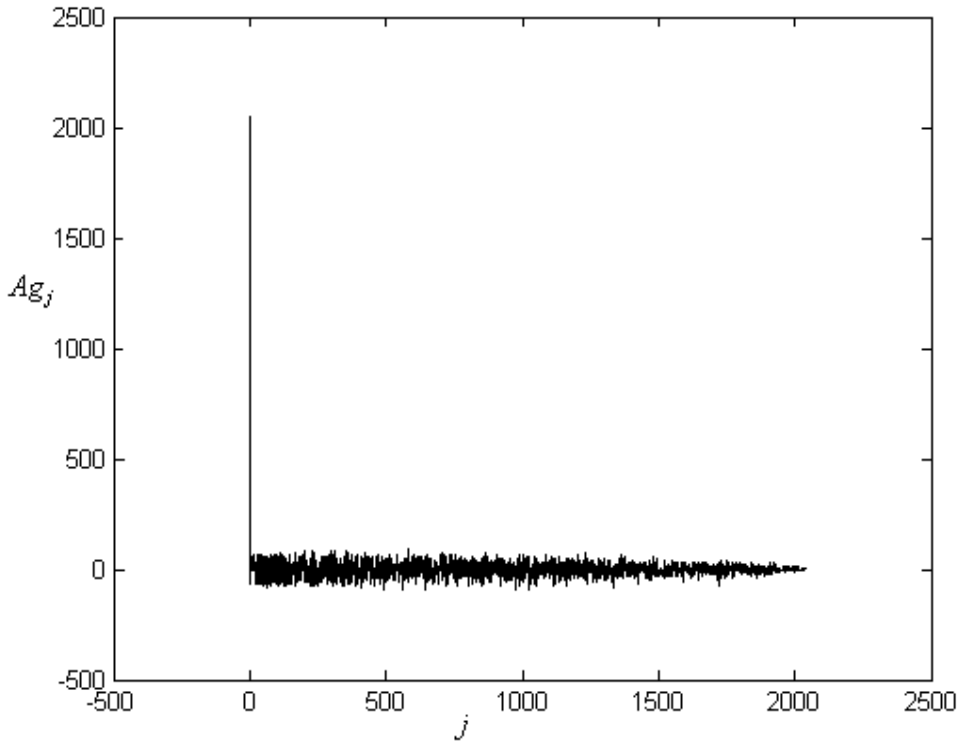


Fig 3.10: Aperiodic autocorrelation of the Gold sequence formed by the octal [4005] & [4215]

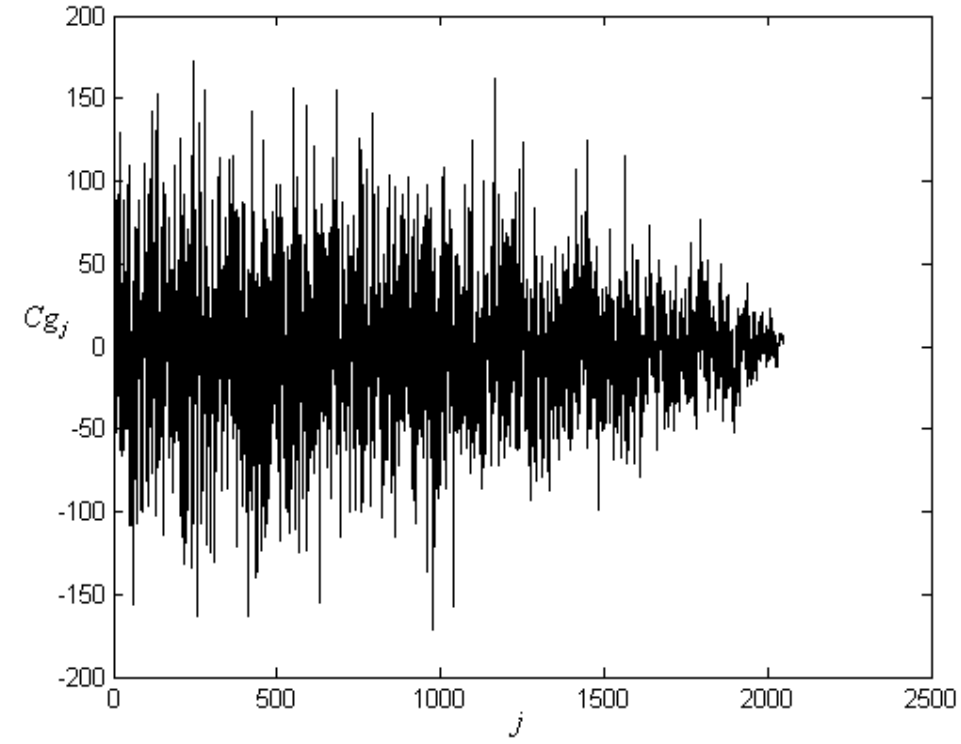


Fig 3.11: Aperiodic autocorrelation of a Gold sequence

CHAPTER 4

CONCATENATED EXTENDED COMPLEMENTARY SEQUENCES (CEC SEQUENCES)

In this chapter, we look at a new so called concatenated extended complementary sequence (CEC) for inter-base station synchronization developed by 3GPP. In Section 4.1, we will first describe the construction principle of the CEC sequence with multiple code offsets. We also highlight the key property of such CEC sequences. In section 4.2, the aperiodic correlation properties of the CEC sequences are presented. An analysis of why CEC sequences with multiple code offsets are poor as compared to *m*- and Gold sequences is also given.

My contribution to this chapter is in the demonstration of the aperiodic autocorrelation properties of CEC with multiple code offsets. Simulations of the cross correlation properties of CEC with multiple code offsets are presented and the analysis of its poor cross correlation properties due to the formation of large peaks, is provided in section 4.2

4.1 CONSTRUCTION PRINCIPLE OF CEC SEQUENCES WITH MULTIPLE CODE OFFSETS

The synchronization code developed by 3GPP for inter-base station synchronization is the CEC sequences with multiple code offsets [12], [15]. A CEC sequence is derived

from a pair of Golay complementary sequence [23]. Consider a pair of complementary sequence, *s(n)* and *g(n)*, each having a length of *N*. The construction principle for the *s(n)* part for the CEC sequence with multiple offsets is shown in Fig 4.1.

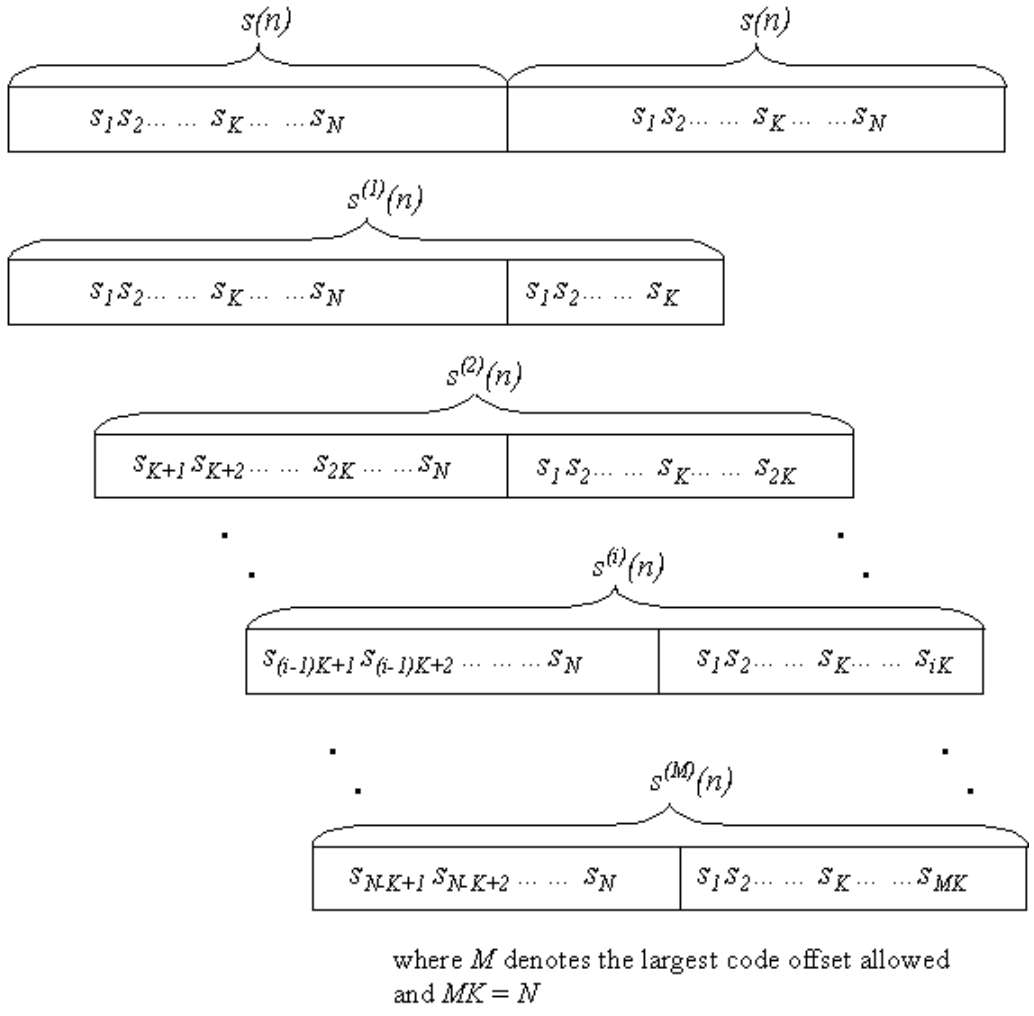
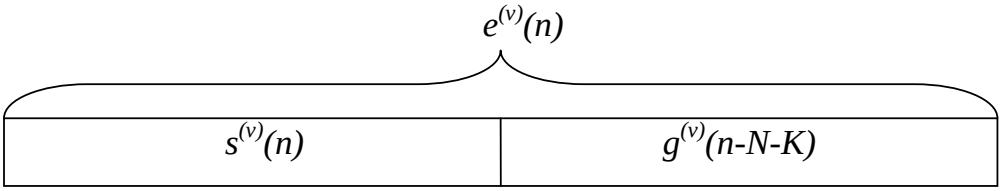


Fig 4.1: Construction principle for different code offsets of the basic sequence *s(n)*

The maximum number of available offsets *M* is given by $M = \frac{N}{K}$ [12], where *N* is the length of either *s(n)* or *g(n)* and *K* is known as the post extension of the basic sequences *s(n)* or *g(n)*.

