NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Electrical and Electronic Engineering

GRADUATION PROJECT

Implementing OFDM communication system
EE-400

Student: Basil Zind El Hadid (20133114)

Supervisor: Assoc.Prof.Dr.OZGUR OZERDEM
Mohammad kmail

Nicosia – 2014

Abstract

Orthogonal Frequency Division Multiplexing (OFDM) is a multi-carrier modulation ique which divides the available spectrum into many carriers. OFDM uses the spectrum ently compared to FDMA by spacing the channels much closer together and making all ers orthogonal to one another to prevent interference between the closely spaced carriers. main advantage of OFDM is their robustness to channel fading in wireless environment. objective of this project is to implement a base band OFDM transmitter and receiver on TLAB. This project concentrates on developing Fast Fourier Transform (FFT) and Inverse Fourier Transform (IFFT). The work also includes in implement data source, QAM lulator, QAM demodulator, and a mapping module. The design uses 16-point FFT and IFFT the processing module which indicate that the processing block contain 16 inputs data. All dules are implemented using MATLAB software simulation programming language.

All processing is executed in MATLAB environment to give the inputs data from data irce next output data is displayed to computer and the results is compared using MATLAB tware. Between figures of QAM Mod/DeMod, and figures of transmit and receive signals.

Table of Contents

List of Figures

Introduction

With the rapid growth of digital communication in recent years, the need for high-speed data transmission has been increased. The mobile telecommunications industry faces the problem of providing the technology that be able to support a variety of services ranging from voice communication with a bit rate of a few kbps to wireless multimedia in which bit rate up to 2 Mbps. Many systems have been proposed and OFDM system has gained much attention for different reasons. Although OFDM was first developed in the 1960s, only in recent years, it has been recognized as an outstanding method for high-speed cellular data communication where its implementation relies on very high-speed digital signal processing. This method has only recently become available with reasonable prices versus performance of hardware implementation.

The aim for this project is to design a baseband digital communication system OFDM processing including FFT (Fast Fourier Transform) and IFFT (Inverse Fast Fourier Transform), mapping (modulator and demodulator), using MATLAB simulation language in AWGN channel.

The remainder of the report is organized as follows. Chapter 1 discusses OFDM technique and comparison with other multiple access methods. Chapter 2 focuses on the Fast Fourier Transform and Inverse Fast Fourier Transform based on decimation in time or decimation in frequency. Chapter 3 includes the simulations and results of the project. Chapter 4 discusses some of the observations during the project and some future enhancements are also mentioned here.

Chapter 1
OFDM Technique

1-1 Introduction

With the rapid growth of digital communication in recent years, the need for high-speed data transmission has increased. The mobile telecommunications industry faces the problem of providing the technology that be able to support a variety of services ranging from voice communication with a bit rate of a few kbps to wireless multimedia in which bit rate up to many Mbps.

Many systems have been proposed and OFDM system based has gained much attention for different reasons. Although OFDM was first developed in the 1960s, only recently has it been recognized as an outstanding method for high-speed cellular data communication where its implementation relies on very high-speed digital signal processing, and this has only recently become available with reasonable prices of hardware implementation.

1-2 Multichannel Transmission

OFDM started in the mid 60's, Chang [2] proposed a method to synthesis band limited signals for multi channel transmission. The idea is to transmit signals simultaneously through a linear band limited channel without inter channel interference (ICI) an inter symbol interference (ISI).

After that, Saltzberg [3] performed an analysis based on Chang's work and he conclude that the focus to design a multi channel transmission must concentrate on reducing crosstalk between adjacent channels rather than on perfecting the individual signals.

In 1971, Weinstein and Ebert [4] made an important contribution to OFDM. Discrete Fourier transform (DFT) method was proposed to perform the base band modulation and demodulation. DFT is an efficient signal processing algorithm. It eliminates the banks of sub carrier oscillators. They used guard space between symbols to combat ICI and ISI problem. This system did not obtain perfect orthogonality between sub carriers over a dispersive channel.

It was Peled and Ruiz [5] in 1980 who introduced cyclic prefix (CP) that solves the orthogonality issue. They filled the guard space with a cyclic extension of the OFDM symbol. It is assume the CP is longer than impulse response of the channel.

1-3 Basic Principles of OFDM

Orthogonal Frequency Division Multiplexing (OFDM) is a multi-carrier transmission technique, which divides the available spectrum into many carriers, each one being modulated by a low rate data stream.

OFDM is similar to FDMA in that the multiple user access is achieved by subdividing the available bandwidth into multiple channels that are then allocated to users. However, OFDM uses the spectrum much more efficiently by spacing the channels much closer together. This is achieved by making all the carriers orthogonal to one another, preventing interference between the closely spaced carriers.

1-4 Orthogonality Defined

Orthogonality is defined for both real and complex valued functions. The functions $\emptyset_n$ *and* $\emptyset_m$ are said to be orthogonal with respect to each other over the interval $a < t < b$ if they satisfy the condition:

$$\int_a^b \emptyset_n(t)\,\emptyset_m^*\,(t)dt = 0\ , \qquad Where\ n \neq m \qquad 1-1$$

OFDM splits the available bandwidth into many narrowband channels (typically 100-8000), each with its own sub-carrier. These sub-carriers are made orthogonal to one another, meaning that each one has an integer number of cycles over a symbol period. Thus the spectrum of each sub-carrier has a "null" at the center frequency of each of the other sub-carriers in the system, as demonstrated in **Figure** 1.1 below. This results in no interference between the sub-carriers, allowing then to be spaced as close as theoretically possible. Because of this, there is no great need for users of the channel to be time-multiplexed, and there is no overhead associated with switching between users. This overcomes the problem of overhead carrier spacing required in FDMA.

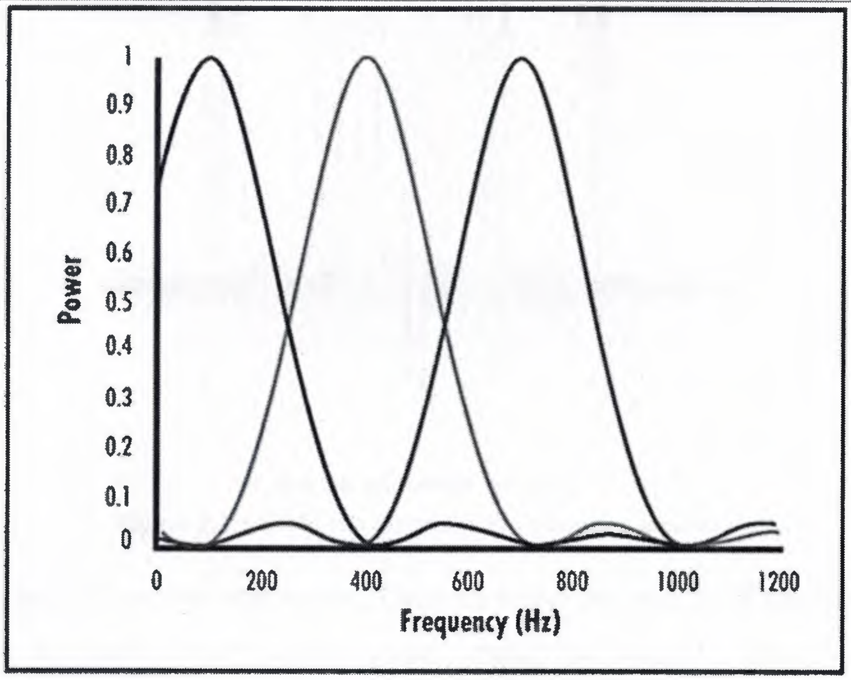
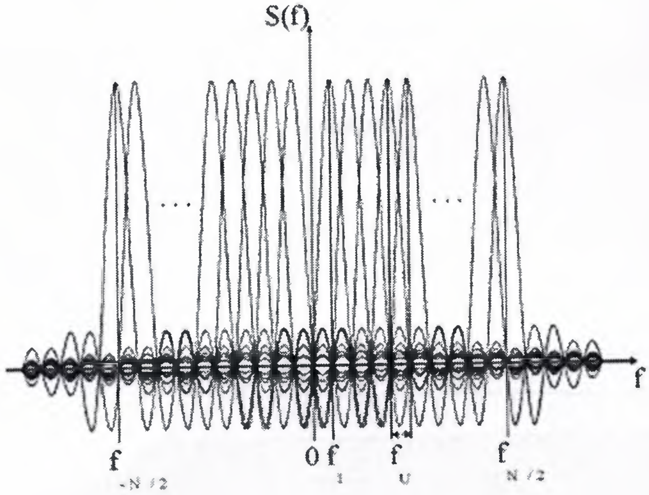


Figure 1.1: Orthogonality of sub-carriers

1-5 OFDM Carriers

As fore mentioned, OFDM is a special form of Multi Carrier Modulation (MCM) and the OFDM time domain waveforms are chosen such that mutual orthogonality is ensured even though sub-carrier spectra may over-lap[3]. With respect to OFDM, it can be stated that orthogonality is an implication of a definite and fixed relationship between all carriers in the collection.

It means that each carrier is positioned such that it occurs at the zero energy frequency point of all other carriers. The sinc function, illustrated in **Figure 1.2** exhibits this property and it is used as a carrier in an OFDM system.



fu is the sub-carrier spacing
Figure 1.2: OFDM sub carriers in the frequency domain

All the subcarriers are sine waves. The area under one period of sine or cosine wave, or any other sinusoidal with some phase angle, is zero[7]. This can be shown diagrammatically.

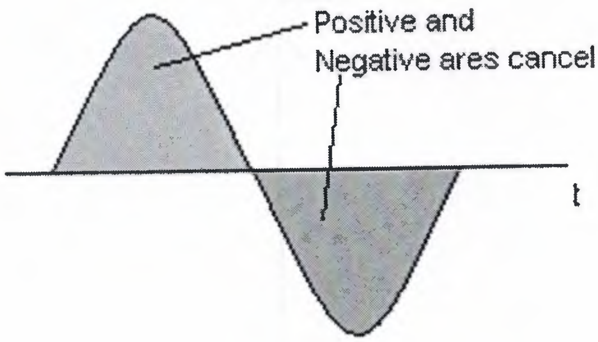
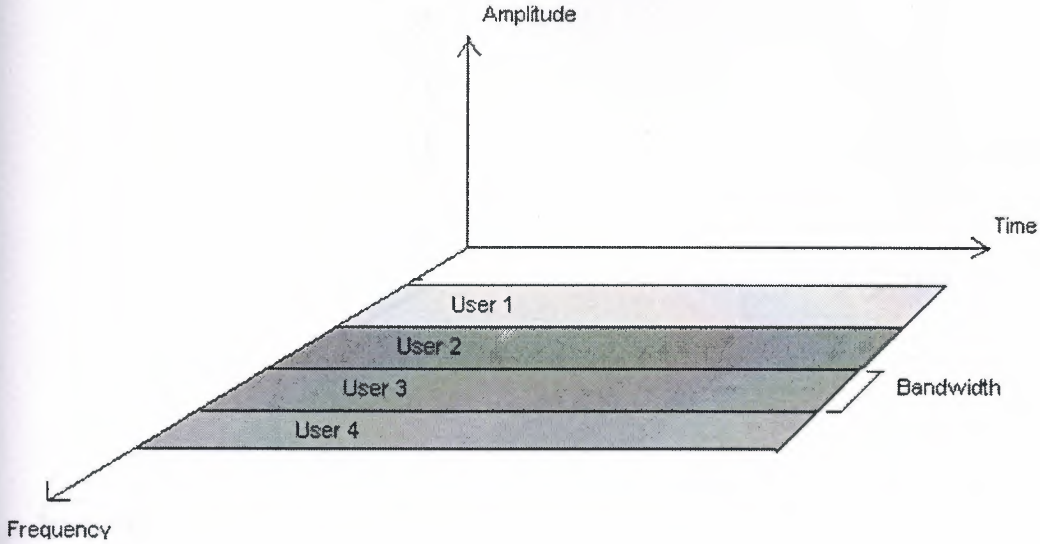


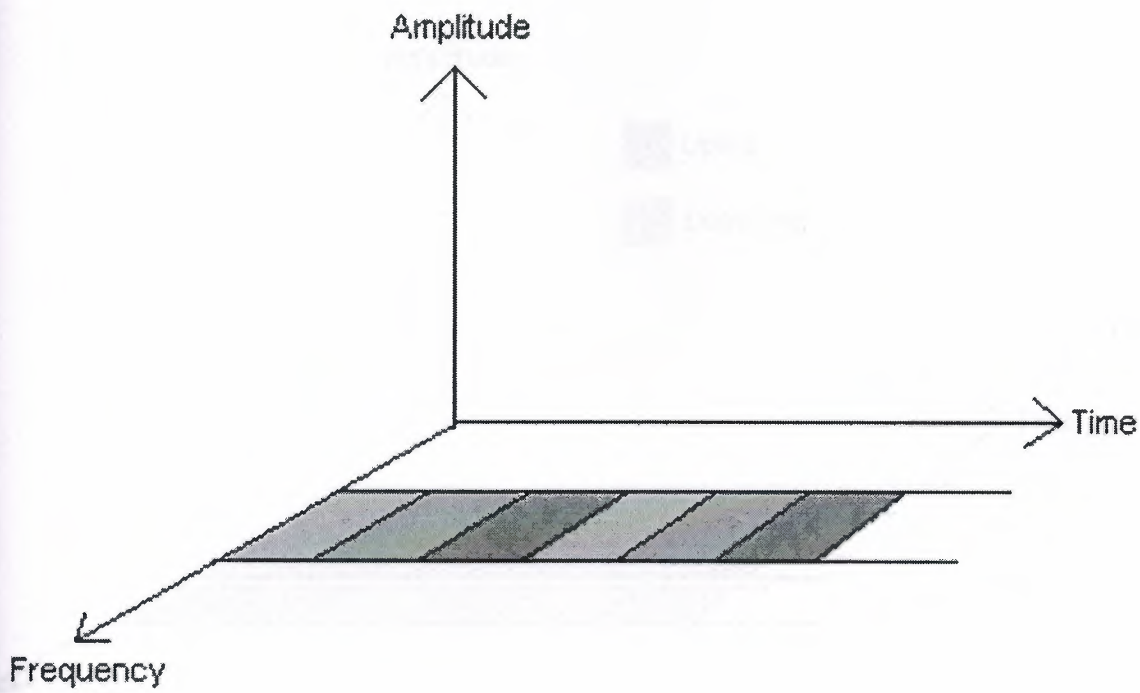
Figure 1.3, a Sine wave

1-6 Comparison of various Multiplexing techniques

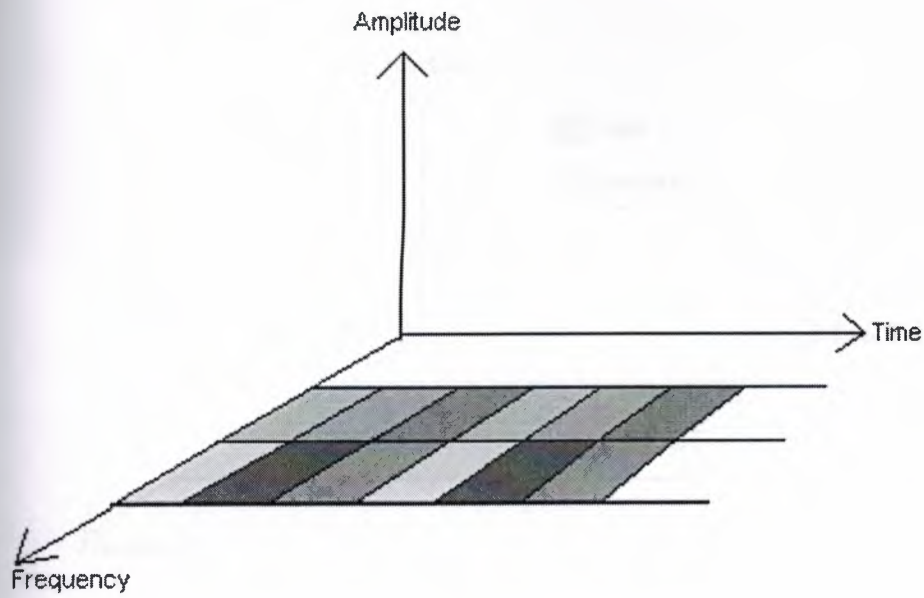
We can compare the different multiplexing techniques with that of OFDM in a diagrammatic way. Shown below are most basic multiplexing techniques of FDM and TDM. There is also a combination of both these scheme, variously applied. The comparison of the various modulation techniques and the OFDM technique is imminent. OFDM uses parallel data streams, and uses many narrowband overlapping digital signals in parallel.



