IN THE NAME OF ALLAH,

THE BENEFICIENT, THE MERCIFUL

ABSTRACT

The end of the last decade saw an immense growth in the field of information technology, worldwide. The concept of a *'Global Village'* brought people closer from all over the world by providing them instant access to information. The quest for better communication devices has resulted in the production of communication systems that have information transfer characteristics approaching the ideal.

In this project, titled *'Encrypted Data Transmission Though HF/VHF Link'*, we have interfaced a *Windows* based personal computer with an HF transceiver to establish a communication system that transmits text data in encrypted form. The encryption algorithm is the *DES, Data Encryption Standard*, which is a F1PS approved cryptographic algorithm. This standard is currently being used in all Federal Institutions of the United States of America.

This report has been divided into three sections. **Section I** is an introduction to data communication. It contains the necessary background required for the study of this report. **Section II** is a detailed account of Cryptography whereas **Section III** gives the details of the DES Algorithm and the hardware implementation of the project.

ACKNOWLEDGEMENTS

This paper owes it's existence to Tele Communication Engineering Department of Military College of Signals which affiliated with National University of Science and Technology.

Our profound gratitude goes to our project advisor Maj Imtiaz Khokhar who generously devoted her precious time to guide us through her golden advice. We wish to thank our co-project advisor Miss Atiya Obaid for giving us valuable suggestions. We are also thankful to Lec Irtiza for making valuable contributions through his discussions in turbulent times.

Our special thanks to the laboratory staffs of electronic, communication, digital and computer, radio laboratories of MCS for their understanding, cooperation and guidance.

We wish to thank the people at NRTC(National Radio Telecommunication Company),specially Capt Daniyal,who guided us in turbulent times.

We also owe a debt of gratitude to our families, whose love and support made this study possible in a peaceful state of mind. :

DECLARATION

All the contents of this project

TO OUR PARENTS,

OUR SOURCES OF INSPIRATION

CONTENTS

SECTION ONE

CHAPTER THREE:DIGITAL DATA COMMUNICATION

SECTION TWO

CHAPTER FOUR :INTRODUCTION TO DA TA ENCRYPTION

CHA PTER FIVE:DA TA ENCR YPTION METHODS

<u>SECTION THREE</u>

CHAPTER EIGHT:INTERFACING THE PC

BIBLIOGRAPHY

APPENDICES

INTRODUCTION

OVERVIEW OF THE PROJECT

"Attention, the Universe' By kingdoms, right wheel! .These profound words , sent by Samuel F.B. Morse in 1838, form the first telegraph message on record. With the transmission of this message, was born the era of electrical communication. Since then, the field of telecommunications has developed continuously with the passage of time. From the audible world of radio, to the visual pleasures of television, this Held of information transfer has become an integral part of modem human life. The very essence of information transfer lies in the fact, that every single communication device built has to perform a single basic function, that of reliable transfer of data with maximum efficiency and a minimum of loss.

This project, titled *'Transmission of Encrypted Data through HF Channel'* is an attempt to setup a communication system that performs the very fundamental task of data communication, yet incorporated with the modern concept of Data Encryption.

PROJECT OBJECTIVES

This project was conducted for Pakistan Army, Signal corps, with a view of expanding their communication capabilities. Pakistan Army Signals Corps is currently using many types of radio sets of various frequencies(HF/VHF), for voice communication between different stations. Through the use of these VHF channels, the department is able to fulfill their requirements of information transfer with aircrafts and ground stations. Moreover, PATCOM (Pakistan Army Tactical Communication), which works in line of sight range, is being used in the field to establish both voice and data communication.

Signal Corps is using TTY Vans for data communication, in which teleprinters were used. Recently all the formations of Pakistan Army Signals Corps have been provided with computers in their TTY Vans instead of teleprinters.

Under hostile conditions, if PATCOM goes out of order, means we are out of communication for both voice and data simultaneously, which is highly undesirable.

The task, assigned to our group was, to use the available HFVHF transceivers for data transmission in addition to voice. Also, the task additional to this was to encrypt the data so as to prevent from any kind of security threats that may arise. Thus, the objectives of this project are as under:

- **To transmit text data using HF sets currently being used for voice communication**

- **To encrypt the data in transmission**

PROJECT IMPLEMENTATION

For implementation of the project, we used PCs with Windows platform and developed a code, which is not only interfaces with the HF transceivers, but also incorporates encryption. The program, named *Cipher*, enables it's user to encrypt the transmitted data using DES algorithm. *DES* stands for *Data Encryption Standard*, and it is an acclaimed standard for cryptography, currently being used in all Federal Agencies of United States.

The *source code* was developed using *Visual Basic 6.0 programming* environment. The implementation details along with the block diagrams of the system are fully explained in Section III of this report.

PROJECT SCOPE to be ammended

The applications of this project are twofold. Not only does it satisfy the needs of Pakistan Army signal Corps for computer based encrypted data transmission using radio as a standby medium, but with slight modifications, it can also be used for providing internet facility to rural areas.

Another factor which highlights the feasibility of our project is it's **portability.** The minimum power requirements of 12 volts allow it to be set up in a mobile environment such automobiles (TTY Van).

Therefore, we believe that this project can prove to be useful in a number of applications if properly implemented.

SECTION ONE

FUNDAMENTALS OF DATA COMMUNICATION

CHAPTER ONE

COMMUNICATION SYSTEMS

A communication, system is one, which transfers data from one place to another. The term *'data'* refers to a signal, which is usually an electrical manifestation of the information being transferred. The advancement in the field of Telecommunications has resulted in the evolution of complex data transfer systems, however, the basic purpose of each remains the same i.e., efficient transfer of information with a minimum loss. All communication systems can therefore be represented by the block diagram shown in Figure 1-1 (Next Page).

The *Input Message,* shown in the diagram, is produced by a Source, and may take any form such as sound waves, visual signals etc. The Input Transducer then converts these signals into electrical form**.** This electrical signal is known as *a Baseband Signal or Message Signal.* Conversion becomes necessary for the next portion of the system to be able to work on the signal. The *Transmitter,* shown by a single block, is actually a complex whole, consisting of a pre emphasizer, a sampler, a quantizer, a coder and modulator. It alters the nature of the input signal to match the characteristics of the *Channel.* The Channel is a medium for transferring information, such as coaxial cable, optical fiber link, air etc. The channel forms the most important part of a communication system as it not only conveys the data, but also has a deteriorating impact on the signal. It acts partly as a filter to attenuate the signal and distort it's waveform.

The receiving end of a communication system begins by placing a *Receiver.* It performs the converse of the *Transmitter* and consists of a demodulator, decoder, a filter and a de-emphasizer. The *Output Transducer* forms the last stage of the system and reproduces the original message, which may be converted in any desirable form.

Figure1-1?

1.1 - PROBLEMS IN SIGNAL TRANSMISSION

Various unwanted and undesirable effects appear during signal transmission. The length of the channel increases attenuation, which results in signal distortion. Distortion is caused because of different attenuation levels and phase shifts for different frequency components. This results in an overall distorted appearance of the signal. This type of distortion is known as *Linear Distortion.* Another type of distortion commonly encountered by communication systems is *Non-linear Distortion.* Non-linearity in signal transmission is due to attenuation, which varies with the signal amplitude. This may be caused due to the presence of non-linear elements in the transmitter or receiver.

Another important factor, which deteriorates the signal, is *Noise.* It refers to undesirable, unwanted signals, which are random in nature. Noise signals are superimposed on the information-bearing signal and result in a corrupted message. Noise can be classified into two types;

- **External Noise**
- **Internal Noise**

External Noise includes interference from signals transmitted on nearby channels, automobile ignition radiation, fluorescent lights, and natural noises from lightning, electrical storms, solar and intergalactic radiations. External noise or *Interference* is usually caused in Radio systems whose receiving antennas intercept several signals at the same time. External noise can be minimized or even be eliminated.

Internal noise is due to the thermal motion of electrons. At any temperature above absolute zero, thermal energy causes microscopic particles to exhibit random motion. The random motion of charged particles such as electrons generates random currents or voltages caused *Thermal Noise.* This type of noise appears in every communication system.

1.2- TYPES OF SIGNALS

The signals used for transmitting information can be categorized into *Digital Data* and *Analog Data*. These two types of signals determine the kind of circuitry to be used at the sending and receiving end. The choice of the kind of transmission, analog or digital, depends upon a number of factors.

Analog signals are continuously varying waves of energy. An analog, signal varies continuously with time. For example, the temperature or the atmospheric pressure of a certain location can vary over a continuous range and can assume infinite possible values.

Digital signals on the other hand vary discretely in time and are constructed with a finite number of symbols. For example, a Morse-coded telegraph message is a digital message constructed from only two symbols, a Mark and a Space. It is therefore a *binary* message. implying only two symbols.

1.3 -MODES OF CHANNEL OPERATION

The direction of data transmission determines the mode in which a channel is operating. Channel operation has the following modes :

- ***Simplex Transmission***
- ***Half Duplex Transmission***
- ***Full Duplex Transmission***

1.3.1-Simplex Transmission

Data in a simplex channel is always one way. Simplex channels are not often used because it is not possible to send back error or control signals to the transmit end. An example of simplex is Television. or Radio.

1.3.2 - Half Duplex Transmission

A half-duplex channel can send and receive, but not at the same time. Only one end transmits at a time, the other end receives. In addition, it is possible to perform error detection and request the sender to retransmit information that arrived corrupted.

1.3.3 - Full Duplex Transmission

In full duplex mode, data can travel in both directions simultaneously. There is no need to switch from transmit to receive mode like in half duplex.

Baseband signals produced by various information sources are not always suitable for direct transmission over a given channel. These signals are usually further modified to facilitate transmission. This conversion process is known as *Modulation.*

CHAPTER TWO
MODULATION

Modulation is a technique used to alter some characteristics of a signal so as to achieve efficient and reliable information transfer. Modulation involves two waveforms, a *Modulating Signal* that represents the message, and a *Carrier Wave* that suits the particular application. A modulator systematically alters the carrier wave in correspondence with the variations of the modulating signal. The resulting *Modulated Wave* thereby carries the message information. It is required that modulation be a reversible process, so that the message can be retrieved by the complementary process of de-modulation .

2.1-MODULATION AND DIGITAL TRANSMISSION

A band pass channel like a telephone channel can transmit sine waves within its bandwidlh but can't transmit digital pulses. If a sequence of binary signals were presented to one end of a telephone network, the digital signals would be so severely distorted that they would be unrecognizable at the receiving end of the circuit. Because the telephone network can transmit voice-band signals in the range 300 to 3,300 Hz, various ways of converting digital information | into speech-like signals have been investigated.

