SIMULATION OF FLIGHT FOLLOWING SYSTEM

BY

MATHEWOS ESSATU

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

FOR THE DEGREE OF MASTER OF SCIENCE

IN ELECTRONIC COMMUNICATION ENGINEERING

IN THE ADDIS ABABA UNIVERSITY

JUNE 1988/9

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

SIMULATION OF FLIGHT FOLLOWING SYSTEM

BY

Mathewos Essatu
Faculty of Technology

Approval by Board of Examiners:

Tesfaye Bayou Chairman, Department Graduate Committee	Signature
Dr. Kissig, Klaus Advisor	V. Kissig
HAILU AYELE Examiner	Hailu Ayele.
MOHAMMED ABDU Examiner	
GEBRE AMMANUEL GESSESSE Examiner	
PROF RAHFu ISOLA SALAUN External Examiner	

TABLE OF CONTENTS

PREFACE

The simulation of the flight following system is primarily chosen in the interest of Ethiopian Airlines need to implement the system to replace the currently used flight following system which is based on verbal communication between aircraft and ground control people.

The first section of this paper discusses the components of data communication system, aspects and objective of data communication system.

The second section laid the emphasis on the fundamental communication concepts parallel transmission, serial transmission and the different types of serial transmission. The theory behind the simulation is basically based on the different types of transmission modes. The third section build the foundation for the characteristics of data transmission, direction of information flow: simplex, half-duplex and full duplex and finally rate of transmission effect on data transmission. In section 4 the different coding technology and structures are presented.

A unique feature of the paper is section 5 : protocol. There are sections on identifying and defining a problem in data communication specially for good understanding between a sender and a receiver.

The principle of operation of the simulation of flight following system is discussed in section 6 and in section 7 conclusion and improvement using the same system for other forms of data are presented.

ACKNOWLEDGMENT

The production of paper of this type can be achieved only through the efforts and contribution on the part of numerous individuals. All the topics are covered from a practical rather than theoretical viewpoint; all the communication software for the system described having been personally designed and implemented.

I wish to express my deep gratitude to Ato Jatany Mudda, Director Manpower and Facility Planning in Ethiopian Airlines who contributed significantly by suggesting the project work and encouragement throughout the effort. A special thanks to my advisor Dr. Ing. Kissig whose advise, and enthusiasm made this undertaking successful. My partner Ato Anbessie Afework who deserves special help in advising and working together, so that the system described here and the system developed by him can interface.

I would like to express my gratitude to all my friends whose help and co-operation made this thesis possible. Last but not least the most important factor in the completion of this work has been the continuing support and encouragement of my colleagues in Ethiopian Airlines.

ror detection the Cyclic Redundancy Check, CRC-16, which results in sending a 16 bit data and 16 redundant bits is made to detect error. Totally for each aircraft identification and position information there are 16 bytes, of which the 8 bytes are Block Check Characters for the 4 pair of bytes, i.e, for frame identification and aircraft identification one pair of BCC byte and, for the position information 3 pairs of BCC bytes. Data representing position, i.e, latitude and longitude are represented by 3 bytes each.

According to the design of the system as well as the protocol, the program was run on IBM PC AT and the result has been displayed .

1.0 INTRODUCTION TO DATA COMMUNICATIONS

Communication plays a very important part in our lives, we are nearly always involved in some form of communication, be it written, oral, audio, or visual. Data communication is one specific area of this broad field.

A number of properties are common to all communication systems, the first of which is that their aim is to transfer information from one point to another. In data communication systems this information is generally called data or a message.

In order to have communication, four system components must be present:-

1) a sender, which generates,
2) a message and places it on
3) a transmission medium and, which carries the message to
4) a receiver.

Fig 1. shows the four systems components of any communication system.

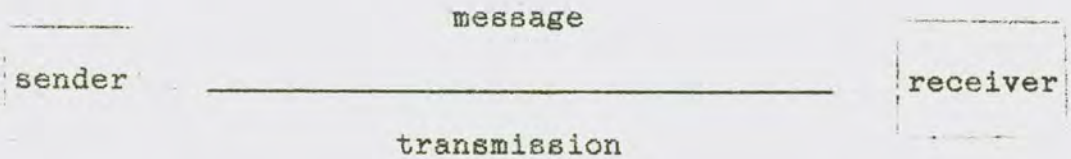


Fig 1 components of communication system

These components are the minimum requirements for any communication process; if any one of them is absent, communication cannot take place. However, each component of the four may vary, depending on the particular system involved.

In addition to having a common ultimate purpose and common basic structure of components, communications systems share three fundamental principles of performance:

First, the message must be understood by the receiver. A multitude of code, speed, and format variations are possible in representing data, just as there is a wide range of languages used in human-human communications.

Second, the characteristics of a communications system are defined and limited by the characteristics of its four basic components: sender, message, transmission medium , and receiver. The kind of message to be conveyed often dictates the characteristics that are most appropriate for a particular sender, transmission medium, or receiver in a communications system. This principle of performance is important to the design of effective communications systems. No single component can be designed or selected without regard for the way it interfaces with the other system components. The limitation of the components must be balanced in order for effective communication to occur. A communication system must be designed and evaluated as a system.

The third, general principle of communications performance is that interference can occur during the transmission process, corrupting the message. Such undesired disturbance in the system is noise, because it is interference that prevents the receiver from correctly interpreting the message that the sender meant to convey.

1.1 Aspects of data communications

Data communications is the function of transporting data from one point to another. Usually the data are encoded in someway. Timely transportation adds value to the information but does not change the inherent information content of the message transported. It is important to note that the transportation between endpoints must be successful in order to have value added by the communication.

The three fundamental aspects of data communications are:

1. Transmission - provides the path over which the information will pass.
2. Communications interface - transforms electrical signals used for transmitting information, to or from a form usable by the receiver or sender.
3. Communications processing - operates upon the information or upon the control characters that precedes, accompany, or follow the information to ensure its successful entry, transmission, and delivery.

Gaining an understanding of transmission involves study of electrical form that messages take during transportation and of the devices used to provide the transport path. Some of these devices comprise transmission media such as wires, microwaves, and satellite equipment; other transmission devices, such as modems, control the electrical form of messages.

Interface involves the transformation of data from a form understandable to the sender to the electrical signals to be transmitted. Similarly, communications interface converts those electrical signals into characters, words, numbers, lines, etc., that can be interpreted by receivers from a printed page or visual display or keyboard device.

Communications processing is perhaps the most complicated (and therefore confusing) aspects of data communications. Eight basic functions of communications processing can be classified conveniently into three areas of responsibility:

1. Editorial responsibility, which includes three basic communications processing functions:
   a) error control
   b) formatting
   c) editing

2. Conversion responsibility which includes two basic com-

munication processing functions:

a) speed conversion

b) code conversion

3. Aribtrator responsibility which includes three basic communications processing functions:

   a) network control

   b) polling

   c) message routing

Communication processing (ensuring successful entry, transmission, and delivery of the information) can be performed in any of a number of devices, using the storage and processing capability of those devices.

The communications processing are capable of front-end processors (FEPs), concentrators, message switches, terminal control units, terminals, computers and modems.

1.2 Objectives of Data Communications System

In general, data communication network are established to collect data from remote points (served by terminals) and to transmit those data to a central point equipped with a computer or another terminal, or to perform the reverse process, or some combination of the two. Data communication

network thus provide access to remotely located computers; they improve the day-to-day controls of a business by providing faster information flow; they provide message switching services to allow terminals to talk to one another. In general, data communications systems offer more reliable and more timely interchange of data among their users and bring the power of computers closer to more users.

The general objectives of most data communication network include-

-Reducing the time and effort required to perform various business tasks
-Capturing business data at their sources.
-Centralizing control over business data.
-Effecting rapid dissemination of information.
-Reducing current and future costs of doing business.
-Supporting expansion of business capacity at reasonable incremental cost as the organizing grows.
-Supporting organizational objectives in centralizing or decentralizing computer systems.
-Supporting improved management control of the organization.

Meeting these objectives gives the user environment the potential to achieve improved resource utilization, improved flexibility to expand or change to meet business requirements, and increased control over support of business opera-

tions. Basically, as users implement and operate their data communications systems, they are attempting to improve their business operations by providing information (both internally and externally) on a more timely basis.

2.0 Fundamental Communication Concepts

Systems that transmit data must have consistent methods of transmission over communication channels. All the systems that are discussed in here transmit binary data, or data forms that are intrinsically binary. Basically, binary data can be sent over communication lines in either serial or parallel mode. The internal transfer of data within modern computers is done in parallel mode. In other words, if the internal structure of the computer uses an 8 bit code element, then all 8 bits of an element are transferred between the main memory and any operational register in the same computer cycle.

2.1 Parallel Transmission

Parallel transmission is a method of transfer in which all the bits of a character are sent simultaneously either over separate lines or on different frequencies of the same line. Parallel transmission uses a low cost transmitter, but the receiver is a high cost item, and it generally also requires higher cost transmission lines than does serial transmis-

sion. It is not used on low speed lines because its primary purpose is to speed up the transmission between two points. Neither is it used on long distances lines because the bits drift back and forth in time relation to one another and may interfere with the bits of the preceding or following character. With parallel transmission, the line speed in bits, is really the line speed in characters since all the bits of a character are sent simultaneously.

Parallel transmission although seldom used outside the computer, is shown in fig 2.1. This figure shows how all 8 bits of the code travel down a channel simultaneously, followed a short time later by eight more bits.

fig 2.1 parallel transmission of an 8 bit code.

