

CHALMERS



Noncoherent Equalizer for DECT System

Master of Science Thesis in the Programme Integrated Electronic System Design

XIAOYU TENG
WEICHAO ZHANG

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden, May 2010

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Noncoherent equalizer for DECT system

Xiaoyu Teng
Weichao Zhang

© Xiaoyu Teng, May 2010.
© Weichao Zhang, May 2010.

Examiner: Lars Svensson

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Cover picture:

A typical strong reflective environment, see detailed information about fading channel in Chapter 3.

Department of Computer Science and Engineering
Göteborg, Sweden May 2010

Abstract

Digital European Cordless Telecommunication (DECT), as a standard for cordless indoor wireless communication in Europe, is widely used in hospitals, offices and factories. But its performance can be constrained by channel dispersion, and this can limit the use of DECT system in reflective environments. In this master thesis project, non-coherent equalization schemes are investigated. Different equalization algorithms are simulated in Matlab and a least mean square (LMS) algorithm based baseband equalizer is implemented in a field programmable gate array (FPGA). But this method is proved insufficient to solve the severe multipath fading problem. Another type of equalizer called passband equalizer is stated afterwards, and a required bit error rate (BER) can be achieved in multipath fading channels up to 500 ns delay spread.

This project includes both simulation and hardware implementation. In this thesis, simulations of two types of equalization methods (LMS baseband and passband equalizer) and implementation of the LMS baseband equalizer in FPGA are introduced. The challenge of future work and its possible refinement are also analyzed.

Key word: DECT, non-coherent equalizer, LMS algorithm.

Acknowledgement

We would like to thank the hardware section of ASCOM (Sweden) AB for giving us this opportunity to work in such an interesting project. We would also like to thank the manager Thomas Harju and our supervisor Mikael Nyström for their generous support. Moreover, we would like to thank our examiners from Chalmers University of Technology, Mats Viberg and Lars Svensson, for their valuable feedback and proof reading on this master thesis.

This thesis work has been performed at ASCOM in Gothenburg from March to July, 2009.

About ASCOM Wireless Solutions:



ASCOM Wireless Solutions leads the way in delivering value for customers by providing them with competitive solutions for wireless on-site communication to support and optimize their mission-critical processes.

Table of Content

1.	Introduction	1
1.1	Background	1
1.2	Problem Description.....	1
1.3	Thesis Outline.....	1
1.4	List of Acronyms.....	2
2.	DECT Systems Description.....	4
2.1	Protocol and Packet Format.....	4
2.2	Transceiver Scheme.....	5
3.	Mobile Radio Channel and Project Specifications	7
3.1	Path loss, Large Scale Fading and Small Scale Fading	7
3.2	Impulse Response of the Wireless Radio Channel.....	8
3.3	Parameters of Mobile radio channel.....	10
3.4	Limitations of This Project.....	12
4.	Modern Electronic Technologies of Digital Signal Processing.....	16
4.1	Digital Signal Processors (DSPs).....	16
4.2	Application Specific Integrated Circuit (ASIC)	17
4.3	Field Programmable Gate Array (FPGA).....	18
5.	FIR Filter and Equalizer Techniques.....	22
5.1	Intersymbol Interference	22
5.2	Digital LTI Filter and Adaptive Filter	22
5.3	Baseband Equalization	24
5.4	Passband Equalization and Fractionally Spaced Equalization	30
6.	System Design and Simulation Result Analysis	31
6.1	Overview of the Design.....	31

6.2	AD Conversion.....	31
6.3	Demodulation Method Simulation and Result analysis.....	33
6.4	LMS Baseband Equalizer and Result Analysis	43
6.5	Limitation of LMS Baseband Equalization for Non-coherent Demodulation.....	49
7.	Hardware Implementation and Verification.....	52
7.1	External PCB Board	52
7.2	FPGA Development Board.....	57
7.3	Basic DSP Arithmetic Implementation on FPGA	59
7.4	Band-Pass Filter and Lowpass Filter	61
7.5	Demodulation	64
7.6	Synchronization.....	65
7.7	LMS Equalizer.....	66
7.8	Resource Utilization	69
7.9	Simulation and Verification	69
8.	Passband Equalization and Analysis of the Result.....	72
8.1	Potential Methods for Improving the Equalizer	72
8.2	Passband Equalization	72
8.3	Performance of Passband Equalization	74
9.	Conclusions and Suggestions for Future Work.....	79
9.1	Conclusion.....	79
9.2	Hardware Implementation Analysis	80
9.3	Issues for Passband Equalizer Implementation	81
10.	References	82

1. Introduction

1.1 Background

As a low cost and reliable digital voice telecommunication system, DECT can provide cordless communications for high traffic density, short range communications, and covers a broad range of applications and environments [1].

But with the popularity of DECT increasing, the system can also run into problems in certain environments. For example, the DECT system is very sensitive to the environments like steel constructed workshops, in which strong reflection of microwaves can occur. Unlike Global System for Mobile communication (GSM), the DECT standard does not have any advanced equalization technologies to fix this multipath propagation problem due to its low cost strategy. As the desire of improving DECT system performance strongly grows and the cost of electronic devices like FPGA and digital signal processors (DSPs) decreases, some simple and efficient methods can be implemented in the DECT system.

Some techniques have already been used to fix multipath fading problems in wireless communications, such as increasing the complexity of the modulation scheme, space diversity and equalization. Previously, the antenna diversity has been integrated to the handsets and base stations of the DECT system [2]. Apparently, the performance can be improved to a certain level. But in order to obtain higher voice quality and data transmission reliability, some other techniques need to be investigated.

The initiator of the project, ASCOM Sweden AB, decided to explore equalization methods for DECT. Thereby, ASCOM Sweden AB started this thesis in March 2009 in cooperation with the Signal and Systems and Computer Engineering departments at Chalmers to investigate the DECT receiver system.

1.2 Problem Description

Due to its good performance and low cost in terms of implementation, the non-coherent receiver has been a very attractive method applied in wireless transceiver design. Equalization is an efficient technique to recover the desired signals from the Inter-symbol Interference (ISI). The purpose of this thesis is to improve the performance of the DECT system in severe ISI environments on the premise of low power cost by investigating a feasible equalization algorithm and integrating it into the non-coherent receiver.

1.3 Thesis Outline

This report consists of 10 chapters.

Chapter 1 is a brief introduction of the project.

Chapter 2 gives an overview of the DECT system.

Chapter 3 gives a general description of the multipath channel.

Chapter 4 states modern electronic techniques for DSP.

Chapter 5 introduces different equalization structures for ISI elimination.

Chapter 6 focuses on the design of an LMS baseband equalizer and analysis of its performance.

Chapter 7 introduces hardware implementation of an LMS baseband equalizer.

Chapter 8 is devoted to the design of a passband equalizer.

Chapter 9 provides the conclusion and some prospective future work.

Chapter 10 lists the references.

1.4 List of Acronyms

Abbreviations

ASIC	Application-Specific Integrated Circuit
CMOS	Complementary Metal-Oxide-Semiconductor
DECT	Digital Enhancement Cordless Telecommunications
DFE	Decision Feedback Equalizer
DSP	Digital Signal Processing
ETSI	European Telecommunications Standards Institute
FDMA	Frequency Division Multiple Access
FPGA	Field Programmable Gate Array
GFSK	Gaussian Frequency Shift Keying
GMSK	Gaussian Minimum Shift Keying
GSM	Global Systems of Mobile communications
ISI	Inter-Symbol Interference
LPF	Low Pass Filter
LMS	Least Mean Square

MLSE	Maximum Likelihood Sequence Estimator
PCB	Printed Circuit Board
RLS	Recursive Least Square
TDMA	Time Division Multiple Access
TDD	Time Division Duplex
VCO	Voltage Controlled Oscillator

2. DECT Systems Description

2.1 Protocol and Packet Format

DECT, also known as Digital European Cordless Telephones, is a standard developed by European Telecommunications Standards Institute (ETSI) and was finalized in 1992 [1].

DECT provides personal communications with cordless indoor communication systems instead of the wired solution. An in-building Private Branch Exchange (PBX) or the Public Switched Telephone Network (PSTN) is used to supply the connection among the headsets [3]. As it is defined in the DECT standard, DECT can provide low power radio access between handsets and fixed base stations at a range of up to a few hundred meters.

TDMA/FDMA/TDD is employed in DECT physical layer. There are 12 pairs of time slots in one frame of 10ms length. The first 12 of them are used for uplink, and the rest are used for downlink. The channel bandwidth is 1738 kHz which is 1.5 times the data rate of 1152 kbps according the specification of the system. The DECT system occupies the radio spectrum from 1880 MHz to 1900 MHz and 10 carrier frequencies are allocated within this spectrum. Each carrier can handle an entire frame including both uplink and downlink which is different from the GSM standard.

Each time slot consists of 480 bits, where the first 32 bits and the last 60 bits are defined as preamble bits and guard time respectively, and the middle 388 bits are data. The structure of one frame is depicted in Figure 1.

480 bits		
Preamble	Data	Guard Time bits
32bits	388 bits	60 bits

Figure 1: The structure of one time slot.

The first 16 bits of preamble consisting of ‘1010101010101010’ are used for clock recovery and frequency offset compensation; the later 16 bits pattern ‘1110100110001010’ is the slot synchronization in digital field. The data sequence is made up by 388 bits including the payload and error detection part, but there is no error correction in this field.

2.2 Transceiver Scheme

2.2.1 Modulation Scheme

The modulation scheme applied in DECT is Gaussian frequency shift keying (GFSK) with modulation index of 0.5, which can also be accepted as a special case of Gaussian minimum shift keying (GMSK) commonly known from GSM. As a binary modulation scheme, the phase of GFSK is rotated by $\pm 90^\circ$ in one symbol duration to present the coming symbol as '1' or '0'. The merit of GFSK is its low cost in hardware and requiring low power amplifier design due to its constant envelope. Spectrum efficiency is another attractive point of GFSK. Since the digital sequence is shaped with a Gaussian low pass filter prior to the frequency modulator, the side lobe level is lowered, which reduces the interference from adjacent carriers effectively. Because of these two advantages, GFSK is widely used in personal wireless communications like DECT and Bluetooth.

The detailed characteristics about GFSK are described in [4] and [5]. In this project, only the implementation is presented. Normally, the GFSK signal is generated by passing a non-return-to-zero (NRZ) sequence through a Gaussian low pass filter followed by an FM-VCO. Another implementation method called quadrature baseband modulation is not used here due to its complexity. The mathematical expression of the Gaussian low pass filter can be written as (2.1), and the block diagram of the GMSK modulator is shown in Figure 2.

$$g(t) = \frac{1}{\sqrt{2\pi}\sigma T} \exp\left(-\frac{t^2}{2\sigma^2 T^2}\right) \quad (2.1)$$

where $\sigma = \sqrt{\ln 2} / 2\pi BT$, and T is used to denote the symbol duration.

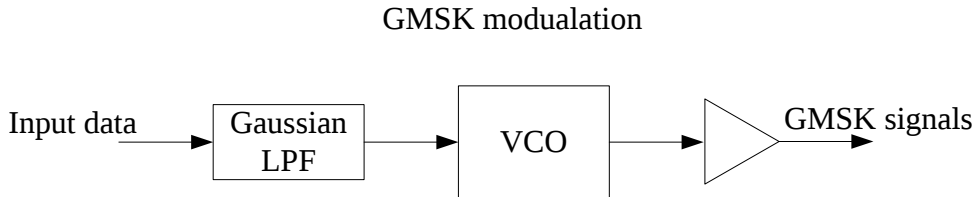


Figure 2: Block diagram of GMSK modulation. LPF: low pass filter, VCO: voltage-controlled oscillator

2.2.2 Demodulation Scheme

In a DECT receiver, it is common to use a quadrature detector to extract the baseband signals from intermediate frequency (IF) signals. The entire demodulator structure is shown in Figure 3. Compared to the coherent demodulation method, the non-coherent demodulation has a slightly higher BER [17]. However, the complexity of a non-coherent demodulator is much lower. There

are also some other demodulation schemes discussed in [1] and [6]. Due to the advantage of low cost, quadrature detector is the main demodulation method to be investigated in this project.

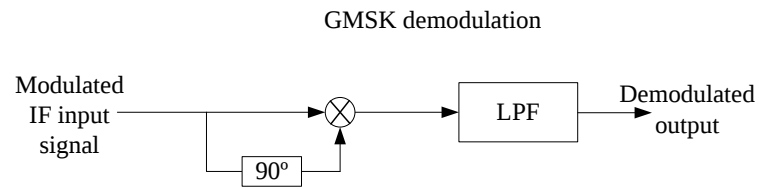


Figure 3: GMSK non-coherent demodulation.

3. Mobile Radio Channel and Project Specifications

3.1 Path loss, Large Scale Fading and Small Scale Fading

The mobile radio channel places fundamental limitations on the performance of wireless communications. Compared to the stationary and predictable wired channel, the wireless channel characteristic is varying with time between the mobile stations and handsets. Thereby, it is very difficult to analyze the wireless channel. In general, the variation of the channel can be attributed to two main categories, the large scale fading effects and the small scale fading effects [7].

Large scale fading effects mainly describe the attenuations of signal power caused by absorption, reflection, scattering and the dissipation of the power radiated between the base stations and mobile handsets. Usually they are called path loss and shadowing. Variations due to path loss occur over long distances (100~1000 meters). Shadowing is caused by obstacles between the transmitter and receiver, and it occurs over distances proportional to the length of the obstacles (10~100 meters).

Small scale fading effects refer to the channel variation caused by the constructive or destructive addition of multipath components. This usually happens over short distances, especially in indoor environments; hence, the multipath fading is the main factor to affect the communication quality of DECT. Figures 4 and 5 illustrate the multipath fading environment and its mathematical model.

Multipath propagation and the speed of a mobile are two main factors influencing small scale fading effects. Multipath propagation, just as its name implies, means that the transmitted signals arrives at the receiver from different paths at slightly different times. This phenomenon, which is caused by reflection and scattering from the objects in the transmission paths, could introduce signal variations in both amplitude and phase when different path components combine at the receiver side. This can cause a severe fluctuation in the signal strength, and distort the original signals severely. Besides, the time dispersion caused by the multipath propagation, will introduce ISI to the received signals. If the time dispersion is severe, this cannot be accepted by the receiver. The motion between the base stations and handsets or the movement of objects in the transmission path can result in Doppler frequency shift, and the incoming waves would experience a random frequency modulation during the transmission. If the Doppler shift is relatively serious, the designed information cannot be extracted without additional techniques at the receiver side.

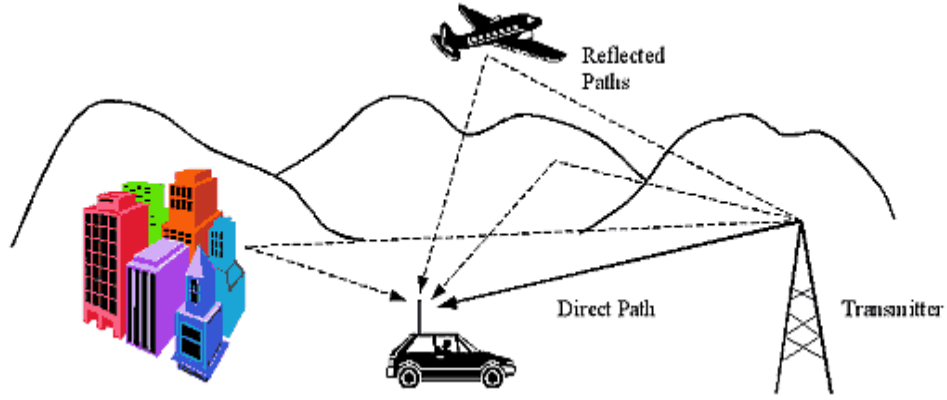


Figure 4: Multipath signals during the transmission.

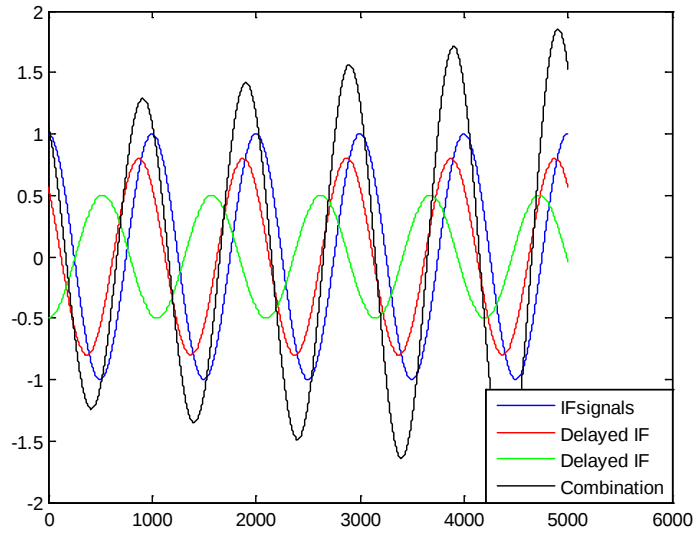


Figure 5: Simulation of multipath transmission.

3.2 Impulse Response of the Wireless Radio Channel

When a single pulse passes through a multipath channel, a pulse train will appear at the receiver with each pulse corresponding to the line-of-sight (LOS) or multipath delay components. So a linear finite impulse response (FIR) filter is quite intuitive to model the time invariant channel [2]. However, in reality, time variation is a feature of wireless radio channels. Thus, an FIR filter with time varying impulse response is more suitable to represent the radio channel. The multipath fading channel model is illustrated in Figure 6.

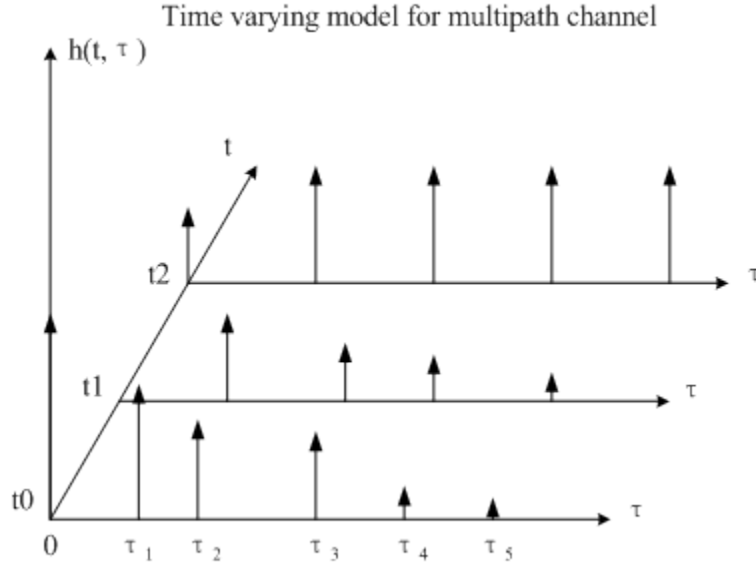


Figure 6: The time varying discrete model for multipath channels.

If there is no LOS component in the channel, like state t1 in Figure 6, the small scale fading envelope obeys a Rayleigh distribution given by

$$p(r) = \begin{cases} \frac{r}{\sigma^2} \exp(-\frac{r^2}{\sigma^2}) & (0 \leq r < \infty) \\ 0 & (r < 0) \end{cases} \quad (3.1)$$

where σ^2 is the time-average power of the received signals and r is the envelope of the received signals.

If a dominant LOS component exists, the envelope is Ricean distributed. The Ricean distribution is defined by

$$p(r) = \begin{cases} \frac{r}{\sigma^2} \exp(-\frac{r^2 + A^2}{\sigma^2}) I_0(\frac{Ar}{\sigma^2}) & (0 \leq r < \infty) \\ 0 & (r < 0) \end{cases} \quad (3.2)$$

Here, A is the peak amplitude of the dominant signal, whereas I_0 is the modified Bessel function of the first kind and order zero.

Usually, the Ricean distribution is specified by a parameter K , which is the ratio between the deterministic signal power and the variance of the multipath as described in (3.3).

$$K = \frac{A^2}{\sigma^2} \quad (3.3)$$

When K equals to 0, the Ricean distribution reduces to the Rayleigh distribution, which also means that the LOS disappears.

3.3 Parameters of Mobile radio channel

The most important characteristics of the channel, including power delay profile, coherence bandwidth, Doppler power spectrum and coherence time, are all derived from the channel autocorrelation and scattering functions.

Denoting the time varying channel impulse response by $c(\tau, t)$, the statistical characteristics of $c(\tau, t)$ are described by its autocorrelation function, given by:

$$A_c(\tau_1, \tau_2; t, \Delta t) = E[c^*(\tau_1; t)c(\tau_2; t + \Delta t)] \quad (3.4)$$

Moreover, when the channel is wide-sense stationary (WSS) channel with uncorrelated scattering (US), (3.4) can be compressed as:

$$E[c^*(\tau_1; t)c(\tau_2; t + \Delta t)] = A_c(\tau_1; \Delta t)\delta[\tau_1 - \tau_2] = A_c(\tau, \Delta t) \quad (3.5)$$

The power delay profile $A_c(\tau)$ is determined by the autocorrelation (3.5) with $\Delta t = 0$. The scattering function is defined as the Fourier transform of $A_c(\tau, \Delta t)$ with respect to Δt :

$$S_c(\tau, \rho) = \int_{-\infty}^{\infty} A_c(\tau, \Delta t)e^{-j2\pi\rho\Delta t}d\Delta t \quad (3.6)$$

a. Rms Delay Spread and Coherent Bandwidth

The mean delay spread μ_{Tm} and root mean square (rms) delay spread σ_{Tm} are defined in terms of the power delay profile $A_c(\tau)$ as:

$$\mu_{Tm} = \frac{\int_0^{\infty} \tau A_c(\tau) d\tau}{\int_0^{\infty} A_c(\tau) d\tau} \quad (3.7)$$

$$\sigma_{Tm} = \sqrt{\frac{\int_0^{\infty} (\tau - \mu_{Tm})^2 A_c(\tau) d\tau}{\int_0^{\infty} A_c(\tau) d\tau}} \quad (3.8)$$

where μ_{Tm} and σ_{Tm} are the mean and rms values of T_m , respectively. The rms delay spread can be used to roughly characterize the channel. Denoting the symbol period by T , when $T > \sigma_{Tm}$, the signals will experience negligible ISI. Conversely, the signals will experience significant ISI when $T < \sigma_{Tm}$.

This characteristic can also be obtained in the frequency domain by taking the Fourier transform of the power delay profile $A_c(\tau)$, resulting in $A_c(\Delta f)$. The frequency B_c , satisfying $A_c(\Delta f) \approx 0$ when $\Delta f > B_c$, is called the coherence bandwidth of the channel.

If the transmitted signal occupies a wider bandwidth compared to the channel coherence bandwidth, given by $B > B_c$, the spectrum magnitude of the signal will experience different fading in different frequencies. This phenomenon is called frequency selective fading. Conversely, if the signal bandwidth is lower than the coherence bandwidth ($B < B_c$), every part of the spectrum will experience similar fading, which is called flat fading. A comparison between frequency selective fading and flat fading is shown in Figure 7.

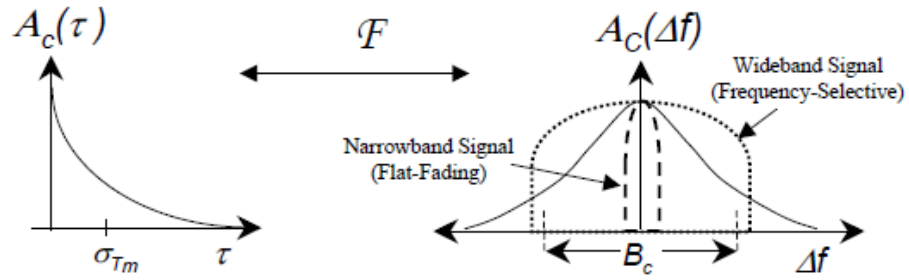


Figure 7: rms delay spread and coherence bandwidth for narrowband and wideband, respectively [7].

b. Doppler Spread and Coherent Time

The time variation of the channel is determined by the Doppler spread, which can be characterized from the Fourier transform of the scattering function with respect to τ , as written in (3.9):

$$S_c(\Delta f, \rho) = \int_{-\infty}^{\infty} S_c(\tau, \rho) e^{-j2\pi\Delta f\tau} d\tau \quad (3.9)$$

When $\Delta f = 0$, $S_c(\rho)$ is the Doppler spread spectrum of the channel. The maximum value of ρ , such that $S_c(\rho)$ is greater than zero, is called the Doppler spread of the channel, denoted by B_D . The coherence time can be obtained by taking the inverse Fourier transform of $S_c(\rho)$, which leads to $T_c \approx 1/B_D$. The relationship between Doppler spread and coherence time is shown in Figure 8.