Similarly the *g(n)* part for the CEC sequence with multiple offsets can be constructed likewise and the final CEC sequence with multiple offsets, $e_v(n)$, has a structure as shown in Fig 4.2.



where *v* denotes the different code offsets

Fig 4.2: Structure of CEC sequence with multiple code offset

$e^{(v)}(n)$ is given by

$$e^{(v)}(n) = s^{(v)}(n) + g^{(v)}(n - N - K) \qquad (4.1)$$

A key advantage of using CEC sequence with multiple code offset is that the orthogonality between different $e^{(v)}(n)$ is preserved. This means that $e^{(1)}(n)$ is orthogonal to $e^{(v)}(n)$, for $v \neq 1$. For example two binary Golay complementary sequences, 11101101 and 00011101, having a length of 8 each, are extended to form a pair of complementary sequences with a length of 1024 using (3.1). In this simulation, the post extension bits, *K*, is taken to be 64 and $M = 1024/64 = 16$. The preservation of the orthogonality for the *s(n)* part is shown in both Fig 4.3 and Fig 4.4. The output of the cyclic correlator for $s^{(3)}(n)$, $g^{(3)}(n)$ and the correlation sum of $s^{(3)}(n)$ and $g^{(3)}(n)$ are shown in Fig 4.3 while The output of the cyclic correlator for $s^{(11)}(n)$, $g^{(11)}(n)$ and the correlation sum of $s^{(11)}(n)$ and $g^{(11)}(n)$ are shown in Fig 4.4.

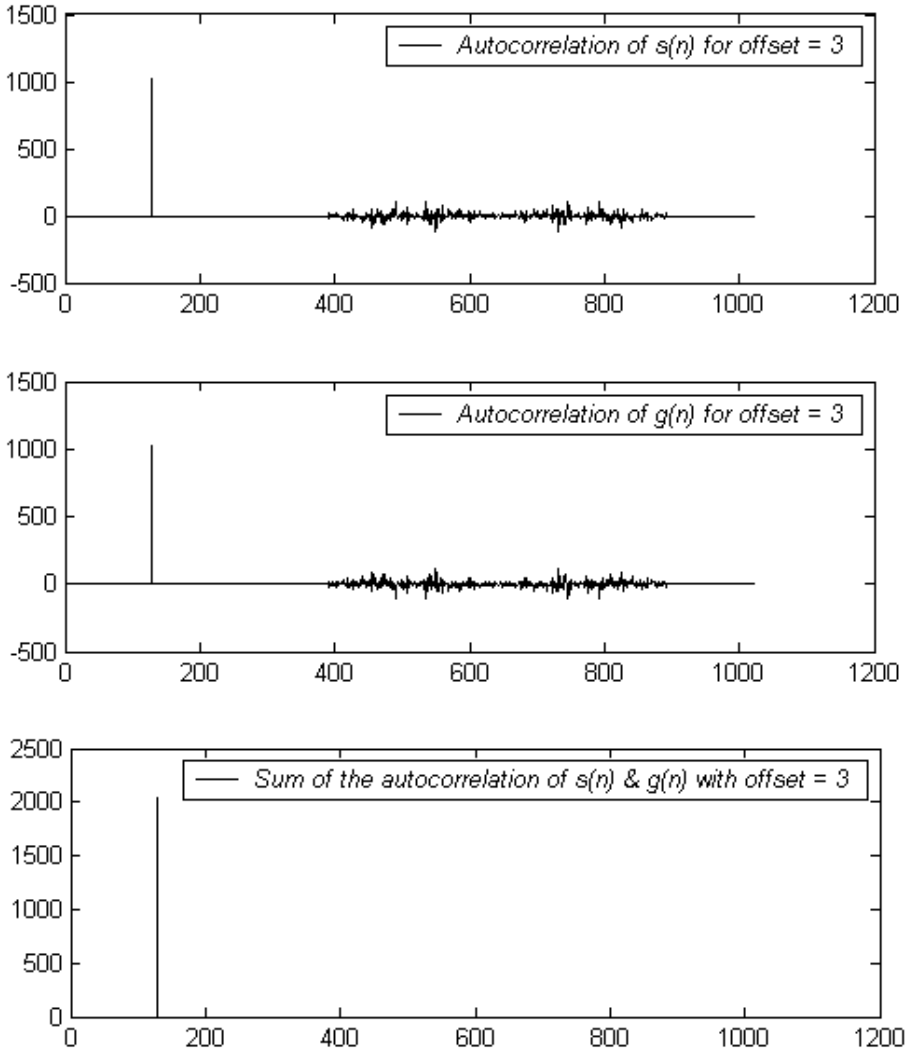


Fig 4.3: Cyclic correlation for $s^{(3)}(n)$ and $g^{(3)}(n)$ and the sum of their correlation

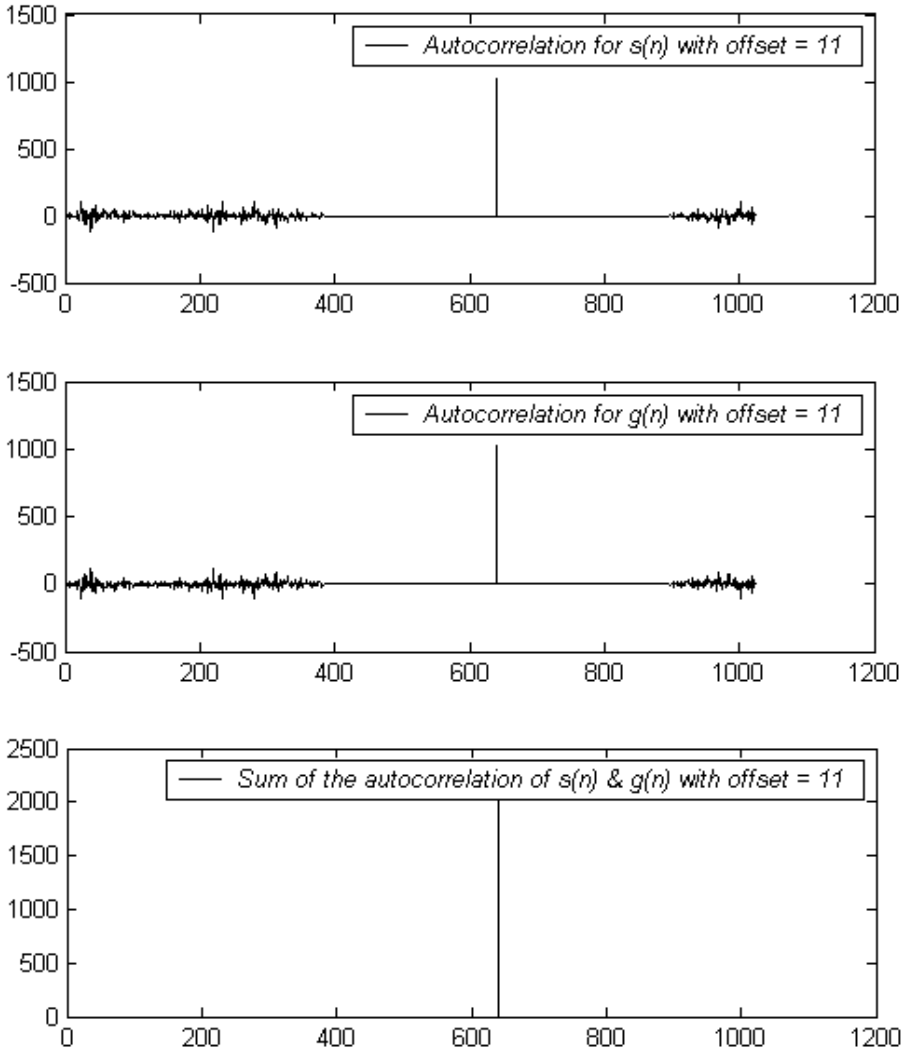


Fig 4.4: Cyclic correlation for $s^{(11)}(n)$ and $g^{(11)}(n)$ and the sum of their correlation

From both Fig 4.3 and 4.4, we observe that each individual $e^{(v)}(n)$ form a set of complementary sequence. This is an advantage because only one CEC sequence with multiple code offset is required for a single RNC, a pair of complementary sequence can be used as synchronization codes for M RNC. It is recommended by 3GPP that Golay complementary sequences of length, N = 1024 with a post extension length, K = 64 be used for inter-base synchronization [12], [15]. Thus a pair of Golay

complementary sequences with $N = 1024$ and $K = 64$ can yield 16 possible codes for synchronization purposes.