2.1.1 - Amplitude Modulation

Figure 2.1 (Next Page) shows how the digital data can be used to change, or *modulate*, the amplitude of a sine wave in sympathy with a digital signal. This technique is known as *amplitude modulation* or AM. The equipment needed to generate such a signal is called a *modulator* and that needed to extract the digital data from the resulting signal is called a *demodulator*. The interface between a computer and a telephone system is called a MODEM (*modulator-demodulator*). Because AM is more sensitive to noise (i.e interference) than other modulation techniques, it is not widely used in data transmission.

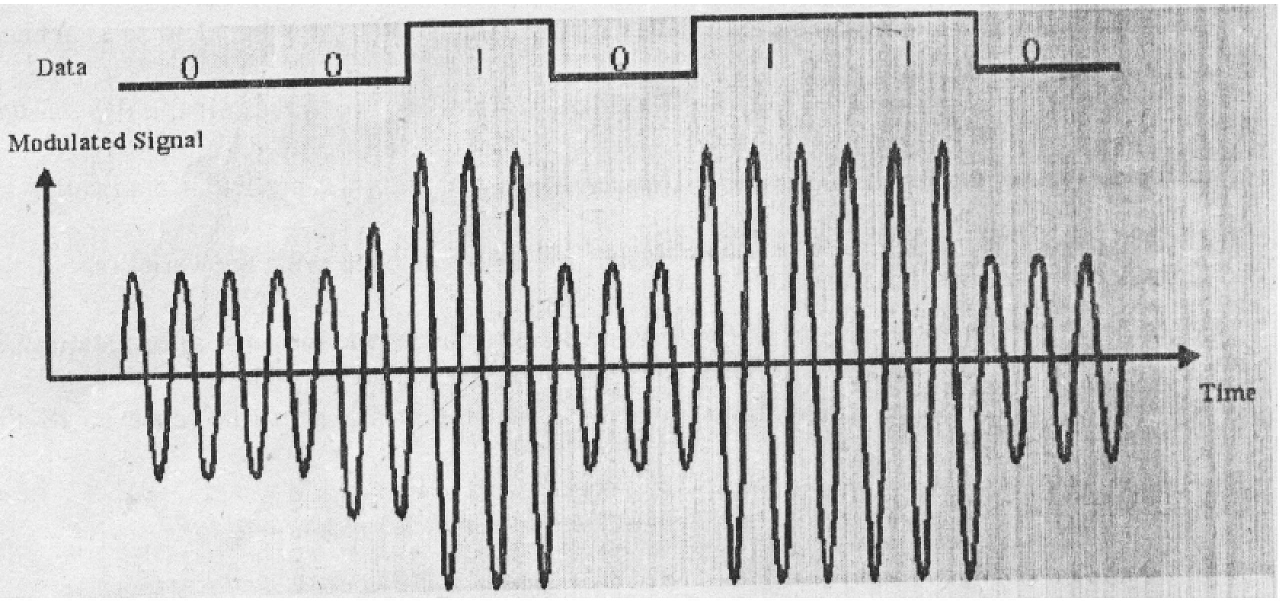

Figure 2-1: Amplitude Modulalion

2.1.2 - Frequency Modulation

Instead of modulating a sine wave by changing its amplitude, it's possible to change its frequency with the digital data. In a binary system, one frequency represents one binary value and a different frequency represents the other. Figure 2.2 (Next Page) shows a *frequency modulated* (FM) signal. FM is widely used because it has a better tolerance to noise than AM (i.e., it is less affected by various forms of interference).

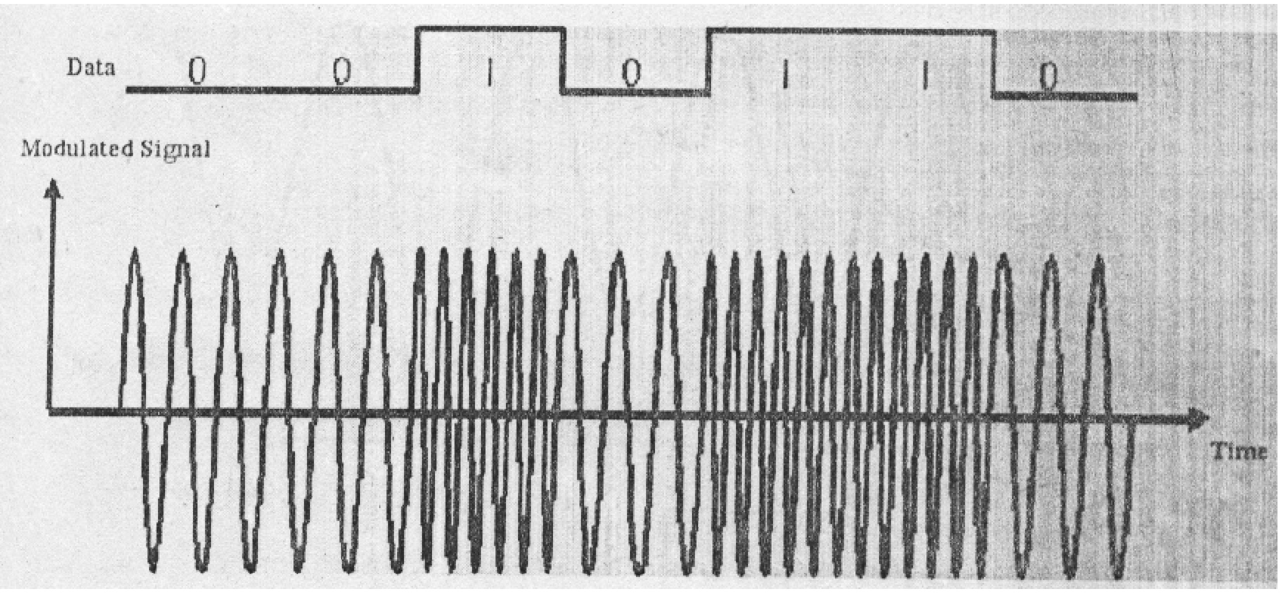

Figure2-2:Frequency Modulation

2.1.3 - Phase Modulation

Figure 2.3-illiisrrates another form of modulation called *phase modulation (* PM).In this case, the *phase* of the sine wave is changed in sympathy with the digital signal. PM is widely used and has fairly similar characteristics to FM. If the phase change corresponding lo a logical I is 180° , and 0°(no change) corresponds lo a logical 0, one bit of information can be transmitted in each time slot (Figure 2.3)- If, however, the phase is shifted by multiples of 90°, two bits at a time can be transmitted (Figure 2,4 - Next Page).