This type of transfer is not normally used in data communications. In data communications, transfer of information is usually done in serial mode. Serial transmission implies that a stream of data is sent over a communication line bit by bit.

2.2 Serial Transmission

The distinguishing difference between serial and parallel transmission is that, in serial transmission, the transmitting device sends a bit followed by a time interval, then a second bit, and so on until all the bits are transmitted. It takes n time cycles to transmit n bits. In parallel transmission, n bits are sent, followed by a time interval, then n bits are sent and so on. In parallel transmission, then, n bits are sent in one time cycle, whereas in serial transmission the same bits take n time cycles.

Serial transmission of an 8 bit code is shown in fig 2.2

Fig 2.2 Serial Transmission of an 8 bit code

Most data communication is performed by serial transmission. Three transmission modes are in common usage: Asynchronous, Synchronous, and Isochronous transmission. Note that all three of these modes of transmission are serial.

2.2.1 Asynchronous Transmission

The first mode of transmission, asynchronous, is often referred to as start-stop transmission. This is because the transmitting device can transmit a character at any time that is convenient and the receiving device will accept that character. Characters can be sent at irregular intervals: for example, 1 character per second or 1 character and then a 10 second wait. To enable the receiver to recognize a character when it arrives, each of the characters that are transmitted has a start bit preceeding, and one or two stop bits following the data signal bits. In fig 2.3, shown below, the data signal bits are the 8 bits code and as can be seen, this code has one start bit and two stop bits.

In summary , a start bit is a signal that is used to inform the receiving terminal to start sampling the incoming data signal at a fixed rate so that it can be interpreted into its proper character structure. A stop bit, which follows the data signal bits, informs the receiving terminal that a character has been received and resets the terminal for

recognition of the next start bit. Synchronization of terminals is re-established upon the reception of each character. Fig 2.3 illustrates asynchronous transmission.

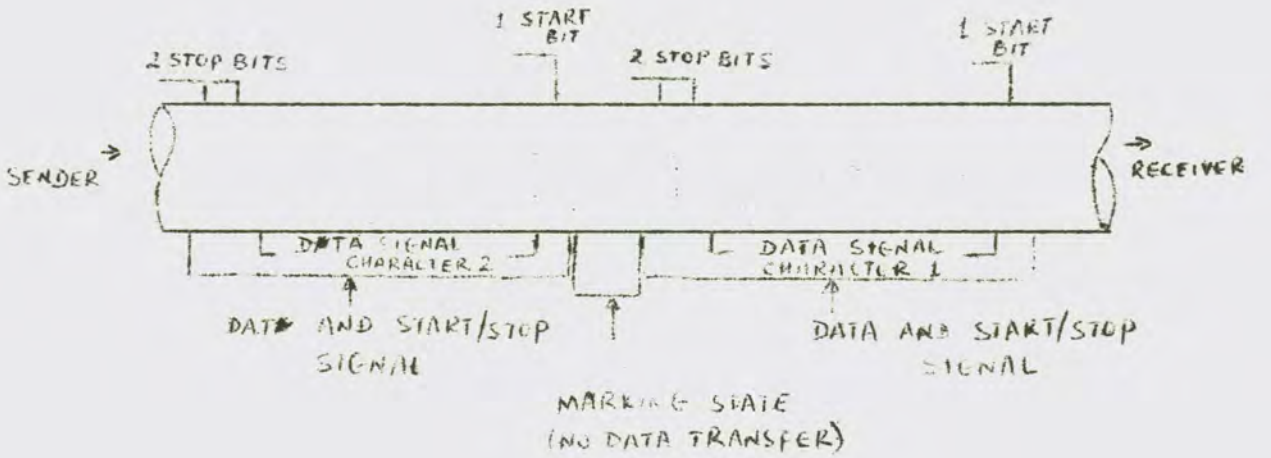


Fig 2.3 Asynchronous Transmission of an 8 bit code.

2.2.2 Synchronous Transmission

The second transmission mode, synchronous, is used for the high speed transmission of a block of characters. In this mode of transmission, both the sending device and the receiving device are operated simultaneously and are resynchronized after each few thousand data signal bits are transmitted. Start/stop bits are not required for each character.

Synchronization is established and maintained either when the line is idle (no data signals being transmitted) or just prior to the transmission of a data signal. This synchronization is established by passing a predetermined group of "sync" characters between the sending and the receiving devices. Fig 2.4 shows how the data signals are contiguous and how one long stream of data bits is transmitted from the sending to the receiving device. In other words, the sending device will send some "sync" characters to the receiving device so that the receiving device can determine the time frame between each of the bits.

The sending device sends a long stream of data bits that may have thousands of bits. The receiving device, knowing what code is being used, counts off the appropriate number of

bits and assumes this is the first character and passes it to the computer. It then counts off the second character and so on.

Synchronous transmission is more efficient in that there are fewer control bits in proportion to the total number of bits transmitted. The synchronization may take only 16 to 32 bits, while the stream of bits in the data signal, may be several thousands bits long.

In asynchronous transmission there is at least one start bit and one stop bit for every character of data. If an error occurs during asynchronous transmission, that error may only destroy one character of data because each character is synchronized with its own start and stop bits. On the other hand, a similar error in synchronous transmission would probably destroy the entire message block by breaking synchronization.

The modems and related equipment for synchronous transmission are more expensive than those used for asynchronous transmission because they must be able to synchronize between themselves.

In summary, synchronous and asynchronous data transmission modes are differentiated by the fact that in asynchronous data transmission, each character is transmitted as a to-

tally independent entity with its own start and stop bits to inform the receiving device that the character is beginning and ending. In synchronous transmission, on the other hand blocks of data are transmitted as units after the transmitter and the receiver have been synchronized.

2.2.3 Isochronous Transmission

Isochronous transmission is a third technique; it combines the element of both synchronous and asynchronous data transmission.

In isochronous transmission, as in asynchronous, each character is required to have both a start bit and a stop bit. However, as in synchronous data transmission, the transmitter and the receiver are synchronized. The synchronization time interval between successive bits is specified to be an integral multiple of the length of one bit. That is, all periods of no transmission consists of one or more one character time intervals. This common timing allows higher precision between the transmitting and the receiving equipment that could be achieved using asynchronous techniques only.

Fig 2.4 illustrates the relationships and differences between asynchronous, synchronous, and isochronous transmission. In asynchronous transmission, there is no determination of the spacing between individual characters

(indefinite time). This requires that both the sending and receiving equipment have clocks to determine the length of a bit, and the receiver must have special recognition (circuitry) to determine the beginning and end of a character. With synchronous transmission, the clocking signal is used to synchronize the receiver to the sender before a long, multicharacter block of data is transmitted. In isochronous transmission, the clocking is supplied by the sending modem, and the receiving modem synchronizes to it for short periods. Each character begins on some multiple of the length of the bit element.

The primary reason for using isochronous transmission in preference to asynchronous transmission is speed. In practice, asynchronous data transmission is generally limited to 1,800 bits/sec by the timing precision of the sending and receiving equipment. By contrast, isochronous data transmission can achieve data transmission rates as high as 9600 bits/sec. Synchronous data communications may be even more rapid than isochronous.

fig 2.4 Comparison of Transmission Modes.

3.0 Characteristics of Data Transmission

A transmission facility or channel is an electrical or electromechanical device that is used to get information from one point to another. The channel may be provided by a pair of wires, a portion of the radio frequency spectrum; or some other medium. Its only purpose is to carry information from one location to another. It is also called a data link.

A data link can be described by its characteristics in the following categories:

-Direction of information flow
-Mode of signal representation
-Rate of transmission
-General accessibility
-Medium
-Error detection

The electrical and physical characteristics of any data link limit its information handling abilities. The selection and design of the transmission paths most appropriate for a particular application depend on appropriate balancing of the inherent advantages and disadvantages of options in each of the categories listed and across the categories, achieving a suitable mix of cost and performance capabilities.

3.1 Direction of Information Flow

The direction(s) in which information can flow over a transmission path is determined by the properties of both the data link and the terminal devices.

3.1.1 Simplex

Simplex - operation allows information to flow in only one direction. Simplex operation is usually determined by the characteristics of a terminal rather than by data link characteristics. That is, use of a send-only terminal device (say a data collection terminal on a factory floor) or a receive-only terminal device (say a line printer) means that the associated data link must operate in simplex mode.

3.1.2 Half-Duplex

Half-duplex - operation allows information to flow in either direction, but not in both directions simultaneously. This mode of operation is exhibited by polite conversation and one-way bridges. The action of reversing the direction of flow in a half-duplex path takes a finite amount of time, which includes the time it takes to recognize the end of transmission in one direction and the time required to flip transmit/receive switches to enable transmission in the opposite direction. This turnaround time is considered to be

"overhead time" and can significantly affect throughput characteristics of the path. Turnaround time is commonly on the order of 20 to 200 millisecond. Turnaround time is one parameter used in the evaluation of transmission facilities. Half-duplex operation is determined both by terminal characteristics (nonoverlapped send and receive) and by data link characteristics (the path may be used in either direction but not in both directions at the same time).

3.1.3 Full Duplex

Full-duplex - operation allows information to flow simultaneously in both directions on the transmission path. This mode of operation is also referred to as full-duplex or FDX or simply duplex. A familiar example of full-duplex operation is a two-way street. Use of full-duplex operation potentially improves the efficiency of a transmission system because it eliminates the line turnaround time required in a half-duplex arrangement.