4.2 APERIODIC CORRELATION PROPERTIES

There are two different ways to perform a correlation of the CEC sequence. Assume that the received signal is $e^{(v)}(n)$, the first approach is to perform a simple aperiodic correlation on $e^{(v)}(n)$ itself. Another approach is to exploit the complementary property of the constituent sequence building the CEC sequence. The receiver will work as follows.

A. First correlate the received signal with a local replica of $s(n)$ aperidoically.
B. Subsequently, correlate the received signal at a $N+K$ chips offset with a local replica of $g(n)$ aperidoically.
C. Finally add up the correlation results from (A) and (B) in order to compute the correlation sum.

The correlation of the received signal in steps (A) to (C) is efficiently implemented by the Enhanced Golay Correlator. This is shown in Fig 4.5.

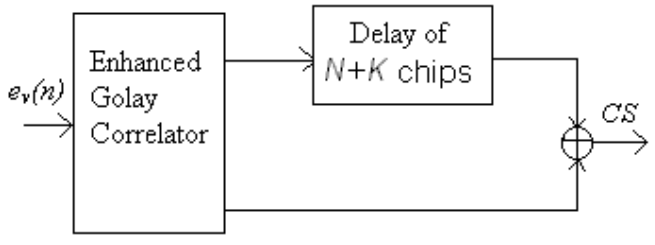


Fig 4.5: Simplified receiver for 3 step correlation procedures.

The correlation sum CS obtained in step (C) is given as follows

$$CS = \eta(e^{(v)}(n), s(n)) + \eta(e^{(v)}(n), g(n-N-K)) \tag{4.2}$$

where $\eta(e^{(v)}(n), s(n))$ denotes the aperiodic correlation between $e^{(v)}(n)$ and $s(n)$.

(4.2) can be developed into

$$\begin{aligned} CS = {} & \eta(s^{(v)}(n), s(n)) + \eta(g^{(v)}(n-N-K), g(n-N-K)) \\ & + \eta(s^{(v)}(n), g(n-N-K)) + \eta(g^{(v)}(n-N-K), s(n)) \end{aligned} \tag{4.3}$$

Since $s^{(v)}(n)$ is formed by cyclically extending $s(n)$ by K chips, $\eta(s^{(v)}(n), s(n))$ is similar to a an autocorrelation of itself up till $N+K$ chips. Thus $\eta(s^{(v)}(n), s(n)) = \eta_{s,s}(n-N) + \eta_{s,s}(n) = \delta(n-N) + \delta(n)$. $\eta_{s,s}(n)$ is the aperiodic cross correlation from 0 to K chips while $\eta_{s,s}(n-N)$ is the aperiodic cross correlation for the cyclically extended K chips. Similarly for $\eta(g^{(v)}(n-N-K), g(n-N-K)) = \eta(g^{(v)}(n), g(n)) = \eta_{g,g}(n-N) + \eta_{g,g}(n) = \delta(n-N) + \delta(n)$. Since we are looking from 0 to K chips, the last two terms in (4.3) can be ignored because $\eta(s^{(v)}(n), g(n-N-K))$ and $\eta(g^{(v)}(n-N-K), s(n))$ will have correlation values only for $> K$ chips, as $g(n-N-K)$ and $g^{(v)}(n-N-K)$ are appended behind $s(n)$ and $s^{(v)}(n)$ respectively. Thus (4.3) can be reduced to

$$\begin{aligned} & CS = \eta(s^{(v)}(n), s(n)) + \eta(g^{(v)}(n-N-K), g(n-N-K)) \\ & = \delta(n) + \delta(n-N) + \delta(n) + \delta(n-N) \\ & = 2\delta(n) + 2\delta(n-N) \end{aligned} \tag{4.4}$$

The resulting autocorrelation sum in (4.4) can be used to detect the received signal as (4.4) indicates that there is no secondary peaks between 0 to K chips.

However in this thesis, we use the first approach. This is to compare with the aperiodic correlation properties of the *m*-sequence, Gold sequence and the concatenated *m*-sequences. To illustrate the aperiodic correlation properties of the CEC sequences with multiple code offsets, once again, the binary Golay complementary sequences, 11101101 and 00011101 are extended to form a pair of

complementary sequences with a length of 1024 using (3.1). *K* is taken to be 64. With {0,1} being replaced by {1, -1}, the aperiodic autocorrelation for the CEC sequences with an offset of 3 and 11, given by η_j, are shown in Fig 4.6 and Fig 4.7 respectively. *j* is the shift of the elements in the CEC. For each CEC with multiple code offset, they have a length of 2176.

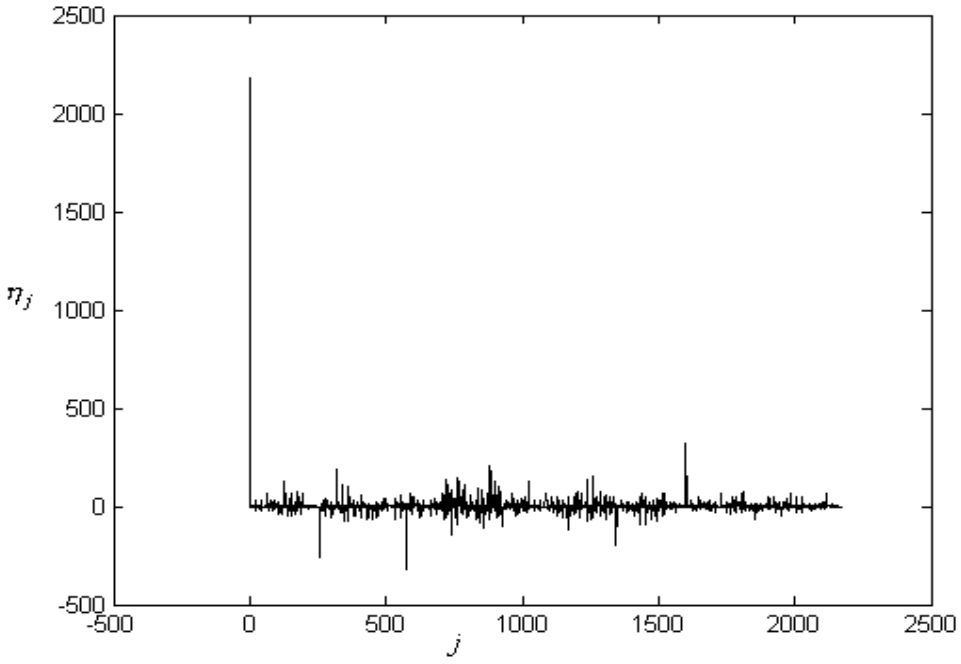


Fig 4.6: Aperiodic autocorrelation property of a CEC sequence formed by code offset 3

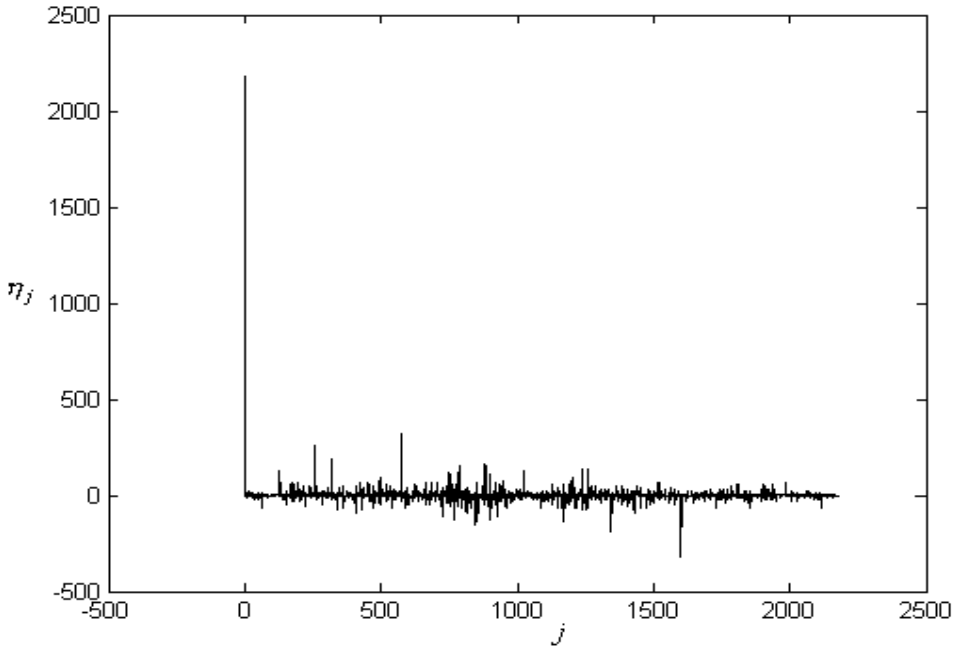


Fig 4.7: Aperiodic autocorrelation property of a CEC sequence formed by code offset 11

From Fig 4.6 and Fig 4.7, we observed that both CEC sequences with offsets 3 and 11 have good autocorrelation property as there are no significant secondary peaks around its main peak. For the CECs with code offsets of 3 and 11, both of their highest side peak is 320 at j =1601 and 577 respectively. Although it is favorable to use CEC sequences with multiple code offset to synchronize cells as this would reduce the number of binary complementary sequence used for synchronization purposes, their cross correlation between each other leaved much to be desired. The cross correlation between the CEC sequence with an offset of 3 and 11 is shown in Fig 4.8, in which we observe one distinct peak at j = 512.