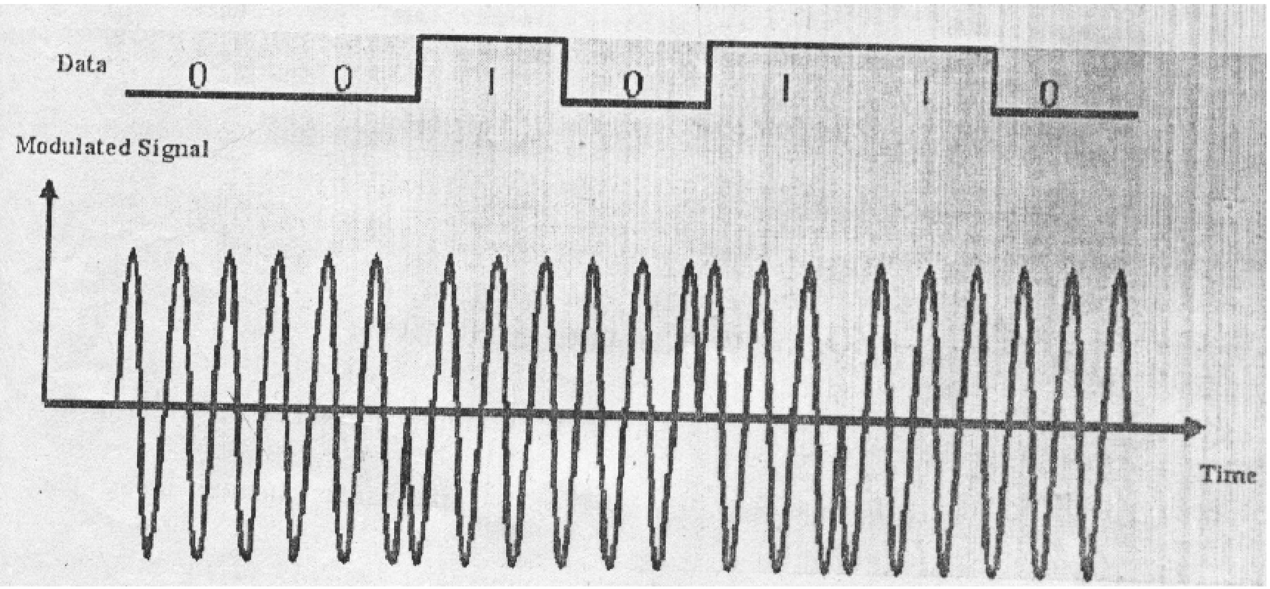

Figure2-3:Phase Modulation

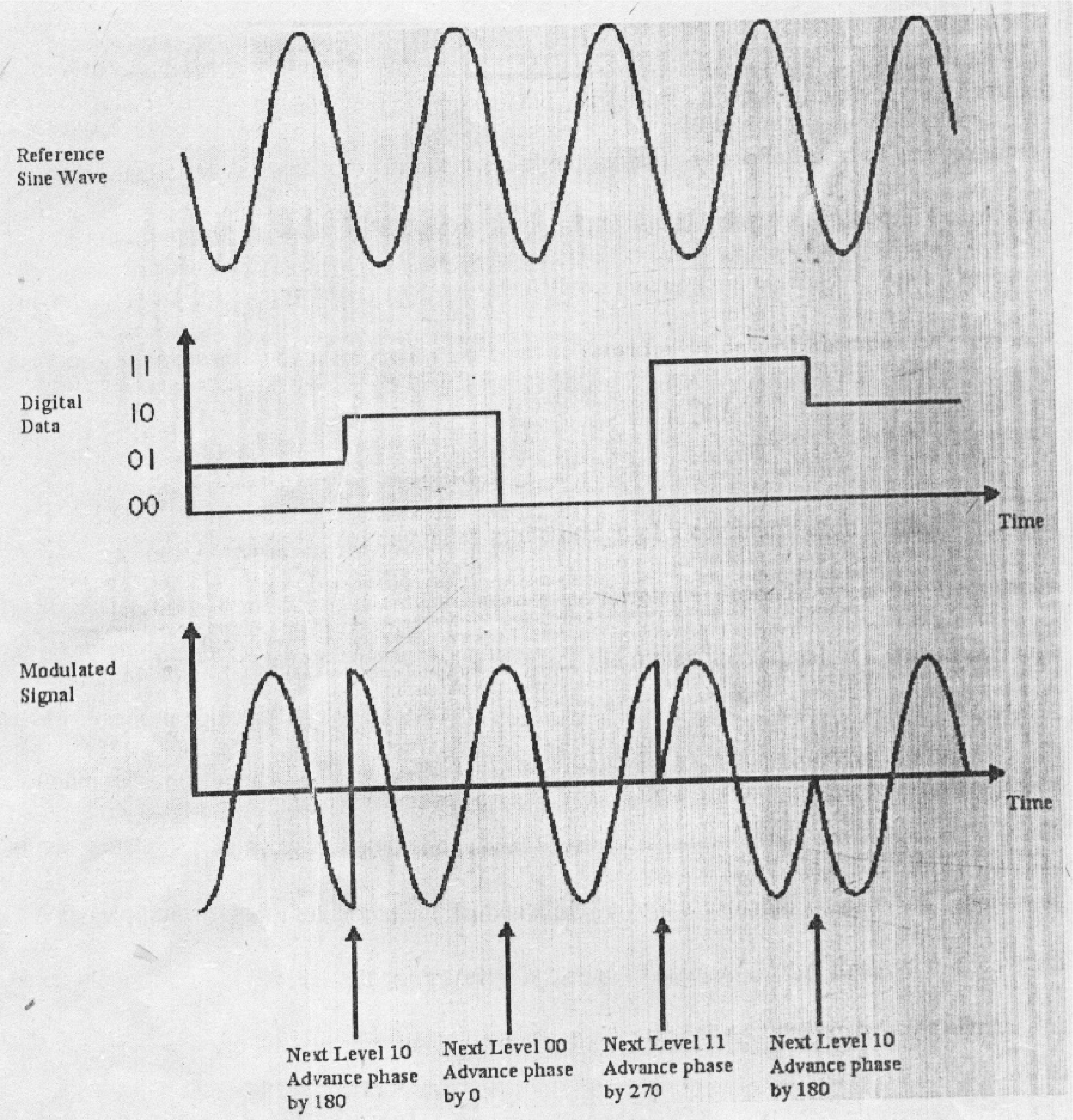
Reference Sine Wave
Digital Data
11
10
01
00
Time
Modulated Signal
Time
Next Level 10 Advance phase by 180
Next Level 00 Advance phase by 0
Next Level 11 Advance phase by 270
Next Level 10 Advance phase by 180

Figure2-4:Differential Phase Modulation

2.2 - MODULATION BENEFITS AND APPLICATIONS

The primary purpose of modulation in a communication system is to generate a modulated signal suited to the characteristics of the particular channel being used for transmission. Other practical benefits and applications and benefits of modulation are as follows:

2.2.1 - Efficient Transmission of Signals

Signal transmission over appreciable distances always involves a traveling electromagnetic wave, with or without a guiding medium. The efficiency of any particular transmission method depends upon the frequency of the signal being transmitted. By exploiting the frequency translation property of *Continuous Wave modulation*, message information can be impressed on a carrier whose frequency has been selected for the desired transmission method.

2.2.2 - Minimum Hardware Limitations

One of the major limitations faced in the design of a communication system is the cost and availability of hardware, hardware whose performance depends upon the frequencies involved. Modulation permits the designer to place the transmitted signal in a frequency range will avoids hardware limitations.

2.2.3 - Reduction of Noise and Interference

FM. and certain other types of modulation have the valuable property of suppressing both noise and interference. This property is called *Wideband Noise Reduction* because it requires the bandwidth of the channel to be much greater than that of the modulating signal.

2.2.4 - Multiplexing

Multiplexing is a process that allows simultaneous transmission of multiple signals over the same channel. *Frequency Division Multiplexing* (FDM) uses *Continuous Wave (CW) modulation* to put each signal on a different carrier frequency, and a bank of filters separates the signals at the destination, *rime Division Multiplexing* (TDM) uses pulse modulation to put samples of different signals in overlapping time slots.

As pointed out previously modulation is used for both analog and digital transmission. To study the transmission of digital data using modulation techniques, it is necessary to study the fundamentals of digital transmission.

CHAPTER THREE

DIGITAL DATA COMMUNICATION

This chapter briefly discusses the differences between two different methods of Serial transmission, namely, asynchronous and synchronous. A protocol establishes a means of communicating between two systems. As long as the sender and receiver each use the same protocol, information can r-e reliably exchanged between them- There are two common protocols used in Serial data communications, the first is known as *Asynchronous*, the second as *Synchronous.*

3.1 - PROTOCOLS

A protocol is a set of rules which governs how data is sent from one point to another. In data communications there are widely accepted protocols for sending data. Both the sender and receiver must use the same protocol when communicating. One such rule is:

BY CONVENTION, THE LEAST SIGNIFICANT BIT IS TRANSMITTED FIRST

3.2 - ASYNCHRONOUS TRANSMISSION

The asynchronous protocol evolved early in the history of telecommunications. It became popular witli the invention of the early tele-typewriters that were used to send telegrams around the world.

Asynchronous systems send data bytes between the sender and receiver by packaging the data in an envelope. This envelope helps transport the character across the transmission link that separates the sender and receiver. The transmitter creates the envelope, and the receiver uses the envelope to extract the data. Each character (data byte) the sender transmits, is preceded with a start bit, and suffixed with a stop bit. These extra bits serve to synchronize the receiver with the sender.

In *asynchronous* serial transmission, each character is packaged in an envelope, and sent cross a single wire, bit by bit, to a receiver. Because no signal lines are used to convey clock (timing) information, this method groups data together into a sequence of bits (five-eight).then prefixes them with a start bit and appends the data with a stop bit.

Figure 3.1 shows the waveform corresponding to a single seven-bit character. In an asynchronous serial transmission system, the clocks at the transmitter and receiver responsible for dividing the data stream into bits are not synchronized. The output from the transmitter sits at a mark state whenever data is not being transmitted and the line is idle. The term *murk* belongs to the early days of data transmission and is represented by a -12V in many systems operating over short distances.

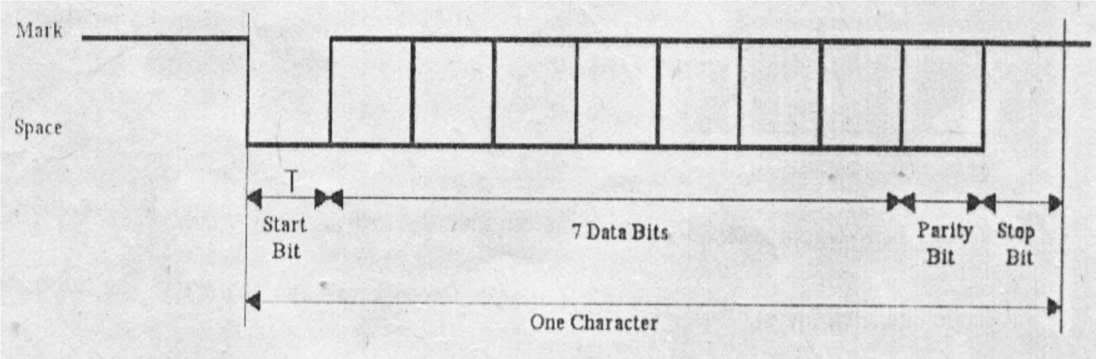

Figure3.1: Asynchronous Transmission

When the receiver sees this logical 0, called a *start bit,* it knows that a character is about to allow. The incoming data stream can then be divided into seven bit periods and the data sampled at the center of each bit. The receiver's clock is not synchronized with the transmitter’s clock and the bits are not sampled exactly in the center.

After seven data bits have been sent, a *parity bit* is transmitted to give a measure of error protection. If the receiver finds that the received parity does not match the calculated parity, an error is flagged and the current character rejected. The parity bit is optional and need not be transmitted.

One or two *stop bits* at a logical 1 level follow the parity bit. The stop bit carries no information and serves only as a spacer between consecutive characters. After the stop bit has been transmitted, a new character may be sent at any time. If the duration of a single bit is *T* seconds, the length of a character is given by start bit plus seven data bits plus the parity bit plus the stop bit = *10T.* Asynchronous transmission is clearly inefficient, since it requires ten data bits to transmit seven bits of useful information.

3.3 - SYNCHRONOUS TRANSMISSION

One of the problems associated with asynchronous transmission is the high overhead associated with transmitting data. Another problem is the complete lack of any; form of error detection. This means the sender has no method of knowing whether the receiver is correctly recognizing the data being transmitted.

In synchronous transmission, grouping characters together, and doing away with the start achieve greater efficiency and stop bits for each character. The information is still enveloped in a similar way as before, but this time more characters are sent between the start and end sequences. In addition, the start and stop bits are replaced with a new format that permits greater flexibility. An extra ending sequence is added to perform error checking.

A shirt type sequence, called a *header,* prefixes each block of characters, and a stop type sequence called a *tail,* suffixes each block of characters. The tail is expanded to include a check code, inserted by the transmitter, and used by the receiver to determine if the data block of characters was received without errors. In this way, synchronous transmission overcomes the two main deficiencies of the asynchronous method, that of inefficiency and lack of error detection.

There are variations of synchronous transmission, which are split into two groups, namely *Character orientated* and *Bit orientated.* In character orientated, information is encoded as characters. In bit orientated, information is encoded using bits or combinations of bits, and is thus more complex than the character orientated version. *Binary Synchronous* is an example of character orientated, and *High Level Data Link Control (HDLC)* is an example of bit oriented.

In asynchronous transmission, if there was no data to transmit, nothing was sent. However in synchronous transmission, because the start bit has been dropped, the receiver must be kept in a state of readiness. This is achieved by sending a special code by the transmitter whenever it has no data to send.

Figure3-2?

SECTION TWO

CYPTOGRAPHY

CHAPTER FOUR

INTRODUCTION TO DATA ENCRYPTION

Within the last decade, there has been a vast increase in the accumulation and communication of digital computer data in both the private and public sectors. Much of this information has a significant value, either directly or indirectly, and requires protection. It is common lo find data transmissions, which constitute monetary transfers of billions of dollars daily. Sensitive information concerning individuals, organizations, and corporate entities is collected by Federal agencies in accordance with statutory requirements and is processed in computer systems. This information requires some type of protection, and the authority responsible for the data may specify cryptographic protection.