Full-duplex operation requires both that the terminal equipment be capable of sending and receiving simultaneously and that the data link provide two independent transmission paths, one in each direction. Many data processing/data communications systems currently do not take advantage of the availability of full-duplex facilities. Consequently, where full-duplex transmission is used, it is often with data

travelling in one direction only and with the other direction of transmission used for control signals. Full-duplex operation is becoming more prevalent, however, with increasing computer-computer communications and the availability of line control protocols(e.g. SDLC, HDLC, ADCCF) that encourage full-duplex operation.

3.2 Rate of Transmission

The signalling speed of a transmission path is measured in bauds, defined as the number of times per second that the signal state (e.g. phase) changes. In contrast, the speed of information transmission is measured in bits/sec, with the basic modulation techniques the signalling rate was always equal to the bit transfer rate.

Communications paths can be grouped by their speeds of information transfer into three basic categories; narrowband, voice grade, and wideband.

Narrowband or sub-voice-grade channels range in speed from 45 to 150 bits per second and are used primarily for telegraph and some low-speed data terminals.

Voice-grade channels(such as those provided by the public-switched network) provide a capacity of approximately 3000 HZ and, when unconditioned, are limited to a maximum of 1800

4.0 Coding Terminology and Structure

A character is a symbol that has a common, constant meaning for some group of people. A character might be the letter A or B, or it might be a number such as 1 or 2. Characters may also be special symbols such as & or ?. Characters in data communications, as in computer systems, are represented by groups of bits. The various groups of bits that represent the set of characters that are the "alphabet" of any given system are called a "coding system," or simply a "code."

A byte is a group of consecutive bits that are treated as a unit or character. One byte is normally comprised of 8 bits and it usually represents one character. However, in data communications some codes in regular use utilize 5, 6, 7, 8, 10 or 11 bits per character. These differences in the number of bits per character arise because the codes have different numbers of characters to represent and different provisions for error checking.

Coding is the representation of one set of symbols by another set of symbols. Information in data communication is normally transmitted serially over a transmission line or channel. Codes for representing the information vary in relation to both the number of bits used to define a single character and in the assignment of bit patterns to each par-

ticular character. In most cases, both the sending and receiving equipment must be designed or programmed to transmit and receive the code that the data communication system is using. It is this equipment that determines whether one or two stop bits be utilized during asynchronous transmission.

The various common codes of data communication are-

ASCII - 8 bit code with 128 valid characters ($2^7 = 128$ characters; the eighth bit is a parity check.). ASCII code is the basic standard code for which most communication equipment are designed.

BAUDOT code - 5 bit code. It is used mainly by older Teletype equipment. It has 62 valid characters. When used in asynchronous transmission, it has 5 bits per character (no parity bit), one start bit and a long stop bit equal to 1.42 bit times. This means it uses 7.42 bit times for one character during transmission from a sending device to a receiving device.

There is also a closely related code to Baudot, called Pseudo-Baudot. Pseudo-Baudot is identical to Baudot code except it uses a stop bit equal to 1.5 bit times, and therefore, it occupies 7.50 bit times during transmission.

Data interchange code is another code that is used on newer teletype equipment. The teletype version of the Data interchange code is an 8 bit code that uses 7 bits to represent the characters and one bit for parity. Data interchange code has 128 valid character combinations (2^7 = 128 characters). Data interchange code is 11 bits per character in transmission with 1 start bit, 7 data bits, 1 parity bit, and 2 stop bits.

The 4-of-8 code is an IBM code. It uses only 4 of the 8 bits of data because the only valid combinations that are recognized are those in which exactly 4 of the 8 bits are in a "1" configuration and the other four are in a "0" configuration. The purpose in allowing only configurations that have four 1's and four 0's is that an accuracy check can be accomplished by ensuring that there are always four 1's and four 0's in the received data character. The price paid for doing this is that there are only 70 valid characters, instead of 256 (2^8 = 256 characters). The 4-of-8 code is 10 bits per character in asynchronous transmission, with 1 start bit, 8 data bits, and 1 stop bits. This code detects errors better than the single parity bit of ASCII or than other codes that use a single parity bit. The 4-of-8 code is used primarily on high speed voice grade lines.

BCD (the Binary Coded Decimal) code - is an extension of older tab-card-oriented Hollerith code. It is a 6 bit code and has 64 valid character combinations (2^6 = 64 characters). BCD code is 9 bits per character in asynchronous transmission with 1 start bit, 6 data bits, 1 parity bit, and 1 stop bit. This code is used primarily on low-speed lines. ¶2ê

The Extended Binary Coded Decimal Interchange Code (EBCDIC) is an IBM system 360/370 code. EBCDIC has 256 valid character combinations (2^8 = 256 characters). This code has 11 bits per character in asynchronous transmission utilizing 1 start bit, 8 data bits 1 parity bit and 1 stop bit.

The codes described above are the basic ones utilized in data transmission. These codes can be used in asynchronous, synchronous, or isochronous transmission. Fig 4.1 summarizes the various start bit, data bits, parity bits, stop bits, and the total bits in asynchronous transmission for the seven different code configurations that have been discussed.

fig 4.1 code configuration for asynchronous transmission.

When designing an information system, it is necessary to consider the different forms and encodings of information that may be needed.

5.0 PROTOCOLS

5.1 Introduction

In data communications, a protocol is a set of rules and procedures established to control transmission between points.

There are two major reasons why protocols are needed in communications networks. First, efficiency can be achieved by reducing the volume of transmission; a protocol reduces the amount of necessary control information by using specified code sequences to identify certain conditions. Second, it is necessary to be able to distinguish between control bits and data bits; a protocol establishes a set of rules so that the receiver can properly interpret the bit stream transmitted from the sender. Without protocols to guide the orderly exchange of data between points in a network, there would be chaos. The protocol determines who talks next in the conversation, how error control is handled, and how the text of messages is to be delineated by control characters.

One responsibility of a protocol is message formatting. The protocol might be used to break lengthy messages into fixed size blocks for transmission. Control characters such as SOH

(start of header), STX (start of text), ETB (end text block) and ETX (end of text) are inserted according to protocol in order to delineate which blocks belong to which message.

There are four types of communication protocols that need to be controlled in a data network.

SESSION - from the start of a "conversation" to the end of that conversation.

PATH - from endpoint to endpoint in a network. (e.g.from application program in a host computer to user at a terminal device).

LINK - between two devices connected by a single transmission line or channel.

HARDWARE - between a device and a line.

These types of communication have different characteristics. A comprehensive data communications protocols include rules that address each type.

Session control establishes and terminates communication and performs accounting procedures. Path control ensures that messages get from one endpoint to another, correctly, and that data flow smoothly through the network without flooding components of the network. Link control ensures that a block of data gets from one end of a data link to the other, correctly. Hardware control ensures that bits can be put onto a line and taken off at the other end and includes both physical and electrical interface rules.

Many different types of protocol are in use today. The most interesting and widely used is the family of link control protocol. The functions of a link control protocol are diverse, including the following.

- Detection and correction of errors
- formatting of blocks
- resolution of contention
- polling
- control of multipoint lines
- sequencing of messages
- interpretation of control characters
- identification procedures
- handling of abnormal circumstances

These responsibilities can be generalized as aspects of communications processing.

A protocol is basically a set of rules for operating a communication system. The rules are designed to solve operating problems in the following areas:

1. Framing - the determination of which eight bit groups constitute characters, and most important, what groups of characters constitute messages.

2. Error control - the detection of errors by means of the longitudinal, vertical, or cyclic redundancy checks; the acceptance of correct messages, and the request for retransmission of faulty messages.

3. Sequence control - the numbering of messages to eliminate duplicate messages, avoid losing messages, and properly identify messages that are retransmitted by the error control system.

4. Transparency - the transmittal of information (such as instrumentation data) that contains bit patterns which resemble the control characters used to implement functions 1, 2, 3 above, without the receiving station identifying those bit patterns as control characters.

5. Line control - the determination, in the case of half-duplex or multipoint line, of which station is going to transmit and which station(s) will receive.

6. Special cases - solving the problem of what a transmitter sends when it has no data to send.

7. Time out control - solving the problem of what to do if message flow suddenly ceases entirely.

8. Startup control - the process of getting transmission started in a communication system that has been idle.

Protocols may be divided into three categories according to the message framing techniques used. These are Character Oriented, Byte Oriented and Bit Oriented.

A Character Oriented protocol uses special characters, such as STX to indicate the beginning of a message, and ETB to indicate the end of a block of text (i.e, the imminent arrival of the block check characters). The classic character oriented protocol is IBM's Binary Synchronous Protocol, known as BISYNC.

Byte count oriented protocols use a header which includes a beginning special character followed by a count that indicates how many characters follow in the data portion of the message and some control information such as which messages have been received correctly to date. The data portion which comes next is the specified length and is followed by block check characters. Digital Equipment Corporation's Digital Data Communication Message Protocol (DDCMP) is an example of this type of protocol.

It is possible to create a protocol that delineates which bits constitute messages by separating those messages with a special flag character such as 01111110. This type of protocol specifies that there shall never be six "1" bits in a row except for the transmission of a flag. Thus, when the receiving station receives a flag character, it knows that the previous 16 bits were the block check characters and that the bit between those 16 and the previous flag constitute the message. This type of protocol is called a

"bit-stuffing" or bit oriented protocol. IBM Synchronous Data Link Control (SDLC), American Standards Institute ADCCP, International Standards Organization HDLC, and CCITT recommendation X.25 are some of the examples for bit oriented type of protocol.