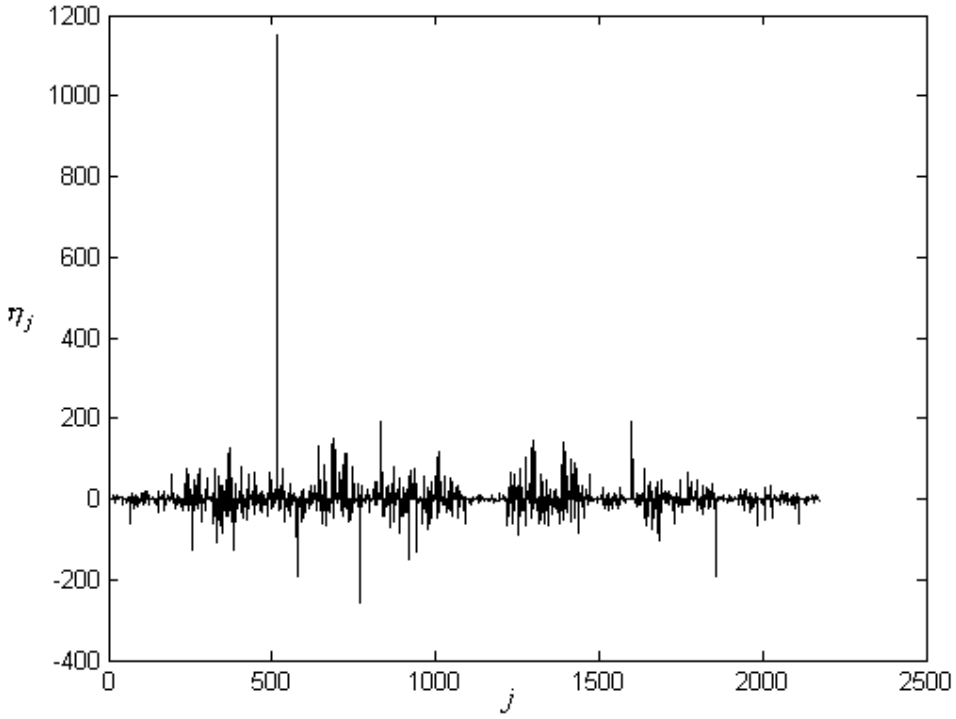


Fig 4.8: Cross correlation between two CEC sequences with code offsets 3 and 11

This implies that if two adjacent RNC uses CEC sequence with offset 3 and 11, interference would occur. To illustrate, see Fig 4.9. Assume there is a Node B in each cell namely 1*a*, 1*b*, 1*c*, 2*a*, 2*b* and 2*c* in the middle. If cells 1*a*, 1*b* and 1*c* are controlled by RNC 1 and are using CEC sequence with a code offset of 3 for synchronization, while cells 2*a*, 2*b* and 2*c* are controlled by RNC 2 using CEC sequence with code offset 11 for synchronization. Then during the initial synchronization phase, if cell 1*c* is transmitting the synchronization sequence while its adjacent cell, cell 2*a* is in the listening mode, cell 2*a* might lock onto a wrong peak when it cross correlate with the synchronization sequence of cell 1*c* and sends an erroneous message to RNC 2.

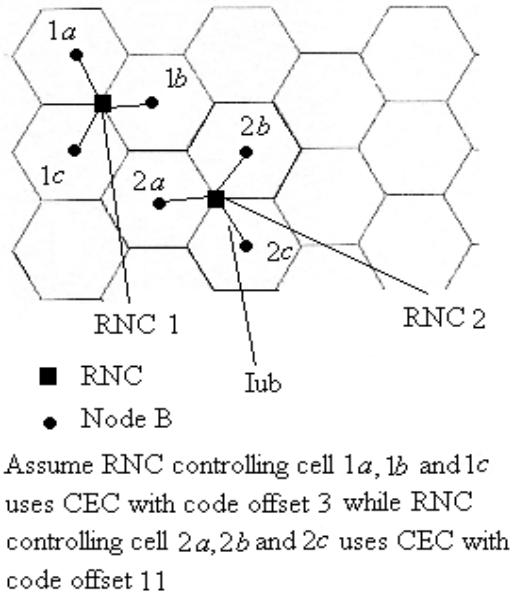


Fig 4.9: Cluster of 3 cells using CEC with different code offsets

So why does the peak appear?

In order to explain this phenomenon, consider the structure of the *s(n)* part of the CEC sequences with offset 3 and 11 as shown in Fig 4.10.

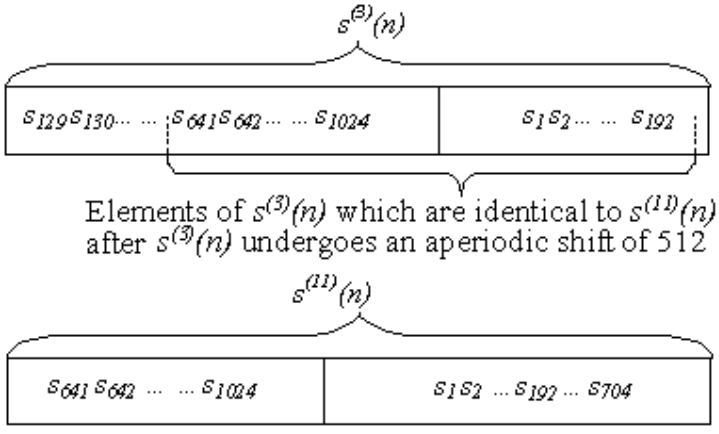


Fig 4.10: Structure of $s^{(3)}(n)$ and $s^{(11)}(n)$

In the calculation of the cross correlation function, we assume $s^{(11)}(n)$ is the reference signal. After a aperiodic shift of 512, $s^{(3)}(n)$ will be identical to $s^{(11)}(n)$ from s_{641} to s_{1024}, and from s_1 to s_{192}, as shown in Fig 4.10 . Hence in the calculation of the cross correlation function, the peak arises.

This phenomenon where two peaks appearing in its cross correlation function will happen to any CEC sequences of any offsets as the sequences are derived from the same CEC sequence.

After looking at this analysis and observing Fig 4.1 again, we can deduce that the cross correlation between two CEC sequence with offset x and y (where $y > x$), with K as the post extension bits, then the first peak will occur at $(y-x)K$ chips Thus the cross correlation between CEC sequences of different multiple offsets is poor due to the large peak. Thus newer synchronization codes are being developed. This will be discussed in the next Chapter.

CHAPTER 5

CONCATENATED SEQUENCES FORMED BY *m*-SEQUENCES

In this chapter, we shall explore a new type of synchronization codes which are formed by the concatenation of *m*-sequences. The motivation behind using concatenated sequences is to increase the numbers of available synchronization codes. In section 5.1 we shall present the structure of the concatenated sequence. The aperiodic autocorrelation and cross correlation will be presented in section 5.2. Finally a comparison of the aperiodic autocorrelation and cross correlation between *m*-sequences, concatenated *m*-sequences, Gold sequences and CEC sequences will be given in section 5.3.

5.1 CONSTRUCTION PRINCIPLE

For the generation of this new sequence, we consider two sequences denoted by *a(n)* and *b(n)*, for either an *m*-sequence or a Gold sequence respectively. These sequences can be generated using Galois generators, and the polynomials (or the values of the octals) used to generate the *m*-sequences can be found in [26]. Let *q(n)* be the concatenated sequence of *a(n)*and *b(n)*. If both *a(n)* and *b(n)* are formed by shift registers consisting respectively of *c* and *d* stages, then the length of *a(n)*, *b(n)* and *q(n)*,denoted respectively by *A*, *B* and *L* will be

$$A = 2^c - 1$$
$$B = 2^d - 1 \qquad (5.1)$$
$$L = 2^c + 2^d - 2$$

The structure of the new sequence is shown in Fig 5.1.

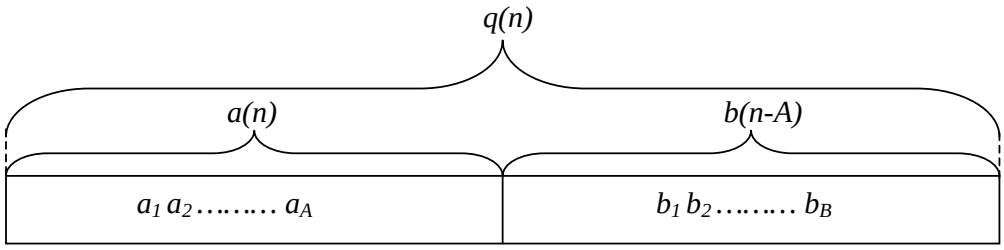


Fig 5.1: Structure of the concatenated sequence

5.2 APERIODIC CORRELATION PROPERTIES

Consider two concatenated sequences, α and β, the aperiodic autocorrelation and cross correlation, given by θ_j and ρ_j and

$$\theta_j = \sum_{i=1}^{L-j} \alpha_i \alpha_{i+j}$$
$$\rho_j = \sum_{i=1}^{L-j} \alpha_i \beta_{i+j} \qquad (5.2)$$

To illustrate the aperiodic autocorrelation and cross correlation, two *m*-sequences formed by the primitive polynomial given by the octals [2011] and [2415] are concatenated. Fig 5.2 shows the aperiodic autocorrelation of the concatenated sequence. In Fig 5.2, for $j \neq 0$, maximum of θ_j is 79 at j = 455. In Fig 5.3, we show the cross correlation of this concatenated sequence with another concatenated sequence formed by concatenating two *m*-sequences formed by the primitive polynomial given by the octals [2157] and [2773].