The rapid growth of computer data banks increases the potential threats to personal privacy. Since data banks often are accessible from remote computer terminals, there is a threat of easy and unauthorized access to personal information from any place in the data communications system. Such information has typically been scattered in remote locations, controlled under separate auspices, and physically or administratively protected. With a telecommunications network of computer systems, what was previously a laborious job of assembling comprehensive dossiers on individuals may become a simple task Thus, both valuable and sensitive information require protection against unauthorized disclosure and modification.

Encryption is a tool, which may be used in data security applications. It is not a panacea. With improper implementation and use, data encryption may only provide an illusion of security. With inadequate understanding of encryption applications, data encryption could deter the utilization of other needed protection techniques. However, with proper management controls, adequate implementation specifications, and applicable usage guidelines, data encryption will not only aid in protecting data communications but can provide protection for a myriad of specific data processing applications.

4.1 - DATA ENCRYPTION

4.1.1 - What is Data Encryption?

Data encryption is a process used to hide the true meaning of data. The word *Encryption* has been coined from the word *Cryptography*, which was derived from the ancient Greek words *kryptos* (hidden) and *graphia* (writing). Encryption is the process of transforming text or data into an unintelligible form called cipher. Reversing the process of encryption and transforming the cipher back into its original form is called *Decryption.* Encryption and decryption comprise the science of cryptography as it is applied to the modern computer.

4.1.2 - How Is Data Encryption Achieved?

Data encryption is achieved through the use of an algorithm that transforms data from its intelligible form to cipher. An algorithm is a set of rules or steps for performing a desired operation. An algorithm can be performed by anything that can be taught or programmed to perform a specific and unambiguous set of instructions.

4.1.3-Where Should Data Encryption Be Used?

Cryptography (Encryption) has historically been used to protect sensitive information in communication. It can be used for protecting computer data transmitted between terminals and computers or between computers. Data is encrypted before transmission and decrypted after it is received. The algorithm used to decrypt the received cipher must be the inverse of the algorithm used to encrypt the transmitted data. In general, a device used to transmit and receive data would contain algorithms for both encryption and decryption.

Encryption can be used between data processing machines and data storage devices such as magnetic tape and magnetic disk. In this application, the data is encrypted before it is written on the storage device and decrypted before it is subsequently read.

Encryption can be used to authenticate the identities of users, terminals, and computers of a data processing system. Passwords have historically been used to differentiate between friend and foe during times of war. Knowledge of the secret password was accepted as authenticating the identity of friends. Unique identification was not necessary and the password was changed for each mission. The DES uses a key, similar to a password, which must be supplied to each group of users of the algorithm. Having the correct key authenticates an individual to a data processing system.

In a similar manner a terminal or a computer may be authenticated as an authorized device of a data processing system. Supplying the correct key to a DES device when requested by the authorization system can authenticate a terminal associated with the device. This authorization system may be a special program or a special computer system, which has been established to control access to the resources and data of the overall system. The authorization system must be initialized with the identities and the authentication keys of all authorized users and devices of the system. This system will issue a challenge for proper identification whenever a device or individual wishes to access the system. Similar password systems are currently in use for computer user authentication. When combined with data encryption technology, authorization systems can authenticate the claimed identities of users and devices without compromising the, passwords or keys by transmitting them through the system.

4.1.4-When Should Data Encryption Be Used?

Data encryption should be used whenever it is the most cost effective method available to protect the confidentiality or integrity of the data. Confidentiality refers to the accidental or intentional disclosure of data to an unauthorized individual. Integrity refers to data, which has not been exposed to accidental or malicious alteration or destruction. Encryption of data prevents unauthorized recipients of the cipher from interpreting its meaning. Encryption can also prevent unauthorized individuals from manipulating the cipher in such a way that the original data is changed in a predetermined manner. To be effective, encryption must cost less than the expected loss if the protection were not provided. Computation or estimation of costs and risks and the decision to employ cryptographic protection are management functions of the authority responsible for the data.

4.1.5 – Necessity of a Data Encryption Standard ?

A data encryption standard is needed to protect sensitive or valuable data within Federal computer systems and networks. Data encryption techniques are needed for controlling access to sensitive data in multi user computer systems, for protecting the integrity of transactions in national and international monetary transfer systems, for disguising sensitive data during transmission, and for authenticating the users and devices of distributed computer systems and networks. A myriad of different encryption algorithms would result: in a-fundamental incompatibility of data communications equipment. Research and development M cryptographic algorithms are difficult areas; redundant and unusable results often occur. The Data Encryption Standard provides a basic method for more effective computer utilization and a high level of protection for computer data.

CHAPTER FIVE

DATA ENCRYPTION METHODS

Encryption is the transformation of data from its original, intelligible form to an unintelligible cipher form. Two basic transformations may be used: *Permutation* and *Substitution.* Permutation changes the order of the individual Symbols comprising the data. In a substitution transformation, the symbols themselves are replaced by other symbols. During permutation the symbols retain their identities but lose their positions. During substitution the symbols retain their positions but lose their original identities.

The set of rules for a particular transformation is expressed in an algorithm. Basic transformations may be combined to form a complex transformation. In a computer system the symbols of the data are groups of one or more binary digits ("1 "s and "0"s) called bits. A group of bits is called a byte. In computer applications the encryption transformation of permutation reorders the bits of the data. The encryption transformation of substitution replaces one bit with another or one byte with another.

5.1 - ENCODING AND ENCIPHERING

Coding or encoding, in a non-cryptographic sense, is often used to mean changing from one intelligible form to another. The American Standard Code for Information Interchange (ASCII) and Morse code are examples of non-cryptographic codes. Reducing the length of a data element without removing any of its information content is called *Compression.*Expanding the length of a data element is usually done for error detection and correction purposes. Even though the form of the data is changed, no attempt is made to prevent unauthorized decoding.

Within basic encryption transformation classes, encoding is usually distinguished from enciphering. A code is a correspondence between code words and data elements. A data element may be a letter, a syllable, a word, a phrase, or a special symbol. Codebooks generally consist of two sections one alphabetized on the data elements for use in encoding and the second alphabetized on the code *words* for use in decoding. Encoding consists of looking up every data element of a message to be transmitted and substituting its codeword equivalent to produce the encoded message. Decoding consists of finding the received code words in the codebook and replacing them with their equivalent data elements, thus reconstructing the original message.

A codebook may be automated to perform the encode and decode functions as just described or an algorithm may be used to automatically encode and decode without looking up the corresponding values in tables. The latter method is preferred when automation is feasible because encoding and simply computing the code equivalent each time it is needed rather than storing an enormously large codebook can perform decoding rapidly.

Enciphering consists of an algorithmic computation involving the *data* itself. The original plaintext data may either be used directly in the computation or may be combined with the results of the computation to form cipher. The cipher that results from such a transformation is generally the same length as the original data that is enciphered.

Ciphers may be thought of as operating on data elements of fixed length and codes as operating on data elements of variable length. Another useful distinction is that a code typically operates on linguistic entities (words) while a cipher operates on syntactic entities (letters or groups of letters). In general computer applications, bits or bytes are used in data encryption algorithms without regard to their linguistic content. Thus the computer encryption transformation of a fixed number of bits or bytes is generally called enciphering.

5.1.1-Block Ciphers

A cipher that is produced by simultaneously transforming a group of message bits into a group of cipher bits is called a *Block Cipher*. In general, the groups are the same size.

5.1.2 - Product Ciphers

Combining the basic transformations of permutation and substitution produces a complex transformation termed a *Product Cipher.* If permutation and substitution operations are applied to a block of data, the resulting cipher is called a product cipher.

5.1.3 - Recirculating Block Product Cipher

Using a permutation operation and a substitution operation alternately, and recirculating the output of one pair of operations back into the input for some number of iterations may construct a block product cipher. Each iteration is called a *round. A* cipher produced in this way is termed a *Recirculating block product cipher.* If a recirculating block product cipher is properly constructed with an unknown key, then the alteration of a single bit of the plaintext block will unpredictably alter each bit of the cipher text block. Altering a bit of the cipher text will also result in an unpredictable change to the plaintext block after decryption.

5.2 - CHARACTERISTICS OF THE DES ALGORITHM

As mentioned before, the encryption algorithm which we implemented is the Data Encryption Standard (DES). Therefore, it will be useful to mention the characteristics of this algorithm, while studying the topic of basic encryption techniques.

The DES algorithm is a recirculating, 64-bit, block product cipher whose security is based on a secret key. DES keys are 64-bit binary vectors consisting of 56 independent information bits and eight parity bits. The parity bits are reserved for error detection purposes and are not used by the encryption algorithm. The 56 information bits are used, by the enciphering operations and are referred to as the *Active Key.* Active keys are generated (selected at random from all possible keys) by each group of authorized users of a particular computer system or set of data. Each user should understand that the key must be protected and that any compromise of the key will compromise all data and resources protected by that key.

In the encryption computation the 64-bit data input is divided into two halves each consisting of 32 bits. One half is used as input to a complex nonlinear function, and the result is it exclusive OR’ed to the other half. After one iteration, or round, the two halves of the data are swapped and the operation is performed again. The DES algorithm uses 16 rounds to produce a recirculating block product cipher. The cipher produced by the algorithm displays no correlation to the input. Every bit of the output depends on every bit of the input and on every bit of the active key.

An important characteristic of the DES algorithm is its flexibility for usage in various data processing applications. Each cipher block is independent of all others allowing encryption or decryption of a single block in a message or data structure. Random access to encrypted data is therefore possible. The algorithm may be used in this straightforward way to form a block cipher or alternatively used with chaining in which the output of the algorithm depends on previous results of the algorithm. The first technique is called the *Electronic Codebook(* ECB) mode and the chaining technique has two examples called the *Cipher Block Chaining* (CBC) mode and the *Cipher Feedback* (CFB) mode. In addition, DES may be used in the *Output Feedback* (OFB) mode to generate a pseudorandom stream of bits which is exclusive OR'ed to the plaintext bits to form cipher.

The DES algorithm is mathematically a one-to-one mapping of the 2" possible input blocks onto all 2^{64} possible output blocks. Since there are 2^{56} possible active keys, there are 2^{56} possible mappings. Selecting one key selects one of the mappings.

The input to the algorithm is under complete specification of the designer of the cryptographic system and the user of the system/Any pattern of 64. bits is acceptable to the algorithm. The format of a data block may be defined for each application. The DES algorithm is composed of two parts: the enciphering (encryption) operation and the deciphering (decryption) operation. The algorithms are functionally identical except that the selected portion of the key used for rounds 1,2,..., 16 during the encryption operation are used in the order 16.15,...,! for the decryption operation. The algorithm uses two 28-bit registers called C and D to hold the 56-bit active key. The key schedule of the algorithm circularly shifts the C and D registers independently, left for encryption and right for decryption. If the bits of die C register are all zeros or all ones (after Permuted Choice 1 is applied to the key) and the bits of the D register are all zeros or all ones, then decryption is identical to encryption. This occurs for four: known keys: (0101010101010101), (FEFEFEFEFEFEFEFE), (IF1F1F1FOEOEOEOE), and (EOEOEOEOFI.FIF1FI). It is likely that, in all other cases, data encrypted twice with the same *key* with not result in plaintext (the original, intelligible data form). This characteristic is beneficial in some data processing applications in that several levels of encipherment can be utilized in a computer network even though some of the keys used could be the same. If an algorithm is its own inverse, then an even number of encryptions under the same key will result in plaintext.