5.2 ERROR DETECTION

In all electrical information transmission systems, due consideration must be given to the effects of noise. Noise is any unwanted signal and may originate from sources as spectacular as lightning strikes or as dirty contacts on telephone switching equipment.

The most important characteristic of noise in telecommunication system is the relatively long duration of the disturbances. A noise burst of 0.01 second duration is not uncommon and sounds like a simple click during a voice conversation. If the 0.01 second noise burst occurs during a 4800 bits per second data transmission, the "simple click" is the death knell for 48 data bits. Thus, when noise causes bits to be received in error, it generally causes a great number of bits to be affected. The periods of high error rate are generally separated by relatively long intervals of low noise, low error rate data reception. Thus the error rate averaged over an hour is typically one error bit in 100,000 bits received.

To determine whether the bits in a character have been properly received, it would be quite simple to append an additional bit to each character and to have that bit be a one or a zero according to the rule that "all transmitted characters shall have an odd number of ones." Thus for example, the character 01001100 would be expanded to 0̲01001100 and the character 01101100 would be expanded to 1̲01101100. The added, underlined bit is called the "parity" bit and using it to make the number of ones odd is called "odd parity." (plainly, one could make the number of ones even and call it "even parity.") In a parity system, the transmitter unit calculates the state of the parity bit and appends it to the character during transmission. The receiving unit calculates the state of the parity bit and compares the calculated value to the value actually received. If they disagree, the receiver knows that a bit has been received in error.

Let us assume that the transmitter sends 101101100. Parity is odd; everything is ok. Let us assume that the second and third bits (counting from the right) are received erroneously. The received character is then 101101010. The parity is still odd and things appear ok despite the double error. In here the parity on each character can detect only errors that affect a single bit (or three, or five, etc.).

Errors that affects two (or four, or six, etc.) bits will not be detected. this problem exists regardless of whether "odd parity" or "even parity" is used.

Consider transmission of the six characters in fig 5.1 . Each bit (except the left-most) in the check character has been computed such that that bit and the bits immediately above it in character 1 through 5 total an odd number of ones. For example, the right-most bit in the check character is a zero because three of the right-most bits in the characters above it are ones; hence there is an odd number of ones.

101101100	character 1
110101111	character 2
001110101	character 3
111100010	character 4
100010111	character 5
010111100	check character

Fig 5.1 Sample Transmission

in all characters, including the check character, the left-most bit is a parity bit.

Let us assume that the transmitter sends 101101100 (character 1). Parity is odd; everything is ok. Let us assume that the second and third bits (counting from the right) are received erroneously. The received character is then 101101010. The parity is still odd and things appear ok despite the double error. This time, however, there is a check character being sent. The error just described increases the number of ones in the second column from three to four (an even number) and decreases the number of ones in the third column (counting from the right) from four to three (excluding the check character).

When the check character sent by the transmitter is compared in the receiver to that which the receiver has calculated from summing the ones in the various "columns" of the received character, the second and third bit positions will be incorrect. The computed check character will require the second bit from the right to be a "1" to increase the number of ones in that column from four to five, and will require the third bit to be a "0" as the receiver has received only three one bits in that column. Thus the calculated check character and the transmitted check character will disagree in the second and third bit positions, which are indeed where the error occurred.

Lest one embrace this scheme as foolproof, consider the case where character 1 and character 3 are received with errors in the second and third bit positions. Now there will be two more ones in the second column and two fewer ones in the third column. The number of ones in each column is now such that the receiver's calculated block check character (BCC) and that sent by the transmitter will be the same.

Thus in the same way that a parity check within the character was defeated by a double error in the character, parity on the columns (referred to as a Longitudinal Redundancy Check or LRC) is defeated if a double error occurs in a column. There are numerous possibilities for double bit errors in characters to occur simultaneously with double bit errors in columns in such a fashion that neither the character parity (referred to as Vertical Redundancy Check or VRC) nor the column parity (LRC) will indicate that errors have occurred. This is an essential problem since errors in communications transmission systems tend to occur in bursts.

The detection systems most effective in detecting errors in communications systems with a minimal amount of hardware (but more than VRC/LRC systems) are the Cyclic Redundancy Checks (CRC).

5.2.1 Cyclic Redundancy Checks (CRC)

CRC calculations are customarily done in a multi-section shift register which feeds into an exclusive-OR gate whose output feeds back to other exclusive-OR gates located in between the sections of the shift registers. An exclusive-OR (XOR) gate is a gate where the output is a "0" if the inputs are both "0" or are both "1". If the inputs differ the output of the XOR gate is a "1". Fig 5.2 shows a typical arrangement of BCC for CRC.

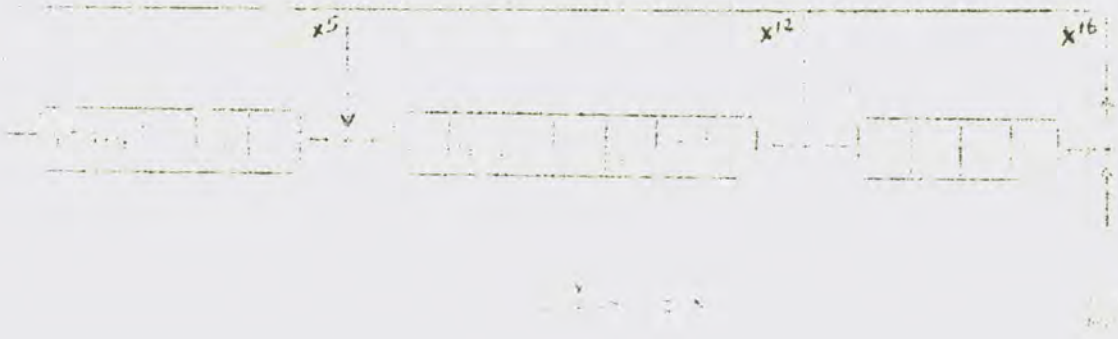


Fig 5.2 Block Check Register Implemented with Shift registers and XOR.

The placement and quantity of the XOR gates vary for CRC-12, CRC-16, and CRC-CCITT, which are the most common Cyclic Redundancy Checks. The block check register example shown in fig 5.2 is the implementation of CRC-CCITT.

When the logic shown in fig 5.2 is used in a transmitter circuit, it is intialized at all zeros. As each bit is presented to the communications line it is also applied to the point marked "A" in fig 5.2 and a shifting pulse is applied to the shift register. Fig 5.3 shows the contents of the shift register assuming that the first bit transmitted was a "1". Note what a dramatic effect just this single bit has and how that effect is spread throughout the register.

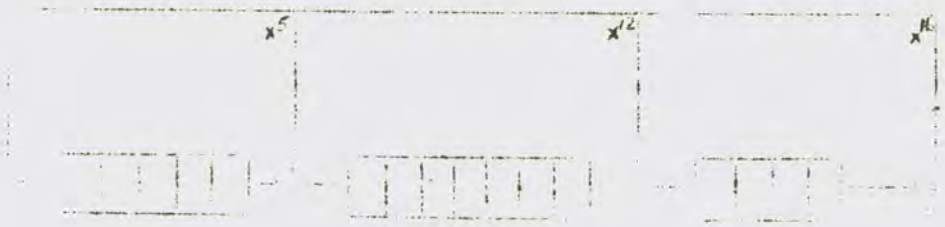


Fig 5.3 Propagation of a "1" bit into Block Check Register

As the "1" bit, shown in fig 5.3, gets shifted on through the segments of the shift register during the transmission of subsequent bits, they will eventually reach some of the XOR gates between the segments of the shift register. Here they will affect the state of the "feedback bits" coming from the XOR gate on the far right-hand side of the diagram.

The general effect to be recognized is that the effect of any bit is reflected in the various bits of the shift register for a considerable time after that bit is transmitted.

In a CRC equipped system, a logical arrangement identical to that used in the transmitter (fig 5) is also used in the receiver. Again the register is initialized to zero and data from the communications line is applied to point A while the shift register contents are shifted once for each bit received. At the conclusion of the message, the transmitting station sends the contents of the transmitter CRC shift register to the receiving station. The X^{15} term, marked "LSB" is sent first. The receiving station applies the incoming bit stream to point A of its CRC logic, just as it did with the preceding data bits.

As the first bit of the received CRC character is applied to point A of the receiver CRC logic, the XOR gate obeys B + B = 0 for B = 0 or B = 1 and produces a "0" if the first bit of the calculated CRC matches the first bit of the CRC being received.

All of the other XOR gates in fig 5 style of CRC logic will obey B + 0 = B for B = "0" or B = "1" and will become essentially "transparent"; a "0" will be shifted into the left end of the shift register. As long as the calculated CRC contained in the receiver CRC shift register continues to match the CRC being received from the transmitting station; the output of the right-most XOR gate will continue to be zero, and the other XOR gates will continue to be "transparent." When all the CRC bits have been shifted, the CRC shift register bits will all be zeros. This will be true regardless of the placement of the XOR gates hence true for all types of CRC. An exception is that SDLC and HDLC start with a preset value in the shift register and end with a special non-zero result.