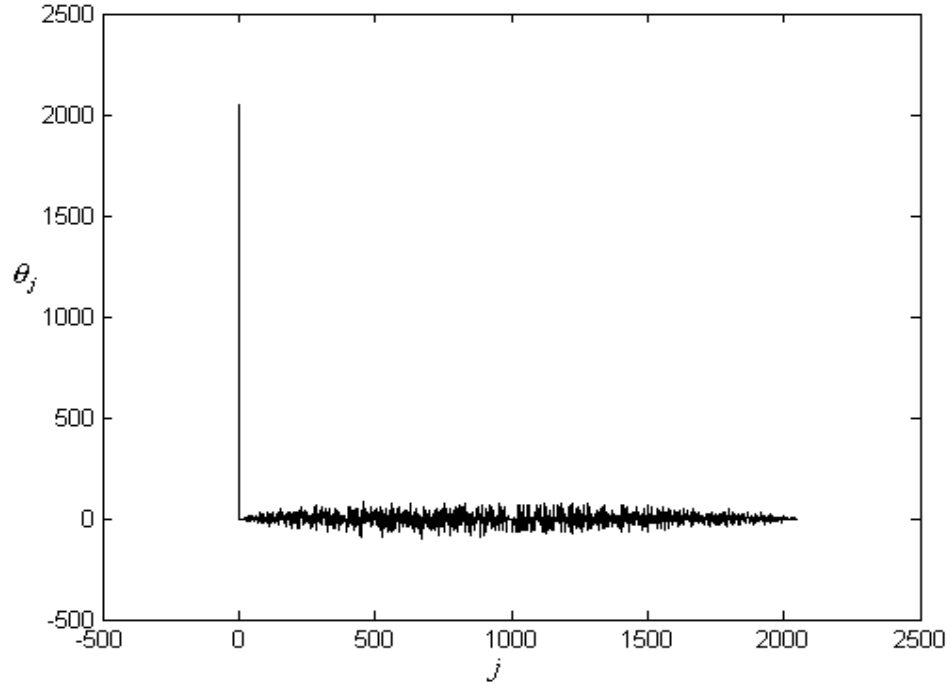


Fig 5.2: Aperiodic autocorrelation of a concatenated *m*-sequences

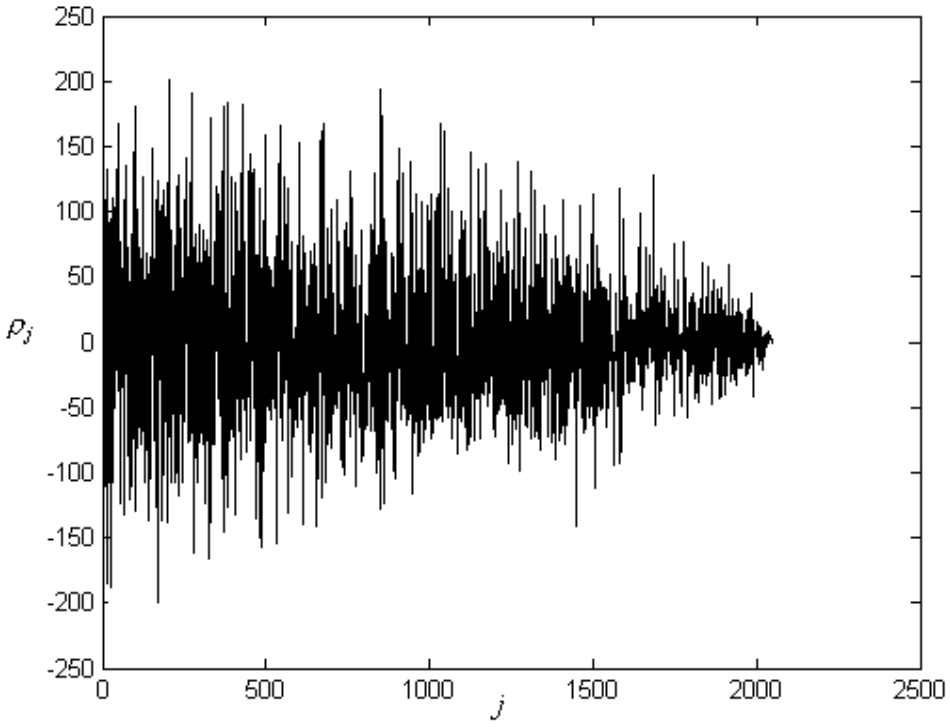


Fig 5.3: Aperiodic cross correlation of two concatenated *m*-sequences

We observe that the aperiodic cross correlation of two concatenated *m*-sequences are bounded. We proceed to find the bound on the concatenated *m*-sequences.

Using the general formulation of the bound given by Sidelnikov [29] where the cross correlation between two sequences denoted by $\eta(Wf(x))$, is given by

$$\eta\left(W\left(f\left(x\right)\right)\right) \leq \left(t - \frac{n}{q-1}\right)\sqrt{q} + \frac{n}{q-1} \qquad (5.3)$$

where $W(f(x))$ is the polynomial generated by the primitive polynomial $f(x)$. $f(x)$ has elements α, α^2,… …, α^t, of the field $GF(q)$ where $q = p^l$, l can be any integer and p is a prime number.

The cross correlation bound of the concatenated sequence in (5.3) can be written as

$$\eta(W(f(x)) \leq (r-2)\sqrt{2^r} + 2 \qquad (5.4)$$

where r is the number of shift registers used to generate the *m*-sequence. The derivation of (5.4) can be found in Appendix D.

To prove the validity of (5.4), different r's are used to generate different lengths of *m*-sequences. The theoretical and simulated bound is presented in table 5.1. The generator polynomials for different degrees can be obtained from [26].

Degree	Theoretical Bound / $\eta(Wf(x))$	Simulation
5	18.97	18
6	34	30
7	58.6	58
8	98	78
9	160.39	106
10	258	204

Table 5.1: Results comparing the theoretical bound and the simulated results for sequences with the same length

5.3 COMPARISON OF THE APERIODIC CORRELATION PROPERTIES

After introducing the concatenated *m*-sequences, a comparison of the aperiodic correlation properties between an *m*-sequence, concatenated *m*-sequences, Gold sequences and CEC sequences summarized in Table 5.2 In the comparison, both the *m*-sequence and Gold sequence have a length of 2047 while the concatenated *m*-sequence has a length of 2046. The CEC sequence that has an offset of 3 is used and its length is 2176.

Sequence	Length	Aperiodic autocorrelation Highest Peak for $j \neq 0$	Highest Peak to Peak ratio	Max Cross Correlation Value
m-sequence	2047	46	0.022	156
Concatenated *m*-sequence	2046	210	0.102	204
Gold sequence	2047	88	0.043	175
CEC sequence with offset 3	2176	320	0.147	320

Table 5.2: Table summarizing the aperiodic correlation properties of all the sequences

Looking at Table 5.2, we observed that CEC sequence possess the Highest Peak to Peak ratio. Moreover the maximum cross correlation between CEC sequences yield distinct peaks. This is shown in Fig 4.7. Therefore using CEC sequences for synchronization of Node Bs in WCDMA TDD mode might be inaccurate. This was elaborated using Fig 4.9.

The concatenated *m*-sequence possesses better cross correlation than a CEC sequence and has much better Highest Peak to Peak ratio compared to the CEC sequence. Although both an *m*-sequence and a Gold sequence has lower aperiodic autocorrelation and cross correlation values than the concatenated *m*-sequences, the number of available concatenated *m*-sequences is more. Moreover the cross correlation of the concatenated *m*-sequence is bounded given by (5.4). Therefore concatenated *m*-sequences are a suitable form of synchronization burst used to synchronize the Node Bs.

CHAPTER 6

CONCLUSION AND FUTURE WORKS

After we have looked at the different synchronization codes for inter-base synchronization in WCDMA TDD mode of communications in the previous chapters, we shall now draw conclusions in section 6.1 and make some recommendations for future works in section 6.2.

6.1 CONCLUSION

In this thesis, we have looked at the primary differences between FDD and TDD mode of WCDMA, one of the few 3G systems adopted by UMTS. We have also discussed the superiority of TDD over FDD mode of communication [8], [9], [11]. Moreover the importance of inter-base station synchronization in WCDMA TDD mode is also highlighted. Subsequently, we also studied how base stations using WCDMA TDD mode of transmission are being synchronized. Although using the GPS is the fastest and easiest method to synchronize all the base station within an RNC, 3GPP hopes to achieve a totally independent and self-reliant system. Moreover, GPS receivers only works when they have good visibility with the space borne satellites, hence, the receiver will not operate accurately under densely urban and indoor environments. Thus synchronization of the base stations via air-interface through the PRACH has been proposed by 3GPP [12], [13], [20] and using CEC

sequences with multiple code offset as synchronization codes [12], [15] were proposed for synchronization purposes.

In the thesis, the correlation properties of CEC sequences with multiple code offsets are presented. Although the usage of binary complementary sequences with different multiple code offsets reduces the number of Golay complementary sequences used in an RNC, we observed a distinct peak appearing in the aperiodic cross correlation function and hence CEC sequences with different multiple offsets should not be used by neighbouring RNCs otherwise interferences will occur when a Node B locks onto the cross correlation peak obtained by correlating with the synchronization code of its adjacent cell controlled by another Node B and sends an erroneous report to the RNC. This is explained with reference to Fig 4.9.