CHAPTER SIX

SECURITY THROUGH ENCRYPTION

Encryption may be implemented in a computer system in-order to combat several possible threats to the security of computer data. These threats are generally categorized as *Transmission threats* and *Storage threats.* Security against these threats is generally termed *Communication security or File security.* The DES algorithm can be used in both applications hut the key will be handled differently.

6.1- TRANSMISSION THREATS

Encryption can be used to prevent the disclosure of data and to detect the modification of transmitted data. Encryption will not combat the threats of accidental or deliberate destruction. Encrypted data can be lost or destroyed as easily as unencrypted data. Adequate backup facilities or copies must be provided to recover from the destruction of either encrypted or unencrypted data. In addition, destruction or loss of the key used to encrypt data is equivalent, to the loss or destruction of the data itself.

The following is a list of threats that are countered with the encryption of transmitted data.

6.1.1 - Spoofing

Spoofing is the threat of accepting a false claim of identity. Spoofing by a computer system penetrator is a serious threat at many places in a computer system. The computer's data communication system is especially vulnerable to spoofing. The identities of' terminals, computers, and users can often be simulated so that the receiving device cannot discern a true identity from a falsely claimed identity.. Data encryption can be used for authentication by requiring that a unique encryption key be associated with each identity. Successful communication using this key mutually authenticates the holders of the key (provided that the key lias not been compromised) and thus prevents spoofing. If the key is not known, false messages cannot be correctly generated and entered into the system and hence message spoofing is prevented.

6.1.2 - Misrouting

The threat of misrouting is directly proportional to the complexity of the communication system and inversely proportional to the reliability of its Components. A simple message routing indicator scheme combined with encryption of the routing indicator may be used to detect misrouting, but prevention can only be accomplished with dedicated lines and permanent connections. In any but geographically local systems, the prevention of misrouting is not economically feasible. However, data encryption can prevent the unauthorized use of misrouted data.

6.1.3 - Passive Wiretapping (Monitoring)

Monitoring of messages during data transmission can occur all along the transmission path in any of several ways. Wiretapping or radio reception of the transmitted data is the most common methods. The transmission is not delayed or altered, only monitored or copied. This threat is difficult to combat in any way other than physically protecting the transmission path or encrypting the data. Plaintext is also vulnerable to monitoring due to radiation, conduction, and acoustic pickup during input and output operations. These threats are prevalent in high voltage CRT terminals, electrically connected devices, and mechanical printing or punching devices. Encryption protects the plaintext from disclosure. The encryption devices should be designed to be an integral part of the original source equipment and the final destination equipment whenever possible. The data encryption devices themselves must be physically protected and designed to minimize electronic emanations.

6.1.4 - Active Wiretapping

With this type of communication threat, the communication line is broken, a high speed receiver-transmitter is installed, t<nd the intercepted data is retransmitted unchanged until special "looked for" event causes the tapping mechanism to modify the data so as to have false information accepted as valid. Communications will be slightly delayed while the

data is being modified but this delay is often not detectable because other variable length delays are already in the communication system. Encryption prevents the penetrator from intelligently modifying the cipher so that the decrypted plaintext is ungarbled (i.e readable and acceptable).Special precautions must be utilized to prevent either the playback threat or the substitution threat The former consists simply of copying a valid encrypted message and playing it back (retransmitting it) to the u unsuspecting receiver. If the key has not been changed, the receiver will correctly decrypt the message and may accept it. For certain types of messages (funds deposits, merchandise order, etc.) this could have disastrous results. The substitution threat consists of replacing blocks or characters of the cipher text with other blocks or characters without actually deciphering the data or having the key. The perpetrator substitutes the cipher of known plaintext. This can be accomplished in the block mode if each block is totally independent from all other, and no other block or message authentication system, is used.

6.1.5 Storage Threats

In addition to combating threats to computer data security during transmission among terminals and computers, encryption may be used effectively for protecting computer data during storage, but the system implementation will be different in the two cases. In the transmission case, the cryptographic key must be available at the two participating locations simultaneously and may be destroyed when that transmission is complete.In the storage case, the key need be at only one location but must be retained for reuse when the data is to be retrieved and used. The computer system or the user must be able to provide the key at the appropriate place and at the appropriate time. The following is a list of threats that are countered with the encryption of stored data:

6.1.5.1 -Theft

Encryption of stored computer data provides protection against the disclosure of stolen data. Data may be stolen from on-line devices (disks, mass storage devices, etc.) by unauthorized access, or from off- line devices (magnetic tape, cards, disk packs, etc.) by physically removing the device and reading it on another computer system. In addition if there is a threat of a computer data storage facility or a computer center being taken over by force, bulk encryption of all data using a common key which is easily erased from the encryption device effectively renders the data unreadable and unusable by destroying the key. This key must be kept in a physically secure location (safe, etc.) so that it may be reentered into the encryption device when the facility has been made secure again. User controlled encryption of private data files renders the data unreadable to other system users.

6.1.5.2 - Residue

Data that is left on magnetic media and not erased after it is no longer needed is called *residue*. Erasing computer data on magnetic storage media may be a very time consuming process. Overwriting data, which is to be discarded in a shared system, can use a significant amount of input and output time if done as standard practice. If sensitive data is always stored in the media in an encrypted form, tapes and disk packs may be returned to their supplier when longer needed or the 'scratched' data tapes may be reused without erasing. Merely destroying the key precludes use of the data. System failures during the erasing of magnetic media are no longer a concern if the media are encrypted. Encryption of stored data with the user's private key obviates the need for clearing temporary storage after use.

6.1.5.3 - Addressing Failure

Random access magnetic storage media have a physical addressing mechanism which positions the data under the reading heads and transfers the data. Software data access methods generally have a complex data structure associated with the stored data to optimize access to it. Both of these mechanisms have a small, but non zero, probability of failure. Encrypting the data by combining the location of the data with the key can prevent accidental reading of the wrong data. Applications of this type in the system depend greatly on the implementation ,of the encrypting device in the proper place in system architecture.

6.2 - THE DES SECURITY

The security provided by the DES algorithm is based on the fact that, if the key is unknown, an unauthorized recipient of encrypted data, knowing some of the data, must perform an unacceptable effort to decipher other encrypted data or recover the key. Even having all but one bit of the key correct does not result in intelligible data.

The only known way of obtaining the key with certainty is by obtaining matched cipher text and plain text and then by exhaustively testing keys by enciphering the known plaintext with each key and comparing the result with the known cipher text. Since 56 independent bits are used in a DES key, 2^{56} such tests are required to guarantee finding a particular key. The expected number of tests to recover the correct key is 2^{55}. At one microsecond per test 1142 years would be required. Under certain conditions (not only knowing matched plain text and cipher text but also the complement of the plain text and the resulting cipher text) the expected effort would be reduced to 571 years. The possibility of 2^{56} keys approximately 70 quadrillion) makes the guessing or computing of any particular key very unlikely given that the guidelines for generating and protecting a key provided in this publication are followed. Of course, one can always reduce the time required to exhaust any cryptoalgorithm by having several devices working in parallel. Time is reduced but initial expenses are increased.

SECTION THREE

IMPLEMENTING THE DES

CHAPTER SEVEN

THE DES ALGORITHM

7.1 - INTRODUCTION

The algorithm is designed to encipher and decipher blocks of data consisting of 64 bits under control of a 64-bit key. Blocks are composed of bits numbered from left to right, i.e., the left most bit of a block is bit one.

Deciphering must be accomplished by using the same key as for enciphering, but with the schedule of addressing the key bits altered so that the deciphering process is the reverse of the enciphering process. A block to be enciphered is subjected to an initial permutation IP, then to a complex key-dependent computation and finally to a permutation which is the inverse of the initial permutation IP^{-1}. The key-dependent computation can be simply defined in terms of a function *f,* called the *Cipher Function,* and a function *KS*, called the *Key Schedule.* A description of the computation is given first, along with details as to how the algorithm is used for encipherment. Next, the use of the algorithm for decipherment is described. Finally, a definition of the cipher function/is given in terms of primitive functions which are called the *Selection Functions S,* and the permutation function P. S,, P and KS of the algorithm are contained in Section 7.2.

The following notation is convenient: Given two blocks L and R of bits, LR denotes the block consisting of the bits of L followed by the bits of R. Since concatenation is associative,B_1B_2…..B_8, for example, denotes the block consisting of the bits of B_1 followed by the bits of B_2 ..followed by the bits of B_8.