Users differentiated by different color schemes
Figure 1.4, Frequency Division Multiple Access (FDMA)



Users differentiated by different color schemes
Figure 1.5, Time Division Multiple Access (TDMA)



Users differentiated by different color schemes FDMA/Time

Figure 1.6, FDMA and TDMA,

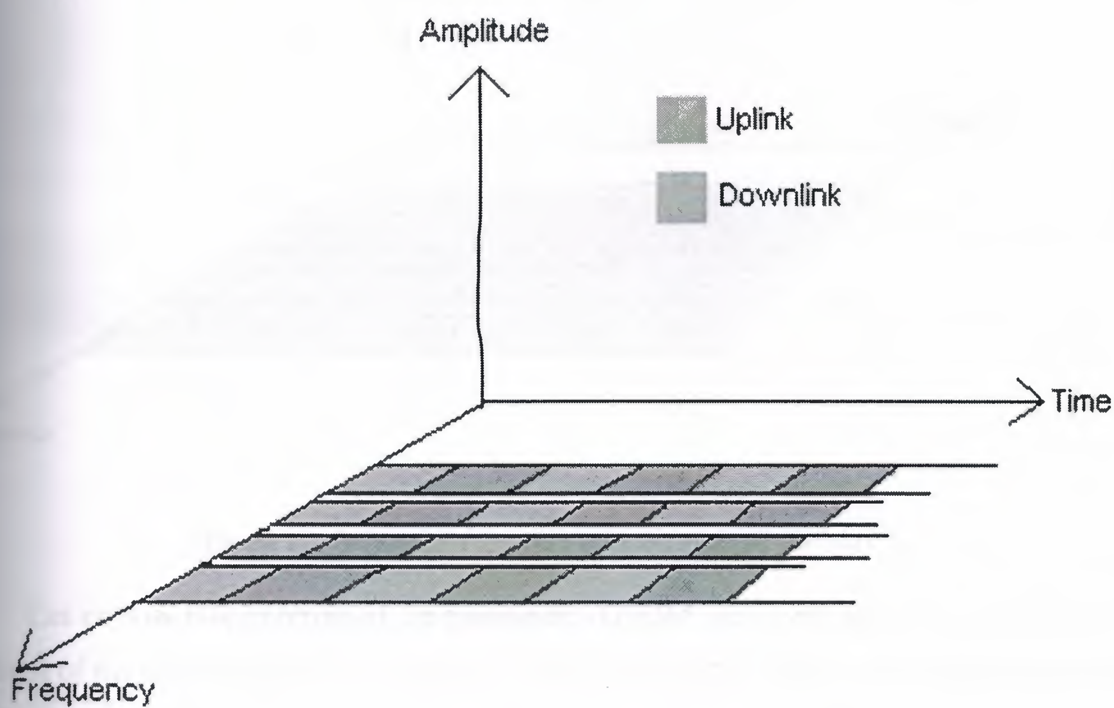


Figure 1.7, FDMA/Time Division Duplexing

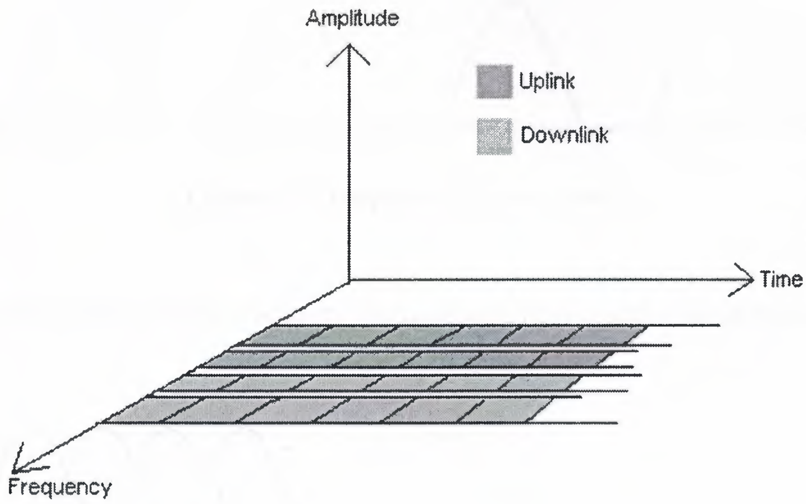


Figure 1.8, Time/ FDMA Division Duplexing

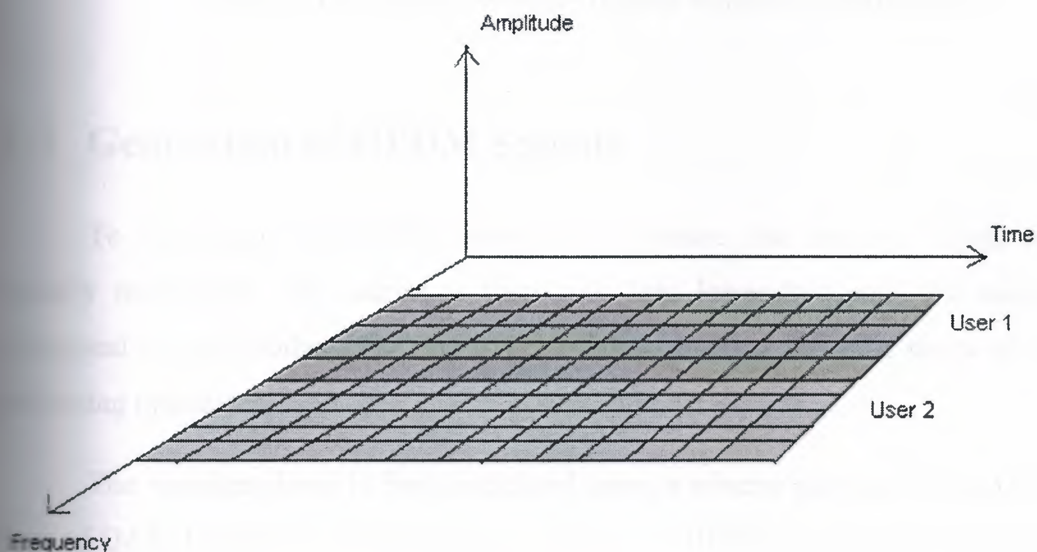


Figure 1.9, Orthogonal Frequency Division Multiple (OFDM)

Let us now talk in terms of the bandwidth. OFDM uses 'orthogonal' carriers and the spectrum of the carriers overlap. But owing to the orthogonality of the carriers, the receiver is able to distinguish between the two adjacent and overlapping carriers. The time to send each symbol is increased (symbol length is increased). So OFDM is not only helpful in saving the bandwidth of the whole system, but also tackles the problems related to fading

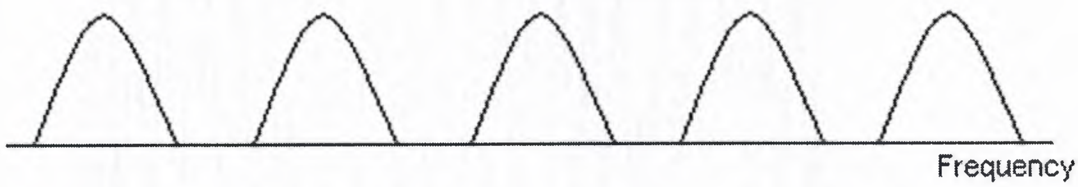


Figure 1.10, Frequency Division spectrum

Conventional Frequency Division Multiplexing multicarrier modulation technique

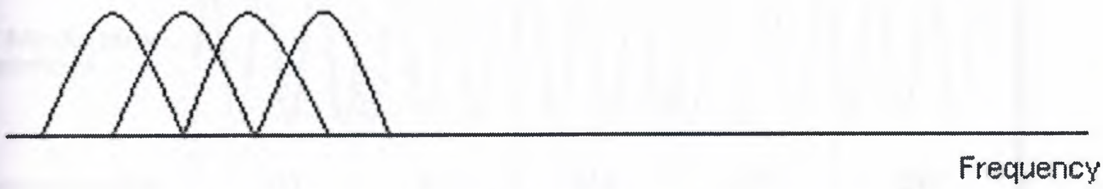


Figure 1.11, Orthogonal Frequency Division Multiple (OFDM) spectrum.

1-7 Generation of OFDM Signals

To implement the OFDM transmission scheme, the message signal must first be digitally modulated. The carrier is then split into lower-frequency sub-carriers that are orthogonal to one another [7]. This is achieved by making use of a series of digital signal processing operations.

The message signal is first modulated using a scheme such as BPSK, QPSK, or some form of QAM (16QAM or 64QAM for example). In BPSK, each data symbol modulates the phase of a higher frequency carrier. **Figure 1.12** shows the time- domain representation of 8 symbols (01011101) modulated within a carrier using BPSK. In the frequency domain, the effect of the phase shifts in the carrier is to expand the bandwidth occupied by the BPSK signal to a sinc function. The zeros (or “nulls”) of the sinc frequency occur at intervals of the symbol frequency.

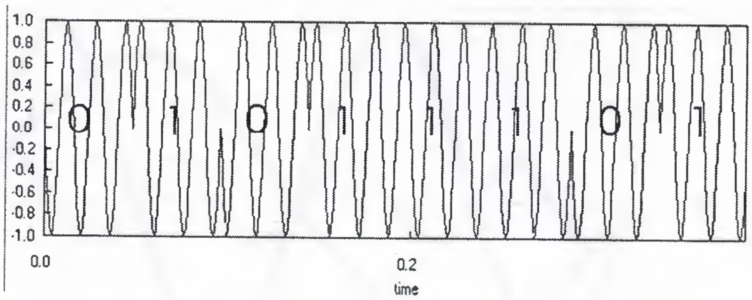


Figure 1.12: Binary Phase-Shift Key (BPSK) representation of "01011101"

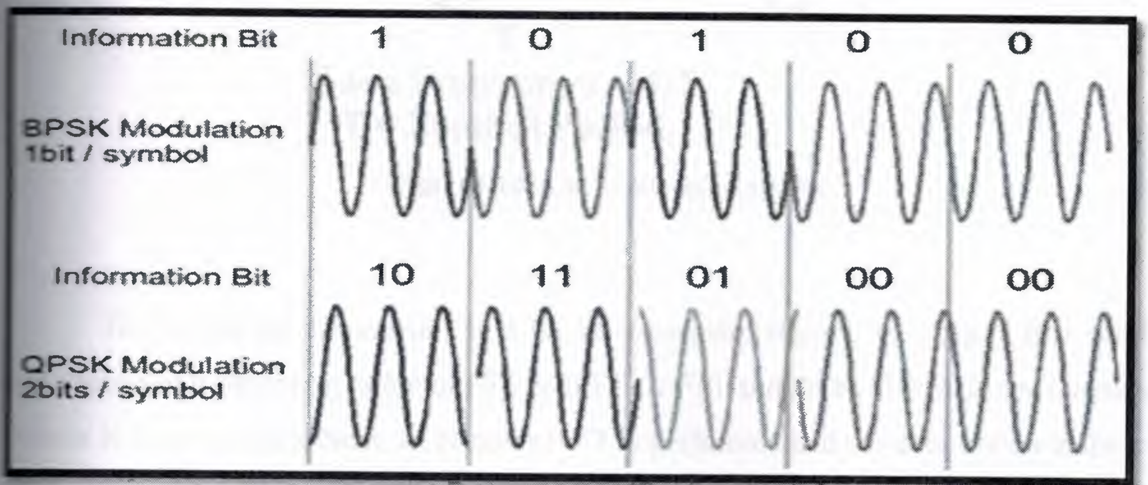


Figure 1.13: Shift Key different method

Originally, multi-carrier systems were implemented through the use of separate local oscillators to generate each individual sub-carrier [5]. This was both inefficient and costly. With the advent of cheap powerful processors, the sub-carriers can now be generated using Fast Fourier Transforms (FFT). The FFT is used to calculate the spectral content of the signal. It moves a signal from the time domain where it is expressed as a series of time events to the frequency domain where it is expressed as the amplitude and phase of a particular frequency. The inverse FFT (IFFT) performs the reciprocal operation.

The underlying principle here is that the FFT can keep tones orthogonal to one another if the tones have an integer number of cycles in a symbol period. In the example **figure 2.3** below, we see signals with 1, 2, and 4 cycles respectively that form an orthogonal set.

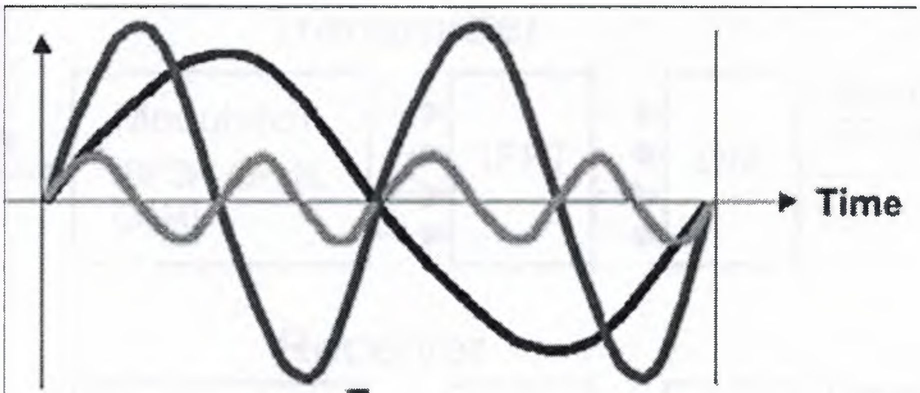


Figure 1.14: A set of orthogonal signals

To convert the sub-carriers to a set of orthogonal signals, the data is first combined into frames of a suitable size for an FFT or IFFT. A FFT should be always in the length of 2N (where N is an integer). Next, an N-point IFFT is performed and the data stream is the output of the transmitter. Thus when the signals of the IFFT output are transmitted sequentially, each of the N channel bits appears at a different sub-carrier frequency.

By using an IFFT process, the spacing of the sub carriers is chosen in such a way that at the frequency where the received signal is evaluated, all other signals is zero. In order for this orthogonality, the receiver and the transmitter must be perfectly synchronized [1]. This means they both must assume exactly the same modulation frequency and the same time-scale for transmission. At the receiver, the exact inverse operations are performed to recover the data. Since the FFT is performed in this stage, the data is back in the frequency domain. It is then demodulated according to the block diagram below.

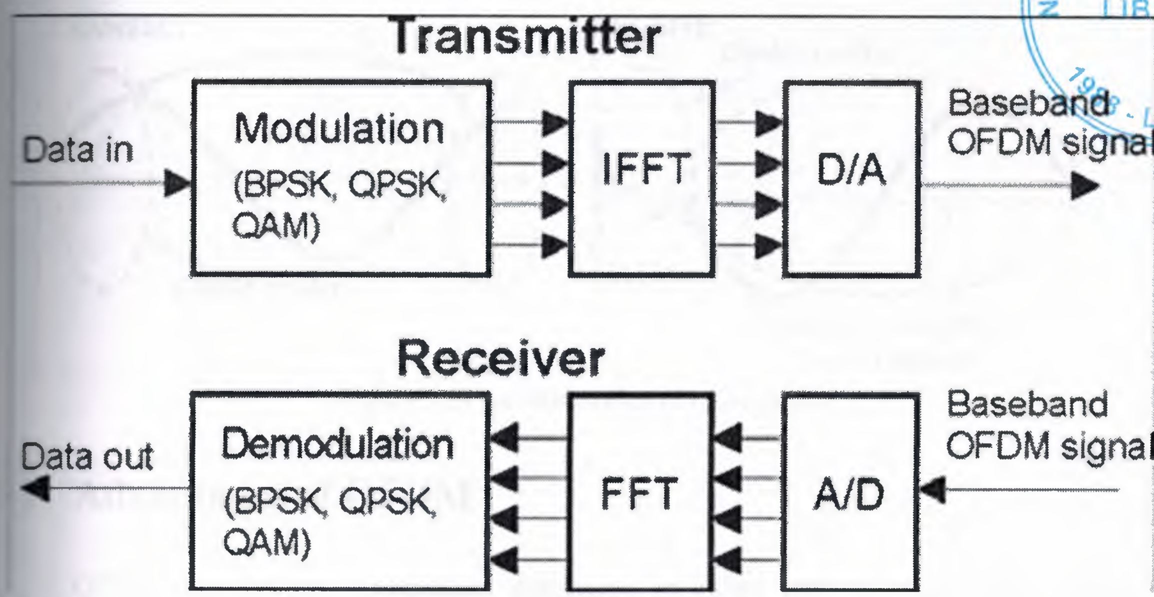


Figure 1.15: Block diagram for OFDM communications

1-8 Guard Period

One of the most important properties of OFDM transmission is its robustness against multi path delay. This is especially important if the signal's sub-carriers are to retain their orthogonality through the transmission process. The addition of a guard period between transmitted symbols can be used to accomplish this. The guard period allows time for multipath signals from the previous symbol to dissipate before information from the current symbol is recorded.

The most effective guard period is a "cyclic prefix", which is appended at the front of every OFDM symbol. The cyclic prefix is a copy of the last part of the OFDM symbol, and is of equal or greater length than the maximum delay spread of the channel (see **Figure 2.16**). Although the insertion of the cyclic prefix imposes a penalty on bandwidth efficiency, it is often the best compromise between performance and efficiency in the presence of inter-symbol interference.

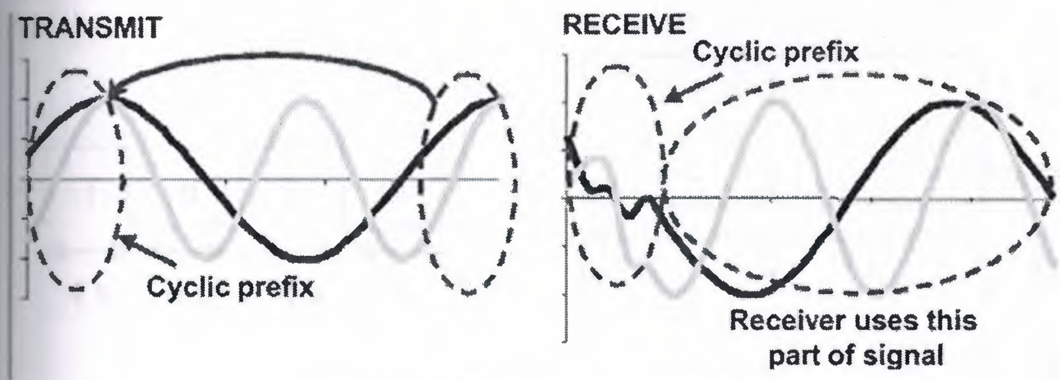


Figure 1.16: Implementation of cyclic prefix

1-9 Advantages of OFDM

OFDM has several advantages compared to other type of modulation technique implemented in wireless system. Below are some of the advantages that describe the uniqueness of OFDM compared to others:

1-9-1 Bandwidth Efficiency

A key aspect of all high-speed communications lies in bandwidth efficiency. This is especially important for wireless communications where all current and future devices are expected to share an already crowded range of carrier frequencies. In OFDM, the frequency band containing the message is divided up into parallel bit streams of lower-frequency carriers, or sub-carriers. These sub-carriers are designed to be orthogonal to one another, such that they can be separated out at the receiver without interference from neighboring carriers. [3] In this manner, OFDM is able to space the channels much closer together, which allows for more efficient use of the spectrum than through simple frequency division multiplexing. The advantage of orthogonality in OFDM does not happen in FDMA where up to 50% of the total bandwidth is wasted due to the extra spacing between channels.

1-9-2 OFDM overcome the effect of ISI

The limitation of sending data in high bit rate is the effect of inter-symbol interference (ISI). As communication systems increase their information transfer speed, the time for each transmission becomes shorter. Since the delay time caused by multi-path remains constant, ISI becomes a limitation in sending high data rate communication. OFDM avoids this problem by sending many low speed transmissions simultaneously. For example **figure 1.17** below shows two ways to transmit the same four pieces of binary data.

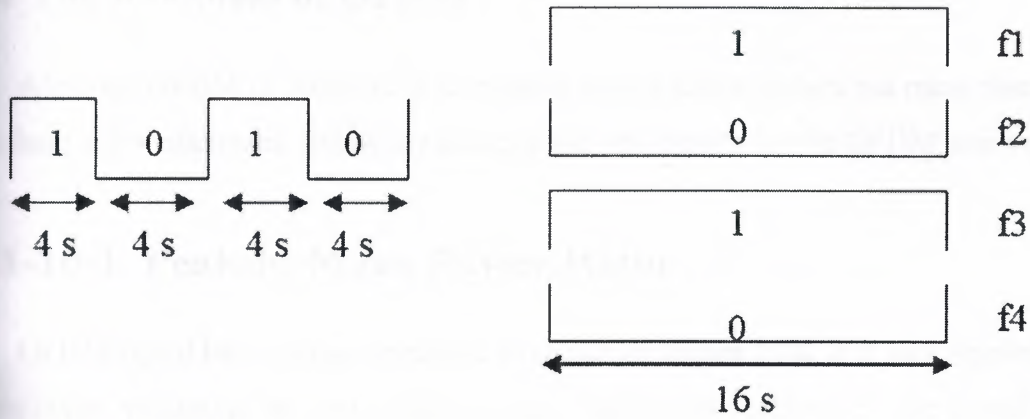


Figure 1.17: Two ways to transmit the same four pieces of binary data

Suppose that this transmission takes four seconds. Then, each piece of data in the left picture has duration of four second. When transmit these data, OFDM would send the four pieces simultaneously as shown on the right. In this case, each piece of data has duration of 16 seconds. This longer duration leads to fewer problems with ISI.

1-9-3 OFDM combats the effect of frequency selective fading and burst error

OFDM is used to spread out a frequency selective fade over many symbols. This effectively randomizes burst errors caused by a deep fade or impulse interference, so that instead of several adjacent symbols being completely destroyed, many symbols are only slightly distorted. This allows successful reconstruction of a majority of them even without forward error correction (FEC) [3]. Because of dividing an entire channel bandwidth into many narrow sub-bands, the frequency response over each individual sub-band is relatively flat. Since each sub-channel covers only a small fraction of original bandwidth, equalization is potentially simpler than in a serial system.

1-10 The weakness of OFDM

Although OFDM is excellent in combating fading effect, it does not mean that OFDM is free from any weaknesses. Below are some of the weaknesses for the OFDM system.

1-10-1 Peak-to-Mean Power Ratio

OFDM signal has varying amplitude as shown by **figure 1.18**. It is very important that the amplitude variations be kept intact as they define the content of the signal. If the amplitude is clipped or modified, then an FFT of the signal would no longer result in the original frequency characteristics, and the modulation may be lost.

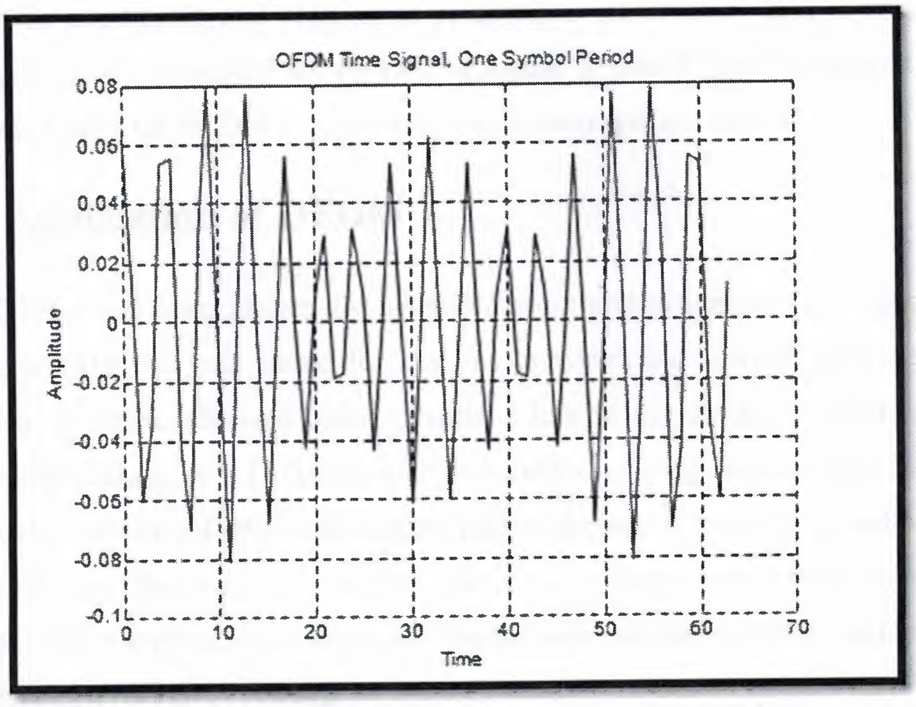


Figure 1.18: Show amplitude varying in OFDM

This is one of the drawbacks of OFDM, the fact that it requires linear amplification. In addition, very large amplitude peaks may occur depending on how the sinusoids line up, so the peak-to-average power ratio is high. This means that the linear amplifier has to have a large dynamic range to avoid distorting the peaks [5]. The result is a linear amplifier with a constant, high bias current resulting in very poor power efficiency.

1-10-2 Synchronization

The other limitation of OFDM in many applications is that it is very sensitive to frequency errors caused by frequency differences between the local oscillators in the transmitter and the receiver. Carrier frequency offset causes a number of impairments including attenuation and rotation of each of the sub carriers and inter- carrier interference (ICI) between sub carriers. In the mobile radio environment, the relative movement between transmitter and receiver causes Doppler frequency shifts, in addition, the carriers can never be perfectly synchronized. These random frequency errors in OFDM system distort orthogonality between sub carriers and thus inter-carrier interference (ICI) occurs.

To optimize the performance of an OFDM link, time and frequency synchronization between the transmitter and receiver is of absolute importance. This is achieved by using known pilot tones embedded in the OFDM signal or attach fine frequency timing tracking algorithms within the OFDM signal's cyclic extension (guard interval).

1-11 Application of OFDM

OFDM has been chosen for several current and future communications systems all over the world. It is well suited for systems in which the channel characteristics make it difficult to maintain adequate communications link performance. In addition to high-speed wireless applications, wired systems such as asynchronous digital subscriber line (ADSL) and cable modem utilize OFDM as its underlying technology to provide a method of delivering high-speed data. Recently, OFDM has also been adopted into several European wireless communications applications such as the digital audio broadcast (DAB) and terrestrial digital video broadcast (DVB-T) systems.

1-11-1 Digital Broadcasting

Standardized in 1995, Digital Audio Broadcasting (DAB) was the first standard to use OFDM. DAB uses a single frequency network, but the efficient handling of multi path delay spread results in improved CD quality sound, new data services, and higher spectrum efficiency. A broadcasting industry group also created Digital Video Broadcasting (DVB) in 1993. DVB produced a set of specifications for the delivery of digital television over cable, DSL and satellite. In 1997 the terrestrial network, Digital Terrestrial Television Broadcasting

DTTB), was standardized [2]. DTTB utilizes OFDM in up to 2,000 and 8,000 sub-carrier modes.

1-11-2 IEEE 802.11a and Wireless LAN

OFDM in the new 5GHz band is comprised of 802.11a. In July 1998, IEEE selected OFDM as the basis for the new 802.11a 5GHz standard in the U.S. targeting a range of data rates up to 54 Mbps.

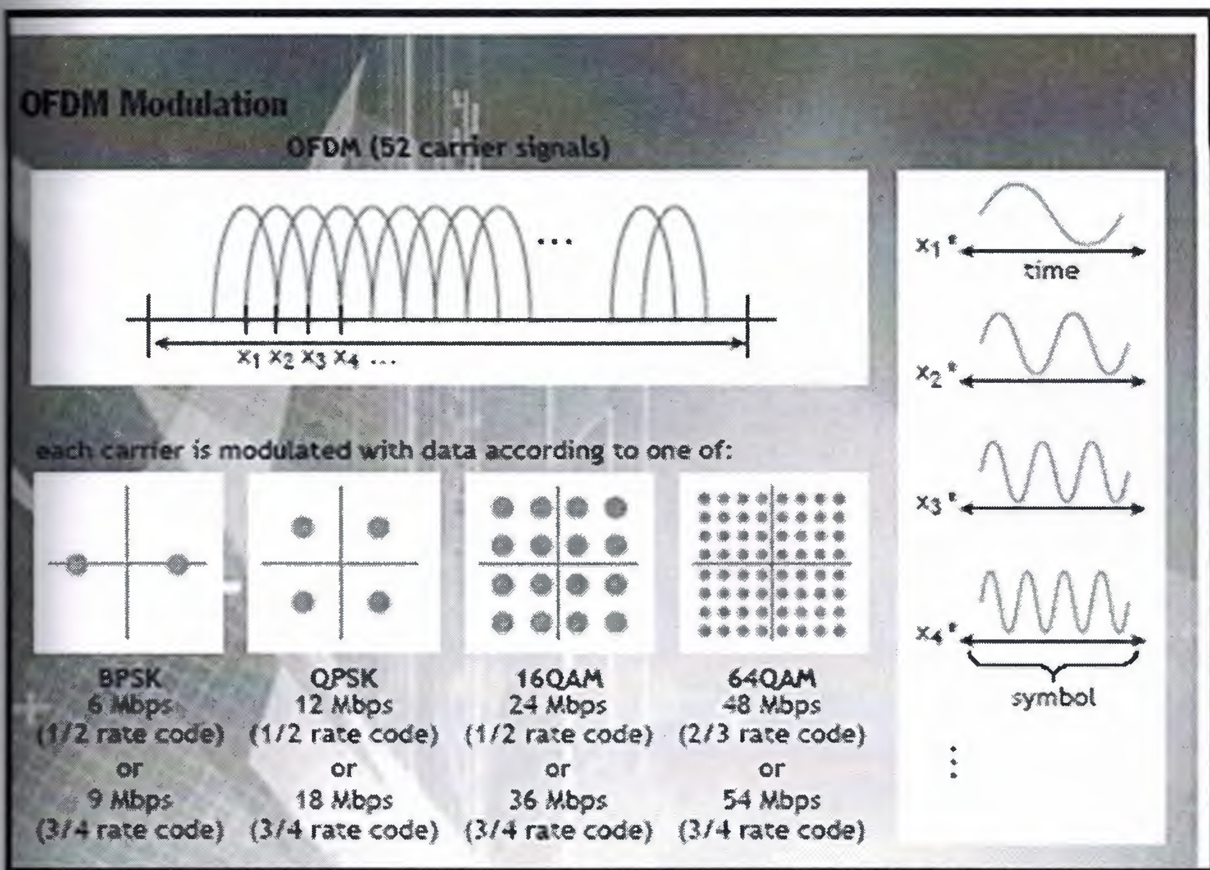


Figure 1.19: OFDM in IEEE 802.11a

1-11-3 Mobile Wireless Communication.

OFDM's capability to work around interfering signals gives it potential to threaten existing CDMA (2.5G and 3G) wireless technology. This is what is allowing the technology to push forward in Europe. In densely populated areas where buildings, vehicles and people can scatter the path of a signal, broadcasters as well as high-speed data providers are anxious to eliminate multi-path effects. According to industry analysts, telecom providers may also be lured to OFDM technology because it could end up causing only a fraction of what it costs to implement 3G wireless technology.

Chapter 2
FFT/IFFT Transform

2-1 Introduction to Fourier Transform

In 1807 the French mathematician and physicist Jean Baptiste Joseph Fourier presented a paper to the Institut de France on the use of sinusoids to represent temperature distributions. The paper made the controversial claim that any continuous periodic signal could be represented by the sum of properly chosen sinusoidal waves. Among the publication review committee were two famous mathematicians: Joseph Louis Lagrange, and Pierre Simon de Laplace. Lagrange objected strongly to publication on the basis that Fourier's approach would not work with signals having discontinuous slopes, such as square waves. Fourier's work was rejected, primarily because of Lagrange's objection, and was not published until the death of Lagrange, some 15 years later [7]. In the meantime, Fourier's time was occupied with political activities, expeditions to Egypt with Napoleon, and trying to avoid the guillotine after the French Revolution!

It turns out that both Fourier and Lagrange were at least partially correct.

Lagrange was correct that a summation of sinusoids cannot exactly form a signal with a corner. However, you can get very close if enough sinusoids are used. (This is described by the Gibbs effect, and is well understood by scientists, engineers, and mathematicians today).

Fourier analysis forms the basis for much of digital signal processing. Simply stated, the Fourier transform (there are actually several members of this family) allows a time domain signal to be converted into its equivalent representation in the frequency domain [9]. Conversely, if the frequency response of a signal is known, the inverse Fourier transform allows the corresponding time domain signal to be determined.

In addition to frequency analysis, these transforms are useful in filter design, since the frequency response of a filter can be obtained by taking the Fourier transform of its impulse response. Conversely, if the frequency response is specified, then the required impulse response can be obtained by taking the inverse Fourier transform of the frequency response [10]. Digital filters can be constructed based on their impulse response, because the coefficients of an FIR filter and its impulse response are identical.

The Fourier transform family (Fourier Transform, Fourier Series, Discrete Time Fourier Series, and Discrete Fourier Transform). These accepted definitions have evolved (not necessarily logically) over the years and depend upon whether the signal is continuous–aperiodic, continuous–periodic, sampled–aperiodic, or sampled–periodic. In this context, the term sampled is the same as discrete.

2-2 The Discrete Fourier Transform

Before going further to discuss on the FFT and IFFT design, it is good to explain a bit on the Fast Fourier Transform and Inverse Fast Fourier Transform operation. The Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) are derived from the main function which is called Discrete Fourier Transform (DFT). The idea of using FFT/IFFT instead of DFT is that the computation of the function can be made faster where this is the main criteria for implementation in the digital signal processing. In DFT the computation for N-point of the DFT will calculate one by one for each point. While for FFT/IFFT, the computation is done simultaneously and this method saves quite a lot of time. Below is the equation showing the DFT and from here the equation is derived to get FFT/IFFT function.

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{\frac{-j2\pi}{N}nk} \qquad 2-1$$

Where x(k) represent the DFT frequency output at the k-the spectral point where k ranges from 0 to N-1. The quantity N represents the number of sample points in the DFT data frame. The quantity x(n) represents the n-th time sample, where n also ranges from 0 to N-1. In general equation, x(n) can be real or complex.

The DFT equation can be re-written into:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \qquad 2-2$$

The quantity W_N^{nk} is:

$$W_N^{nK} = e^{\frac{-j2\pi}{N}nk} \qquad 2-3$$

Then IDFT equation can be written into:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} x(n)\, e^{\frac{j2\pi}{N}nk} = \sum_{k=0}^{N-1} x(n)\, W_N^{-nK}, n = 0,1,\dots\dots\dots, N-1 \qquad 2-4$$

So we find that every value X(k) of DFT signal need N complex multiplying for x(n) with W_N^{nK} and need $(N-1)$ complex adding, that mean the complexity is $[N^2]$.

$$X = [X(0)\ \ X(1) \dots . X(N-1)]^T \qquad 2-5$$

X vector is N time component:

$$x = [x(0)\ \ x(1) \dots . x(N-1)]^T \qquad 2-6$$

And can be written using the Fourier transform of intermittent transformation matrix T_N shape:

$$X = T_N x \qquad 2-7$$

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N^0 & \vdots & W_N^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & & \vdots \\ 1 & W_N^{N-1} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ \vdots \\ x(N-1) \end{bmatrix} \qquad 2-8$$

Because of the great importance of the Fourier transform of intermittent and many applications had to be quick to find ways to display these methods is a Fast Fourier Transform FFT.

2-3 Fast Fourier Transform

Fast Fourier Transform is not the new conversion added to conversions used in signal processing. It is a set of algorithms and methods of calculating fast Fourier transform intermittent DFT, where algorithms are based on the properties related to the conversion twiddle factorsW_N^{nk} that appear in the Fourier transform of intermittent relationship.

Devolve these methods to divide the Fourier transform of intermittent component to the N point Discrete Fourier transformations on a number of denominators number N, and these methods are:

1- Decimation in Time.

2- Decimation in Frequency.

Where these methods are based on the number of points N is the Fourier transform of the power of the number (2) mean ($N = 2^M$)

That's called roads that support this format in ways that basis (2). And there are ways based on the foundation (4) In addition to the algorithms and other methods, all of which depends on the number N compound and to the extent that a composite algorithm be more effective.

2-3-1 Decimation in Time Algorithm

Suppose that the number of points is N conversion is even, in this way we divide tin the time domain to the two vectors of $\left(\frac{N}{2}\right)$ point for each of the elements of even and odd elements.

Odd elements:

$$x = [x(1)\, x(3) \ldots .x(N-1)]^T \qquad 2-9$$

Even elements:

$$x = [x(0)x(2) \ldots .x(N-2)]^T \qquad 2-10$$

As you divide the vector contained in the domain to the frequency of the two vector $\left(\frac{N}{2}\right)$ point for each of the elements of the first half and the second half.

Vector the first half:

$$X = \left[X(0)X(1) \ldots .X\left(\frac{N}{2}-1\right)\right]^T \qquad 2-11$$

Vector the second half:

$$X = \left[X\left(\frac{N}{2}\right)X\left(\frac{N}{2}+1\right) \ldots .X(N-1)\right]^T \qquad 2-12$$

We left the first half of the vector X in terms of the radical elements of even and odd vector x, and can simply write vector frequency contained as a product of the total hit matrix first measure $\left(\frac{N}{2} \times \frac{N}{2}\right)$ vector values odd temporal matrix of other measurement the same vector even values as in the following figure:

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ \vdots \\ X\left(\frac{N}{2}-1\right) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^{2*1} & 1 & \cdots & W_N^{2*\left(\frac{N}{2}-1\right)} \\ 1 & W_N^{2*2} & W_N^{2*2} & \cdots & \\ \vdots & \vdots & W_N^{2*4} & \ddots & \vdots \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \vdots & & \cdots & \cdots & \\ 1 & W_N^{2*\left(\frac{N}{2}-1\right)} & \cdots & \cdots & W_N^{2*\left(\frac{N}{2}-1\right)*\left(\frac{N}{2}-1\right)} \end{bmatrix} \begin{bmatrix} x(0) \\ x(2) \\ \vdots \\ \vdots \\ x(N-2) \end{bmatrix}$$

$$+ \begin{bmatrix} 1 & 1 & & 1 \\ W_N^1 * 1 & W_N^1 * W_N^{2*1} & \cdots & W_N^1 * W_N^{2*\left(\frac{N}{2}-1\right)} \\ W_N^2 * 1 & W_N^2 * W_N^{2*2} & \cdots & \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ W_N^{\left(\frac{N}{2}-1\right)} * 1 & W_N^{\left(\frac{N}{2}-1\right)} * W_N^{2*\left(\frac{N}{2}-1\right)} & \cdots & W_N^{\left(\frac{N}{2}-1\right)} * W_N^{2*\left(\frac{N}{2}-1\right)*\left(\frac{N}{2}-1\right)} \end{bmatrix} \begin{bmatrix} x(1) \\ x(3) \\ \vdots \\ \vdots \\ x(N-1) \end{bmatrix}$$

The matrix show that the impact of the odd element $x(n+1)$with a number $(n+1)$ values of the time element in $X(k)$ values of reciprocating in the first half of any case $\left\{k = 0,1,\ldots,\frac{N}{2}-1\right\}$

Equal to the effect of the even element $x(n)$ the number (n) multiplied by the factor W_N^kin the element $X(k)$:

$$W_N^{(n+1)k} = W_N^k \,.\, W_N^{nk} \qquad 2-13$$

The resulting matrix in writing the previous result from the first rows multiplied with W_N^k where $\left\{k = 0,1,\ldots,\frac{N}{2}\right\}$ becomes the following form:

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ \vdots \\ X\left(\frac{N}{2}-1\right) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^{2*1} & W_N^{2*2} & \cdots & W_N^{2*\left(\frac{N}{2}-1\right)} \\ 1 & W_N^{2*2} & W_N^{2*4} & \ddots & \vdots \\ \vdots & \vdots & \cdots & \cdots & \vdots \\ 1 & W_N^{2*\left(\frac{N}{2}-1\right)} & \cdots & \cdots & W_N^{2*\left(\frac{N}{2}-1\right)*\left(\frac{N}{2}-1\right)} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ \vdots \\ x(N-2) \end{bmatrix}$$

$$+ \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & W_N^{2*1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & W_N^{\left(\frac{N}{2}-1\right)} \end{bmatrix} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ \vdots \\ x(N-1) \end{bmatrix}$$

We note that the matrix is a matrix T_N intermittent Fourier transform of the number of points $\left(\frac{N}{2}\right)$which is equal to $\left(\frac{T_N}{2}\right)$ because:

$$W_N^{2nk} = e^{-\frac{j2\pi}{N}2nk} = e^{-\frac{j2\pi}{\frac{N}{2}}nk} = W_N^{nk} \qquad 2-14$$

In the same way the previous treatment can be written the second half of the vector X in terms of marital and individual elements of the vector x .

The effect of the even element doubles x (n) with n number of values in the temporal element $X\left(\frac{N}{2}+k\right)$

Values of the frequency in the second half $\left\{k = 0,1, \ldots, \frac{N}{2}-1\right\}$ is equal to its impact on the element X (k) because:

$$W_N^{n\left(k+\frac{N}{2}\right)} = W_N^{nk} \,.\, W_N^{n\frac{N}{2}} = W_N^{nk} \qquad 2-15$$

The impact of the odd element $x(n+1)$ with $(n+1)$ number of values in the temporal element $X\left(\frac{N}{2}+k\right)$ values of the frequency in the second half $\left\{k = 0,1, \ldots, \frac{N}{2}-1\right\}$ is equal impact on the element X (k) multiplied by (-1)

$$W_N^{(n+1)\left(k+\frac{N}{2}\right)} = W_N^{(n+1)k} \,.\, W_N^{(n+1)\frac{N}{2}} = -W_N^{(n+1)k} \qquad 2-16$$

In total, we find that we have moved from the Fourier transform of intermittent operation on the N component to the processes of the Fourier transform of intermittent operation, where it is all on the $\left(\frac{N}{2}\right)$ component. In addition to the $\left(\frac{N}{2}\right)$ process transactions W_N^k in the first half in addition to the $\left(\frac{N}{2}\right)$process transactions hit W_N^k in the second half. And $\left(\frac{N}{2}\right)$collection process for each T_N and $\left(\frac{N}{2}\right)$ the process of asking each T_N any mathematical operation and N (**Figure 2-1**) illustrates the time-domain algorithm to differentiate.

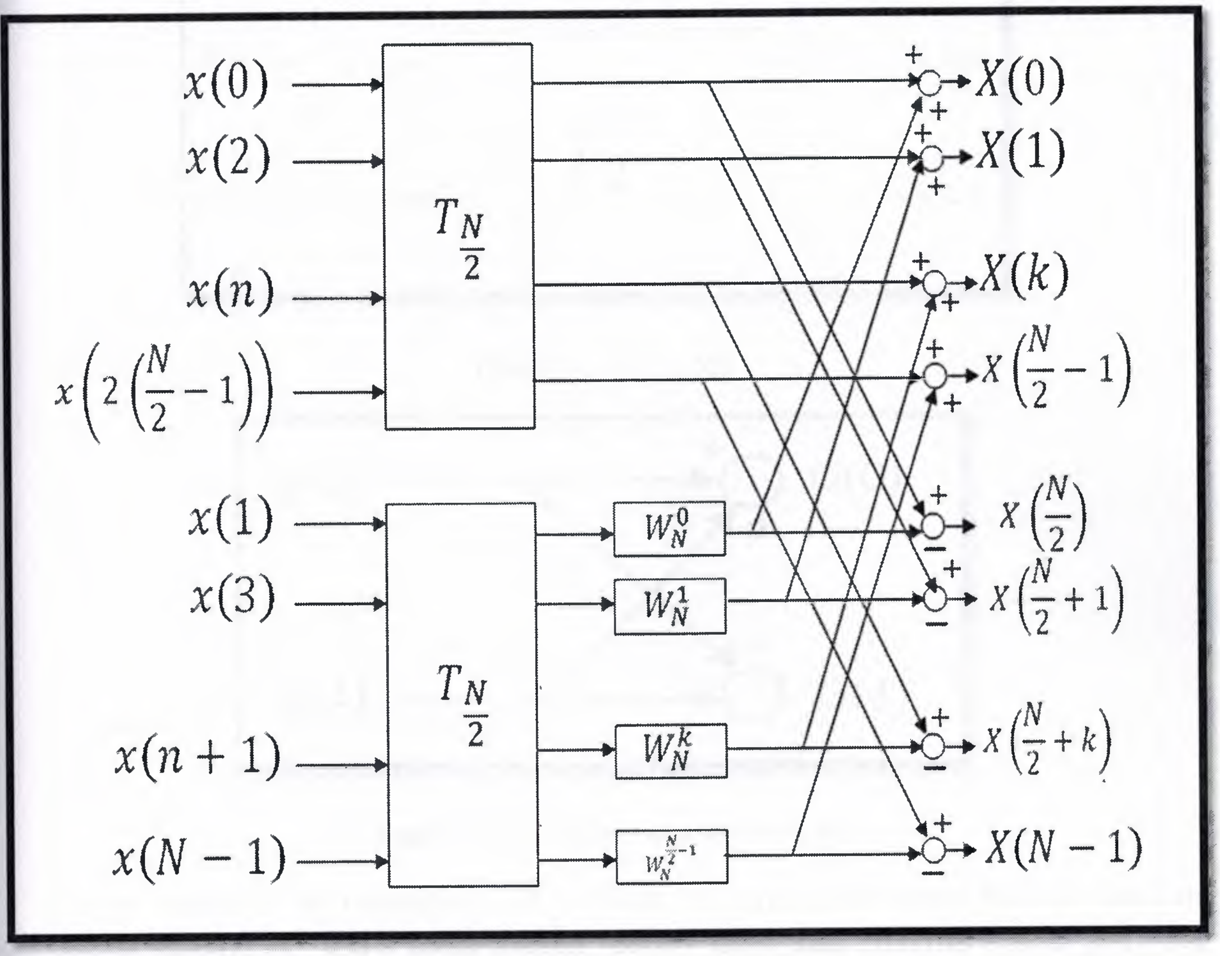


Figure 2.1: DIT algorithm

Though the $\left(\frac{N}{2}\right)$ to divided on 2 , it can be made on what we have done T_N again , any move each process are the Fourier transform $\left(\frac{N}{2}\right)$ component , to the processes of the Fourier transform is the $\left(\frac{N}{4}\right)$ for each element of them (ie, four operations T_N / 4). Will be practical to conduct $\left(\frac{N}{4}\right)$ process hit each two $\left(\frac{T_N}{2}\right)$any $\left(\frac{N}{2}\right)$ process hit a two $\left(\frac{T_N}{2}\right)$ and $\left(\frac{N}{4}\right)$collection process for each of the operations $\left(\frac{T_N}{4}\right)$ four any N addition process.

Thus, if the $2^M = N$ it can return from one stage to another division of radiation elements of time in each time to two of the elements of even and odd elements , and so on until we get to the operations T_N which includes the processes of addition and subtraction only and does not include beatings . Thus we get the M phase of each of them except the last contains $\left(\frac{N}{2}\right)$ and N multiplication process of addition and subtraction. T_N The latter include the N collection process only. Thus, the total processes required to calculate the Fourier transform is intermittent $(M-1) \times \frac{N}{2}$ multiplication and $(M \times N)$ the process of collection, and **figures** (2-2) and (2-3) illustrate the operations in the last stage.

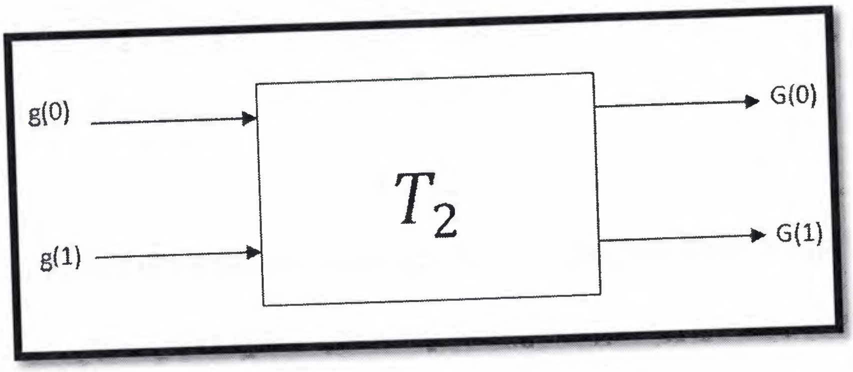


Figure 2.2: DIT process

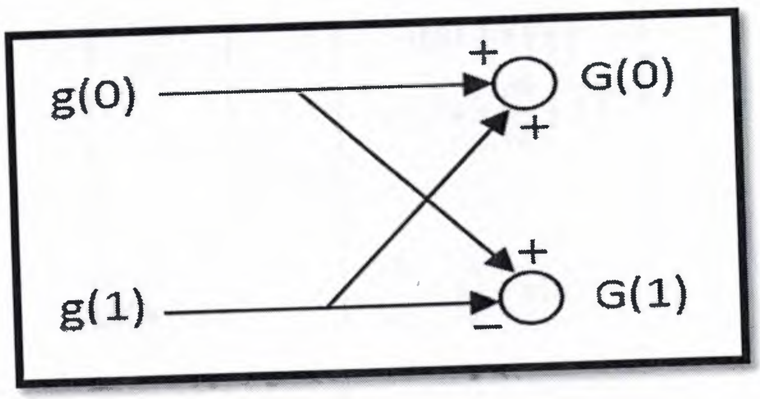


Figure 2.3: The operations in the last stage

For example, the imposition of $N = 1024$, the expense of direct Fourier transform calculation requires N^2 that's mean million process more than DIT that's need just 4600 multiplying process.

2-3-2 Decimation in Frequency Algorithm

This method relies on the fact that the number of points Fourier transform intermittent divisible by (2) in this way we divide the vector:

$$x = [x(0)\ \ x(1) \ldots .x(N-1)]^T \qquad 2-17$$

In the time domain to the two vectors of $\left(\frac{N}{2}\right)$ point for each of the elements of the first half and the second half .

Vector the first half:

$$x = \left[x(0)\ x(1) \ldots . x\left(\frac{N}{2} - 1\right)\right]^T \qquad 2-18$$

Vector the second half:

$$x = \left[x\left(\frac{N}{2}\right)\ x\left(\frac{N}{2}+1\right) \ldots . x(N-1)\right]^T \qquad 2-19$$

As we divide the vector:

$$X = [X(0)\ X(1) \ldots . X(N-1)]^T \qquad 2-20$$

Into two vectors of elements of odd and even elements, odd elements vector:

$$X = [X(1)\ X(3) \ldots . X(N-1)]^T \qquad 2-21$$

Even elements vector:

$$\begin{bmatrix} X(0) \\ X(2) \\ \vdots \\ \vdots \\ \vdots \\ X(N-2) \end{bmatrix} = T_{N/2} \begin{bmatrix} x(0) + x\left(\frac{N}{2}\right) \\ x(1) + x\left(\frac{N}{2}+1\right) \\ \vdots \\ \vdots \\ x\left(\frac{N}{2}-1\right) + x(N-1) \end{bmatrix}$$

$$\begin{bmatrix} X(1) \\ X(3) \\ \vdots \\ \vdots \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & W_N^{2*1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W_N^{\left(\frac{N}{2}-1\right)} \end{bmatrix} T_{N/2} \begin{bmatrix} x(0) - x\left(\frac{N}{2}\right) \\ x(1) - x\left(\frac{N}{2}+1\right) \\ \vdots \\ \vdots \\ x\left(\frac{N}{2}-1\right) - x(N-1) \end{bmatrix}$$

And so we move on account of DISCRETE Fourier transform on N point to tow DISCRETE Fourier transform of the $\left(\frac{N}{2}\right)$ points. But with $\left(\frac{N}{2}\right)$ multiplication process and N adding process. The discussion is also to differentiate the time domain .

And the total number of operations required to calculate the DISCRETE Fourier transform is $(M-1) \times \frac{N}{2}$ multiplication and $(M \times N)$ addition ad process (Figure 2-4) illustrates the algorithm of DIF.

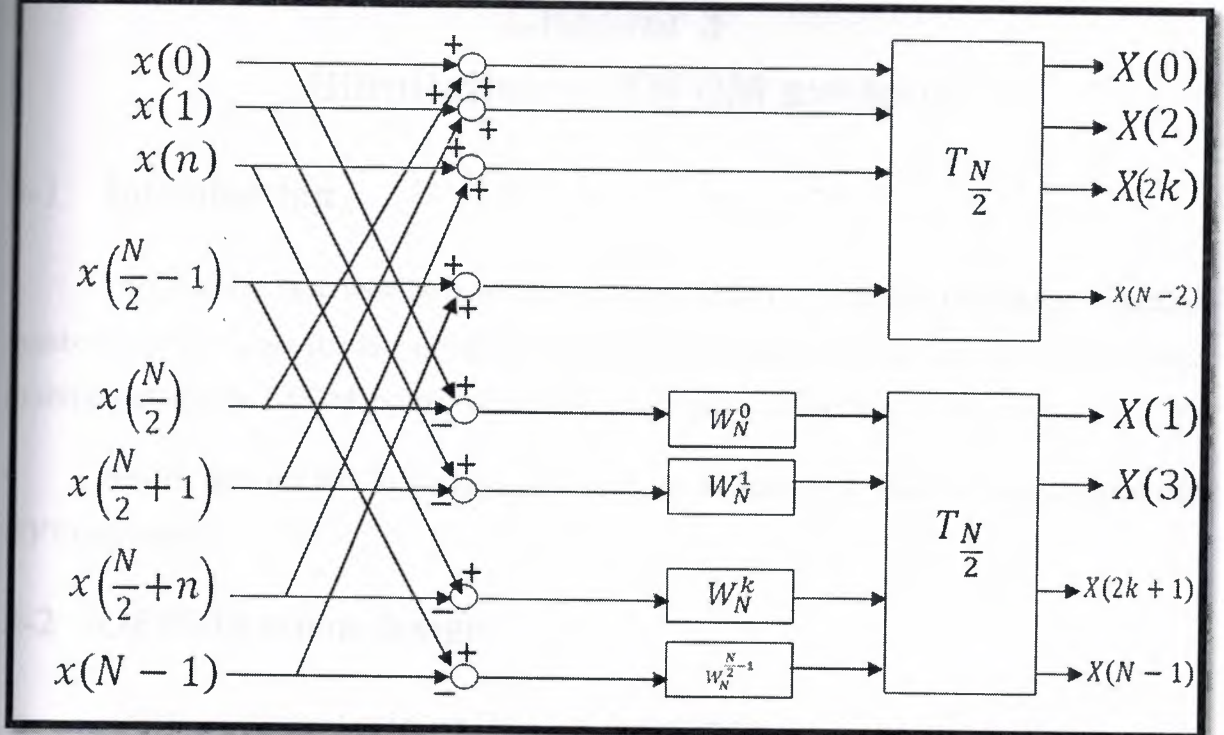


Figure 2.4: DIF algorithm

Chapter 3
Simulation of OFDM system

3-1 Introduction

This section describes how to simulate an OFDM system by generating a random numbers and modulate the bits using QAM modulation, next transmit the modulated bit after multiplexed them in OFDM technique in AWGN channel with signal to noise ratio.

And receive the bits in frequency domain, so we convert it back to time domain with (FFT) algorithm.

3-2 OFDM system design

We design a system as **figure (3-1)** that shown below:

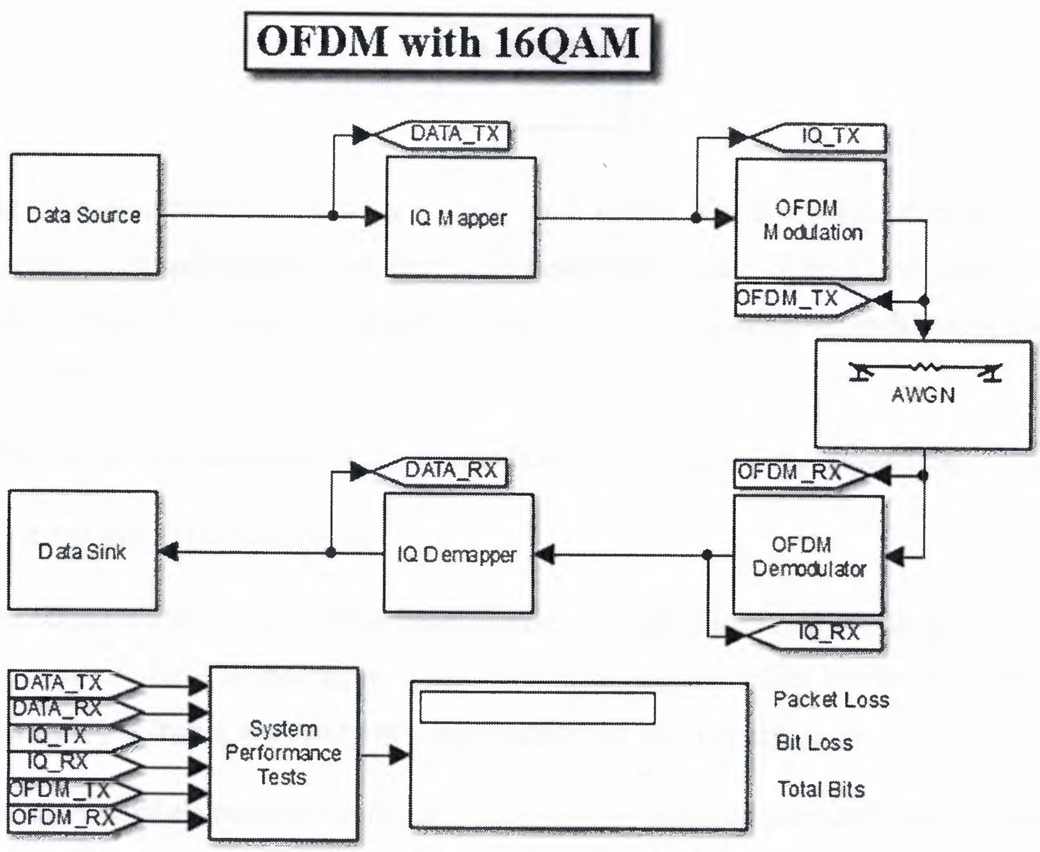


Figure 3.1: OFDM system

3-3 Data source

Using data source as shown below in **figure (3-2)**:

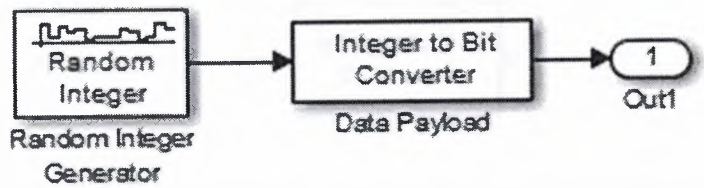
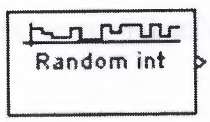


Figure 3.2: Data source

3-3-1 Random Integer Generator

It generates uniformly distributed random integers in the range [0, M-1], where M is the M-ary number defined in the dialog box.



The M-ary number can be either a scalar or a vector. If it is a scalar, then all output random variables are independent and identically distributed (i.i.d.). If the M-ary number is a vector, then its length must equal the length of the Initial seed; in this case each output has its own output range.

If the Initial seed parameter is a constant, then the resulting noise is repeatable.

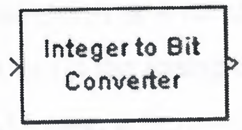
- *Attributes of Output Signal:*

The output signal can be a frame-based matrix, a sample-based row or column vector, or a sample-based one-dimensional array. These attributes are controlled by the Frame-based outputs, Samples per frame, and Interpret vector parameters as 1-D parameters.

The number of elements in the Initial seed parameter becomes the number of columns in a frame-based output or the number of elements in a sample-based vector output. Also, the shape (row or column) of the Initial seed parameter becomes the shape of a sample-based two-dimensional output signal.

3-3-2 Integer to Bit Converter

It maps each integer (or fixed-point value) in the input vector to a group of bits in the output vector.



The block maps each integer value (or stored integer when you use a fixed point input) to a group of M bits, using the selection for the Output bit order to determine the most significant bit. The resulting output vector length is M times the input vector length.

When you set the Number of bits per integer parameter to M and Treat input values as to Unsigned, then the input values must be between 0 and 2M-1. When you set Number of bits per integer to M and Treat input values as to Signed, then the input values must be between – 2M-1 and 2M-1-1. During simulation, the block performs a run-time check and issues an error if any input value is outside of the appropriate range. When the block generates code, it does not perform this run-time check.

This block is single-rate and single-channel. It accepts a length N column vector or a scalar-valued (N = 1) input signal and outputs a length N·M column vector.

The block can accept the data types int8, uint8, int16, uint16, int32, uint32, single, double, and fixed point.

3-4 Quadrature Amplitude Modulation (QAM)

The simplest type of digital modulation involves transmitting a sequence of waveforms ("symbols") $si(t)$ of equal duration T where each waveform is chosen independently from a set of M. This allows us to transmit up to $b = log_2(M)$ bits per symbol.

Common sets of such symbols are those where the real and imaginary parts of the complex baseband signal are each modulated in amplitude. This known as Quadrature Amplitude Modulation (QAM).

Quadrature amplitude modulation (QAM) is both an analog and a digital modulation scheme. It conveys two analog message signals, or two digital bit streams, by changing (modulating) the amplitudes of two carrier waves, using the amplitude-shift keying (ASK) digital modulation scheme or amplitude modulation (AM) analog modulation scheme. The two carrier waves, usually sinusoids, are out of phase with each other by 90° and are thus called quadrature carriers or quadrature components — hence the name of the scheme. The modulated waves are summed, and the resulting waveform is a combination of both phase-shift keying (PSK) and amplitude-shift keying (ASK), or (in the analog case) of phase modulation (PM) and amplitude modulation. In the digital QAM case, a finite number of at least two phases and at least two amplitudes are used. PSK modulators are often designed using the QAM principle, but are not considered as QAM since the amplitude of the modulated carrier signal is constant. QAM is used extensively as a modulation scheme for digital telecommunication systems. Arbitrarily high spectral efficiencies can be achieved with QAM by setting a suitable constellation size, limited only by the noise level and linearity of the communications channel.

3-4-1 Constellation Diagrams

Each of the possible symbols is a combination of a real and a complex value and can be plotted as a point in the complex plane. Each point represents one of the M possible symbols. This plot is called a constellation. For example, the plots below show the constellations used by the 802.11n WLANs as given in the IEEE standard:

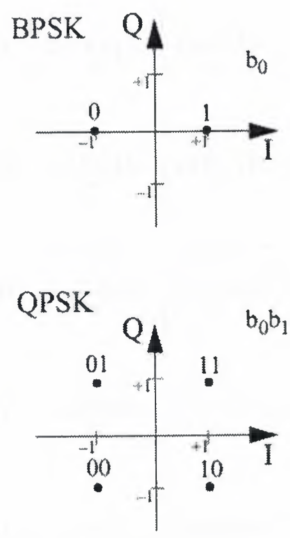


Figure 3.3: BPSK and QPSK

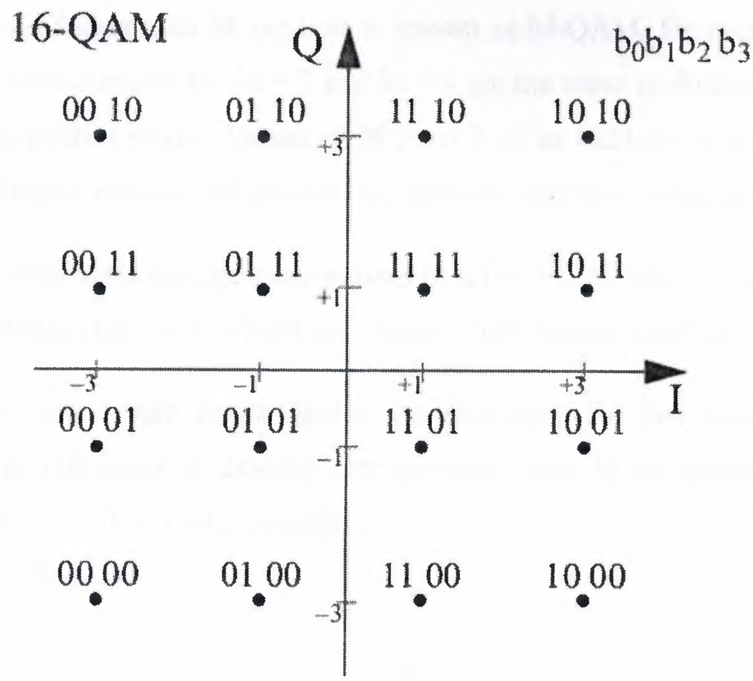


Figure 3.4: 16-QAM

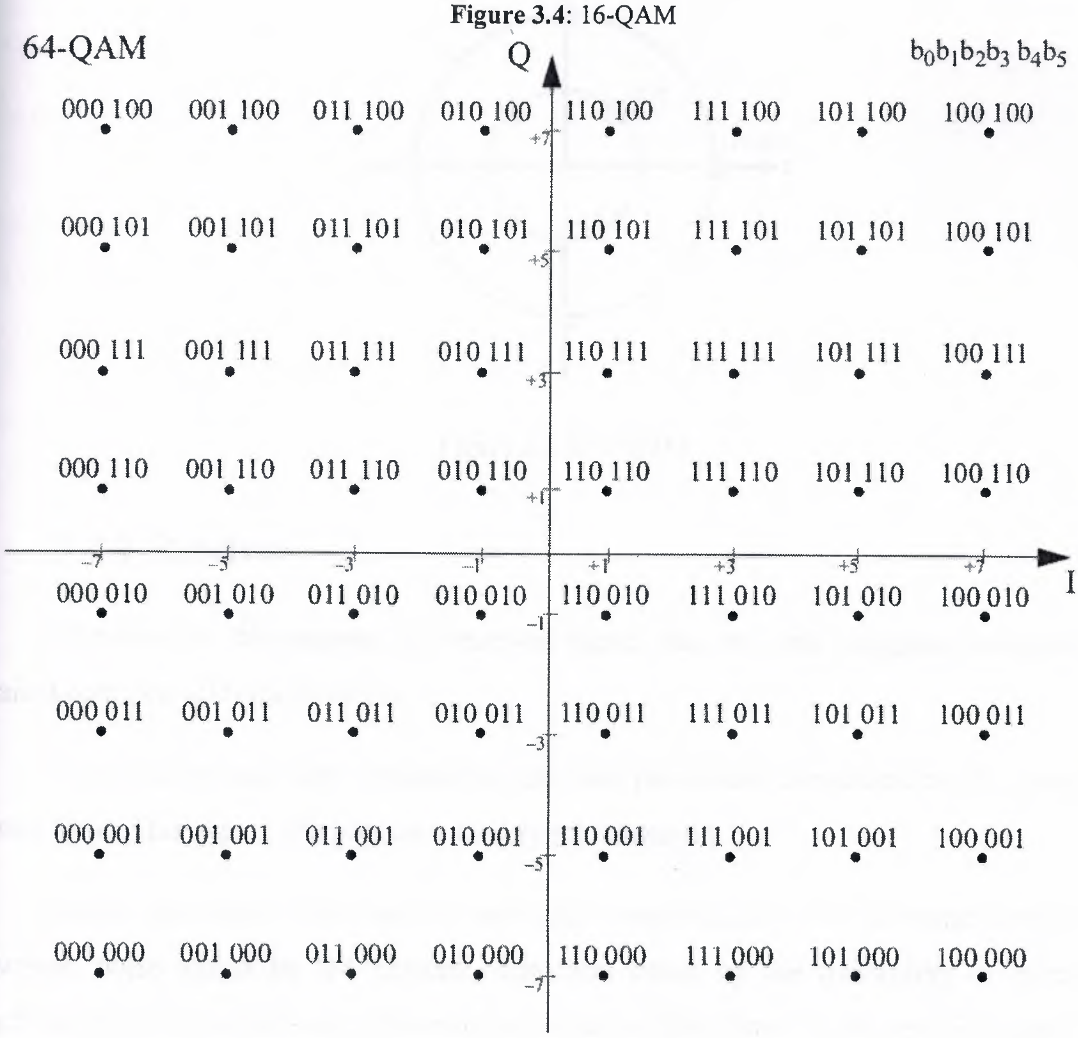


Figure 3.5: 64-QAM

QAM modulation with M symbols is known as M-QAM, for example 16-QAM, 256-QAM, etc. The constellations for M = 2 and M = 4 are the same as those for BPSK and QPSK modulation as explained above. Values of M from 2 up to 1024 (10 bits per symbol) or even more are used. Higher values of M are used on channels with low levels of noise and distortion.

Constellation sizes that are even powers of 2 (M = 2;4;16;64; : : :)are typically used to make the constellation the same in both axes and simplify implementation.

However, non-square constellations are also used for low values of M or where maximum power efficiency is desired. For example, here is an example of a non-square constellation for M = 8 (b = 3 bits/symbol):

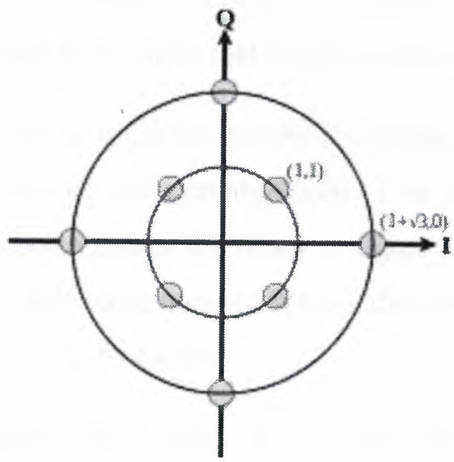


Figure 3.6: M=3 QAM

3-4-2 Receiver

The receiver decomposes the received signal into real and imaginary components to obtain a complex (2D) received value.

The receiver must also estimate the gain and phase shift introduced by the channel and correct for it. This is part of a process called synchronization.

Ideally, the result is that the received value would match one of the transmitted symbols. However, noise added by the channel, distortion cause by the transmitter or receiver, or synchronization errors will cause the received value to differ from the transmitted value.

Typically the noise is uniformly distributed in angle and all symbols are equally probable. In this case an optimum receiver should choose the constellation point nearest the received value to minimize the symbol error rate.

We can visualize the receiver as using "decision thresholds" drawn around each constellation point that define when that symbol is selected for a received value.

Synchronization can be particularly difficult on wireless channels with multipath propagation because the phase and amplitude are constantly changing. Known "pilot" symbols embedded within packet headers are typically used for synchronization.

3-4-3 Errors

Errors happen when additive noise or distortion cause the received value to be closer to a different constellation point than to the point that was transmitted.

The symbol error rate can be approximated by computing the probability that the noise voltage exceeds the distance to the decision threshold. This happens if either the real or imaginary part of the additive noise causes the received signal component to cross a decision threshold. For typical normally distributed noise (AWGN), this can be computed using the error (erf() or erfc()) or Q ("Marcum's Q") functions.

Constellations are usually gray coded. This means that the bits corresponding to adjacent points in the constellation differ by only one bit. Since errors between adjacent points are the most likely, gray coding minimizes the bit error rate.

3-5 IQ (M) Mapper

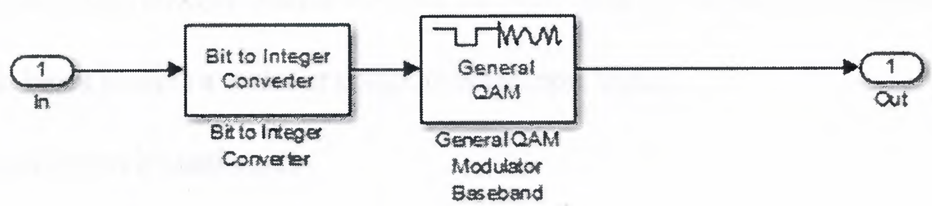
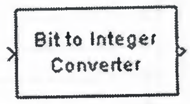


Figure 3.7: IQ (M) Mapper

3-5-1 Bit to Integer Converter

It maps groups of bits in the input vector to integers in the output vector. M defines how many bits are mapped for each output integer.

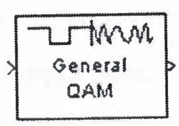


For unsigned integers, if M is the Number of bits per integer, then the block maps each group of M bits to an integer between 0 and 2M-1. As a result, the output vector length is 1/M times the input vector length. For signed integers, if M is the Number of bits per integer , then the block maps each group of M bits to an integer between –2M-1 and 2M-1-1.

This block accepts a column vector input signal with an integer multiple equal to the value you specify for Number of bits per integer parameter. The block accepts double, single, boolean, int8, uint8, int16, uint16, int32, uint32 and ufix1 input data types.

3-5-2 General QAM Modulator Baseband

It modulates using quadrature amplitude modulation. The output is a baseband representation of the modulated signal.



The Signal constellation parameter defines the constellation by listing its points in a length-M vector of complex numbers. The input signal values must be integers between 0 and M-1. The block maps an input integer m to the (m+1)st value in the Signal constellation vector.

This block accepts a scalar or column vector input signal.

Constellation Visualization

The General QAM Modulator Baseband block provides the capability to visualize a signal constellation from the block mask. This Constellation Visualization feature allows you to visualize a signal constellation for specific block parameters.

3-6 OFDM Modulation

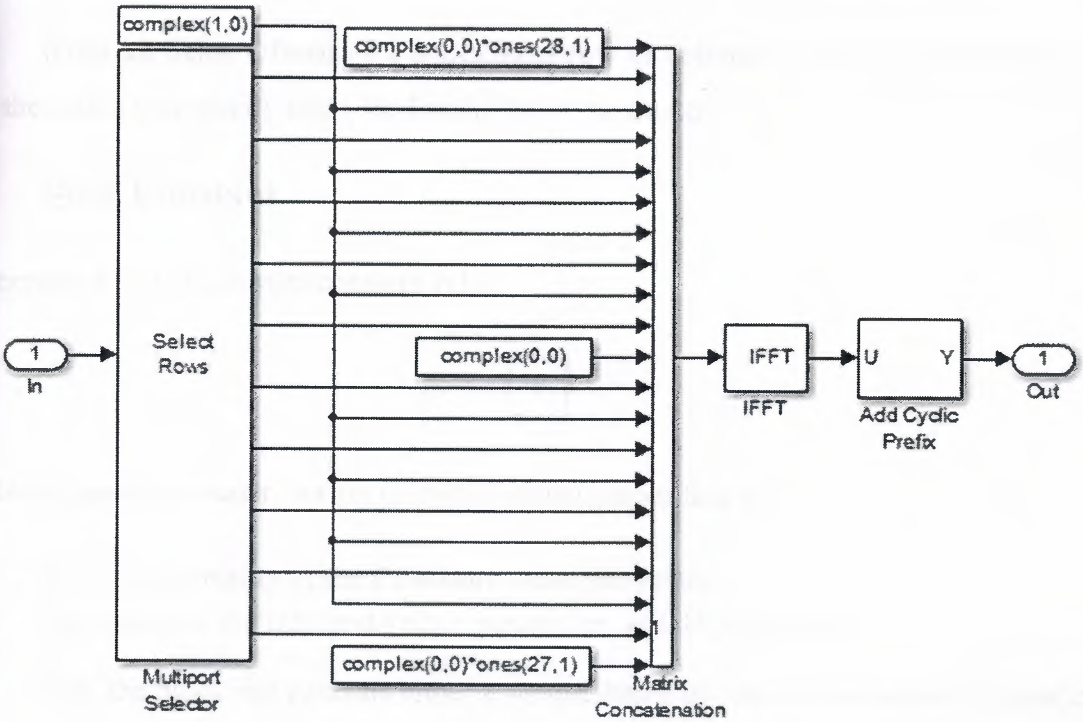
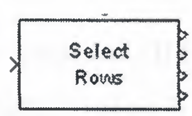


Figure 3.8: OFDM Modulation

3-6-1 Multiport Selector

It extracts multiple subsets of rows or columns from M-by-N input matrix u, and propagates each new sub matrix to a distinct output port. The block treats an un-oriented length-M vector input as an M-by-1 matrix.



The Indices to output parameter is a cell array whose k_{th} cell contains a one-dimensional indexing expression specifying the subset of input rows or columns to be propagated to the k_{th} output port. The total number of cells in the array determines the number of output ports on the block.

When you set the Select parameter to Rows, the block uses the one-dimensional indices you specify to select matrix rows, and all elements on the chosen rows are included. When you set the Select parameter to Columns, the block uses the one-dimensional indices you specify to select matrix columns, and all elements on the chosen columns are included. A given input row or column can appear any number of times in any of the outputs, or not at all.

When an index references a nonexistent row or column of the input, the block reacts with the action you specify using the Invalid index parameter.

3-6-2 Constant

It generates a real or complex constant value.

The block generates scalar, vector, or matrix output, depending on:

- The dimensionality of the **Constant value** parameter
- The setting of the Interpret vector parameters as 1-D parameter

Also, the block can generate either a sample-based or frame-based signal, depending on the setting of the Sampling mode.

The output of the block has the same dimensions and elements as the Constant value parameter. If you specify for this parameter a vector that you want the block to interpret as a vector, select the Interpret vector parameters as 1-D parameter. Otherwise, if you specify a vector for the Constant value parameter, the block treats that vector as a matrix.

3-6-3 IFFT

It computes the inverse fast Fourier transform (IFFT) of each row of a sample-based 1-by-P input vector, or across the first dimension (P) of an N-D input array.

When you specify an FFT length not equal to the length of the input vector, (or first dimension of the input array), the block implements zero padding or modulo-M, (FFT length) data wrapping. This occurs before the IFFT operation, as per Orfanidis.

$$y = ifft(u, M) \qquad \% P \leq M$$

$$y(:, l) = ifft(datawrap(u(:, l), M)) \qquad \% P > M;\ l = 1, \ldots, N$$

When the input length, P, is greater than the FFT length, M, you may see magnitude increases in your IFFT output. These magnitude increases occur because the IFFT block uses modulo-M data wrapping to preserve all available input samples.

To avoid such magnitude increases, you can truncate the length of your input sample, P, to the FFT length, M. To do so, place a Pad block before the IFFT block in your model.

The k_{th} entry of the l_{th} output channel, y(k, l), is equal to the k_{th} point of the M-point inverse discrete Fourier transform (IDFT) of the l_{th} input channel:

$$y(k,l) = \frac{1}{M}\sum_{p=1}^{P} u(p,l)e^{2j\pi(p-1)(k-1)/M} \qquad\qquad k = 1, \dots\dots, M$$

The output of this block has the same dimensions as the input. If the input signal has a floating-point data type, the data type of the output signal uses the same floating-point data type. Otherwise, the output can be any fixed-point data type. The block computes scaled and unscaled versions of the IFFT.

A. FFTW Implementation

The FFTW implementation provides an optimized FFT calculation including support for power-of-two and non-power-of-two transform lengths in both simulation and code generation. Generated code using the FFTW implementation will be restricted to MATLAB® host computers. The data type must be floating-point. Refer to Simulink® Coder™.

B. Radix-2 Implementation

The Radix-2 implementation supports bit-reversed processing, fixed or floating-point data, and allows the block to provide portable C-code generation using the Simulink Coder. The dimension M of the M-by-N input matrix, must be a power of two. To work with other input sizes, use the Pad block to pad or truncate these dimensions to powers of two, or if possible choose the FFTW implementation.

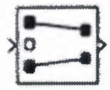
With Radix-2 selected, the block implements one or more of the following algorithms:

- Butterfly operation
- Double-signal algorithm
- Half-length algorithm

- Radix-2 decimation-in-time (DIT) algorithm
- Radix-2 decimation-in-frequency (DIF) algorithm

3-7 Selector

It generates as output selected or reordered elements of an input vector, matrix, or multidimensional signal.



A Selector block accepts vector, matrix, or multidimensional signals as input. The parameter dialog box and the block's appearance change to reflect the number of dimensions of the input.

Based on the value you enter for the Number of input dimensions parameter, a table of indexing settings is displayed. Each row of the table corresponds to one of the input dimensions in Number of input dimensions. For each dimension, you define the elements of the signal to work with. Specify a vector signal as a 1-D signal and a matrix signal as a 2-D signal. When you configure the Selector block for multidimensional signal operations, the block icon changes.

3-8 AWGN Channel

The AWGN Channel block adds white Gaussian noise to a real or complex input signal. When the input signal is real, this block adds real Gaussian noise and produces a real output signal. When the input signal is complex, this block adds complex Gaussian noise and produces a complex output signal. This block inherits its sample time from the input signal.

This block accepts a scalar-valued, vector, or matrix input signal with a data type of type single or double. The output signal inherits port data types from the signals that drive the block.

3-8-1 Signal Processing and Input Dimensions

This block can process multichannel signals. When you set the Input Processing parameter to Columns as channels (frame based), the block accepts an M-by-N input signal. M specifies the number of samples per channel and N specifies the number of channels. Both M and N can be equal to 1. The block adds frames of length-M Gaussian noise to each of the N channels, using a distinct random distribution per channel.

3-9 OFDM Demodulator

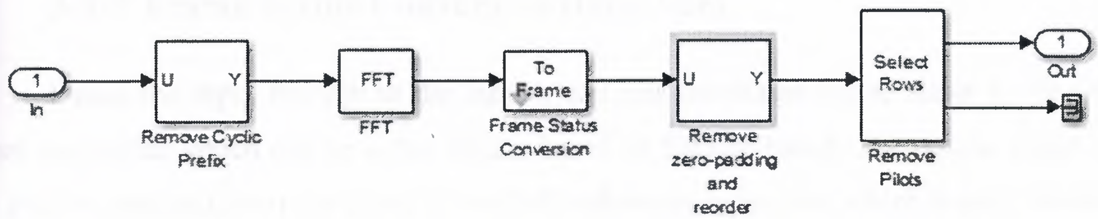
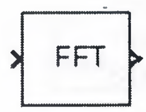


Figure 3.9: OFDM Demodulator

3-9-1 FFT



It computes the fast Fourier transform (FFT) of each row of a sample-based 1-by-P input vector, u, or across the first dimension (P) of an *N*-D input array, u. For user-specified FFT lengths, not equal to P, zero padding or modulo-length data wrapping occurs before the FFT operation, as per Orfanidis.

$$y = fft(u,M) \qquad \% P \leq M$$

$$y(:,l) = fft(datawrap(u(:,l),M)) \qquad \% P > M;\ l = 1,\ldots,N$$

When the input length, P, is greater than the FFT length, M, you may see magnitude increases in your FFT output. These magnitude increases occur because the FFT block uses modulo-M data wrapping to preserve all available input samples.

To avoid such magnitude increases, you can truncate the length of your input sample, P, to the FFT length, M. To do so, place a Pad block before the FFT block in your model.

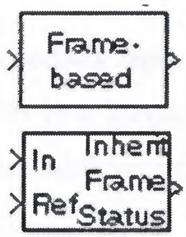
The k_{th} entry of the l_{th} output channel, y(k, l), equals the k_{th} point of the M-point discrete Fourier transform (DFT) of the l_{th} input channel:

$$y(k,l) = \frac{1}{M}\sum_{p=1}^{P} u(p,l)e^{2j\pi(p-1)(k-1)/M} \qquad k = 1, \dots\dots, M$$

The block uses one of two possible FFT implementations. You can select an implementation based on the FFTW library or an implementation based on a collection of Radix-2 algorithms. You can select Auto to allow the block to choose the implementation.

3-9-2 Frame Status Conversion (Obsolete)

Passes the input through to the output, and sets the output frame status to the Output signal parameter, which can be either Frame-based or Sample-based. The output frame status can also be inherited from the signal at the Ref (reference) input port, which is made visible by selecting the Inherit output frame status from Ref input port check box.



When the Output signal parameter setting or the inherited signal's frame status differs from the input frame status, the block changes the input frame status accordingly, but does not otherwise alter the signal. In particular, the block does not rebuffer or resize 2-D inputs. Because 1-D vectors cannot be frame based, when the input is a length-M 1-D vector, and the Output signal parameter is set to Frame-based, the output is a frame-based M-by-1 matrix (that is, a single channel).

When the Output signal parameter or the inherited signal's frame status matches the input frame status, the block passes the input through to the output unaltered.

3-10 IQ (D) Mapper

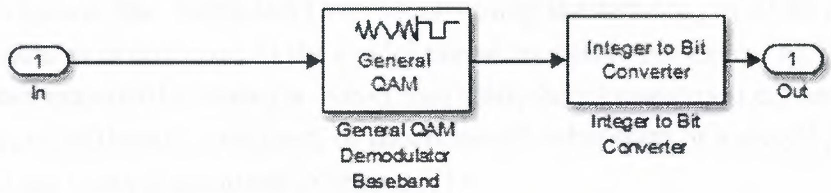
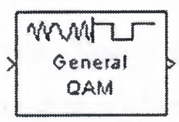


Figure 3.10: IQ (D) Mapper

3-10-1 General QAM Demodulator Baseband

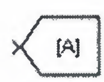
The General QAM Demodulator Baseband block demodulates a signal that was modulated using quadrature amplitude modulation. The input is a baseband representation of the modulated signal.



The input must be a discrete-time complex signal. The Signal constellation parameter defines the constellation by listing its points in a length-M vector of complex numbers. The block maps the *m*th point in the Signal constellation vector to the integer *m*-1.

3-11 Goto

The Goto block passes its input to its corresponding From blocks. The input can be a real- or complex-valued signal or vector of any data type. From and Goto blocks allow you to pass a signal from one block to another without actually connecting them.



A Goto block can pass its input signal to more than one From block, although a From block can receive a signal from only one Goto block. The input to that Goto block is passed to the From blocks associated with it as though the blocks were physically connected. Gotoblocks and From blocks are matched by the use of Goto tags.

The Tag Visibility parameter determines whether the location of From blocks that access the signal is limited:

- local, the default, means that From and Goto blocks using the same tag must be in the same subsystem. A local tag name is enclosed in brackets ([]).
- scoped means that From and Goto blocks using the same tag must be in the same subsystem or at any level in the model hierarchy below the Goto Tag Visibility block that does not entail crossing a nonvirtual subsystem boundary, i.e., the boundary of an atomic, conditionally executed, or function-call subsystem or a model reference. A scoped tag name is enclosed in braces ({}).

global means that From and Goto blocks using the same tag can be anywhere in the model except in locations that span nonvirtual subsystem boundaries.

The rule that From-Goto block connections cannot cross nonvirtual subsystem boundaries has the following exception. A Goto block connected to a state port in one conditionally executed subsystem is visible to a From block inside another conditionally executed subsystem.

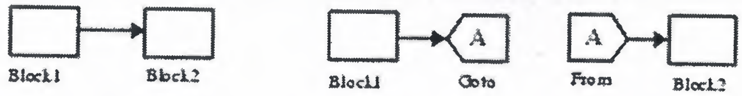
Use local tags when the Goto and From blocks using the same tag name reside in the same subsystem. You must use global or scoped tags when the Goto and From blocks using the same tag name reside in different subsystems. When you define a tag as global, all uses of that tag access the same signal. A tag defined as scoped can be used in more than one place in the model.

3-12 From

The From block accepts a signal from a corresponding Goto block, then passes it as output. The data type of the output is the same as that of the input from the Goto block. From and Goto blocks allow you to pass a signal from one block to another without actually connecting them. To associate a Goto block with a From block, enter the Goto block's tag in the Goto Tag parameter.

A From block can receive its signal from only one Goto block, although a Goto block can pass its signal to more than one From block.

This figure shows that using a Goto block and a From block is equivalent to connecting the blocks to which those blocks are connected. In the model at the left, Block1 passes a signal to Block2. That model is equivalent to the model at the right, which connects Block1 to the Goto block, passes that signal to the From block, then on to Block2.



3-13 System Performance Test

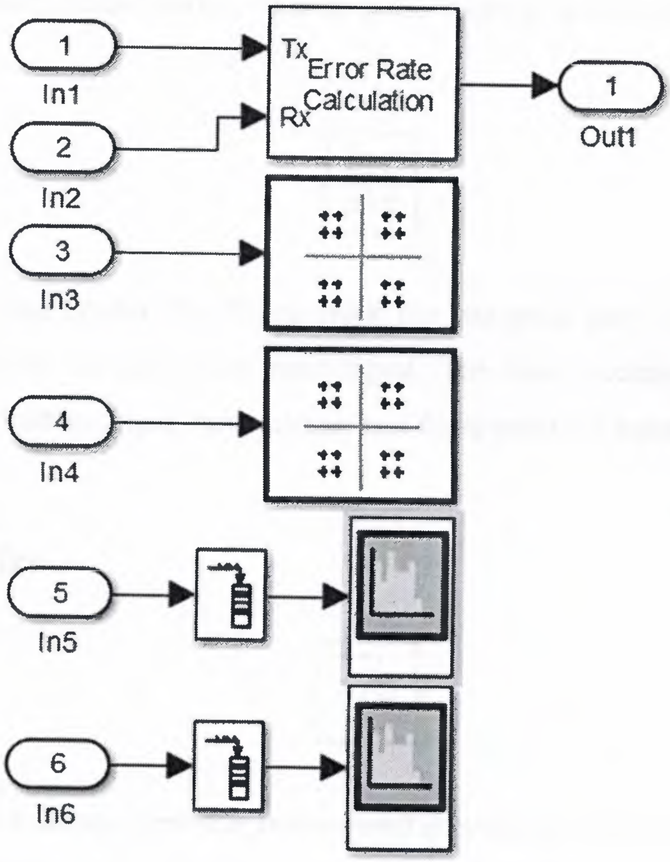
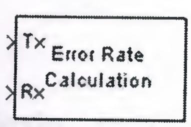


Figure 3.11: System Performance Test

3-13-1 Error Rate Calculation

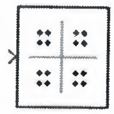
The Error Rate Calculation block compares input data from a transmitter with input data from a receiver. It calculates the error rate as a running statistic, by dividing the total number of unequal pairs of data elements by the total number of input data elements from one source.



Use this block to compute either symbol or bit error rate, because it does not consider the magnitude of the difference between input data elements. If the inputs are bits, then the block computes the bit error rate. If the inputs are symbols, then it computes the symbol error rate.

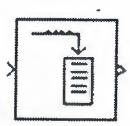
3-13-2 Discrete-Time Scatter Plot Scope

The Discrete-Time Scatter Plot Scope block displays scatter plots of a modulated signal, to reveal the modulation characteristics, such as pulse shaping or channel distortions of the signal.

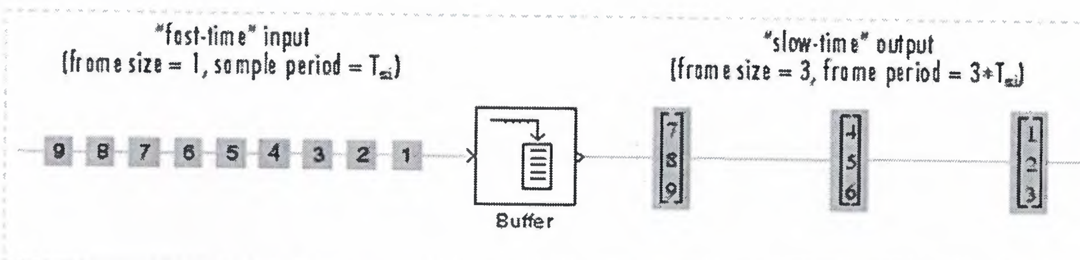


The Discrete-Time Scatter Plot Scope block has one input port. This block accepts a complex scalar-valued or column vector input signal. The block accepts a signal with the following data types: double, single, base integer, and fixed-point for input, but will cast it as double.

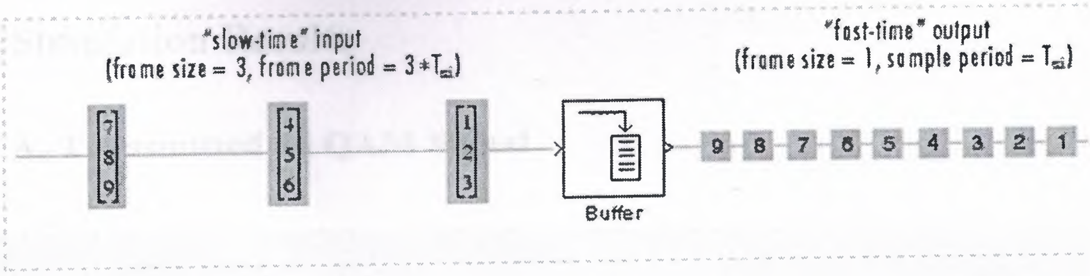
3-13-3 Buffer



The Buffer block always performs frame-based processing. The block redistributes the data in each column of the input to produce an output with a different frame size. Buffering a signal to a larger frame size yields an output with a *slower* frame rate than the input. For example, consider the following illustration for scalar input.



Buffering a signal to a smaller frame size yields an output with a faster frame rate than the input. For example, consider the following illustration of scalar output.



The block coordinates the output *frame size* and *frame rate* of nonoverlapping buffers such that the sample period of the signal is the same at both the input and output: Tso = Tsi.

This block supports triggered subsystems when the block input and output rates are the same.

3-13-4 Spectrum Analyzer

The Spectrum Analyzer block, hereafter referred to as the scope, displays frequency spectra of signals. The Spectrum Analyzer block accepts input signals with the following characteristics:

- Discrete sample time
- Real- or complex-valued
- Fixed number of channels of variable length
- Floating- or fixed-point data type

3-14 Simulation Results

A. Transmitted 16 QAM Signal

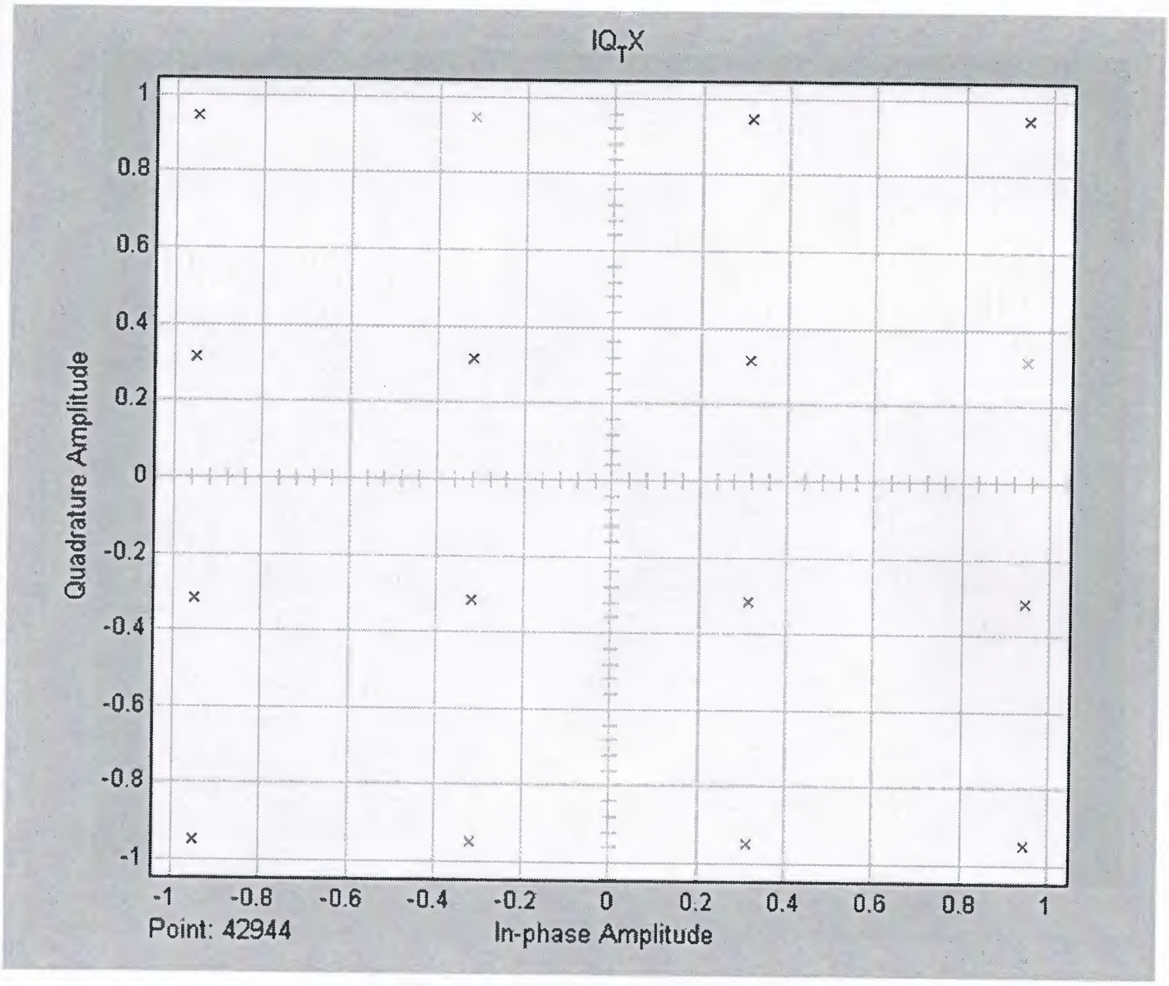


Figure 3.12: Transmitted 16 QAM Signal

B. Transmitted OFDM Signal

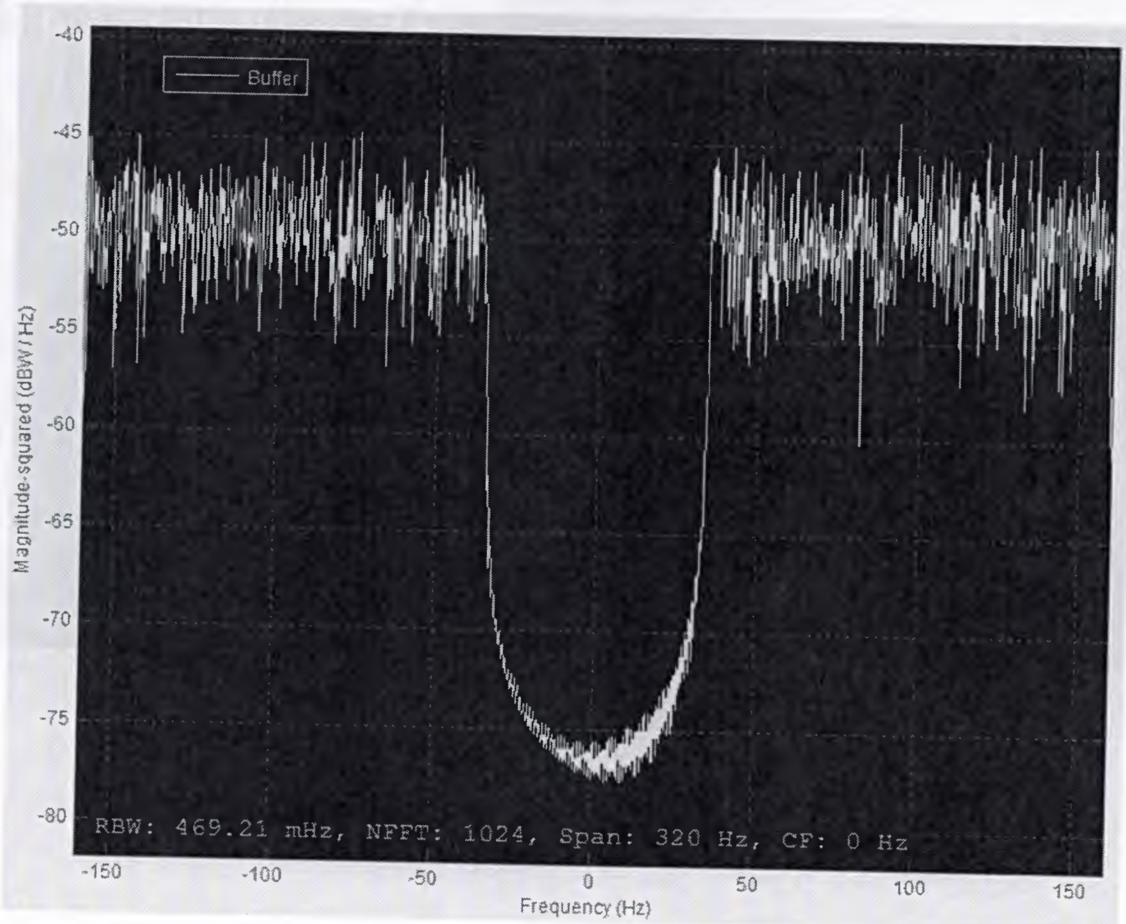


Figure 3.13: Transmitted OFDM Signal

C. Received OFDM Signal

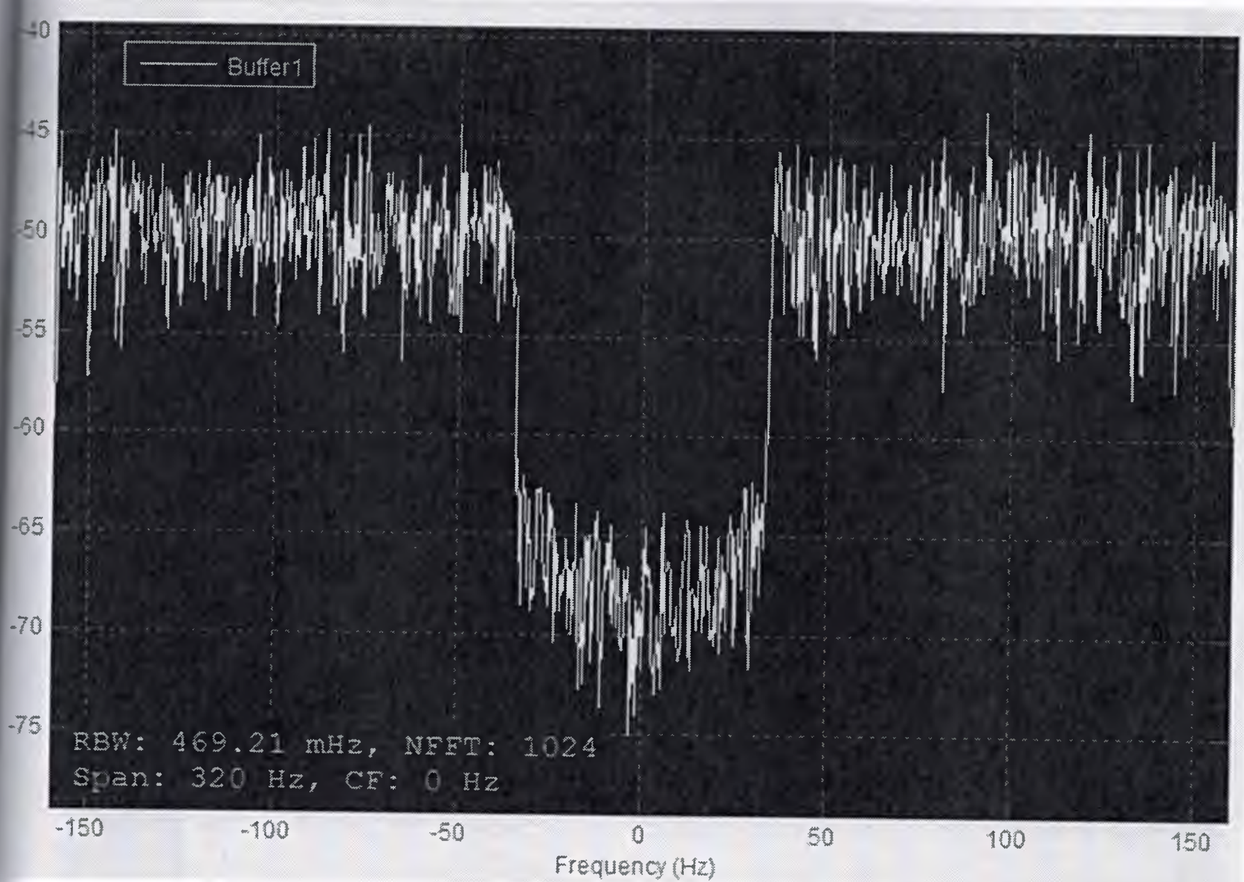


Figure 3.14: Received OFDM Signal

D. Received 16 QAM Signal

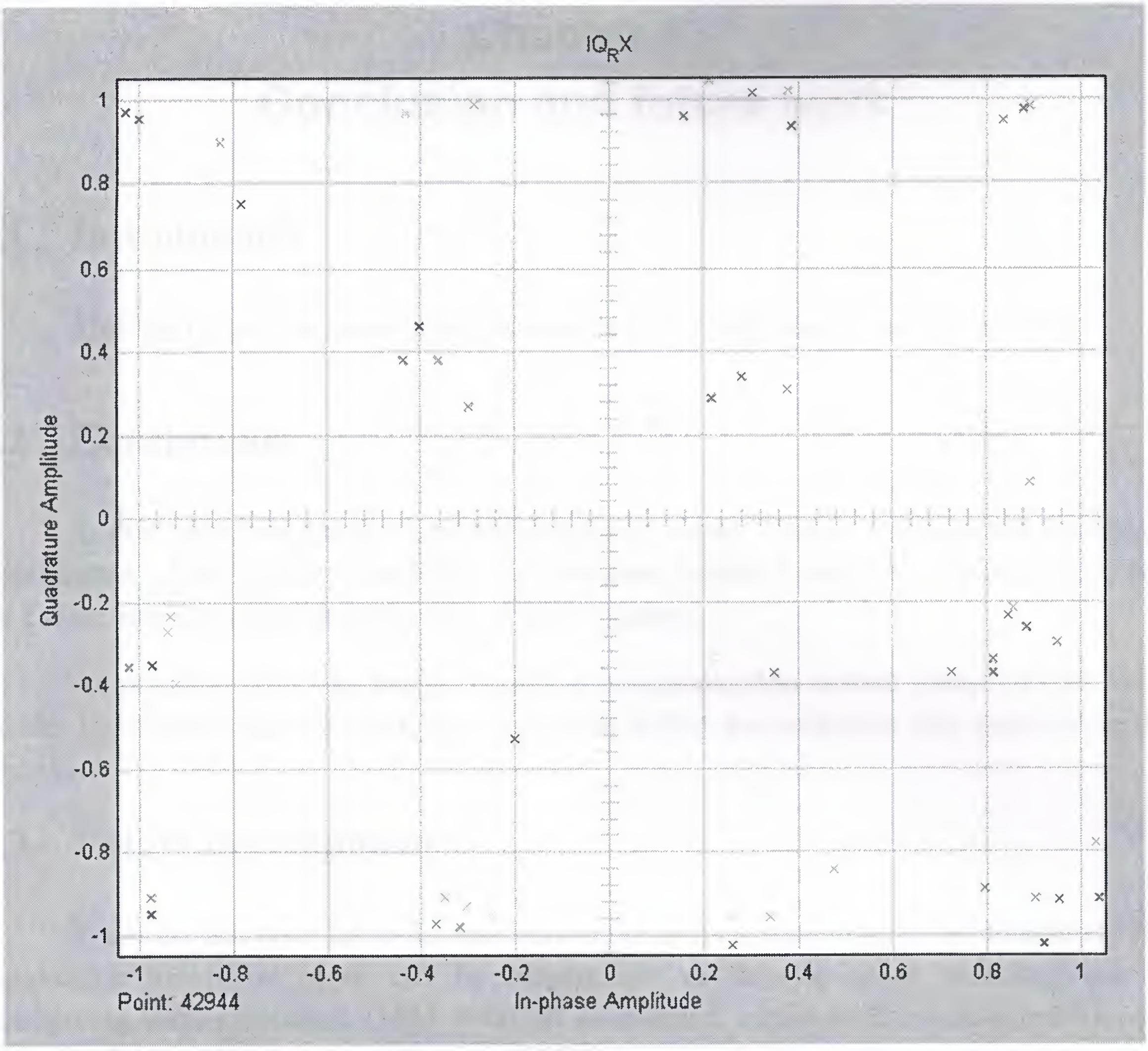


Figure 3.15: Received 16 QAM Signal

E. Performance Measurement

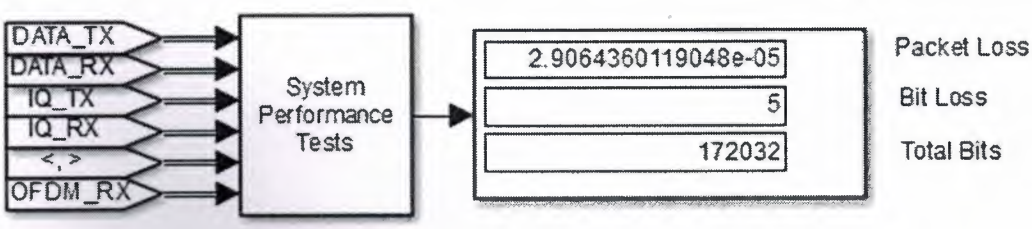


Figure 3.16: Performance Measurement

Chapter 4
Conclusion and future work

4.1 Introduction

This last chapter includes conclusion and some of suggested future work.

4.2 Conclusions

In our work, we explain theoretical concept about OFDM system which consist of (Modulation – Fast Fourier Transform – Inverse Fast Fourier Transform - Cyclic prefix) and we list the advantage, and disadvantage of this technique.

As practical work we design complete communication system using OFDM Multi Carrier Modulation and we show the results that reflect the parameter that specified in the simulation.

4.3 Future development

There are always a lot to do, one may leave or miss something to be done later. The suggestions for future work can be summarized as develop other modules such as interleaving, error correction, QAM or QPSK modulation, cyclic prefix module and RF part. These modules will make a complete set of OFDM system for transmitter and receiver.

References

[1] Dusan Matiae, "OFDM as a possible modulation technique for multimedia applications in the range of mm waves", TUD-TVS, 1998.

[2] R.W Chang, "Synthesis of Band limited Orthogonal Signals for Multichannel Data Transmission", Bell System Tech. J., pp.1775-1776, Dec 1996.

[3] B. R. Saltzberg, "Performance of an Efficient Parallel DataTransmission System", IEEE Trans. Comm., pp. 805-811, Dec 1967.

[4] S. B Weinstein and P.M. Ebert, "Data Transmission by Frequency Division Multiplexing Using the Discrete Fourier Transform", IEEE Transactions on Communication Technology", Vol. COM-19, pp. 628-634, October 1971.

[5] A. Peled an A. Ruiz, "Frequency Domain Data Transmission using Reduced Computational Complexity Algorithms", In Proc. IEEE Int. conf. Acoust., Speech, Signal Processing, pp. 964-967, Denver, CO, 1980.

[6] "An Introduction to OFDM", International Engineering Consortium (IEC), http://www.iec.org/online/tutorial/ofdm/topic04.html.

[7] "Orthogonal Frequency Division Multiplexing (OFDM) Explained", Magisn Networks, Inc., February 8, 2001, www.magisnetworks.com.

[8] Erich Cosby, 2001, "Orthogonal Frequency Division Multiplexing (OFDM): Tutorial and Analysis", www.eng.jcu.edu.au/eric/thesis/Thesis.htm.

[9] Implementation of 8 point IFFT and FFT for OFDM system, Nor Hafizah Bt Abdul Satar, UTM, 2004.

[10] Advanced Electronic Communication Systems, Wayne Tomasi, 5th edition, Prentice.