Subsequently, a new set of concatenated sequences, formed by concatenating two *m*-sequences is developed and their aperiodic correlation properties are presented. In 5.3, we do a comparison between the aperiodic correlation properties of an *m*-sequence, concatenated *m*-sequences, CEC sequences with multiple code offsets and a Gold sequence with nearly similar length. We observed that the aperiodic autocorrelation of concatenated sequence has a relatively lower highest peak-to-peak ratio than CEC sequences with multiple offsets. Moreover no distinct peaks appear in the cross correlation of the concatenated sequences. A bound for the cross correlation between concatenated sequences is given in (5.4). As the concatenated *m*-sequence increases the number of available sequences, it can be considered as a possible alternative to the CEC sequences with multiple code offsets, as a synchronization cell burst.

6.2 RECOMMENDATIONS FOR FUTURE WORKS

For future work, new codes with better aperiodic autocorrelation and cross correlation function can be developed. A synchronization code should have zero values for a few chips next to the main peak in order to prevent self-interference, and its cross correlation should be as low as possible to prevent interference to other Node Bs. Moreover codes with better security features can also be developed. However in the design of new codes, the length of the code must be equal or less than 2400 chips and the usage of this new codes must not result in a receiver structure which is too complex, which in turn would increase the cost of the receiver.

A possibility would be to explore the use of polyphase complementary sequences for the synchronization of the Node Bs. Firstly there are many available polyphase complementary sequences and another advantage is the availability of low costs of Read Only Memory (ROM) that can be used to store such sequences with a length of 2400 chips or less. As the number of possible Golay complementary sequences of length 1024 would be $(1024)! \times 2^{1023}$, using (3.6), considering different phases for the Golay complementary sequence of this length would greatly increase the number of available synchronization codes. Moreover the Enhanced Golay Correlator could be used to detect the polyphase complementary sequences. Thus it is a code worth exploring.

REFERENCES

Internet References

1. "Evolution of Mobile Communications"; available at www.iec.org/online/tutorials/umts/topic01.html
2. "GSM – The Wireless Evolution"; available at http://www.gsmworld.com
3. " Glossary for Mobile Networks "; available at http://www.monetmobile.com/assets/Glossary.pdf
4. "Enhanced Data rates for GSM Evolution", available at http://www.nuntius.com/solutions24.html
5. " 3GSM", available at http://www.ida.gov.sg/
6. " 3G – Third Generation Wireless System"; available at http://www.info.gov.hk/digital21/eng/knowledge/3g.html
7. " 3G Wireless"; available at http://www.commsdesign.com/
8. "The advantage of TDD over FDD in Wireless Data Applications"; available at http://www.three-g.net/3g_tdd_advantages.pdf
9. "Tutorial on TDD Systems"; available at http://www.fcc.gov/ realaudio/mt120301.pdf
10. "3G Technology", available at http://www.three-g.net/3g_technology.html
11. "The FDD/TDD DEBATE - frequency division duplexing vs. time division duplexing for fixed wireless networks - Technology Information", available at http://www.findarticles.com/p/articles/mi_m0GTV/is_4_17/ai_60010601

Papers and Journals

12. 3G TR 25-836, “Node B Synchronization for TDD”, Release 4; available at http://www.3gpp.org/
13. 3GPP TR 25-402, “Synchronization in UTRAN Stage 2”, Release 5; available at http://www.3gpp.org/
14. 3GPP TSG-RAN R1-99-g42, “Synchronization of Node B’s in TDD via selected PRACH timeslots”, source: Siemens; available at http://www.3gpp.org/
15. 3GPP TSG RAN R1-00-1351, “CEC sequences with multiple offsets for Node B sync in UTRA TDD”, source: Siemens, Mitsubishi Electric; available at http://www.3gpp.org/
16. 3G TS 25.211, “Technical Specification Group Radio Access Network: Physical Channels and Mapping of Transport Channels onto Physical Channels (FDD)”; available at http://www.3gpp.org/
17. 3G TS 25.221, “Technical Specification Group Radio Access Network: Physical Channels and Mapping of Transport Channels onto Physical Channels (TDD)”; available at http://www.3gpp.org/
18. 3G TS 25.223, “Technical Specification Group Radio Access Network: Spreading and modulation (TDD)”; available at http://www.3gpp.org/
19. B. Hofmann-Wellenhof, H. Lichtenegger and J. Collins, “Global Positioning System: Theory and Practice”, Spring-Verlag Wien New York, 1992
20. 3GPP TR 25-838, “Node B Synchronization for TDD (Iub/Iur aspects)”, Release 4; available at http://www.3gpp.org/
21. 3GPP TR 25-105, “BS Radio transmission and Reception (TDD)”, Release 5; available at http://www.3gpp.org/

22. 3G TS 25.224, “Technical Specification Group Radio Access Network: Physical Layer Procedures (TDD)”, Release 5; available at http://www.3gpp.org/
23. M.J.E. Golay, “Complementary Series”, *IRE Trans. Information Theory*, vol. IT-7, pp. 82-87, April 1961
24. S. Budisin, “Golay Complementary Sequences are superior to PN Sequences”, *IEEE International Conference*, pp. 101-104, Sep1992
25. S. Budisin, “New Complementary Pairs of Sequences”, *IEEE Electronics Letters*, Vol 26, No. 13, pp. 881-883, June 1990,
26. Roger L. Peterson, Rodger E. Ziemer & David E. Borth, “Introduction to spread-spectrum communications”, Englewood Cliffs, NJ: Prentice Hall, 1995
27. S. W. Golomb, “Shift-register sequences and spread-spectrum communications”, *IEEE Third International Symposium on Spread Spectrum Techniques and Applications*, Finland, pp. 14-15, July 1994
28. R.Gold, “Maximal Recursive Sequences with 3-valued recursive Cross-Correlation Functions”, *IEEE Tans. Information Theory (Correspondence)*, vol. IT-14, pp.154-156, January 1968
29. V.M. Sidelnikov, “On Mutual Correlation of Sequences”, *Soviet Math. Dokl*, Vol. 12, pp 197-201, 1971

APPENDIX A

GOOD CORRELATION PROPERTIES OF A SYNCHRONIZATION CODE

Assume the transmitted signal is *a(t)c(t)*, where *a(t)* is the modulation signal and *c(t)* is spreading code waveform. And the reference signal at the receiver is *c(t)*, with period of *T* and possessing the following autocorrelation properties,

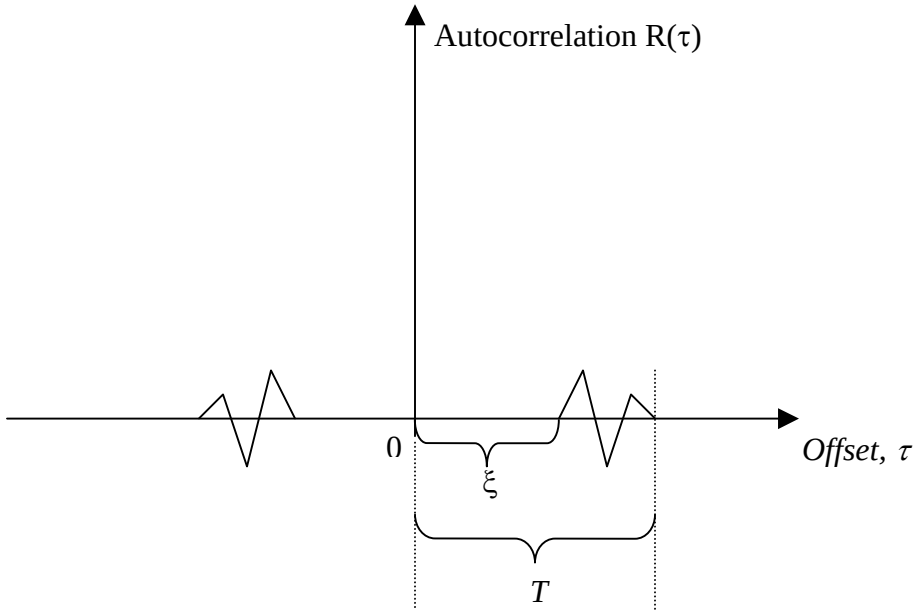


Fig A1: Autocorrelation Property of *c(t)*

If the propagation channel is multipath channel, and *a(t)c(t)* and *a(t-τ) c(t-τ)* are received in the receiver, then the demodulation of the received signal is

$$\begin{aligned}
&\int_0^T [a(t)c(t) + a(t-\tau)c(t-\tau)]c(t)dt \\
&= \int_0^T a(t)c(t)c(t)dt + \int_0^T a(t-\tau)c(t-\tau)c(t)dt \\
&= a(t) + a(t-\tau)\int_0^T c(t-\tau)c(t)dt \\
&= a(t) \qquad for\ |\tau| < \xi
\end{aligned} \tag{A1}$$

In the above equation, we use the fact that, $\int_0^T c(t-\tau)c(t)dt = 0,\ for\ |\tau| < \xi$

Therefore, as long as the signal delay fulfills the relationship $|\tau| < \xi$, the delayed version of the transmitted signal will not lead to self-interference.