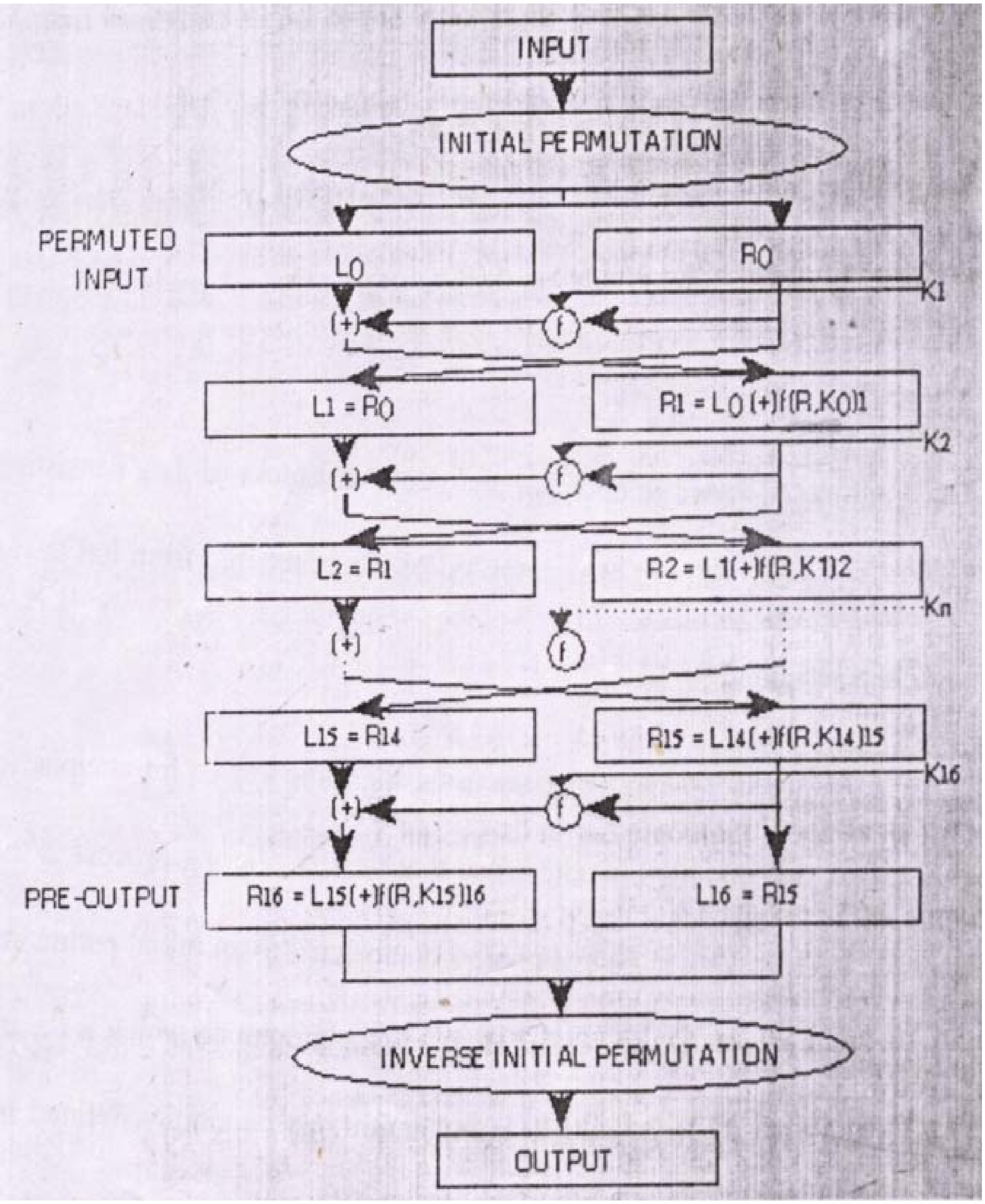

FIGURE 7-1:Enciphering Computation

7.2 - ENCIPHERING

A sketch of the enciphering computation is given in Figure 7.1. The 64 its of the input block to be enciphered are First subjected to the following permutation, called the initial permutation *IP*:

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

That is the permuted input has bit 58 of the input as its first bit, bit 50 as its second bit, and so on with bit 7 as its last bit. The permuted input block is then the input to a complex key dependent computation described below. The output of that computation, called the*preoutput*,| is then subjected to the following permutation which is the inverse of the initial permutation:

IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

That is, the output of the algorithm has bit 40 of the pre output block as its first bit, bit 8 as its second bit, and so on, until bit 25 of the pre output block is the last bit of the output.

The computation, which uses the permuted input block as its input to produce pre output block, consists, but for a final interchange of blocks, of 16 iterations of a calculation that is described below in terms of the cipher function which operates on two blocks, one of 32 bits and one of 48 bits, and produces a block of 32 bits.

Let the 64 bits of the input block to an iteration consist of a 32 bit block ***L*** followed by a 32 bit block ***R***. Using the notation defined in the introduction, the input block is then ***LR*** **.**Let ***K*** be a block of 48 bits chosen from the 64-bit key. Then the output ***LR*** of an iteration with input ***LR*** is defined by:

$$\textbf{(1)} \qquad \mathbf{L' = R}$$

$$\mathbf{R' = L(+)f(R,K)}$$

Where (+) denotes bit-by-bit addition modulo 2.

As remarked before, the input of the first iteration of the calculation is the permuted input block. If ***L'R'*** is the output of the 16th iteration then ***R'L'*** is the pre output lock. At each iteration, a different block ***K*** of key bits is chosen from the 64-bit key designated by ***KEY***.

With more notation we can describe the iterations of the computation in more detail. Let ***KS*** be a function which takes an integer *n* in the range from 1 to 16 and a 64-bit block KEY as input and yields as output a 48-bit block Kn which is a permuted selection of bits from KEY.

That is:

$$(2) \qquad K_n = KS(n,KEY)$$

with ***K_n*** determined by the bits in 48 distinct bit positions of ***KEY***. ***KS*** is called schedule because the block ***K*** used in the ***n^{th}*** iteration of (l) is the block ***K***, determined by (2).

As before, let the permuted input block be ***LR***. Finally, let ***L_0*** and ***R_0*** be respectively ***L*** and ***R*** and let ***Ln*** and ***Rh*** be respectively ***L'*** and ***R'*** of (l) when ***L*** and ***R*** are respectively ***L_{n-1}*** and ***R_{n-1}*** and ***K*** is ***$K_{n:}$*** that is, when *n* is in the range from 1 to 16,

$$(3)\ L_n = R_{n-1}$$

$$R_n\, n = L_{n-1}(+)f(R_{n.l}, K_n)$$

The preoutput block is then $R_{16}L_{16}$.

The key schedule KS of the algorithm is described in detail in Section 7.4. The key schedule produces the 16 Kn which are required for the algorithm.

7.3 – DECIPHERING

The permutation IP"' applied to the pre output block is the inverse of the initial permutation IP applied to the input. Further, from (1) it follows that:

$$(4) \quad R=L' \\ L=R'(+)f(L',K)$$

Consequently, to decipher it is only necessary to apply the very same algorithm to an enciphered message block, taking care that at each iteration of the computation the same block of key bits K is used during decipherment as was used during the encipherment of the block. Using the notation of the previous section, this can be expressed by the equations:

$$(5) \qquad R_{n-1} = L_n \\ L_{n-1} = R_n \ (+) \ f(L_n, K_n)$$

where now $\mathbf{R_{16}L_{16}}$, is the permuted input block for the deciphering calculation and $\mathbf{L_0}$ and $\mathbf{R_0}$ is the preoutput block. That is, for the decipherment calculation with $\mathbf{R_{16}L_{16}}$ as the permuted input, $\mathbf{K_{16}}$ is used in the first iteration, $\mathbf{K_{15}}$ in the second, and so on, with $\mathbf{K_1}$ used in the 16th iteration.

7.3.1 - The Cipher Function

A sketch of the calculation of ***f*****(R, K)** is given in Figure 7.2

Let **E** denote a function, which takes a block of 32, bits as input and yields a block of 48 bits as output. Let E be such that the 48 bits of its output, written as 8 blocks of 6 bits each, are obtained by selecting the bits in its inputs in order according to the table given below figure 7-2:

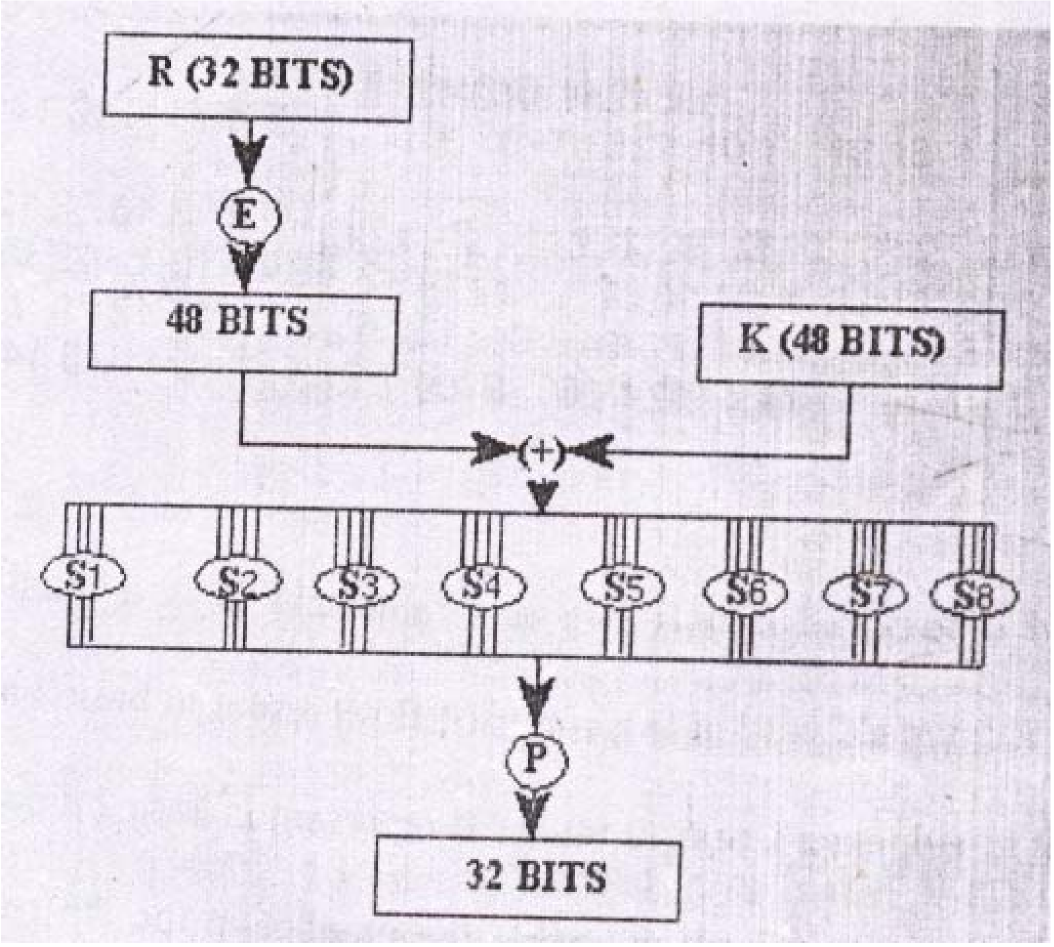

FIGURE7-2: Calculation of **_f_(R, K)**

_E_ BIT-SELECTION TABLE

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Thus the first three bits of **E(R)** are the bits in positions 32, 1 and 2 of **R** while the last 2 its of **E(R)** are the bits in positions 32 and 1.

Each of the unique selection functions **S_1, S_2,….S_8,** takes a 6-bit block as input and yields a 4-bit block as output and is illustrated by using a table containing the recommended **S_1** :

S_1

Column Number

Row No	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

If **S_1** is the function defined in this table and **B** is a block of 6 bits, then **S_1(B)** is determined as follows:

The first and last bits of **B** represent in base 2 a number in the range 0 to 3. Let that number be *i*. The middle 4 bits of **B** represent in base 2 a number in the range 0 to 15.Let that number be *j*. Look up in the table the number in the i^{th} row and j^{th} column. It is a number in the range 0 to 15 and is uniquely represented by a 4-bit block. That block is the output **S_1(B)** of S_1 for the input **B**. For example, for input 011011 the row is 01, that is row 1, and the column is determined by 1101, that is column 13. In row 1 column 13 appears 5 so that the output is 0101. Selection functions S_1,S_2,...,S_8 of the algorithm appear in Section 7.4.