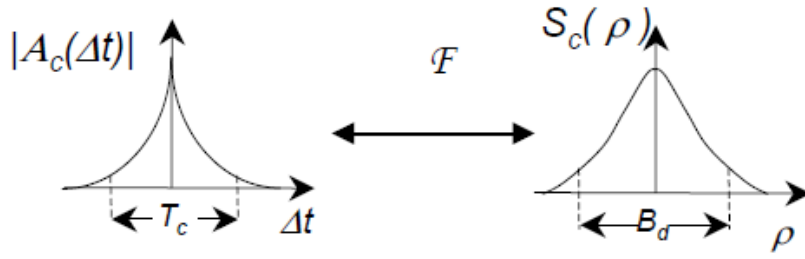


Figure 8: Coherence time and Doppler spread [7].

Therefore, the channel can be classified as either slow fading channel or fast fading channel. In the slow fading channel, the channel impulse response changes at a rate much slower than the transmitted symbols, given by $T < T_c$. In the fast fading channel, the channel impulse response changes faster than the symbols, i.e. $T > T_c$.

3.4 Limitations of This Project

3.4.1 Channel Specification

The content above in this chapter is the general knowledge of wireless radio channels. There are also some channel specifications for DECT communication systems. As mentioned before, DECT is an indoor telecommunication application. The protocol of DECT specifies the data rate as 1.152 Mbps, which means that the symbol duration is 0.868ms. When the delay spread of indoor environments is longer than 10% of the symbol time, the intersymbol interference cannot be negligible [8]. In other words, the transmitted signal would experience frequency selective fading. That is, significant ISI would occur at the receiver.

DECT is an application that only covers a range up to several hundred meters, which implies that it is impossible to make a call in a high-speed vehicle. The maximum motion speed of the handsets is up to 1.5 m/s like normal walking speed. Therefore, the Doppler spectrum is much smaller than the signal bandwidth, and the slow fading will occur. In this case, a linear FIR filter can be used to model the DECT channel in one time slot, as depicted in Figure 9.

Time invariant model for DECT channel

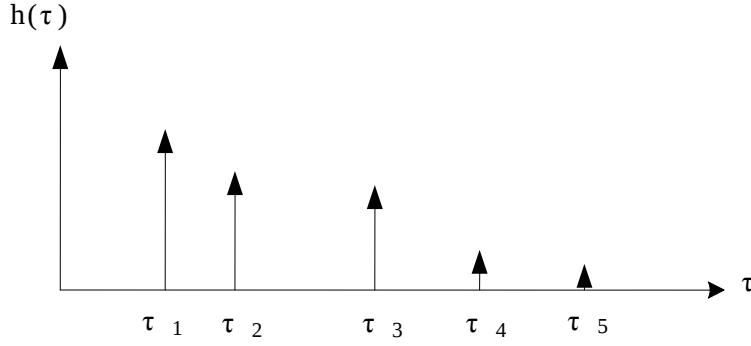


Figure 9: Discrete time model for the DECT channel.

Two models of the multipath channel are used in our simulations. One is the channel impulse response with each multipath component having equal gain (as model D in Table 1); in the other model, each path gain is attenuated as the delay is increasing (as model E in Table 1). In Table 1, each value indicates the delay spread versus the path gain.

Model	Path 1	Path 2	Path 3	Path 4	Path 5	Path 6
A (ns/dB)	0/0	50/-10	100/-13	150/-16	200/-19	250/-22
B (ns/dB)	0/0	50/-5	100/-8	150/-15	200/-21	25/-29
C (ns/dB)	0/0	100/-17	200/-20	300/-23	400/-26	500/-29
D (ns/dB)	0/-6	50/0	100/0	150/0	200/0	250/0
E (ns/dB)	0/0	100/0	200/-7	300/-13	400/-21	500/-28
F (ns/dB)	0/0	100/-10	200/-13	300/-16	400/-19	500/-26

Table 1: The standard discrete-time channel models in a wireless radio channel [9].

3.4.2 System specification

In order to simulate the system in a more practical way, a connection is built between a handset and a signal generator, which is similar to the connection to a base station. The uplink and downlink signals can be measured by connecting the oscilloscope to the external board as shown in Figure 10. The wave snapshots on the oscilloscope include the IF signals, the transmitted digital data and the demodulated or equalized signals, as shown in Figure 11, which can be used in Matlab and Modelsim simulations.

In the implementation, a pre-purchased FPGA development board (Low Power Reference Platform from Arrow Electronics) is used as the main development platform. A differential IF signal is extracted from the headset as input to the FPGA. A sequence of digital data is output from the FPGA and fed back to the headset. The system clock of the handset with the frequency of 10.368 MHz is also used as a clock input to the FPGA and the external ADC board. This also constrains the maximum clock frequency of FPGA at 10.368 MHz.



Figure 10: The connection between the handset and the oscilloscope.



Figure 11: Snapshot on the oscilloscope.

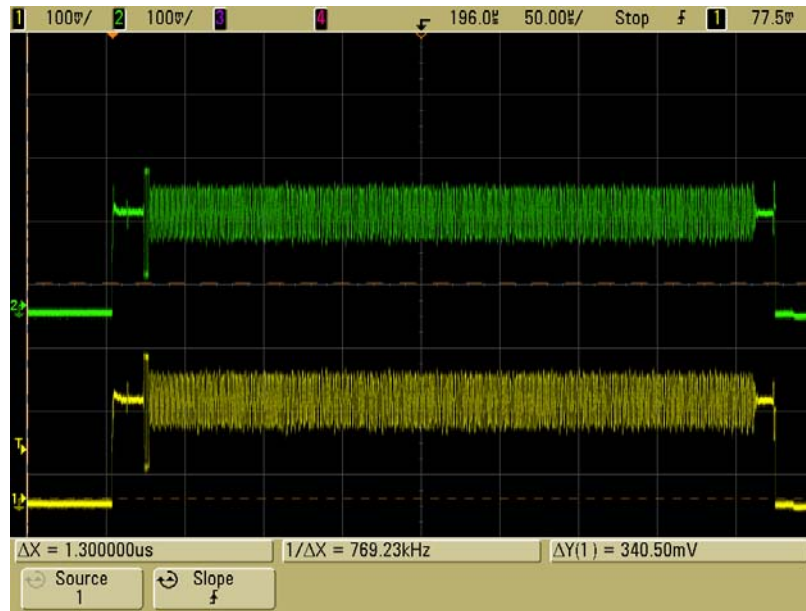


Figure 12: The differential IF signal sampled in the oscilloscope.

4. Modern Electronic Technologies of Digital Signal Processing

Figure 13 shows a typical application of a digital signal processing (DSP) system, which can be found in most wireless communication systems and as in this project. The analog signal is firstly fed through an analog anti-aliasing filter to suppress the unwanted frequency components. It is then followed by an analog-to-digital converter (ADC), which is normally implemented with a sample-and-hold component and a quantization circuit to convert the analog signal into digital domain. The digital signal processing circuits perform the next steps to process the digital signal (such as filtering, which is the most frequently used). After the DSP system, we could further process the data or generate an analog output signal (such as an audio signal) through a digital-to-analog converter (DAC) [10].

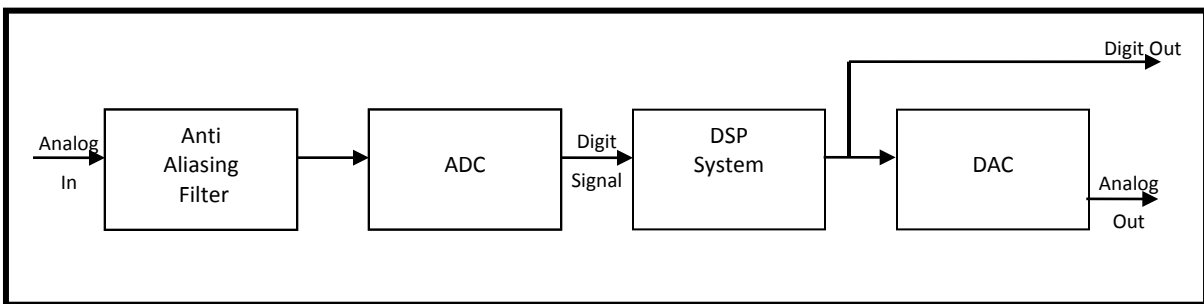


Figure 13: A typical DSP application.

The interfaces, between the analog and digital world, ADC and DAC are very important components in current research and industry. A fully-differential ADC is used in this project and will be introduced later. The DSP components, which are always the key part in any projects, can be implemented with many alternatives but the most common ones are DSP, ASIC and FPGA. In this chapter, we give an introduction to the most popular integrated circuits (IC) used for digital signal processing, and also some comparisons between them.

4.1 Digital Signal Processors (DSPs)

Digital signal processors (DSPs) normally refer to specially designed processors for digital signal processing. Most DSPs are sequential instruction based processors that provide fast mathematical computing, such as shift and addition, multiplication and addition. But unlike ordinary microprocessors, DSPs are often used as a type of embedded processors that are built into another piece of equipment and used for a special group of tasks [11]. In this case, the DSPs assist the general purpose microprocessors such as microcontrollers. This has been widely seen in cellular telephones, automobiles and plenty of advanced scientific instruments.

In fact, not many people use the word “DSP” to mean digital signal processor but only hardware engineers. On the other hand, it always refers to digital signal processing as in communication engineering. Mostly, digital signal processors are designed for digital signal processing. For example, a real world signal (mostly in analog form) can be converted into digital data and then be analyzed. This analysis is always done in digital domain because when a signal has been converted into digits; its components can be isolated, analyzed and rearranged more easily than in analog [11]. And after DSP system finishing its work, these digital data can be converted back to an analog signal with improved quality, such as removed interference, amplified amplitude, etc, as has been described in Figure 14. For this purpose, they usually have built-in mathematical blocks such as multipliers, arithmetic logic units (ALU) and shifters that are special for DSP functions.

DSPs can be classified by their dynamic input/output range that is the processor’s data width (the number of bits it manipulates) and the arithmetic type. Typically, they are 32-bit, 16-bit or floating-point, fixed-point. Each type of DSP is suited for a particular range of applications. Such as, 32-bit floating-point processor is used for image processing, 3-D graphics and scientific simulations and 16-bit fixed-point processor is usually for speech processing. But mostly, there is usually a waste of resource since the dynamic range is fixed for each DSP.

4.2 Application Specific Integrated Circuit (ASIC)

ASIC, as the name indicates, is an integrated circuit which is designed for a particular use. Normally, an ASIC is dedicated to a single function, or a few functions that are unchangeable. The development period of an ASIC is relatively long when comparing with a programmable circuit that ASICs could take a year or even more to finish. Despite the cost of an ASIC design, it could be very cost effective when the amount is high, and normally it gives better performance than its opponents. In most cases, it is possible to make an ASIC to meet the exact requirement for a specific application and use an exact number of components for this application to avoid any additional waste of resources. Nowadays, mixed signal ASIC enables both analogue and digital functions incorporating into one piece of IC. Therefore, ASIC is widely used in high volume products such as cell phone, audio/video player, automobile or other similar applications.

Traditionally, ASICs were designed by directly entering the silicon layout for the specified function, and this requires much longer development times than nowadays. The situation has changed with the improvements in computer-aided design (CAD) tools, and this enables more complicated design of circuits. Designers often use a hardware description language such as VHDL or Verilog to describe the functionality of their circuits, and use a compiler to generate the silicon layout automatically. Since the development and manufacturing of an ASIC is very expensive both in time and cost, there are different levels of customization to reduce the cost. From the least customizable level, they are Gate Array level, Cell level and Full-custom designs [12]. The more customization required, the more cost will be needed and also there will be higher risks of redesign as well as higher performance. A general ASIC design flow is shown in Figure 14. As shown, designing an ASIC is a long procedure, and there is a risk that defects could be found out after tapeout for which the cost would be as high as redoing it again.

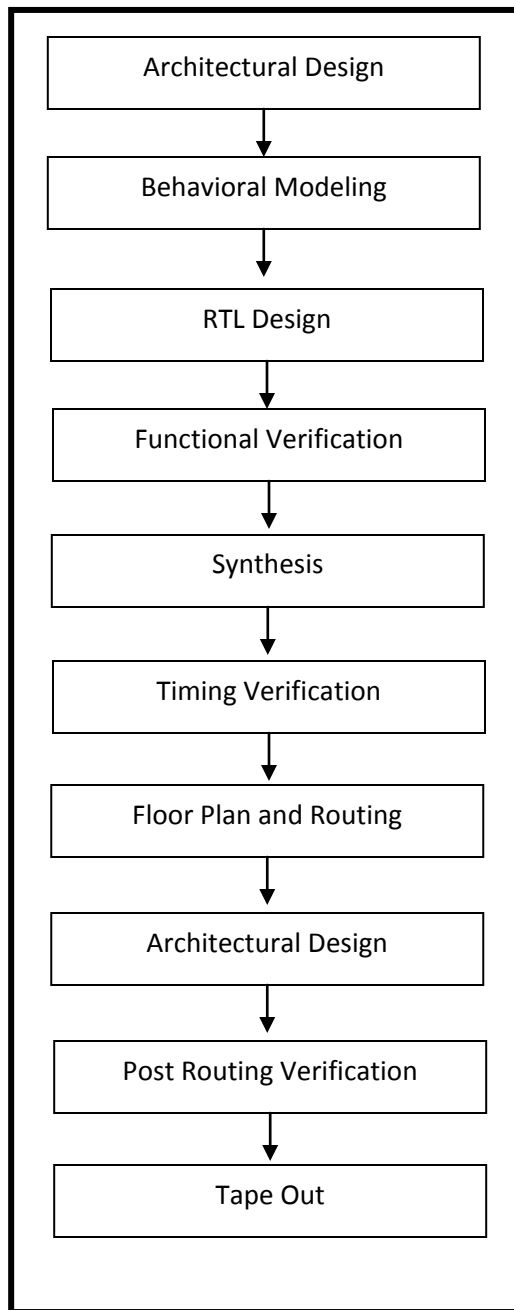


Figure 14: General ASIC design flow.

4.3 Field Programmable Gate Array (FPGA)

4.3.1 FPGA

FPGA has experienced a tremendous growth in recent years, and it has become a major player in the electronic industry. FPGA is an integrated circuit that can be electrically reprogrammed to become any kind of digital circuit or system by the customer or designer. FPGA contains an array of user-programmable logical blocks that could be designed to implement combinatorial and sequential circuits. FPGA could be considered as a simple “glue logics” technology that provides

programmable connectivity between these “logic blocks” where the programmability is based on either anti-fuse, EPROM or SRAM [13]. FPGA could be argued as a type of ASIC technology, since FPGA is an application specified IC. But classic ASIC design requires additional processing steps beyond those required for an FPGA. These additional steps provide ASIC with performance and power advantage, but also with high non-recurring engineering (NRE) costs. On the other hand, FPGA could gain a lot more time to meet the time-to-market requirement, and allows design errors being recognized at the late stage of development to be corrected with very low costs.

The most common FPGA architecture, as illustrated in Figure 15, consists of an array of programmable blocks of different types including logic block, memory and in recent devices DSP block. These blocks are surrounded by a programmable routing channel that allows the interconnection of the blocks to be customizable. These channels are surrounded by programmable input/outputs that connect the FPGA to the other circuits.

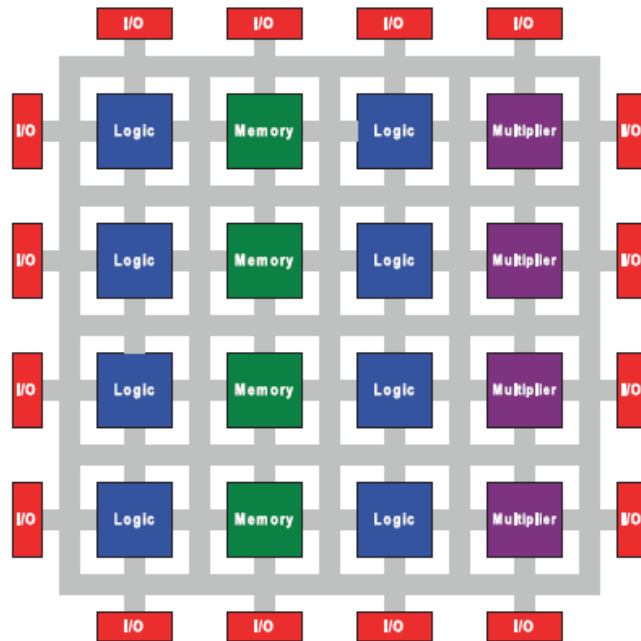


Figure 15: A simplified FPGA structure.

The logic block is the smallest programmable unit in the FPGA device architecture. A typical logic block consists of a 4-input look-up table (LUT) and a flip flop, as shown in Figure 16. There is only one output, which can be either the registered or the unregistered LUT output, and four inputs for the LUT and one clock input. In modern FPGAs, these logic blocks could also be multipliers, DSP blocks, embedded memories, etc. Some FPGA families employ PLL in the devices to enable multi-clock system and eliminate synchronization problem.

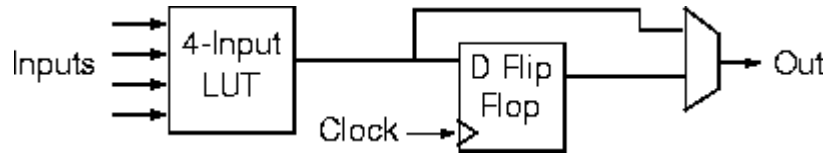


Figure 16: Structure of logic block [38].

The functions of an FPGA are usually defined by hardware description language (HDL) or a schematic design. For a bigger system, HDL is often easier for the designer. By using electronic design automation (EDA) tools, this function or behavior description is translated into a netlist which conveys the connectivity information. This netlist can be fitted into the FPGA by using place-and-route and then a binary file is generated which could be used to configure the FPGA. Most modern FPGA vendors provide a library of predefined circuits or components called IP cores, which could be in RTL (register transfer level) level or netlist level, to simplify the design of a complex system. For example, in the Altera FPGA, a predefined library called Megafunction is provided in their EDA tool, which contains thousands of IPs such as DSP functions, I/O prototypes, memories and embedded processors that are optimized structure for Altera FPGA architectures [14].

4.3.2 FPGA Versus DSP and ASIC

Traditionally, DSP processors have overwhelming advantages over FPGAs. FPGAs were used as a co-processor or controller while the DSP processors deal with the main calculations. However, as the cost of FPGA decreases and the higher needs for processing capabilities, high-end, DSP-oriented FPGAs already have a huge advantage over high-performance DSP processors in terms of performance/cost [15].

In most wireless applications, power is also a major consideration for designers. Previously, FPGA was viewed as too power-hungry as compared to DSP processors. But thanks to the development of CMOS technologies, there has been a number of low-power FPGAs which could compete with DSP processors in terms of power. Also, FPGAs, which are not constrained by a specific application or hardware structure, are much more flexible than DSP processors. Due to the highly flexible architecture, FPGAs can take advantage of their highly parallel architecture and offer an advantage in terms of performance/power.

On the other hand, DSP applications implemented on FPGAs typically take much more time and effort than on a DSP processor, since most FPGA designers still use low level hardware description languages, while DSP processors are mostly programmed in the more system level C language. Although there has been some effort on high level FPGA design tools such as System Generator [39] from Xilinx and DSP builder [40] from Altera, these tools have not been widely used yet. They are always hard to understand and require engineers to do their work in a new and unfamiliar way.

Till now, there are still big gaps from FPGA to ASIC in terms of higher performance, higher capacity, lower power, more efficiency, mixed signal integration performance and unit cost for high volume, although most FPGA vendors claims that their high-end FPGAs are a compelling

proposition for ASICs design. According to recent experimental measurement [16], there is still a quite significant gap between FPGAs and ASICs. For modern FPGAs, there are 21 times differences in silicon area, 3 to 4 times in critical path timing and 12 times for dynamic power consumption. However, due to their architecture, FPGA designs are much faster and easier and there is almost no recurring expenses when compare to ASIC design. So they are much more suitable for experimental applications and prototyping solutions. In wireless applications, FPGAs could be used to cope with the time-to-market issue and defect debugging purpose and then move on to build up their own ASICs later on. So, in this thesis project, FPGA is used to fulfill the experimental purpose.

5. FIR Filter and Equalizer Techniques

In this section, ISI and the solutions to ISI are discussed. Several types of equalization methods are introduced in this chapter.

5.1 Intersymbol Interference

Intersymbol interference is a common distortion in telecommunication, which means that one received symbol is interfered by the neighboring symbols [18]. The following equations are referenced from [19]. Denoting the received signal by $r(n)$, $r(n)$ equals to the convolution result between the transmitted signal $x(n)$ and the discrete channel impulse response h_n with length of $2L+1$, and n is used to denote the discrete time index:

$$r(n) = \sum_{l=-L}^L x(n-l)h_l + \omega(n) \quad (5.1)$$

Here, $\omega(n)$ is additive white Gaussian noise (AWGN). The equation can also be written as:

$$r(n) = x(n)h_0 + \sum_{l=-L}^{-1} x(n-l)h_l + \sum_{k=1}^L x(n-l)h_l + \omega(n) \quad (5.2)$$

where $x(n)$ is the dominating symbol in the receiver at time n . There is also interference from the previous and latter symbols, which is inferred by the second and third terms in (5.2). In this case, the receiver cannot make an accurate decision when $x(n)$ comes. Hence, some techniques should be employed in the receiver to reduce the interference.

One of the effective solutions for ISI distortion is equalization to compensate or reduce the ISI effect.

5.2 Digital LTI Filter and Adaptive Filter

Digital demodulators and equalizers can be considered as digital filters with specified functions. A digital filter performs mathematical operations on sampled, discrete-time signals to reduce or enhance certain attributes in time or frequency domain [20]. This is in contrast to analog filters, which work on continuous-time signals.

The most commonly used digital filters are linear and time invariant (LTI), which means that the impulse response of the filter interacts with the input signals through a linear convolution. The linear convolution process in the discrete domain is formally defined by:

$$y(n) = x(n) * f(n) = \sum_l x(l) f(n-l) = \sum_l f(l) x(n-l) \quad (5.3)$$

where $f(n)$ is the filter's impulse response, $x(n)$ is the input signal and $y(n)$ is the convolved output.

Digital filters can be classified as FIR filters and infinite impulse response (IIR) filters, respectively [32]. As its name indicates, FIR filters consist of a finite number of sample values, which reduce the above convolution sum to a finite sum for each output sample instant. An IIR filter, on the other hand, requires an infinite sum. The output of an FIR and an IIR filter at time n in terms of their input are described in (5.4) and (5.5), respectively:

$$y(n) = \sum_{l=0}^L a_l x(n-l) \quad (5.4)$$

$$y(n) = \sum_{l=0}^{L1} a_l x(n-l) - \sum_{l=1}^{L2} b_l y(n-l) \quad (5.5)$$

The filter can also be characterized by its system function. The transfer function of FIR and IIR filters in the Z-domain are depicted in (5.6) and (5.7), respectively:

$$H(z) = \sum_{l=0}^L a_l z^{-l} \quad (5.6)$$

$$H(z) = \frac{\sum_{l=0}^{L1} a_l z^{-l}}{1 + \sum_{l=1}^{L2} b_l z^{-l}} \quad (5.7)$$

Some important differences between FIR and IIR filters are summarized as follows [21]:

- Normally, FIR filters require substantially more filter coefficients or taps than IIR filters for equivalent responses to perform the same functionality.
- An FIR filter can be designed to have a linear phase response.
- Certain types of FIR filters can be configured in a symmetrical structure to save computational resources. Symmetrical even order FIR filters will require half the number of multipliers and consume less logic space.
- An IIR filter contains poles. Stability is determined by ensuring that the magnitudes of the poles are less than unity, i.e. the poles are within the unit circle.
- FIR filters are implemented directly and always stable due to no poles, which can be inferred from (5.6) and (5.7).