APPENDIX B

RECURSIVE ALGORITHM TO GENERATE COMPLEMNETARY SEQUENCES

In [25], Budisin gives the following algorithm to generate complementary sequences.

$$\begin{aligned} a_0(i) &= \delta(i) \\ b_0(i) &= \delta(i) \\ a_n(i) &= a_{n-1}(i) + W_n b_{n-1}(i - 2^{P_n}) \\ b_n(i) &= a_{n-1}(i) - W_n b_{n-1}(i - 2^{P_n}) \end{aligned} \tag{B1}$$

where a_n and b_n are two complementary sequences, each of length $N = 2^n$, in the nth iteration, $\delta(i)$ is the Kronecker delta function, $i = 0,1,\ldots, N\text{-}1$, $n = 0,1,\ldots, N\text{-}1$ and W_n is an arbitrary complex number of unit magnitude. P_n is any permutation of numbers from 0 to N-1 [25].

To illustrate, we will do 2 iterations (n = 2). Assume P_n = [0, 1] (since P_n is any permutation of numbers from 0 to 3) and W_n = [1, 1]. This means the length of the complementary sequence is 2 in the first iteration, and 4 after second iteration. Thus in the first iteration (n = 1), P_1 = 0 and W_1 = 1. Since there are two elements for both a_1 and b_1, i exist from 0 to 1. Therefore

$$\begin{aligned} a_0(i) &= \delta(i) \\ b_0(i) &= \delta(i) \\ a_1(i) &= a_0(i) + W_0 b_0(i-1) = \delta(i) + \delta(i-1) \\ b_1(i) &= a_0(i) - W_0 b_0(i-1) = \delta(i) - \delta(i-1) \end{aligned} \tag{B2}$$

To illustrate it graphically, see Fig B1.

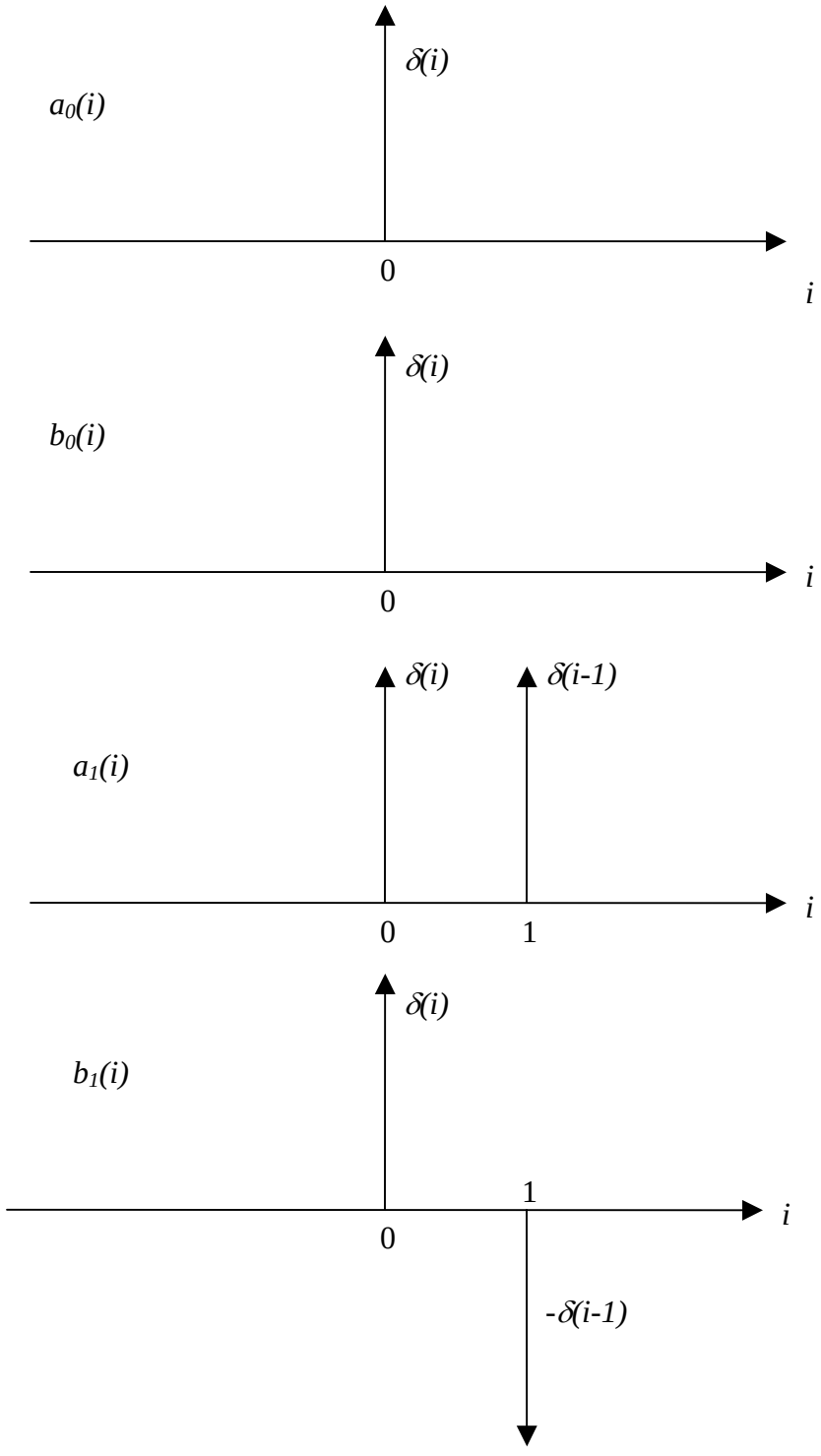


Fig B1: Graphical representation of the of 1st iteration

Therefore the complementary sequences after the first iterations yields

$$\begin{aligned} a_1 &= [1 \;\; 1] \\ b_1 &= [1 \;\; -1] \end{aligned} \tag{B3}$$

In the second iteration, $P_1 = 1$ and $W_1 = 1$. Since there are four elements for both a_2 and b_2, i exist from 0 to 3.

$$\begin{aligned} a_2(i) &= a_1(i) + W_1 b_1(i-2) = a_1(i) + b_1(i-2) = \delta(i) + \delta(i-1) + \delta(i-2) - \delta(i-3) \\ b_2(i) &= a_1(i) - W_1 b_1(i-2) = a_1(i) - b_1(i-2) = \delta(i) + \delta(i-1) - \delta(i-2) + \delta(i-3) \end{aligned} \tag{B4}$$

The graphical representation is given in Fig B2.

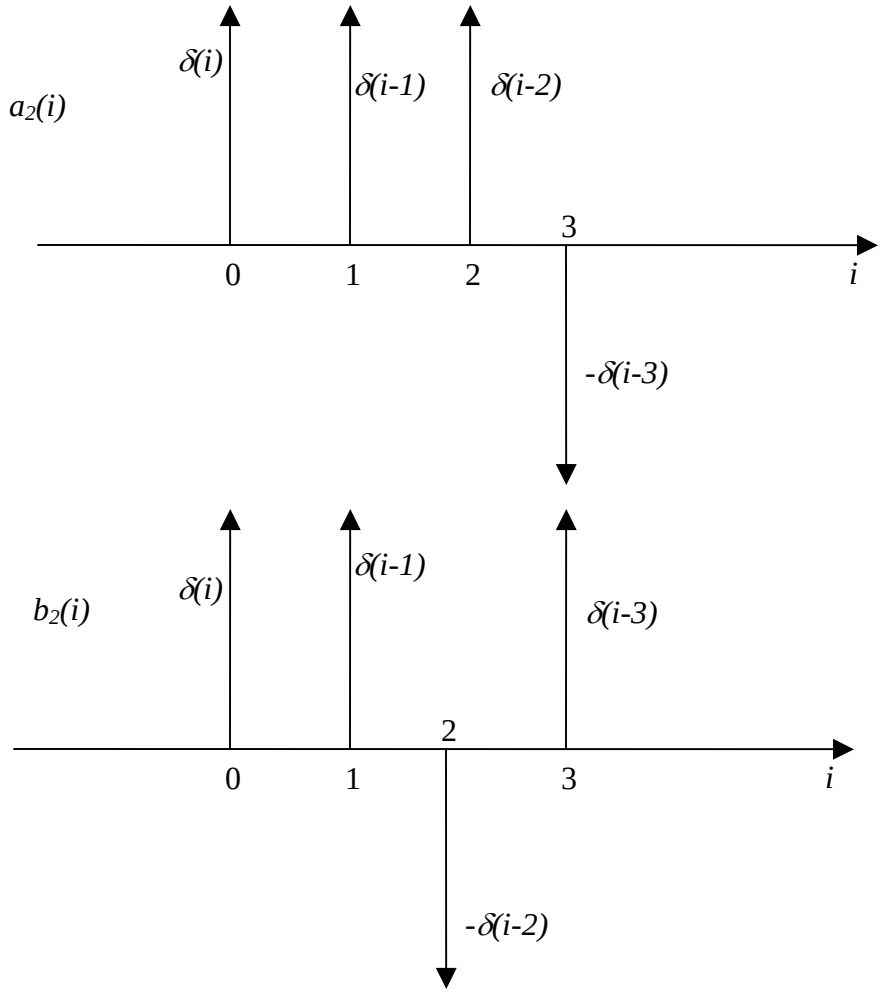


Fig B2: Graphical representation of the of 2nd iteration

Therefore the complementary sequences after the second iterations yields

$$a_2 = [1\ \ 1\ \ 1\ \ -1]$$
$$b_2 = [1\ \ 1\ \ -1\ \ 1] \qquad \text{(B5)}$$

The iteration can go on to form longer complementary sequences. For Example the third iteration is given in Fig B3.

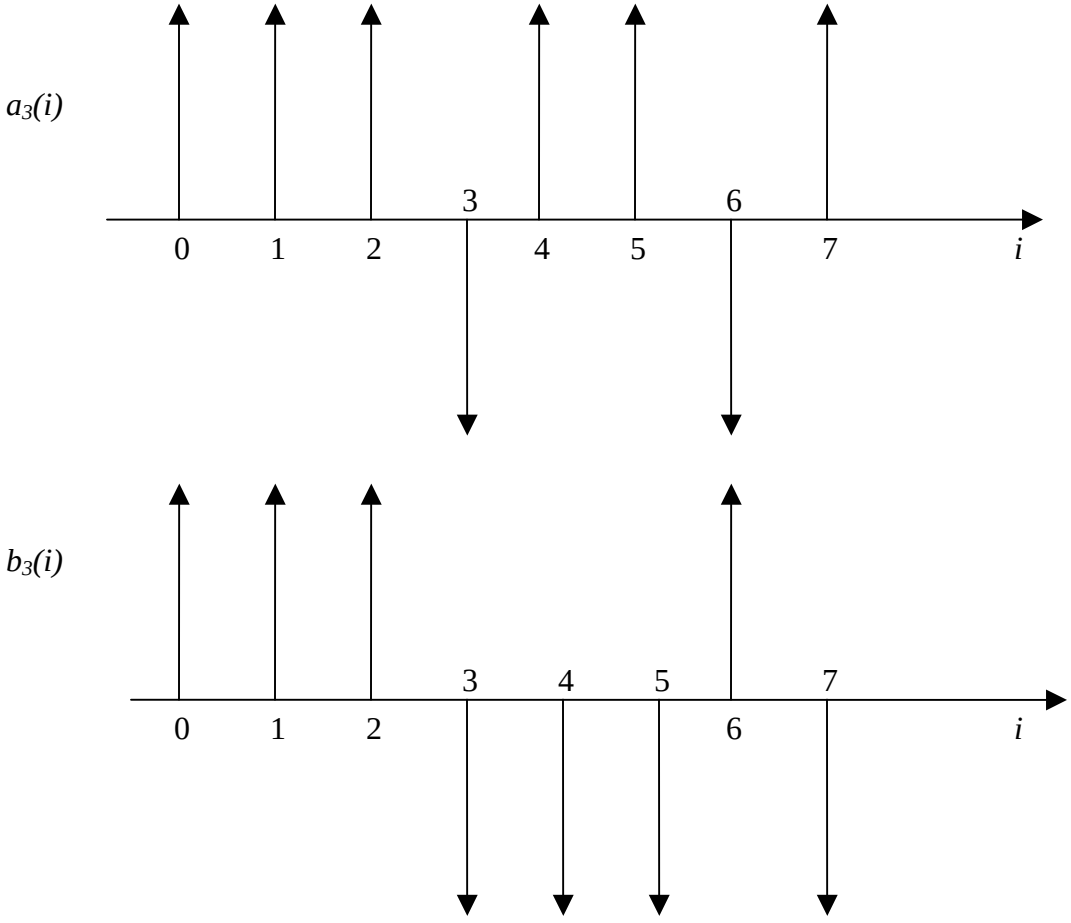


Fig B3: Graphical representation of the of 3rd iteration

Therefore the complementary sequences after the third iterations yields

$$a_3 = [1\ \ 1\ \ 1\ \ -1\ \ 1\ \ 1\ \ -1\ \ 1]$$
$$b_3 = [1\ \ 1\ \ 1\ \ -1\ -1\ -1\ \ \ 1\ -1] \qquad \text{(B6)}$$

APPENDIX C

EULER FUNCTION AND PROPERTIES

The following is obtained from http://www.cut-the-knot.com/blue/Euler.html

The Euler's function φ for integer m, $\varphi(m)$, is defined as the number of positive integers not greater than and coprime to *m*. Two integers are coprime if they share no common positive factors (divisors) except 1. A few first values: $\varphi(1) = 1$, $\varphi(2) = 1$, $\varphi(3) = 2$, $\varphi(4) = 2$, $\varphi(5) = 4$, $\varphi(6) = 2$, $\varphi(7) = 6$, $\varphi(8) = 4$, $\varphi(9) = 6$, $\varphi(10) = 4$, $\varphi(11) = 10$, $\varphi(12) = 4$, $\varphi(13) = 12$, and so on.

For example 8 is coprime with 1, 3, 5 and 7, so $\varphi(8) = 4$ and 9 is coprime with 1, 2, 4, 5, 7 and 8, thus $\varphi(9) = 6$.

Property 1

The first property of Euler's function, for prime p, $\varphi(p) = p - 1$.

Property 2

The Euler's function, φ, is multiplicative which means that $\varphi(m_1m_2) = \varphi(m_1)\ \varphi\ (m_2)$ for coprime m_1 and m_2 or g.c.d $(m_1,m_2) = 1$.

For example $\varphi(12) = \varphi(3)\ \varphi\ (4) = 2 \times 2 = 4$ and **NOT** $\varphi(12) = \varphi(2)\ \varphi\ (6) = 1 \times 2 = 2$

APPENDIX D

CORRELATION BOUND

Using the general formulation of the bound given by Sidelnikov [29] where the cross correlation between two sequences denoted by $\eta(Wf(x))$, is given by

$$\eta\left(W\left(f\left(x\right)\right)\right) \leq \left(t - \frac{n}{q-1}\right)\sqrt{q} + \frac{n}{q-1} \qquad \text{(D1)}$$

where $W(f(x))$ is the polynomial generated by the primitive polynomial $f(x)$. $f(x)$ has elements α, α^2,… …, α^t, of the field $GF(q)$ where $q = p^l$, l can be any integer and p is a prime number.

We consider two m-sequences generated by polynomials of degree j and k. Then the length of the entire sequence $n = 2^j + 2^k - 2 = L$ (5.1).

Let $W(f_1(x))$ and $W(f_2(x))$ be the polynomials generated by the primitive polynomial $f_1(x)$ and $f_2(x)$ whose degree are j and k respectively, in the field $GF(p^j)$ and $GF(p^k)$. If $j > k$, the elements of $f_2(x)$ will also be present in the field $GF(p^j)$. So let $f_1(x)$ and $f_2(x)$ be represented mathematically by

$$\begin{aligned} W(f_1(x)) &= a_A x^A + a_{A-1}x^{A-1} + a_{A-2}x^{A-2} + \ldots\ldots\ldots + a_1 x^1 \\ W(f_2(x)) &= b_B x^B + b_{B-1}x^{B-1} + b_{B-2}x^{B-2} + \ldots\ldots\ldots + b_1 x^1 \end{aligned} \qquad \text{(D2)}$$

with $A > B$ where A and B are the numbers of elements of the sequences given in (5.1) and a_A, a_{A-1}, ……, a_1 and b_B, b_{B-1}, ……, b_1 are coefficients in the field $GF(p^j)$.

When both sequences are concatenated with $f_2(x)$ being attached to $f_1(x)$, the new sequence, $W(f_3(x))$, formed can be mathematically represented as

$$\begin{aligned} W(f_3(\alpha)) &= x^B\left(a_A x^A + a_{A-1}x^{A-1} + a_{A-2}x^{A-2} + \ldots\ldots\ldots + a_1 x^1\right) + b_B x^B + b_{B-1}x^{B-1} + b_{B-2}x^{B-2} + \ldots\ldots\ldots + b_1 x^1 \\ &= a_A x^{A+B} + a_{A-1}x^{A+B-1} + a_{A-2}x^{A+B-2} + \ldots\ldots\ldots + b_B\alpha^B + b_{B-1}\alpha^{B-1} + b_{B-2}\alpha^{B-2} + \ldots\ldots\ldots + b_1\alpha^1 \end{aligned} \quad \text{(D3)}$$

This new sequence will have elements in the field $GF(p^j)$.

Using (5.1), $A + B = L$, which is the total length of the new sequence and the sequences used for concatenation are generated by primitive polynomial with degree c and d, hence the cross correlation bound of the concatenated sequence can be written as

$$\eta\left(W\left(f\left(x\right)\right)\right) \le \left(r - \frac{L}{p^r - 1}\right)\sqrt{p^r} + \frac{L}{p^r - 1} \quad \text{(D4)}$$

where $r = \max[c,d]$.

For the case when both j and k are similar, $\frac{L}{r-1} = \frac{2^j + 2^k - 2}{r-1} = \frac{2^j + 2^j - 2}{2^j - 1} = 2$ and since we are dealing with binary sequences, $p = 2$, hence the cross correlation bound in (C4) reduces to

$$\eta(W(f(x)) \le (r-2)\sqrt{2^r} + 2 \quad \text{(D5)}$$

APPENDIX E

DETAILS OF PUBLICATIONS

In the process of doing the M.Eng project, the following papers are being published

1. Y.K. Khit and T. T. Tjhung, "Design of Concatenated Sequences for Inter-Base Synchronization in WCDMA TDD mode", *9th IEEE International Conference on Telecommunications 2002 (ICT 2002)*, vol. 2, pp 197-201, Beijing (China), Jun 2002
2. Y.K. Khit and T. T. Tjhung, "Concatenated Sequences for Inter-Base Station Synchronization in WCDMA TDD mode", *8th IEEE International Conference on Communication Systems 2002 (ICCS 2002)*, Singapore, Nov 2002
3. Y.K. Khit, T. T. Tjhung and Y. Wang, "Sequences for Inter-Base Synchronization in WCDMA TDD mode", *Fourth International Conference on Information, Communications & Signal Processing 2003 (ICICS 2003)*, Singapore, Dec 2003