The permutation function P yields a 32-bit output from a 32-bit input by permuting the, bits of the input block. The following table defines such a function:

P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

The output **P(L)** for the function **P** defined by this table is obtained from the input **L** by taking the 16th bit of **L** as the first bit of **P(L)**, the 7th bit as the second bit of **P(L)**, and so on until the 25th bit of **L** is taken as the 32nd bit of **P(L)**. The permutation function P of the algorithm is repeated in the Section 7.4.

Now let $S_{1,}$,...,S_8 be eight distinct selection functions, let **P** be the permutation function and let **E** be the function defined above.

To define ***f*****(R,K)** we first define B_1,...,B8 to be blocks of 6 bits each for which

$$(6) \quad \mathbf{B_1B_2,...B_8 = K(+)E(R)}$$

(7)

The block. ***f*****(R,K)** is then defined to be

$$\mathbf{(8) \quad P(S_1(B_1)S_2(B_2)...S_8(B_8))}$$

Thus **K(+)E(R)** is first divided into the 8 blocks as indicated in (6). Then each $\mathbf{B_i}$ is taken as an input to $\mathbf{S}_i$, and the 8 blocks $\mathbf{(S_1(B_1)S_2(B_2)...S_8(B_8))}$ of 4 bits each are consolidated into a single block of 32 bits which forms the input to **P**. The output (7) is then the output of the function for the inputs **R** and **K**.

7.4 - PRIMITIVE FUNCTIONS FOR THE DATA ENCRYPTION ALGORITHM

The choice of the primitive functions **KS, $S_{1...}$ S_8** and **P** are critical to the strength of an encipherment resulting from the algorithm. Specified below is the recommended set of functions, describing **S_1,...,S_8** and **P** in the same way they are described in the algorithm. For the interpretation of the tables describing these functions, see the discussion in the body of the algorithm. - :

The primitive functions **S_1,...,S_8** are:

S_1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

The primitive function P is:

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

$\mathbf{K_n}$. for 1<=n<= 16, is the block of 48 bits in (2) of the algorithm. Hence, to describe **KS**. it is sufficient to describe the calculation of $\mathbf{K_n}$ from **KEY** for *n* =1, 2, ...,16. That calculation is illustrated in Figure 7.3. To complete the definition of **KS** it is therefore sufficient to describe the two permuted choices, as well as the schedule of left shifts. One bit in each 8-bit byte of the KEY may be utilized for error detection in key generation, distribution and storage.Bits 8, 16...., 64 are for use in assuring that each byte is of odd parity.

Permuted choice I is determined by the following table:

PC-1

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

The table has been divided into two parts, with the first part determining how the bits of $\mathbf{C_0}$ are chosen, and the second part determining how the bits of $\mathbf{D_0}$ are chosen. The bits of **KEY** are numbered 1 through 64. The bits; of $\mathbf{C_0}$ are respectively bits 57, 49, 41,..., 44 and 36 of **KEY**, with the bits of $\mathbf{D_0}$ being bits 63, 55, 47,..., 12 and 4 of **KEY**. With $\mathbf{C_0}$ and $\mathbf{D_0}$ defined, we now define how the blocks $\mathbf{C_n}$ and $\mathbf{D_n}$ are obtained from the blocks $\mathbf{C_{n-1}}$ and $\mathbf{D}_{n-1}$ respectively, for *n* == 1, 2,..., 16. That is accomplished by adhering to the following schedule of left shifts of the individual blocks:

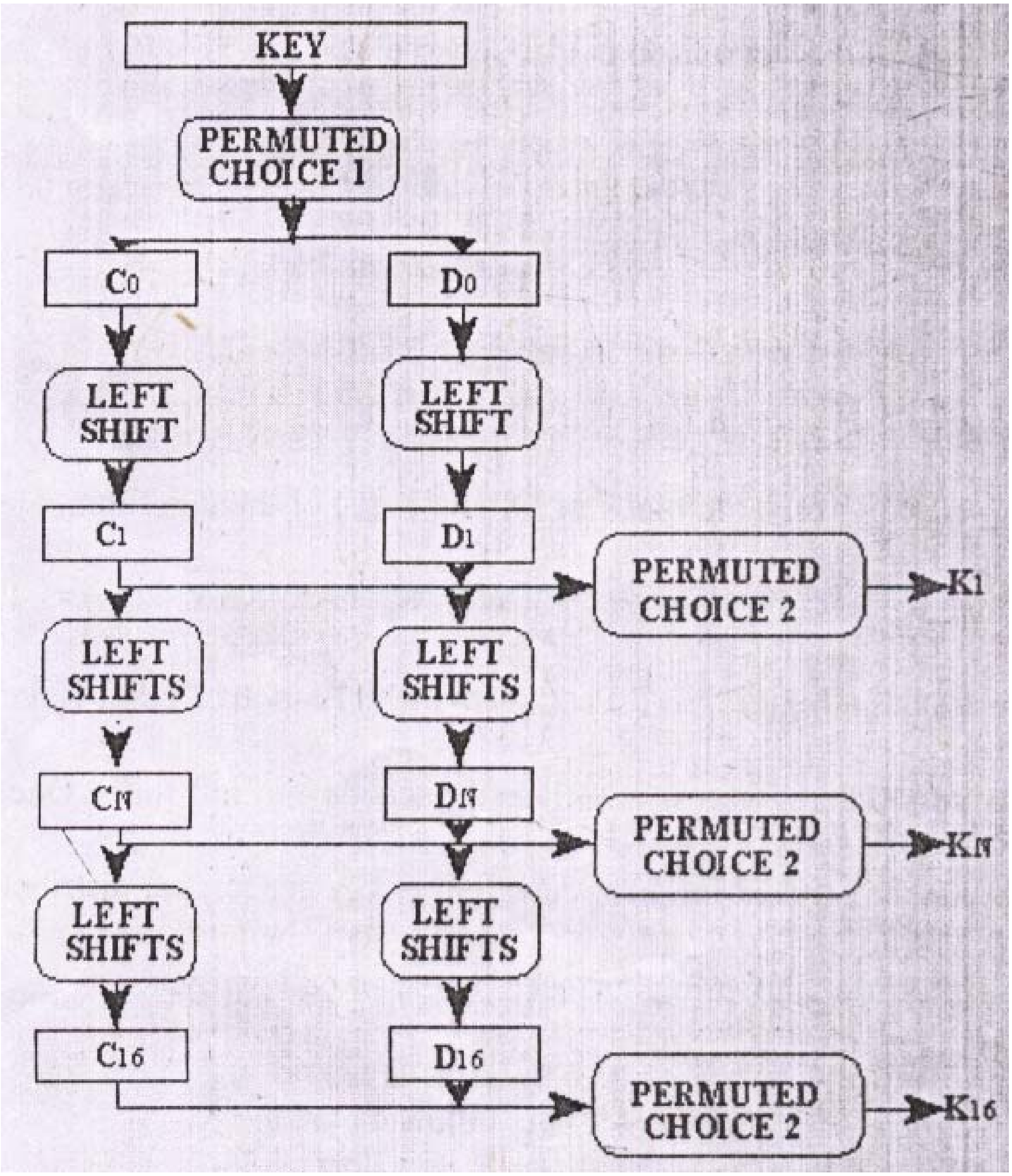

Figure7-3 Key Schedule Calculation

Iteration Number	Number of Left Shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

For example, $\mathbf{C_3}$ and $\mathbf{D_3}$ are obtained from $\mathbf{C_2}$ and $\mathbf{D_2}$, respectively, by two left shifts, and $\mathbf{C_{16}}$ and $\mathbf{D_{16}}$& are obtained from $\mathbf{C_{15}}$ and $\mathbf{D_{15}}$, respectively, by one left shift. In all cases, by a single left shift is meant a rotation of the bits one place to the left, so that after one left shift the hits in the 28 positions are the bits that were previously in positions 2,3,..., 28,1.

The following table determines permuted choice 2:

PC-2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Therefore, the first bit of $\mathbf{K_n}$ is the 14th bit of $\mathbf{C_nD_n}$, the second bit the 17th, and so on with the 47th bit the 29th, and the 48th bit the 32nd.

CHAPTER EIGHT

INTERFACING THE PC

8.1 - BLOCK DIAGRAM

The block diagram of the hardware implementation of the project is shown in Figure 8.1, shown on the next page.

8.1.1 - The Personal Computer

The computer in the project is used to send the data in the form *of typed* phrase, sentence or the data in the bulk form as a file. For this purpose, a software program, named *Cipher* was established in t'ne language Visual Basic 6.0. The typed data or the data in the form of a file is opened in the text editor that is designed in Visual Basic 6.0. Once the data is typed or the file to be sent is chosen, there is present a sent button, which is pressed to send the data.

Once the data is sent by the send button to the serial port, the encryption is performed on the data using the *Data Encryption Standard* **(DES)**. As the port was opened, the 2206 IC via Maxim 232 receives the encrypted data.

The encrypted data is transmitted. When EXAR 2211 IC receives encrypted data, this data is sent to the computer via maxim 232. This encrypted data is stored in the buffer of the serial port for a while. If in this time the receive button in the receiver computer is pressed then the encrypted data transmitted is first decrypted and then displayed on the screen of the receiver computer.

Again the coding in the receiver computer is done in visual basic 6.0. The function of the computer has been explained for simplex data communication. For duplex communication, there is present both, the **Send** button as well as the **Receive** button in the receiver as well as the transmitter computer. The software is designed in the Visual Basic 6.0 due to the reason that this language provides a way of serial communication in a beautiful environment with a very easy coding technique.