The most important property of CRCs is that, due to the feedback arrangements, the exact state of the shift register is dependent on a great deal of past history. Hence it is highly unlikely that a burst of errors could produce a CRC calculation that was the same as that for the data as originally transmitted before the errors occurred.

CRC requires no bit per character like parity does, but there are two check characters at the end of each block of characters, since the CRC shift register is two character lengths long. These two characters are referred to as the Block Check Characters (BCC).

5.2.2 Mathematical Presentation of CRC

A cyclic code message consists of a specific number of data bits and a BCC. Let n equal the total number of bits in the message and k equals the number of data bits; then n - k equals the number of bits in the BCC.

The code message is derived from two polynomials which are algebraic representations of two binary words, the generator polynomial P(x) and the message polynomial G(x). The generator polynomial is the type of code used (CRC-12, CRC-16 and CRC-CCITT); the message polynomial is the string of serial data bits. The polynomials are usually represented algebraically by a string of terms in powers of x such as $x^n+...+x^3+x^2+x^1+x^0$. In binary form, a 1 is placed in each position that contains a term; absence of a term is indicated by a 0. The convention followed in the following presentation is to place the x^0 bit at the right. For example, if a polynomial is given as x^4+x+1, binary representation is 10011.

Given a message polynomial G(x) and a generator polynomial P(x), to construct a code message polynomial F(x) that is evenly divisible by P(x) :

1. multiply the message G(x) by x^{n-k} where n-k is the number of bits in the BCC.
2. divide the resulting product $x^{n-k}(G(x))$ by the generator polynomial P(x).
3. disregard the quotient and add the remainder C(x) to the product to yield the code message polynomial F(x), which is represented as $x^{n-k}(G(x)) + C(x)$.

The division is performed in binary without carries or borrows. In this case the remainder is always one bit less than the divisor. The remainder is the BCC and the divisor is the generator polynomial; therefore, the bit length of the BCC is always one less than the number of bits in the generator polynomial.

For example,

1. given

   message polynomial G(x) = 110011 (x^5+x^4+x+1)

   generator polynomial P(x) = 11001 (x^4+x^3+1)

   G(x) contains 6 data bits

   P(x) contains 5 bits and will yield a BCC with 4 bits; therefore n-k = 4.

2. multiplying G(x) by x^{n-k}

$$x^{n-k}(G(x)) = x^4(x^5+x^4+x+1) = x^9+x^8+x^5+x^4$$

$$= 1100110000.$$

3. this product is divided by P(x)

```
1100110000  11001
11001       100001
     10000
     11001
      1001  <--  remainder = C(x) = BCC
```

4. the remainder C(x) is added to $x^{n-k}(G(x))$ to give

$$F(x) = 1100111001.$$

The code message polynomial is transmitted. The receiving station divides it by the same generator polynomial. If there is no error, the division will produce no remainder and it is assumed that the message is correct. A remainder indicates an error. The division is shown below .

```
1100111001  11001
11001       100001
     11001
     11001
     00000 <--  no remainder.
```

In typical data communication hardware, the BCC is computed and accumulated in a shift register. The configuration of the register is based on the CRC code to be implemented. The number of stages in the register is equal to the degree of the generating polynomial; the number of XOR elements is also a function of the polynomial.

CRC-12 is applied to synchronous system that use 6 bit characters. The BCC accumulation is 12 bits. The generator polynomial is $x^{12}+x^{11}+x^3+x^2+1$ with prime factors of $(x+1)$ and $(x^{11}+x^2+1)$. It provides error detection of bursts up to 12 bits in length.

CRC-CCITT is the standard used to compute a BCC for European systems. When operating with eight bit characters, the BCC accumulation is 16 bits. The generator polynomial is $x^{16}+x^{12}+x^5+1$. It provides error detection of bursts up to 16 bits in length.

CRC-16 is applied to synchronous systems that use eight bit characters. The BCC accumulation is 16 bits. The generator polynomial is $x^{16}+x^{15}+x^2+1$. It provides error detection of bursts up to 16 bit in length.

6.0 Principle of Operation of the Simulation of Flight Following System

Flight Following System is one part of all airline activities. In this system aircraft positions and conditions are monitored instantaneously from the start of engine, take-off, cruise, descent and landing. At each phase of the flight the aircraft will send message to ground station, the position and if required additional information about the flight within a specified interval of time. The ground station will receive the message and advise sections requiring the information for better planning and efficient operation. If the aircraft didn't send a message within the specified interval of time, the ground station will try to contact him and ask him to send the information. If the aircraft again didn't respond, the ground station will contact the people at the aircraft's last reporting point and tell them to search the aircraft. If these people don't get response from the aircraft, they will launch search and rescue operation.

Currently in Ethiopian Airlines the system uses verbal communication, i.e, HF transmission for longer distance from the base station and VHF transmission for shorter distances. Using HF and VHF communication system, sometimes there is a possibility where the aircraft sends message and the ground station couldn't receive the message. This is due to noise or too much interference.

When data are transmitted using microwave transmission or satellite transmission, the possibility of losing or distortion of data is significantly less than either the VHF or HF transmission.

There is a system known as INMARSAT (International Martime Satellite Organization) where all aircrafts will have a link to any station or aircraft via satellite and when this system is implemented the data to the ground station from the aircraft via satellite will make use of the system described and simulated here.

The whole communication between the aircraft and the ground station and then after, to the different concerned sections can be automated, that is to say, without human interference. Data can be automatically sent to the ground station repeatedly within a specified interval of time and the same data at the ground station can be distributed to the concerned sections as well as be displayed on a computer screen. This method of communication, when implemented, will improve information availability and distribution of it. Because filing of information could also be made for an agreed period of time, so that retrieval of the information at a latter date will help in reconstructing operation scenarios.

In this project the above system is simulated using one computer as an aircraft and another computer as a ground station.

The computer used as an aircraft will send serially through wire position data and aircraft identification to another computer which simulates the ground station. The ground station computer will receive this data and after processing, will indicate the position of the aircraft on the screen where the background is the world map. The position of different aircrafts can be simulated by sending different identification for the different aircrafts and displaying a unique character for each aircraft. In doing so the relative position of the aircrafts and estimated time of arrival can be monitored on one screen.

6.1 Methodology

As it has been indicated in the previous sections there could be a number of ways by which data can be transmitted from the sender to the receiver. Each and every system has its own particular characteristics and depending upon the characteristics one may choose, from the different possibilities, the one which fits the system specific requirement.

In the flight following simulation system the type of transmission chosen is the serial synchronous data communication, which is mostly used in most of todays data communication systems, at a transmission rate of 100 bauds.

Data containing aircraft identification, latitude and longitude firstly entered or keyed into the computer simulating the aircraft. Since the latitude has a maximum of 3 characters and the longitude has a maximum of 4 characters, it is required to send four characters for longitude and 3 characters for latitude. The problem of sending two different number of characters for one position information can be solved by assuming latitude from 90°S to 90°N as from 0° to 180° respectively and longitude from 180°W to 180°E as from 0° to 360° respectively. Therefore, it is possible to send 3 characters for latitude as well as longitude. The number of characters required for the aircraft identification depends on the number of aircrafts required to monitor and in this simulation system it is assumed that there are nine (9) aircrafts that are to be monitored. This aircraft identification data is transmitted as a one byte ASCII character code and the latitude and longitude data are transmitted as a three byte ASCII code each. Totally there are seven (7) bytes to be transmitted serially for every position information for every aircraft.

The seven bytes serially transmitted will be received at the other computer simulating the ground station. In here the bytes are interpreted and a code or unique character is displayed for the identified aircraft on the screen of the computer simulating the ground station, where the background of the screen is the map of the world. The same data can be transmitted to other computers which might simulate the other concerned departments or sections on the aircraft's position.

In data communication there should be a rule by which there will be full correspondence between the sender and the receiver. This can be achieved by setting up rules, whereby the receiver could recognize that a data is coming and transmission of data is completed as well as which set of bits are to be interpreted as a character or frame depending upon the type of transmission. This is what is called designing protocol.

6.1.1 Bit Synchronization

In the project work, since once data are sent to the port of the computer the data will stay at the port as long as it is reset or different data are transmitted, a data and a strobe line are used, and data and strobe signals are transmitted at the same time. The strobe line will be high when

ever the voltage on the data line corresponds to a new data and will go low immediately. That means the strobe line indicates the frequency of the data transmission.

The receiver will receive the data line and the strobe line, and will accept the data only when the strobe line is high. In this way all the seven bytes will be transmitted and for each bit there is one strobe signal, that is, a high voltage level on the strobe line.

6.1.2 Framing

Up to this point the receiver knows when data is available and when it is not at the input port. But there is no means of identifying which set of bits constitute a transmitted byte and which sets of bytes constitute the position information, that is, the seven bytes (one frame).

It has been indicated in section 5 that one of the reasons to design protocol is to solve operating problems in the framing area. To make the receiver understand the start of a frame, a code to be interpreted on the receiver side as a start of frame, has to be sent. It is possible to choose any code that will not affect the primary objective of the system. For this particular system any code can be chosen but it is preferably not to be a code that represent a num-

ber, since all the data for this particular systems are numeric. The ASCII code 00101010 which represent a '.' is chosen to indicate start of a frame.

6.1.3 Error Detection

The receiver now knows which of the data line voltages correspond to a data and also knows where the frame starts. But still the data can be corrupted by noise, and there is a possibility where the code representing the start of the frame is error free, and the position and/or the aircraft identification data can be erroneous. In this case, the position and/or the aircraft identified, indicated on the screen do not represent the exact position and/or aircraft, or might be out of the range of position or aircraft identification for the particular system. That means there must be a way, either to correct the error or detect and discard the data, since it doesn't represent the actual information that is required.

One of the most effective method of detecting error in data communication is the Cyclic Redundancy Check (CRC). Since the data transmission uses 8 bits ASCII code, the CRC-16 which uses eight bit characters is selected from the most common CRC checks (CRC-12, CRC-16 and CRC-CCITT). CRC-16 provides error detection of bursts up to 16 bits. As

discussed in section 5, the hardware implementation shown in fig 5.2 is converted to a software CRC calculation and the flow chart and the program is presented in Appendix 2.

So far including the start of frame code, totally 8 bytes will be transmitted for each position information and for each aircraft. For error correction, two bytes of data will have two bytes of BCC and, overall the 8 bytes data will have 8 bytes of BCC, and therefor for one position information 16 bytes will be transmitted.

Layout of the data transmission or flow is shown below in fig 6.1 and in fig 6.2 World map and the nine aircrafts as displayed on the receiver PC screen is shown.

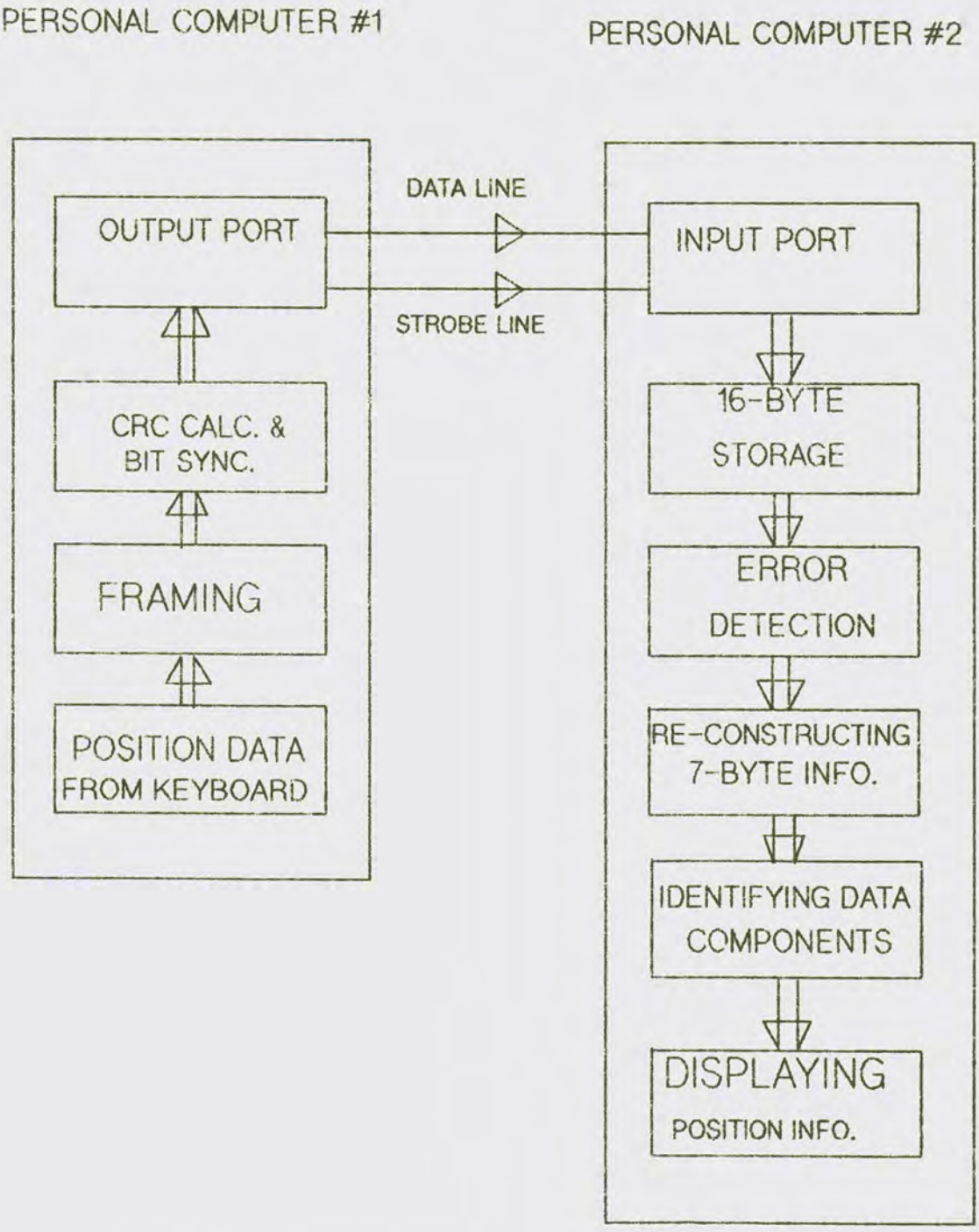


FIG 6.1 Data Flow between the PC's

767 + AIE 5 N 38 E
* AIF 80 N 80 E
- AIZ 5 N -38 E
727 o AHK 40 N -50 E
! AHL 10 N 0 E
AHM 40 N 80 E
737 & AJA 7 N 20 E
707 % AIV 50 N -85 E
@ ACQ 50 N 20 E
FLIGHT FOLLOWING SYSTEM

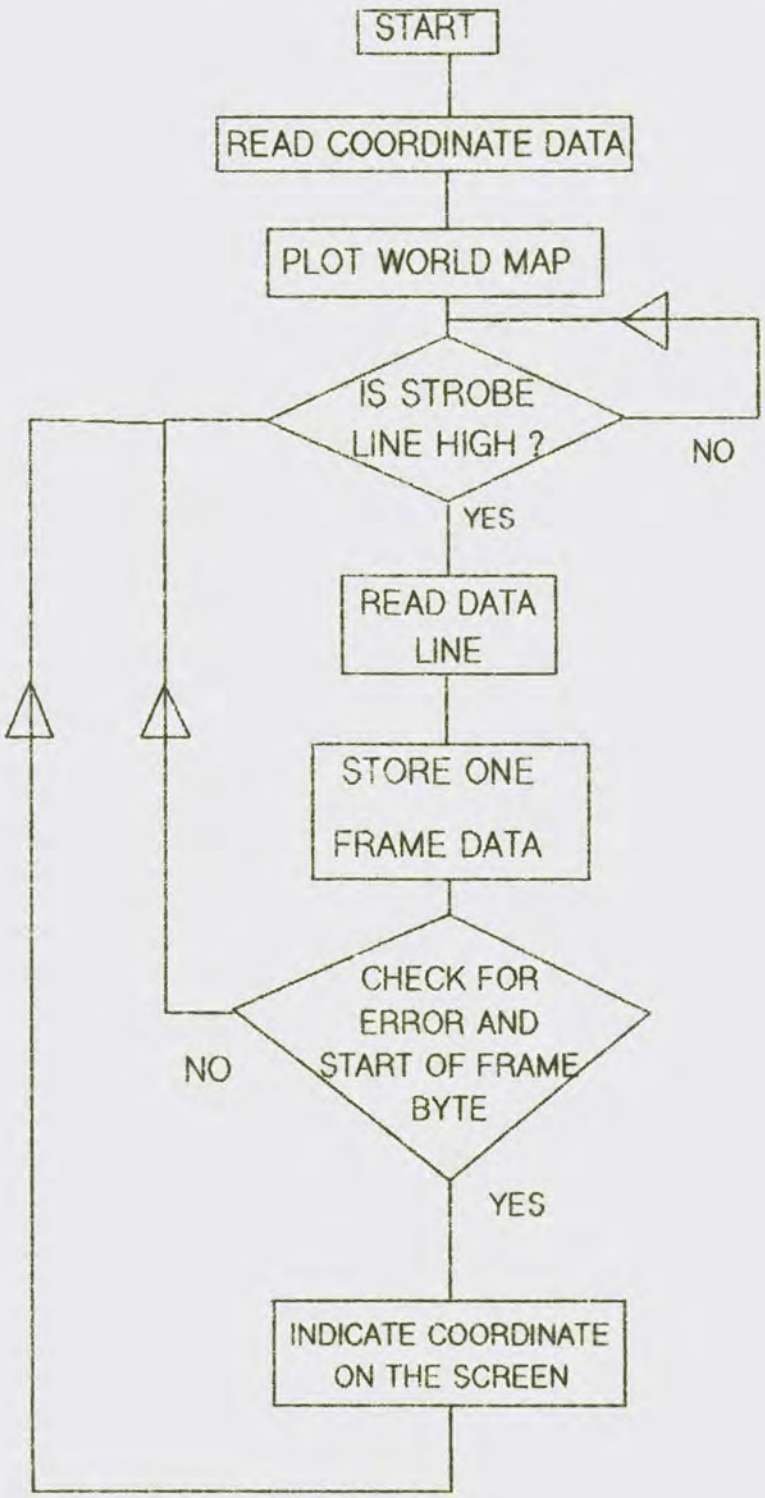
Fig. 6.2

7.0 Conclusion

In this thesis simulation of flight following system is done. The basic principle in this simulation is transmitting data and receiving the same data. In this process all the possible means of transmitting, receiving, error detection and synchronization methods have been seen. For the specific simulation system, as described in the text, some methods are employed and some of them are from my own way to fulfill the requirements for the system.

The methods used in this simulation system can be expanded to some other forms of data transfer like file transfer, computer-computer communication etc. and not only for position data.

FLOW CHART OF RECEIVER PROGRAM



APPENDIX I

SOURCE PROGRAM OF RECEIVER COMPUTER

```
10  CLS:KEY OFF
20 REM LPRINT CHR$(27);"3";CHR$(24)
30 LOCATE 25,1
40 PRINT "              F L I G H T     F O L L O W I  N  G
S Y S T E M   "
50 REM DRAWING BOARDER LINES
60 LINE (1,279)-(720,279)
70 LINE (1,347)-(720,347)
80 LINE (0,1)-(0,348)
90 LINE (719,1)-(719,348)
100 LINE (1,1)-(720,1)
110 REM READING COORDINATES OF WORLD MAP
120 OPEN "map2.bas" FOR INPUT AS #1
130 FOR I=1 TO 465
140 INPUT #1,Y,X
150 IF Y=85 AND X=-60 THEN LINE (259,11)-(259,11)
160 IF Y > 70 THEN Y=1.07*Y
170 IF Y > 78 THEN Y=1.2*Y/1.07
180 X1=1.7*X+360
190 Y1=-1.7*Y+185
200 IF I=1 THEN PSET(X1,Y1)
210 IF X=-72 AND Y=13 THEN PSET(X1,Y1)
220 IF X=43 AND Y=12 THEN PSET(X1,Y1)
```

```
230 IF Y=-35 AND X=150 THEN PSET(X1,Y1)
240 IF Y=-25 AND X=45 THEN PSET(X1,Y1)
250 IF X=-78 AND Y=9 THEN PSET(X1,Y1)
260 REM DRAWING WORLD MAP
270 LINE -(X1,Y1),,,&HAAAA
280 NEXT I
290 X2=0
300 Y2=0
310 N767ED=0:N767FD=0:N767ZD=0
320 N737AD=0:N727KD=0:N727LD=0
330 N727MD=0:N707VD=0:N707QD=0
340 REM DEFINING AIRCRAFT IDENTIFICATION CHARACTER
350 LOCATE 21,1:PRINT " + * - o ! # & % @ "
360 DIM A1(150),A2(150),A3(150),A4(150),A5(150),A6(150)
,A7(150),A8(150),A9(150),A10(150)
370 GET(8,280)-(20,290),A1
380 GET(20,280)-(40,290),A2
390 GET(40,280)-(60,290),A3
400 GET(60,280)-(75,290),A4
410 GET(75,280)-(95,290),A5
420 GET(95,280)-(112,290),A6
430 GET(112,280)-(130,290),A7
440 GET(130,280)-(150,290),A8
450 GET(150,280)-(165,290),A9
460 GET(165,280)-(190,290),A10
470 LOCATE 21,1:PRINT"                                    "
480 LINE (0,1)-(0,348)
```

```
490 ICOUNT = 0
500 ICOUNT=ICOUNT+1
510 IF ICOUNT > 16 GOTO 850
520 N=0
530 OUT &H378,0
540 REM WAITING FOR DATA
550 WAIT &H378,2
560 A=INP(&H378)
570 C=1 AND A
580 D=C*2^N
590 IF N=0 THEN C1=C
600 B1=C1 OR D
610 C1=B1
620 N=N+1
630 IF N > 7 GOTO 670
640 WAIT &H378,2,2
650 GOTO 550
660 REM RECONSTRUCTING DATA
670 ON ICOUNT GOTO 680,690,700,710,720,730,740,750,760,770
,780,790,800,810,820,830
680 FRBCC=B1:GOTO 500
690 BBCC=B1:GOTO 500
700 D42=B1:GOTO 500
710 B=B1:GOTO 500
720 V11BCC=B1:GOTO 500
730 V12BCC=B1:GOTO 500
740 V11=B1:GOTO 500
```

```
750 V12=B1:GOTO 500
760 V13BCC=B1:GOTO 500
770 V21BCC=B1:GOTO 500
780 V13=B1:GOTO 500
790 V21=B1:GOTO 500
800 V22BCC=B1:GOTO 500
810 V23BCC=B1:GOTO 500
820 V22=B1:GOTO 500
830 V23=B1:GOTO 500
840 REM CRC CALCULATION
850 A1=0:B1=0
860 ICOUNT = 1
870 ON ICOUNT GOTO 880,890,900,910,920,930,940,950
880 C=42:D=B:GOTO 960
890 C=FRBCC:D=BBCC:GOTO 960
900 C=V11:D=V12:GOTO 960
910 C=V11BCC:D=V12BCC:GOTO 960
920 C=V13:D=V21:GOTO 960
930 C=V13BCC:D=V21BCC:GOTO 960
940 C=V22:D=V23:GOTO 960
950 C=V22BCC:D=V23BCC
960 REM C AND D ARE DATA REGISTERS
970 REM A AND B ARE BCC REGISTERS
980 D1 = D
990 C1 = C
1000 FOR COUNT = 1 TO 16
1010 N0 = (D1 AND 1) XOR (B1 AND 1)
```

```
1020 N1 = (INT(B1 AND 2)/2) XOR N0
1030 N2 = (INT(A1 AND 64)/64) XOR N0
1040 B1 = INT(B1/2)
1050 B1 = B1 OR ((A1 AND 1)*2^7)
1060 A1 = INT((A1/2)) OR (N0*2^7)
1070 IF N1 = 1 THEN B1=B1 OR 1
1080 IF N1 = 0 THEN B1=B1 AND 254
1090 IF N2 = 1 THEN A1=A1 OR 32
1100 IF N2 = 0 THEN A1=A1 AND 223
1110 D1 =INT(D1/2)
1120 D1 = D1 OR ((C1 AND 1)*2^7)
1130 C1 = INT(C1/2)
1140 NEXT COUNT
1150 REM CHECKING FOR ERROR
1160 IF ICOUNT=2 AND (A1=0 AND B1=0) THEN CD42=D42
1170 IF ICOUNT=2 AND (A1=0 AND B1=0) THEN CB=B
1180 IF ICOUNT=2 AND (A1<>0 OR B1<>0) THEN GOTO 490
1190 IF ICOUNT=4 AND (A1=0 AND B1=0) THEN CV11=V11
1200 IF ICOUNT=4 AND (A1=0 AND B1=0) THEN CV12=V12
1210 IF ICOUNT=4 AND (A1<>0 OR B1<>0) THEN GOTO 490
1220 IF ICOUNT=6 AND (A1=0 AND B1=0) THEN CV13=V13
1230 IF ICOUNT=6 AND (A1=0 AND B1=0) THEN CV21=V21
1240 IF ICOUNT=6 AND (A1<>0 OR B1<>0) THEN GOTO 490
1250 IF ICOUNT=8 AND (A1=0 AND B1=0) THEN CV22=V22
1260 IF ICOUNT=8 AND (A1=0 AND B1=0) THEN CV23=V23
1270 IF ICOUNT=8 AND (A1<>0 AND B1<>0) THEN GOTO 490
1280 ICOUNT=ICOUNT+1
```

```
1290 IF ICOUNT=3 THEN A1=0
1300 IF ICOUNT=3 THEN B1=0
1310 IF ICOUNT=5 THEN B1=0
1320 IF ICOUNT=5 THEN A1=0
1330 IF ICOUNT=7 THEN A1=0
1340 IF ICOUNT=7 THEN B1=0
1350 IF ICOUNT > 8 THEN GOTO 1380
1360 GOTO 870
1370 REM CHECKING FOR FRAME IDENTIFICATION
1380 IF CD42=42 THEN GOTO 1410
1390 GOTO 490
1400 REM CONSTRUCTING POSITION AND AIRCRAFT IDENTIFICATION
1410 V1=V11*100+V12*10+V13:V2=V21*100+V22*10+V23
1420 ALONG$="E":ALAT$="N"
1430 C=B:Y=V1-90:X=V2-180
1440 X1=1.7*X+360
1450 Y1=-1.7*Y+185
1460 IF X<0 THEN ALONG$="W"
1470 IF X<0 THEN X=ABS(X)
1480 IF Y<0 THEN ALAT$="S"
1490 IF Y<0 THEN Y=ABS(Y)
1500 REM DISPLAYING POSITION AND AIRCRAFT CODE
1510 ON C GOTO 1520,1600,1660,1720,1780,1840,1900,1970,2030
1520 IF N767ED <> 0  THEN PUT(X767ED,Y767ED),A1
1530 PUT(X1,Y1),A1:N767ED=1
1540 X767ED=X1:Y767ED=Y1
1550 LOCATE 21,15
```

```
1560 PRINT "                              "
1570 LOCATE 21,15
1580 PRINT "767 + AIE" Y;ALAT$;X;ALONG$
1590 GOTO 490
1600 IF N767FD <> 0  THEN PUT(X767FD,Y767FD),A2
1610 PUT(X1,Y1),A2:N767FD=1
1620 X767FD=X1:Y767FD=Y1
1630 LOCATE 21,40
1640 PRINT "* AIF" Y;ALAT$;X;ALONG$
1650 GOTO 490
1660 IF N767ZD <> 0  THEN PUT(X767ZD,Y767ZD),A3
1670 PUT(X1,Y1),A3:N767ZD=1
1680 X767ZD=X1:Y767ZD=Y1
1690 LOCATE 21,62
1700 PRINT "- AIZ" Y;ALAT$;X;ALONG$
1710 GOTO 490
1720 IF N727KD <> 0  THEN PUT(X727KD,Y727KD),A4
1730 PUT(X1,Y1),A4:N727KD=1
1740 X727KD=X1:Y727KD=Y1
1750 LOCATE 22,15
1760 PRINT "727 o AHK" Y;ALAT$;X;ALONG$
1770 GOTO 490
1780 IF N727LD <> 0 THEN PUT(X727LD,Y727LD),A5
1790 PUT(X1,Y1),A5:N727LD=1
1800 X727LD=X1:Y727LD=Y1
1810 LOCATE 22,40
1820 PRINT "! AHL" Y;ALAT$;X;ALONG$
```

```
1830 GOTO 490
1840 IF N727MD <> 0 THEN PUT(X727MD,Y727MD),A6
1850 PUT(X1,Y1),A6:N727MD=1
1860 X727MD=X1:Y727MD=Y1
1870 LOCATE 22,62
1880 PRINT "# AHM" Y;ALAT$;X;ALONG$
1890 GOTO 490
1900 IF N737AD <> 0 THEN PUT(X737AD,Y737AD),A7
1910 PUT(X1,Y1),A7:N737AD=1
1920 X737AD=X1:Y737AD=Y1
1930 LOCATE 23,15
1940 PRINT "737 & AJA" Y;ALAT$;X;ALONG$
1950 REM IF N=4 GOTO 470
1960 GOTO 490
1970 IF N707VD <> 0 THEN PUT(X707VD,Y707VD),A8
1980 PUT(X1,Y1),A8:N707VD=1
1990 LOCATE 23,40
2000 PRINT "707 % AIV" Y;ALAT$;X;ALONG$
2010 REM IF N=4 GOTO 470
2020 GOTO 490
2030 IF N707QD <> 0 THEN PUT(X707QD,Y707QD),A9
2040 PUT(X1,Y1),A9:N707QD=1
2050 X707QD=X1:Y707QD=Y1
2060 LOCATE 23,62
2070 PRINT "@ ACQ" Y;ALAT$;X;ALONG$
2080 REM IF N=4 GOTO 470
2090 GOTO 490:END
```

SOURCE PROGRAM OF TRANSMITTER COMPUTER

```
10 ICOUNT=1
20 PRINT ""
30 OUT &H378,0
40 T=1
50 REM DATA INPUT
60 INPUT "ENTER AIRCRAFT IDENTIFICATION NUMBER  (VALID  NUM-
BERS FROM 1 - 9) ";B
70  IF  (B  > 9 OR B < 1) THEN PRINT "ILLEGAL AIRCRAFT IDEN-
TIFICATION TRY AGAIN"
80 IF (B > 9 OR B < 1) THEN GOTO 40
90 PRINT "  "
100 INPUT "ENTER LATITUDE  (VALID NUMBERS FROM  -90  ->  90)
";V1
110  IF  (V1  < -90 OR V1 > 90) THEN PRINT "ILLEGAL LATITUDE
TRY AGAIN"
120 IF (V1 < -90 OR V1 > 90) THEN GOTO 100
130 V1=V1+90
140 V11=INT(V1/100)
150 V12=INT((V1-V11*100)/10)
160 V13=INT(V1-V11*100-V12*10)
170 PRINT "  "
180 INPUT "ENTER LONGITUDE (VALID NUMBERS FROM -180 ->  180)
";V2
190 IF (V2 < -180 OR V2 > 180) THEN PRINT "ILLEGAL LONGITUDE
TRY AGAIN"
```

```
200 IF (V2 < -180 OR V2 > 180) THEN GOTO 180
210 V2=V2+180
220 V21=INT(V2/100)
230 V22=INT((V2-V21*100)/10)
240 V23=INT(V2-V21*100-V22*10)
250 REM  BCC CALCULATION
260 IBCC=1
270 ON IBCC GOTO 280,290,300,310
280 C=42:D=B:GOTO 320
290 C=V11:D=V12:GOTO 320
300 C=V13:D=V21:GOTO 320
310 C=V22:D=V23
320 A1=0:B1=0
330 D1=D:C1=C
340 FOR COUNT = 1 TO 16
350 N0 = (D1 AND 1) XOR (B1 AND 1)
360 N1=(INT(B1 AND 2)/2) XOR N0
370 N2 = (INT(A1 AND 64)/64) XOR N0
380 B1=INT(B1/2)
390 B1 = B1 OR ((A1 AND 1)*2^7)
400 A1=INT((A1/2)) OR (N0*2^7)
410 IF N1 = 1 THEN B1=B1 OR 1
420 IF N1 = 0 THEN B1=B1 AND 254
430 IF N2 = 1 THEN A1=A1 OR 32
440 IF N2 = 0 THEN A1=A1 AND 223
450 D1 = INT(D1/2)
460 D1=D1 OR ((C1 AND 1)*2^7):C1=INT(C1/2)
```

```
470 NEXT COUNT
480 ON IBCC GOTO 490,500,510,520
490 FRBCC=A1:BBCC=B1:GOTO 530
500 V11BCC=A1:V12BCC=B1:GOTO 530
510 V13BCC=A1:V21BCC=B1:GOTO 530
520 V22BCC=A1:V23BCC=B1
530 IBCC=IBCC+1
540 IF IBCC > 4 THEN GOTO 560
550 GOTO 270
560 ON ICOUNT GOTO 570,580,590,600,610,620,630,640,650,660
,670,680,690,700,710,720
570 DT=FRBCC:GOTO 730
580 DT=BBCC:GOTO 730
590 DT=42:GOTO 730
600 DT=B:GOTO 730
610 DT=V11BCC:GOTO 730
620 DT=V12BCC:GOTO 730
630 DT=V11:GOTO 730
640 DT=V12:GOTO 730
650 DT=V13BCC:GOTO 730
660 DT=V21BCC:GOTO 730
670 DT=V13:GOTO 730
680 DT=V21:GOTO 730
690 DT=V22BCC:GOTO 730
700 DT=V23BCC:GOTO 730
710 DT=V22:GOTO 730
720 DT=V23
```

```
730 REM 16 BYTE SERIAL TRANSMISSION
740 N=1
750 OUT &H378,0
760 C=2 OR DT
770 OUT &H378,C
780 N=N+1
790 IF N>8 THEN ICOUNT = ICOUNT +1
800 K=C AND 1
810 OUT &H378,K
812 IF ICOUNT > 16 THEN PRINT ""
820 IF ICOUNT > 16 THEN GOTO 860
822 IF ICOUNT > 16 THEN GOTO 10
830 IF N>8 THEN GOTO 560
840 DT=INT(DT/2)
850 GOTO 750
860 FOR I=1 TO 2000:NEXT I
870 ICOUNT=1:T=T+1
880 IF T>2 GOTO 10
890 GOTO 560
900 END
```

APPENDIX II

SOURCE PROGRAM OF CRC CALCULATION

```
10 C=42:D=8
20 A=0:B=0
30 REM C AND D ARE DATA REGISTERS
40 REM A AND B ARE BCC REGISTERS
50 D1 = D
70 C1 = C
90 FOR COUNT = 1 TO 16
100 N0 = (D1 AND 1) XOR (B1 AND 1)
110 N1 = (INT(B1 AND 2)/2) XOR N0
120 N2 = (INT(A1 AND 64)/64) XOR N0
130 B1 = INT(B1/2)
140 B1 = B1 OR ((A1 AND 1)*2^7)
150 A1 = INT((A1/2)) OR (N0*2^7)
160 IF N1 = 1 THEN B1=B1 OR 1
170 IF N1 = 0 THEN B1=B1 AND 254
180 IF N2 = 1 THEN A1=A1 OR 32
190 IF N2 = 0 THEN A1=A1 AND 223
200 D1 =INT(D1/2)
210 D1 = D1 OR ((C1 AND 1)*2^7)
220 C1 = INT(C1/2)
222 PRINT C1,D1
230 NEXT COUNT
240 PRINT A1;B1
```

REFERENCES

1. John E. McNamara, "Technical Aspects of Data Communication," Second Edition Digital Equipment Corporation USA 1982.

2. Jery FitzGerald, Tom S. Eason, "Fundamentals of Data Communications," John Wiley/Hamilton Publication New York 1978.

3. Robert Cole, "Computer Communications," Department of Computer Science University College, London. Second Edition Published by Macmillan Education Ltd. Houndmills, Basingstoke, Hampshire RG21 2XS & London 1986.

4. Richard Deasington, "A Practical Guide to Computer Communications and Networking," Second Edition Ellis Horwood

Ltd. Market Cross House, Cooper Street, Chichester, West Sussex; P0191EB, England 1984.

5. D. W.Davies, D. L.A Barber, W. L. Price and C. M. Solomonides, "Computer Networks and their Protocols," A Wiley Interscience Publication John Wiley and Sons 1979.

6. Mary E. S. Loomis, "Data Communications," Prentice-Hall, Inc., Englewood Cliff, New Jersey 07632 1983.

7. Franklin F. Kuo, "Protocols and Techniques for Data Communications Networks," Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632 1981.

8. Vijay Ahuja, Ph. D. "Design and Analysis of Computer Communication Networks," McGraw-Hill Inc. New York 1986.

DECLARATION

I, the undersigned, declare that this thesis is my work and that all sources of material used for this thesis have been duly acknowledged.

NAME : Mathewos Essatu

Signature : [signature]

Date of submission : July 5, 1988