- IIR filters can directly emulate analog filters. Standard transformations can be used to convert Butterworth, Bessel, and other conventional analog filters into digital filters.
- IIR filters performance is more sensitive to quantization error.

Due to potential stability problems and more sensitivity to quantization error, IIR filters are not used in this project, even though it has some advantages over FIR filters. Therefore, all filters involved in this project are designed as FIR filters. Equalizers, as a special form of digital filters, will be introduced in following pages.

5.3 Baseband Equalization

According to their position at the receiver, equalizers can be divided into two categories: one is baseband equalizer, and the other is passband equalizer. Simple block diagrams of these two types of equalizer are shown in Figure 17.

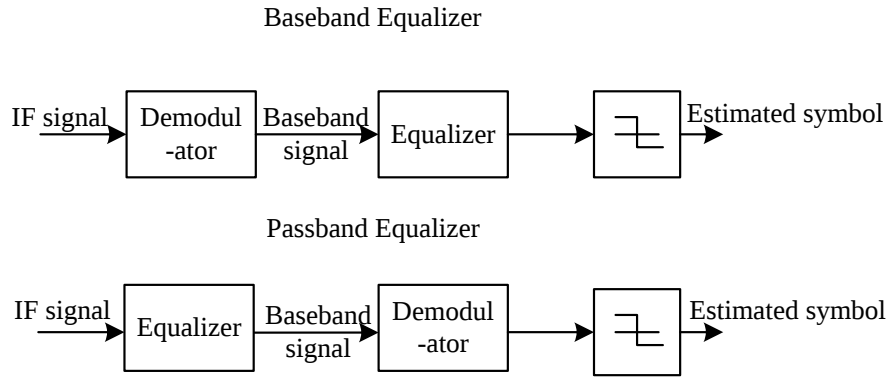


Figure 17: Block diagrams of baseband and passband equalizers, respectively.

Equalization can also be classified as either linear equalization or non-linear equalization in terms of its characteristics. Baseband linear equalization, as a simpler method in equalization techniques, is discussed firstly.

5.3.1 Linear Equalization

Linear equalization mainly includes zero forcing equalization and minimum mean square error (MMSE) equalization. Zero forcing equalization does not consider the effect of noise, and it is therefore seldom used in practical design. Hence, MMSE and its adaptive form are investigated in this section. The coefficients of MMSE equalizers are chosen to minimize the mean-square error, which consists of the sum of the squares of all the ISI terms plus the noise power at the equalizer output [17]. Therefore, the signal-to-distortion ratio could be maximized. The MSE criterion is defined as (5.8), and the minimizing coefficient vector is given by (5.9) [19].

$$MSE = E(|\mathbf{C}_L^T \mathbf{R}_L(n) - d(n)|^2) \quad (5.8)$$

where n is used to denote the discrete time index that represents as $t_n = nT$; the subscript L is used to denote the number of delay stages in the equalizer; $\mathbf{R}_L(n) = [r(n) \ r(n-1) \ \dots \ r(n-L+1)]^T$ is the received vector, $d(n)$ is the desired symbol and $\mathbf{C}_L = [c_0 \ c_1 \ \dots \ c_{L-1}]^T$ is the vector of equalizer weights.

The vector of MMSE equalizer coefficients, which minimizes (5.8), is given by:

$$\mathbf{C}_L = \mathbf{R}_{rr}^{-1} \mathbf{R}_{rd} \quad (5.9)$$

where

$$\begin{aligned} \mathbf{R}_{rr} &= E[\mathbf{R}_L(n) \mathbf{R}_L(n)^T] \\ &= E \begin{bmatrix} r(n)^2 & r(n)r(n-1) & \dots & r(n)r(n-L+1) \\ r(n-1)r(n) & r(n-1)^2 & \dots & r(n-1)r(n-L+1) \\ \dots & \dots & \dots & \dots \\ r(n-L+1)r(n) & r(n-L+1)r(n-1) & \dots & r(n-L+1)^2 \end{bmatrix} \end{aligned} \quad (5.10)$$

and

$$\begin{aligned} \mathbf{R}_{rd} &= E[\mathbf{R}_L(n) d(n)] \\ &= E[d(n)r(n) \ d(n)r(n-1) \ \dots \ d(n)r(n-L+1)]^T \end{aligned} \quad (5.11)$$

As an LTI filter, the coefficients of the MMSE equalizer do not change over time. But in typical DSP fields, such as speech processing, echo cancellation, radar, sonar, seismology, or biomedicine, the applications require the filter coefficients to vary over time. A filter with adjustable coefficients is called an adaptive filter [32]. Two basic algorithms for adaptive filtering will be taken into consideration in the following sections, i.e. LMS and RLS.

a. LMS Algorithm

LMS and RLS, as two basic algorithms, are widely used in adaptive filtering. In the LMS algorithm, the update at time n is computed as (5.12)-(5.14):

$$\hat{d}(n) = \mathbf{C}_L^T(n-1) \mathbf{R}_L(n) \quad (5.12)$$

$$e(n) = d(n) - \hat{d}(n) \quad (5.13)$$

$$\mathbf{C}_L(n) = \mathbf{C}_L(n-1) + \mu e(n) \mathbf{R}_L(n) \quad (5.14)$$

where $\hat{d}(n)$ is the estimated output of equalizer, $e(n)$ is the prediction error and μ is the step size.

Apparently, the desired symbol $d(n)$ cannot always be known by the receiver side. In order to keep updating all the time, the equalizer has to work in two modes.

The first is the training mode: in this mode, the equalizer knows a part of the desired symbols (training sequence). By detecting the training sequence in the received signals, the adaptive algorithm can compute and minimize the error by adjusting the tap weights [1]. The length of the training sequence depends on the number of equalizer coefficients, which must be determined on beforehand. Normally, the duration of the training mode is no more than tens of symbols. It must be noted that if the channel has changed, the equalizer must be retrained. In this project, since the channel is time invariant during one time slot, the equalizer is not retrained in that duration, but it must be trained when next time slot comes.

The direct decision mode comes after the training mode. To keep adaptation after the training sequence, $d(n)$ in (5.13) directly depends on the output of receiver, as $d_{\text{direct}}(n)$ in Figure 18, and thus, some errors can happen when $d_{\text{direct}}(n)$ is incorrectly estimated. But in practical tests, the BER is lower when having the direct decision mode as compared to not updating during the data mode. The block diagram of these two modes is depicted in Figure 18.

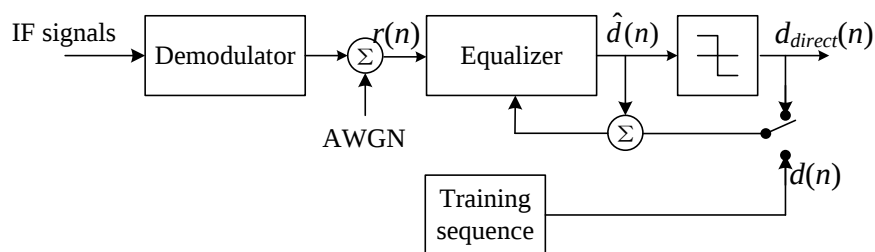


Figure 18: The block diagram of training tracking equalizer.

In the LMS equalizers, several important issues should be emphasized. Firstly, the vector $\mathbf{C}(0)$ must be initialized. Theoretically, it can be chosen arbitrarily, but in this project, it is defined as an all zero vector, since each initialization corresponds to a new time slot.

Secondly, when the transmitted signals are specified, the convergence speed only depends on the factor of μ (given in (5.14)) called step size. The step size needs to be well defined to fit the system requirements. Otherwise, the LMS algorithm can be either unstable or converge very slowly. To prevent the adaptation from becoming unstable, the value of μ must satisfy [32]:

$$0 < \mu < \frac{2}{\lambda_{\max}} \quad (5.15)$$

where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{R}_{rr}$.

Thirdly, in this project the equalizer is quite sensitive to the synchronization result. When a training sequence is used in the equalization, the synchronization of the training sequence must be

in a high level of accuracy. If not, the weights of equalizer cannot be trained sufficiently and may cause even more destructive effects to the received signals.

b. RLS Algorithm

LMS algorithm is a low cost computation method and easier to implement in hardware. However, it usually needs tens or even hundreds of iterations to converge. Sometimes, this is unacceptably slow. Instead, RLS can give a higher convergence speed with the trade off in computation complexity.

Initialize $\mathbf{C}(0) = \mathbf{0}$, $\mathbf{P}(0) = \delta \mathbf{I}_{LL}$, where $\mathbf{I}_{LL}$ is an $L \times L$ identity matrix, and δ is a large positive constant, The RLS algorithm computation is then described as follows [33]:

Compute the filter output:

$$\hat{d}(n) = \mathbf{C}_L^T(n-1) \mathbf{R}_L(n) \quad (5.16)$$

Compute the error:

$$e(n) = d(n) - \hat{d}(n) \quad (5.17)$$

Compute the Kalman gain vector and the inverse of correlation matrix:

$$\mathbf{K}_L(n) = \frac{\mathbf{P}(n-1) \mathbf{R}_L(n)}{\omega + \mathbf{R}_L^T(n) \mathbf{P}(n-1) \mathbf{R}_L(n)} \quad (5.18)$$

$$\mathbf{P}(n) = \frac{1}{\omega} [\mathbf{P}(n-1) - \mathbf{K}_L(n) \mathbf{R}_L^T(n) \mathbf{P}(n-1)] \quad (5.19)$$

Update the coefficient vector of filter:

$$\mathbf{C}_L(n) = \mathbf{C}_L(n-1) + \mathbf{K}_L(n) e(n) \quad (5.20)$$

In (5.18), ω is the weighting factor that can tune the performance of the RLS equalizer. If the channel is time invariant, ω can be set to one. Usually, it is in the range of (0.8, 1). The value of ω cannot affect the convergence speed of the equalizer, but can determine the tracking ability of the RLS equalizer [1]. When it is smaller, the tracking ability is better, but a smaller value of ω leads higher noise sensitivity. The convergence speed of LMS and RLS is illustrated in Figure 19.

As mentioned above, the convergence speed can be increased a lot by using RLS instead of LMS, but the cost is much higher in terms of computation complexity.

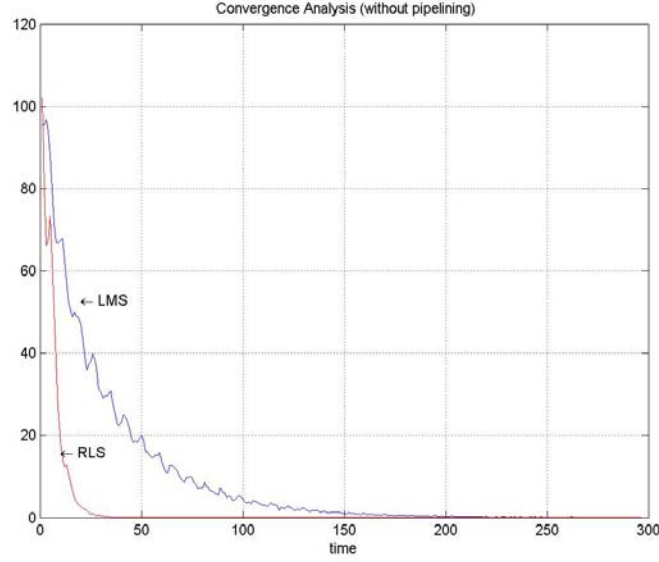


Figure 19: the convergence speed analysis of RLS and LMS.

The computation complexity of these two algorithms is listed in Table 2. Because of the huge computation cost of the RLS algorithm, it is not possible to implement the RLS adaptive equalizer on the FPGA development board given in this project.

Algorithm	Computational Load	
	Multiplication	Division
LMS	$2L+1$	-
RLS(Kalman)	$2.5L^2+4.5L$	2

Table 2: The computation complexity of LMS and RLS.

5.3.2 Nonlinear Equalization

When the ISI is much more severe, the linear filter may not recover the distorted signals very well, and a nonlinear equalizer can be an effective method in this case. Two main types of nonlinear equalizers are described in this section. One is decision feedback equalizer (DFE) and the other is maximum likelihood sequence estimator (MLSE).

a. Decision Feedback Equalization (DFE)

The DFE equalizer consists of a feedforward filter (FFF) with the received symbols as its input and a feedback filter (FBF) with the output detector as its input. The structure of DFE is shown in Figure 20. Once a symbol is detected and decided, by using the feedback filter, the ISI it induces on future symbols can be subtracted before the detection of the coming symbols.

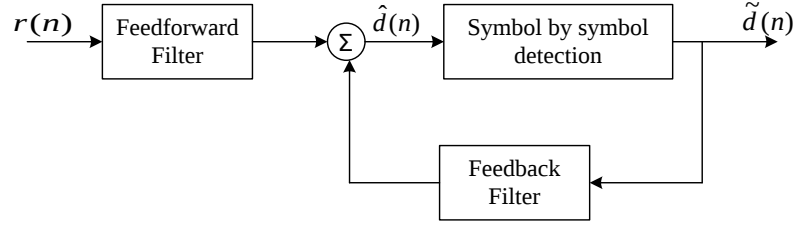


Figure 20: The structure of DFE.

The mathematical expression of a DFE is given by following equations:

$$\hat{d}(n) = \sum_{l=-L_1+1}^0 a_l r(n-l) + \sum_{l=1}^{L_2} b_l \tilde{d}(n-l) \quad (5.21)$$

where a_l and b_l denotes the coefficients of FFF and FBF, respectively.

Either ZF or MMSE criterion can be employed to design the feedforward filter. Moreover, the adaptive methods like LMS and RLS are also suitable for DFE. For instance, if LMS is employed, the computations of the DFE coefficients are shown below [33]:

Compute the error:

$$e(n) = d(n) - \hat{d}(n) \quad (5.22)$$

Update the coefficients of two filters:

$$\mathbf{a}_{L1}(n) = \mathbf{a}_{L1}(n-1) + \mu \mathbf{R}_{L1}(n) e(n) \quad (5.23)$$

$$\mathbf{b}_{L2}(n) = \mathbf{b}_{L2}(n-1) + \mu \tilde{\mathbf{d}}_{L2}(n) e(n) \quad (5.24)$$

where $\mathbf{a}(n)$ and $\mathbf{b}(n)$ is the weights vector of FFF and FBF at time n , respectively. The subscripts of $L1$ and $L2$ are used to denote the length of these two filters.

Although the DFE equalization shows superior performance over the linear equalization, there are also defects of the DFE equalizer. For example, once an error happens in the decision part, the input of the FBF is not the one as expected, and the ISI effects from previous symbols is wrongly estimated, then the error propagation occurs. This will degrade the signal seriously.

b. Maximum Likelihood Sequence Estimator (MLSE)

MLSE can avoid many problems in the equalizers discussed above. MLSE estimates the transmitted signals according to the knowledge of the channel, which is different from the equalization methods discussed before. The structure is the same as the upper plot of Figure 17, when replacing the equalizer block by a MLSE estimation algorithm block. Usually, the Viterbi algorithm [17] is employed in the MLSE receiver. MLSE is the optimal receiver with a sacrifice on the computation complexity. The computation complexity increases exponentially as the

number of signal states increase. Hence, it is not considered in this project. Figure 21 illustrates the BER by using different equalizers.

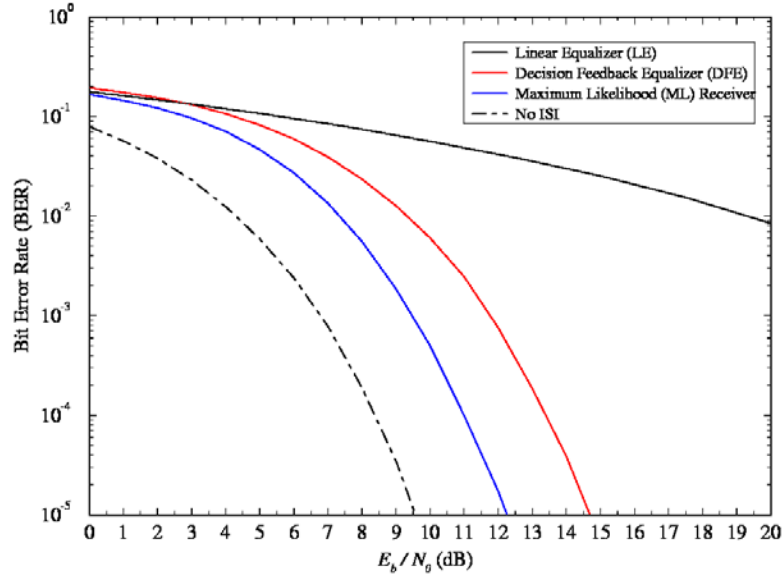


Figure 21: BER with different type of equalizer.

5.4 Passband Equalization and Fractionally Spaced Equalization

The equalizer does not only work with baseband signals but it can also be used with passband signals. For example, passband equalization can be an effective option when the demodulator introduces some nonlinear effects, which is relatively tough for the linear baseband equalizer to deal with. Since the sampling rate of passband signals is greater than the symbol rate according to the Nyquist criterion, an over-sampling equalizer, known as fractionally spaced equalizer (FSE), is used for the passband equalization. The sampling rate is at least as fast as the Nyquist rate.

6. System Design and Simulation Result Analysis

6.1 Overview of the Design

A system block diagram of the design is shown in Figure 22. The function of each component is described in following sections.

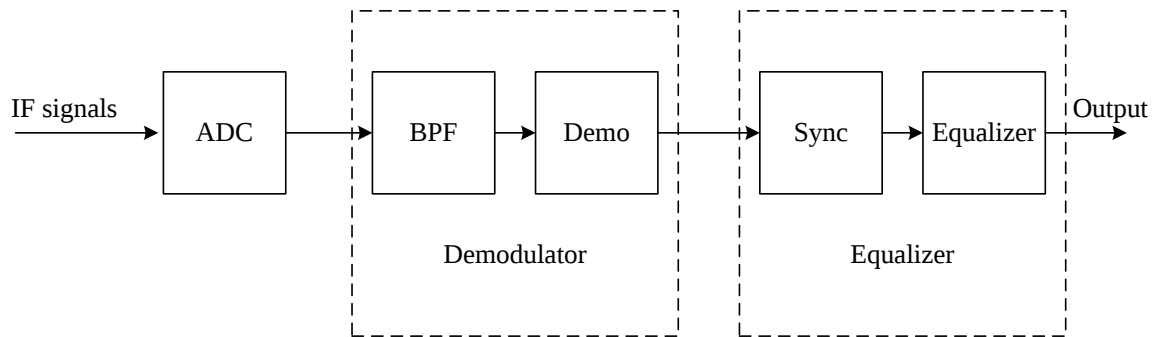


Figure 22: System block diagram of non-coherent equalizer.

6.2 AD Conversion

An FPGA works in digital world and the signals transmitted over air have to be analogue. So, an analog to digital converter is needed to convert the analog signals to digital bits. In order to get identical results between the Modelsim and Matlab simulation, the data used in Matlab simulations are also based on the output of the AD converter.

The data sampled from the AD converter is ten bits; the Peak to Peak value of the ADC is 2V. It is a ten bits quantization (2^{10} levels) to the raw analog data. The sampled output from the oscilloscope is 10 bits within the range of (-32768, 32767), so it needs to be rescaled. The analog data, the sampled data and the rescaled data are shown in Figures 23 - 25.

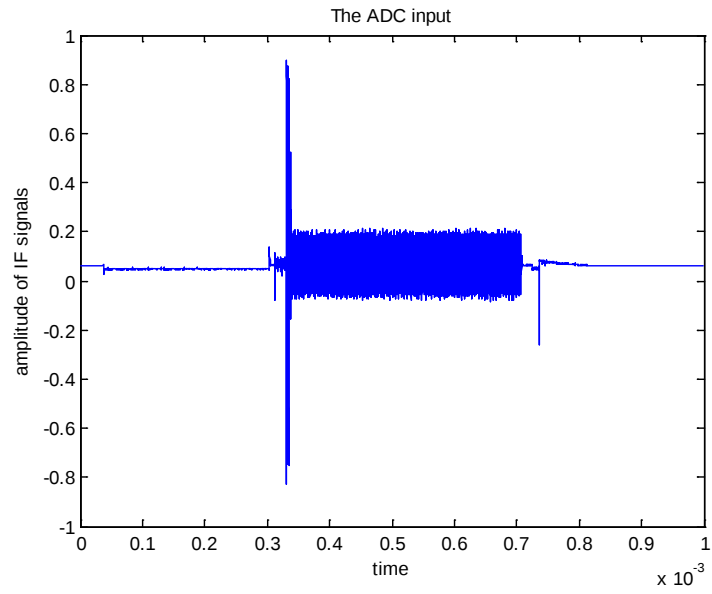


Figure 23: The IF signal before AD conversion.

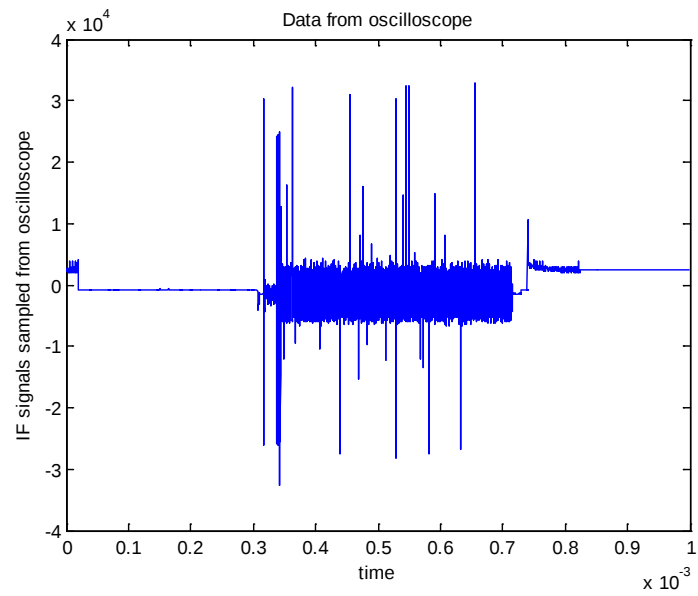


Figure 24: The output from the ADC sampled from the oscilloscope.

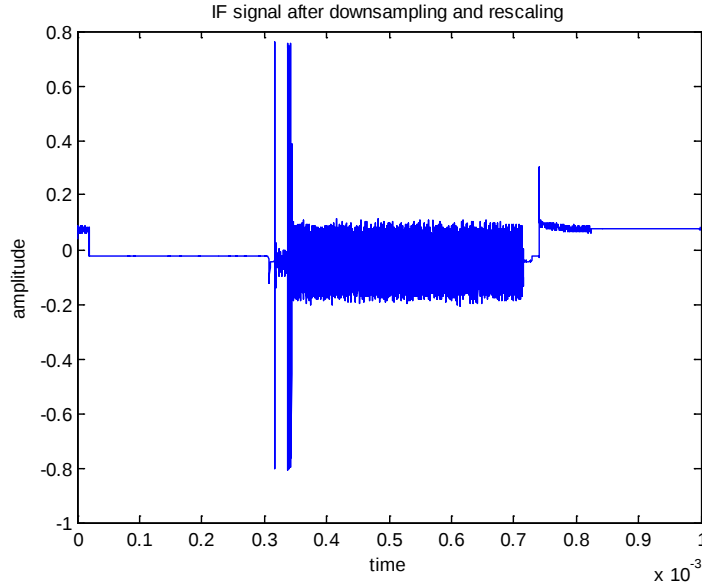


Figure 25: The rescaled version of Figure 24.

6.3 Demodulation Method Simulation and Result analysis

The demodulation structure consists of two parts: an IF bandpass filter and a demodulator. The function of each component is listed below.

IF bandpass filter: the input digital signals from the ADC contain both high frequency noise and DC voltage offset. But only the intermediate frequency signals are of interest. An FIR bandpass filter is used to remove these interferences.

Demodulator: the demodulator component converts the IF signals to baseband. It basically consists of one delay path (referred to as shift registers in hardware), a multiplier and a lowpass filter. The incoming IF signals are firstly self-multiplied, which generates mixed signals including both the baseband component and the high frequency component. And then the high frequency component of the mixed signals is filtered out by a lowpass filter (LPF). The demodulation process is shown in Figure 3. In order to get higher spectrum efficiency, DECT uses GFSK as the modulation scheme. The baseband digital pulse and the baseband Gaussian pulse are shown in Figures 26 and 27, respectively.

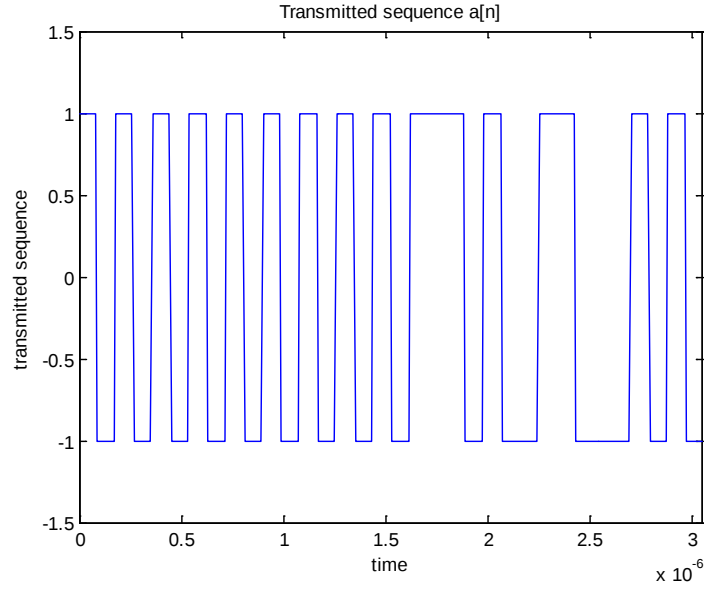


Figure 26: Digital NRZ sequence.

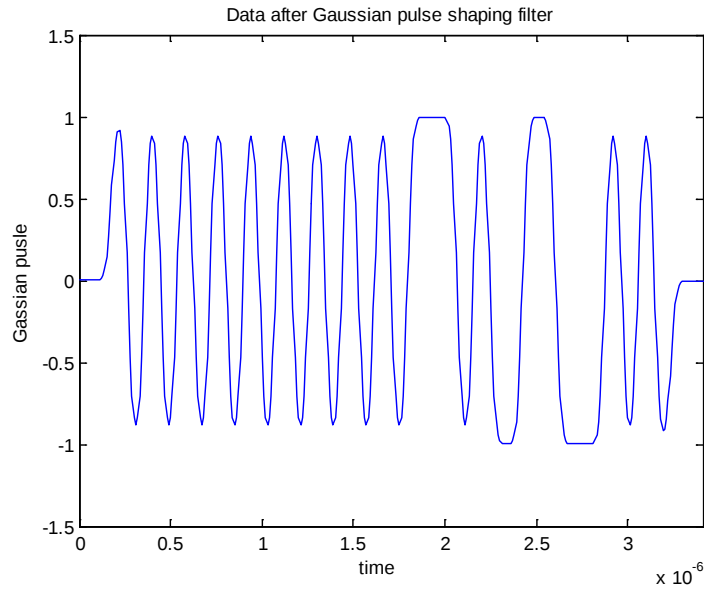


Figure 27: Gaussian pulse sequence.

The Gaussian pulse sequence is modulated to passband after the Gaussian pulse shaping filter. In the DECT system, the carrier frequency of the IF signals is 864 kHz. The mathematical expression of the modulation is given by (6.1), where t is used to denote the continuous time due to the analog modulation. Figure 28 shows a representative example of a GFSK modulated signal.

$$s(t) = \cos \left[2\pi f_c t + \frac{2\pi f_d}{T} \int_{-\infty}^t \sum_{n=1}^N a(n) g(v - nT) dv \right] \quad (6.1)$$

where f_c is the carrier frequency, f_d is the deviation frequency and $g(t)$ is the Gaussian pulse.

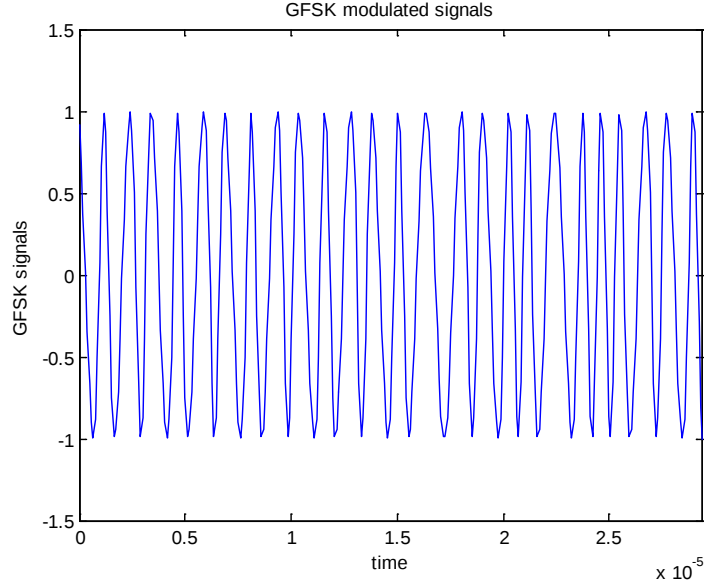


Figure 28: GFSK Modulated signal with modulation index of 0.5.

A quadrature detector is used as the demodulation scheme, which is a standard choice. The continuous time block diagram is illustrated in Figure 29. Denoting the delay in terms of 90 degree phase shift by τ_o , $m(t)$ is given by (6.2).

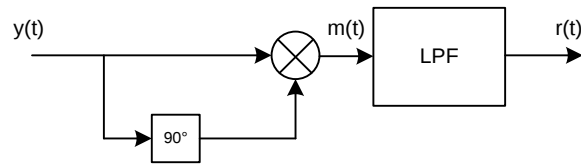


Figure 29: Block diagram of quadrature detector in continuous time domain.

$$\begin{aligned}
m(t) &= y(t)y(t - \tau_0) \\
&= \cos \left[2\pi f_c t + \frac{2\pi f_d}{T} \int_{-\infty}^t \sum_n a_n g(v - nT) dv \right] \cos \left[2\pi f_c (t - \tau_0) + \frac{2\pi f_d}{T} \int_{-\infty}^{t - \tau_0} \sum_n a_n g(v - nT) dv \right] \\
&= 1/2 \left\{ \cos \left[2\pi f_c \tau_0 + \frac{2\pi f_d}{T} \int_{t - \tau_0}^t a_n g(v) dv \right] \right\} \\
&\quad + 1/2 \left\{ \cos \left[2\pi (f_c t + f_c t - f_c \tau_0) + \frac{2\pi f_d}{T} \int_{-\infty}^t \sum_n a_n g(v - nT) dv + \frac{2\pi f_d}{T} \int_{-\infty}^{t - \tau_0} \sum_n a_n g(v - nT) dv \right] \right\}
\end{aligned} \tag{6.2}$$

After lowpass filtering, only the first term in (6.2) is preserved. Let $2\pi f_c t = \pi/2$, then the baseband component of (6.2) reduces to:

$$\begin{aligned}
\cos \left[2\pi f_c \tau_0 + \frac{2\pi f_d}{T} \int_{t - \tau_0}^t a_n g(v) dv \right] &= \cos \left[\frac{\pi}{2} + \frac{2\pi f_d}{T} \int_{t - \tau_0}^t a_n g(v) dv \right] \\
&= -\sin \left[\frac{2\pi f_d}{T} \int_{t - \tau_0}^t a_n g(v) dv \right] \approx -\frac{2\pi f_d}{T} \int_{t - \tau_0}^t a_n g(v) dv
\end{aligned} \tag{6.3}$$

The approximation in (6.3) is based on the property of $\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$, so when x is relatively small, we have $\sin x \approx x$. If the integration interval is relatively small, the result of the integration is low enough to satisfy the property mentioned above. Thus, we can make the decision according to the sign of the demodulator output. When it is negative, the transmitted bit is 1, and vice versa.

Two issues in the demodulation should be noticed. One is calculating the demodulation delay τ_0 and the other is the design of LPF in the quadrature detector.

The delay of τ_0 is given by

$$\begin{cases} 2\pi f_c \tau_0 = \pi/2 \\ f_c = 3/T4 \end{cases} \Rightarrow \tau_0 = \frac{T}{3} \tag{6.4}$$

where T is the symbol duration.

In this project, all signals are processed in the digital domain, so all signals denoted in Figure 29 should be discrete, as shown in Figure 30 (n is used to denote discrete time index that represents $t_n = nT_s$, where $T_s = 1/\text{sampling frequency}$). According to the result of (6.4), if the sampling rate of ADC is an integer multiple of 3, the number of samples corresponding to τ_0 is easy to define. In this project, the sampling rate is predefined as 10368 kHz which is 9 times of the symbol rate. So the number of samples corresponding to τ_0 is 3.

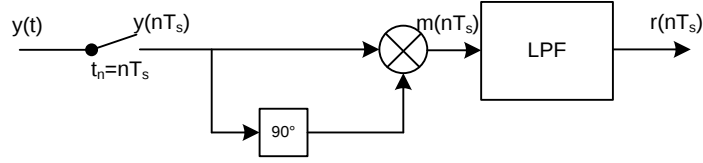


Figure 30: Block diagram of quadrature demodulator in discrete time domain.

As shown in Figures 29 and 30, the LPF is applied after the multiply operation. The cutoff frequency of the LPF is one of the main factors in the demodulation. Usually, it is determined by the 3 dB law. Figure 31 shows the spectrum of $y(nT_s)$ in the ideal channel.

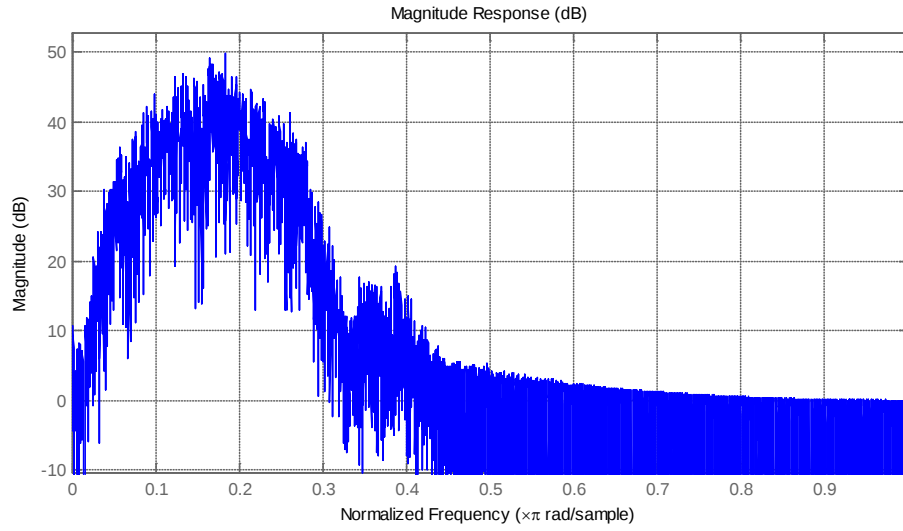


Figure 31: The spectrum of IF signals $y(nT_s)$.

Figure 32 depicts the spectrum of the signal $m(nT_s)$, which includes both the baseband and passband components.

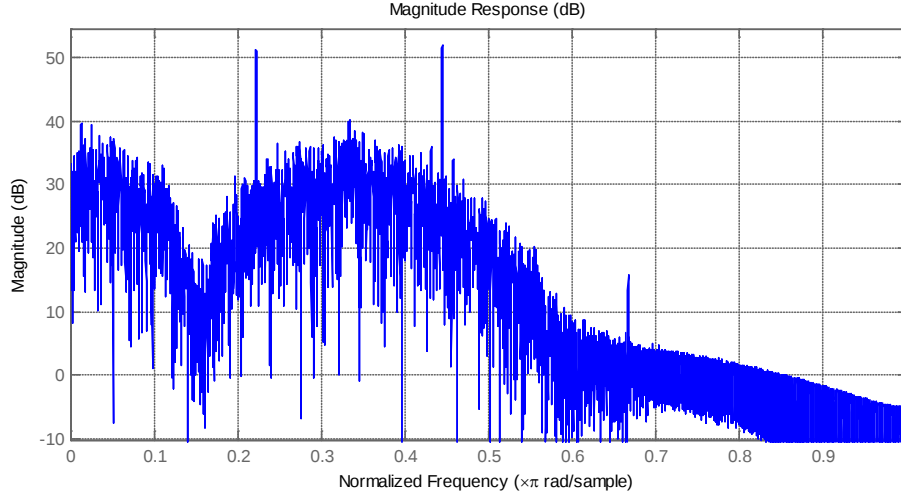


Figure 32: The spectrum of mixed signal $m(nT_s)$.

According to the 3 dB cutoff law, the first trough of the spectrum is the ideal point for the cutoff frequency, which equals 0.15. Two kinds of window functions, rectangle and hamming, are tested in the filter design, and they are shown in Figure 33.

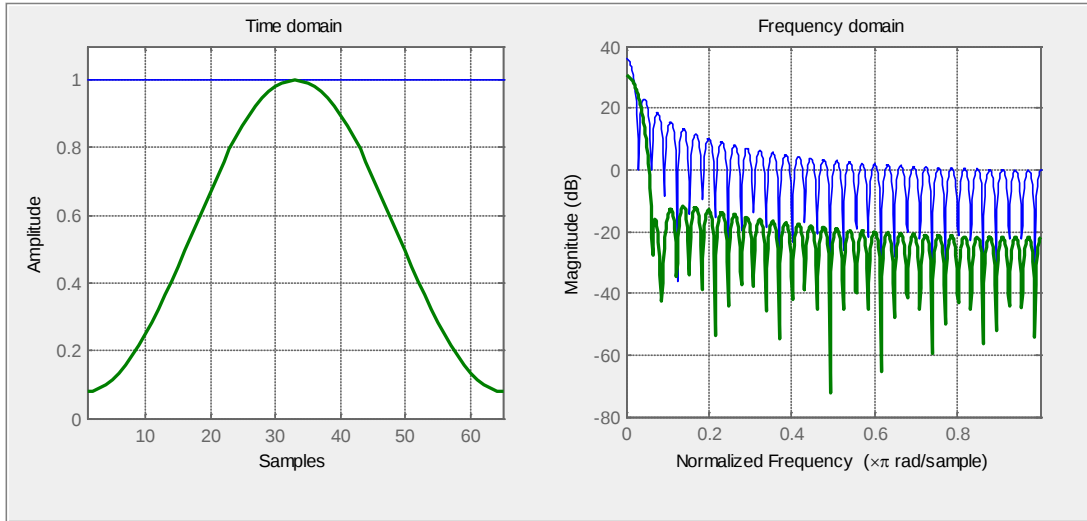


Figure 33: Comparison between a rectangle window and a hamming window in both time domain (left) and frequency domain (right).

The FIR low pass filters constructed by these two windows are shown in Figures 34 and 35. Figure 34 shows the frequency response with length of 32, and Figure 35 corresponds to the filter length of 64. When the filter length is 32, the LPF with hamming window has better stopband attenuation but a wider transition band, and the LPF with rectangle window is in contrast to this.

When increasing the length to 64, the LPF with hamming window can have a better transition band, and there is almost no improvement in the rectangle filter's spectrum.

Therefore, the hamming LPF with filter length of 64 is used in the demodulation. Apparently, the longer the filter, the higher the performance, but the higher computation complexity and longer delay will be introduced.

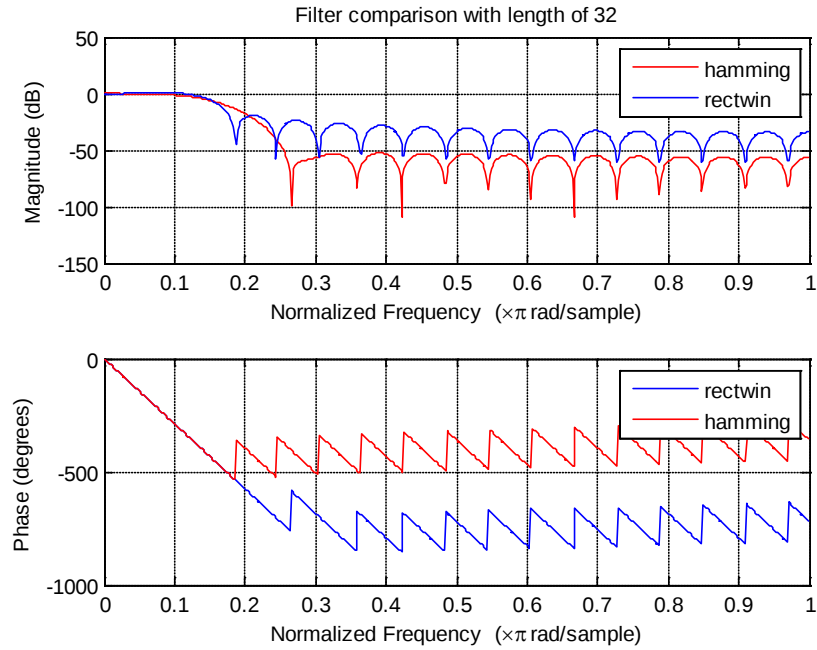


Figure 34: FIR filter comparison with length 32.

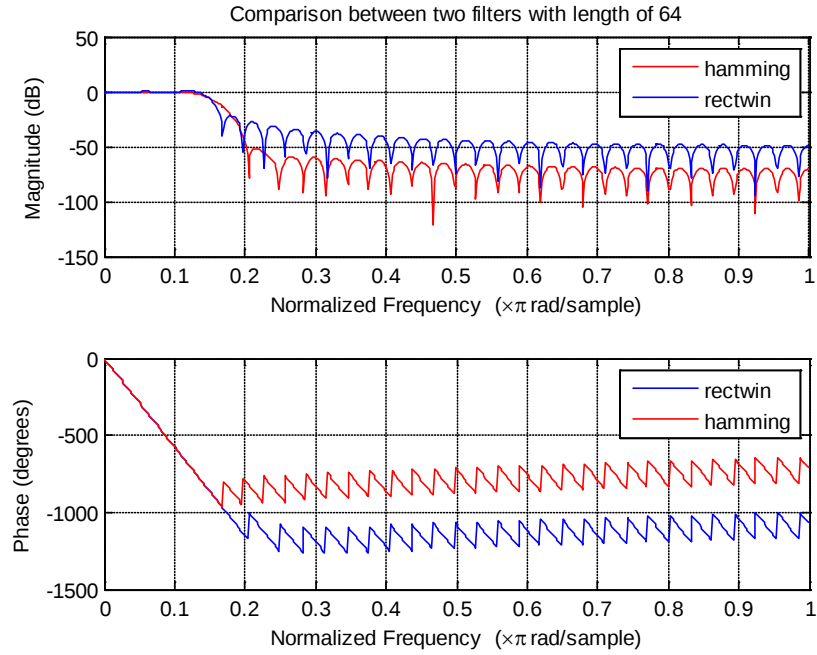


Figure 35: FIR filter comparison with length 64.

All components depicted in Figure 30 have been defined above, and Figure 36 shows the demodulated result of a noise free IF signal from an ideal channel.

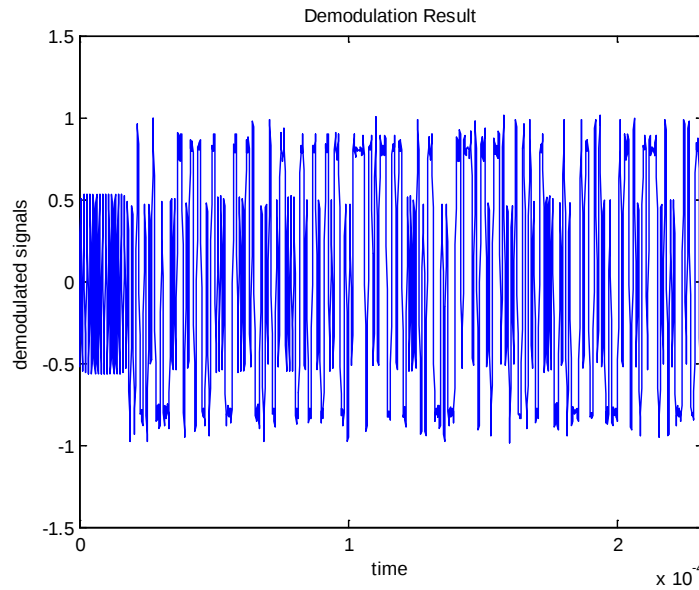


Figure 36: The demodulated signal.

Without taking the multipath fading effects into consideration, there are also some other interferences degrading the performance of demodulation. In the ideal ADC, due to the identical

direct current (DC) offset of the differential outputs, the DC offset can be removed after subtraction. However, in a practical application, the DC offset of the differential IF signals will be dynamic within a small range, which is destructive for the performance of the demodulator. If there is an extra DC offset adding to the result of (6.3), the decision is not only decided by the sign of the demodulated results anymore. Therefore, a component for removing the DC offset of the IF signals should be constructed prior to the demodulator. Figure 37 shows the spectrum of the IF signal $y(nT_s)$ with DC offset.

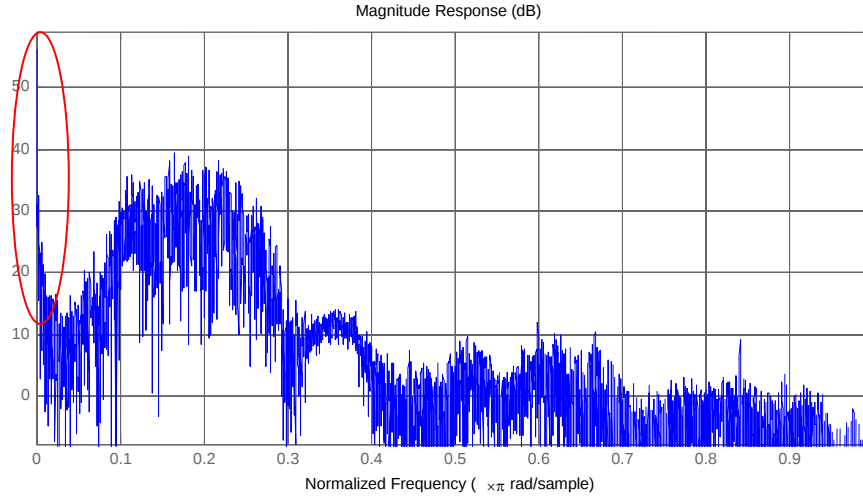


Figure 37: The spectrum of IF signals with DC offset.

Two methods are tested to remove the DC offset. One is directly subtracting a constant value from the IF signals. By continuous tests, The DC offset always fluctuates round 0.08. After subtracting this constant value, the new DC offset could fall into an acceptable level, and the demodulation is not so sensitive to this DC offset as before.

Although the demodulation performs better by this method, it is still a risk when the ADC is changed. Apparently, it is unrealistic to measure and tune that constant value time and again in a practical implementation. A more common method for removing the offset is to filter out the DC offset by either a high pass filter (HPF) or a bandpass filter (BPF).

A 128 order FIR type of BPF is used in this project, by which both DC offset and the noise outside the signal band is removed. Figure 38 shows the frequency response of this bandpass filter.

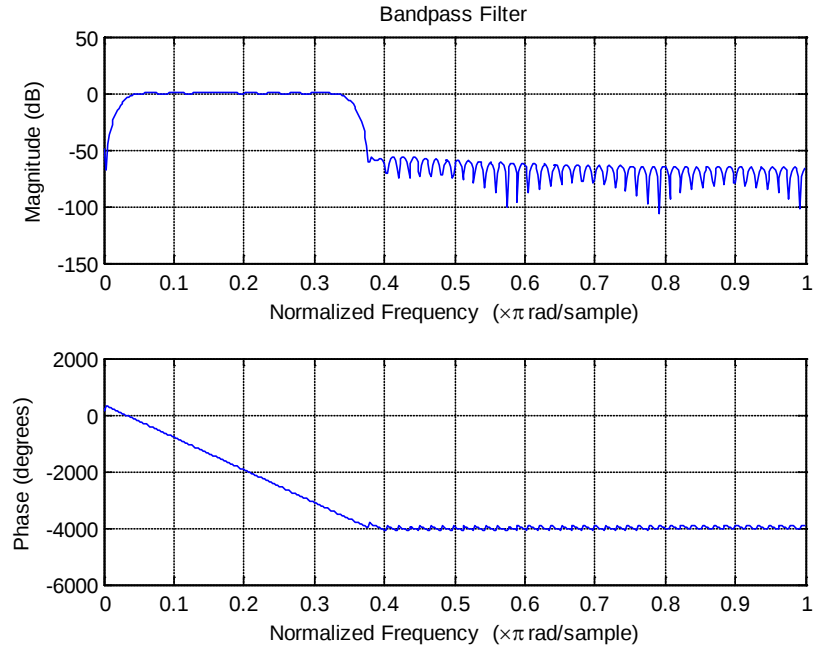


Figure 38: The frequency response of the bandpass filter.

With all components discussed before, an entire structure of the digital non-coherent demodulator is available. Figures 39 and 40 depict the demodulation results in the channel with different levels of fading. Since The GMSK modulation is constant envelope modulation, the envelope of the IF signals is kept constant in the normal channel, as shown in Figure 39. The demodulated result satisfies the BER requirement. However, when the multipath fading is severe, the envelope of the IF signal is no longer constant, as shown in Figure 40. Moreover, the BER is increased. The BER of the baseband signal in the lower plot of Figure 40 is higher than 5%, which degrades the audio quality significantly. Thus, equalization techniques should be used.

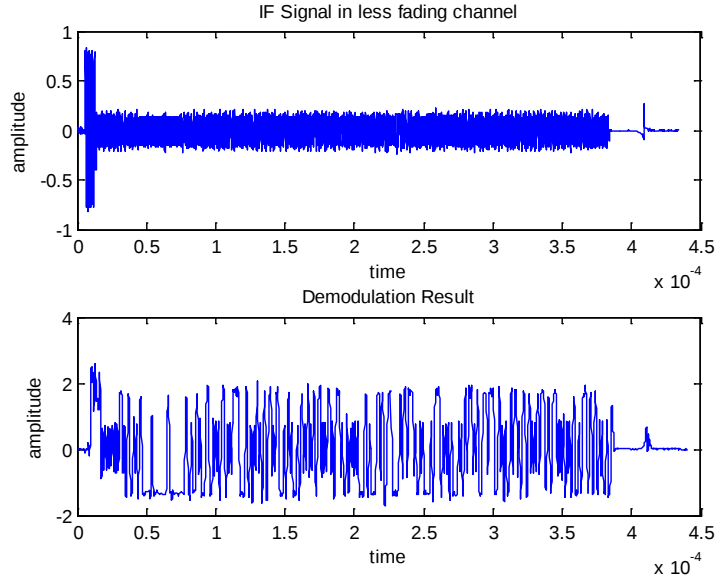


Figure 39: Demodulation result with less fading effects.

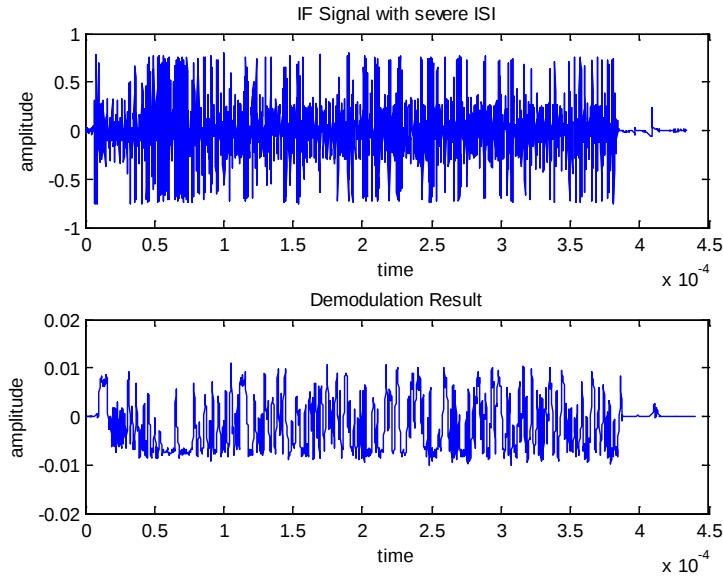


Figure 40: Demodulation result with severe ISI effects.

6.4 LMS Baseband Equalizer and Result Analysis

An LMS baseband equalizer as the core component is built into the system due to its low cost in terms of implementation. Two components are included in this block, namely: synchronizer and equalizer. A brief introduction to each component is listed below.

Synchronizer: this component is used to detect where the start of the training sequence in each time slot is, and thus enable the equalizer.

Equalizer: as the essential part of our design, the equalizer is designed as a fractionally spaced equalizer due to its less sensitivity to the synchronization accuracy. When the synchronization component finds the start of training sequence, the equalizer updates its coefficients according to the training sequence in the training mode, and then keeps updating according to direct decision. According to the sampling rate of the receiver, nine samples are used to represent one symbol, and the coefficients are updated every 9 samples, which maintains the stationarity of the LMS adaptive filter.

6.4.1 Synchronization

The first 16 bits in the synchronization field of the data package are used as the syncword and training sequence, which are “1 1 1 0 1 0 0 1 0 0 0 1 0 1 0 1”.

A maximum value can be found by taking the cross correlation between the demodulated signals and the syncword, as shown in Figure 41. Since the equalizer is a fractionally spaced equalizer, the cross correlation should also be taken in an oversampled mode. The amplitudes of the demodulated signals are dynamic with respect to the dynamic amplitude of the received IF signals. So, only the sign of each sample is treated in the cross correlation. As 1 symbol is represented by 9 samples, the ideal maximum value of the cross-correlation is 72. But it seldom reaches this ideal value due to distortion. The peak value is mostly around 60 according to the results from a large amount of simulations, and a value of 55 is defined as the threshold for searching for the peak. The synchronization will fail, if the peak is lower than 55, and the equalizer will not be activated. The synchronization error is within 2 samples, which does not degrade the performance of the baseband equalizer.

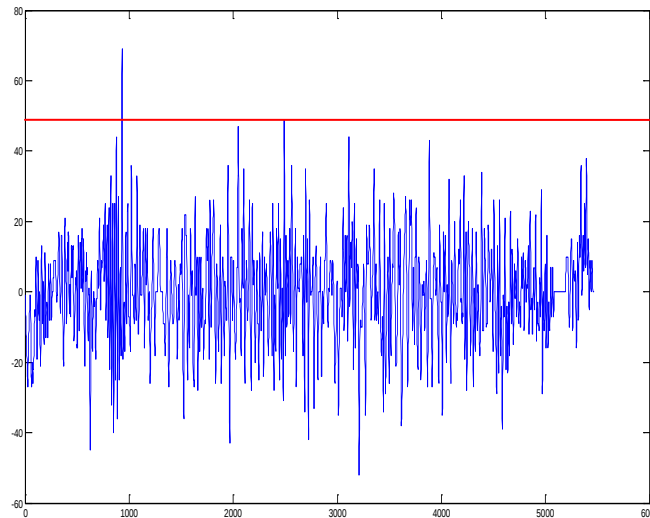


Figure 41: The output of the synchronization component (cross correlator).

6.4.2 LMS baseband Equalization

Figure 40 illustrates the influence of severe multipath fading. To eliminate this distortion, the baseband equalizer is cascaded with the demodulator. Due to the higher computation complexity of the RLS algorithm, the LMS baseband equalizer will be discussed in this section. The structure is shown in Figure 18.

The length of the equalizer depends on the length of the delay spread. That is to say, it depends on the number of symbols from the past and future that can affect the current symbol. The maximum delay spread of channel model D and E is 250ns and 500ns, respectively, which is at most 2/3 of one symbol duration. The final length for the baseband equalizer is defined as 27. This corresponds to 3 times the symbol duration.

As discussed before, LMS has a disadvantage of low convergence speed, and the length of the training sequence (16 bits) used in the equalizer is too short to learn the channel. To compensate for this problem, a large step size is used in the equalizer. The value of 0.0625 is selected from the range (0.001, 0.3), according to simulation results in Matlab. In addition, this value can be expressed as “0.0001” in binary form, for which the multiplication in LMS algorithm can be replaced by a shift operation (4 bits left shift) in the hardware implementation.

After all factors of the LMS equalizer are defined, the equalizer works according to the principles given in (5.12)-(5.14).

In this project, the signal to noise ratio (SNR) is fixed. So, the SNR-BER plot cannot be used to evaluate the performance of the equalizer. Instead, the histogram of the baseband signals and instantaneous BER are used. The following figures show the baseband equalizer performance with different ISI distortion. Figures 42 - 44 show that when multipath fading (channel model D) is not very severe, the equalizer can recover the data correctly. The BER of the equalizer output is 0. The histograms in Figure 44 illustrate that when using the baseband equalizer, the baseband output is equally distributed around the decision level of 0. However, when the equalizer is not used, the baseband output is severely distorted.

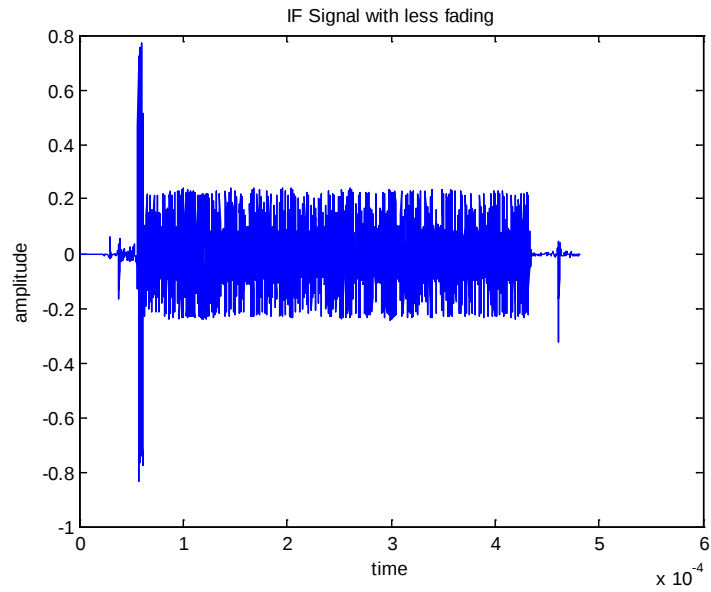


Figure 42: IF signal in a less severe fading channel.

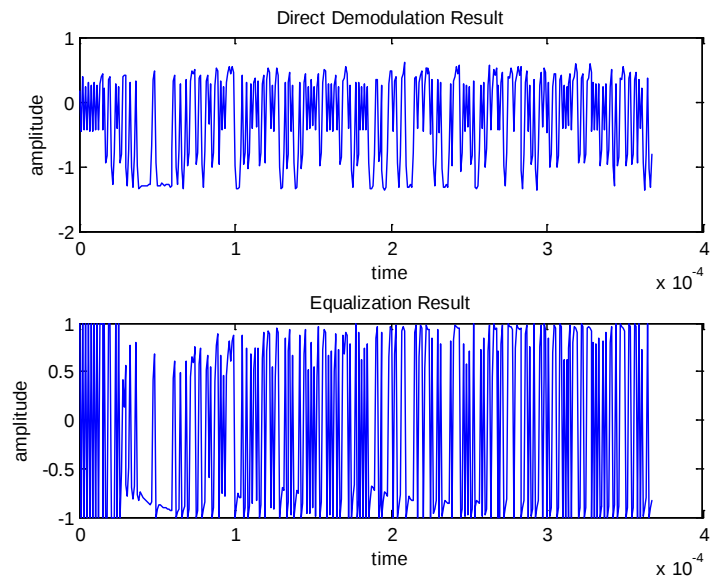


Figure 43: Comparison on baseband signal with or without the baseband equalizer.

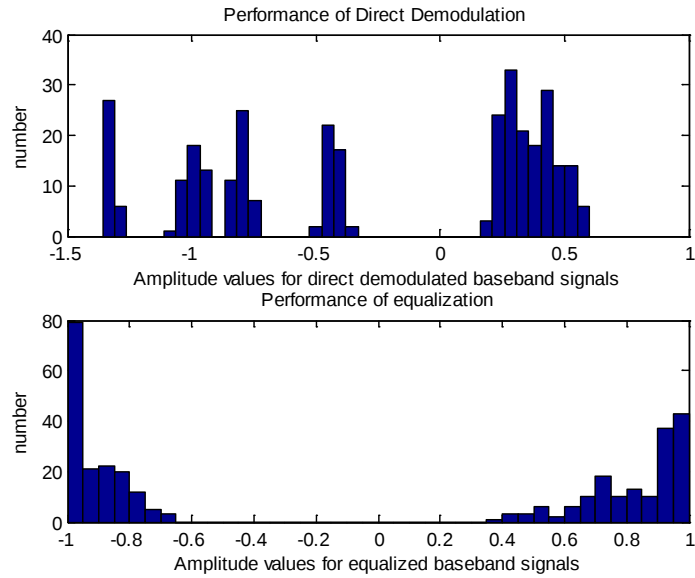


Figure 44: Histogram comparison on the results of Figure 43.

When the distortion is even worse, as shown in Figure 45, the BER of the baseband signals (upper plot Figure 46) is 7.5% without using the equalizer. When the equalizer is used, the BER is six times lower than before. The histograms in Figure 47 also indicate that some negative symbols are interfered to the positive part when the equalizer is not used.

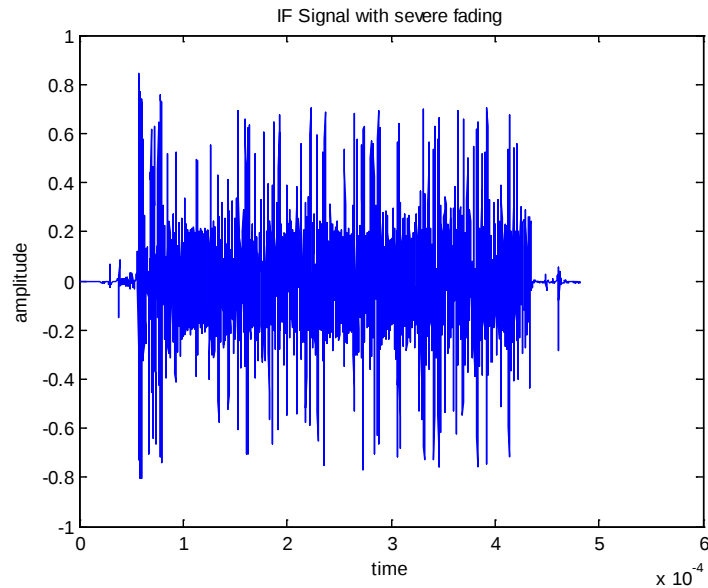


Figure 45: IF signal in a severe multipath fading channel (model E).

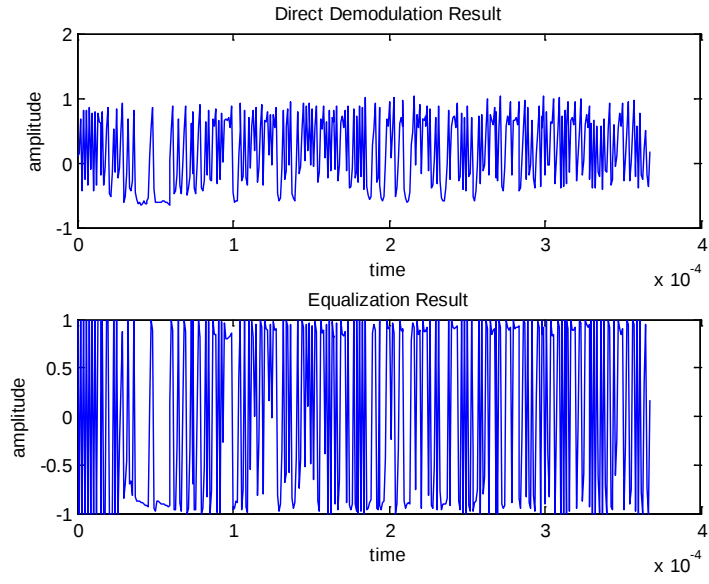


Figure 46: The comparison between with and without the baseband equalizer.

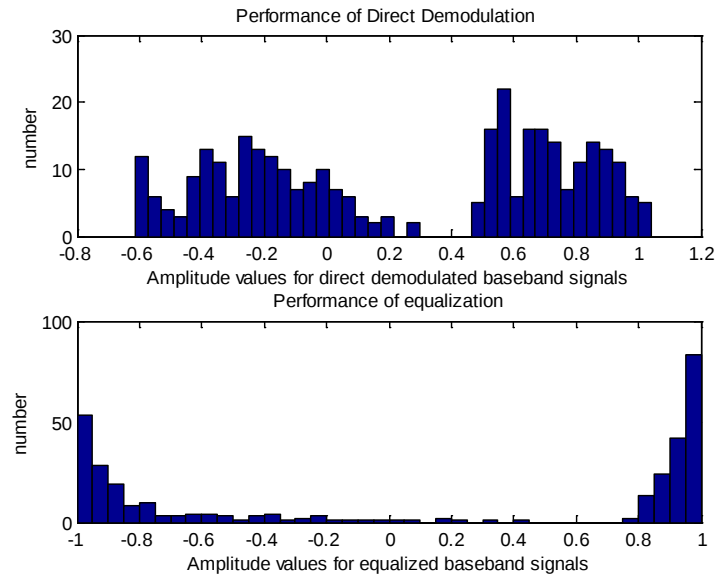


Figure 47: Histogram comparison on the results of Figure 46.

6.5 Limitation of LMS Baseband Equalization for Non-coherent Demodulation

From the results of a large amount of simulations, we have found that the baseband equalizer can only improve the performance to some extent. When the multipath fading is particularly severe, the desired data cannot be recovered from the demodulated signals. Moreover, there are no peaks over the pre-defined threshold of 55 in the synchronization process, as shown in the lower plot of Figure 48 (channel model E). Therefore, the equalizer does not work in such cases.

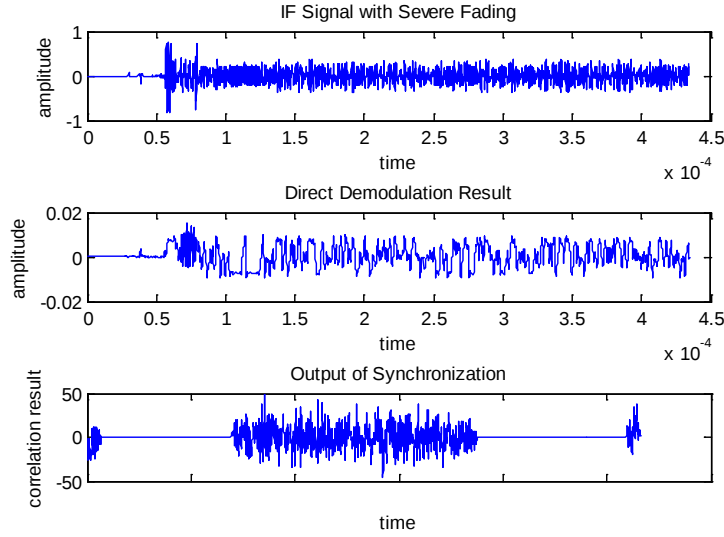


Figure 48: Results when severe ISI occurs.

In order to check the performance of the LMS baseband equalization, the demodulated signals are forced passing through the equalizer by finding the synchronization position manually. However, the result is not good, as expected. An analysis of this problem is presented as follows:

Here we use continuous time signals to do the analysis. In the ideal channel, the noise-free IF signal is written as:

$$y(t) = \cos[\omega_c t + \varphi(t)] \quad (6.6)$$

where ω_c is the center frequency of the carrier, $\varphi(t) = \frac{2\pi f_d}{T} \int_{-\infty}^t \sum_n a_n g(v - nT) dv$ is the information, and t is used to denote the continuous time.



Figure 49: Two-ray channel model.

When a 2-ray channel with equal gain is used as the simulation channel as in Figure 49, $y(t)$ can be written as:

$$y(t) = A \cos[\omega_c t + \varphi(t)] + A \cos[\omega_c (t - \tau) + \varphi(t - \tau)] \quad (6.7)$$

The first term of (6.7) is the desired IF signals without any delay. The second term is from the path with delay τ . In the receiver, we apply quadrature detection (shown in Figure 50) to demodulate the received signals. The demodulation process of $y(t)$ can be written as (6.8)-(6.10), where τ_0 is used to denote the delay corresponding to 90° rotation.

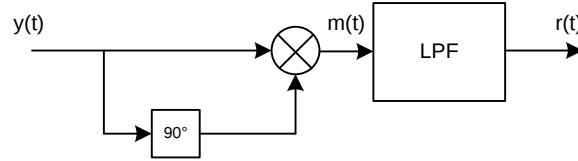


Figure 50: Block diagram of demodulation system.

$$\begin{aligned} m(t) &= y(t)y(t - \tau_0) \\ m(t) &= \{A \cos[\omega_c t + \varphi(t)] + A \cos[\omega_c (t - \tau) + \varphi(t - \tau)]\} \times \\ &\quad \{A \cos[\omega_c (t - \tau_0) + \varphi(t - \tau_0)] + A \cos[\omega_c (t - \tau - \tau_0) + \varphi(t - \tau - \tau_0)]\} \end{aligned} \quad (6.8)$$

After passing through LPF, the baseband signal $r(t)$ can be written as:

$$\begin{aligned} r(t) &= A^2 \cos[\omega_c \tau_0 + \varphi(t) - \varphi(t - \tau_0)] + A^2 \cos[\omega_c (\tau_0 + \tau) + \varphi(t) - \varphi(t - \tau_0 - \tau)] / 2 \\ &\quad + A^2 \cos[\omega_c (\tau_0 - \tau) + \varphi(t - \tau) - \varphi(t - \tau_0)] / 2 \end{aligned} \quad (6.9)$$

Since $\omega_c \tau_0 = \frac{\pi}{2}$, (6.9) can be written as:

$$\begin{aligned} r(t) &= -A^2 \sin[\varphi(t) - \varphi(t - \tau_0)] + A^2 \cos[\omega_c (\tau_0 + \tau) + \varphi(t) - \varphi(t - \tau_0 - \tau)] / 2 \\ &\quad + A^2 \cos[\omega_c (\tau_0 - \tau) + \varphi(t - \tau) - \varphi(t - \tau_0)] / 2 \end{aligned} \quad (6.10)$$

If there is no multipath fading, (6.10) only contains the first term. According to (6.3), the demodulation results are decided by the sign of the first term. However, when multipath fading exists, the results will be interfered by nonlinear effects, as the second and the third terms in

(6.10), and this is non-negligible. So, when the fading is smaller, the nonlinear terms can be approximated as a linear effect which can be effectively fixed by a linear equalizer. But, when the delay spread is longer, the linear baseband equalizer is not so effective to solve the nonlinear interference introduced by the quadrature demodulation.

7. Hardware Implementation and Verification

7.1 External PCB Board

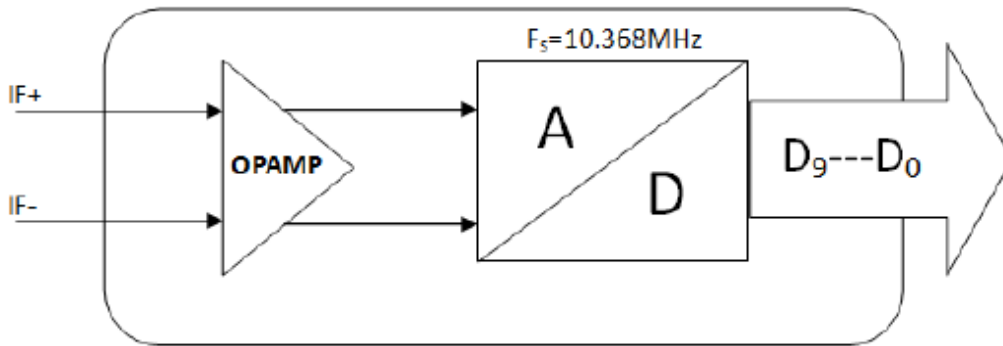


Figure 51: Block diagram of external PCB.

As has been shown in the system diagram, before the FPGA there is an extra board as in Figure 51 for converting the analog IF signal to digital form for digital signal processing within the FPGA. The analog IF input signal from the DECT system as shown in the above diagram is in the form of differential signaling. Differential signaling is a method of transmitting information electrically by means of two complementary signals sent on two separate wires [22]. Due to its inherent resistance to external noise, differential signaling is becoming popular in high speed data acquisition, and nowadays high speed and high accurate ADC's inputs are differential. Therefore, to achieve the noise resistance of differential signaling, fully differential amplifier and differential analog to digital converter are chosen for the analog to digital conversion.

7.1.1 Differential amplifier

A fully differential amplifier is similar to a standard voltage feedback operational amplifier, see Figure 52. Both types of amplifier have differential inputs. But fully differential amplifiers have differential outputs, while a standard operational amplifier's output is single ended. The output of a fully differential amplifier can be controlled by the output common mode voltage (V_{ocm}), which could be controlled independently of the differential input voltage. The usage of V_{ocm} input to the amplifier is to set the output common mode voltage. There are several advantages by using a fully differential amplifier: [23]



Figure 52: Fully differential amplifier versus standard operational amplifier [23].

- Increased noise immunity: unavoidably, when signals are transmitted from one place to another, noise is coupled into the wires. In a differential system, keeping the wires as close as possible to one another makes the noise coupled into the signals as a common mode voltage. Since the differential signaling rejects common mode voltages, the system is more immune to external noise [23].
- Twice output voltage swing: because of the change in phase between the differential outputs, the output voltage swing increases by a factor of two over the standard single ended output with the same voltage swing. Figure 53 [23] illustrates this condition, where V_{od} is the output voltage, defined as:

$$V_{od} = (V_{out+}) - (V_{out-}) \quad (7.1)$$

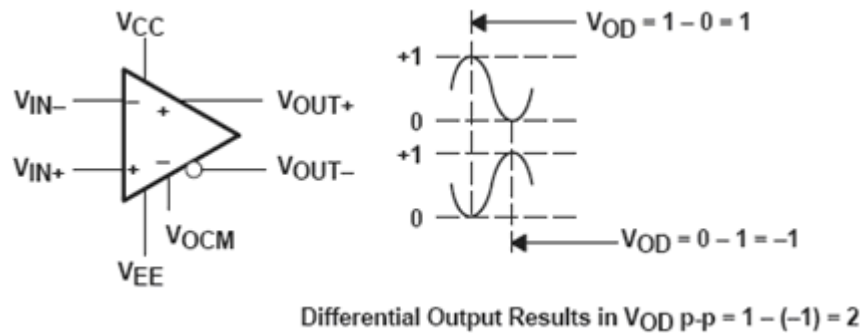


Figure 53: Fully differential amplifier has double output voltage swing.

- Besides all mentioned above, the fully differential amplifier could reduce the even order harmonics and is ideal for low voltage systems, thanks to the increased output voltage swing.

In the application of the fully differential amplifier, there could be two possible feedback paths, one for each side. Figure 54 shows the most typical configuration of a fully differential amplifier with negative feedback to control the gain. In a real design, it is important to keep the two paths symmetric. If we define the differential input as V_{id} and differential output as V_{od} then:

$$V_{id} = (V_{in+}) - (V_{in-}) \quad (7.2)$$

$$V_{od} = (V_{out+}) - (V_{out-}) \quad (7.3)$$

If the feedback loops are symmetry, then

$$V_{od} = (V_{out+}) - (V_{out-}) = A \times V_{id} \quad (7.4)$$

where $A = \frac{R_f}{R_g}$

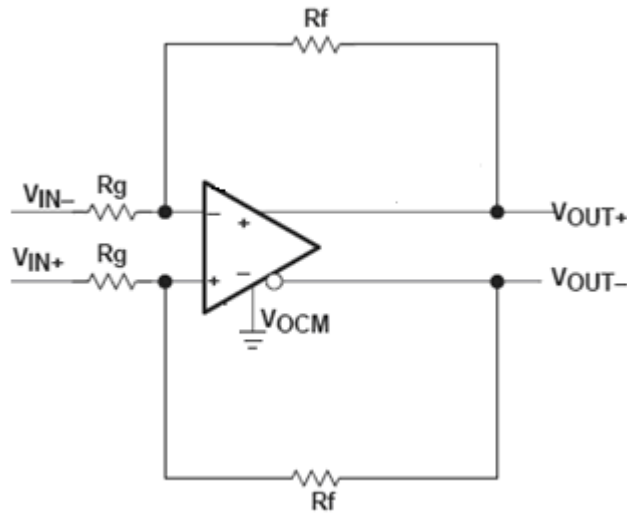


Figure 54: Typical connection of a fully differential amplifier.

In this project, as the input range of the IF signal is from -0.3 V to 0.3 V and the desired input range for the ADC is from -1 V to 1 V, R_f s are set to be 3.3 k Ω and R_g s to be 1 k Ω to have a 3 times amplification.

7.1.2 Differential ADC

As the advantages of differential signaling have been stated in previous pages, a differential input ADC is chosen to be used in the external PCB. The pin allocation is shown in Figure 55 [24].

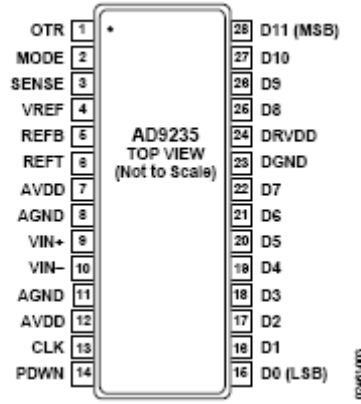


Figure 55: Pin allocation of AD9235.

As we can see, there are 12 bits for digital outputs. But only the first 10 bits are used in this design due to the limitation of I/Os of the FPGA development board. Also, 10 bits resolution has been proved to be sufficient according to Matlab simulation. The output code could be expressed by the following equation [24]:

$$Digit = \frac{2^n ((Vin+) - (Vin-))}{2V_{REF}} \quad (7.5)$$

The pin function descriptions are shown in Table 3 [24].

Pin No.	Mnemonic	Description
1	OTR	Out-of-Range Indicator.
2	MODE	Data Format and Clock Duty Cycle Stabilizer (DCS) Mode Selection.
3	SENSE	Reference Mode Selection.
4	VREF	Voltage Reference Input/Output.
5	REFB	Differential Reference (-).
6	REFT	Differential Reference (+).
7, 12	AVDD	Analog Power Supply.
8, 11	AGND	Analog Ground.
9	VIN+	Analog Input Pin (+).
10	VIN-	Analog Input Pin (-).
13	CLK	Clock Input Pin.
14	PDWN	Power-Down Function Selection (Active High).
15 to 22, 25 to 28	D0 (LSB) to D11 (MSB)	Data Output Bits.
23	DGND	Digital Output Ground.
24	DRVDD	Digital Output Driver Supply.

Table 3: Pin function of AD9235.

The MODE pin is a multi level input that controls the data format and duty cycle stabilizer state. The mode selections are stated in Table 4. In this design, the MOED pin is connected to 2/3 AVDD to get a two's complement output and enabled duty cycle stabilizer.

MODE Voltage	Data Format	Duty Cycle Stabilizer
AVDD	Twos Complement	Disabled
2/3 AVDD	Twos Complement	Enabled
1/3 AVDD	Offset Binary	Enabled
AGND (Default)	Offset Binary	Disabled

Table 4: MODE pin configuration of AD9235.

SENSE pin is used to configure the reference voltage into one of the four possible states which are shown in Table 5. To get a stable internal voltage ladder and 2V peak-to-peak voltage, where the input voltage is from -1 to 1 V, the SENSE pin is connected to ground to select the Internal Fixed Reference mode.

Selected Mode	SENSE Voltage	Internal Switch Position	Resulting VREF (V)	Resulting Differential Span (V p-p)
External Reference	AVDD	N/A	N/A	$2 \times \text{External Reference}$
Internal Fixed Reference	VREF	SENSE	0.5	1.0
Programmable Reference	0.2 V to VREF	SENSE	$0.5 \times (1 + R2/R1)$	$2 \times \text{VREF}$ (See Figure 40)
Internal Fixed Reference	AGND to 0.2 V	Internal Divider	1.0	2.0

Table 5: SENSE and VRET pin configuration of AD9235.

Further, VREF is grounded, and CLK, as the clock input pin used as sampling frequency, is connected to 10.368MHz which is taken from the DECT system to get synchronized with the system. Another reason of choosing 10.368MHz as system clock is that it is easy to do the demodulation. As has been shown in previous chapters, the demodulation is to multiply the signal with itself delayed. The demodulation delay equals to 1/3 of the symbol period, while the clock frequency = 10.368MHz = 9/symbol period, which makes the demodulation easy to implement (see equation (6.4)). The 10bits digital outputs are connected to the FPGA board by using an I/O header.

7.2 FPGA Development Board

The selection of FPGA development board is narrowed down to Low Power Reference Platform (LPRP) from Arrow Electronics, Inc [25] because it was pre-purchased by the company and its powerfulness by means of FPGA device. This board contains of a low power Altera Cyclone III FPGA EP3C25, onboard battery for power supply, 12 user I/Os, an ADC and a DAC (which was not used), two clocks and several push buttons and LEDs.

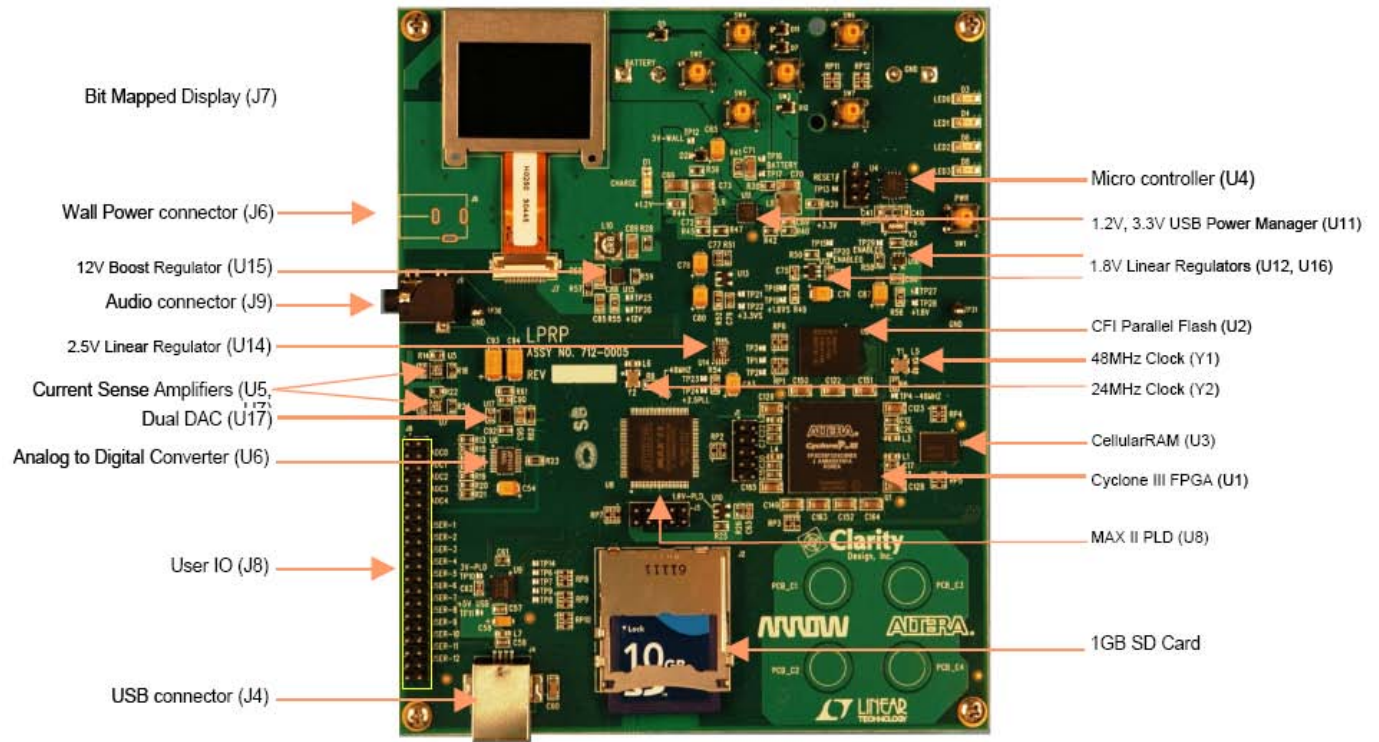


Figure 56: A layout of LPRP FPGA development board.

Figure 56 shows the layout of the LPRP FPGA development board. It features the devices listed below [26].

The EP3C25 Cyclone III low cost and low power FPGA can be seen in the middle right of the board, and is marked as U1 in the figure. This FPGA is used as the main processor in this design. It finishes the tasks including filtering, demodulation, synchronization and equalization. This FPGA contains large amount of resources, and is enough for a big DSP design and low power device which employs 65-nm processing. More features and resources of this Cyclone III FPGA are introduced in later pages.

- Power converters from Linear Technology, including the LT3455 which works as a power supply for all the components.
- Cellular RAM, NOR Flash and removable SD memory devices that could all be used as external memory. But since this design does need much memory resources, only the onboard memory of the FPGA is used such as the storage of filter coefficients.
- 1.1 inch monochrome grayscale display: this display could be used as an output of any signals and there is a software driver for this device from the board package.

- Audio CODEC and headphone amplifier and ADC which have not been used in this design.
- Buttons & LEDs: Several buttons have been used, such as Reset, Start, Mode Switch, etc. All LEDs have been used as indicators.
- General Purpose I/O header: This I/O header has been used as the main input and output for the FPGA board including 10 bits input from the ADC, 10 MHz clock input and demodulated data output.

7.2.1 EP3C25

EP3C25 Cyclone III FPGA works as the main processor in this design, and it is in charge of most of the digital signal processing and control tasks. This Cyclone III chip features cost-optimized and memory-rich minimized power consumption and is therefore ideal for wireless communication systems. The device feature of this FPGA has been shown in following Table 6 [27]. As we can see, it has rich resources and is sufficient for such a medium sized DSP and controlling design. Altera supplies an embedded soft core solution Nios, which enables a combination of VHDL design and C programming.

Feature	EP3C25
Logic Elements	24,624
Memory (kb)	594
Multipliers	66
PLLS	4
Global Clock Networks	20

Table 6: Cyclone III FPGA features.

7.3 Basic DSP Arithmetic Implementation on FPGA

7.3.1 Number Representation

An integer or fraction number could be represented by fixed point or floating point form, and it is preferable to decide it in the early stage of the project. Generally, fixed point representation has higher speed and lower cost, while floating point implementation has higher dynamic range and no need for scaling which is more attractive for complex systems. To decrease the use of the FPGA resource, we choose to use fixed point representation in this design, which is also more suitable for FPGA.

The data processed by the FPGA, such as the input signal, the coefficients of the filter and the output signal could be positive or negative. Accordingly, it is necessary to employ a signed number for expressing all the data in FPGA. In signed system, the magnitude and sign are represented by separate bits. The first bit (i.e. the MSB) represents the sign and the remaining bits the magnitude in this design. So a decimal number X represented by N bits in signed form is given by

$$X = \begin{cases} \sum_{n=0}^{N-2} x(n)2^n & X \geq 0 \\ -\sum_{n=0}^{N-2} x(n)2^n & X < 0 \end{cases} \quad (7.6)$$

where $x(n)$ is the n th bit of the binary number x (could be either 0 or 1). The digit $x(0)$ is the least significant bit (LSB) and the digit $x(N-1)$ is the most significant bit (MSB) which represents the sign of number X . The dynamic range of this representation is $[-(2^{N-1}-1), 2^{N-1}-1]$. The advantage of this signed representation is simplified prevention of overflows, but the disadvantage is that the addition must be split depending on which operand is larger.

Two's complement representation is one of the signed representations and the most popular signed numbering system for DSP use. This is because it is possible to add several signed numbers, and any overflow could be ignored as long as the sum is within N bits' range. Further, it is easy to do the subtraction in this arithmetic operation, where it is just adding the inverted subtrahend number to the subtrahend. Two's complement numbers can also be used for implementing modulo 2^N arithmetic operation, which is useful for multiplication and division. A two's complement representation of a signed integer is given by:

$$X = \begin{cases} \sum_{n=0}^{N-2} x(n)2^n & X \geq 0 \\ -2^{N-1} + \sum_{n=0}^{N-2} x(n)2^n & X < 0 \end{cases} \quad (7.7)$$

Plus, two's complement number is included in the VHDL and Verilog library which makes it easier to read and program.

7.3.2 Adder/Subtractor

Addition and subtraction is the most common computation block for DSP applications such as filters. A basic N bits adder/subtractor consists of N full adders (FA). A full adder can be expressed by the following Boolean equations:

$$SUM(n) = X(n) \quad \text{XOR} \quad Y(n) \quad \text{XOR} \quad C(n) \quad (7.8)$$

which define the sum bit and X and Y are the operand. The carry bit C can be expressed by:

$$C(n+1) = (X(n) \quad \text{AND} \quad Y(n)) \quad \text{OR} \quad (X(n) \quad \text{AND} \quad C(n)) \quad \text{OR} \quad (Y(n) \quad \text{AND} \quad C(n)) \quad (7.9)$$

According to Altera EDA tool Quartus II 9.0 handbook [14], the synthesis tool could choose between each structure and find the most suitable structure to meet the timing and power requirement.

7.3.3 Multiplier

Multiplication is another widely used arithmetic operation in DSP applications. Most modern FPGAs have embedded multiplier blocks that are optimized to support multiplier intensive applications such as FIR filters, FFT and encoders. These embedded multipliers are dedicatedly designed for fast multiplication, resulting in efficient resource utilization, improved performance and data throughput as compared to using logic blocks. Similarly to the latest device from Altera, Cyclone III devices offer up to 288 embedded multiplier blocks and supports individual 18-bit $\times$ 18-bit multiplications per block, or two individual 9-bit $\times$ 9-bit multiplications per block. Besides the embedded multipliers, Cyclone III contains a combination of on chip resources and external interfaces that help increasing the performance, reduce system cost, and lower the power consumption of digital signal processing. It focuses on optimizing the device for applications and benefits from an abundance of parallel processing resources, including video and image processing, intermediate frequency modems used in wireless communication systems, and multi-channel communications and video systems [27].

To infer the multipliers for Altera devices, there is a built-in library called Megafunction in Quartus II software. From there, MegaWizard Plug-In Manager could help to create the Megafunctions in the Quartus II GUI that instantiate a multiplier. The MegaWizard Plug-In Manager provides a GUI to customize and parameterize Megafunctions, and ensures that all Megafunction parameters are properly configured [14]. The MegaWizard Plug-In Manager instantiates the Megafunction with the correct parameters and generates a Megafunction variation file in HDL. Besides, the synthesis tools will look for multipliers and convert them to Megafunctions, or may map them directly to built-in multipliers.

7.4 Band-Pass Filter and Lowpass Filter

After the AD conversion, we found a big offset which is around 0.7V after amplification in the real implementation. According to simulation and real tests, this DC offset could affect the demodulation very severely as discussed before. By analysis, it was defined that this DC offset is from one of the differential input signals, since the two wires are not symmetrical to each other. To solve this, a band-pass filter was used to filter out the DC offset as well as the out of band noise. According to simulation, a 128 order bandpass filter was implemented, as the DC offset could severely affect the performance. An FIR filter was chosen to implement this band-pass filter as it has better stability properties than IIR filters. There are several structures for FIR filter implementation as stated below:

7.4.1 Finite Impulse Response filter basic structure

The L-order FIR filter with an input of $x[n]$ and output of $y[n]$ is graphically interpreted as in Figure 57.

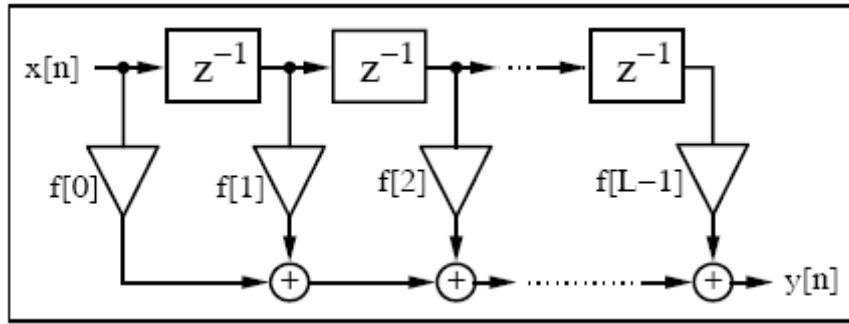


Figure 57: A structure of FIR filter

The filter consists of a collection of taps (i.e. $f[n]$), delay components, adders and multipliers. One of the operands to the multipliers is the FIR coefficients (the taps), and the other is the delayed input samples.

To implement this structure with a digital circuit, L multiplications and $L-1$ additions are needed. So to implement a 64-tap's filter, which is quite common in real life, 64 multipliers and 63 adders will be needed. An FIR filter, on the other hand, is quite costly in area and timing. Some options to improve this structure to reduce its complexity are stated in following parts.

7.4.2 Transposed FIR filter

An FIR filter model could also be expressed by another structure as shown in Figure 58 [10].

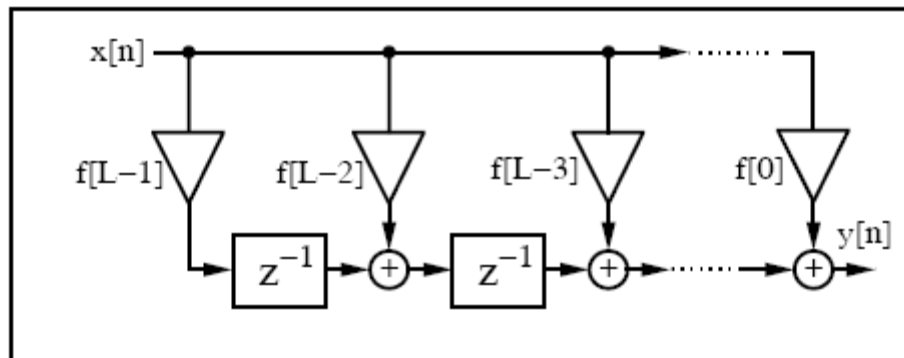


Figure 58: Structure of transposed FIR filter.

This structure is derived from the standard structure of Figure 57 by exchanging the input and output; inverting the direction of signal flow; and by substituting an adder by a fork.

This model is called the transposed FIR filter. This structure is the preferred implementation of an FIR filter. The advantage of this one over the previous one is that an extra shift register for $x[n]$ is not needed, and there is no need for an extra pipeline stage for the adder tree of the products. But for this structure, the number of multiplications and additions is still very high.

7.4.3 Symmetric FIR filters

The center of an FIR's impulse response is an important point of symmetry. Such a linear and time invariant causal system as described above is a causal filter. This filter's output only depends on the past and present inputs.

Since the linear phase filter coefficients are symmetrical or asymmetrical, the number of the multiplications can be reduced by first adding pairs of samples and then perform the multiplication. Such a structure can efficiently reduce the number of multipliers L , as shown in Figure 59 [10]. This symmetric architecture of FIR filter implementation has a multiplier budget per filter cycle exactly half of that in the previous cases (i.e. L versus $L/2$), while the number of adders remains unchanged at $L-1$.

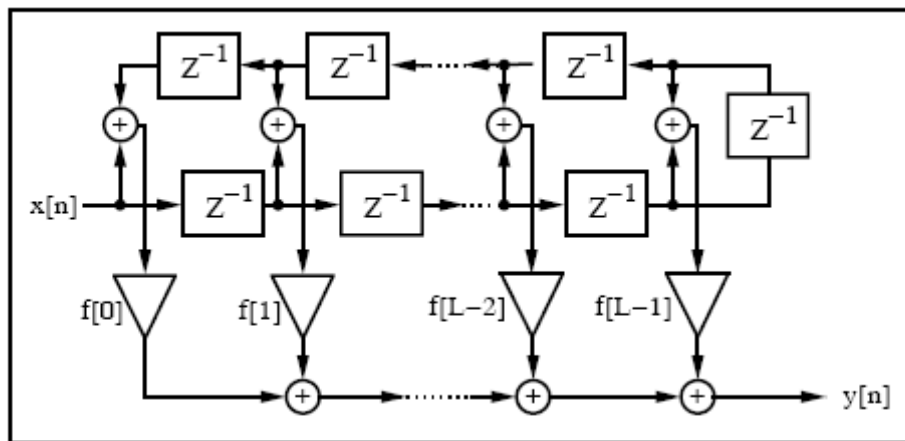


Figure 59: Structure of symmetric FIR filter.

The coefficients are generated, for example, by using Matlab's filter design tools. Since the coefficients are symmetric, a symmetric FIR filter structure (as stated in the figure above) is used to implement this band-pass filter.

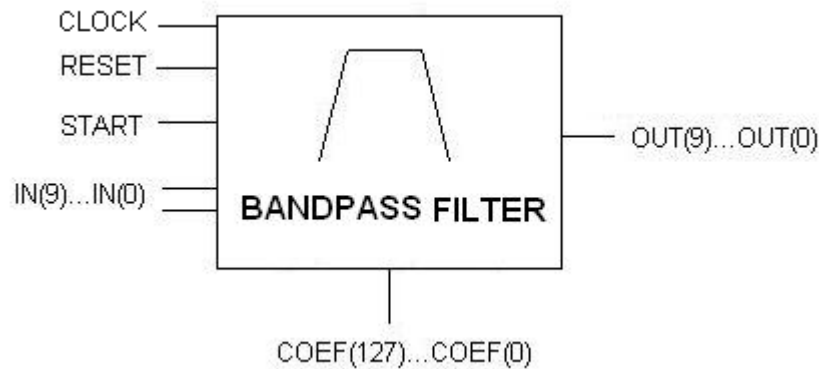


Figure 60: Schematic of a bandpass filter implementation.

A diagram of this pre-filter is shown in Figure 60. It has clock, reset, and start as control inputs; and 10 bits input and output signals. The FIR filter has 128 coefficients.

CLOCK: system clock connects to 10.368 MHz.

RESET: synchronous reset, resets the filter to an initial state where all delay paths are set to zeros.

START: if this input is active, the filter will stop working and keep all the delayed inputs.

IN, OUT: input, output signal of the filter.

COEF: 128 coefficient inputs to the filter, each coefficient has 16 bits.

This implementation of the symmetric FIR bandpass filter contains $L/2$ multipliers and $L-1$ adders, where L equals 128 in this case. The lowpass filter for demodulation uses the same FIR structure as the bandpass filter, but with a lower order of $L=32$. The filter coefficients are generated by Matlab.

7.5 Demodulation

A non-coherent quadrature demodulation scheme is chosen for implementation in this design. The quadrature demodulator structure is shown in Figure 61. As shown below, it is simple to implement, as it only contains one multiplier and a delay path for the down conversion path, and a 32 order lowpass filter after.

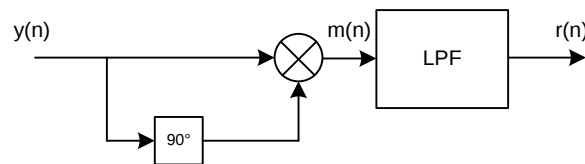


Figure 61: Structure of quadrature demodulation.

The symbol of this demodulator is shown in Figure 62. It has clock, reset and start for control inputs that reset the demodulator to an initial state and stop the demodulator, 10 bits inputs and outputs, and 32 coefficients inputs.

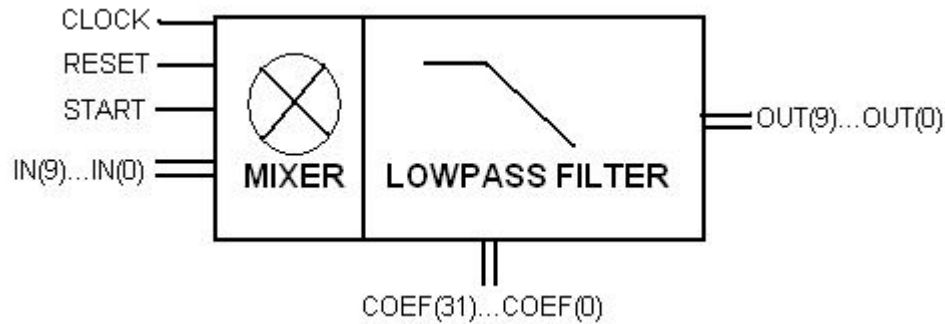


Figure 62: Symbol diagram of the demodulator.

7.6 Synchronization

To find the starting point of the equalization, synchronization needs to be done based on the demodulated signal. Every DECT slot contains 480 bits as shown in Figure 1.

The first 16 bits of the preamble consist of '1010101010101010', which is used for clock recovery and frequency offset compensation. The last 16 bit pattern '1110100110001010' is the slot synchronization in the digital field. Despite its autocorrelation function being far from optimum (it is not a Dirac delta function), it is a reasonable choice for the synchronization purpose.

Instead of taking an cross-correlation by doing several multiplications, a simplified digit correlation is used for synchronization. This method works like this: if the number in the pattern in the synch word is '1' then the corresponding positioned number in the demodulated digit is added to the sum value. On the other hand, if the number in the pattern is '0', then the corresponding demodulated digit is subtracted from the sum value. The figure below shows result of the digit correlation between the demodulated result and the synch word. As we can see, there is an obvious peak value in the beginning of the plot in Figure 63. By doing this, the ideal peak value is 72, because one symbol contains 9 samples and there are 8 times of additions according to the pattern above. Based on a real test, we choose 55 as the synchronization threshold.

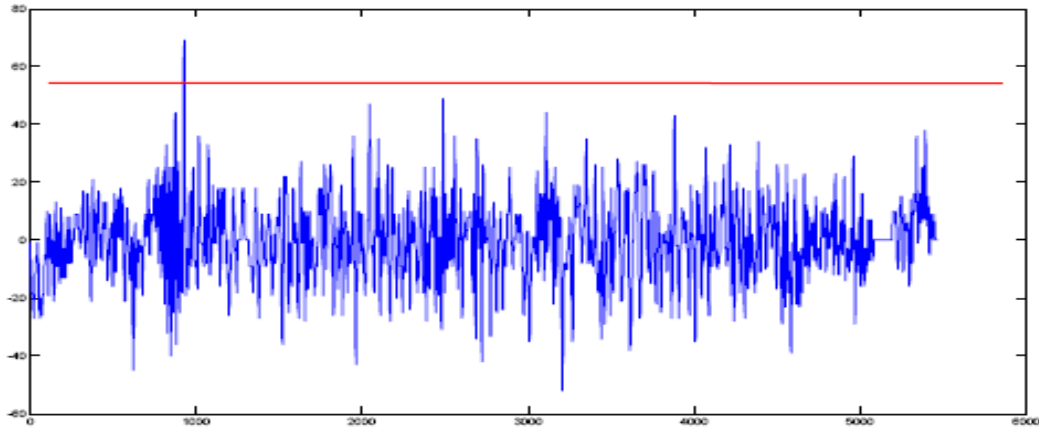


Figure 63: Digit autocorrelation of demodulated signal and training pattern.

Therefore, there is no need for multiplications in this synchronization, and only a sequence of accumulators are needed to perform this task. The symbol of synchronization is shown in Figure 64. The synchronization takes the demodulated signal from the demodulator as input, and outputs the synchronized indicator. The synchronization indicator will activate if the synch word is found and it remains activated for one slot time.

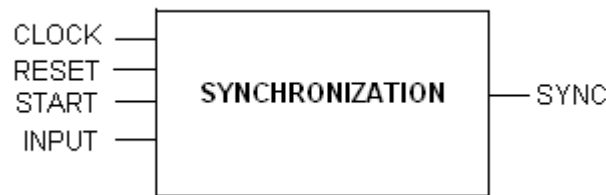


Figure 64: Symbol diagram of synchronization.

7.7 LMS Equalizer

The coefficients of the filters stated above (i.e. FIR filters or IIR filters) do not change over time. But in this project, the channel slowly changes over time. This requires that the filter coefficients are adjusted over time, depending on the input signal as the channel changes. Due to the adjustment and updating of the filter coefficients, implementation of an adaptive filter is much more complex than LTI filters. But recently, journal publications such as IEEE Transactions on signal processing show the feasibility [28] and stability of implementation of LMS filter in FPGA [29].

An LMS equalizer is used as a channel equalizer in this project. Unlike fixed coefficient filters that have fixed weights or coefficients, LMS equalizer needs an algorithm to update the filter coefficients. Among all kinds of adaptive algorithms, the LMS algorithm is known for its simplicity, and better performance in different running environments [30]. On the other hand, Recursive Least Squares algorithm is another one widely used in adaptive filter. Although RLS

algorithm has better convergence speed, LMS requires much less computation complexity as has been discussed in previous chapters.

According to the LMS adaptive updating algorithm (equations (5.12) - (5.14)), a possible implementation represented by a block diagram is shown in Figure 65 [10].

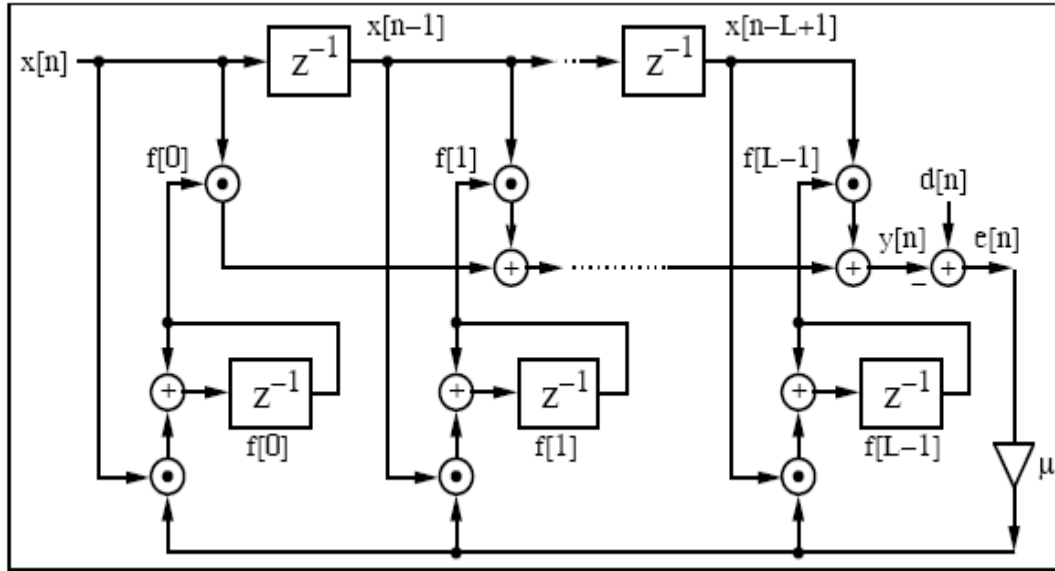


Figure 65: LMS equalizer implementation.

This LMS algorithm implementation uses an FIR filter structure. As shown in the figure above, the main component of the filter consists of $L-1$ delay components, L coefficients and coefficients update blocks, where L is the length of the equalizer. The delay registers can be implemented by D Flip-Flops. The filter output $y[n]$ is subtracted from the desired signal $d[n]$ to produce an error signal. The error signal is fed back to the weight updating components to produce the next sets of filter coefficients. The weight updating components perform logics calculation according to the equation. The operations needed in the algorithm include two multiplications and one addition.

The step size parameter μ is a decimal number, and multiplying by a decimal number is equivalent of dividing by its reciprocal. Generally, implementation of division is a complicated algorithm. In order to avoid implementing complicated and area consuming division, or floating-point multiplication, arithmetic shift operation can be used instead to simplify and shorten the most critical path of the design. The arithmetic shift operation on a 2's complement integer shifts the number n bits to the right, while preserving the sign bit (the most significant bit). By shifting the number n bits to the right, it is equivalent to multiplying this number by 2^{-n} . Therefore, in order to achieve simplicity and feasibility, the multiplication by μ can be replaced by an arithmetic shift operation if μ is restricted to be $\mu = 2^{-n}$, where n is a positive integer. How to choose the number of μ has been discussed in other chapters.

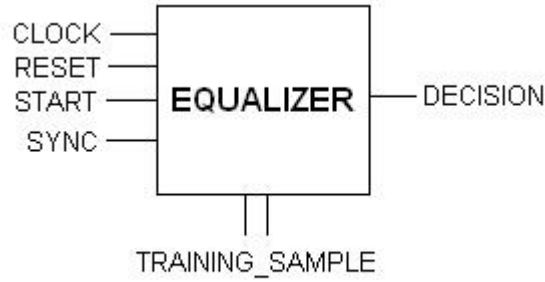


Figure 66: Symbol diagram of the baseband equalizer.

After the synchronization component finds the beginning of the synch word, the LMS equalizer will be activated. This component symbol as shown in Figure 66 takes a demodulated signal with the exact starting position of the training sequence as input and updates the coefficients according to the adaptive equation. It has clock, reset and start as control inputs, and sync to indicate the starting of equalization. Once the equalizer is activated, the equalizer will start equalization which is divided into two modes.

The equalizer starts updating the coefficients during the training mode, which updates according to the training sequence that is pre-stored and calculates the error. After 144 clock cycles which are the length of the training sequence period, the equalizer keeps updating according to the decision directed feedback to calculate the error and continues until the end of one slot. The equalizer outputs the equalized decision digital data.

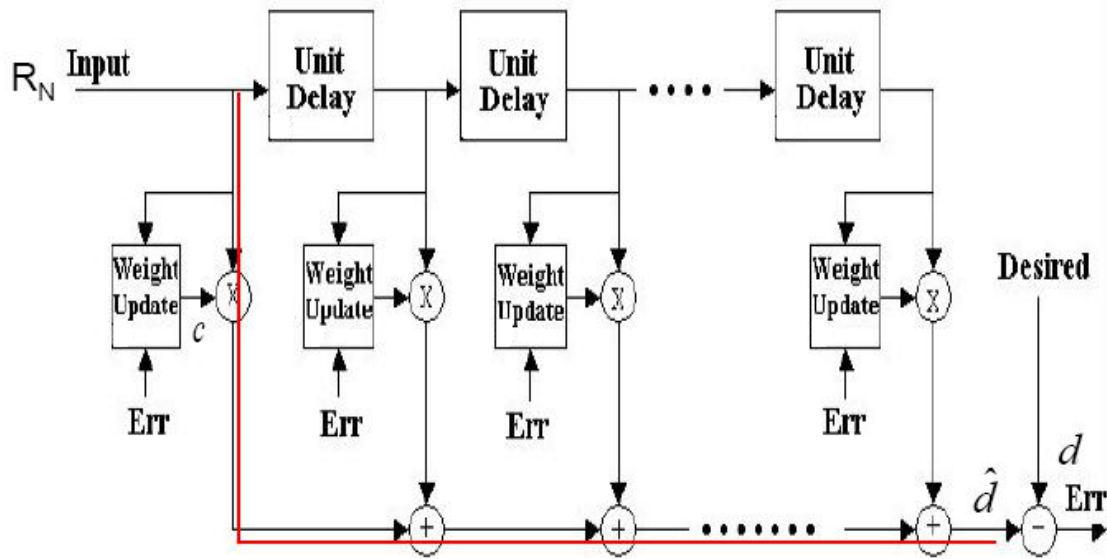


Figure 67: Critical path of LMS equalizer.

The critical path of the equalizer has been marked red in Figure 67, which includes one multiplier and L serial additions, which can be quite a long path for large values of L. This critical path is too long to meet the time requirement of 10.368 MHz according to Quartus II synthesize result. So, a register was inserted in the middle of the accumulator, where it is before the L/2-1 adder. That is to say this equalizer is pipelined into two. After pipelining, there is no timing problem.

7.8 Resource Utilization

Table 7 shows the resource utilization in FPGA.

	Logic Elements	Dedicated Multiplier(9×9)
LMS Equalizer	1036 (4%)	94 (70%)
Demodulator	981 (3%)	23 (17%)
Synch	274 (1%)	0
Pre-filter	1290(5%)	112(80%)
All	17354(60%)	117(89%)

Table 7: Resource utilization of FPGA.

The table shows the most resource consuming components in the FPGA when they are synthesized separately. The LMS equalizer uses a large number of embedded DSP components. But according to a publication [30], the DSP component uses could be halved by a improved structure where this needs to be further modified. As the pre-filter has quite a large number of coefficients, it uses the most resources in the design. But this could be decreased or removed by a modified external ADC board.

Another issue that needs some attention in this table is that the overall use of logic elements are more than the sum up of all four components. This is because the overall use of multipliers is more than the FPGA has. So, half of the multipliers have been implemented by logic elements. Another big use of logic elements, which has not been shown in the table, is the delay registers used to compensate the filters' delay and storing data when the system is searching for the sync word. The length of this component could vary as the equalizer and synchronization changes.

7.9 Simulation and Verification

The design has been simulated by using Modelsim, and a screenshot of the simulation result of the baseband equalizer in channel model E is shown in Figure 68. The Input_tmp and correct are the signals sampled from a real test, and converted to binary format by using Matlab and read into

testbench. Input_tmp signal is the output from the ADC and correct is the transmitted data. Equa_deci is the equalized signal after decision making. Thus, the result can be compared with the transmitted data. But this comparison cannot be done automatically in Modelsim due to small jitters between the transmitted and received signals. This comparison was done in Matlab instead. In the lower part of Figure 68, the equalized signal and error pattern are shown. After the training updating, the baseband signal has been dragged close to 1 and -1, and the errors are close to zero except some small glitches.

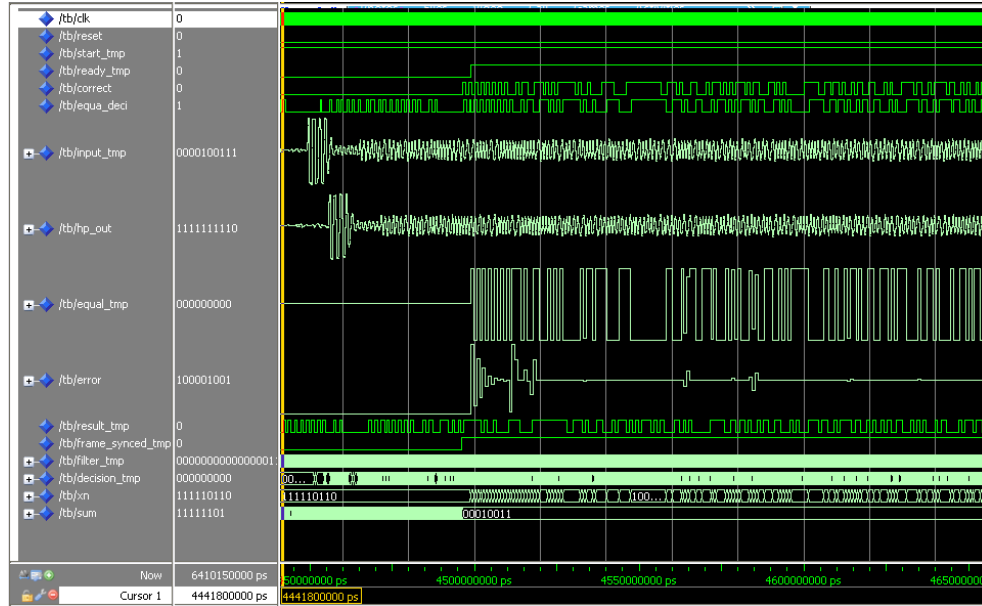


Figure 68: Simulation results in Modelsim (channel model E).

Testbench based verification is used to verify the FPGA design. The term testbench refers to some codes used to create a pre-determined input sequence to a design, and then optionally observing the response [31]. Figure 69 shows a testbench with a design under verification (DUV). The testbench provides inputs to the design and checks the outputs. So for a testbench system, there is no input or output. Moreover, a testbench is not synthesizable and it is just used for simulations.

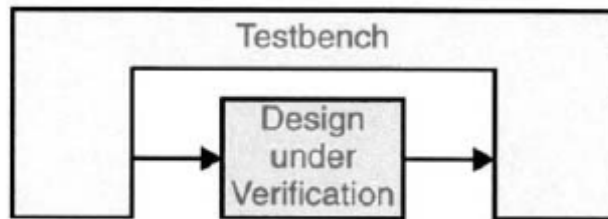


Figure 69: General structure for testbench based verification.

The IF signals which are sampled from the real test, are input into the design by testbench; and then the outputs are compared with the Matlab results. This method of verification was used to verify the demodulation result. To ensure that the synthesized results are equal to the functional description in the RTL level, the netlist file which is generated by Quarus II is also verified by the testbench.

8. Passband Equalization and Analysis of the Result

8.1 Potential Methods for Improving the Equalizer

In Chapter 6, we showed that the demodulation method can introduce a nonlinear variation that cannot be compensated by a linear equalizer, so some modification on this baseband equalizer structure is necessary.

The first possible solution is to replace the demodulator with a coherent demodulator. According to the method of coherent demodulation as described in [35], it will not introduce any nonlinear variations to the results. This method requires a phase lock loop (PLL), for example, Costas Loop as described in [36]. This is the trade-off of using coherent demodulation as the computation complexity is quite high. Besides, the coherent demodulation will not satisfy the pre-requirement of this project, non-coherent demodulation.

The second possible solution is to change the equalizer. According to some publications, there are two potential methods. Firstly, a non-linear equalizer filter can be used. It means that the weights of the transversal equalization filter will be polynomials instead of constants. Obviously, this algorithm is quite complicated and difficult to design, both in Matlab simulation and in hardware implementation. Secondly, a passband equalizer can be an effective way to fix the problems. This will be discussed in detail in Section 8.2.

8.2 Passband Equalization

The multipath channel can be modeled as a linear FIR transversal filter. So the RF signals transmitted in the air are also distorted by linear effects. The down conversion from RF to IF can also be considered as a linear operation. Hence, it can be predicted that a passband equalizer can be a reasonable solution for the multipath fading.

Figure 70 shows a block diagram of a passband equalizer. In this equalizer, the LMS adaptive algorithm is still used. But now the input is the distorted IF signals. After the equalization process, the output is fed into the same non-coherent demodulator. According to the methods of training and tracking for the equalizer, it is very important to find the position of the training sequence. Consequently, synchronization should be done prior to the passband equalization.

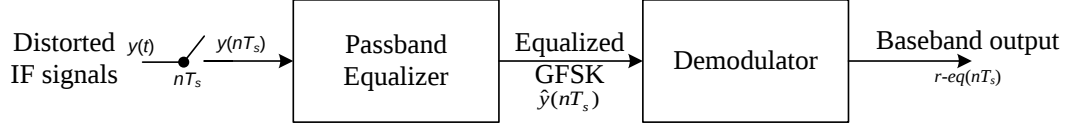


Figure 70: The diagram of passband equalizer.

The syncword is still from the same field of data package. However, a baseband discrete pulse sequence is incompatible now. Instead, the synch pattern discussed in Chapter 6.4.1 must be modulated to passband as depicted in Figure 71. This wave is also used as a training sequence for the passband equalizer.

In the passband equalization, the LMS algorithm is still used. However, in the baseband equalization, the coefficients are updated every 9 samples. On the contrary, the passband equalizer coefficients are updated every sample, which requires more computation than the baseband equalizer during the coefficients updating. One widely known issue is that the LMS algorithm has a feature of low convergence speed. Thus, the equalizer coefficients cannot be sufficiently updated to fix the multipath fading effects with such a short training sequence. Reference [37] provides an idea to achieve a good adaptation by using more than 200 iterations for training. It means that the equalizer would retrain the same training sequence many times until it converges. However, this will introduce a large delay for the system. In our simulation, we find that 20 loops are sufficient to obtain relatively perfect coefficients. Even by using 20 loops, a delay equivalent to half of one time slot is introduced. This is one limitation for further implementation on hardware. This issue and a possible solution are described in Chapter 9.

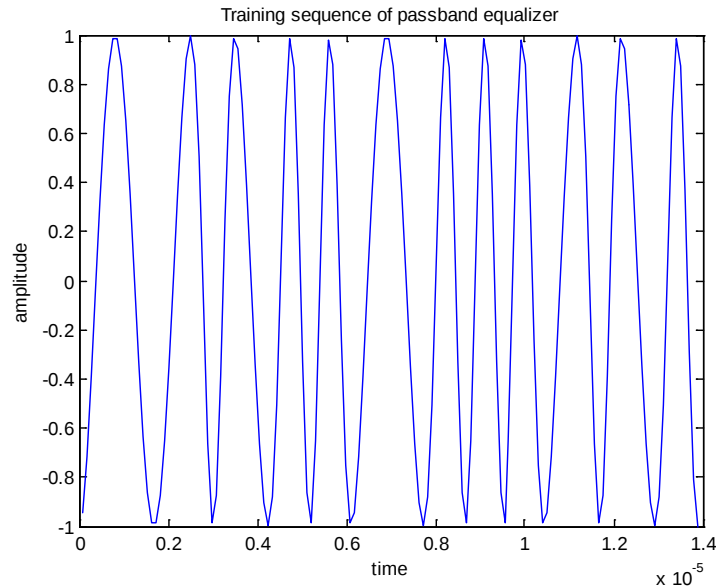


Figure 71: The generated passband training sequence.

8.3 Performance of Passband Equalization

When the modulation index is fixed at 0.5, the performance can be improved significantly by using the passband equalizer, with the training sequence as shown in Figure 71. Figures 72 and 73 depict the performance of the passband equalizer. The IF signal in the upper plot of Figure 72 is distorted severely. It is easy to see that the amplitude and the phase experience a severe distortion, and the desired information cannot be extracted from the demodulation result, as shown in the upper plot of Figure 73. The BER of the baseband signal is over 10%. However, when the passband equalizer is used to filter the IF signal, the distortion is compensated, as shown in the lower plot of Figure 72. The demodulation result of the equalized signal is shown in the lower plot of Figure 73, the instantaneous BER of which is lower than 1%.

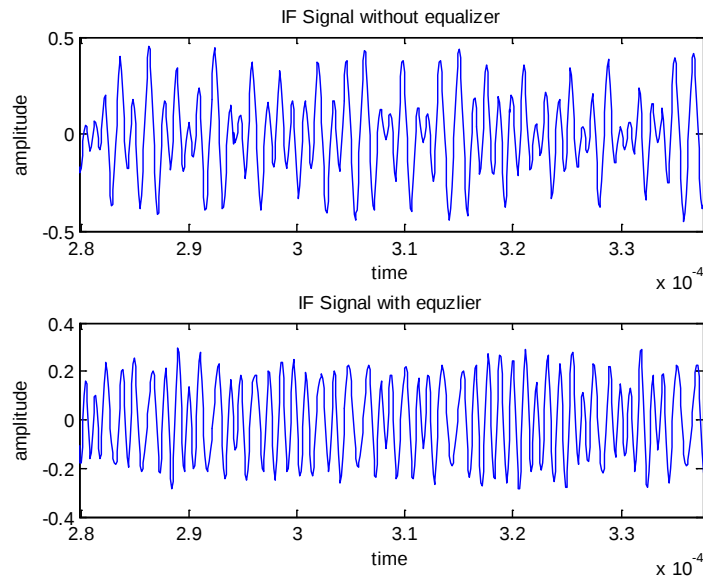


Figure 72: IF signals before and after the passband equalization (channel model E).

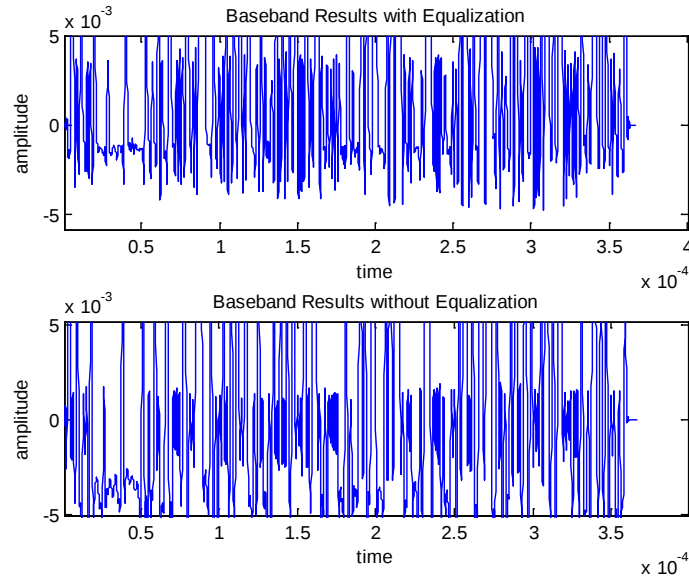


Figure 73: Comparison between the demodulation results of the IF signals in Figure 72.

By a large number of simulations, it has been verified that the performance can be improved significantly when a passband equalizer is used. The average BER can be decreased to 0.05%, which is ten times better than that without using the equalizer. The following figures depict more of the performance of the passband equalizer in the channel with different levels of fading.

Figures 74 - 76 show the passband equalization performance in a severe multipath fading channel. The two plots in Figure 74 are the input and output of the passband equalizer. From its zoom-in version in Figure 75, we see that both the amplitude distortion and the phase distortion are compensated. The lower plot of Figure 76 shows the baseband results after using the passband equalizer. The BER of the baseband signal is decreased from 10% to 0.8% when the equalizer is used.

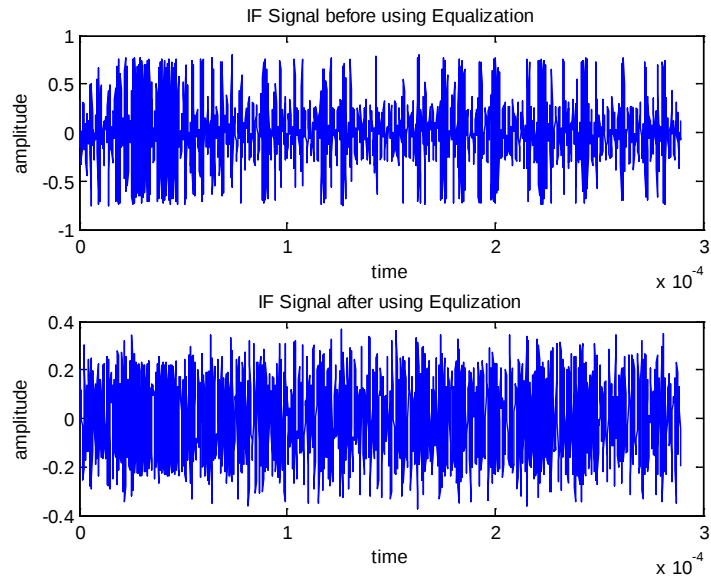


Figure 74: The comparison between with and without using the passband equalizer (channel model E).

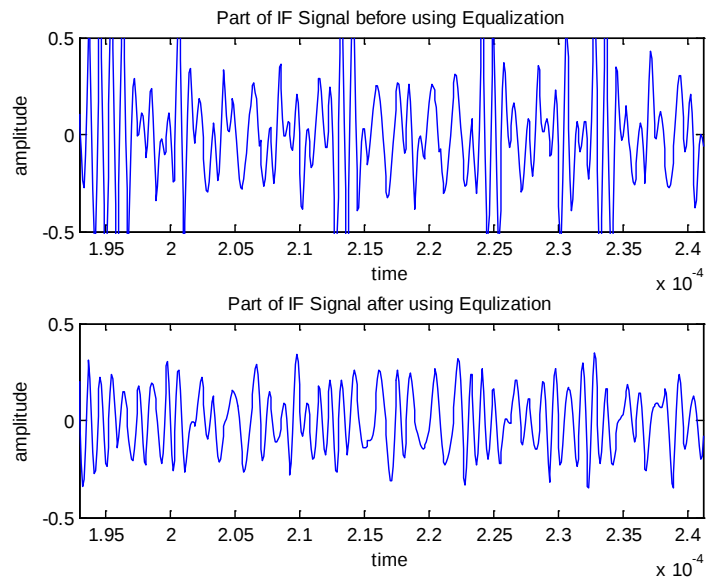


Figure 75: Zoom in version of Figure 74.

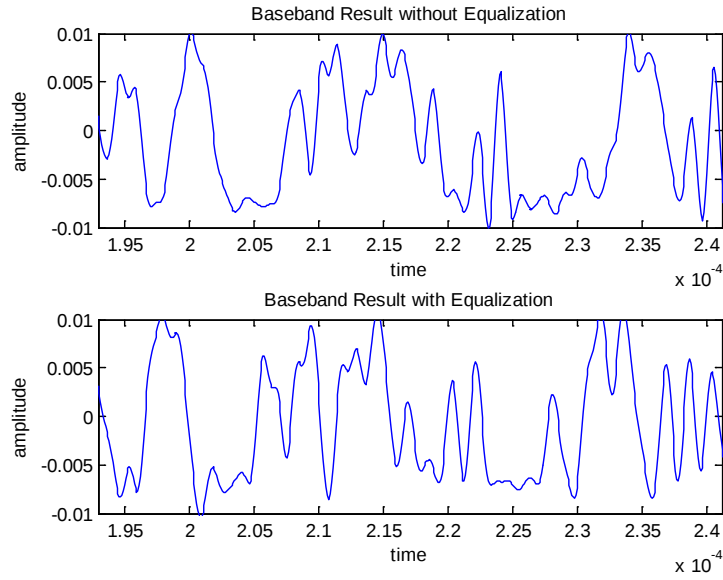


Figure 76: Comparison between the baseband demodulated signals.

Figures 77 - 79 show the performance when particularly severe ISI occurs. Without using the equalizer, the demodulated baseband signal cannot be synchronized and it results in a frame error rate (FER). After using the equalizer, the BER of this slot is decreased to 1%.

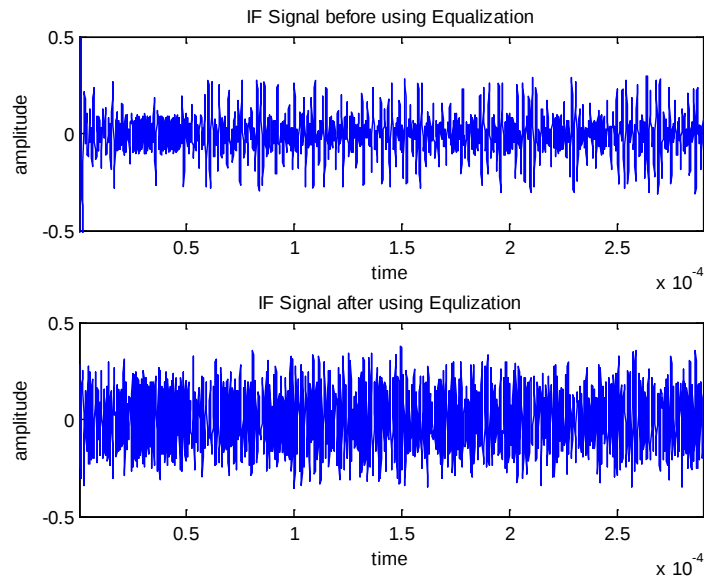


Figure 77: The comparison between with and without using the passband equalizer (channel model E).

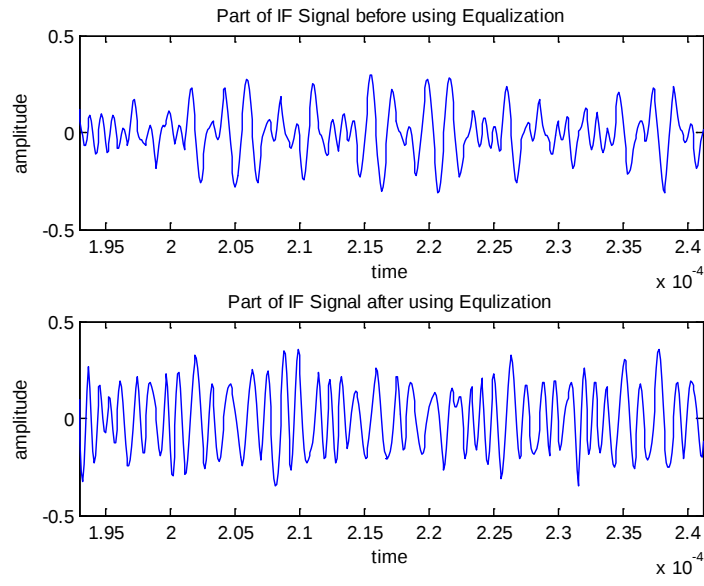


Figure 78: Zoom in version of Figure 77.

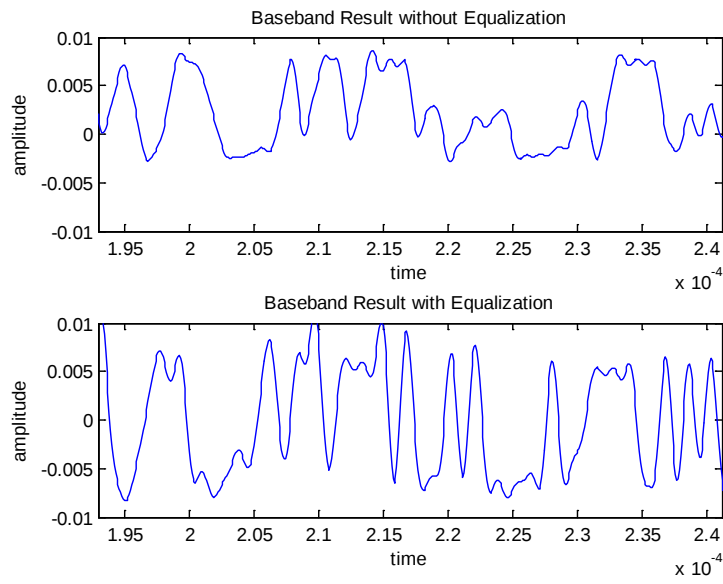


Figure 79: Baseband demodulated signals corresponding to IF signals in Figure 78.

9. Conclusions and Suggestions for Future Work

9.1 Conclusion

An implementation of an LMS based baseband equalizer for DECT system on FPGA has been shown in this report. Implementing the equalizer in an FPGA is a power, area and speed efficient option, where the whole design could be included by using a Cyclone III FPGA as shown in this paper. To the best of our knowledge, there are no publications about DECT equalization design combined with FPGA implementation. Furthermore, due to the re-programmability of FPGA the processor could be tailored for other uses, or this design could be embedded into any DECT systems which have an FPGA on board.

This project has shown that the equalization is an attractive topic in DECT communication. Although the DECT system was not designed for equalization and multi-path effect channels, we showed the possibility of implementing an equalizer in the DECT system in this paper. According to our analysis and real tests, a baseband equalizer combined with a non-coherent demodulator can fix the multi-path effect for a DECT system in low time dispersive channels, but it is not sufficient to deal with more severe fading channels as described in Chapter 6. However, according to the theory that multi-path effect is a linear effect on the transmitted signal, linear equalization can be an efficient way to solve this. So a passband equalizer is introduced, where the equalizer works on the IF signal instead of the demodulated signal. According to the simulation results, this could effectively fix the multi-path effect of the DECT system and lower the bit error rate to meet the requirement. The following table shows the BER for non-coherent demodulation with no equalizer, baseband equalizer and passband equalizer according to a Matlab simulation.

Comparison between different schemes			
schemes	Without Equalizer	Baseband Equalizer	Passband Equalizer
BER	$\geq 2\%$	$< 0.5\%$	$< 0.05\%$

Table 8: BER comparison of different schemes (channel model E).

As shown in Table 8, a baseband equalizer could improve the BER performance, but it is still far away from the requirement of 0.1%. In contrast, a passband equalizer as shown in the last column has a significant improvement on BER, which is lower than the requirement. Therefore, passband equalization is a potential solution for DECT systems in severe multi-path channels.

9.2 Hardware Implementation Analysis

According to the simulation results, a passband equalizer has shown an attractive performance for DECT systems in fading channels. An equalizer will be a necessary component for the DECT system, if the ISI effects cannot be solved by other alternatives. Besides, introducing an equalizer to the DECT system would be a cost efficient way to solve ISI effects.

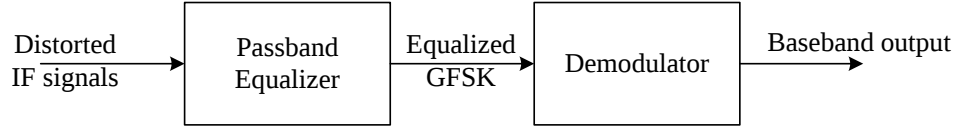


Figure 80: Block diagram of a passband equalizer system.

The passband equalizer system includes passband synchronization, passband equalizer and non-coherent demodulator as shown in the above system diagram. Passband synchronization can be implemented by using cross-correlation to find the syncword. This cross-correlation will be more complicated in computation than baseband synchronization, as multiplications are used in cross-correlation rather than additions and subtractions in baseband synchronization. The number of multipliers (k) required in passband synchronization is given by

$$k = 16 \times R_s / R_b \quad (9.1)$$

where 16 is the length of synch words, R_s is sampling rate for passband signal and R_b is the bit rate.

The length of the passband equalizer is estimated to be about twice that of the baseband equalizer. The number of multipliers in the passband equalizer will be doubled as well. By using the LMS equalizer structure as in [30], the number of multipliers used in the LMS equalizer could be halved. So the computation complexity for a passband equalizer would be kept at the same level as the current design.

According to the simulation results, the performance of the equalizer will not be affected if the sampling frequency decreased from 9 times of symbol rate to 6 times of symbol rate. But further decrease can affect the BER performance. Therefore, by decreasing the sampling frequency, the computation complexity could decrease to 2/3 of the original design. Although the passband synchronization is more complex than the baseband synchronization, the overall computation complexity can be lowered by further optimization. Therefore, a medium sized FPGA as the one used in this project will be sufficient to implement the passband equalizer.

9.3 Issues for Passband Equalizer Implementation

Basically, passband equalization has a similar structure as baseband equalization. So the functional implementation of a passband equalizer is just a small modification of the baseband equalizer. Several issues should be taken into consideration when integrating the equalizer into the DECT system.

As stated before, the modulation index, which is defined at 0.5 in DECT standard, is an important factor for equalizer training sequence generation and passband synchronization. But the modulation index of 0.5 can differ from one base station to another, due to differences in the analog devices. Further, in one base station the modulation index varies from one slot to another. So, it is impossible to generate the training sequence and make passband synchronization without knowing the exact modulation index. A possible solution, given in [37], fixes this issue by searching for the maximum likelihood modulation index by cross-correlation.

More importantly, passband equalizer coefficients updating needs several iterations to converge as mentioned in previous sections. Although this method has an advantage in low computation complexity and is easy to implement, the iterations introduce a large delay, which is estimated about half of one time slot. This will cause problem when a passband equalizer is integrated into the DECT system. A possible optimization method is that the initial values of the equalizer coefficients are specified by a certain algorithm, which can decrease the number of iterations a lot. Another way to improve this is to introduce a higher frequency clock to speed up the computation.

The last issue is the passband synchronization. Since the amplitude of received IF signals varies for each time slot, it is complicated to define a threshold for passband synchronization. One way to improve the accuracy of passband synchronization is to access more information from the system, such as the timing window signal from the system.

To support the hardware implementation and analyze the DECT channel in the real world, a multi-path channel model needs to be built in Matlab according to the real test. This will provide an ideal simulation environment for signal processing without internal and external interference. It is possible to analyze the relationship between SNR and BER based on this model. In addition, the tolerance of DECT to the channel delay spread could be simulated.

The Viterbi algorithm is another widely used equalization algorithm. This algorithm used in DECT systems has been discussed in several papers. They introduced non-coherent demodulation and a non-linear Viterbi equalizer on baseband to solve the ISI effect. According to their analysis result, the Viterbi equalizer might be another effective solution, although its computational complexity is significantly higher than our proposed solution.

10. References

- [1] Theodore S. Rappaport, “Wireless Communications: Principle and Practice”, second edition, Prentice Hall, 2002.
- [2] MATS RYDSTRÖM, “Indoor reflective RF channels and their impact on the DECT system”, 2003.
- [3] DECT, http://en.wikipedia.org/wiki/Digital_Enhanced_Cordless_Telecommunications, Oct. 2009.
- [4] Kazuaki Mupota, Kenkichi Hirade, “GMSK Modulation for Digital Mobile Radio Telephony”, IEEE Transaction on Communications, Vol. COM-29, No. 7, pp. 1044-1050 Jul. 1981.
- [5] Ing. Jan ŠEBESTA, “GMSK Modulation in DSP”, http://www.feec.vutbr.cz/EEICT/2003/fsbornik/03-PGS/01-Electronics/33-sebesta_jan.pdf, 2003
- [6] Roel Schiphorst, Fokke Hoeksema and Kees Slump, “Bluetooth demodulation algorithms and their performance”, 2nd Karlsruhe Workshop on Software Radios, Vol. 2, pp. 99-106, 2002
- [7] Andrea Goldsmith, “Wireless Communications”, Cambridge University Press, 2005.
- [8] DECT, <http://www.wirelesscommunication.nl/reference/chaptr01/telephon/dect.htm>, Oct. 2009.
- [9] ASCOM Technical Report, “DECT BER mätningar för olika kanalmodeller”, Oct. 2009.
- [10] U. Meyer-Baese, “Digital Signal Processing with Field Programmable Gate Arrays”, Second Edition, Springer, 2007.
- [11] DSP Tutorial, <http://www.wave-report.com/tutorials/DSP.htm>, Oct. 2009.
- [12] ASIC basics tutorial, <http://www.radio-electronics.com/info/data/semicond/asic/asic.php>, Oct. 2009.
- [13] R. Woods, J. Mcallister, R. Turner, Y. Yi, G. Lightbody, “FPGA-based Implementation of Signal Processing Systems”, Wiley, 2008.
- [14] Altera Corporation, “Quartus II Handbook Version 9.0”, 2009.
- [15] DSP DesignLine, FPGAs vs. DSPs: A look at the unanswered questions, <http://www.dspdesignline.com/showArticle.jhtml?articleID=196802403>, Oct. 2009.
- [16] I. Kuon, J. Rose, “Measuring the Gap Between FPGAs and ASICs, Computer-Aided Design of Integrated Circuits and Systems”, IEEE Transactions on Vol 26, Issue 2, Feb. 2007.
- [17] John G. Proakis, “Digital Communications”, Fourth Edition, McGraw Hill Higher Education, 2006.
- [18] ISI, http://en.wikipedia.org/wiki/Intersymbol_interference, Oct. 2009.
- [19] Rodger E. Ziemer and Roger L. Peterson, “Introduction to Digital Communications”, second edition, Prentice Hall, 2000.
- [20] Digital filter, http://en.wikipedia.org/wiki/Digital_filter, Oct. 2009.
- [21] C. Chou, J. Evans, FPGA Implementation of Digital Filters, http://www.itc.ku.edu/Projects/FPGA/Digital_Filters.pdf, Oct. 2009.
- [22] Differential signaling, http://en.wikipedia.org/wiki/Differential_signaling, Oct. 2009.
- [23] J. Karki, Fully-Differential Amplifiers, AAP Precision Analog, Jan, 2002.
- [24] Analog Devices, 12-Bit, 20/40/65 MSPS, 3V A/D Converter, AD9235 Datasheet, 2004.
- [25] Arrow Electronics, Inc., <http://www.arrow.com/>, Oct. 2009.
- [26] Arrow Electronics, Inc., LPRP Reference Guide, <http://www.arrow.com/events-training/atsf/portable-applications-workshop/lprp-referenceguide.pdf>, 2007.

- [27] Altera Corporation, "Cyclone III Device Handbook", Jul. 2009.
- [28] A. Lin, K. Gugel, J. Principe, "Feasibility of fixed-point transversal adaptive filters in FPGA devices with embedded DSP blocks, System-on-Chip for Real-Time Applications", The 3rd IEEE International Workshop, Jul. 2003.
- [29] A. Elhossini, S. Areibi, R. Dony, "An FPGA Implementation of the LMS Adaptive Filter for Audio Processing", Reconfigurable Computing and FPGA's, IEEE International Conference, Sep. 2006.
- [30] G. Yecai, H. Longqing, Z. Yanping, "Design and Implementation of Adaptive Equalizer Based On FPGA, Electronic Measurement and Instruments", ICEMI '07. 8th International Conference, Aug. 2007.
- [31] J. Bergeron, "Writing Testbenches: Functional Verification of HDL Models", Second Edition, Springer, Feb. 2003.
- [32] John G. Proakis and Dimitris G. Manolakis, "Digital Signal Processing Principles, Algorithms, and Applications", Fourth Edition, Prentice Hall; Apr, 2006.
- [33] Upamanyu Madhow, "Fundamentals of Digital Communication", Cambridge University Press 2008.
- [34] Carlos A. Belfiore and John H. Park, "Decision Feedback Equalization", Proceedings of the IEEE, Vol. 67, No. 8, pp. 1143-1156, 1979.
- [35] Suzuki, H. Yamao, Y. Kikuchi, H., "A Single-Chip MSK Coherent Demodulator for Mobile Radio Transmission", Vol. 34, pp. 157-168, Nov 1985.
- [36] Jeff Feigin, "Practical Costas loop design", 2002.
- [37] Francisco J. Casajtis-Quirus and Jose Manuel Paez-Borrillo, "Improving DECT Performance with Band-Pass Equalization", Vol. 2, pp. 554-558, May 1997.
- [38] FPGA, <http://en.wikipedia.org/wiki/Fpga>, Oct. 2009.
- [39] Xilinx, Inc, <http://www.xilinx.com>, Oct. 2009.
- [40] Altera Corporation, <http://www.altera.com>, Oct. 2009.