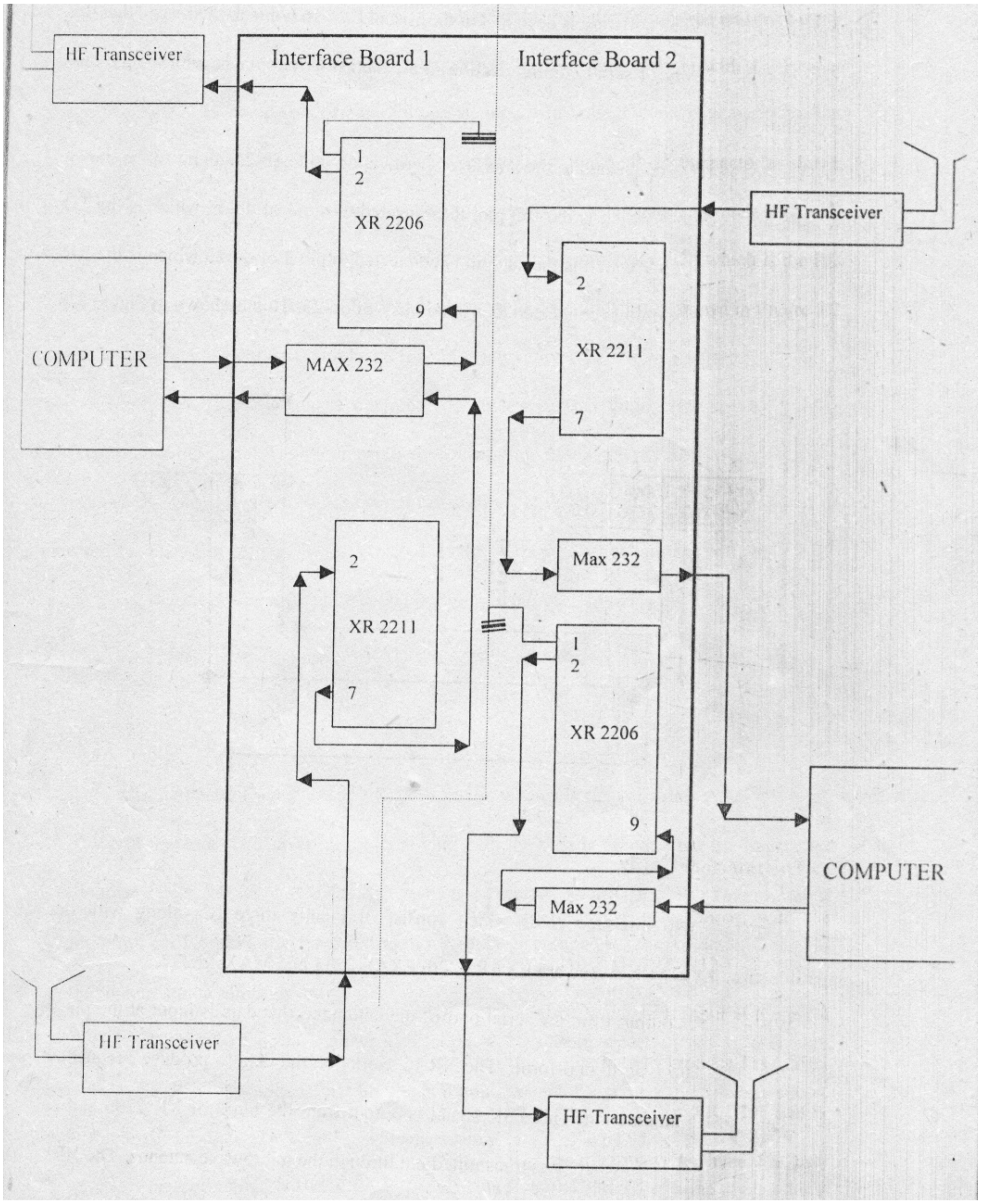
HF Transceiver
Interface Board 1
Interface Board 2
1
2
XR 2206
9
HF Transceiver
2
XR 2211
7
COMPUTER
MAX 232
2
XR 2211
7
Max 232
1
2
XR 2206
9
COMPUTER
Max 232
HF Transceiver
HF Transceiver

Figure:8-1

The computer is connected to the interface board via db-09 to db-25 connector as shown below. The pin 2 of db-9, which is the RD pin, is connected to pin 2 of db-25, which is the TD pin. Similarly pin 3 of db-9, which is the TD pin, is connected to pin 3 of db-25, which is the RD pin. The ground pin 5 of db-9 is connected to ground pin 7 of db-25. This is shown in Figure 8.2 below:

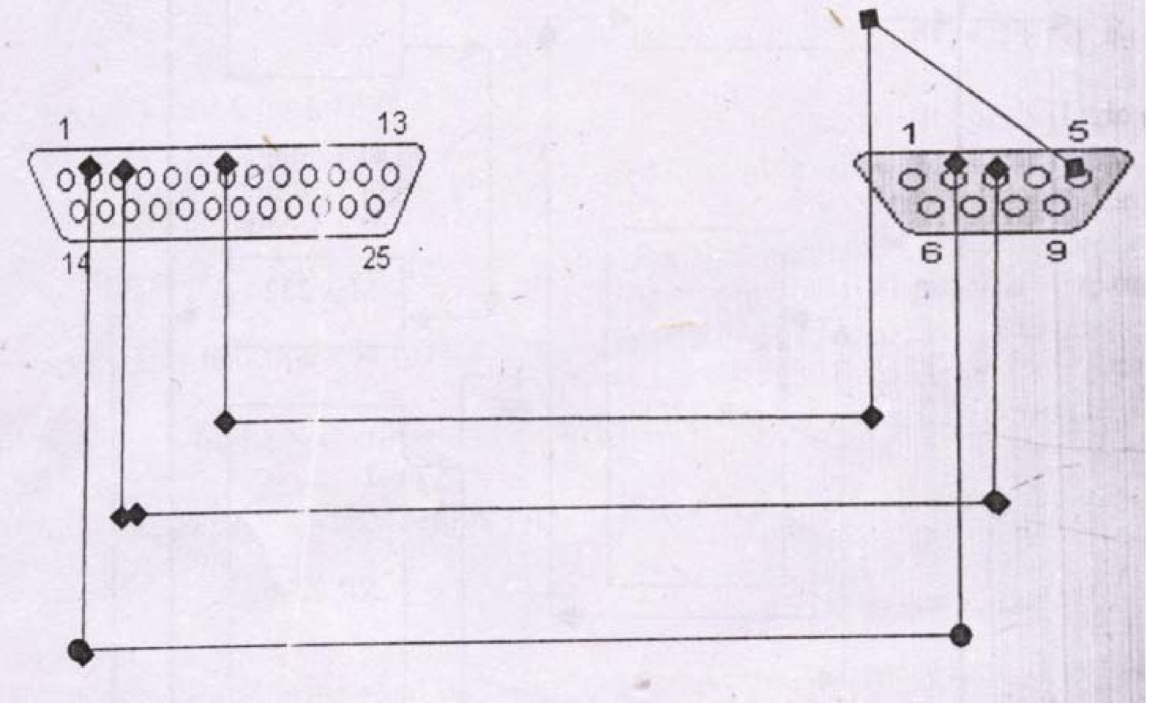

Figure8-2

8.I.2-The Interface Card:

There are two inteiface cards which consist of mainly three ICs along with the compatible circuitry. The ICs used are EXAR 2206, EXAR 2211 and MAXIM 232.

When data is output from the serial port of the computer, this data is input at the pin 9 of XR 2206. The data is in digital form.. The XR 1C works on the data to produce *Frequency Shift Keying* **(FSK)** *modulation*. This FSK signal is output from the pin 2 of XR 2206 and is sent to HF transceiver from where it is transmitted out through the microwave antenna. The XR 2206 is a monolithic function generator integrated circuit capable of producing high quality sine, square, triangle, ramp and pulse waveforms of high stability and accuracy. The output waveforms can be both amplitude and frequency modulated by an external voltage. Frequency of operation can be selected externally over a range of 0.01 Hz to more than 1 MHz.

The XR 2206 comprises of four functional blocks; a voltage-controlled oscillator, an analog multiplier and sine shaper; a unity gain buffer amplifier; and a set of current switches.

The VCO actually produces an output frequency proportional to an input current, which is produced by a resistor from the timing terminals to ground. The current switches route of one of the timing pins current to the VCO controlled by an FSK input pin, two discrete output frequencies can be independently produced for FSK Generation Applications.

For generating FSK, the XR can be operated with two separated timing resistors $\mathbf{R_1}$ and $\mathbf{R_2}$ connected to the timing pin 7 and 8 respectively. Depending on the polarity of the logic signal at pin 9, either one or the other of these timing resistors is activated. If pin 9 is open circuited or connected to bias voltage greater than 2, $\mathbf{R_1}$ is activated. If the voltage level at pin 9 is less than 1, only $\mathbf{R_2}$ is activated. The output frequency can be keyed between two levels, f_1 and f_2 as

$$\mathbf{f_1= 1/R_1C \quad and \quad f_2=l/R_2C}$$

The specification and the Data sheet of XR 2206 can be viewed in the Appendix A.

When the data is received at the receiving end, the data is output from the transceiver and sent to pin 2 of XR 2211 IC. The data, which it receives, is in FSK, The *Data* is output from the pin 7 of XR 2211 IC and is fed in to the receiver computer through Maxim 232. *The* outputted data is in the digital form.

The XR 22.J 1 is monolithic phase lock loop (PLL) system especially designed for data communications. It is well suited for FSK modem applications. It operates over a wide supply voltage range of 4.5 V to 20 V and a wide frequency range of 0.01 Hz to 300 KHz. It can accommodate analog signals between 2mV and 3V, and can be interfaced with conventional DTL ITL and ECL logic families. The circuit consists of a basic PLL for tracking an input signal within the pass band, a quadrature phase detector, which provides carrier detection, and a FSK voltage comparator, which provides FSK demodulation. External components are used to independently set the center frequency, bandwidth and output delay. An internal voltage reference proportional to the power supply provides ratio metric operation for low system performance variations with power supply changes.

The main PLL within the XR-2211 is constructed from an Input preamplifier, Analog Multiplier used as a phase detector, and a precision voltage controlled oscillator (VCO). The preamplifier is used as a limiter such that input signal above typically 2mV rms are amplified to a constant high level signal. The VCO is actually a current controlled oscillator with its nominal input current (f_0) set by a resistor (R_0) to ground and its driving current with a resistor (R_1) from the phase detector.

The other sections of the XR 2211 act to determine if the VCO is driven above or below the center frequency (FSK Comparator); produce both active high and active low outputs to indicate when the main PLL is in lock (quadrature phase detector and lock detector comparator). The specification and the Data Sheet of XR 2211 can be viewed in the Appendix A.

The MAXIM 232 IC is used as a voltage changer. It changes the voltage from 12 V to 5V and from 5 V to 12V. This 1C is placed in between the TTL devices and the serial port of the computer. At the serial port of the computer there is present 12 V. But the TTL devices work at 5V. So when the data is sent through the serial port to the XR 2206, Maxim 232 is used in between in order to change the voltage level from 12 volts to 5 volts. Similarly, when the computer receives data, it receives it through the XR 2211 IC via the MAXIM 232. The MAXIM 232 brings the voltage level up from 5 V to 12 V, which is suitable for serial port operation.

The specification and the Data sheet of MAXIM 232 can be viewed in the Appendix A: