VUS1/2
VU0001508

DATE DUE

THE/ 1988/2
Box THE 00002

RESERVE
THESIS
004.68
SIM

FOOTSCRAY INSTITUTE OF TECHNOLOGY
LIBRARY

6/1308815d

X

NOT FOR LOAN

BRN: 380723

A Local Area Network Using Spread Spectrum Techniques

A Thesis submitted for examination for the Degree of Master of Engineering (Electrical)

By

A. L. Simcock

BSc(Hons)(Electronics)

University of Kent at Canterbury,

Canterbury.

England.

Department of Electrical and Electronic Engineering,

Footscray Institute of Technology,

Victoria, Australia.

December, 1988

Table of Contents

Appendices

List of Figures

List of Tables

Acknowledgements

This research project was supported by a grant from the Australian Telecommunications and Electronics Research Board (ATERB).

The project was supervised by Dr. C. naRanong and Mr. P. Leung, for whose advice and support I am grateful.

My warm thanks are extended to many people, in particular:-

+ My wife, Moira, for typing this thesis

+ My friends and colleagues at F.I.T., for their help and encouragement

+ Mr. F. Hayward, for laying-out the printed circuit boards

+ Mr. R. Ives, Mr. M. Dow, Mr. M. Faulkner and Mr. J. MacLeod, for their many informative discussions

+ Dr. R. Malyniak and Dr. A. Kalam for their support, proof reading and criticism of this thesis.

Abstract

Local Area Networks (**LANs**) have become fairly common in recent years. This thesis considers the application of Spread Spectrum (**SS**) communication techniques to the implementation of a LAN.

Basic SS techniques are described and a model of a LAN using these techniques is developed. The major difference between this LAN and most other LAN implementations is that in this system many nodes (or stations) may transmit concurrently. The primary feature of this model is the ability to distinguish between transmissions intended for different users by the address code used to modulate the data. This is called Code Division Multiple Access (**CDMA**). The properties of the codes, allocated to each node in the network, are of paramount importance in a CDMA system. Several code allocation schemes are considered in order to find one with the most suitable properties. System timing and synchronisation concepts are also considered, since they, together with the code allocation scheme selected, determine the characteristics of the system.

The hardware designed to implement and control network functions is described. Tests carried out to verify the operation of the independent hardware modules are also detailed. The ability of the system to synchronise and transmit data in the presence of noise is tested, prior to complete system testing and throughput measurement.

CDMA is used to separate information on different channels but there is still the need to consider possible contention within a channel. A Carrier Sense Multiple Access with Collision Detection (**CSMA/CD**) mechanism is used for this purpose. A theoretical model of a suitable mechanism is developed and simulated. The theoretical and simulated results are compared and used to predict the results expected from the hardware. Finally the system's traffic handling characteristics are tested and the results compared with those predicted. System flexibility is investigated by programming an alternative protocol into each node and again examining the throughput characteristics.

This thesis concludes with a discussion of the major features highlighted by this development project. Theoretical, simulated and actual measured results are also compared. Cost and certain system performance parameters are compared with those of a more conventional CSMA/CD network, such as Ethernet. Possible system enhancements and avenues for future work are then discussed.

1 INTRODUCTION

1.1 Summary

This thesis contains the description of the research carried out towards the production of a Local Area Network (**LAN**) based on Spread Spectrum (**SS**) techniques [1]. The concept of using Spread Spectrum techniques is attractive since simultaneous transmissions by several users is possible, using a Code Division Multiple Access (**CDMA**) mechanism.

The evolution of the requirement, a brief introduction to the essential theory and a precis of the contemporary Spread Spectrum LAN research projects are all outlined in chapter 1. An outline of the project design criteria and system specifications are included in chapter 2. The choice of code generation and distribution is crucial to the performance of a CDMA system. Chapter 3 briefly outlines the different coding schemes considered before a final choice was made. After considering several schemes, a synchronised system was adopted as an appropriate code allocation scheme. Timing and code related matters are inseperable in a synchronised system; because of this, aspects of code recognition and collision detection and certain system timing concepts are also described in chapter 3.

Chapter 4 describes the hardware required to generate the codes and meet the timing requirements etc outlined in previous chapter. Photographs of the two printed circuit boards are also included to aid description. Where individual circuit modules have been tested in isolation the tests descriptions are included with the details of the hardware of that particular module. Multi-module and integration tests are, however, detailed separately in chapter 5. Since the hardware is described in terms of the individual circuit modules the complete circuit diagrams, for each of the two boards which constitute a system node, are included in appendices rather than the main section of the text.

The hardware described in chapters 4 and 5 represents the physical components of the system; it is the software which controls the hardware, however, in which the higher level system protocols are embedded. Theoretical and simulation models of channel throughput were developed in order to predict system performance before the control software was written. These models are described in chapter 6. The control software and the tests carried out to verify system performance parameters are described in chapter 7.

A discussion of system performance parameters and ideas for further research are included with an overall evaluation of the research project in the conclusions section, chapter 8.

The aim of this thesis is to describe the main points of the research carried out towards the development of a Spread Spectrum LAN. For this reason all of the control, test and peripheral programs which were written are included as appendices rather than in the main body of the text. Details of special purpose test equipment, photographs of the circuit boards and complete circuit diagrams are also included as appendices.

1.2 The Emergence of Local Area Networks

The advent of the microprocessor has had a significant impact on society. The substantial reduction in the cost of computing power has resulted in the widescale use of computers for all kinds of data manipulation and storage. Magnetic storage devices such as floppy disks have, to some extent, replaced paper as a means of intra and inter company data transfer. Local Area Networks (LANs) were primarily developed to allow the sharing of expensive resources, but their development has also allowed the sharing of data between interested parties without the need to manually transport disks, paper tapes etc. The expanding need for data communication channels in LANs has increased the pressure on the available transmission bandwidth. This has led to developments in two broad areas : the extension of the usable spectrum through the use of even higher frequency carriers and the development of complex networking and multiple access algorithms.

The ever increasing use of LANs has brought with it two problems with respect to the protection of data. First it is necessary to protect the data against interference, be it deliberate or otherwise, in order to ensure delivery of correct information to the correct user. Second, as more proprietory information is appearing on data bases, it is becoming increasingly necessary to protect the information against prying or interception. Society has, indeed, become information conscious to the extent that the acquisition and protection of data has become an industry in its own right.

SS techniques, which were originally developed in the mid 1950s [2] for military use as a means of providing a jam resistant communication link, have only recently been considered seriously in a commercial or civilian environment [3]. Areas of interest include mobile radio, radio telephony, amateur radio and satellite communications [4].

1.3 Spread Spectrum in LANs

The adaptation of SS techniques to the development of a LAN appears to offer some advantages in the areas of concern mentioned above. Its well known interference rejection capabilities [5] could be used to advantage in allowing the system to work in an environment where electrical interference might be a problem and where the use of alternatives such as fibre optic cables might not be appropriate[6,7]. SS systems, by their very nature, also offer possible benefits [2,8,9] in the area of data privacy.

One definition of SS [5] which adequately reflects the major characteristics of this technique is as follows:-

> "Spread Spectrum is a means of transmission in which the signal occupies a bandwidth in excess of the minimum necessary to send the information. The band spreading is accomplished by means of a code which is independent of the data. A synchronised version of the same

> code is then used at the receiver in order to despread the signal and facilitate data recovery."

What this definition means is that the data is modulated with the spreading code and that, through a correlation process, the receiver restores the signal to its normal bandwidth and, in doing so, it enhances the desired signal while substantially reducing the effects of interference and noise. Overall, the communication channel capacity is not "wasted" because with a suitable selection of spreading codes (having low crosscorrelation properties) several nodes may simultaneously transmit messages over the same channel, with low, predictable, interference between nodes.

A LAN, where multiple nodes use a single communication channel - one pair at a time, is a prime candidate for the application of SS techniques. By using CDMA techniques, alternatively called Spread Spectrum Multiple Access (**SSMA**), it is possible to have several multi-pair communications in progress simultaneously [10,11]. Being inherently multi-user, CDMA techniques have the potential to increase the network's throughput, under conditions of heavy usage, by allowing many pairs of nodes to communicate simultaneously, thus reducing contentions.

The three primary methods of spreading the data transmissions are [5,12]:-

- Direct Sequence - in which the data is directly modulated with a fast pseudo random code sequence.

- Frequency Hopping - in which the carrier which is modulating the data is caused to hop from one frequency to another in a pseudo random manner.

- Time Hopping - in which the time channel allocated to a particular transmission is changed in a pseudo random manner.

1.4 Literature Survey and Existing Systems

A survey of literature revealed many papers on Spread Spectrum techniques and many more on Local Area Networks, but only four papers which described research which combined both [13 - 16]. Several more papers relating to one of the four existing research projects were published subsequently. These papers described the project at various stages of its development, the most recent [17] was used as a reference. The projects described were at various stages of completion, varying from systems which had been extensively tested to those which had been simulated, but for which no hardware had been built.

The characteristics of each system in brief are as follows:-

1.4.1 System A [13 and 17]

Smythe and Spracklen [13] describe their SS LAN system, and the software simulations used to test it, in general terms. Spracklen et. al. [17], however, give a much more comprehensive description of the system. The system, called SPREADNET, is a bus topology system which is connected via fibre optics, and is aimed specifically towards military applications. In this network, which can be up to 2 km. in length, as many as 150 users can be accommodated. Direct sequence modulation is used and encoded data is transmitted in baseband at 100 M bits/sec. Each node controller uses a 12 MHz MC68000 microprocessor.

A source code address allocation scheme is used, where the receiver must be informed which demodulation code to use. Upon receipt of a packet for transmission the node controller requests a virtual communications channel ie. requests address codes to be used for the transmission. The management system then informs the transmitter and receiver of the choice of codes. Contentions are avoided by this elaborate network management interchange before logical code to physical device mapping is achieved.

All codes allocated in this manner are encrypted before delivery and are protected by a two key cryptographic algorithm in conjunction with the Data Encryption Standard (DES). Dynamically allocated codes are also used by the individual node controllers to transmit network management information around the system.

Synchronisation is achieved by means of polarity coincidence correlators, and digital correlation is performed continuously to overcome tracking problems. Modulation and demodulation are handled by dedicated hardware (custom programmed logic arrays), and a sliding correlator is used in conjunction with polarity coincidence correlation for data recovery.

1.4.2 System B [14]

System B was developed, to investigate network efficiency and performance criteria and compare them with the classic Frequency Division Multiple Access (FDMA) ALOHA type of network. The system is cable connected, and used "uplink" and "downlink" separation of transmission and reception paths interfaced via a "Repeater and Sync generator". Bus topology and direct sequence (DS) modulation are used. Bit synchronisation is achieved via a sophisticated mechanism whereby the Repeater and Sync generator periodically interrupts traffic to inject sync information for each user. This must be done with sufficient frequency so that the local clocks do not drift by more than one chip bit period between updates.

The system is designed to accommodate voice and data. Voice or data bits are transmitted in packets. Different processing gains are used in the system, and data is modulated using a variable number, between 64 and 512, of chip bits per data bit, whereas a processing gain of 167 is used for voice traffic. This gives an effective modulated data rate of 10.9 MHz for voice traffic of 64 kHz. Bits of each packet, either voice or data, are modulo 2 added to the direct address code sequence. The resulting binary array then balance modulates a high frequency carrier which is transmitted in the uplink direction. The Multiple Access protocol is via point to point CDMA, with the destination address being used, but no code allocation details are included.

1.4.3 System C [15]

This system utilises what the designers call "a non-processed fibre optics bus", where non-processed is defined as supressing the entire command and address part of the bus. A distributed star topology is used to implement the bus, which is capable of supporting up to 25 users. Distributed 5 port optical couplers are used to allow a bus do be developed which does not use handshaking or a master-slave configuration.

Fibre optic interconnections are used and modulation takes the form of Frequency Hopping superimposed upon a Direct Sequence modulated data signal. 15 frequencies with a separation of 15 MHz centred on 155 MHz are used for the frequency hopping component, and a processing gain of 127 is used for the direct sequence part. Surface wave acoustic convolvers centred on 155 MHz are used for synchronisation.

1.4.4 System D [16]

For this system a fibre optic connected ring network and a separate synchronisation bus are used. Both synchronous and asynchronous data is catered for in the system.Node addresses are cyclically bit shifted versions of the one m-sequence. The sync signal sent from the Sync Node is also a bit shifted m-sequence. The system employs dual phase locked loops for synshronisation demodulation and data separation.

Each receiver removes its own modulated data signal from the received signal, so node #2 would see the same signal as node #1 minus any transmission addressed to node #1. This prevents the transmission for node #1 from recirculating in the system. Each node conducts an "in-channel" busy test before transmitting. When a node is transmitting it switches its receiver and transmitter m-sequence generators to the destination address. This ensures that no collisions can occur in that channel as the node would remove from any signal any component modulated with the same address code. It does, however, mean that a node may not transmit and receive at the same time.

The research described in these papers appeared to be concentrating on the high speed, extremely sophisticated end of possible applications. Considerable use was made of custom designed VLSI devices, multiple high speed correlators etc. Network management, sophisticated synchronisation, layered SS techniques, encryption and dynamic address code allocation were some of the other advanced features of the systems referenced. Three of the systems used, or proposed using, fibre optic interconnections and only one used cables. This concentration on the "**high end**" of possible applications left considerable scope for the development of a **SS LAN** which could find applications where the advanced features described above might not be appropriate (see System Specifications, section 2.4) and systems such as those described above may be too complicated.

1.5 Autocorrelation, Crosscorrelation and Orthogonality

The correlation between two signals gives an indication of the similarity between the signals, as a function of the time displacement (τ). The **crosscorrelation** between two signals **v(t)** and **w(t)** is defined by the scalar product shown below [18]

$$R_{vw}(\tau) = \langle v(t), w(t-\tau) \rangle \qquad (1)$$

Thus, if the correlation value is high the signals have a marked degree of similarity; and if the correlation value is very low the signals are essentially dissimilar. The **autocorrelation** function is defined as the correlation of a signal **v(t)** with a time displaced version of itself **v(t-τ)** and is given by:-

$$R_{vv}(\tau) \quad = \quad R_{v}(\tau) \quad = \quad < v(t)\,,\, v(t-\tau) > \qquad (2)$$

If $|R_v(\tau)|$ is large then the inference is that **v(t)** is very similar (proportional) to ± **v(t)** for that particular value of τ . Conversely if $R_v(\tau) = 0$ then, at that particular value of τ, **v(t)** and **v(t-τ)** are said to be **orthogonal** .

For a signal **z(t)** which is a summation of two signals **v(t)** and **w(t)**, such that **z(t) = v(t) + w(t)**, the correlation with one signal **v(t)** would be given by

$$R_{zv}(\tau) \; = \; R_v(\tau) \; + \; R_{wv}(\tau) \qquad (3)$$

and if the component signals are orthogonal at this particular value of τ ie. if

$$R_{vw}(\tau) \; = \; R_{wv}(\tau) \; = \; 0 \qquad (4)$$

then

$$Rzv(\tau) = Rv(\tau) \qquad (5)$$

This extraction of a single correlation function from a signal which is the summation of two or more signals is the basis upon which CDMA relies.

1.6 Features Required in CDMA Codes

In a CDMA system where one node is distinguished from another by means of the code used to modulate its data, the selection and allocation of codes is of prime importance, The address codes used must have certain properties [5,8] i.e. they:-

i) must be easy to generate.

ii) have randomness properties.

iii) have selectable length (long) periods.

iv) must be deterministic i.e. can be synchronised.

v) must be orthogonal

and if the code is intended to impart any substantial degree of encryption they must also be:-

vi) difficult to reconstruct from a short segment.

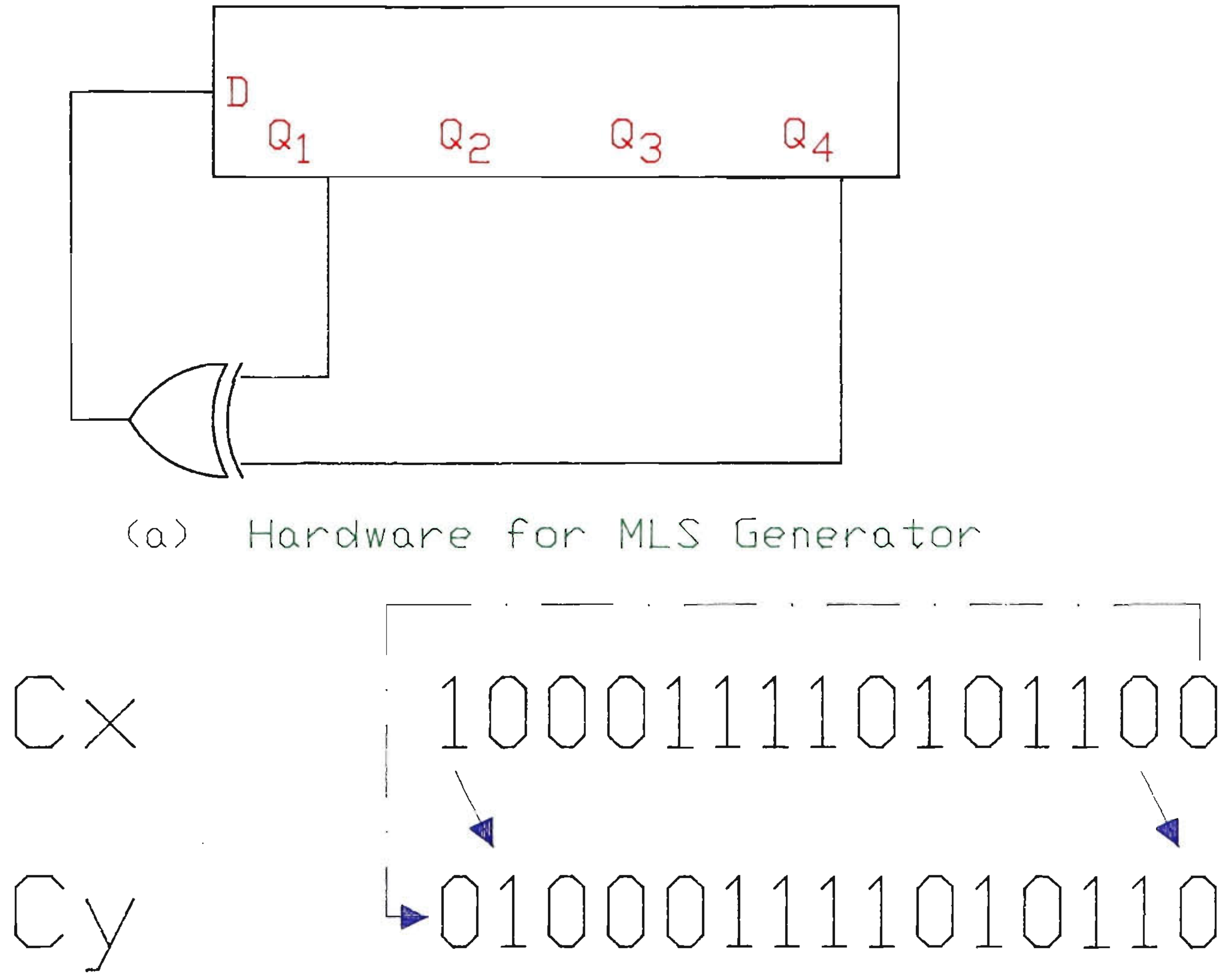
It can be seen that properties **i)**, **iv)** and **vi)** are somewhat contradictory since, if it is simple, deterministic and easy to synchronise for the intended receiver, it will probably prove to be relatively easy for an interceptor to "**break**" the code. Since, however, cryptographic integrity is not a primary requisite of this work, property **vi)** may be omitted thus opening the way to use codes which have only the first five properties.

1.7 Maximal Length Sequences

The most commonly used type of code which meets these five basic properties is the Maximal Length Sequence (**MLS** or **m-sequence**). The normal method of generating these codes is using Linear Feedback Shift Registers (LFSR) as illustrated in Figure 1a. For a code to be a maximal length code it must have a length, **N**, which is equal to 2^n-1, where **n** is the number of shift register stages in the generator. Tables are available [19] which show the feedback taps necessary to generate m-sequences for up to **34** shift register stages. In an m-sequence all binary **n-tuples**, except the all zero state, are present. By setting the initial contents of the shift registers to **0001** the pattern shown as code C_x in Figure 1b would result. This pattern repeats after $2^4-1 = 15$ bits. Figure 1b also shows code C_y which is the result of cyclically shifting code C_x by 1 bit such that $C_{x_k} = C_{y_{k-1}}$ where **k** represents the bit position.

The signals **v(t)** and **w(t)** in section 1.5 were continuous signals. The autocorrelation and crosscorrelation functions do, however, have a direct translation into discrete pseudo random signals such as m-sequences. By defining the ±**1** sequence $C_x'_k = 1-2C_{x_k}$ (where $C_{x_k} = 0,1$ - see Figure 1b) then the autocorrelation function is defined as

$$R_{C_x'}(\tau) = \sum_{1}^{N} C_x'_k \, C_x'_{k-\tau} \qquad (6)$$



(a) Hardware for MLS Generator

(b) Cyclic Shifting to Form Codes

Figure 1 Maximal Length Sequence Generator

This is the **Periodic Autocorrelation** of the m-sequence taken over the length, **N**, of the sequence, and τ is in whole bit increments. [5]

The **Periodic Crosscorrelation** of two sequences **C**x' and **C**y' would similarly be given by

$$R_{C_{x}'C_{y}'} = \sum_{1}^{N} C_{x}'{}_{k} C_{y}'{}_{k} \qquad (7)$$

which is a bit by bit summation of the number of bits in code **C**x' which are the same as those in code **C**y' over the length, **N**, of the codes. The periodic autocorrelation and crosscorrelation functions are of extreme importance when considering SS systems.

1.8 Basic Spread Spectrum Theory

This research project uses the Direct Sequence method of data spreading, and the description will concentrate on this aspect. The pseudo random binary sequence used to spread the data is also commonly known as the "**Chip**" sequence. Figure 2 illustrates a conceptual model of a single channel direct sequence system. In this system the chip sequence is used directly to modulate the data in baseband.

Each data bit **d(t)**, as illustrated in Figure 3, of energy E_b and duration **T**, may be represented by [5]

$$d(t) = \pm \sqrt{(E_b/T)} \qquad (8)$$

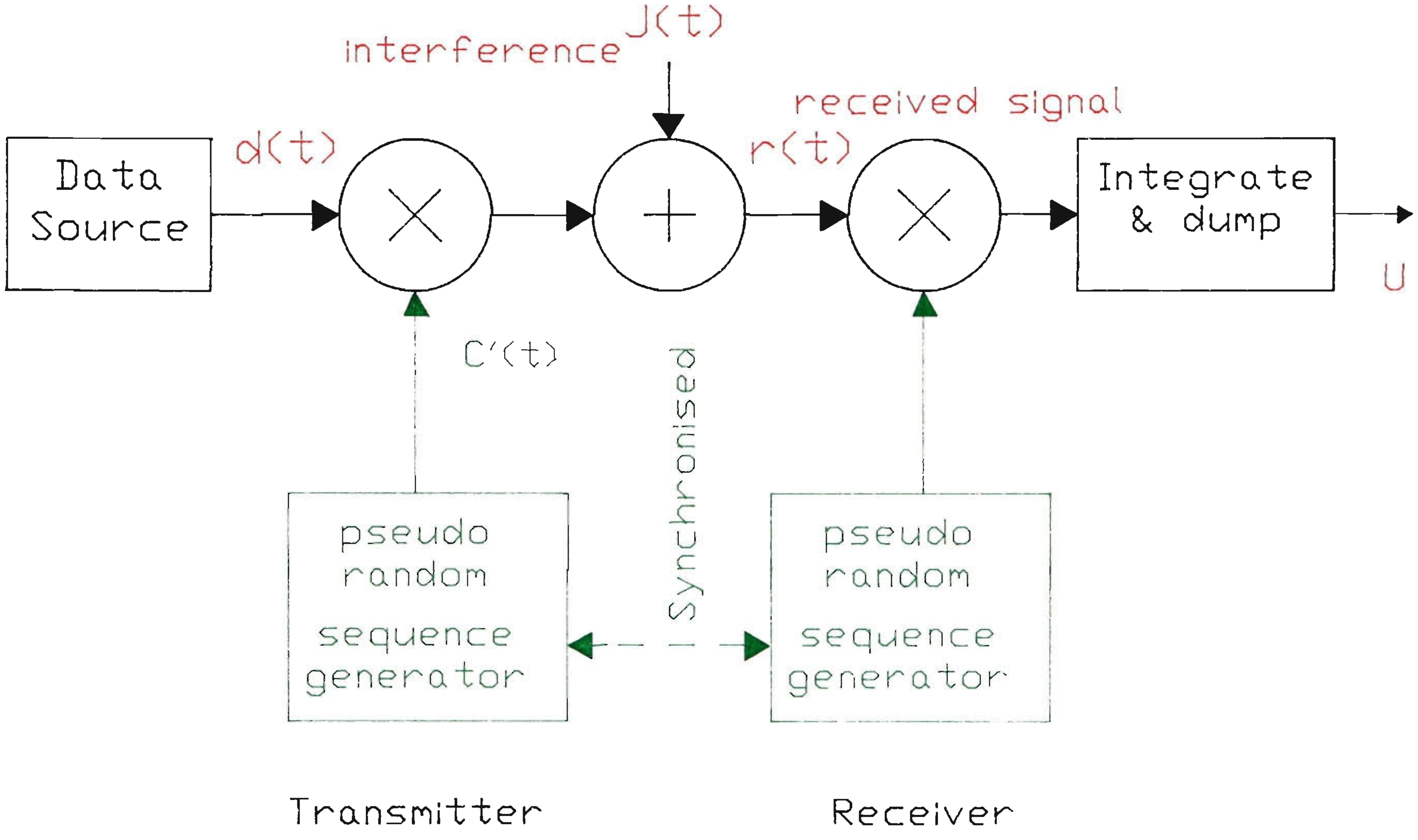


Figure 2 Single Channel Direct Sequence System

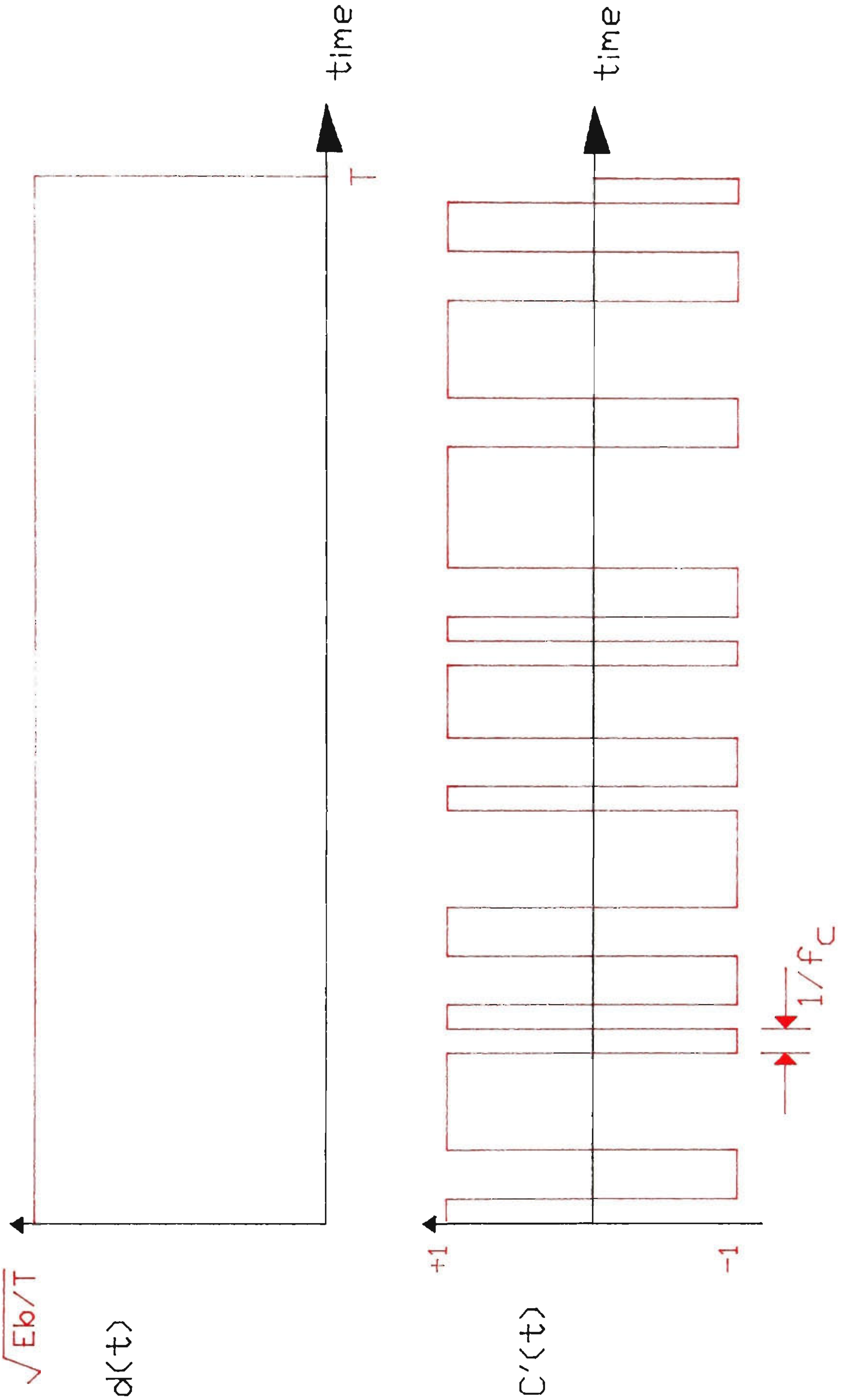


Figure 3 Relationship Between Data and Chip Signals

This one dimensional data bit is multiplied by a two level ± 1 chip sequence, **C'(t)**, running at a frequency of f_c chips/sec, ie. a total of f_cT chips per data bit. The resultant sequence **d(t)C'(t)** is then a f_cT dimensional signal. The relationship between the data and chip signals is shown in Figure 3

The ratio of the dimensionality of the chip sequence to the data is called the "**Processing Gain**", and is defined as :-

$$\text{Processing Gain} = G_p = \frac{B_{SS}}{B_D} \qquad (9)$$

Where B_{SS} = Bandwidth of the SS Signal and B_D = Bandwidth of the Data Signal

The numerical value of the processing gain is an indication of a power improvement factor, which a receiver, possessing a copy of the transmitter's chip sequence, can achieve by a process of correlation. The processing gain gives an indication of the interference rejection capability [5].

If the "chipped data" signal, **d(t)C'(t)**, is transmitted in the presence of an interfering signal **J(t)**, the received signal would be

$$r(t) = d(t)C'(t) + J(t) \qquad 0 \leq t \leq T \qquad (10)$$

At the receiver the correlation process is carried out by multiplying the received signal, **r(t)**, with the same chip sequence used to encode the transmitted data, and then integrating the result over a period **T** to produce a decision variable **U**, where

$$U = \sqrt{(E_b/T)} \int_0^T r(t)C'(t)\ dt \tag{11}$$

From equation (11) it can be determined whether $+\sqrt{(E_b/T)}$ or $-\sqrt{(E_b/T)}$ was sent, depending on whether **U** is positive or negative. The integrand in equation (11) may then be expanded as follows

$$r(t)C(t) = d(t)C'^2(t) + J(t)C'(t) \tag{12}$$

And, since **C'(t)** $= \pm$ **1** ($\therefore$ **C'^2(t)** $=$ **1**), the above equation may be rewritten as

$$r(t)C'(t) = d(t) + J(t)C'(t) \tag{13}$$

With suitable filtering the high frequency component **J(t)C'(t)** may be removed and the original data signal, **d(t)**, recovered.

This fact can be illustrated by inspection of the the spectral occupancy of the signals shown in Figure 4, where the power spectral densities of the data signal, **d(t)**, and the spread data signal, **d(t)C'(t)**, are sketched. The data signal may be extracted by using a filter with a bandwidth of f_d Hz. The fraction of power due to the interfering signal which can pass through this filter is then **1/f_cT**. Thus the data can have an effective power advantage over the interfering signal of **n** $=$ f_cT, i.e. the processing gain, defined in equation (9) above.

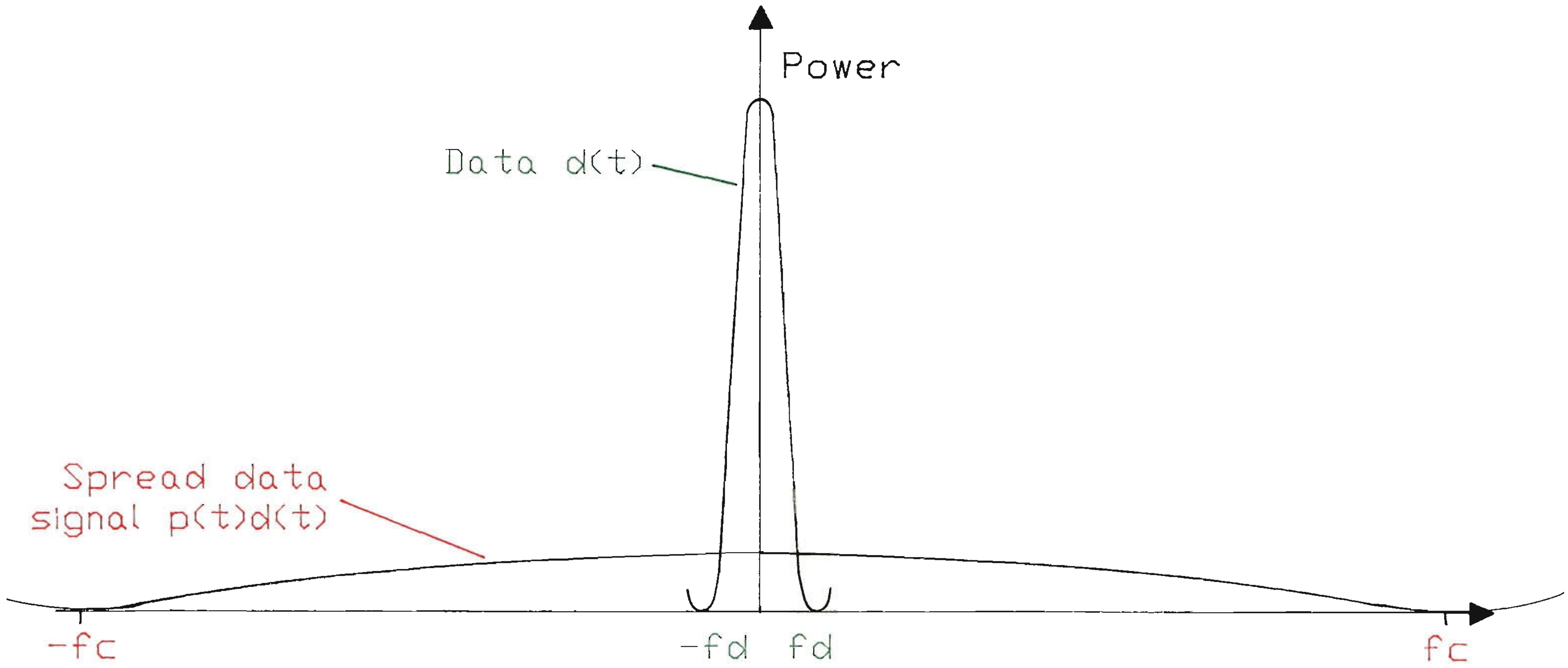


Figure 4 Spectral Occupancy of Signals

In equation (13) the jamming signal **J(t)** may be an external interference signal, eg. nearby high voltage equipment, and/or it may be interference from other transmitting nodes. The rejection of the other nodes transmissions as noise depends upon the orthogonality of the codes [20] used for spreading each node's data.

1.9 Multiple Access in a Spread Spectrum System

The concept of simultaneous multi-access is illustrated in Figure 5. This diagram shows a number of transmitting nodes (**n**) and a single receiver (node **#x**) (where $\mathbf{1} \le \mathbf{x} \le \mathbf{n}$). At each transmitter the data bit, $\mathbf{d_n(t)}$, is modulated with an address code, $\mathbf{C_n'(t)}$, to form the output, $\mathbf{s_n(t)}$, such that

$$s_n(t) = C_n'(t)\, d_n(t) \tag{14}$$

The signal from each transmitter is then "**added into**" the transmission medium as illustrated in Figure 6, to produce a total transmission signal **T(t)** of the form

$$T(t) \quad = s_1(t) + s_2(t) + s_3(t) + \ldots\ldots\ s_n(t) \tag{15}$$

$$= \quad \sum_{1}^{n} s_n(t) \tag{16}$$

The transmission medium may be subjected to external noise or interference, **J(t)**, which would then result in a received signal of the form

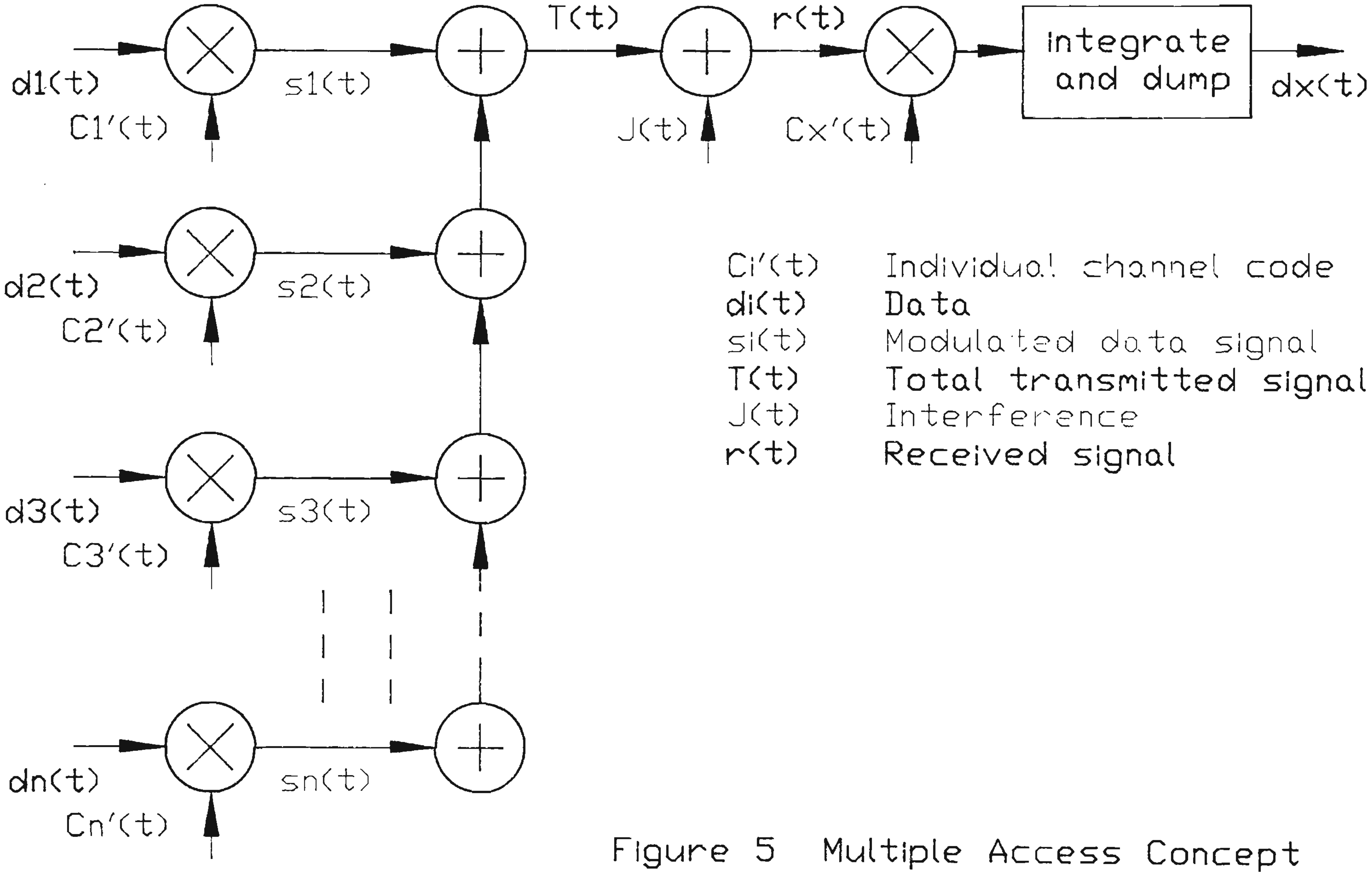


Figure 5 Multiple Access Concept

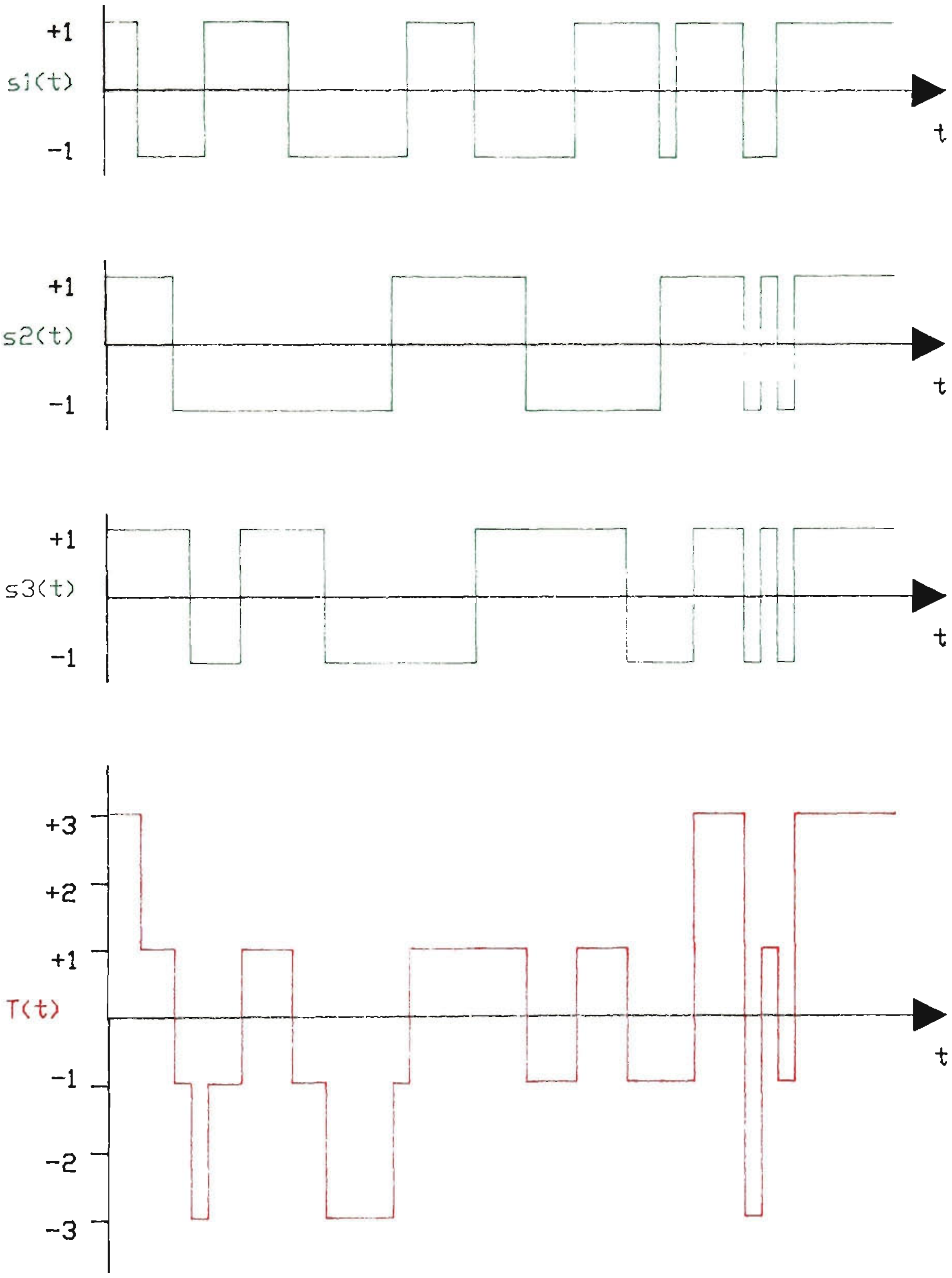


Figure 6 Typical Transmitter Waveforms

$$r(t) = T(t) + J(t) \qquad (17)$$

At the receiver the received signal, **r(t)**, is demodulated by multiplying with the address code for that node, **C_x'(t)**, and, from equation (11)

$$r(t)C_x'(t) = d_x(t) + \sum_{1}^{x-1} s_n(t)C_x'(t) + \sum_{x+1}^{n} s_n(t)C_x'(t) + J(t)C_x'(t) \qquad (18)$$

By suitable filtering the data signal, **d_x(t)**, may be extracted from all of the other components indicated in equation (18). In this manner any receiver may extract its own data from all of the other signals which have been simultaneously combined onto the network, ie. the receiver may access its own data by using CDMA.

2 OUTLINE OF THE RESEARCH

2.1 Summary

Chapter 2 outlines the aims of this research project and the the breakdown of the overall task into smaller packages with identifiable milestones. A brief description of the potential advantages of SS technology in a network situation is then given. Finally, the system's network specifications are detailed.

2.2 Design Aims

The primary aim of this research is to produce a LAN using SS techniques, and to evaluate some of the performance parameters of the system produced. The project should use either twisted pair or parallel pair cables as the interconnection medium in order to maintain the simplicity of the design. The overall network should use standard, off the shelf, components wherever possible.

In order to satisfy these requirements the research was divided into separate sections :-

- Selection of appropriate coding sequences.

- Development of timing, synchronisation and decoding philosophies for these sequences.

- Transmission of a synchronising sequence, and subsequent clock recovery.

- Development of a driving technique suitable to ensure that the connection or disconnection of a station (or a number of stations) would not substantially affect the electrical characteristics of the network.

- Design of data recovery hardware.

- Design of microprocessor control circuit for the system.

- Investigation into the hardware and software protocol structure to fulfil the requirements of the lower levels of the International Standards Organisation (ISO) seven layer model.

- Demonstration of concurrent transmissions.

2.3 Spread Spectrum vs Current Networks

Current bus and ring LAN topologies typified by Ethernet and Cambridge ring [21,22] respectively employ some form of (asynchronous) time division multiplexing. This may be in the (more or less) fixed compartment format of Cambridge Ring or the "packet message" type format of Ethernet. Whichever system is used, any node wishing to access the network must wait until a time when no other users are transmitting. The node will then transmit at a high frequency in the order of 10 MBits/sec into the system. The overall effect is that each node emits bursts of data at high bit rates for relatively short periods, and then remains dormant for a much longer period.

Special chip sets (e.g. Intel 82586), capable of this high speed "**bursty**" type of transmission have been developed thus easing the designer's problem but still leaving a requirement for high speed storage to allow the node processor time to assimilate the data. In systems such as these, even nodes requiring very modest data rates of a few bits per second would need to be able to accommodate these high speed transmissions.

In a SS system, however, the node would only need to be able to cater for a data rate much more modest than the 10 M bits/sec of Ethernet. This reduces the need for specialised high speed or customised devices at each node and increases the use of standard, commercially available, components [23]. This may well allow a LAN to be used in areas, hitherto considered unsuitable, because the cost component outweighed the possible advantages.

2.4 System Specifications

The primary specifications of the system are as follows :-

- The system will use a bus topology as shown in Figure 7, with separate Data/Network and Synchronisation lines.

- The node interconnections will be via twisted pair or parallel pair wires for simplicity and ease of connection/disconnection. The cable is terminated with an appropriate resistance at both ends in order to minimise mismatches and reflections. Using this mechanism means that connections may be made at any point in the system with "**Tap-in**" connectors and not only at discrete intervals as would be the case with a fibre optic system (requiring optical couplers) or a co- axial system (requiring "**T**" pieces). It means, furthermore, that in order to accommodate a new node there would be no need to break the line to insert an optical coupler or "**T**" piece if a conveniently situated connection point was not already available.

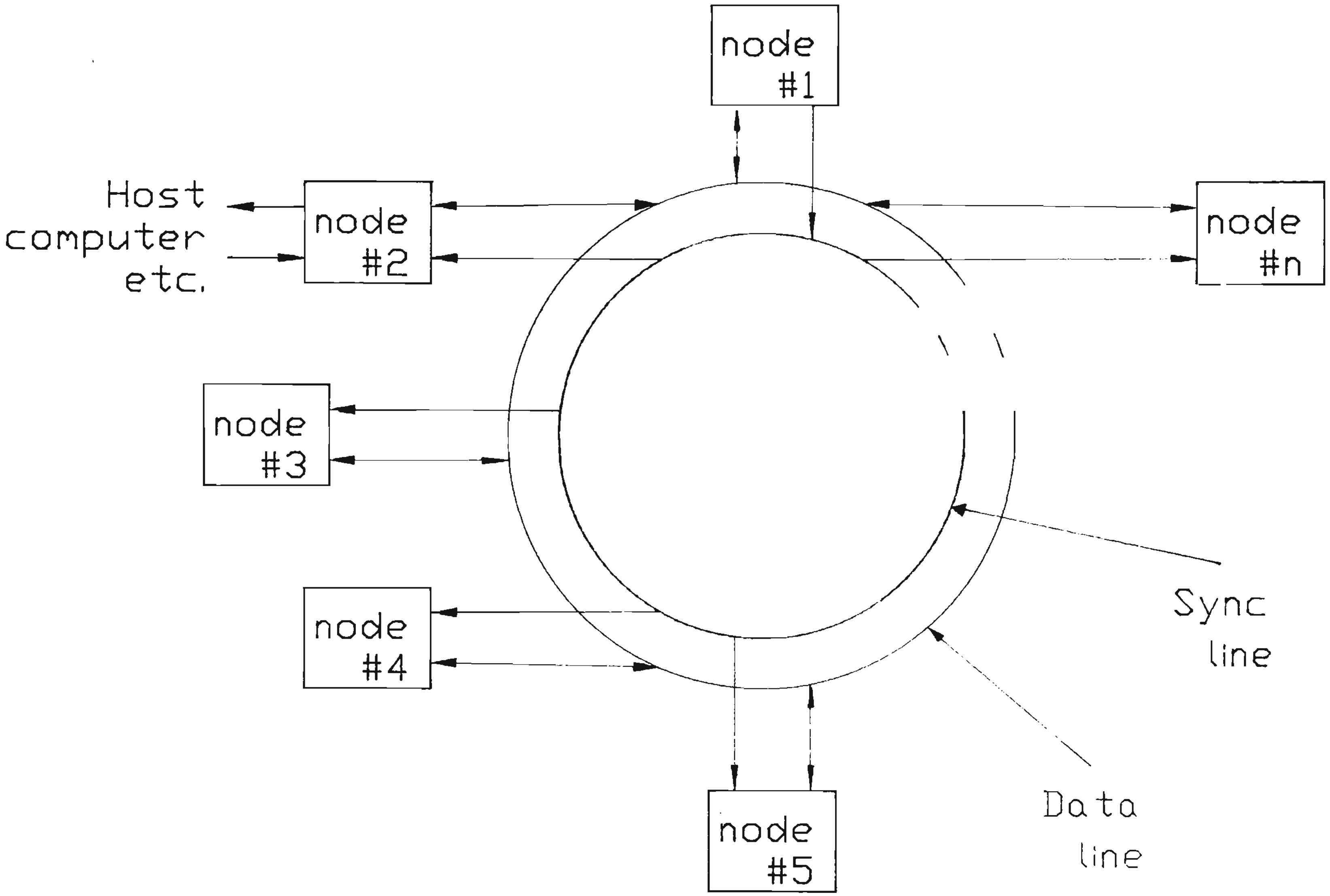


Figure 7 Network Bus Structure

- High impedance current drivers will be used to interface onto the network line, thus ensuring minimum perturbation to overall network traffic when nodes connect or disconnect.

- These drivers are not necessary for the synchronisation line. One node (eg. node **#1**) acts as the sync transmitter and all other nodes use the transmission on the sync line to recover clock and code synchronisation information. The overall synchronisation means that all data may be transmitted in a synchronous, rather than an asynchronous, format. This has significant implications in the choice of code allocation scheme and in the implementation of the system timing and data recovery sections (as shown in chapters 3 and 4).

- The system will use the point to point communication protocol. Each node will be allocated its own chip sequence, or spreading code, which will act as an address for that node; thus if node **#1** wishes to transmit to node **#2** then node **#1** will modulate its data with the chip sequence for node **#2**. Hence the transmitter of each node must be able to modulate its data with any recipient's address code. In this manner simultaneously multi-pair transmissions are allowed with transmission channels being identified by the codes being used to modulate the data, ie. CDMA. It is obvious, however, that no two nodes may simultaneously transmit to the same recipient, otherwise a collision would occur. The multiple access features reduce the probability of collisions occurring but do not eliminate them totally.

- A Carrier Sense Multiple Access with Collision Detection, (**CSMA/CD**), type of protocol will be used to detect and resolve intra-channel collisions. Each transmitter will have a built-in demodulator to pre-check for traffic in the desired channel and to continuously check for collisions whilst transmitting. Each node, therefore, requires two demodulators - one for monitoring the channel for its own received messages and the other for collision avoidance and detection. The first demodulator works exclusively with its own address code whereas the code used in the second is selectable. Further to the code allocation scheme, it was decided that the code length (the number of chip bits per data bit, ie. the processing gain, **Gp**) should be in the order of 1000. This number was chosen because there is a rule of thumb which indicates that the number of potential users is approximately 10% of the processing gain [5]. This would allow a system possibly capable of supporting up to 100 users.

3 CODES AND SYSTEM TIMING

3.1 Summary

Chapter 3 outlines the properties required of the codes in a CDMA system. A brief description of each coding scheme considered is given. This is followed by a description of the important timing characteristics of the system.

3.2 Coding Scheme Selection

Several possible generation and allocation schemes were considered before one was finally chosen. The properties of major concern were :-

- Simplicity of generation and synchronisation.
- Crosscorrelation between members of the code set.

The first three schemes considered (schemes A, B and C) were designed to allow the network to be operated asynchronously, ie. the data bits from each user do not have to coincide. The other two schemes considered (schemes D and E) were designed for synchronous system operation, where data bits from every user have a fixed start time.

3.2.1 Scheme A - Segments of Long m-sequences

In this scheme the concept was to allocate to each node a segment of **M** bits of a much longer code of length **N**. Every node would generate the same long MLS, each node would then be identified by the segment of this long code allocated to it. The crosscorrelation between these segments transmitted simultaneously would determine the interference between one transmission and another.

Lindholm [24] found, in his consideration of the moments of the distribution of weights of M-tuples of MLSs, that this type of system gave more predictable crosscorrelation values as **M** tends towards **N**. This figures would, in the limit, result in the full code length **N** being used and in fact producing the autocorrelation function of the MLS. For this reason this scheme was not considered further.

3.2.2 Scheme B - Different m-sequences of the Same Length

This scheme considered the use of different m-sequences. Each node would be allocated a series of tap settings which would be used to produce a different m-sequence. Each m-sequence would be of identical length since shift registers of the same length would be used in each node. Producing m-sequences in this manner is a relatively straight-forward procedure and tables of polynomials for the production of such sequences are readily available [19]. Some sequences do, however, require many taps to produce, and since each transmitter must be able to transmit to any receiver, this implies that each node must have sophisticated hardware in order to produce any code allocated.

The crosscorrelation between sequences produced by Linear Feedback Shift Registers (LFSRs) of the same length but with different tappings was considered. A problem arises with this scheme in that there are only 60 m-sequences of length 1023 which can be produced. Within this 60 limit the maximum number in any **Maximally Connected Set** * is only 3 indicating that the crosscorrelation between sequences could be as high as 38% [20], unless extreme care is exercised in the selection of sequences allocated. The number of codes available for allocation under this scheme would, therefore, restrict the number of users to 60. This, coupled with the level of correlation interference indicated above, was considered significant enough to prevent further investigation of this scheme.

***Note** A maximally connected set is one in which all members of the set act as part of a **preferred pair** with any other member of the set [20]. A preferred pair is a pair of m-sequences of period **N = 2^n-1** which has a "**3 valued**" crosscorrelation function where the peak magnitude of the crosscorrelation is given by [27]:-

$$|R(t)| = \begin{cases} 2^{(n+1)/2} + 1 & \text{for n odd} \\ 2^{(n+2)/2} + 1 & \text{for n even} \qquad n = \text{mod } 4 \end{cases} \qquad (19)$$

For **N = 1023** the maximum crosscorrelation for a preferred pair is **65**, and the three correlation values are **-65**, **-1** and **+63**. (Note the peak value **65** is given by **$2^{(10+2)/2}+1$**).

3.2.3 Scheme C - Gold Codes

The third scheme considered the use of GOLD codes allocated to each node. Gold codes are a class of periodic sequences which provide larger sets of sequences with predictable crosscorrelation properties [20,25]. A set of Gold sequences of period **N = 2^n-1** consists of **N+2** sequences. The adoption of this scheme would, however, require each sequence generator to have a 20 bit shift register, to generate the 1023 bit codes. Two such registers would be required to cater for the fixed receiver and the switchable transmitter. The transmitter stage would also require a method of pre-setting the LSFR with the required seed in order to generate the particular Gold code required. The correlation between members of the Gold Code set is the same as that between the preferred pairs of m-sequences used to generate them [20] (see also equation (19) above). Although there would be sufficient members of the Gold Code set, the correlation values offer no improvement over the scheme considered above.

3.2.4 Scheme D - Phase Shifted Versions of a Single m-sequence

Taking advantage of the fact that the whole system could be synchronised means that a code allocation scheme, using MLSs, could be devised such that the codes allocated would not be produced by different shift registers, rather that a single m-sequence would be used throughout and each node would be allocated a **phase** or **bit shifted** version of the same sequence. The crosscorrelation between codes of each node would then become the autocorrelation of the single code. Codes allocated on this phase shifted basis would have the following format:-

$$C_{n-1}'_{k} = C_{n}'_{k-s} \tag{20}$$

n indicates the **n**th code and **s** is the separation factor, ie. the phase shift between successive codes.

Using separate synchronisation and data lines means that the synchronisation overhead could be kept to a minimum by using phase zero of the network's m-sequence as the synchronisation pattern. Locking to the sync code automatically synchronises the address codes which are phase shifted versions of the same sequence.

With this scheme, however, each user injects approximately 0.1% error in the correlation process of every other user. This may be explained by inspection of the autocorrelation function of a m-sequence as shown in Figure 8 and by consideration of the autocorrelation value of a two level m-sequence, **Cx'**, which, from equation (6), is given by [5]

$$|R_{Cx'}(t)| = \begin{cases} N & \tau = 0, N, 2N, \\ -1 & \text{otherwise} \end{cases} \tag{21}$$

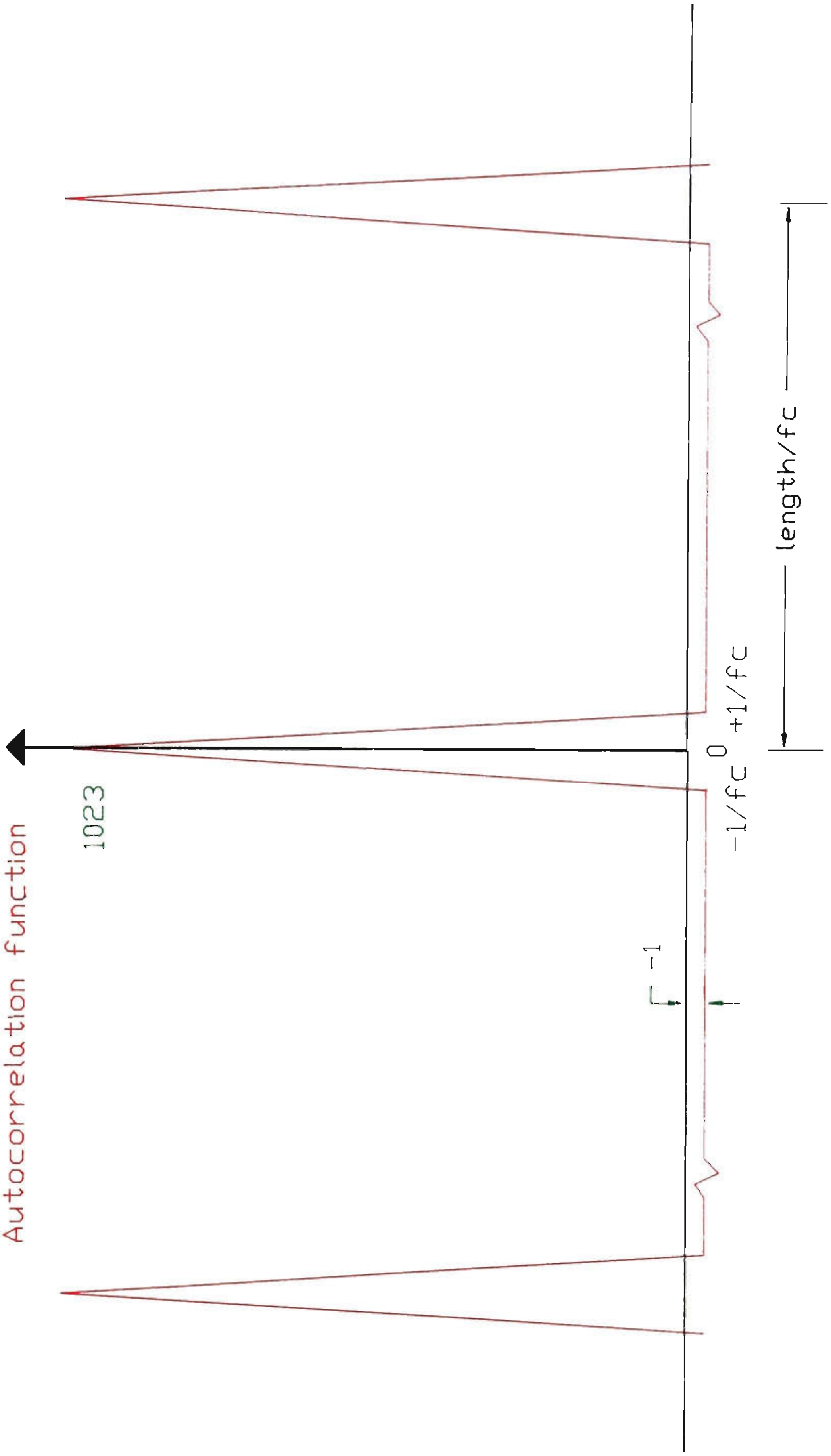


Figure 8 Autocorrelation Function of a 1023 Bit MLS

It can be seen that for zero phase shift the correlation is equivalent to the length of the code (i.e. 1023 in this case) and that for all other phase shift positions the correlation is -1. This residual -1 effect from every code being simultaneously transmitted causes this decrease in the correlation output at the receiver. It is obviously unsatisfactory to have this progressive level of interference caused by other users, and directly proportional to the number of these other users. These observations are supported by the findings of Hasegawa, Hirosaki and Sawai [16] in their performance analysis of a fibre optics system.

This scheme has two basic drawbacks, firstly the cross channel interference inherent in m-sequences and secondly the fact that the code length is an odd number, 1023. A more suitable code would be a binary multiple in length, generators and correlators etc. could then use binary counters directly, and would have ideal "**two valued**" autocorrelation values of **0** and **N**, where **N = 1024** in this case. This being the case it was decided to attempt to modify the m-sequences to eliminate this residual interference effect inherent in the m-sequences.

3.2.5 Scheme E - Modified Maximal Length Sequences

This scheme uses "**Modified**" **Maximal Length Sequences**. Each node is allocated one of these Modified MLSs, which is essentially a MLS cyclically bit shifted within its **1023** bit length, refer also to Figure 1b, with an additional binary zero transmitted as bit **1024**. This address code generation concept is illustrated in Figure 10. Because of this extra logic zero all codes produce an additional correlation factor of "**+ 1**" when compared to the m-sequence value shown in equation (21). This modification to the MLS produces a code with the ideal **1024, 0** crosscorrelation characteristics illustrated in Figure 9. The correlation values produced by code members of this system will then be given by :-

$$|R_m Cx'(t)| = \begin{cases} N+1 & \tau = 0, N, 2N, \\ 0 & \text{otherwise} \end{cases} \qquad (22)$$

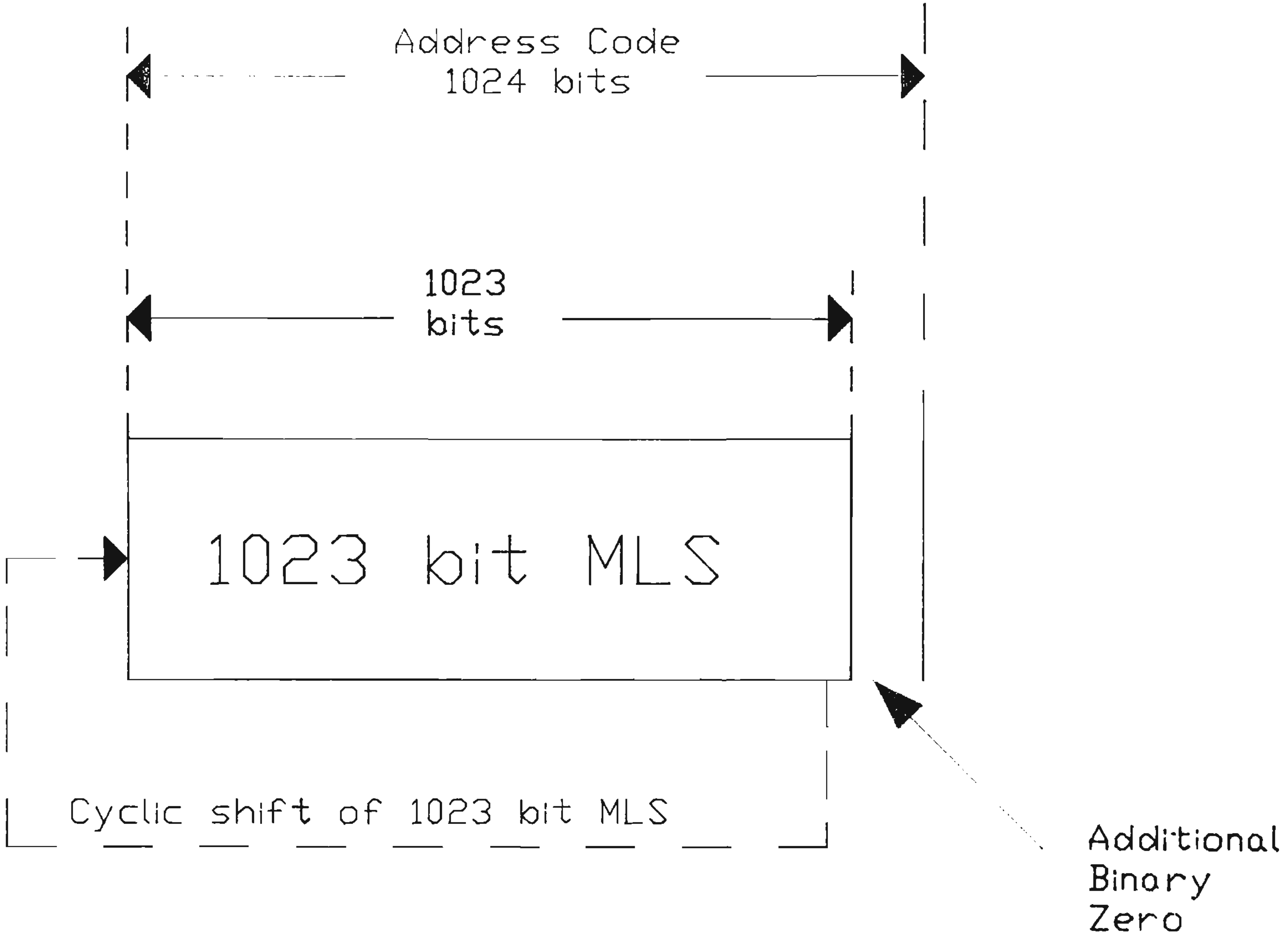


Figure 9 "Modified" MLS Address Code Generation

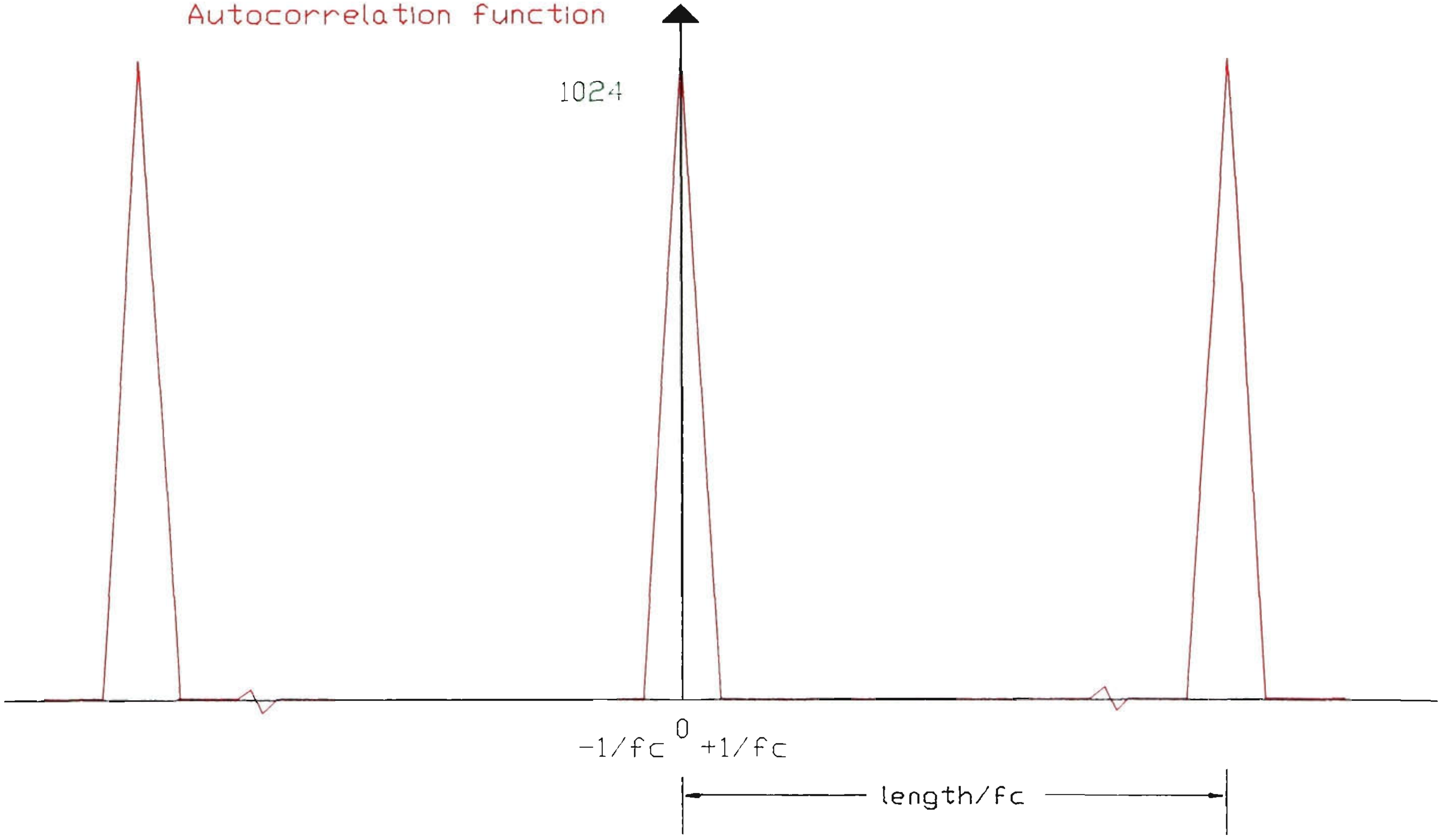


Figure 10 Autocorrelation Function of "Modified" 1024 Bit MLS

3.3 System Timing Concepts

This section gives a description of the important timing characteristics of the system. Physical network dimensions and delay features are also considered. Since the code allocation scheme is so intimately involved with system timing, those aspects of timing important to code allocation are also described. Timing and code allocation are vital to data recovery, and for this reason the recovery of data from transmissions subjected to delays is also considered.

As previously noted, this system allows a node to be connected at any point along the interconnection medium. Node **#1** is the station transmitting the synchronisation code (ie. phase zero of the system's "Modified" MLS) onto the sync line, and this node is connected in the centre of both the Data and Sync transmission lines. The overall configuration may be represented as shown in Figure 11. The synchronisation information travels in both directions, along the Sync line, towards nodes **#2** and **#5**, which are assumed to be at the ends of the transmission lines. Both transmission lines are terminated in an appropriate resistance to minimise reflections. The clock timing lines in Figure 11 show an example of possible system delays where the Sync signal takes 1 bit period to reach nodes **#2** and **#5**. At a Chip rate of 1.2288 MHz (ie. 1200 Hz data rate multiplied by the 1024 processing gain) this 1 bit delay represents one **Chip Bit** clock period, **Tc** = 813.8 ns. This represents a distance of 244 metres from node **#1** to node **#2** or node **#5**, or a total distance from node **#2** to node **#5** of 488 metres.

Since node **#1** is the synchronisation transmitter it represents the system time reference zero ($\mathbf{T}_{ref}$) and it is this time reference which is used to to generate the synchronisation code ($\mathbf{C}_{ref'k}$), which is the zero phase of the system coding sequence. This signal takes 1 bit period to arrive at node **#2**. Node **#2** locks onto this as a reference to absolute system time zero. Node **#2** has therefore, a time reference, $\mathbf{T}_{ref-1}$, or 1 bit delayed from the system norm. Node **#5** is, similarly, 1 bit delayed, and nodes **#3** and **#4** are each 1/2 bit delayed.

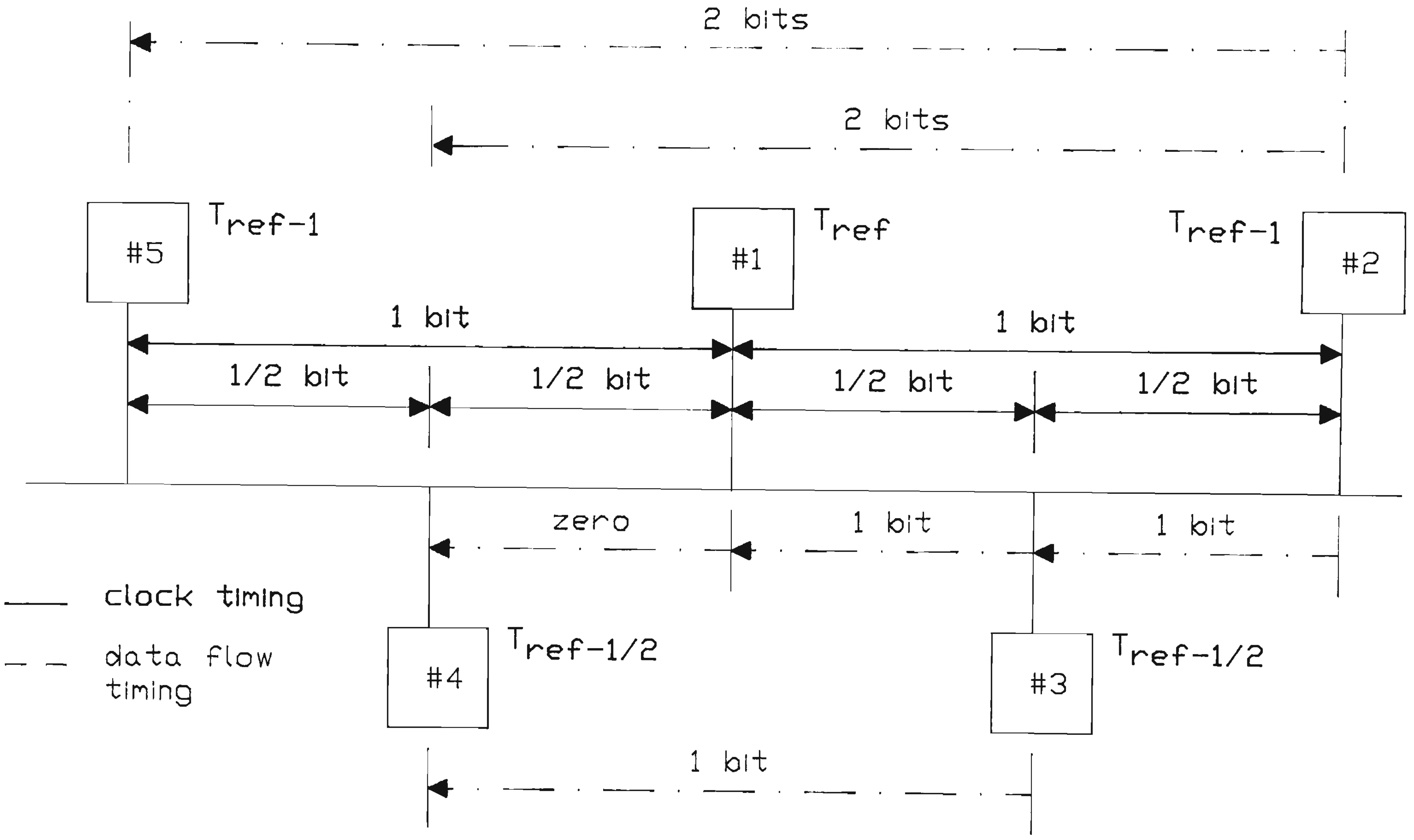


Figure 11 System Delay and Timing Aspects

If node **#1** transmits a message to node **#2** using a code which has a separation factor, **s**, of **2** (ie. $C_{ref}'_{k-2}$) it will arrive 1 bit delayed at node **#2** (ie. it should look like $C_{ref}'_{k-3}$), but since node **#2** has a relative time reference which is also 1 bit delayed the two delays will cancel and the transmission will appear to node **#2** to be using code $C_{ref}'_{k-2}$. Thus, the code which appears as $C_{ref}'_{k-3}$ with respect to system time reference, T_{ref} will appear as $C_{ref}'_{k-2}$ with respect to node **#2**'s time reference of T_{ref-1}.

If, on the other hand, node **#2** transmits to node **#1**, the timing reference of node **#2** (T_{ref-1}) used to select the transmission address code will mean that any address code selected (eg. one subjected to a separation factor, **s**, of **x**, which produces code $C_{ref}'_{k-x}$) will appear 1 bit delayed (as $C_{ref}'_{k-(x+1)}$) with respect to node**#1**'s time reference, T_{ref}. The transmission from node **#2** will also be subject to a propagation delay of 1 bit before reaching node **#1**. The transmission would, therefore, appear to be using code $C_{ref}'_{k-(x+2)}$ with respect to node **#1**'s time reference of T_{ref}, ie. the transmission code appears to have a separation factor of **x + 2** as far as node **#1** is concerned. In this case the time reference delay is added to the data propagation delay to produce this effective delay of **2** bits. Table 1 lists examples from Figure 11 to explain the situation. In this table column **A** is the transmitting node, column **B** is the receiving node and column **C** is the total delay as seen by the receiver. This delay is made up of timing reference delay plus data propagation delay. The calculation of the effective delay for the first two rows in this table, ie. node **#1** to node **#2** and node **#2** to node **#1**, is shown above. These delays are with reference to the receiver's own timing reference.

Table 1 System Timing and Delays

From A	To B	Delay (Bits) as seen by receiver C
1	2	Zero
2	1	2
2	5	2
3	2	Zero
2	3	1
1	4	Zero
3	1	1
3	4	1
2	4	2

It can be seen that the relative delay is dependent on the respective positions of the nodes in the system. One way of overcoming the difficulties posed by these position dependent delays would be to allocate fixed positions to each node and measure all time delays etc. and then provide a look-up table to each node so that suitable compensation could be made. This solution detracts from the flexibility of the system, but could be used to match the transmitting node's ID with the delay characteristics in the look-up table in order to provide a means of verifying the transmitter's identity. This would be an attractive proposition if security was a major requirement of the system. Another solution is to adopt a code allocation scheme such that codes allocated to each node are selected using a separation factor, **s**, which is greater than the maximum accumulation of delays in the system.

ie $\mathbf{s} >$ maximum system delay

Since the maximum system delay is dependent on the length of the transmission lines used, ie. between node **#2** and node **#5** in the example shown in Figure 11, the maximum delay can be calculated when the code allocations are being made. In this way no valid code, no matter what delays it is subjected to, could possibly mimic any other valid code.

3.4 Sliding Correlator

The fact that transmissions are subject to different propagation delays implies that the receiver must be able to detect its own messages at any point in a time window, with delays up to the maximum propagation delay in the system. With the two bit delay used in the example above, it means that a receiver expecting to demodulate a transmission using code $\mathbf{Cx'_k}$ must be able to demodulate in the range $\mathbf{Cx'_k}$ to $\mathbf{Cx'_{k-2}}$. To this end each receiver incorporates a "**Sliding Correlator**" with which it can scan across this 2 bit window looking for its own data. If no such transmission is present the correlation should be zero across the whole 2 bit window and the search process can begin again. If a valid transmission is present the receiver should be able to determine the delay at which the correlation peaks and use this delay factor throughout the transmission, as the delay will remain constant.

Up to this point all aspects of code allocation and modulation/demodulation had been considered in the light of whole bit delays between codes. This is, obviously, impractical in a real system as the delay is node placement dependent, and the nodes may be positioned at any point along the transmission line. The sliding correlator designed for the system is capable, therefore, of delaying the local correlating code in increments, **v**, of 1/16th of a bit upto a maximum of 2 bits. The local correlating code thus takes the form:-

$$Cx'_{k-v} \tag{23}$$

If the received and local codes are separated by more than 1 full bit the correaltion will be 0. As the delay between the correlating codes reduces the correlation will increase uniformly towards a peak of 1024, as shown in Figure 10.

3.5 Collision Mechanisms and Characteristics

Having seen that the output of the integrator is dependent on the relative partial chip bit overlap of the transmitter and the receiver, it is important to consider the possible situation surrounding collisions. A collision would occur when two transmitters simultaneously use the same destination address code. For positively colliding transmissions the effect would be to simulate a received signal where the energy of the data bit being encoded varies between $\sqrt{(E_b/T)}$ and $\sqrt{(2E_b/T)}$, depending on the relative overlap of the two signals with the local correlating code. The result will be to produce a decision variable with a value between **U** and **2U**, refer to equation (11). For negatively colliding signals, ie. subtractive interference, the value of the decision variable will be in the range **0** to **U**. For collisions involving more than two transmissions, the value of the decision variable may be extrapolated from the relative polarities and overlaps of the colliding signals.

The node should, therefore, be capable of making a decision based on the magnitude of the actual final value of the decision variable, and not just on the fact that a prescribed threshold has been reached. The shape of the correlator output (ie. the value of the decision variable) will not follow the characteristic inverted cone, illustrated in Figure 10, across the 2 bit sliding correlator window when colliding signals are present. The node should, therefore, be capable of using this information when deciding whether a collision has occurred.

4 HARDWARE

4.1 Summary

This chapter describes the hardware which has been developed for the project. The description of the hardware will commence with a functional description of the major building blocks (and their interactions) which make up the complete circuit. This will be followed by a description of the individual circuits in block diagram format, before each circuit is described in detail. The report is designed in this manner in order that an overall understanding of the operation of the hardware may be obtained, unobscured by the detail necessary to describe some of the more unique aspects of the circuits designed.

4.2 Overview of Development Requirements

It was decided to produce 3 system nodes. This number allows the simultaneous communication and interference rejection concepts to be adequately demonstrated, thus satisfying the design aims of the project.The system is housed in a modular 19 inch rack. A printed circuit backplane, designed for 64 way indirect edge connectors, is connected to the lower connector slot of the two available. Double or single Eurocard boards may be used in this system. The indirect edge connectors used are "**a**" and "**c**" contact 2 x 32 way units. During the development stages individual circuits were wire wrapped onto single or double Eurocards. Double Eurocard sized printed circuits were designed for sub sections which had been developed and tested. Each node of the mature system is made up of two printed circuit boards of double Eurocard size. For development purposes up to four complete nodes can be housed in the rack.

Power requirements for the whole development system are as follows :-

- +5 volts at 4 amps, ±12 volts at 200 ma., and -5 volts at 100 ma.

The need to design modules to be able to be used with different node addresses, and to be controlled via the same backplane has resulted in the need for some degree of flexibility in backplane pin allocation. Individually wire wrapping the upper connectors allowed this flexibility. This is necessary in the development environment, where several nodes must be accomodated together in the single 19 inch chassis, but is not a feature of a "**production**" node.

4.3 Brief Overview of Hardware

Figure 12a shows a block diagram layout of the circuits necessary to perform all of the functions required in this Spread Spectrum LAN. The individual sections are dispersed between two boards, designated the "**Microprocessor**" and "**Peripheral**" boards, **M** and **P** for short, although the microprocessor board contains peripheral as well as microprocessor components. Each individual circuit is identified as being on board **M** or board **P**. Figures 12b and 12c are photographs of the completed Microprocessor and Peripheral boards respectively. Figure 12a also identifies which board houses each circuit module. Signals common to all nodes are transmitted via a commercial printed circuit backplane on the lower of the two edge connectors, and intra node control takes place via individual wire wrapped connections on the upper edge connector. Complete circuit diagrams of the two circuit boards are included at Appendix A for reference.

4.3.1 Microprocessor Control Circuit - Board M

This is the central control and decision making component in each node unit. This section is responsible for storage of information in transit to and from the network and is also responsible for controlling the implementation of the LAN protocols.

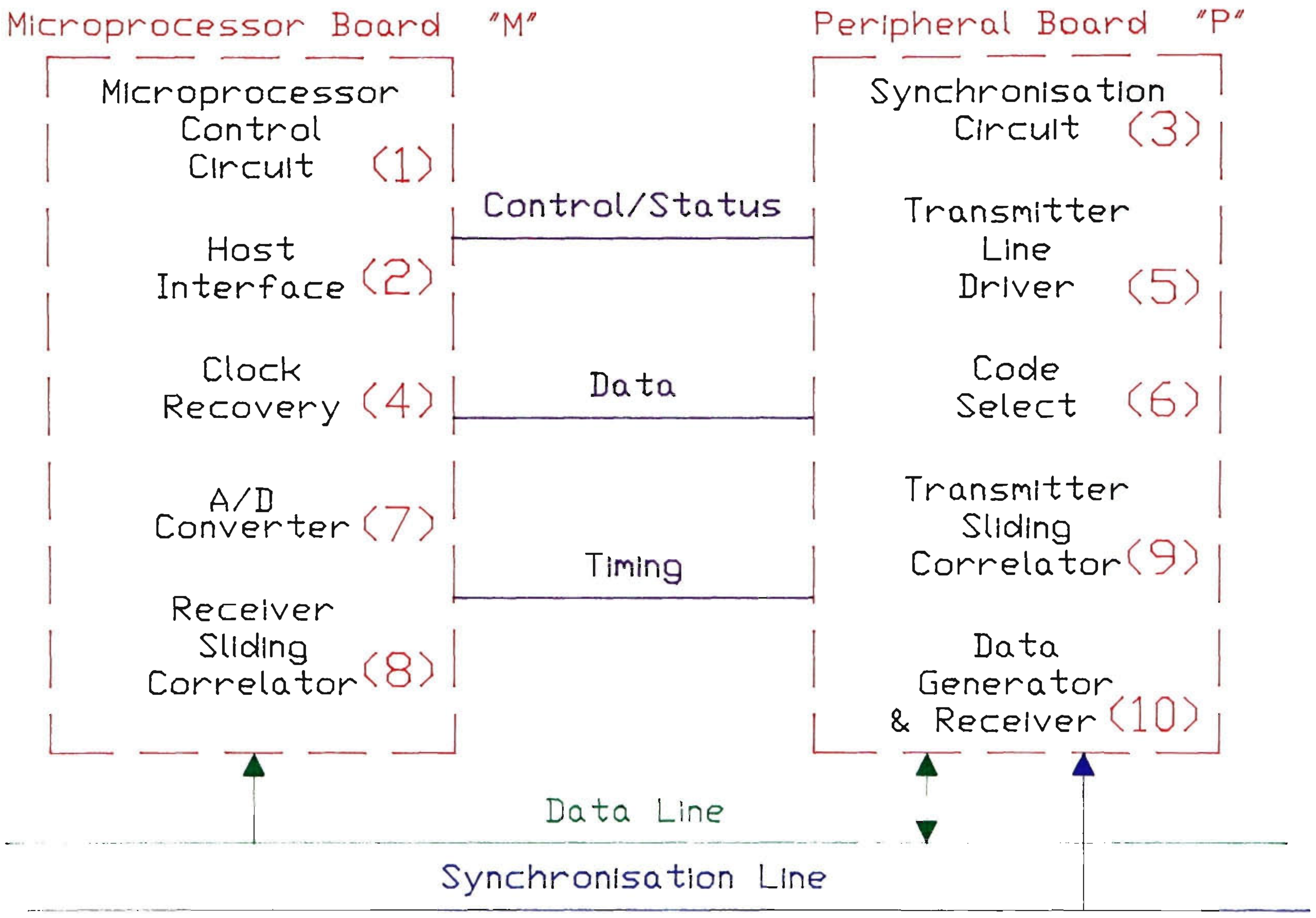

Microprocessor Board "M"
Peripheral Board "P"
Microprocessor Control Circuit (1)
Host Interface (2)
Clock Recovery (4)
A/D Converter (7)
Receiver Sliding Correlator (8)
Synchronisation Circuit (3)
Transmitter Line Driver (5)
Code Select (6)
Transmitter Sliding Correlator (9)
Data Generator & Receiver (10)
Control/Status
Data
Timing
Data Line
Synchronisation Line

Figure 12a Overall System Block Diagram

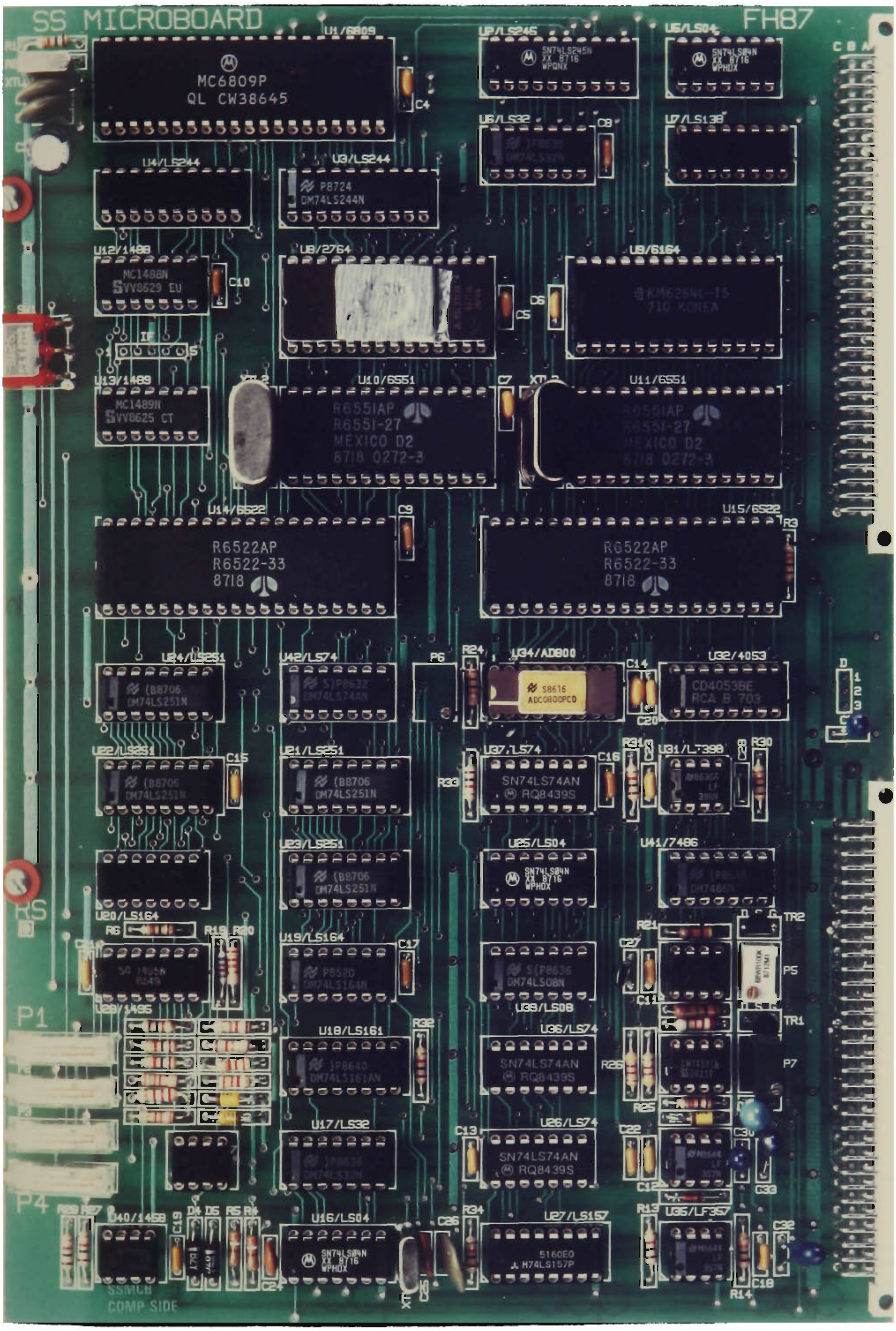


Figure 12b Photograph of Microprocessor Board

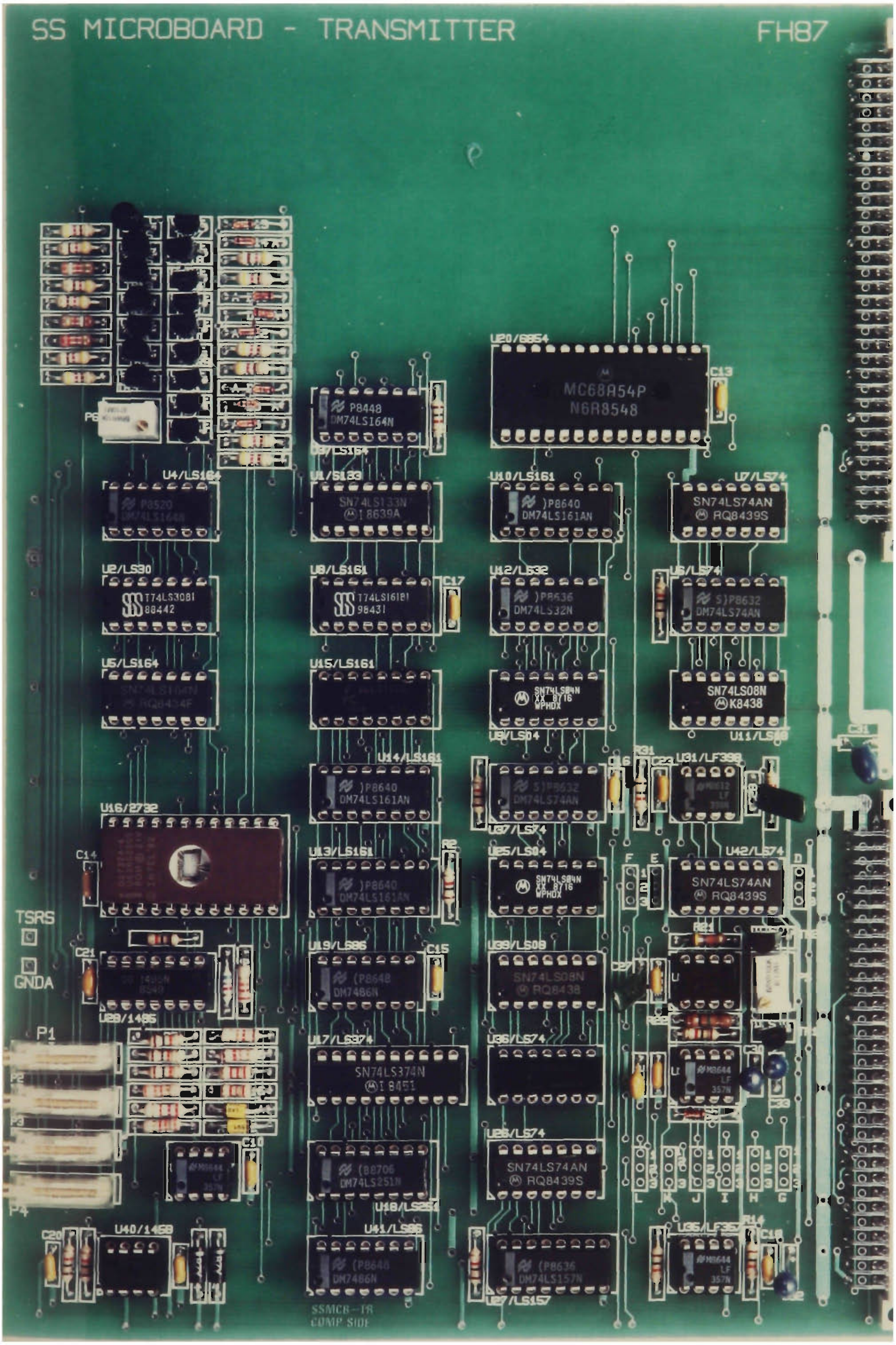


Figure 12c Photograph of Peripheral Board

4.3.2 Host Interface - Board M

The host may be an active bi-directional device such as a computer system, or a relatively passive, mainly receiver device such as a printer unit. The task of this unit is, however, the same; information received from the transmission medium must be transferred to the host device in a manner suitable for that device, and information must be accepted from the host for subsequent transfer, via the spread spectrum network, to some desired receiver.

4.3.3 Synchronisation Circuit - Board P

This circuit maintains node synchronisation from a sync signal (Address Code 0 is used for this purpose) transmitted on the sync line. This synchronisation is essential since reference to time frame 0 is necessary in order to be able to generate different address codes for the transmitter and to detect information encoded with the node's own address code.

4.3.4 Clock Recovery - Board M

The clock recovery circuit uses the signal on the sync line to extract timing information in order to correct any drift in the local clock circuits.

4.3.5 Transmitter Line Driver - Board P

This circuit is the high impedence line driver used to transmit modulated data onto the transmission medium by adding it to the signals already present. A high impedence driver is necessary to ensure isolation for concurrently transmitting nodes.

4.3.6 Code Select - Board P

This circuit selects the address code to be used to modulate the transmitter's data, ie. the circuit selects the destination for each transmission.

4.3.7 A/D Converter - Board M

This section incorporates an A/D converter and an analogue multiplexer, to allow the A/D converter to be used for two separate inputs. The A/D circuit is used to analyse data obtained from scanning across the possible 2 bit delay window by the receiver whilst looking for transmissions intended for this particular node. It is also used by the transmitter to search for transmissions modulated with the intended destination's code, i.e. a channel occupancy check. The control circuit can select which signal is used as input to the A/D. This selection allows the transmitter to monitor its own transmissions and to ensure no collisions occur during transmission.

4.3.8 Receiver Sliding Correlator and Demodulator - Board M

This section allows the node to search, across the allowable 2 bit delay window, for transmissions modulated with the receiver address code, Up to 32 measurements can be made in 1/16th bit increments across the window. Each measurement is the correlation over 1024 chip bits, i.e. the **periodic** correlation. The correlation value is then sampled and held whilst waiting to be used by the A/D converter. The microprocessor control circuit may then make a decision as to the presence or absence of a valid transmission, with or without collisions, as proposed in section 3.5.

4.3.9 Transmitter Sliding Correlator and Demodulator - Board P

This section enables the transmitter to search, across the allowable 2 bit delay window, for existing transmissions modulated with the intended recipient's address code. In this manner a Carrier Sense Multiple Access with Collision Detection (**CSMA/CD**) type of protocol may be adopted to resolve intra channel contentions. The circuit may also be used to monitor traffic during transmissions in order to detect any possible collisions, thus performing the "**CD**" part of the above protocol.

4.3.10 Data Generator/Receiver - Board P

This circuit performs the necessary second level protocol generation such as frame creation, flag insertion, frame check sequence, etc. It is also used to check the received data for these same protocol features.

4.4 Detailed Hardware Description

In this part of the report the individual circuits are described in greater detail. Initially, the functional blocks of each circuit will be explained and then the unique and/or less commonplace parts of the circuit will be examined in more detail. The report is designed in this manner to avoid unnecessary description of standard circuits or techniques.

4.4.1 Microprocessor Control Circuit

The functional block diagram for this circuit is shown in Figure 13. The circuit is based on a Motorola MC 6809 microprocessor operating at a clock frequency of 1MHz. The two asynchronous serial interface devices are included (SY 6551), primarily to support the development exercise as only one will be needed in the mature system. Two parallel input/output devices (SY 6522) are included, giving 32 programmable

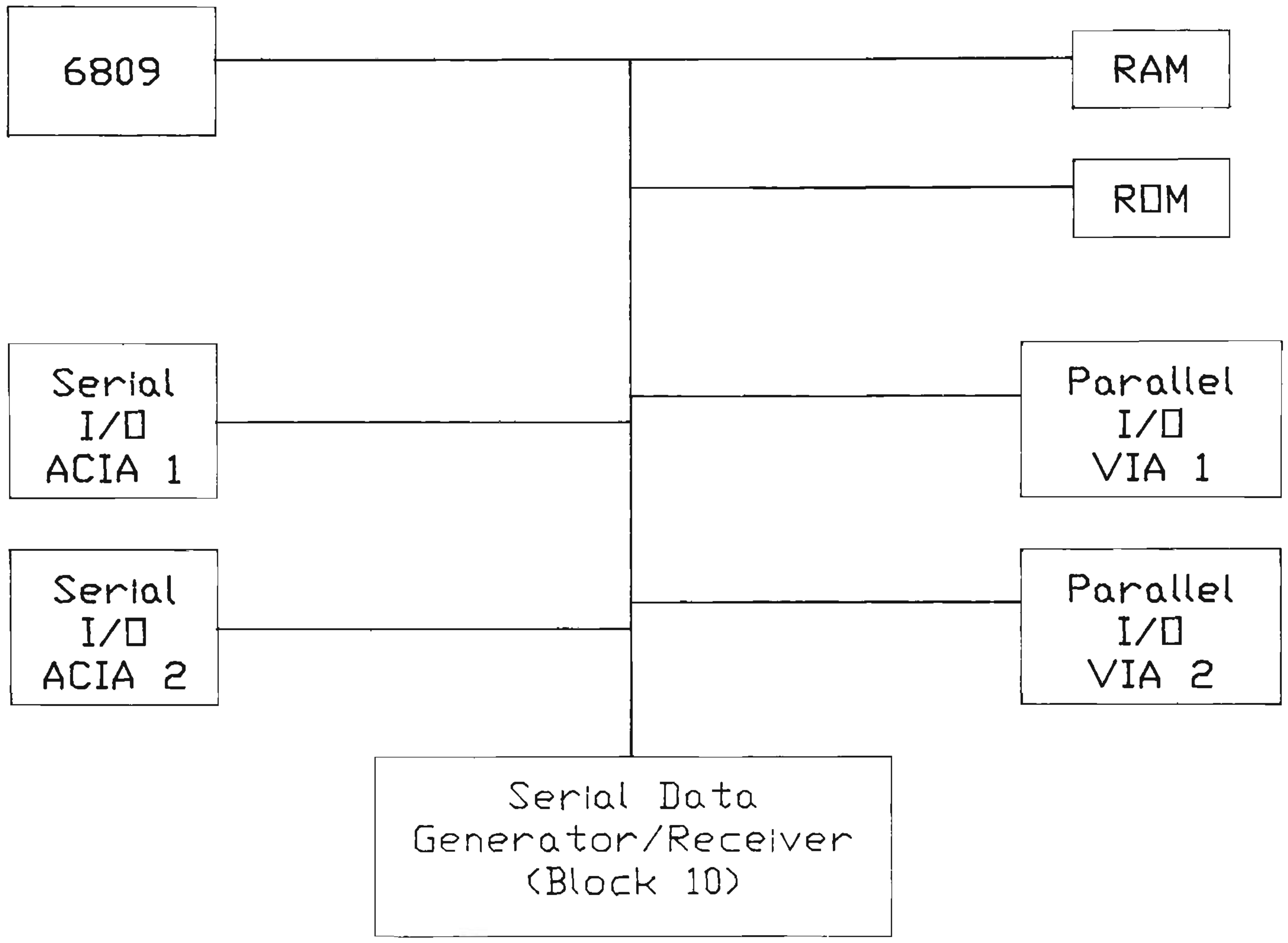


Figure 13 Microprocessor Control Circuit Block Diagram

input/output lines. The circuit is designed with 8K x 8 of EPROM (2764) and 8K x 8 static RAM (6164), which is expendable to 16K x 8. The memory map of the system is included as Figure 14. Figures 13 and 14 also show the Serial Data Generator/Receiver, which uses an Advanced Data Link Contoller (ADLC - MC 6854), to be part of the controller circuit, but its use will be considered separately in the Data Generator/Receiver section. The description of the actual use of the components in this circuit will be better illustrated when either the controller software or the individual circuit being controlled are described. The connections between the controller and the circuits being controlled are through the parallel input output devices (VIAs). Figure 15 is a list of the control signals and an identification of the circuits to which they relate.

4.4.2 Host Interface

The host interface is, essentially, a part of the microprocessor control circuit but is identified separately since it interfaces to the host machine rather than the "personality" circuits which comprise the Spread Spectrum LAN node. The interface uses an Asynchronous Communications Interface Adapter (ACIA) operating to the V24 format. The device may be programmed to suit the number of data and stop bits, parity etc. required by the host computer. Messages may then be buffered in the node's RAM whilst awaiting transmission into the network

4.4.3 Synchronisation Circuits

Figure 16 shows the synchronisation, and closely associated code selection circuitry. For simplicity of generation, the modified MLSs, which form the address codes, are stored in a Read Only Memory. The reference (or synchronisation) code is stored in data bit 0 of 1024 consecutive memory locations with bits 1 to 7 being used for 7 other node address codes. For the demonstration system this number is more than adequate.

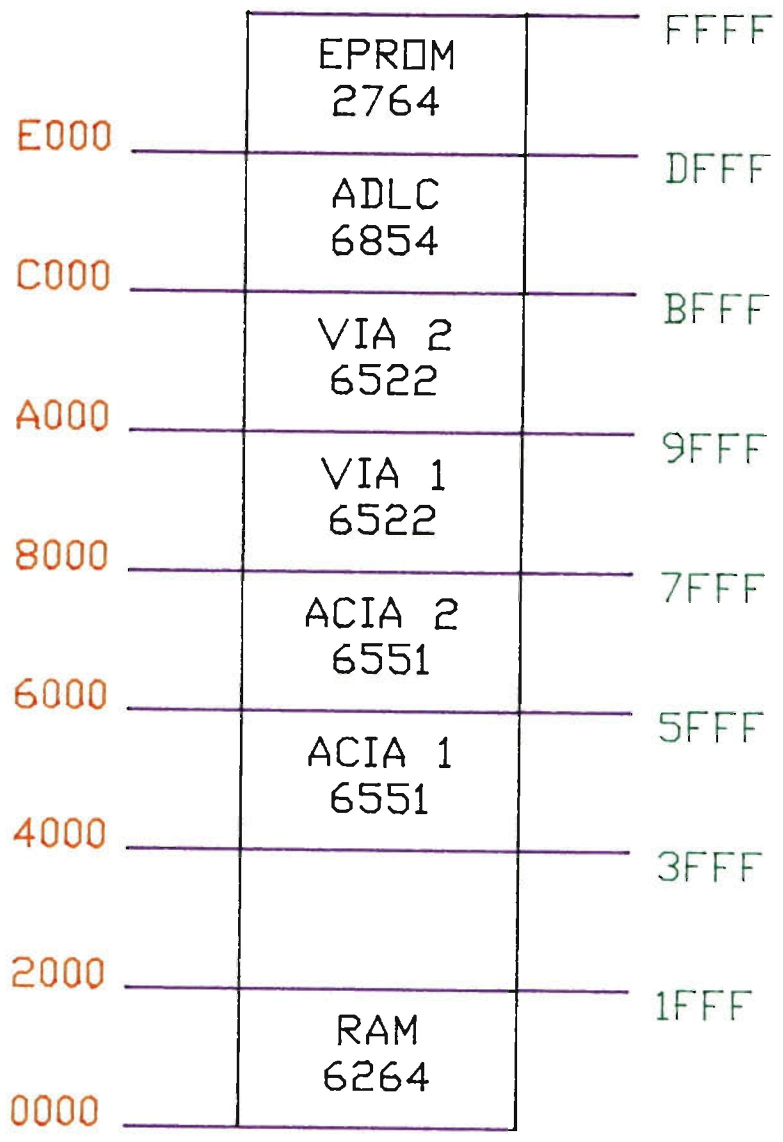


Figure 14 Memory Map for Microprocessor Control Circuit

Chip ID	Pin Number	Port Address	Pin Function	Description		Destination or Input From
U 15	2	8001	PA 0	ADC 800 Bit 0		U 34 (13)
	3		PA 1	1		(14)
	4		PA 2	2		(16)
	5		PA 3	3		(17)
	6		PA 4	4		(1)
	7		PA 5	5		(2)
	8		PA 6	6		(3)
	9		PA 7	7		(4)
U 15	10	8000	PB 0	Rx Sliding Correlator	S0	U 21 (11)
	11		PB 1		S1	(10)
	12		PB 2		S2	(9)
	13		PB 3		S3	(7)
	14		PB 4		S4	U 27 (1)
	15		PB 5	Tx or Rx to ADC		U 32 (11)
	16		PB 6			
	17		PB 7	Tx Data Enable		TC a17
U 14	2	A001	PA 0	Rx Data Bit 01		U 42 (8)
	3		PA 1	Rx Data Check Bit 02		U 42 (5)
	4		PA 2	Tx Data Bit 01		TC a2
	5		PA 3	Tx Data Check Bit 02		TC a3
	6		PA 4	Rx Data Clock 1200 Hz		TC a9
	7		PA 5			
	8		PA 6			
	9		PA 7			
U 14	10	A000	PB 0	Tx Sliding Correlator	S0	U 23 (11)
	11		PB 1		S1	(10)
	12		PB 2		S2	(9)
	13		PB 3		S3	(7)
	14		PB 4		S4	TC a4
	15		PB 5	Tx Code Select S0		TC a13
	16		PB 6	S1		TC a14
	17		PB 7	S2		TC a15
U 14	39	A00C	CA 2	Start ADC Conversion		U 34 (6)
	40	A00D	CA 1	End Of Conversion		U 34 (9)

Figure 15 V.I.A. Control Signal Pin Connections

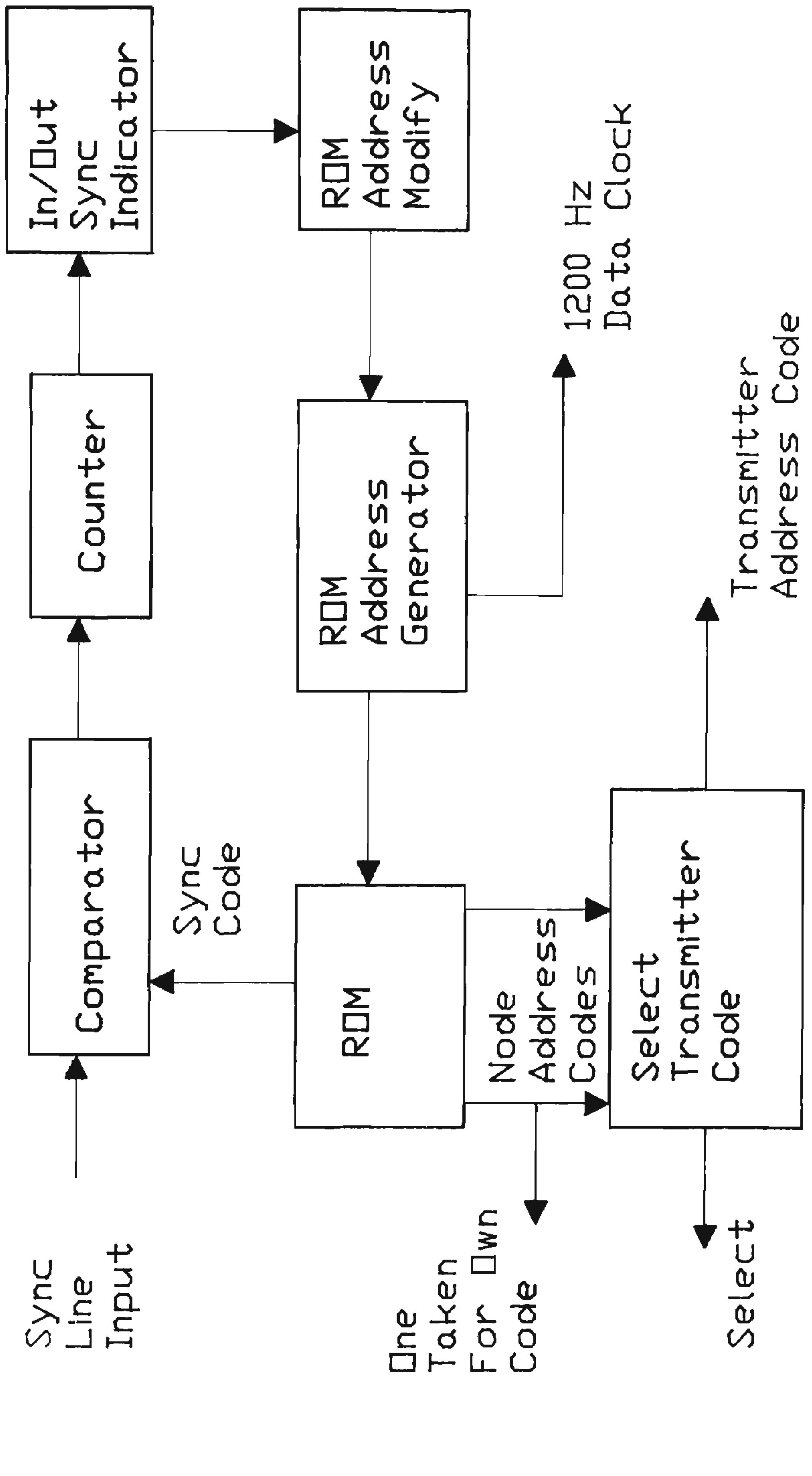


Figure 16 Sync Circuit and Code Select Block Diagram

The locally generated synchronisation code is compared with the signal on the sync line in the comparator section. It is possible that all 1024 phases of the local sync code must be tried before it matches with the sync line signal. During the search operation each phase is tested for **32** bits. If **20** consecutive bits are matched then code synchronisation is assumed and the circuit reverts to the "**In Sync**" mode and the "**In/Out Sync**" indicator set accordingly. If, after **32** bits, the matching criterion is not met then the ROM Address Modify circuit is activated. This circuit removes one clock pulse from the clock driving the ROM Address Generator circuit, thus effectively stepping the locally generated code one phase backwards with respect to the sync line reference signal. **32** bits of this phase are then tested, as described above.

The criteria for recognising loss of sync, when the circuit was already in the **In Sync** mode, differ from the criteria used when the circuit was previously "**Out of Sync**". When already **In Sync** it is assumed that errors on the synchronistion line are possible and are in fact more probable than a total loss of sync. This being the case the comparator circuit remains on the formerly "**in sync phase**" for **128** bits looking for **20** consecutive matches. Whenever **20** consecutive matches are found the counter circuit is reset and the sync indicator set to **In Sync**. If, after **128** bits of testing, **20** consecutive matching bits not are found the sync indicator is changed to **Out of Sync** and the "**Sync Search**" mode is entered. Figure 17 is a timing diagram of some of the more important waveforms in this circuit, and Figure 18 is a circuit schematic for the sync circuit.

The waveforms in Figure 17 are related to test points on the circuit schematic. In Figure 17 waveform **A** is the clock and waveform **B** is the ouput of the **20** bit comparator (showing no valid **20** bit comparison), and waveform **C** is the output of the search counter circuit. When a count of **32** is reached waveform C goes to a logic **1** causing the address modify waveforms **D** and **E** to be generated. These waveforms are **NAND**ed together to form the address modify clock enable signal, **F**, which is used to gate the clock signal, **G**, used to generate ROM addresses. The loss of a clock edge in this manner causes the output of the ROM to be altered by **1** bit in phase relationship with the reference signal on the sync line. **32** bits of the new phase are than compared with this reference signal.

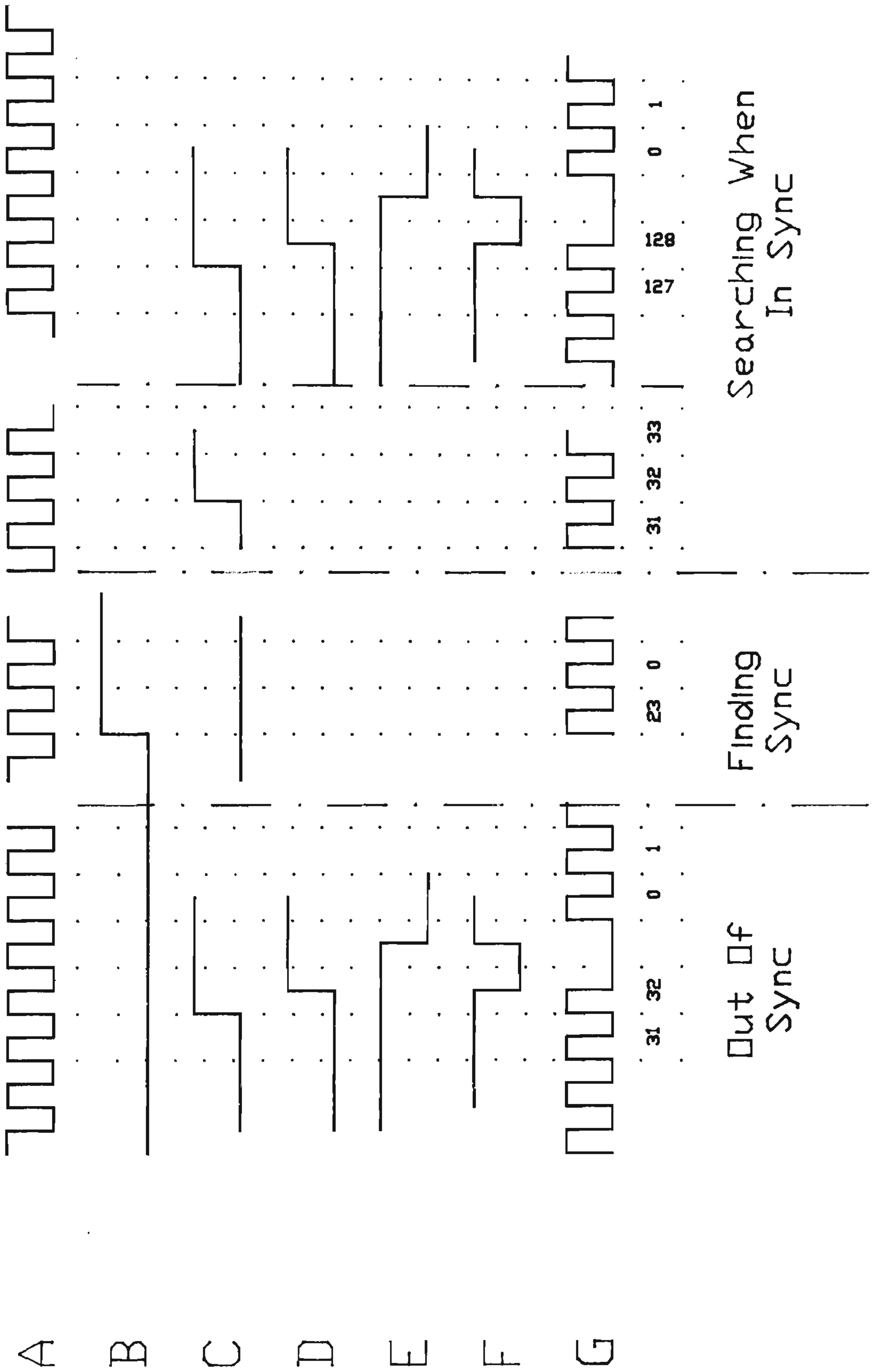


Figure 17 Timing Diagram for Sync Circuit

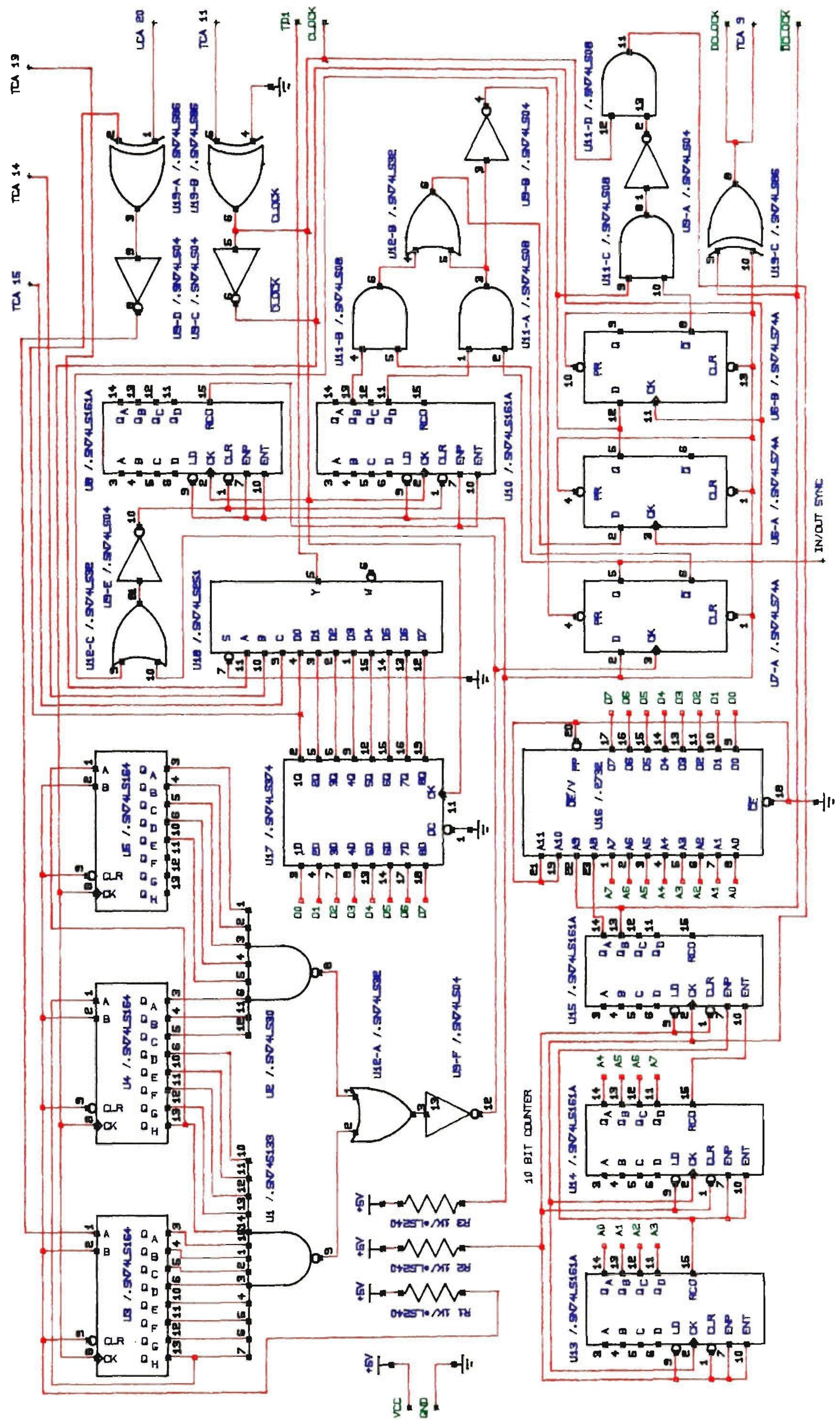


FIGURE 18 SCHEMATIC DIAGRAM FOR SYNCHRONISATION CIRCUIT

The second part of Figure 17 shows the acquisition of a valid **20** bit comparison. After a number of bits have been compared (**23** is only used as an example) a sequence of **20** consecutive bits in agreement are found and the comparator output goes to a logic **1**. This causes the counter to be reset to **0** and remain in this reset condition whilst the comparator output remains at a logic **1**. If for some reason, phase synchronisation is lost (as shown in the third part of Figure 17) the comparator output returns to a logic **0** and the counter is re-enabled. After **32** bits the address modify clock enable circuitry is not activated, however, as the circuit was previously in an **In Sync** condition. If a valid comparison is not found within **128** bits, however, the clock enable circuitry is enabled and one bit removed from the clock to the ROM address generator circuit, and the whole circuit reverts to the **Sync Search** mode.

It can be seen that, if the comparator circuit requires **20** valid bits from **32, one** error in **32** may cause a valid phase not to be recognised as phase **0**, as the error may make **20** bit comparison impossible before the next phase is tested. Thus a **3%** error rate could, apparantly, cause a complete failure to find sync. The situation is not quite as bad as this in practice since any **20** consecutive valid bits out of **32** would be sufficient to find sync, and in burst error conditions this could correspond to an effective error rate of **11 in 32** or **34.4%**. When the circuit is already in sync, however, finding **20** error free bits out of **128** is a much simpler task and, if they are uniformly spread, the circuit could withstand **6** errrors within the **128** bits and still find a valid **20** bit comparison. Thus an error rate of **1 in 20** or **5%** is unlikely to cause loss of sync problems. If the errors occur in bursts, then any burst sequence which permitted **20** error free bits in **128** would, again, not interfere with the circuit remaining in synchronisation. The circuit could, therefore, in theory withstand **107** errors within any **128** bit window and still not lose sync. This corresponds to an error rate of **83.6%**.

Having estimated that the circuit should be able to find synchronisation in an error environment of between **3** and **34%**, and that the circuit should be able to retain this synchronisation in an error environment of between **5** and **83%**, it was thought necessary to investigate the circuit's error rejection characteristics more thoroughly.

The worst case time to find sync, from recognition of loss of sync, should be the time to test 32 bits for 1023 non-valid sync phases and then 20 bits of the valid reference phase 0. The worst case time is therefore:-

$$1023 \times 32 \times Tc + (20 \times Tc) = 32756 \times Tc$$

$$= \underline{26.7 \text{ ms}}$$

$$\text{where } Tc = 1 / (1.2288 \times 10^6) = \underline{814 \text{ ns}}.$$

In order to test the performance of the synchronisation system in the presence of errors it was necessary to use an error generator which could be programmed to produce the error rate required, and which could be synchronised with the clock rate of the synchronisation sequence. No such device was available and so one had to be designed, built and tested. As this error generator is not, exactly, part of the mainstream research of this project the details of its contruction and testing are included at Appendix B rather than in the main text; but since it was designed specifically for this project, the description is comprehensive. A brief summary of the device is as follows:-

> The error rate is programmable between **1 x 10^{-6}** and **9 x 10^{-1}**. The output is either an error signal or, if data is injected at the input, a data signal which includes generated errors. The input and output are **TTL** compatible. In order to synchronise to a particular data rate, the generator requires a data clock input, which is also **TTL** compatible. The data clock rate may vary between **1 Hz** and **1.5 MHz**. Error insertion can be turned off and on with one switch, therefore constant connection and reconnection of the data signal is not necessary. The error rate quoted is the average, long term, error rate and includes multiple bit errors ie. when 3 consecutive bits are in

error this counts as 3 errors rather than a single error. The output of the error generator approximates to a **Gaussian** distribution based on the preset rate selected. as can be seen from Appendix B,

4.4.3.1 Performance Tests on the Sync Circuit

Using the error generator described briefly above, errors were injected into the signal on the synchronisation line. The error rate was increased from **1%** and the **In/Out Sync** indicator monitored to determine whether the circuit had lost sync and failed to re-find sync within the allowed 128 bits. As the error rate was increased, from **1%**, no sync dropouts were encountered until **10%** error rate had been reached. From this point **10** readings were taken at error rates between **10%** and **90%**. The number of sync dropouts vs the error rate is plotted in Figure 19. It can be seen from this graph that the maintenance of synchronisation was not a problem, even with error rates as high as **30%**. From this point dropouts increased until they became a serious problem above **50%** errors and from **60%** errors the circuit could not reliably stay in sync. These figures match closely with the predicted figures and indicate that the circuit will, indeed, maintain synchronisation in an extremely error prone environment.

Having proven the circuit's ability to retain synchronisation it was then necessary to determine the circuit's ability to achieve synchronisation in an error prone environment. For these tests the error generator was used as described above and an additional test circuit to force sync loss was also used. This circuit removed a clock pulse from the ROM Address generator input, causing an incorrect phase to be tested thus ensuring sync dropout. This method of generating sync dropout causes the next phase to be injected into the comparator circuit thus ensuring sync dropout and also requiring all 1023 alternate phases to be tested before the reference phase **0** was again tested. In this manner the the time taken to regain sync may be used as an indicator to determine the circuit's resynchronisation characteristics. A counter/timer was used to measure the time the circuit then took to regain synchronisation. The counter/timer was started by the **In/Out Sync** indicator falling from logic **1** to logic **0** and stopped by the sync achieved signal ie. a **0** to **1** transition on the **In/Out Sync** indicator.

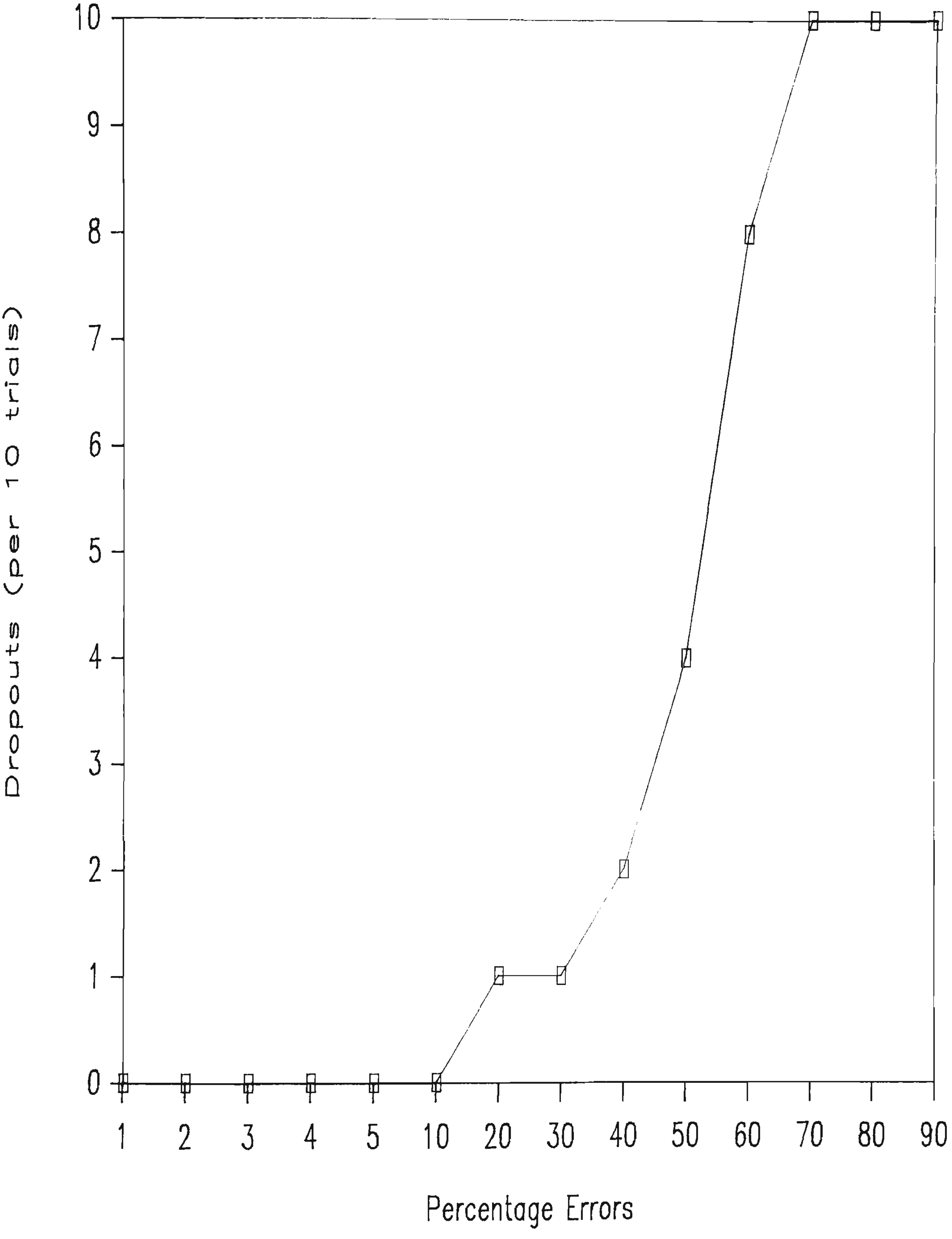


Figure 19 Sync Dropout vs Error Rate

Tests began with an error rate of **1%** which was increased in **1%** steps. 10 measurements were taken at each error rate. In the absence of errors the circuit should regain synchronisation within **26.7ms**, as shown previously. If the circuit failed to find sync on the first pass, thus requiring all 1024 phases to be re-tested , the time taken to achieve sync would be two cycles ie. approximately **54 ms**. All test figures are, therefore, normalised to the **26.7 ms** cycle and given as the number of cycles required before sync is achieved. Figure 20 shows a graph of "**the average number of cycles required to regain synchronisation**" vs "**error rate**". It can be seen that below **7%** the circuit found synchronisation on the first pass every time and that after this point the number of passes required increased rather rapidly. Having found sync, however, it is very unlikely that the circuit would lose sync again, as sync loss below **30%** error rate was extremely rare. The performance criterion for the successful operation of this circuit must, therefore, be its ability to find synchronisation reliably at the first attempt in an environment with an error rate as high as **7%**. Since this is a very high error rate, the circuit was considered suitable for use to synchronise the Spread Spectrum LAN.

4.4.4 Clock Recovery

The basic circuit functions in the clock recovery circuit are shown in Figure 21. Figure 22 shows timing diagrams for this circuit with the waveforms corresponding to the test points indicated in Figure 21. In keeping with the overall design aims of the project, i.e. keep the design simple and to use standard components wherever possible , the circuit does not employ sophisticated phase locked loops as may be expected.

In order to provide the ability to step the sliding correlator in 1/16th bit intervals it is necessary to locally generate a frequency of 16 times that used to produce the synchronisation data sequence. A local crystal controlled oscillator operating at 19.6608 MHz (i.e. 16 x 1.2288 MHz) is used for this purpose. The output of this local oscillator, **A**, is used to clock a divide by 16 circuit, the output of which acts as a locally generated 1.2288 MHz chip rate clock, **B**. The signal, **C**, on the sync line is input to a pulse generator such that every transition, whether it be positive or negative edge, causes a narrow pulse, **D**, to be

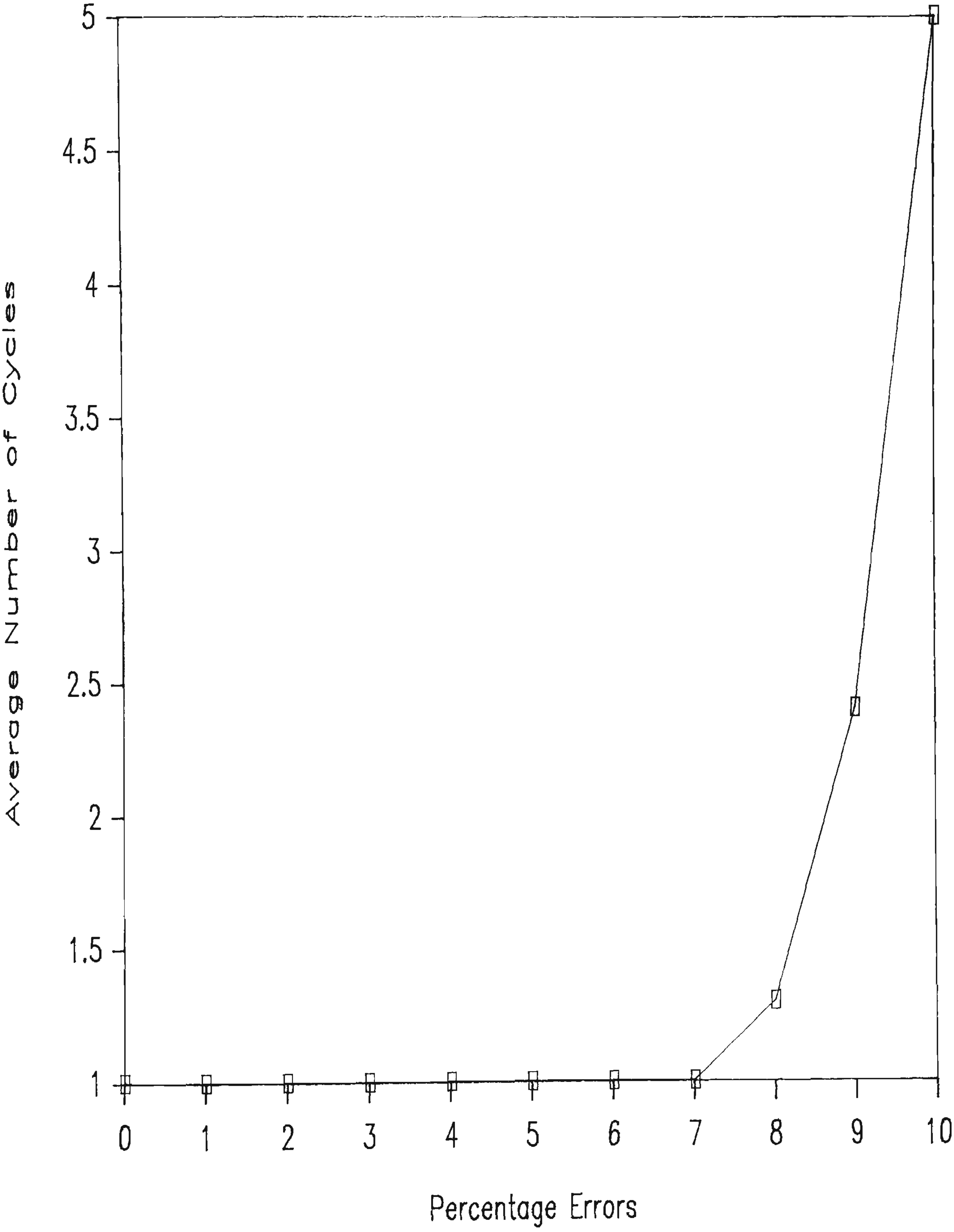


Figure 20 Cycles to Regain Sync vs Error Rate

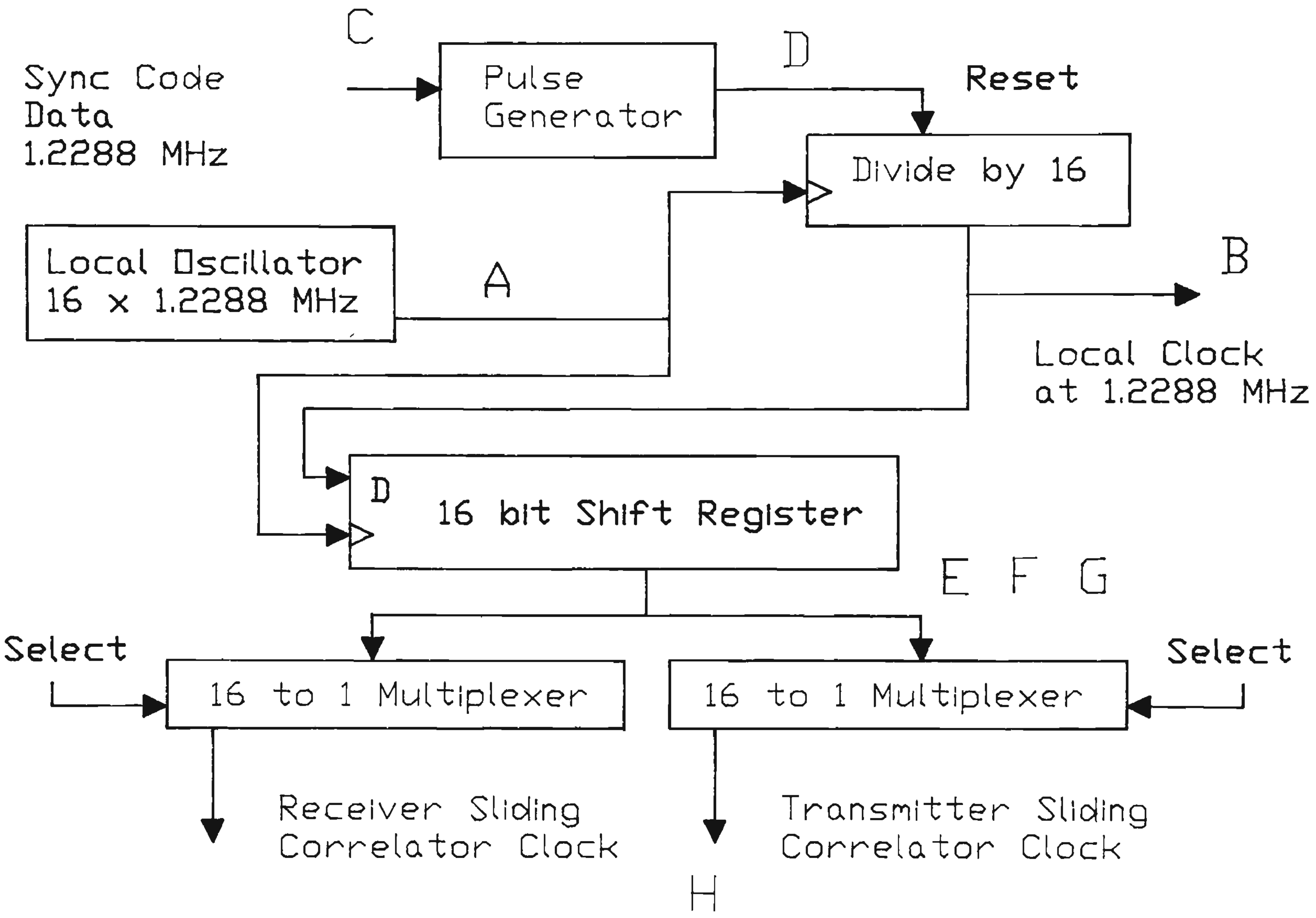


Figure 21 Clock Recovery Generator Block Diagram

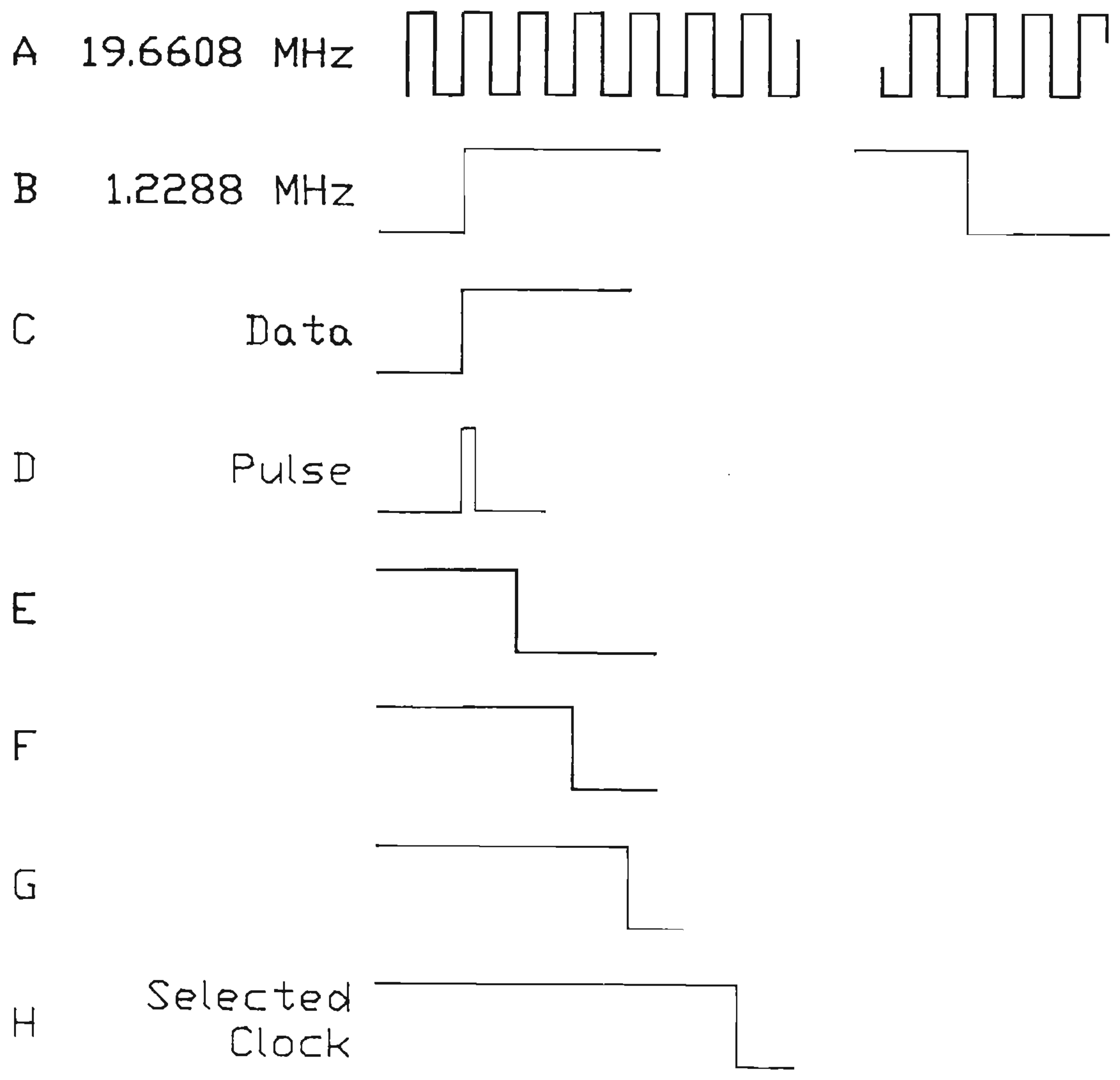


Figure 22 Timing For Clock Recovery

generated. These pulses are used to reset the divide by 16 counter such that the output, i.e. the locally generated 1.2288 MHz clock, is synchronised with the synchronisation signal on the sync line. The output of the divide by 16 circuit is, therefore, subject to jitter of up to 1/16th of a bit caused by any frequency differences between the local oscillator and that of the sync transmitter. Within the limits of this 1/16th bit jitter the local and reference clocks are thereby locked in phase and frequency.

To produce the clock necessary to create the 1/16th bit steps required for the sliding correlator the locally generated 1.2288 MHz clock is input as a data signal to a 16 bit **serial-in parallel-out** shift register clocked by the high speed local oscillator. The data outputs, **E** , **F**, **G**, of this shift register thus represent the 1.2288 MHz clock delayed in 1/16th bit increments. The 16 outputs of the shift register are then input to a 16 to 1 data selector/multiplexer and the delayed version of the 1.2288 MHz required, **H**, may then be obtained at the output by selecting which input is to be connected to the output. Two seperate data selectors are used to provide independently selectable signals suitable for the transmitter and receiver sliding correlators.

4.4.5 Transmitter Line Driver

Each node must be able to add its signal to those already be present on the data transmission line. To this end it is obvious that the transmitter line driver must not load the output of every other transmitter. The multiple access concept used in this network was described in section 1.9. The block diagram for the functional layout is shown in Figure 5. In order not to load other node's transmissions, and so as to approach (as far as possible) the idealised multiple transmission characteristics shown in Figure 6, each transmitter incorporates a high impedance current driver output stage.

Before the driver stage could be designed it was necessary to consider the properties of the transmission line chosen for this development model. For simplicity of connection and disconnection Balanced Twin Feeder with the following characteristics was used :-

- Clear polythene insulation

- Cores are 7 x 0.25 mm plain copper stranded

- Dim 9.7 mm x 1.8 mm

- Characteristic impedance 300 Ω

- Capacitance 13.2 pF/m

- Attenuation per 10 metres 0.12 dB at 10 MHz & 1.68 dB at 1000 MHz

For development purposes, a 100 metre drum of this cable was used, as an interconnection medium. This is terminated at both ends in the characteristic impedance in order to minimise mismatches and reflections.

The block diagram of the circuit designed to interface to this transmission line is shown in Figure 23. It was decided that the circuit should act as a constant current source (as near ideal as possible). In order to transform the **1** and **0** level binary into a two level, ± **1**, signal the driver should be able to source and sink current. If, for example, the circuit were to source current to represent a logic **1** and remain passive for the logic **0** signal, it would be impossible for a logic **0** signal to have the desired effect, or indeed to be detected in the presence of any logic **1** signal, ie. the system would be performing a logical **OR** function rather than **ADDING** signals. For this reason, it was decided to represent a logic **1** by adding **1** "unit" voltage to the existing voltage on the transmission line and to represent a logic **0** by subtracting **1** "unit" voltage from the existing voltage. This is brought about by sourcing or sinking current at an appropriate rate.

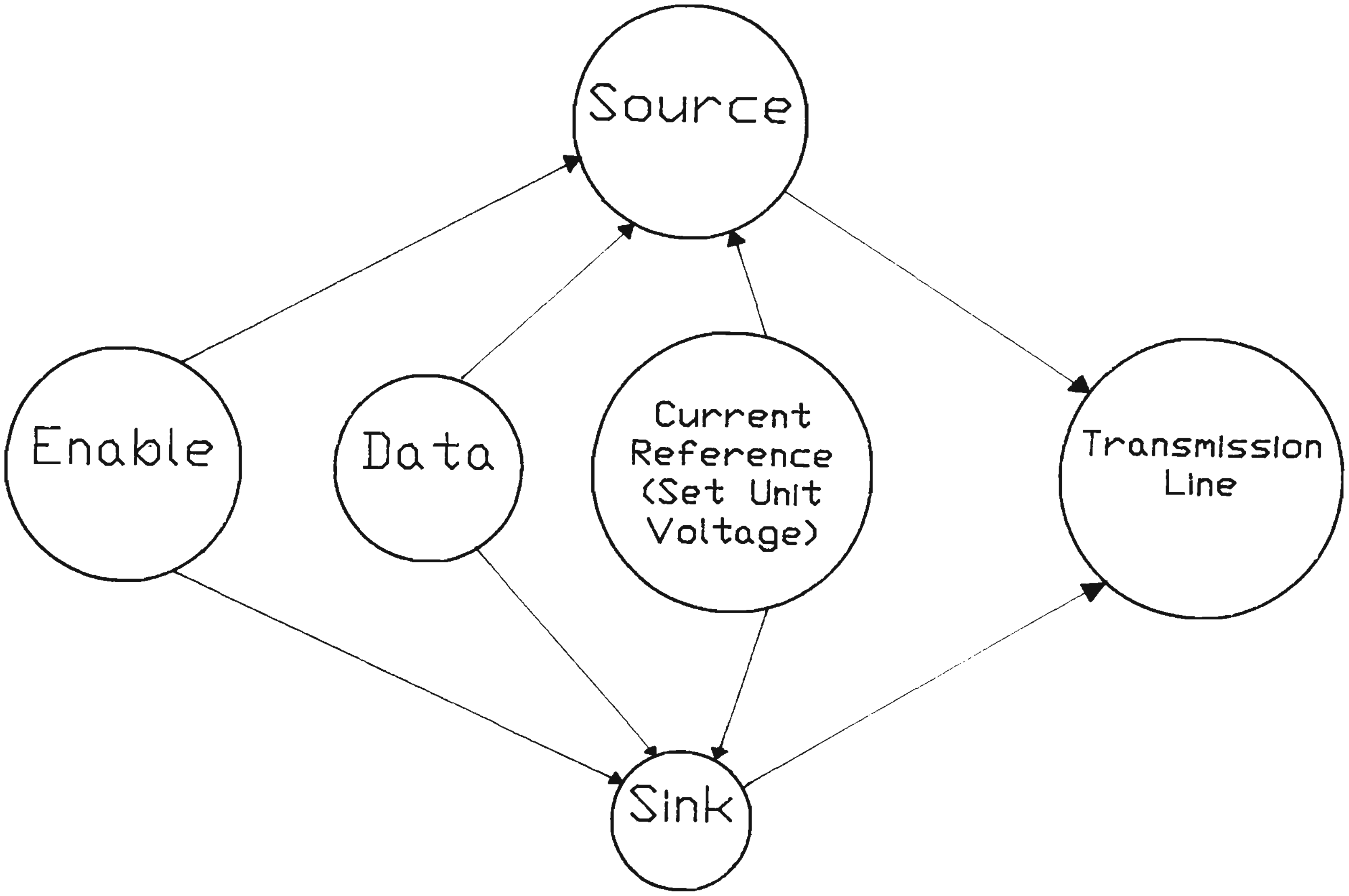


Figure 23 Transmitter Line Driver Block Diagram

Since this circuit would always be either sourcing or sinking current it was necessary to include an enable input so that both source and sink circuits could be disabled. And, as the driver present a high impedance interface to the transmission line, this means that the node may be electrically disconnected from the network.

A schematic diagram of the driver circuit is shown in Figure 24. The circuit makes considerable use of "**CURRENT MIRRORS**" to create the effect of an ideal current source. The current mirror **M2** is the current reference shown in Figure 23. Current flowing in the left hand transistor is determined by the 10kΩ variable resistor, P6, this current is reflected in the right hand transistor which is connected to the enable section. This 10kΩ variable resistor is used to set the reference current and hence the value of the "unit" voltage. The value selected for the unit voltage must be considered when selecting the scale factor for the multiplier (section 4.4.8.2) and the input amplitude adjustment for the integrate and dump (section 4.4.8.3). The enable section is made up of two identical NPN transistors. The right hand transistor is biased at 2.1 volts, determined by the three diode drops. The enable, **TCA 17** in Figure 23, is an input to comparator **C1**. If the enable input is greater than 2.1 volts then **TR5** will be on and current will flow in **M1**. If, however, the enable input is less than 2.1 volts then **TR8** will be on and current to **M2** will all be supplied by **TR8** and no current will flow in **M1**. **M1** is also a current mirror, if the enable input is greater than 2.1 volts then, as stated, all of the current for **M2** will flow through **TR5** and thus this current will be reflected down to **M3**.

M3 and comparator **C2** make up the data section. **TR14** is biased like **TR8** at 2.1 volts so that if the data input is greater than 2.1 volts then **TR11** will be on and **TR14** off, hence all of **M3's** current will flow through **TR11**. A data input greater than 2.1 volts represents a binary one and current must be sourced on to the line. This is done by **M4**, the current flowing in **TR11** is reflected by **M4** and is sourced onto the line, causing the line voltage to increase by **1** unit voltage.

If the data input is less than 2.1 volts (i.e. binary zero) then **TR11** will be off and all of **M3's** current will flow through **TR14**. This current will be reflected by **M5** and then by **M6** which will draw the same current off the line, (i.e. sink the reference current) causing the line voltage to decrease by **1** unit voltage.

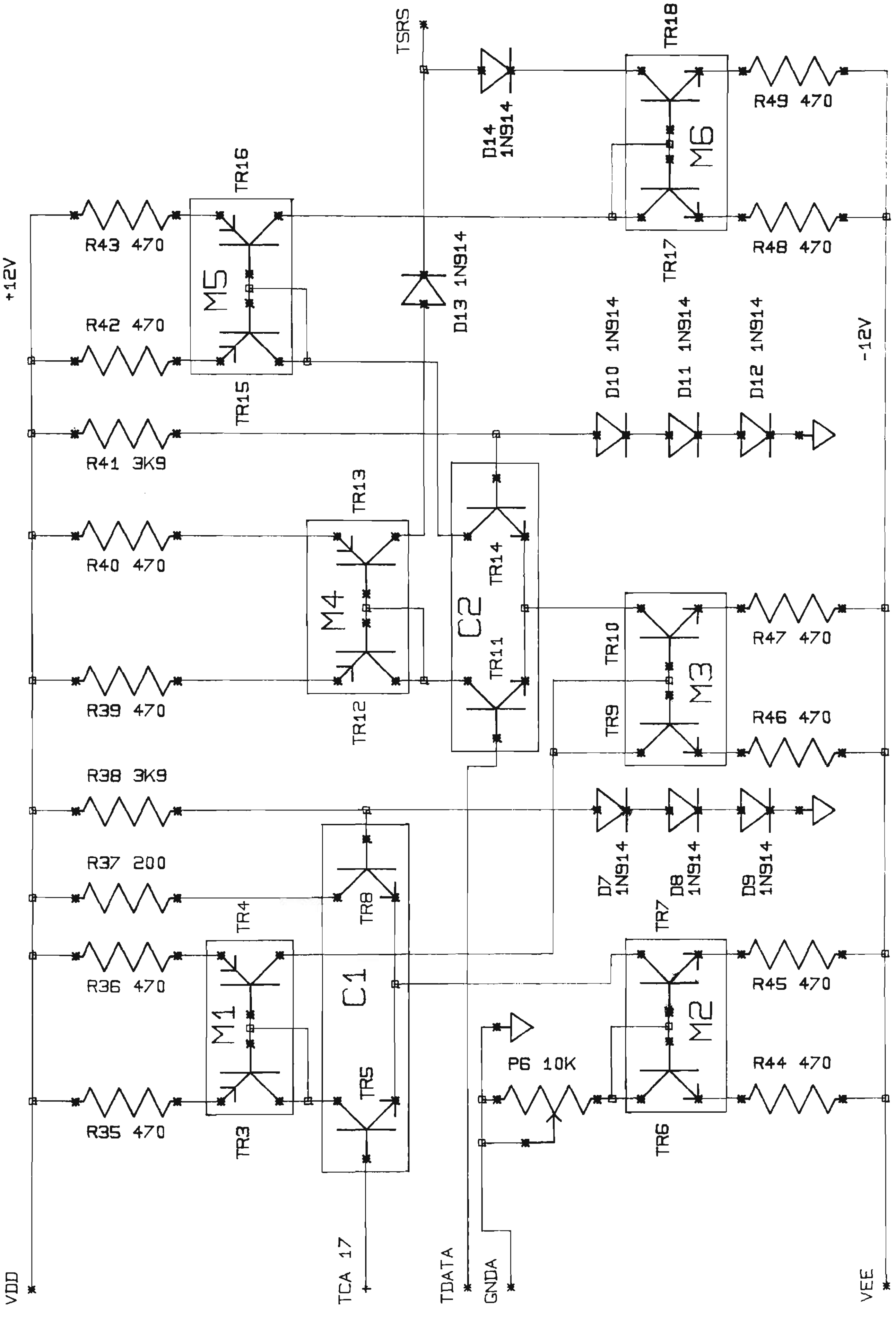


FIGURE 24 LINE DRIVER - CIRCUIT SCHEMATIC

Diodes **D13** and **D14** are used to ensure current does not flow the wrong way when the particular section is not in use. If the enable is off, ie. 2.1 volts, then no current will be reflected into **M3** and hence **M4**, **M5**, and **M6**. Transistors **TR13** and **TR18** will both be off and the circuit will present a high impedance interface to the network.

4.4.6 Code Select

Because of the nature of the code synchronisation and generation, the code selection circuit is greatly dependent on the sync circuit and is shown included in Figure 16 with the sync circuit. The ROM is pre-programmed with the synchronisation plus seven other address codes. The synchronisation code is programmed into the bit D_0 position of the ROM and the other bits D_1 to D_7 give the, already synchronised, address codes used for demonstration system testing. The outputs from the ROM are all latched, to remove ROM address-generation-caused glitches. Three select lines from the microprocessor control circuit are used to select any one of the remaining seven codes to act as a destination address. The local address code is extracted after latching, but prior to the selection circuit.

4.4.7 Analogue to Digital Converter (ADC)

The analogue to digital converter circuit contains much more than the name implies. Figure 25 shows a block diagram of this circuit. The analogue to digital converter is, actually, multiplexed between the transmitter and receiver circuits, the selection being made by the microprocessor control circuit using a single select line. In order to maintain the value of the signal, to be converted until the converter is available the signal must firstly be sampled and the value of the sample held until it has been used. Seperate sample and hold circuits are incorporated into the transmitter and receiver circuits. Commercial LF398 sample and hold ICs are used in conjunction with a 14053 analogue multiplexer. The analogue to digital converter is an eight bit ADC 800 device. With reference voltages of ± **5** volts the full scale value of **$00** represents **+5** volts, **$80** represents **0** volts and **$FF** represents **-5** volts (where **$** signifies hexadecimal notation). When rationalised these values equate to **0** to ±**5** volts being represented by **0** to ±**128** (decimal)

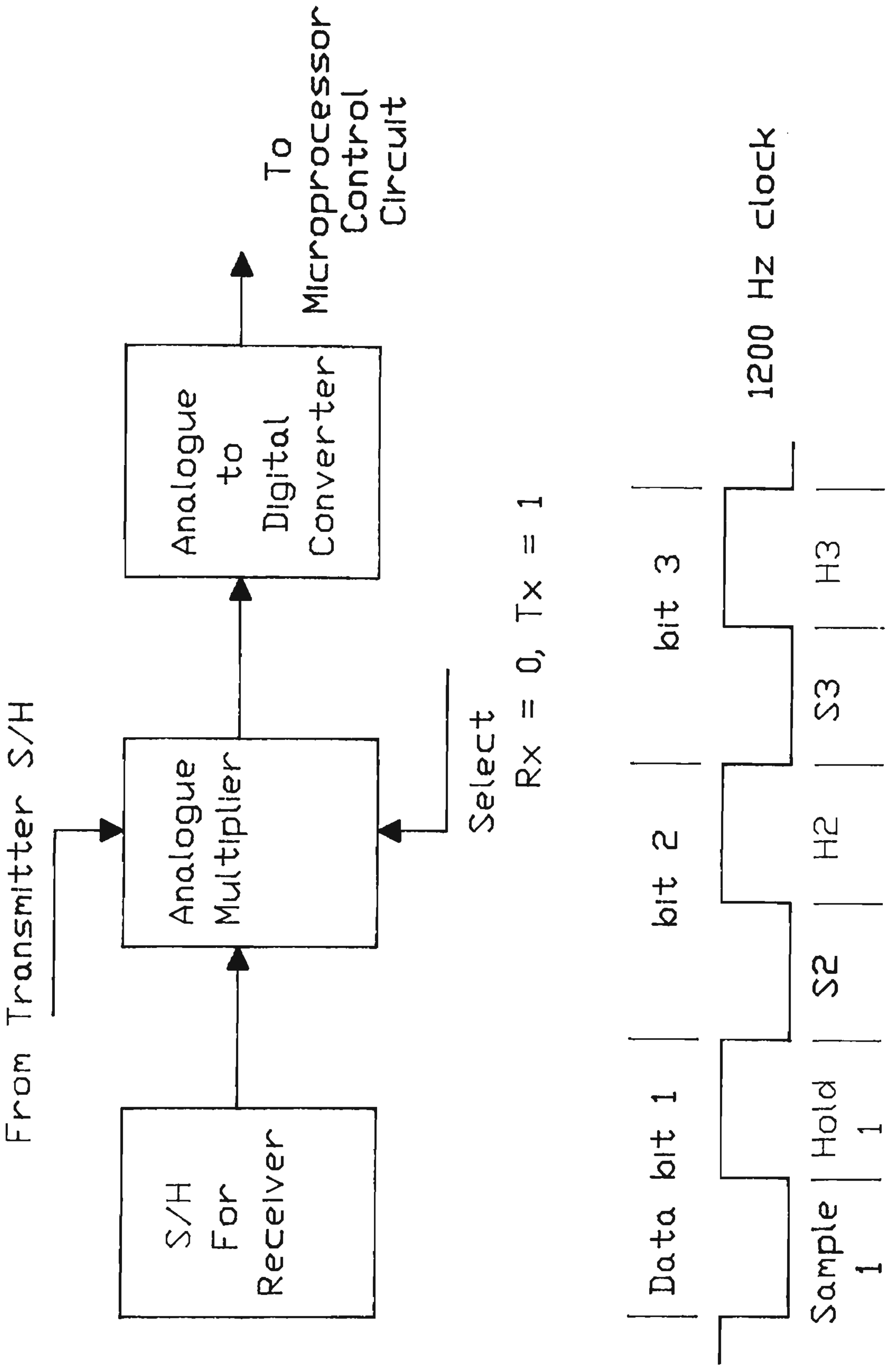


Figure 25 A to D Converter Block Diagram

respectively. The input to this circuit is from the output stage of the integrate and dump filter in the transmitter or receiver circuit (see Figure 26). The sample and hold circuit uses the inverted 1200 Hz data clock as its tracking and holding clock, as indicated in Figure 25. This gives **416 μs** for the sample and hold IC to track the output of the integrate and dump filter (of either the receiver or the transmitter). This substantial amount of time ensures that expensive fast tracking sample and hold devices are not necessary. The remaining **416 μs** in the hold period is then available for the microprocessor to select and process the ADC value. The clock which the ADC uses for its conversions is the 1.2288 MHz divided by 2. This gives a clock rate of 614.4 kHz. The maximum conversion time (quoted as 40 clock cycles) is, therefore, **65.1 μs**.

4.4.8 Receiver

Figure 26 is a block diagram of the receiver circuit. The receiver may be considered as 3 sections for description purposes:-

i) The Sliding Correlator

ii) The Analogue Multiplier

iii) The Integrate and Dump Filter

4.4.8.1 The Sliding Correlator

Figure 27 is a block diagram of the sliding correlator. This circuit uses the output of the clock selection circuitry in the clock recovery section (see Figure 21) to re-time the address code selected to be used for demodulation and data recovery purposes. This address code may be

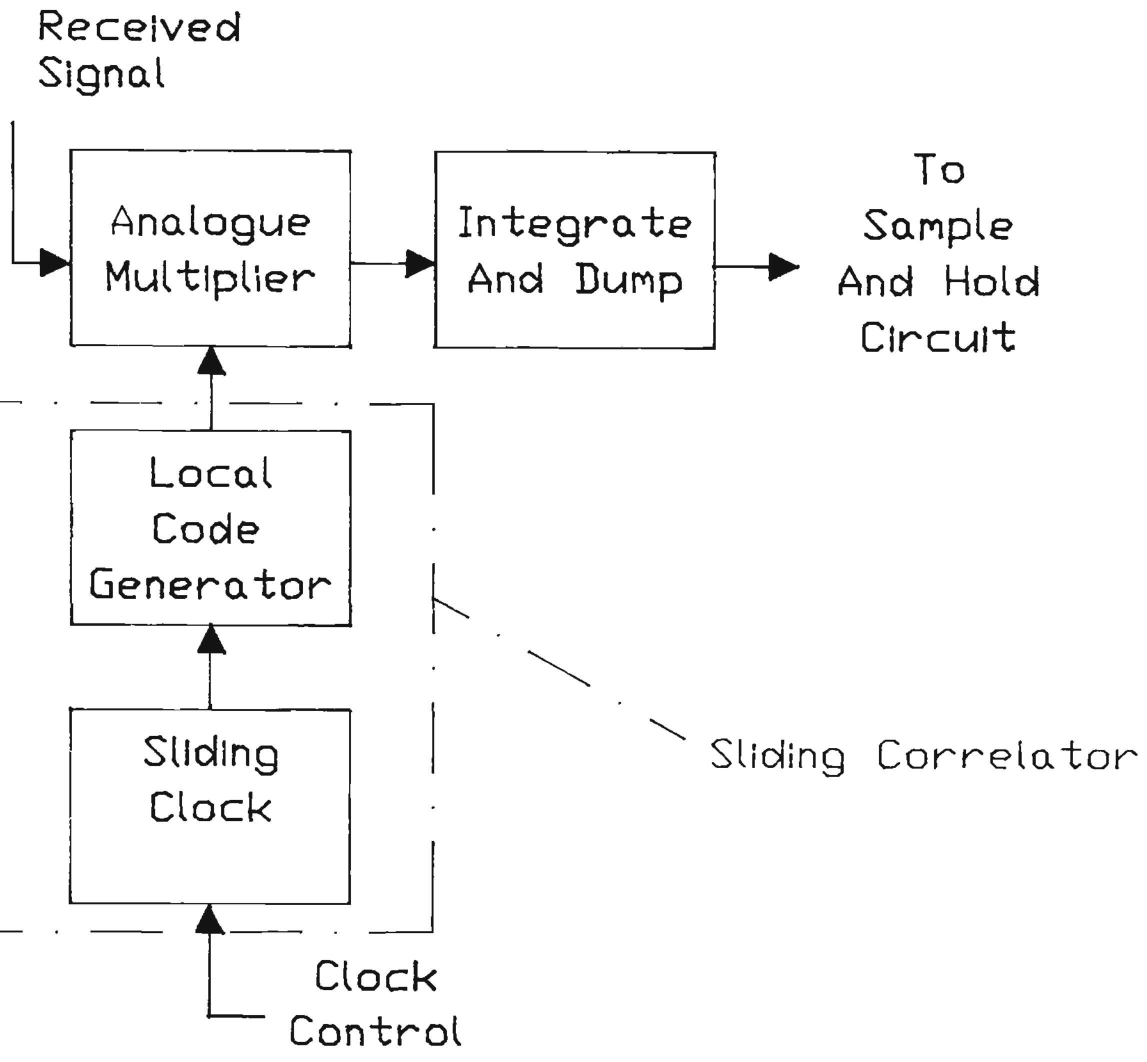


Figure 26 Receiver Block Diagram

the node's own code (if this is the receiver sliding correlator) or the code being used for transmission (if this is the transmitter sliding correlator). One of 16 seperate 1.2288 MHz clock phases may be used to clock the first "**D**" type flip flop, and thus re-time the address code in 1/16th bit increments over a whole bit position. For delays in excess of a single bit the address code is passed through a second **D** type giving a 2 bit total delay available. Four select bits are used to select the clock delay phase and a fifth is used to select < 1 bit or ≥ 1 bit delay. In this way the whole 2 bit window required for the sliding correlator function may be generated giving 32 x 1/16th bit correlator sample positions.

4.4.8.2 The Analogue Multiplier

The major component of this section is the RS 1495 Analogue Multiplier IC. Figure 28 shows a basic block diagram of the functional components of this circuit. The full calculation of parameters for the use of this device is shown in Appendix C. The inputs into the RS 1495 for the two signals to be multiplied are designated $\mathbf{V_Y}$ and $\mathbf{V_X}$. The received signal, **r(t)**, is connected to $\mathbf{V_Y}$ and the local code, **Cx'(t)** is connected to $\mathbf{V_X}$, as shown in Figure 28. The two signals are multiplied together in order to facilitate data recovery as shown in equation (18) in section 1.9.

Since the modulating address code, **Cx'(t)**, is a ± **1** signal, then remultiplying by the same code will give $\mathbf{Cx'^2}$ which will always be **+1**. This **+1** value will be integrated over the full 1024 bits which represent a data bit, thus reproducing the original data bit. The multiplication **Cx'(t) Cy'(t)** (where **Cy'(t)** represents any other address code) will, on average, produce as many **+1** values as it would **-1** values over the 1024 bits, thus cancelling each other out and producing no accumulated value at the output of the integrate and dump filter. Separate **X** and **Y** offset adjustments are available on the RS1495, to remove any DC components, and a scale factor adjustment is also available, allowing the multiplication output to be adjusted to suit power supply limitations and line signal voltage parameters etc. A separate level shifter circuit (made up of a LF741 operational amplifier) is included which also transforms the balanced output signal into a single line output signal. This output signal may then be level shifted, using an output offset

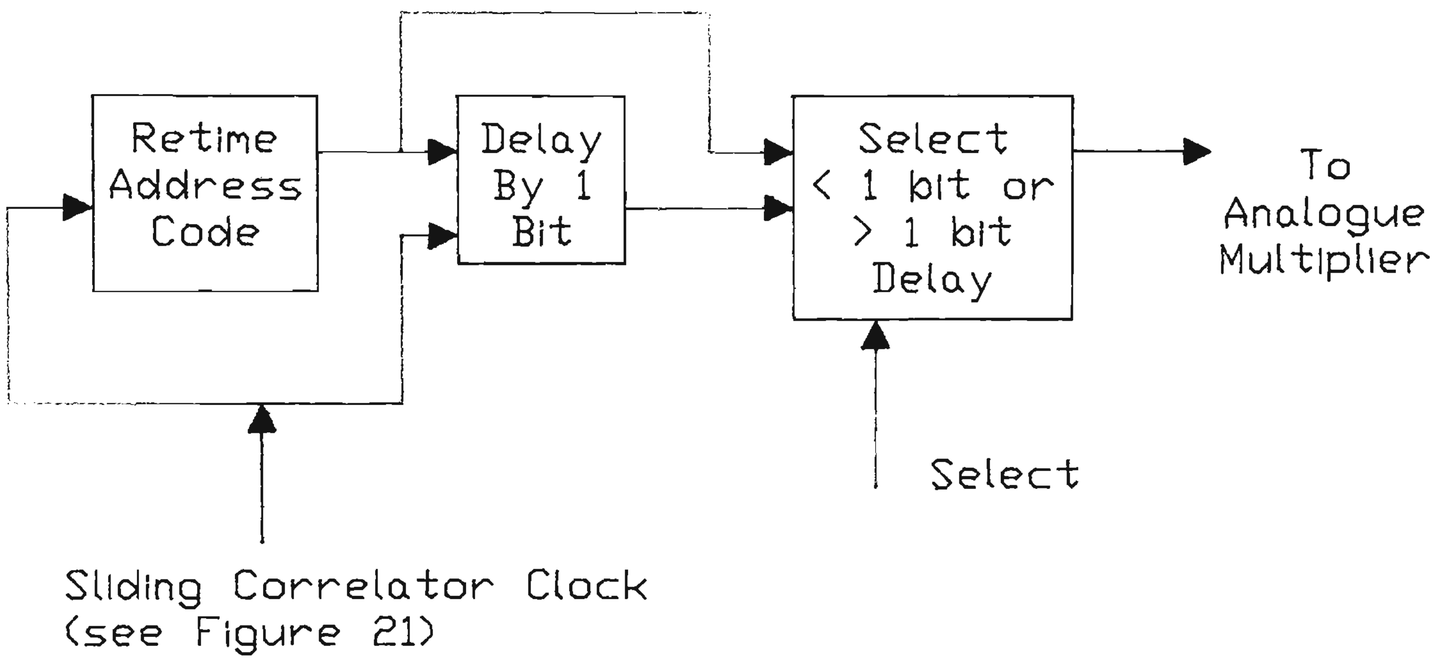


Figure 27 Sliding Correlator Block Diagram

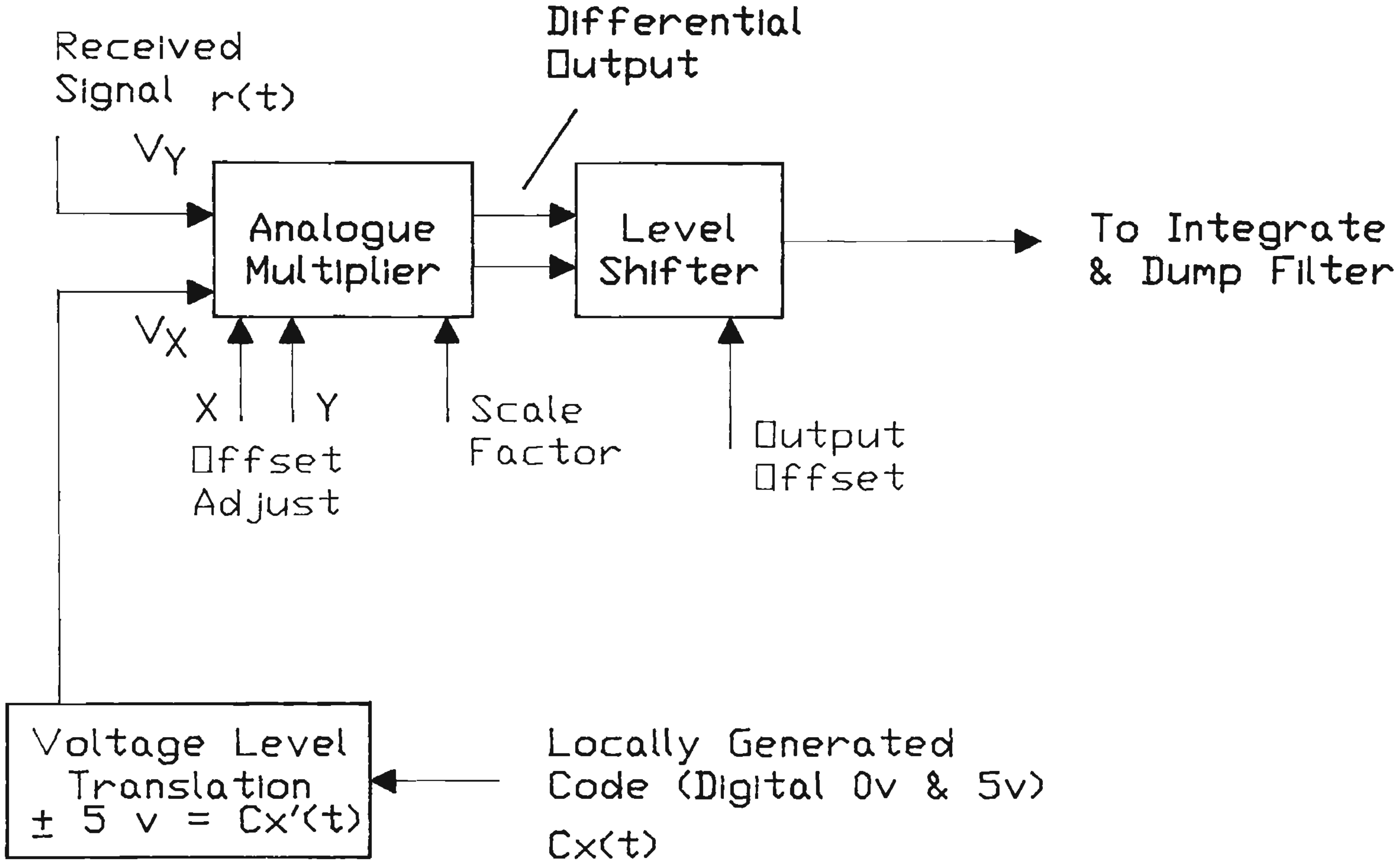


Figure 28 Analogue Multiplier Block Diagram

adjustment, in order to remove any residual DC offset before the signal is presented to the integrate and dump filter.

4.4.8.3 Integrate and Dump

A block diagram of this circuit, with appropriate timing waveforms, is included in Figure 29. The integrator consists of a LF741 operational amplifier in an integrator configuration with a serial input resistor and a feedback capacitor. The values of the components are chosen so as to ensure linear integration over a data bit period of **833 μs**. The input amplitude adjustment allows the value of the integrated output to be chosen within the range ±**1** to ±**10** volts. The standard integration value was selected to be ±**2.5** volts. This is set to give half scale readings on the Analogue to Digital converter, thus leaving the range between ±(**2.5** to **5** volts) available for collision detection. The integrator output value is held, in the Sample and Hold circuit, on the rising edge of the 1200 Hz data clock. One bit of the address code clock later (1.2288 MHz) a pulse one clock bit wide is generated which causes the integrator output to be reset to zero by discharging the integrator capacitor using FET switches. This dumping of the previous value ensures no residual or carry over effect from one data bit to the next.

Besides routing the integrator output to the ADC circuit, so that any collisions may be detected on an individual bit basis, the signal is also input to a comparator and voltage translation circuit to reproduce a digital signal representing the logic **1** or the logic **0** value of the, now decoded, data bit. This digital signal is then clocked into the data generator/receiver section for byte and protocol evaluation etc.

4.4.9 Transmitter

Figure 30 shows a block diagram of the transmitter circuit. The circuit is, in fact, a collection of circuits previously described. The address code to be used for transmission, i.e. the destination's code, is selected and a replicated receiver used to pre-check for existing transmissions which are using this code, i.e. a

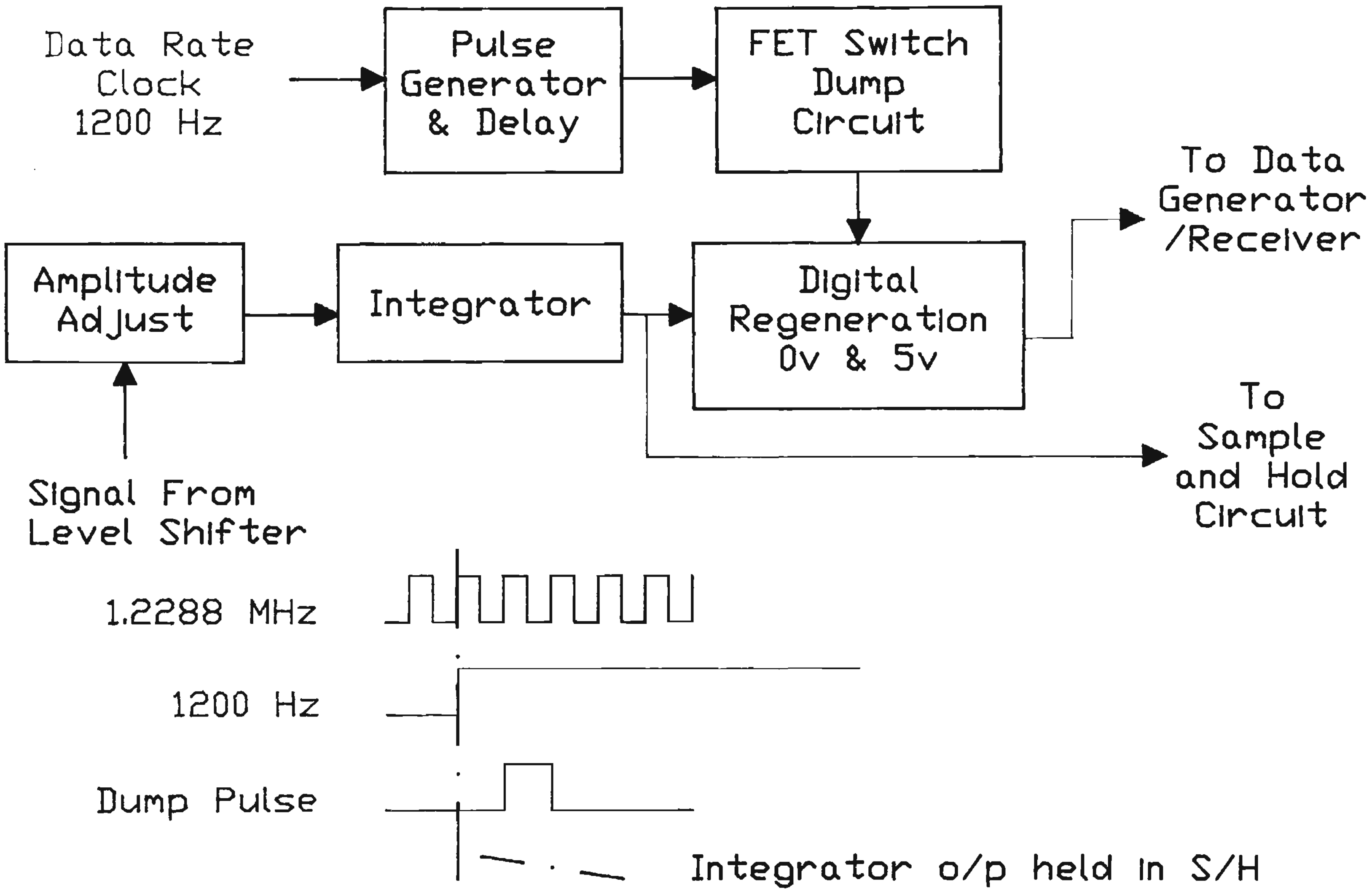


Figure 29 Integrate & Dump Circuit and Timing Diagram

channel occupancy check. If no existing transmissions are detected over the full 2 bit window, which is available using the sliding correlator, then the output from the data generator is multiplied with the selected address code before being routed to the transmitter line driver for injection into the transmission medium. The transmission medium is also constantly monitored by demodulating using the destination's address code to ensure correct, collision free, transmission of the required data. The transmitter may, thereby, validate its own transmissions and take remedial action in the event of a collision being detected.

4.4.10 Data Generator/Receiver

The Data Generator/Receiver is shown as part of the microprocessor control circuit in Figure 13, although it appears on board **P** with the transmitter circuitry. A Motorola Advanced Data Link Controller, MC 6854 [26], device is used to provide the level 2 protocol for the International Standards Organisation (**ISO**), Open System Interconnection (**OSI**), seven layer model for a Local Area Network. The initial data preamble, required for synchronisation, is controlled by the microprocessor which sets the ADLC to transmit a Mark on the idle condition. The destination address code is then selected and the the transmitter line driver enabled. These conditions are maintained for 32 data bit periods. After this phase the ADLC will be configured to generate and detect the appropriate protocol to ensure data transmission and reception. The protocol includes flag detection and synchronisation, zero insertion and deletion, frame check sequence generation and validation, etc. The transmitter and receiver sections are clocked at the 1200 Hz data clock rate.

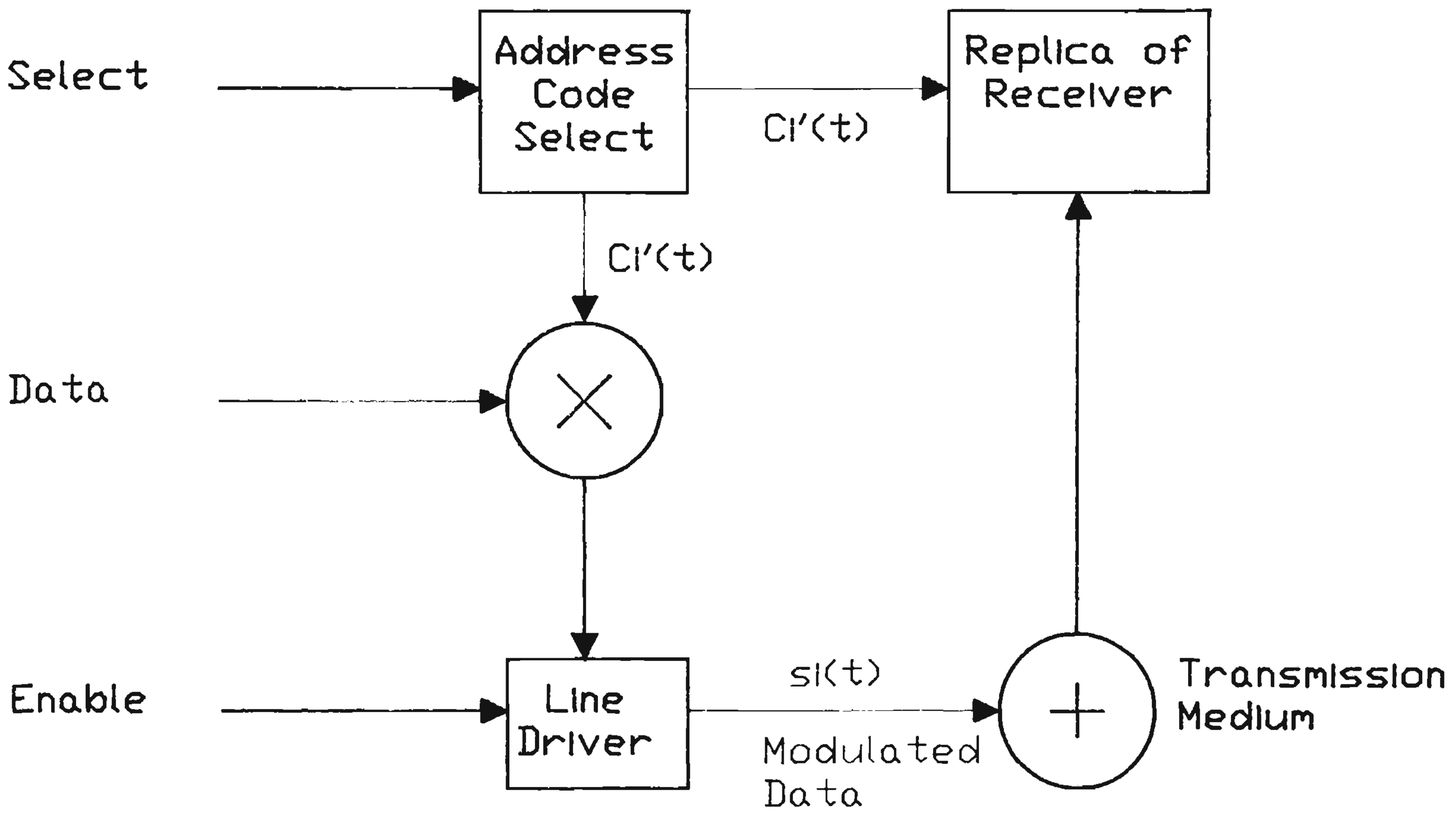


Figure 30 Transmitter Circuit Block Diagram

5 HARDWARE FUNCTIONAL TESTING

5.1 Summary

This chapter describes the system tests carried out on the hardware, as opposed to the tests of the individual circuit elements described in chapter 4. The circuit configurations used for test purposes are also outlined. Test results are also related back to the theoretical results predicted in sections 3.4 and 3.5.

5.2 Bit Error Rate (BER) Test

The first test to be carried out to determine some of the characteristics of the hardware was the determination of the hardware address code demodulation in the presence of other signals and external interference. This test was, therefore, designed to prove the concept illustrated mathematically in equation (18) in section 1.9. Three transmitting nodes and two receivers were used for this test. This equipment was configured in a similar manner to that shown in Figure 31.

The purpose of the test was to determine the Bit Error Rate (**BER**) produced under different levels of Signal to Noise (**S/N**). The word noise is used here to represent total interfering signals and includes both co-channel interference (from the two other transmitters) and external interference which took the form of Added White Gaussian Noise (**AWGN**). The analogue White Gaussian noise signal was produced from the Error Generator which is described in Appendix B by taking the **2^{31}-1** bit **MLS** at 36 MHz and passing this through a simple low pass filter with a 3 MHz cut-off. The output of this MLS would produce a spectrum of a **(sinc x)^2** function illustrated in Figure 32 with its first null at the clock frequencey of 36 MHz. By band limiting this to 3 MHz, only the central part of the spectrum is used, i.e. the section with a relatively constant output level. The band limited output spectrum is shown in Figure 33a. True white noise contains all

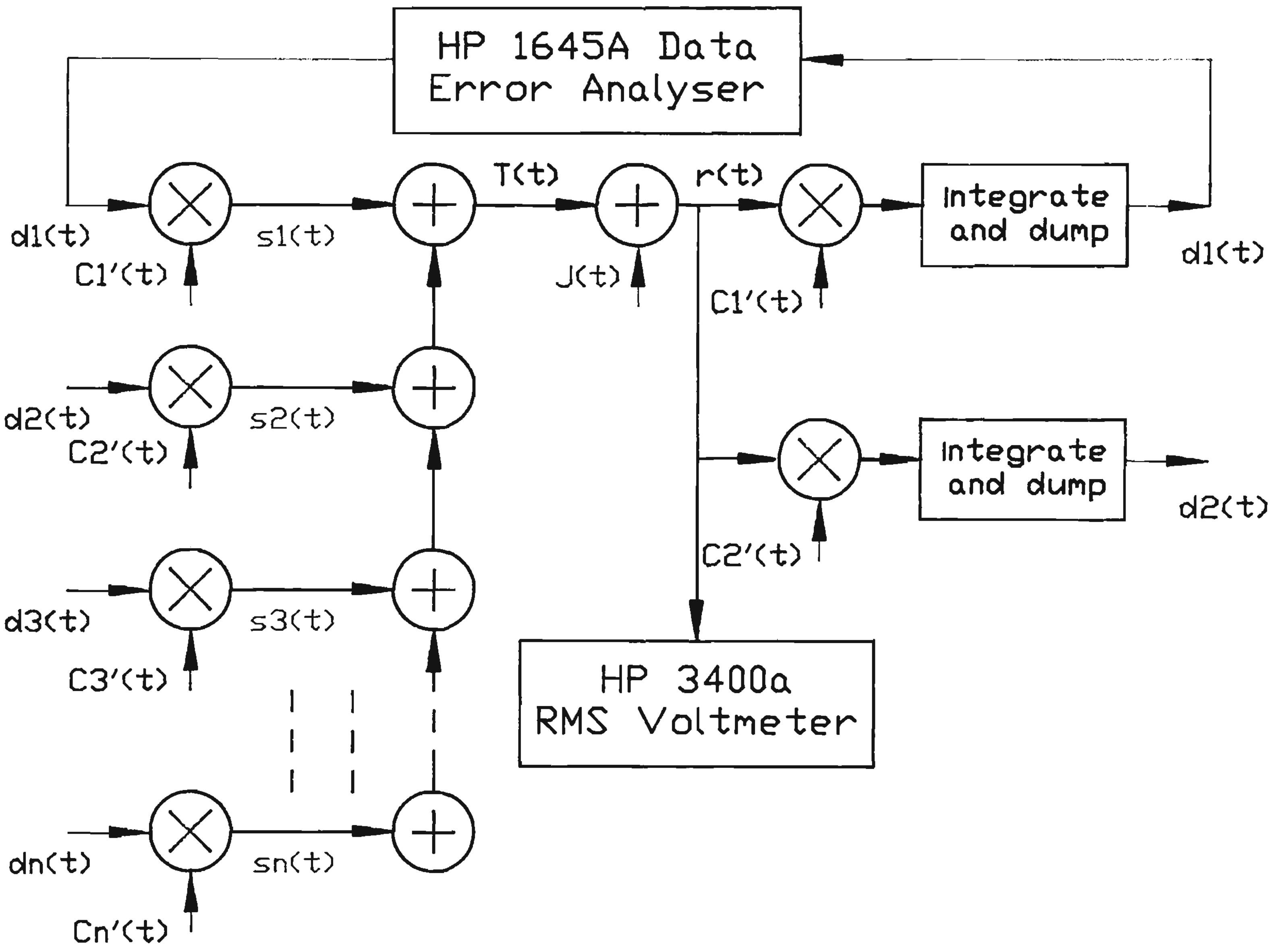


Figure 31 Bit Error Rate Test Hardware Configuration

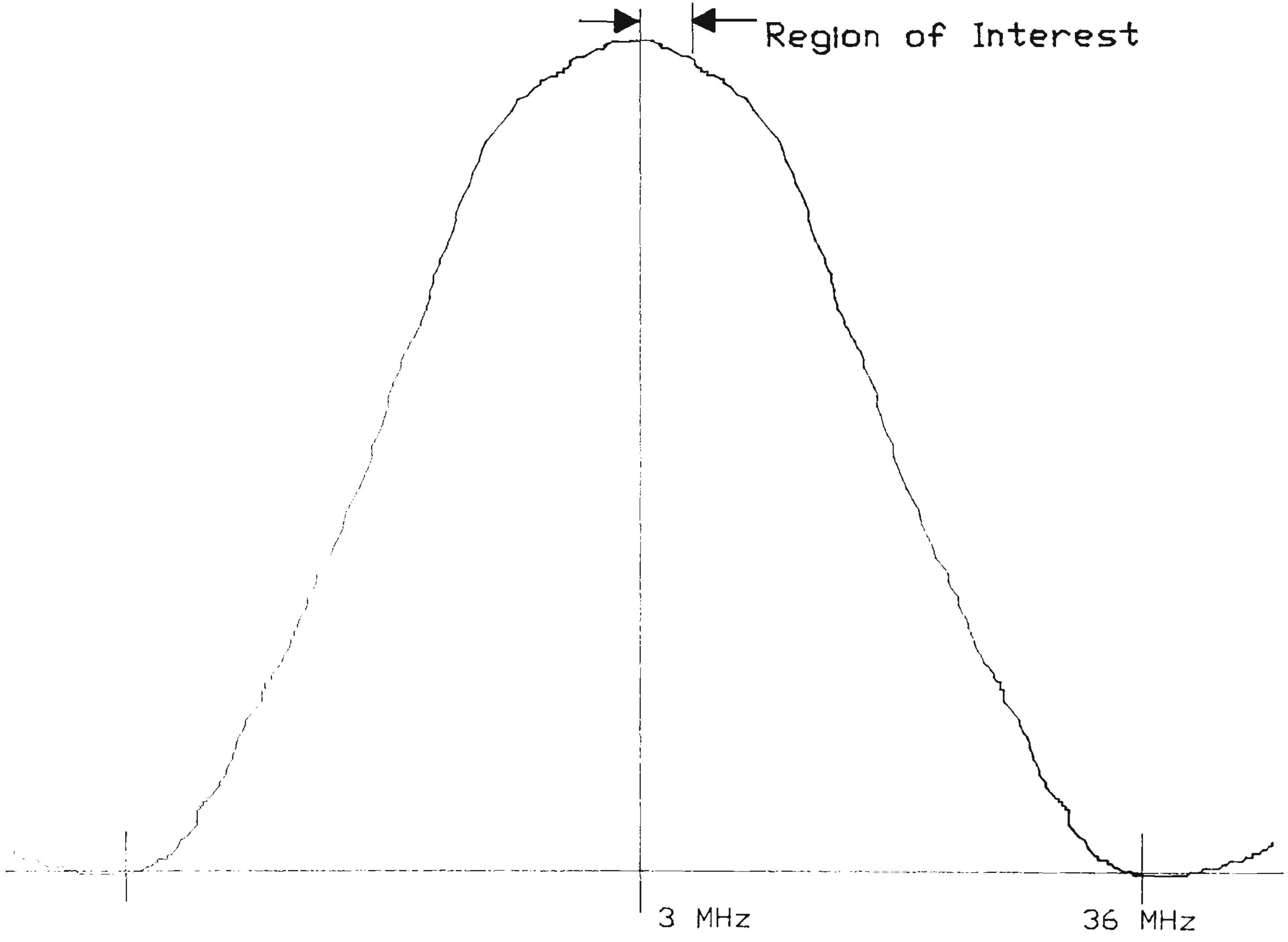


Figure 32 White Noise Generation

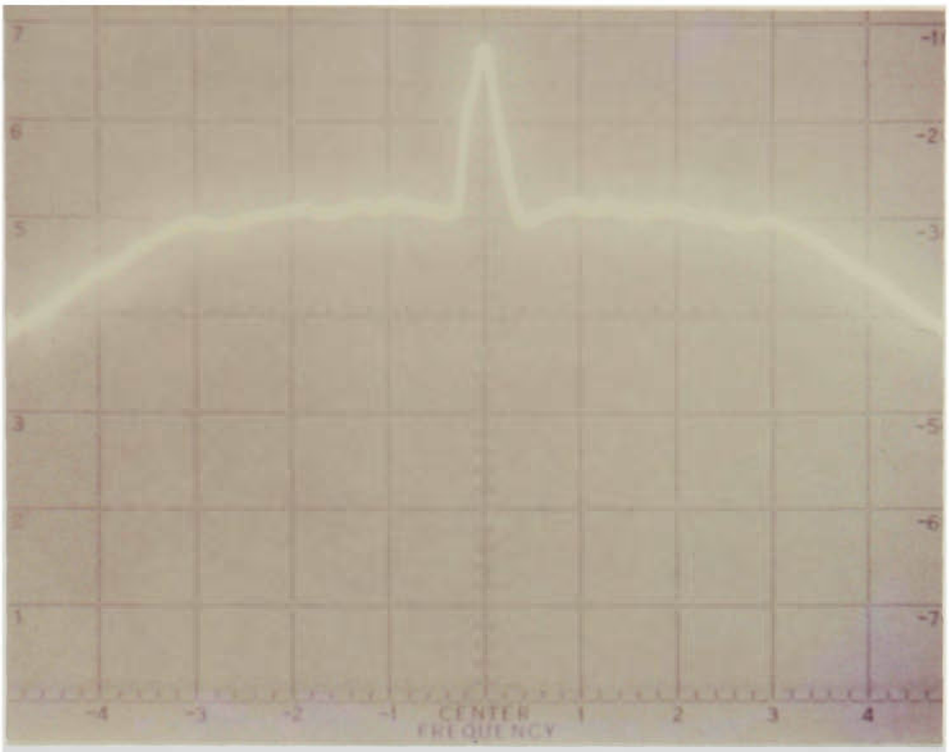


Figure 33a Photograph of Bandlimited Spectrum

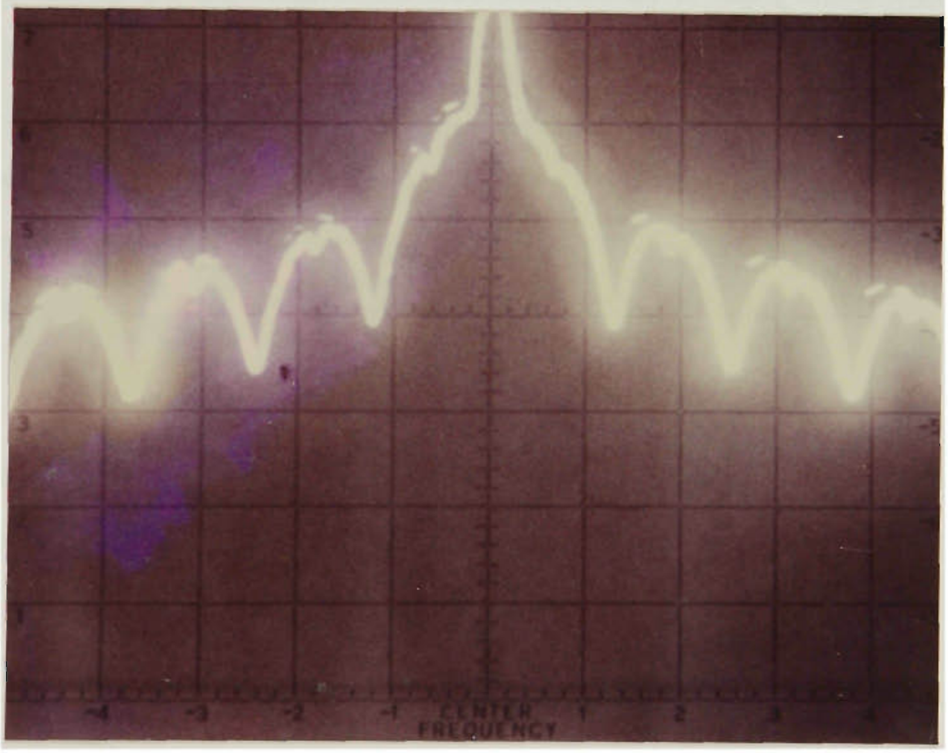


Figure 33b Photograph of Single Transmitter Spectrum

frequencies with equal energy but this signal actually contains discrete frequencies with a frequency spacing of:-

$$f = f_{clock} / (2^n - 1) = (3 \times 10^6) / (2^{34} - 1) = 0.0167\text{Hz}$$

With discrete spectral line spacing of **0.0167 Hz** over the whole 3 MHz band this generator could be considered to approximate to White Gaussian noise. The white noise was band limited to 3 MHz, as the 3dB bandwidth of the RS 1495 Analogue Multiplier (i.e. the receiver) is given as typically 3 MHz.

The three transmitter signals, $\mathbf{s_1(t)}$ to $\mathbf{s_3(t)}$, were summed with the AWGN signal, **J(t)**, before being injected into the transmission medium, to arrive at the receivers as signal **r(t)**. The HP 1645A Data Error Analyser acted as a data source for transmitter **#1** and the output of receiver **#1** was routed to the same device for analysis. Transrnitter **#2** and receiver **#2** were used merely to demonstrate simultaneous data communication, no measurements were made on channel **#2**. Transmitter **#3** is included as an additional source of co-channel interference. The transmission rate of the three transmitters is 1.2288 MHz and Figure 33b shows the spectrum of the received signal at **r(t)** with only one transmitter being input to the summing amplifier and no additional noise signal, (ie. **J(t) = 0**). Figure 33c shows the spectrum at **r(t)** with three transmitters and added noise at a signal to noise ratio of **-11** dB.

The upper trace in Figure 33d shows the time domain of this same received signal. The lower trace shows the data recovered, $\mathbf{d_1(t)}$, from this signal.

The output of the summing amplifier was tuned to increase the noise content in the output and the HP 3400A RMS voltmeter was used to measure the signal to noise values. The HP 1645A has a limited error count before overflow occurs, therefore at higher error rates the number of bits selected to be included in each test was much lower than for the low values of error rate where $\mathbf{10^7}$ data bits were transmitted.

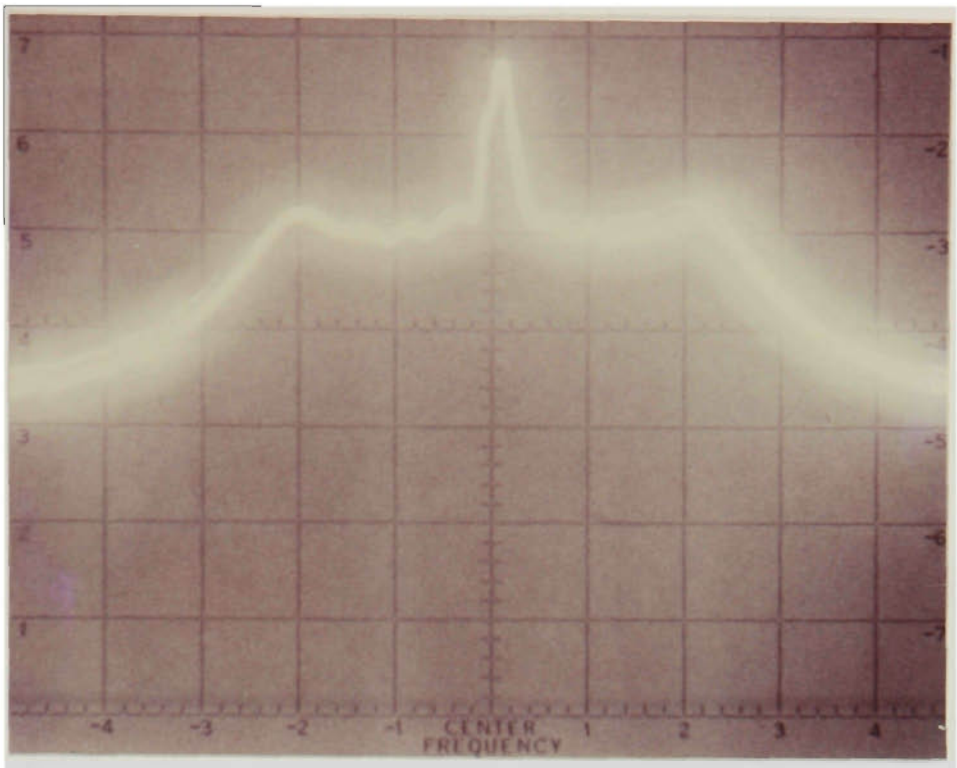


Figure 33c Photograph of Noise + Multi-Transmitters Spectrum

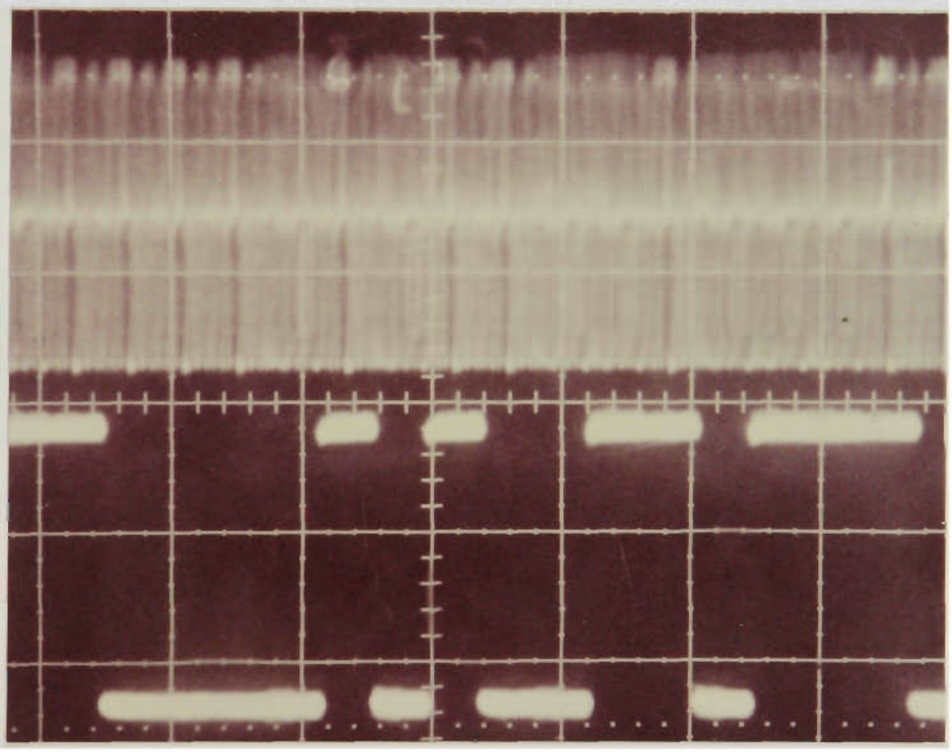


Figure 33d Photograph of Received Signal / Recovered Data

10 **BER** samples were taken at each selected **S/N** value and the **average** of these **10** tests plotted against the **S/N ratio** to produce the graph shown in Figure 34.* It can be seen from this graph that with approximately **-13** dB signal to noise ratio the Bit Error Rate is as high as **2 x 10**$^{-2}$ but at **-10** dB **S/N** the **BER** drops so low that no errors were recorded in a **24** hour test period.

These results show that this type of Spread Spectrum system may be used to transmit data in the presence of noise, even when the total interference is significantly greater than the signal.

5.3 Sliding Correlator Test

The sliding correlator was included in the receiver circuitry in order to be able to detect signals subject to delay of up to **2** bits, caused by propagation delays and timing differences (see chapter 3). A peak of correlation is found where the two signals are in phase, and the correlation value drops uniformly until, with the signals more than ± **1** bit separated, zero correlation is apparant beween them. It is obvious, therefore, that for maximum demodulation efficiency the receiver should ensure that the demodulation process is carried out at the peak of the correlation function.

In order to carry out these tests a transmitter was modified slightly so that the code selected, to modulate the transmitted data, could be delayed in **1/16**th chip bit increments up to **2** complete bits. The coded transmission was then injected at one end of the 100 metres of cable used for the development system transmission line, and the receiver positioned at the opposite end of the line in order that both intentional

* **NOTE** The noise component in the S/N values is meant to include all interference, ie. both co-channel and AWGN.

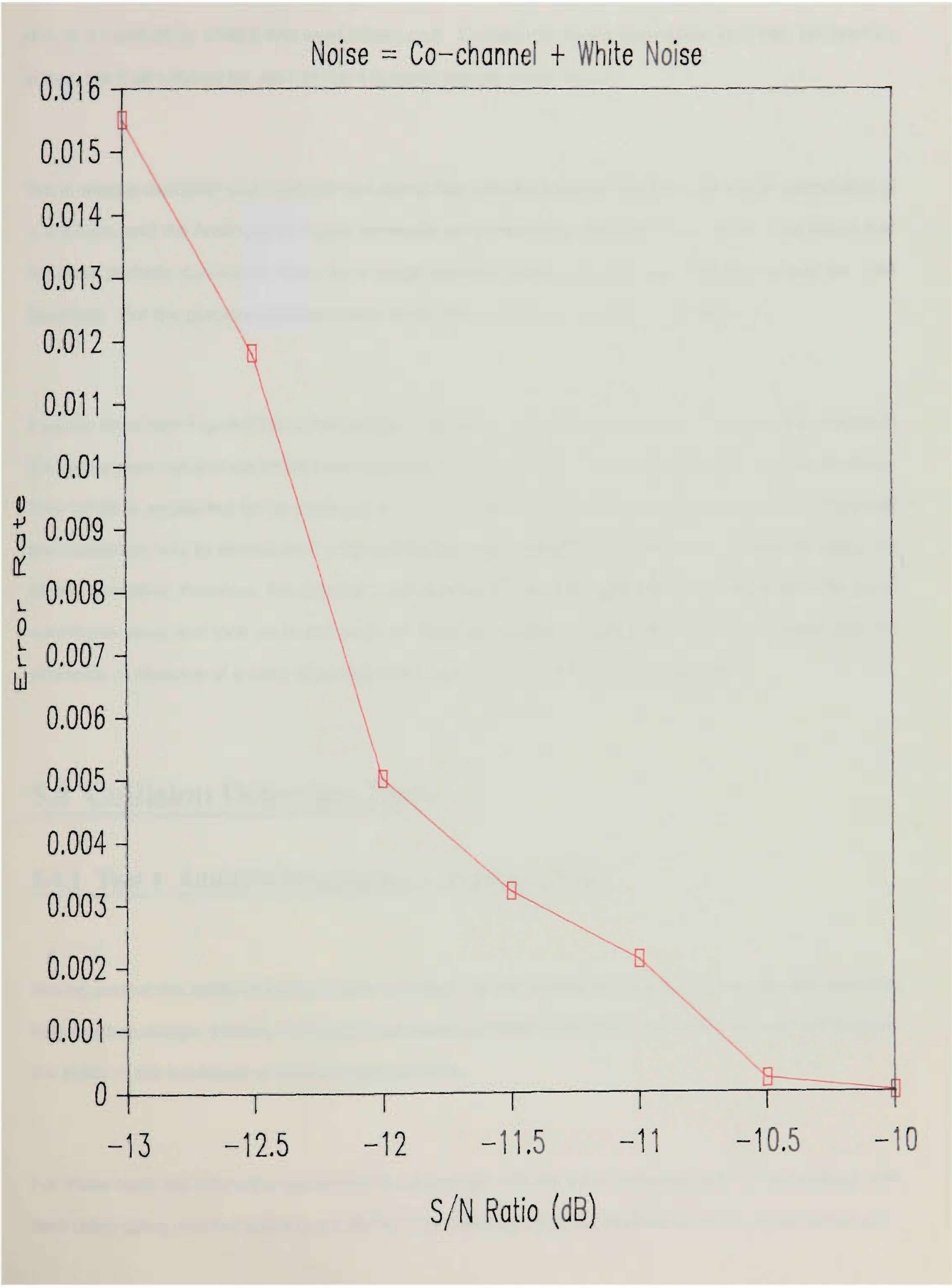


Figure 34 Bit Error Rate vs S/N Ratio

and the propagation delay effects caused by the line would be included in the results. Intentional delays of 1, 9, 21 and 25 (x 1/16th) bits were introduced. Correlation values were taken in **1/16**th bit intervals across the **2** bit window for each of the **4** delayed signals noted above.

The analogue multiplier and integrate and dump filter circuits were adjusted to give a peak correlation of ± **2.5** volts, and the Analogue to Digital converter set to reference voltages of ± **5** volts. This meant that the peak periodic correlation value for a single received signal, over the full 1024 bits, should be ± **64** (decimal). For the purpose of comparison in this test, only magnitudes are considered.

It can be seen from Figure 35 that the correlation output does, indeed, follow the trend indicated in section 3.4, with a peak value of about **64** occurring approximately **1/2** bit past the deliberately introduced delay. This **1/2** bit is accounted for by propagation, clock recovery and processing delays etc. From this peak the correlation falls to almost **zero** when the signals are separated by more than **1** full bit. By using the sliding correlator, therefore, the receiver could determine the delay time which corresponds to the peak correlation value and lock on to this peak for data demodulation purposes. It is also apparent that the presence or absence of a valid signal anywhere in the **2** bit window is easily detected.

5.4 Collision Detection Tests

5.4.1 Test 1 Additive interference (overlap ≤ 1 bit)

Having proved the ability of the hardware to detect signals anywhere in the **2** bit delay window specified in the system design, thereby verifying the concepts outlined in chapter 3, the next step was to determine the ability of the hardware to detect signal collisions.

For these tests the reference transmitter is co-located with the sync generator and is transmitting with **zero** delay using channel address **#1**, **C_1'(t)**. The colliding signal is provided by the modified transmitter,

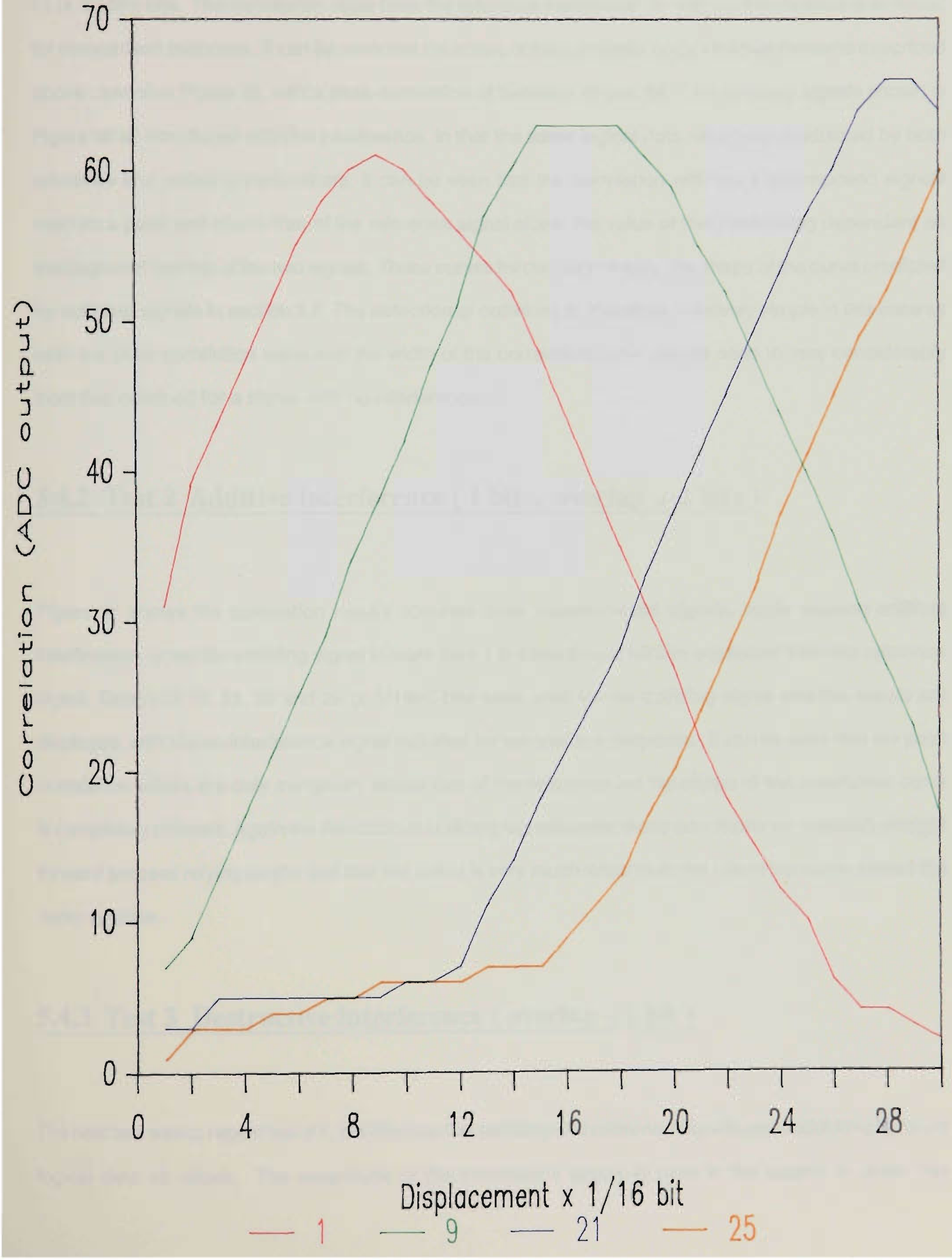


Figure 35 Correlation With Delayed Signals

which generates the same address code with selectable simulated propagation delays, Figure 36 shows the output of the correlator when the colliding transmission is subject to intentional delays of 1, 5, 9 and 13 (**x 1/16**th) bits. The correlation value from the reference transmitter, ie. with no interference is included for comparison purposes. It can be seen that the shape of the correlator output follows the trend described above , see also Figure 35, with a peak correlation of between **60** and **64**. The colliding signals shown in Figure 36 all introduced additive interference, in that the same logical data value was modulated by both reference and colliding transmitters. It can be seen that the correlation with the superimposed signals reaches a peak well above that of the reference signal alone; the value of this peak being dependent on the degree of overlap of the two signals. These curves follow, very closely, the shape of the curve predicted for colliding signals in section 3.5. The detection of collisions is, therefore, relatively simple in this case as both the peak correlation value and the width of the correlation curve can be seen to vary considerably from that obtained for a signal with no interference.

5.4.2 Test 2 Additive interference (1 bit ≤ overlap ≤ 2 bits)

Figure 37 shows the correlation results obtained from superimposed signals, again causing additive interference, when the colliding signal is more than **1** but less than **2** full bits separated from the reference signal. Delays of 17, 21, 25 and 29 (**x 1/16**th) bits were used for the colliding signal and the results are displayed, with the no interference signal included for comparison purposes. It can be seen that the peak correlation values are only marginally above that of the reference but the shape of the correlation curve is completely different. Again the detection of colliding signals under these conditions is a relatively straight forward process relying on the fact that the curve is very much wider than the reference curve across the delay window.

5.4.3 Test 3 Destructive interference (overlap ≤1 bit)

The next test was to repeat test #1, but this time the colliding and reference signals are modulating different logical data bit values. The magnitude of the correlator's output is used in the graphs in order that

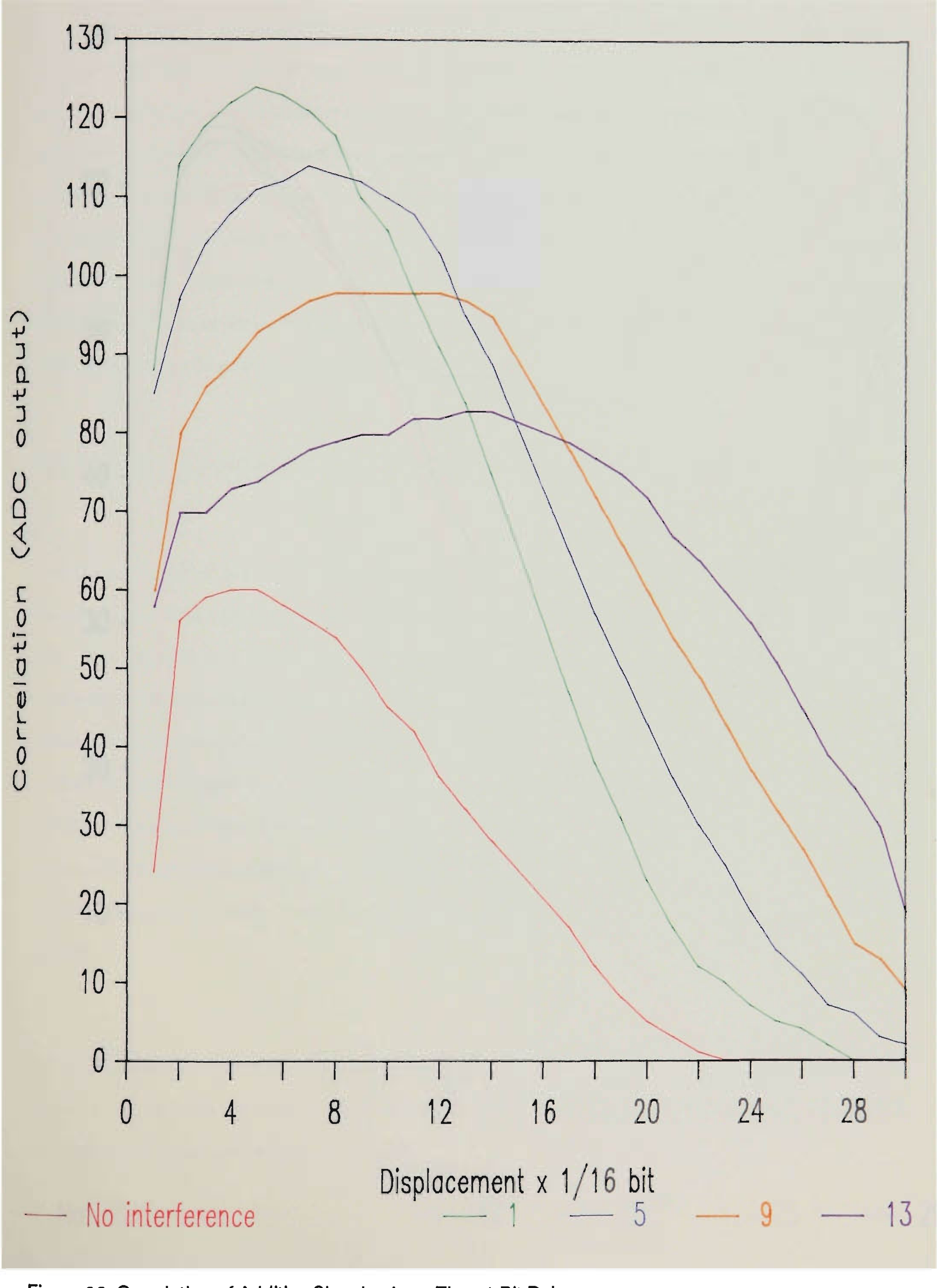


Figure 36 Correlation of Additive Signals - Less Than 1 Bit Delay

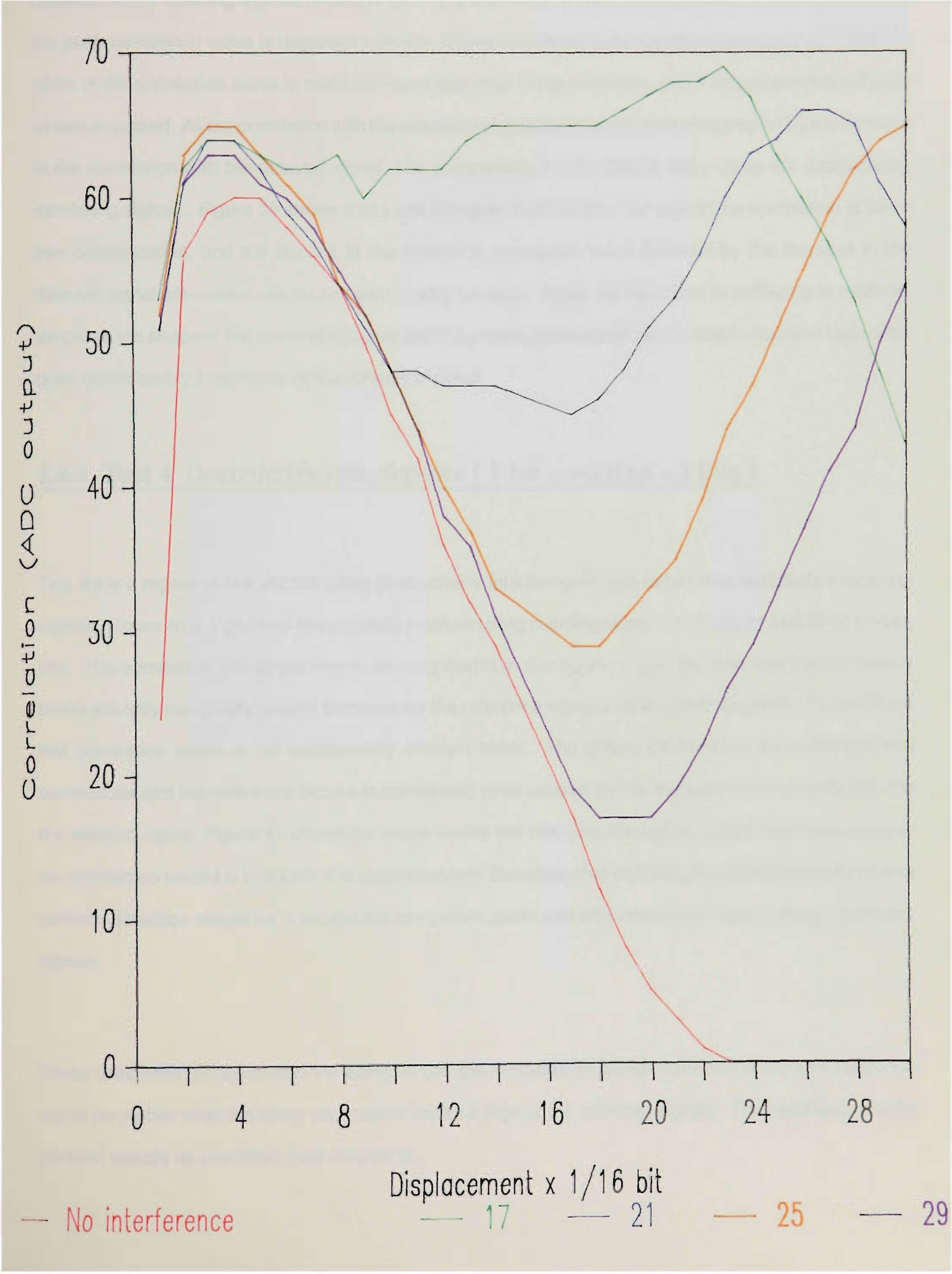


Figure 37 Correlation of Additive Signals - Greater Than 1 Bit Delay

comparisons with the results of the previous tests may be made. Figure 38 shows the correlation results obtained when colliding signals of delays of 1, 5, 9 and 13 (**x 1/16**th) bits are used. It can be seen that the peak correlation value is dependent on the degree of overlap with the reference signal and that the width of the correlation curve is much narrower than that of the reference, and it has also a second peak as was expected. As the correlation with the reference signal declines it is compensated for by the increase in the correlation with the delayed signal. The comparison is not obvious since these are destructively interfering signals. Figure 39 shows these results again, but this time the sign of the correlation is taken into consideration, and the decline in the reference correlation value followed by the increase in the delayed signal correlation value can more readily be seen. Again the detection of collisions is relatively simple as the shape of the correlation curve and the peak magnitude of the correlation function both differ quite considerably from those of the reference signal.

5.4.4 Test 4 Destructive interference (1 bit ≤overlap ≤2 bits)

Test **#4** is a repeat of test **#2** but using destructively interfering signals rather than additively interfering signals. Figure 40 is a graph of the correlation values using colliding signals of 17, 21, 25 and 29 (**x 1/16**th) bits. The correlation values are shown as magnitudes in this figure. It can be seen that the correlation peaks are only marginally greater than that for the reference signal and also that the width of each of the first correlation peaks is not substantially different either. The difference between the superimposed correlations and the reference occurs in the second peak caused by the increase in the correlation with the delayed signal. Figure 41 shows the same results but this time the sign as well as the magnitude of the correlation results is included. It is quite apparent, therefore, that including the sign makes the task of collision detection simple as **+ve** and **-ve** correlation peaks can only come from destructively interfering signals.

These tests have demonstrated the ability to use the hardware to detect collisions with some reference signal no matter what the delay and relative data bit sign of the colliding signals. The hardware circuits perform exactly as predicted (see chapter 3).

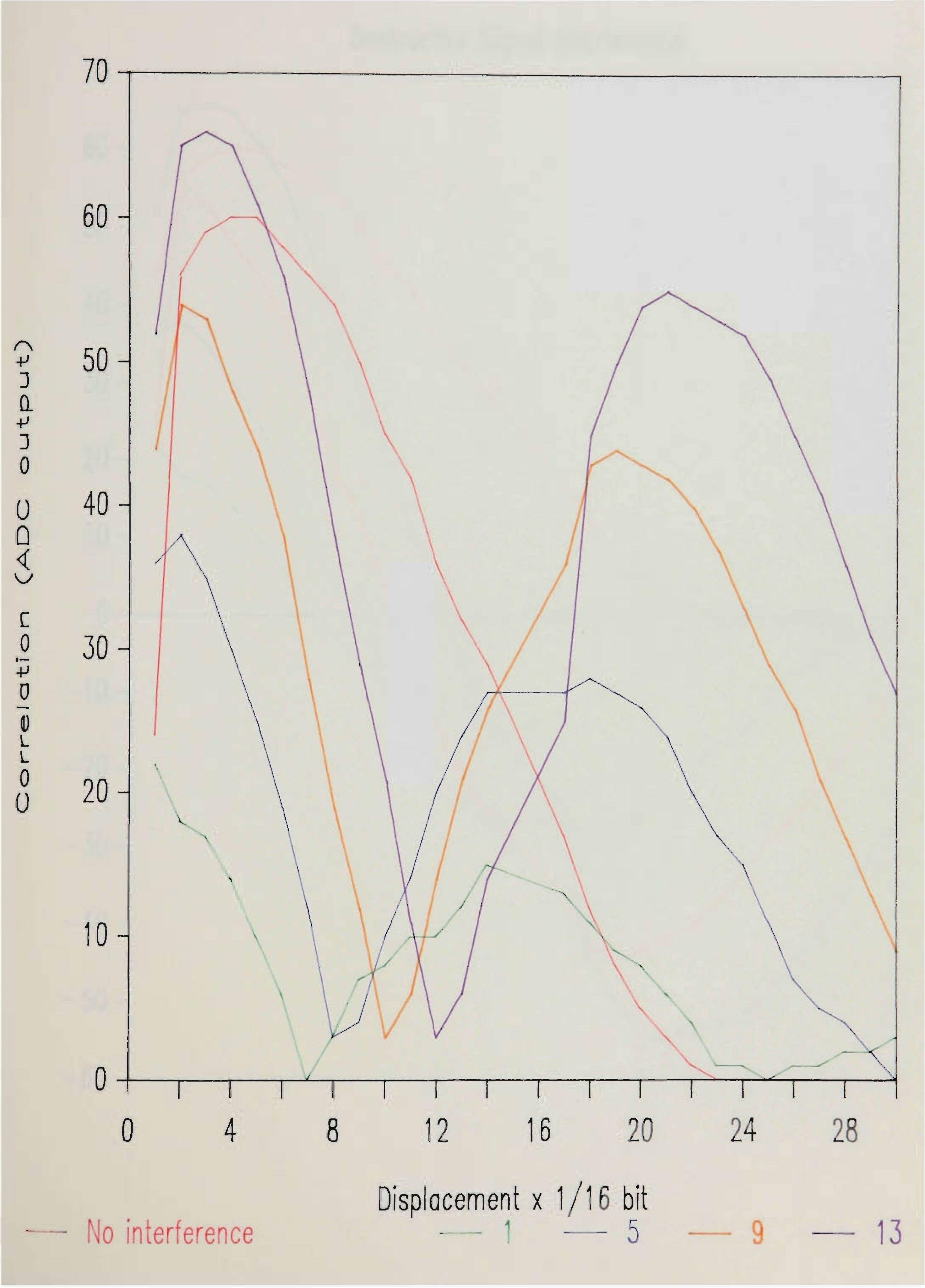


Figure 38 Correlation of Destructive Signals - Less Than 1 Bit Delay

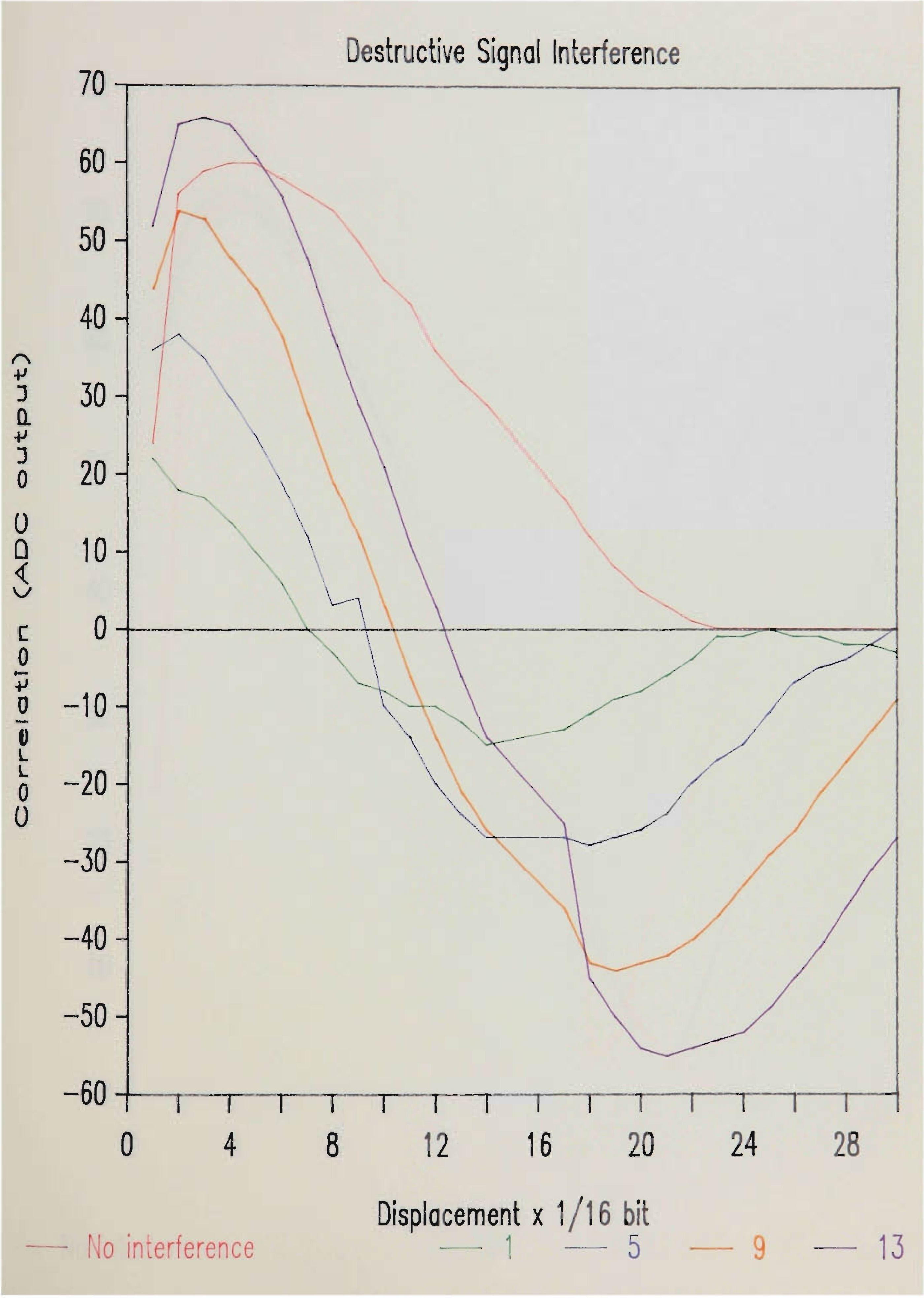


Figure 39 Positive and Negative Correlation - Less Than 1 Bit Delay

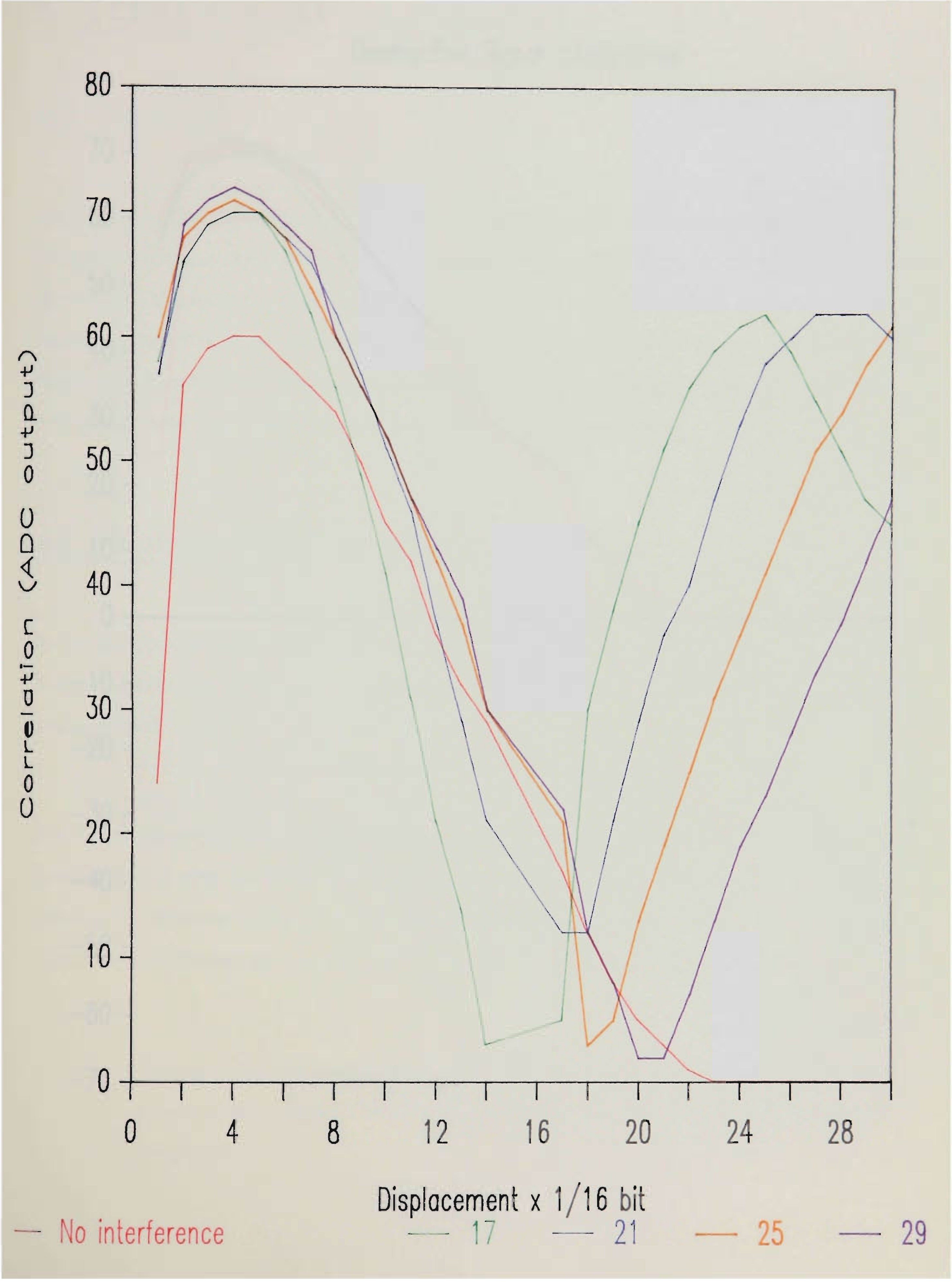


Figure 40 Correlation of Destructive Signals - Greater Than 1 Bit Delay

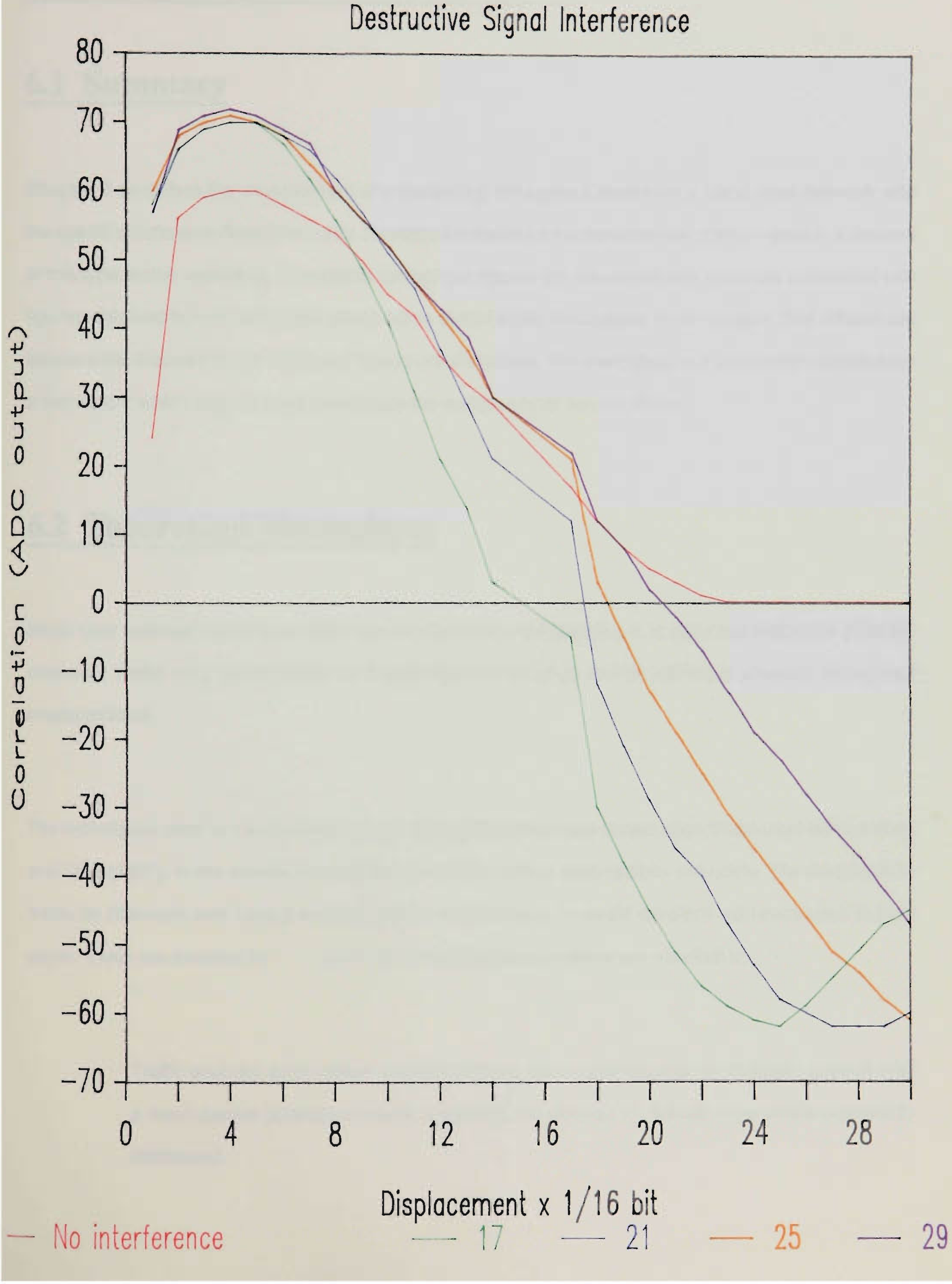


Figure 41 Positive and Negative Correlation - Greater Than 1 Bit Delay

6 THROUGHPUT MODEL

6.1 Summary

Chapter 6 describes the development of a theoretical throughput model for a Local Area Network with the specifications described previously. A protocol suitable for the transmission of information in a network of this type is also explained. Theoretical throughput figures are calculated and these are compared with figures obtained from a simulation designed to model traffic throughput in this system. The differences between the theoretical and simulated figures are discussed. The theoretical and simulation models form a framework which may be used to evaluate the system under test conditions.

6.2 Theoretical Throughput

Since user address codes have been chosen for their orthogonality, it is assumed that each (**CDMA**) channel's traffic may be regarded as independent in order to model individual channel throughput characteristics.

The techniques used in the derivation of this throughput model are based upon those used by Kleinrock and Tobagi [27]. In the development of this model the various assumptions are made. The assumptions made by Kleinrock and Tobagi are included for convenience, to avoid constant reference back to their paper, these are denoted by " >" and new or modified assumptions are denoted by "+"

> Traffic sources, on a system and per channel basis, are regarded as **Poisson** sources with a mean packet generation rate of λ packets per second, i.e. independent and exponentially distributed.

> Packets are of constant length of **T** seconds duration, such that the average number of packets generated in this **T** second period is **S** = λ**T**. Thus the input rate is normalised to the transmission time **T**. **S** may then be called the **Channel Throughput** or **Channel Utilisation**.

> The **Offered Channel Traffic** is denoted by **G** (where **G** $\geq$ **S**). **G** is equal to the **new traffic** plus **the traffic requiring retransmission owing to some earlier collision**.

> $\bar{\mathbf{X}}$ is the **Mean Retransmission**, or **Back-off**, time and is assumed large compared to **T**.

> In this model **non-persistent CSMA** is used where, upon busy sensing and recognition the transmitter will immediately back off an arbitrary time period $\bar{\mathbf{X}}$, and not continue to sense, waiting for the channel to become free.

+ No positive acknowledgement scheme is incorporated, collisions being detected (in a finite time τ) by each station involved in the transmission of the colliding packets. After collision detection, packet transmission will be continued for a further period τ to jam the channel in order to ensure that the other station(s) involved in the collision will actually detect it. The time taken to detect the actual collision plus the jamming time is collectively called the collision detection time. The ratio of the collision detection time τ to the transmission time **T** is given by **a** where **a** = τ/**T**.

+ **Busy Sensing** is achieved in a period equal to the **Collision Detection** time **a**, where **a** is normalised to **T** as shown above.

+ Propagation delays are negligible compared to **T** and are set to zero.

Let **t** be the arrival time, at a node, of a packet to be transmitted. The station will sense the channel for **a** seconds and, sensing the channel free, will transmit immediately ie. at time (**t** + **a**), as shown in Figure 42. * Another station obtaining a packet between **t** and (**t** + **a**) will sense the channel free and also transmit, thus causing a collision. This period of **a** seconds is classified as the "**Vulnerable Period**", and if no other station receives a packet for transmission in the period between t and (t + a) , then the first's packet will be successfully transmitted.

If (**t** + **Y**) is the time of arrival of the last packet (where $\mathbf{0} \leq \mathbf{Y} < \mathbf{a}$), and (**t** + **Y** + **a**) is the time of transmission of this packet, then at time (**t** + **Y** + 2**a**) stations transmitting should have recognised the collision. Each station will continue to transmit for a period of **a** seconds after collision detection, thus at time (**t** + **Y** + 3**a**) all stations will have stopped transmitting. Since no transmissions are occurring during the first **a** seconds of this period, while the first station is sensing the channel, the average duration of an **unsuccessful transmission** is ($\overline{\mathbf{Y}}$ + 2**a**), and the duration of a **successful transmission** is equal to **1**. The time when transmissions are occurring is defined as that time when traffic is actually flowing and excludes the initial sensing period. This transmission time, whether for a successful or an unsuccessful transmission, is called the **Transmission Period** (**TP**) or **Busy Period**. Any period between two successive **TP**s is called an **Idle Period**. A Busy Period followed by an Idle Period is called a **Cycle**. If $\overline{\mathbf{B}}$ is the expected duration of a Busy Period, and $\bar{\mathbf{I}}$ the expected duration of an Idle period then the duration of a cycle would be $\overline{\mathbf{B}} + \bar{\mathbf{I}}$. Using renewal theory, the average **Channel Utilisation**, **S**, may then be defined as

$$S = \frac{\overline{U}}{\overline{B} + \bar{I}} \tag{24}$$

* **NOTE** Upon commencement of transmission the busy sensing circuit is reset and re-activated as a collision detection circuit.

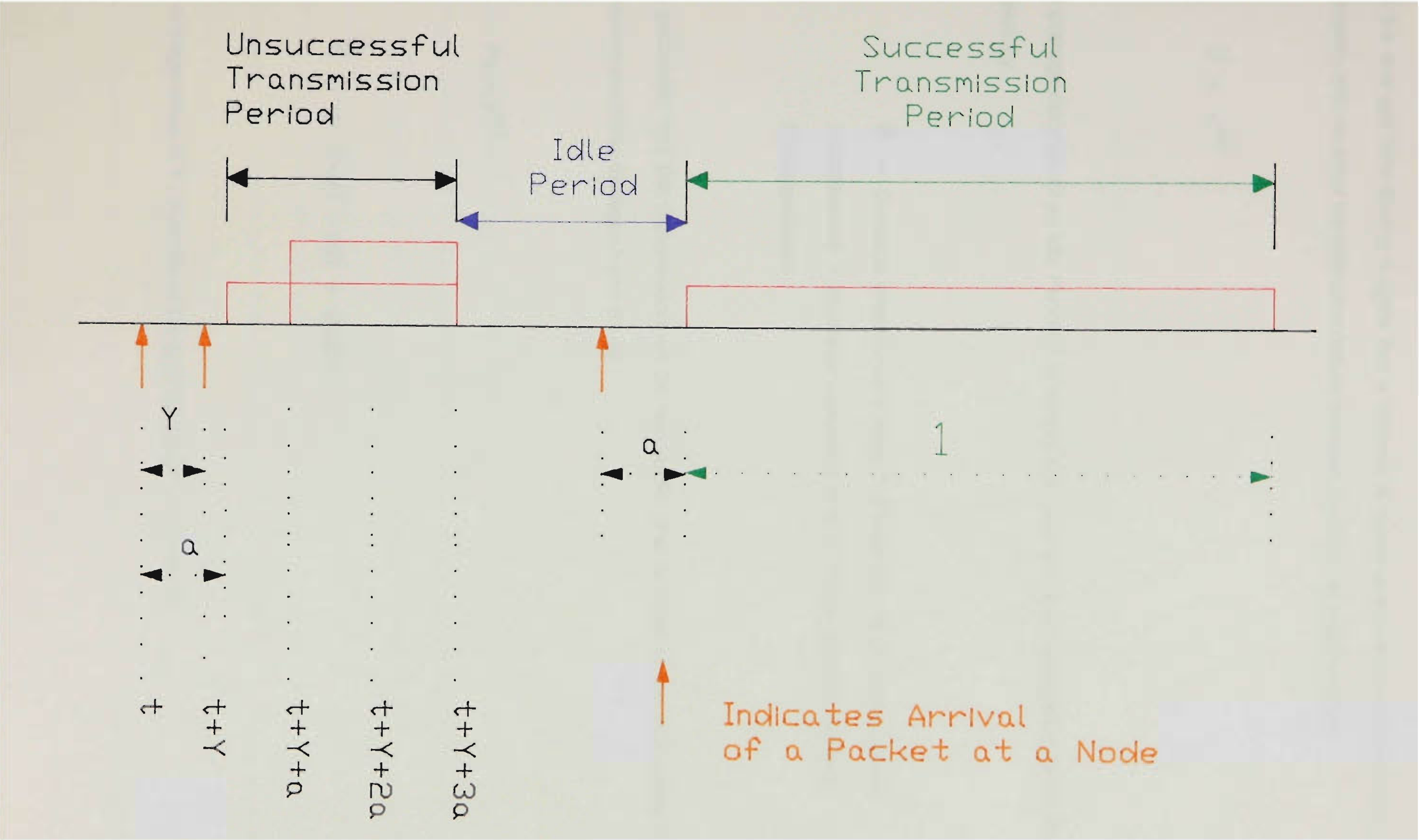

Unsuccessful Transmission Period
Idle Period
Successful Transmission Period
Y
a
a
1
t
t+Y
t+Y+a
t+Y+2a
t+Y+3a
Indicates Arrival of a Packet at a Node

Figure 42 Channel Transmission Characteristics

$\overline{\mathbf{U}}$ is the average time during a cycle that a channel is being used without conflict. $\overline{\mathbf{U}}$ is given by the probability that no other transmission occurs between **t** and (**t** + **a**) and therefore

$$\overline{U} = e^{-aG} \tag{25}$$

The average duration of an Idle Period, $\bar{\mathbf{I}}$, is simply **1/G** , and the average duration of the Busy Period $\overline{\mathbf{B}}$ is given by

$\overline{\mathbf{B}}$ = (Average unsuccessful time) * (Probability of an unsuccessful transmission) + (Average successful time) * (Probability of a successful transmission)

The probability that the transmission will be successful (**Ps**) is equal to the probability that no other transmissions occur between **t** and (**t** + **a**).

$$Ps = e^{-aG}$$

$$\therefore \quad \overline{B} = (1 - Ps)(\overline{Y} + 2a) + (Ps)(1) \tag{26}$$

The average value of **Y** , from Kleinrock and Tobagi [27], is given by

$$\overline{Y} = a - \frac{(1 - e^{-aG})}{G} \tag{27}$$

Substituting into equation (26) gives

$$\overline{B} = 3a - \frac{(1 - e^{-aG})}{G} - 3ae^{-aG} + e^{-aG} + e^{-aG}\frac{(1 - e^{-aG})}{G} \tag{28}$$

Using the values for **I**, **U** and **B** and substituting into equation (26) gives

$$S = \frac{\overline{U}}{\overline{B} + \overline{I}} = \frac{Ge^{-aG}}{(3aG + (e^{-aG}(2 + G - 3aG - e^{-aG})))} \tag{29}$$

Equation (29) above represents the throughput, **S**, in terms of the **Offered Traffic**, **G**, and the **busy sensing period**, **a**. Figure 43 is a graph of **S** vs **G**, and the curves represent different values of **a**. These curves show the expected characteristic shape with the throughput increasing as the offered traffic increases until a breakpoint is reached. Above this point the probability of collisions occurring is such that increasing the offered traffic actually causes a reduction in throughput as the channel capacity is progressively taken up with more re-transmissions and less new traffic. It can be seen from Figure 43 that the busy sensing time is a highly important parameter in the throughput model and that reducing the sensing time causes the breakpoint in throughput to occur at higher levels of offered traffic. This point is highlighted by the list of maximum throughput figures for each of these curves, which is included as Table 2. It shows that an increase in peak throughput corresponds to a decrease in the value of the vulnerable period **a**. The table also includes the offered traffic at which the peak occurs.

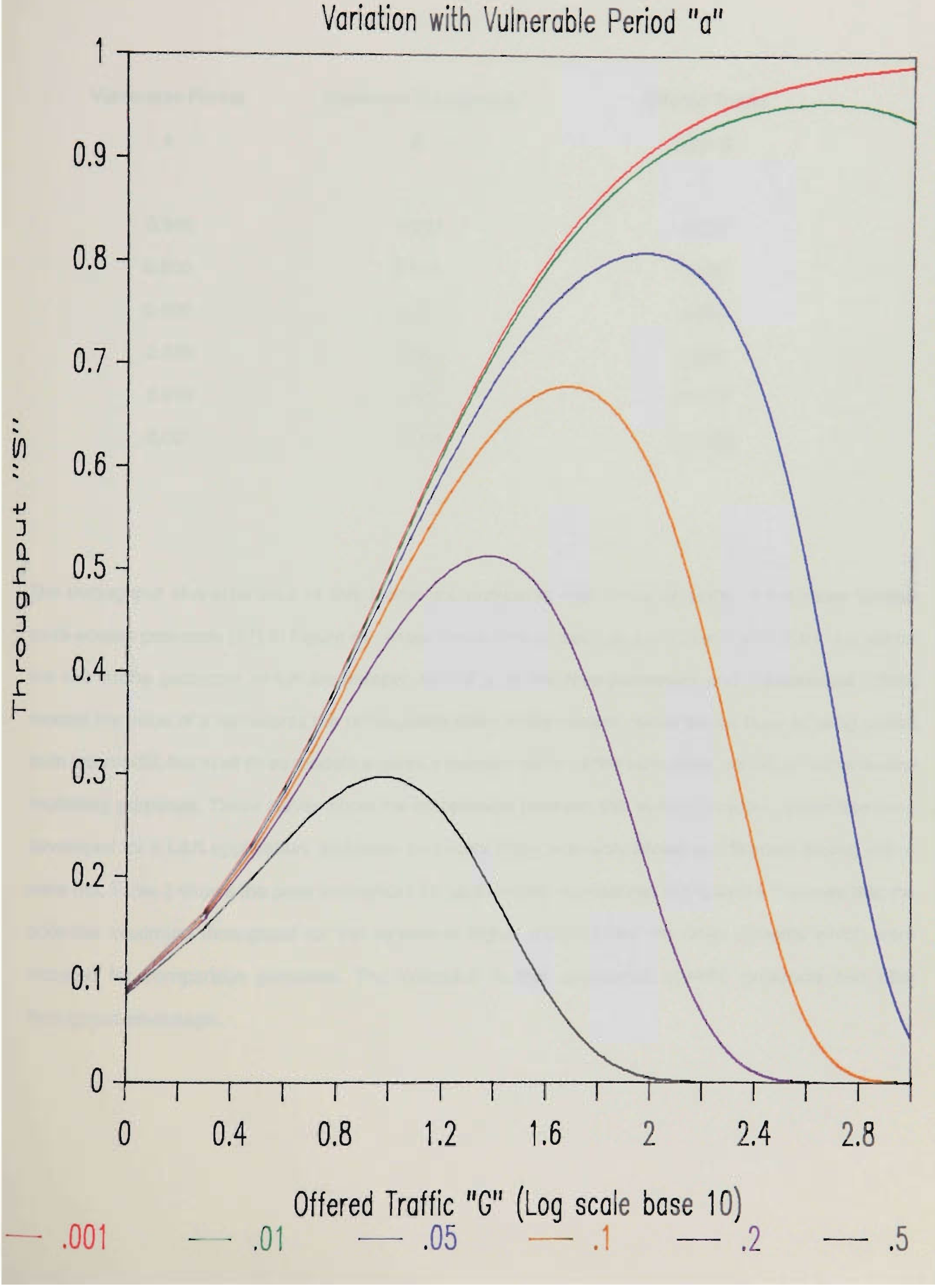


Figure 43 Throughput vs Offered Traffic

Table 2 Peak Throughput as a Function of Vulnerable Period

Vulnerable Period a	Maximum Throughput S	Offered Traffic max G
0.500	0.297	0.900
0.200	0.514	2.400
0.100	0.697	4.700
0.050	0.809	9.500
0.010	0.955	47.400
0.001	0.989	100.000

The throughput characteristics of this model are compared with those of some of the more familiar multi-access protocols [27] in Figure 44. These curves are plotted using a value of **a** = **0.01**, except for the two **Aloha** protocols which are independent of **a**. In the **Non-persistent** and **1-persistent** CSMA models the value of **a** represents the propagation delay in the system, rather than a busy sensing period as in this model, but in all three models **a** gives a representation of the vulnerable period for transmission modelling purposes. These curves show the comparison between this system's model, which has been developed for a **LAN** application, and other protocols (most noticably **Aloha** and **Slotted Aloha**) which were not. Table 3 shows the peak throughput for each model represented in Figure 44. It shows that the potential maximum throughput for this system is higher than that for the other systems which were included for comparison purposes. The indication is that application specific protocols can offer throughput advantage.

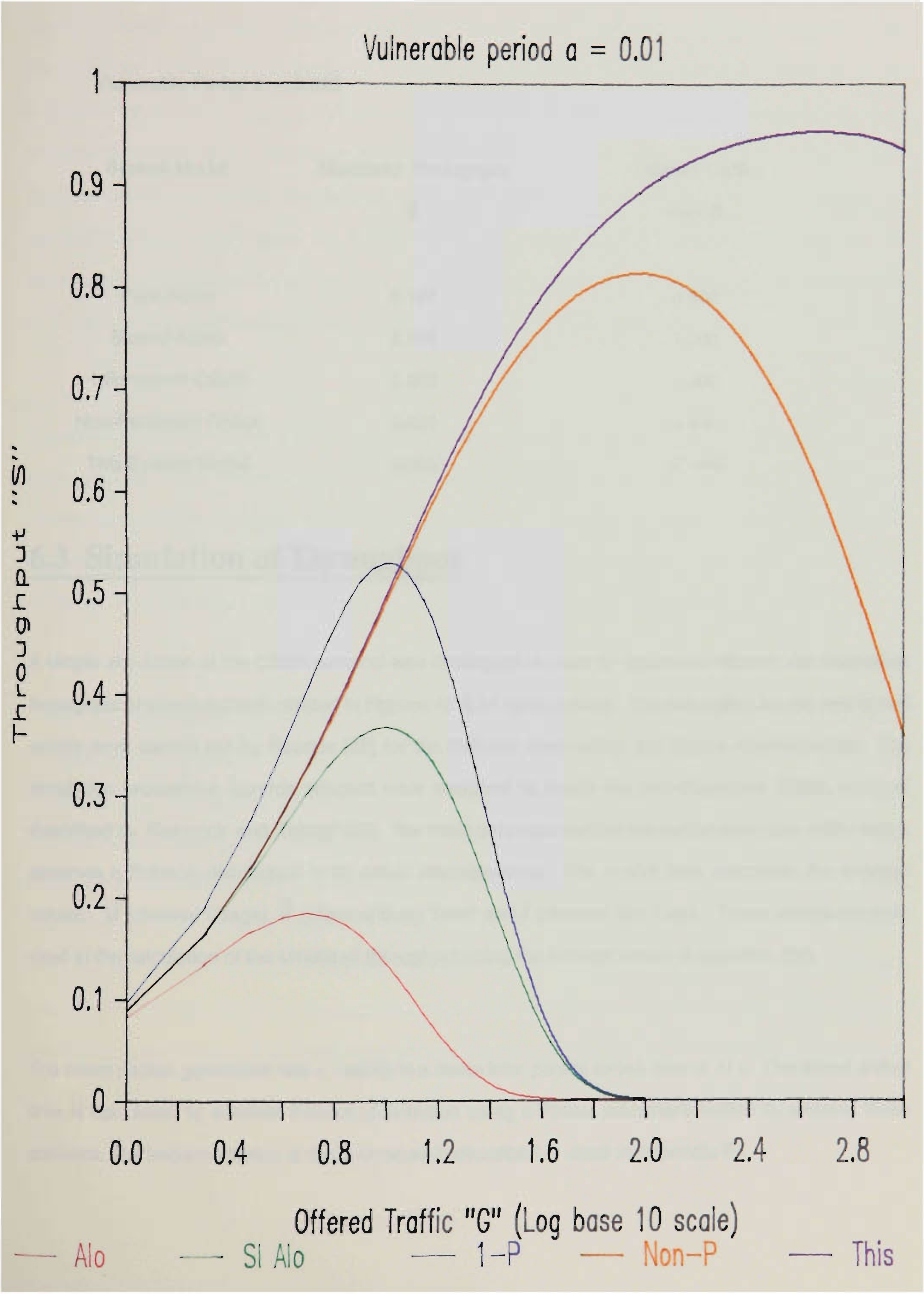


Figure 44 Comparison of Throughput Characteristics

Table 3 Peak Throughput as a Function of System Model

Vulnerable Period **a** = 0.010

System Model	Maximum Throughput S	Offered Traffic max **G**
Pure Aloha	0.184	0.500
Slotted Aloha	0.368	1.000
1-Persistent CSMA	0.529	1.000
Non-Persistent CSMA	0.815	9.400
This System Model	0.955	47.400

6.3 Simulation of Throughput

A simple simulation of the CSMA protocol was developed in order to determine whether the theoretical throughput characteristics illustrated in Figures 43 & 44 were realistic. The simulation model was based on the work carried out by Rounds [28] for the National Aeronautics and Space Administration. The simulation procedures Rounds adopted were designed to model the Non-Persistent CSMA protocol described by Kleinrock and Tobagi [27]. The basic concepts are that the model generates traffic which assumes a Poisson distribution in its arrival characteristics. The model then calculates the average values:- $\bar{U}$ (channel Usage), $\bar{B}$ (channel Busy Time) and $\bar{I}$ (channel Idle Time). These values are then used in the calculation of the simulated throughput using the formula shown in equation (24).

The mean packet generation rate λ results in a mean inter packet arrival time of $1/\lambda$. The actual arrival time is calculated to simulate Poisson distribution using a Pascal procedure written to emulate these statistics. The implementation of this procedure is described in detail in Appendix F.

If the arrival time is less than the vulnerable period **a** then the packet is marked as a collision, and the Busy Period set to the unsuccessful transmission period values (see Figure 42). If the arrival time is greater than the Busy Period then the packet is successful and the Busy Period and Usage values set to the successful transmission values (see Figure 42). The Idle period is then the difference between the busy period and the arrival time of the latest packet. If a packet arrives after the vulnerable period but before the Busy period has expired, then its busy sensing circuitry is assumed to detect that the channel is in use and that packet is rejected, i.e. it causes neither a collision nor a successful transmission. This simulates re-scheduling of the packet with a mean re-transmission delay, $\overline{\mathbf{X}}$, tending towards infinity. The **U**, **B** and **I** values are then averaged before being used to calculate the **Simulated Throughput, ST**.

The specifications of Rounds' model had to be modified in order to simulate the protocol described above rather than the Non-Persistent protocol. Parameters relating to channel usage were modified and the concepts relating to Propagation Delay altered to suit the Busy Sensing period implemented in the protocol for this system. Rounds' program was completely redesigned to accomodate these changes and then re-written in Pascal. This simple simulation program is included in Appendix D.

Table 4 shows the output of a typical simulation run. The figures produced represent the calculated values of **Theoretical Thoughput, TS**, (calculated using the formula shown in Equation (18)), the **Simulated Throughput, ST**, the **Offered Traffic, G**, in terms of traffic offered per packet transmission period (T), and also the number of collisions detected, the number of packets arriving during channel busy periods and, finally, the number of packets being successfully transmitted. The packets arriving during a channel busy period are called "**CLASHES**", and are subject to the large re-scheduling delays discussed previously. The busy period is set to a value of **(1 + a)** to represent the channel occupancy plus sensing time, thereby ensuring the inter-packet gap necessary for this protocol. The program cycles through **50** iterations for every offered traffic rate, so that the number of collisions represents collisions per **50** traffic packets offered.

Table 4 Simulated & Theoretical Throughput Figures

Theoretical	Simulated	Offered	Collisions	Clashes	Successes
0.091	0.091	0.1	0	0	50
0.166	0.166	0.2	0	0	50
0.284	0.284	0.4	0	1	49
0.441	0.441	0.8	0	11	39
0.495	0.496	1	0	18	32
0.658	0.659	2	0	31	29
0.818	0.791	5	0	41	9
0.888	0.865	10	0	45	5
0.910	0.892	15	0	46	4
0.919	0.883	20	0	47	3
0.902	0.818	50	16	24	10
0.886	0.731	60	33	7	10
0.865	0.610	70	36	8	6
0.841	0.414	80	47	0	3
0.841	0.324	80	46	2	2
0.781	0.000	100	50	0	0

The Theoretical and Simulated throughput values are plotted against the Offered Traffic in Figure 45. It can be seen that at low levels of offered traffic the Simulated and Theoretical values are very similar. The simulated throughput tends to show signs of reaching a peak at much lower levels of offered traffic, however, as the effects of collisions and packet rejection begin to accumulate. The reason for this discrepency is that the theoretical model values are calculated using steady state conditions and mean packet inter-arrival times whereas the simulated values attempt to take into consideration the potential for "**bursty**" traffic, which is more realistic in practical local area networks.

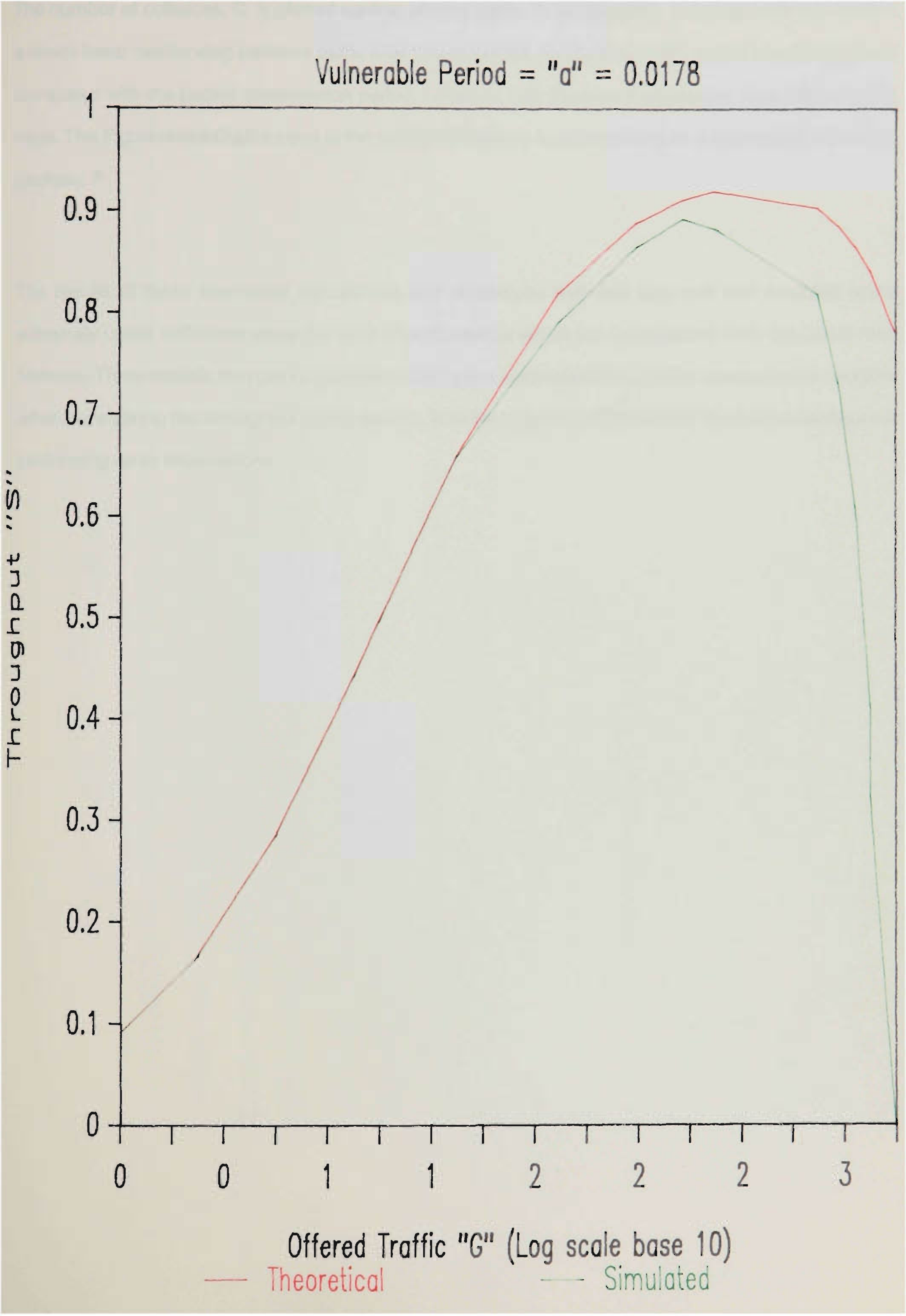


Figure 45 Theoretical and Simulated Throughput

The number of collisions, **C**, is plotted against offered traffic, **G**, in Figure 46. It can be seen that there is a direct linear relationship between traffic load and collisions. As the vulnerable period is relatively small compared with the packet transmission period, collisions only become a problem at high offered traffic rates. This Figure also includes a plot of the number of clashes, **L**, and the number of successfully delivered packets, **P**.

The results of these theoretical calculations and simulations matched very well and provided some extremely useful indicators about the level of performance which can be expected from this Local Area Network. These models also gave a guide as to the type of tests to perform and the parameters to measure when considering the throughput of this system, in order to gain confidence that the system hardware is performing up to expectations.

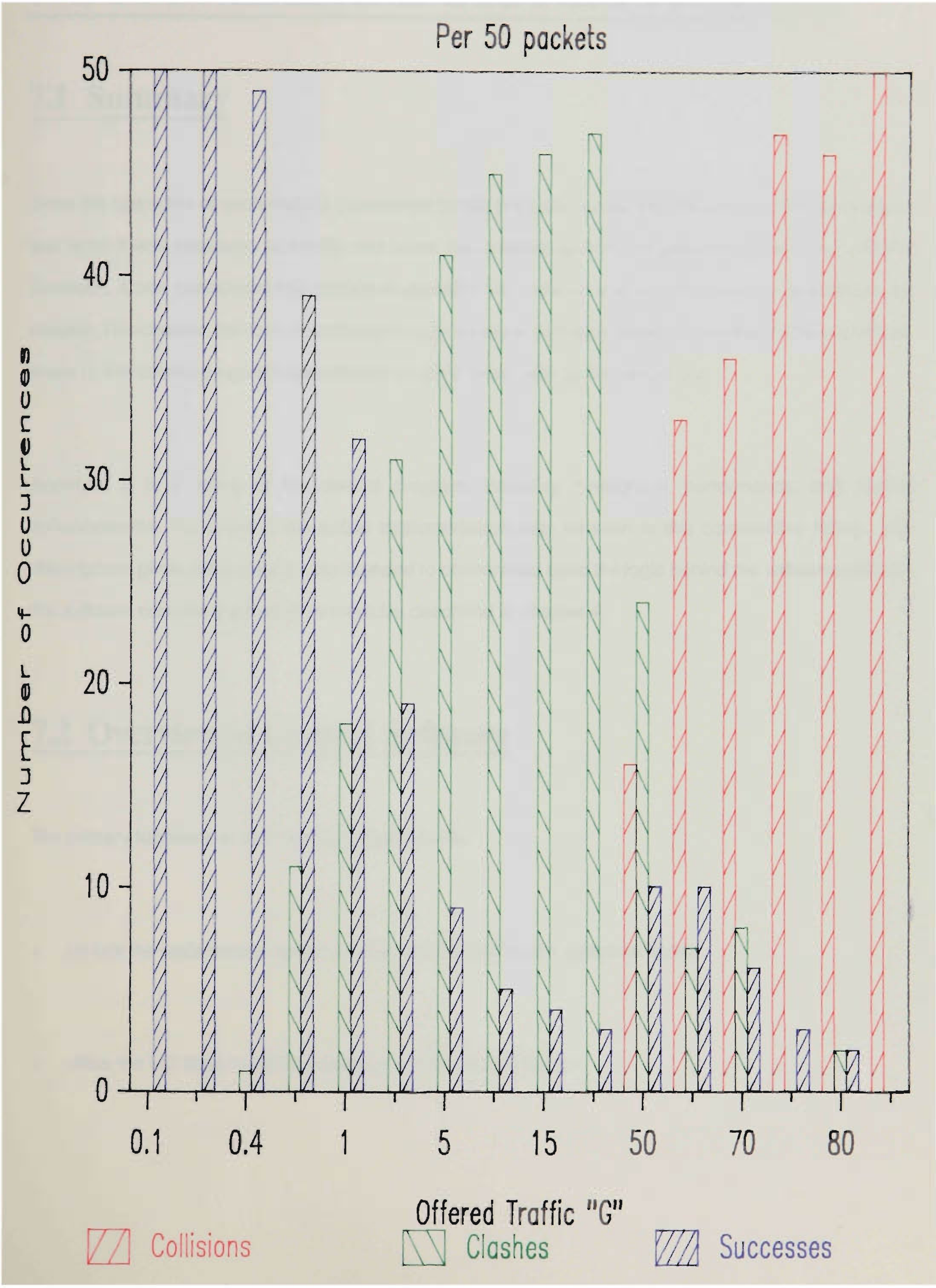


Figure 46 Number of Collisions vs Offered Traffic

7 SOFTWARE AND SYSTEM TESTING

7.1 Summary

Since the operation of each node is determined by the program loaded into the control microprocessor, and since it was necessary to modify and adapt the standard operational program to insert special test functions, it was considered appropriate to describe the software and system testing in one integrated chapter. This chapter, therfore, describes the node software, test layouts and procedures and the additions made to the standard operational software to allow these tests to be carried out.

Appendix E is a listing of the control program, including throughput, performance, and testing enhancements. The detail of the actual implementation may be seen in this commented listing. The descriptions given in this chapter are intended to concentrate upon the logic behind the software and how the software controls the hardware modules described in chapter 4.

7.2 Overview of Control Software

The primary functions of the control software are to:

- control the node spread spectrum interface i.e. ISO level 1 (physical layer).

- utilise the MC 6854 to fulfil the Data Link protocol ISO level 2.

7.3 Control of the Spread Spectrum Interface

The basic concept for the software is that everything is related to the time frame of a single 1200 Hz clock pulse. During one such time period all functions relating to **SS** interface input/output, HDLC input/output and Host or Test interface input/output must be serviced. With this as the prime directive no input can be missed neither can any opportunity to transmit be missed. The whole system is, thus, designed on a polling basis with every input or output being tested once and once only during each 1200 Hz clock pulse.

7.3.1 Inputs

Tranmissions intended for a particular node are modulated with the code for that particular receiver. Because of propagation delays etc. the transmission may appear at the receiver anywhere within a 2 bit window. The receiver must continuously scan across this window looking for such transmissions. The first function of the software is to control the scanning of the receiver whilst it is looking for a valid signal. This is done by issuing a 5 bit output signal, PB0 to PB4 on VIA **#1**, to control the receiver sliding correlator. The received signal is then modulated with the delayed locally generated signal and integrated over a 1024 chip bit period (ie. one 1200 Hz bit period). This value is sampled and held before being input to the ADC 800. One ADC 800 is used and the input of Receiver Sample and Hold or Transmitter Sample and Hold is selected using a single bit PB5 of VIA **#1**.

Since the Sample and Hold circuits track whilst the 1200 Hz clock is low and hold while it is logic 1 the software waits until PA4 on VIA **#2**, ie. the 1200 Hz clock is a logic 1 (see also Figure 25, section 4.4.7). The receiver signal is selected and a Start Conversion (**SC**)pulse sent to the ADC 800 using the CA2 output of VIA **#2**. The End Of Conversion (**EOC**) signal from the ADC should occur in a maximum of **65.1 ms** (see section 4.4.7.). The software waits for this EOC response on the CA1 input of VIA **#2** before reading the converted value using PA0 to PA7 of VIA **#1**.

It has been noted in the past that some 8 bit A/D convertors occasionally fail to respond to a **SC** pulse. For this reason an anti lock-out mechanism has been designed into the ADC read routine. When the **SC** pulse is generated, timer 1 in VIA **#2** is loaded with a value of **80** and instructed to decrement at a 1 MHz rate. If the **EOC** pulse fails to arrive before the counter reaches **0** (i.e. **80 ms**) a second **SC** pulse is generated and the whole cycle repeated.

The input from the ADC is then checked to see if it is in the range allowed for valid signals. With magnitude values ranging from **0** to **127** (the MSB being a sign bit) the range **0** to **15** is designated invalid as it could be due to interference. The range **80** to **127** is designated as invalid as it can only have been generated by colliding signals (see section 6.4). The range **16** to **79** is designated to represent valid signals. This validation is achieved by means of a look-up table, so the various ranges can be modified to suit conditions if necessary.

If the signal for this particular delay position is invalid no further action is taken and the delay counter incremented for the next sample. If the bit is valid then the magnitude of the correlation peak is saved so that the correlation characteristics can be examined. This is done so that the receiver may be automatically tuned to lock onto the peak correlation delay position for reception of this transmission.

Once the correlation peak is found and the receiver locks-on, the scanning stops and each received data bit is checked for validity as described above. Every single bit in a received message is validated in this manner.

7.3.2 Outputs

When the transmitter section has a packet to transmit, the signal will be modulated with the code for the intended recipient. This addess code is selected using a **3** bit address which selects one of **8** destination addresses using PB5 - PB7 of VIA **#2**.

Having selected the destination address, the first task is to scan the allowable 2 bit window for existing signals with the same destination's address code. This is achieved in a similar manner to that described for the receiver section but the transmitter sliding correlator delay is controlled using bits PB0 - PB4 of VIA **#2**. The ADC is then reset to look at the receiver sliding correlator. If no other transmissions are detected after scanning across the 2 bit window, then the transmitter line driver is enabled using bit PB7 of VIA **#1** and the transmitter sends a **64** data bit preamble prior to enabling the ADLC transmitter. This ensures that any reciever will be guaranteed to be able to lock-on to the correlation peak of this transmission before the ADLC sends its opening flag (see section 7.4.2.).

When the transmitter has detected a vacant channel and commenced its own transmission, the transmitter sliding correlator is locked-on to the peak of the transmitters own signal and the transmitters output is monitored to ensure that no other transmitter has commenced transrnissions and has caused a signal collision. Data validation is carried out as described in section 7.3.1. When a collision is detected the transmitter sends a jamming signal so that every node can detect the collision. The actual protocols used, in the event of a channel being identified as busy or a collision being detected, may require one of two actions to be taken:

- Back off a random time which is long compared with the packet transmission period.

- Back off a random time which is short compared with the packet transmission period.

Both have their relative merits and are discussed in the testing part of this chapter in sections 7.6 and 7.7.

In the absence of busy or collisions, at the end of the preamble transmission the ADLC transmitter is re-programmed as a data transmitter and the ADLC transmitter control software activated.

7.4 ADLC Control

The "**Data Link**" protocol is essentially the High Level Data Link Control (**HDLC**). The message (or packet) is defined in the following manner:-

i) Opening Flag (8 bits)

ii) Address Field (8 bits)

iii) Control Field (8 bits)

iv) Data Field (up to 200 bytes of 8 bits/byte)

v) Cyclic Redundancy Check (**CRC**) (16 bits)

vi) Closing Flag (8 bits)

7.4.1 Receiver Control

Having established physical, logic level communications, with the receiver locked-on to the signal, the ADLC (MC 6854) receiver section is reset and the recovered data signal routed to the ADLC for the next level of protocol checking.

The receiver section of the ADLC checks the received data signal bit by bit looking for an opening flag. The detection of an opening flag indicates the acquisition of data synchronisation. The 8 bits following the flag is the address field. In conventional systems this is normally used to contain the destination station's address, but since, in this system, the destination address is defined by the modulation code used, this field is used to convey the identity of the originating node. The control field is normally used to convey data link control information but, again, this is not necessary and is left vacant in this system. The next field is the, variable length, data field which is set to a maximum of **200** bytes, for convenience, in this system. Transfer of data into the "**Last Data Address**" of the ADLC causes a **CRC** of **16** bits to be sent immediately after the last data byte. This **CRC** is then followed by a closing flag.

Detection of the closing flag at the receiver causes the previous **16** bits to be checked for **CRC** information against the data accumulated for this packet. Verification of the **CRC** causes the ADLC to set its Frame Valid status bit, otherwise the Error bit is set. All receiver status bits are tested and the results made available for appropriate responses to be made by the next higher level of protocol.

Upon completion of a frame the software is in a position to cross check the Frame Valid status output from the ADLC against the individual bit validation obtained from the bit level interface to the SS system. This double level of checking may be used to convey added confidence in the integrity of the received data.

Should the signal connection become broken, for some reason, or the transmitter detects a collision and prematurely ceases transmissions then an abort signal will be received. This event is detected by the system and the fact made available to the next level software for appropriate action to be taken.

Since this system is capable of simultaneous reception and transmission, communication between the transmitter and receiver sections is essential. When the receiver has collected a new ADC value it looks to see if there is a transmission in progress, and, if so, switches the ADC to accept a signal from the

Transmitter sliding correlator by setting bit A5 of VIA **#1**, in order to prepare the system to read from the transmitter section.

7.4.2 Transmitter Control

On completion of the synchronisation preamble the ADLC transmitter is reset and the first byte of the packet, ie. the originator's address, is loaded into the Frame Continue **FIFO** of the ADLC. This causes an opening flag, followed by the address field information, to be sent. The status register is checked to determine if the transmitter data register is empty. Each data bit transmitted is monitored, as previously described, to ensure that colliding signals do not interfere with the transmission. Detection of a collision results in the transmission of a jamming signal (which is the Abort signal), and resetting of the transmitter section of the ADLC, followed by the re-start of transmitter scan procedures.

In the absence of collisions the final byte of the packet being sent is loaded into the Frame Terminate address of the ADLC. Putting a data byte into this address, rather than the Frame Continue address, causes the ADLC to automatically send a **16** bit **CRC** after the data, this is then followed by the transmission of a closing flag.

When the whole packet has been successfully transmitted the ADLC transmitter is disabled and, if there is another packet to be sent, the whole scan process re-started using the new destination's address code.

7.5 Connection Testing

The incorporation of the ADLC allows inter node communication testing to be conducted with relative ease. With three stations (address codes $**20**, $**40** and $**60**) node numbers **#1**, **#2** and **#3**, the following test was conducted. The node control software was modified slightly such that when a packet was received, the originator's address was removed and replaced with the receiving node's address, the packet was then transmitted to the third node such that a complete loop was formed. The instigator of the loop counted the responses and stopped after **100** packets had passed round the loop. Passing the same packet round the loop meant that the integrity of the data could be validated at each node.

The transmission time of a packet is made up of the following components:-

i) A 32 bit channel occupancy scan

ii) A 64 bit sync preamble

iii) An 8 bit opening flag

iv) 16 bits of address and control information

v) 200 bytes of 8 bits of data = 1600 bits

vi) 16 bits of CRC

vii) An 8 bit closing flag

The total number of bit periods required to send a packet is, therefore, **1744** bits, which at a data rate of 1200 Hz corresponds to a packet transmission time of **1.45 seconds**. Since each packet needs three transmissions to complete the loop and there are **100** packets, the total time for this test was **7 minutes 15 seconds**. Repeating the test **10** times resulted in no errors being recorded at any of the three nodes.

7.6 Throughput Performance Testing

Whilst the results of the previous tests were excellent, these tests were not considered comprehensive enough to qualify as system performance tests. It was decided to develop tests to attempt to measure the individual channel throughput and compare the measured results with the theoretical and simulated results described chapter 6.

7.6.1 Test Configuration and General Procedures

Three separate tests were carried out to verify the predicted performance characteristics. Figure **47** shows the equipment configuration adopted during these tests. node **#1** was designated the receiver, node **#2** was the interference generator and node **#3** was the traffic generator.

The value used for the packet transmission time was modified from that described in Section 7.5. The reason for this is that the ADLC uses a bit oriented protocol to establish data transparency and this involves the addition of logic zeros to ensure no data transmission replicates flag or abort signals etc. This "**bit stuffing**" means that more bits will be sent than is absolutely warranted to represent the data. This is obviously data dependent and so value of **1800** was used for the total number of bits per packet. This represents only a **3.2%** overhead on the absolute minimum value of **1744** quoted earlier, and was considered quite reasonable. This value of **1800** then represents a packet time of **1.5 seconds**.

The packet arrival times at the nodes is assumed to conform to **Poisson** distribution (see chapter 6). It was, therefore, necessary to produce a random number generator which produces values with this

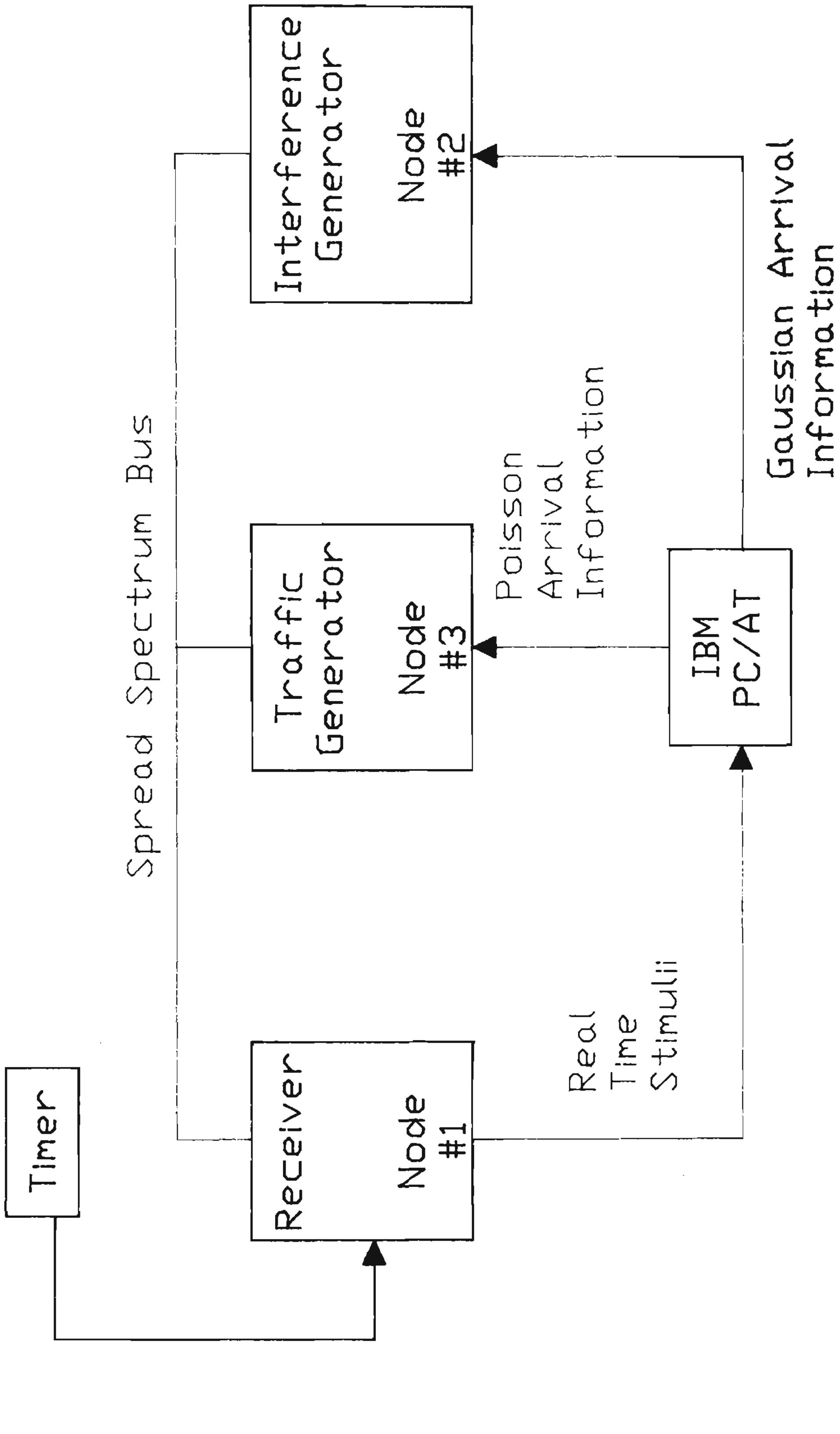


Figure 47 Hardware Configuration For Performance Tests

particular distribution. The **Poisson** characteristic arrival times were then used for the traffic generator (node **#3**). The interference generator, used only in tests **#2** and **#3**, was included to represent the accumulated potential traffic from other nodes which could be transmitting in the chosen channel. Since an accumulation of **Poisson** events approximates to a **Gaussian** distribution of events, a **Gaussian** distribution random number generator was also required to provide packet arrival data for the interference generator (node **#2**). A computer program which generates the required statistics was written, this program is described and listed in Appendix F.

When a program such as this is run on a system like an IBM PC/AT there is a need to synchronise the IBM to real world events. In the test configuration shown in Figure 47 this synchronisation is required to achieve **Poisson** and/or **Gaussian** arrival characteristics and is obtained in the following manner :-

If each packet is regarded as **1800** bits long, then a packet arrival rate of **G = 1** may be obtained by counting **1800** at a rate of 1200 Hz.

Packet arrival times may then be normalised to a the packet transmision time by counting "**n**" clock pulses at a rate of 1200 Hz ie. **G = n/1800**.

The value "**n**" is loaded into the **T2** counter on VIA **#1**, this counter is then decremented by the 1200 Hz data clock. When the count reaches zero, a stimulus is sent to the PC which then calculates

a) A **Poisson** distribution value

b) A **Gaussian** distribution value

both with a mean value of **n/2**.

These values are converted to Hexadecimal numbers and the **Poisson** number sent to node **#3** and the **Gaussian** number sent to node **#2**.

These numbers are loaded into the **T2** counter of VIA **#1** in the respective nodes, the counters being decremented by the 1200 Hz data clock.

When the counter is decremented to zero it represents the "**arrival**" of a packet for transmission. These "arrivals" are then subject to **Poisson** or **Gaussian** distribution characteristics at the rate of "**n**" packets per packet period, ie. these "**arrivals**" represent **G**, the "**Offered Traffic**".

In order to be able to determine the channel utilisation, it was then necessary to repeat the experiment with different values of offered traffic and count the number of successful arrivals. To give repeatability to the test procedure, a counter was used to interrupt node **#1** after **256** seconds (this represents **170** packets). The number of successful arrivals, divided by **170**, would then give the channel occupancy.

7.6.2 Test Results

Three separate tests were carried out in order to develop a realistic idea of the system channel throughput.

7.6.2.1 Test #1 Peak Channel Occupancy - no collisions

This test was designed to produce a figure which could be used as a target for peak channel occupancy. No interference node was used in this test. Only the receiver, node **#1**, and the traffic generator, node **#3**, were used. The transmitter was designed to operate using the protocols described in section 6.2.

The pre-transmission channel scan period is **32** bits (ie. "**a**" = 32/1800 = 0.018)

Table 5 shows the results for this test. For each value of offered traffic the test was run **3** times and the resultant number of arrivals is the average of these **3** runs. The channel occupancy is calculated from the number of arrivals divided by the channel capacity (ie. **170** packets per test run).

Table 5 Measured Peak Throughput - no interference

Theoretical Traffic "G"	Delivered Packets	$Log_{10}G$	Throughput "S" = Delivered/170
0.1	17	0.000	0.100
0.2	34	0.301	0.200
0.4	59	0.602	0.347
0.8	75	0.903	0.441
1.0	89	1.000	0.524
2.0	113	1.301	0.665
5.0	141	1.699	0.829
10.0	146	2.000	0.859
15.0	150	2.176	0.882
20.0	152	2.301	0.894

Figure 48 is a graph of the results of these tests. It can be seen, by comparing this graph with Figure 45, that the peak channel occupancy is around **89%**. This figure is actually dependent on the mean inter-arrival period, since packets arriving while the channel is still occupied with exisiting traffic are re-scheduled with a very long back off time, **X** (ie. tending to infinity for the purpose of

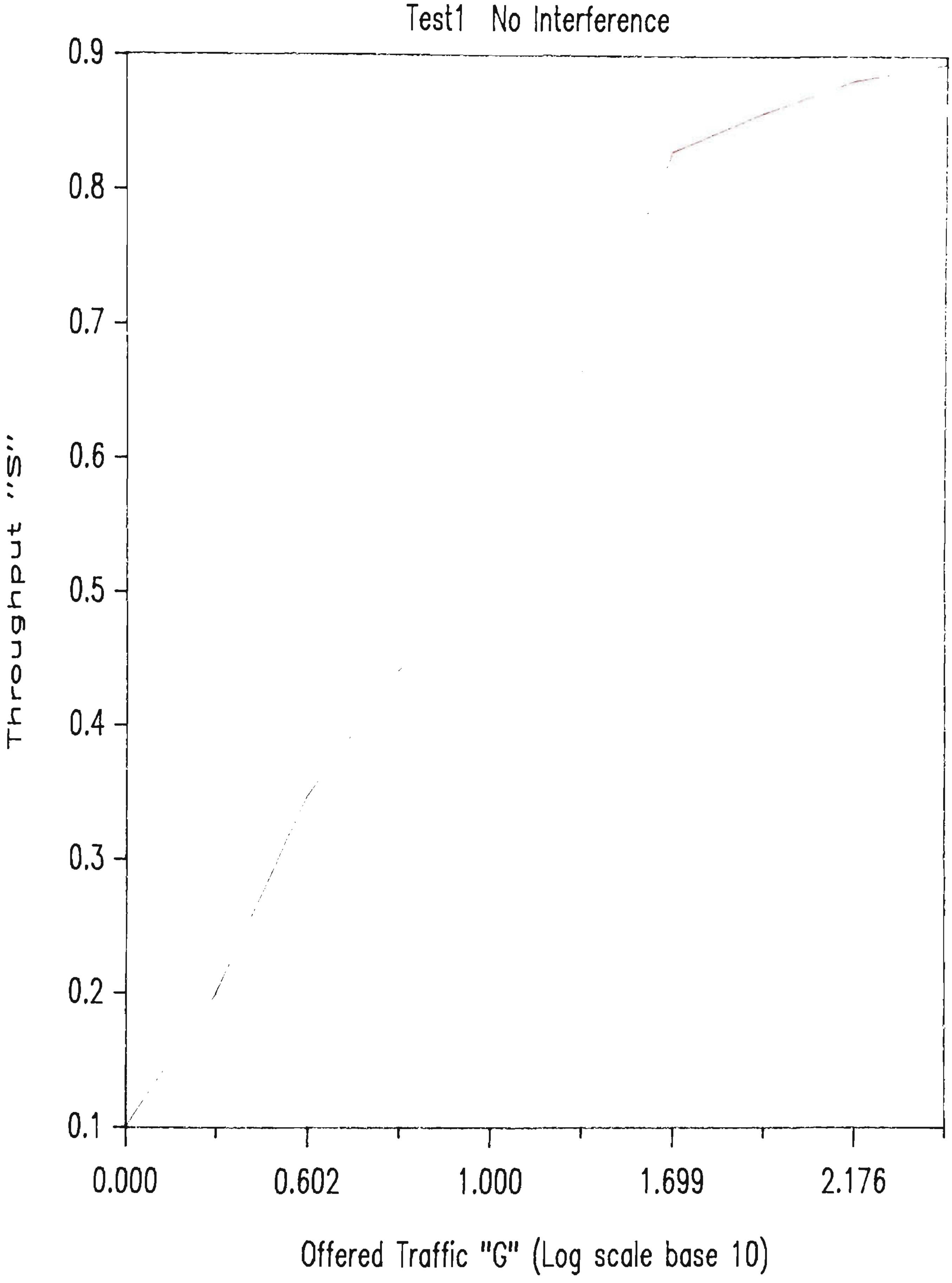


Figure 48 System Throughput - No Interference

these tests). Only packets which arrive when the channel is not actually occupied are accepted for transmission, thus, if the mean inter-arrival time is large the channel could wait, unoccupied, for a packet to transmit. With an offered traffic rate of **20**, therefore, the channel could be unoccupied for up to **5%** of a packet period, waiting for a new packet to transmit. Under these circumstances a peak channel occupancy in the order of **90%** is not unreasonable. This figure will then represent a guide to the channel occupancy in the presence of interfering traffic and will also serve as a measure of the contention handling characteristics of the system.

7.6.2.2 Test #2 Throughput - with collisions

The parameters and protocols for this test are very similar to those used in test **#1**. This test uses, however, both the Traffic and Interference generators shown in Figure 47. The presence of the two packet generators means that collisions will occur, the packet arrival rate, representing the offered traffic. The number of successful arrivals was counted at various offered traffic rates.

Table 6 shows the results of these measurements, and Figure 49 shows a graph of the results. It can be seen that, the channel throughput increases uniformly with offered traffic until the effects of collisions and re-scheduling begin to take effect. The peak channel throughput of approximately **78%** occurs at an offered traffic rate of **5** packets per packet slot. Below this rate the inter-arrival period is largely responsible for the non-utilised time between successive packets, and thus the lower throughput. Above this rate the probability of the arrivals being in the vulnerable period, thus causing collisions, begins to rise, and more of the channel time is occupied with pre-transmission scanning and channel jamming caused by transmissions. At offered traffic rates above **5** packets per slot, therefore, the throughput falls noticeably.

Table 6 Measured Throughput - with interference

Theoretical Traffic "G"	Delivered Packets	$Log_{10}G$	Throughput "S" = Delivered/170
0.1	17	0.000	0.100
0.2	33	0.301	0.194
0.4	49	0.602	0.288
0.8	80	0.903	0.471
1.0	90	1.000	0.529
2.0	125	1.301	0.735
5.0	131	1.699	0.771
10.0	123	2.000	0.724
15.0	100	2.176	0.588
20.0	69	2.301	0.406

These characteristics follow the trends predicted in the Theoretical Model (see Figure 43) and in the Simulation Model (see Figure 45), although the break point in both the simulation and theoretical models occurs at higher rates of offered traffic. These results demonstrate the validity of the models described in chapter 6.

7.6.2.3 Test #3 Short (Pseudo random) Re-scheduling Times

Because of the limitations highlighted by the preceding tests it was decided to investigate an alternative protocol, with a view to increasing the peak throughput and maintaining this peak level. The test procedures were the same as those described for test **#2**, above. The alternative takes the form of drastically reducing the re-scheduling time to a random value in the range **1** to **255** bits. For this reason a pseudo random number generator, taking the form of an **8** bit MLS, was included in the control software. Each transmitter loads the MLS generator with a different seed value, to ensure variation between transmitters. The MLS is "**clocked**" once every polling cycle.

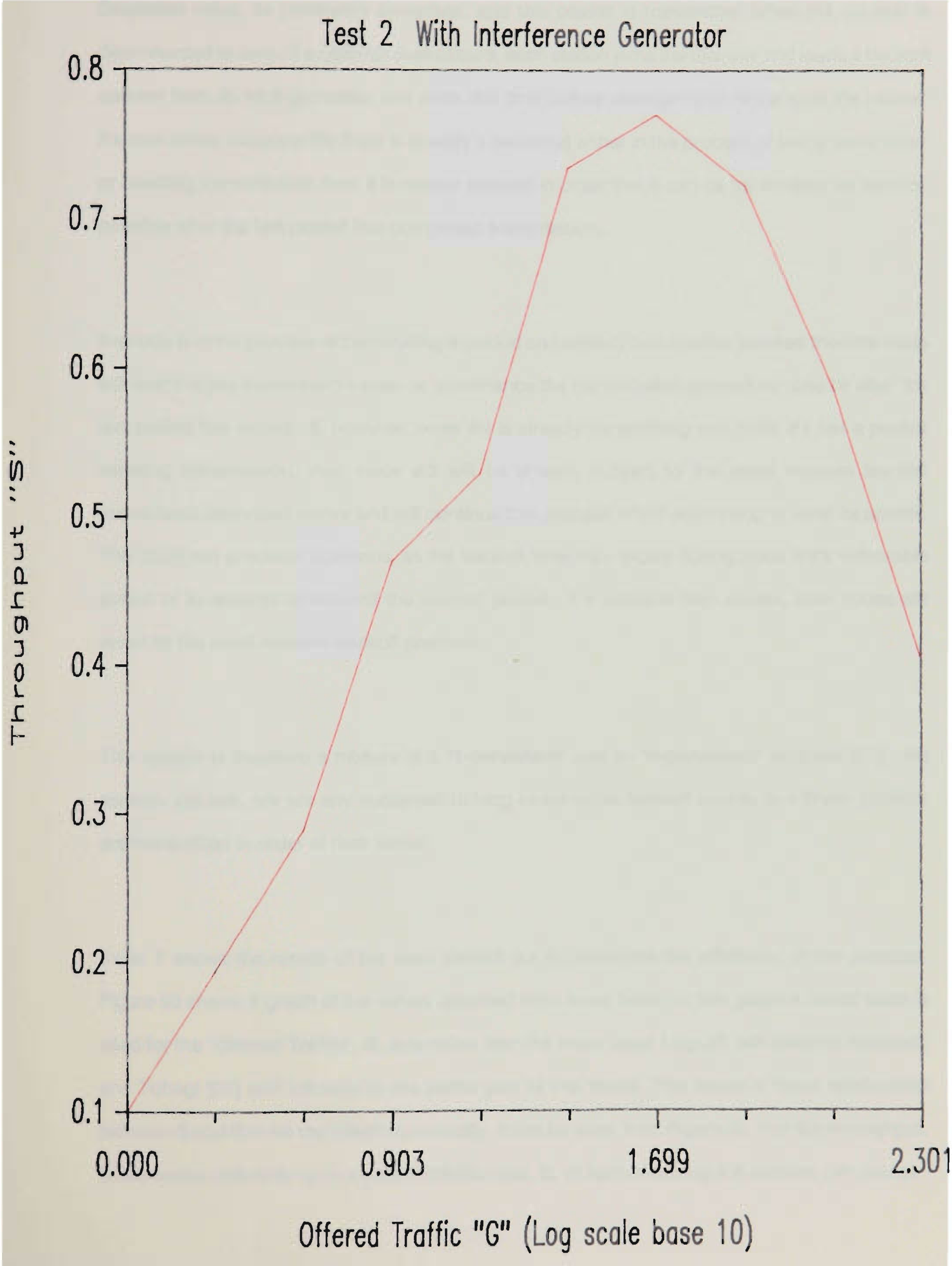


Figure 49 System Throughput - With Interference Generator

When a new packet arrives, if the channel is vacant the **T2** counter is loaded with the **Poisson** or **Gaussian** value, as previously described, and this packet is transmitted when the counter is decremented to zero. If a collision then occurs, each station jams the channel and loads a backoff counter from its MLS generator and waits this time before attempting to re-transmit the packet. If a new arrival occurs while there is already a packet(s) either in the process of being transmitted or awaiting transmission, then it is merely stacked in order that it can be transmitted as soon as possible after the last packet has completed transmission.

If a node is in the process of transmitting a packet and already has another stacked then the node will start the pre-transmission scan (ie. commence the transmission procedure) **one** bit after the last packet has ended. If, however, node **#2** is already transmitting and node **#3** has a packet awaiting transmission, then node **#3** will be already subject to the small random backoff procedures described above and will continue this process whilst attempting to send its packet. This does not preclude collisions, as the backoff time may expire during node **#2**'s vulnerable period of its attempt to transmit the second packet. If a collision then results, both nodes will revert to the small random backoff protocol.

This system is therefore a mixture of a "**1-persistent**" and an "**n-persistent**" protocol [27]. No packets are lost, nor are any subjected to long re-schedule backoff counts and finally packets are transmitted in order of their arrival.

Table 7 shows the results of the tests carried out to determine the efficiency of this protocol. Figure 50 shows a graph of the values obtained from these tests. In this graph a **linear** scale is used for the "**Offered Traffic**", **G**, axis rather than the more usual **Log$_{10}$G** axis used by Kleinrock and Tobagi [27] and followed in the earlier part of this thesis. This allows a linear relationship between **S** and **G** to be identified more readily. It can be seen, from Figure 50, that the throughput, **S**, increases uniformly up to an offered traffic rate, **G**, of approximately **0.8** packets per packet

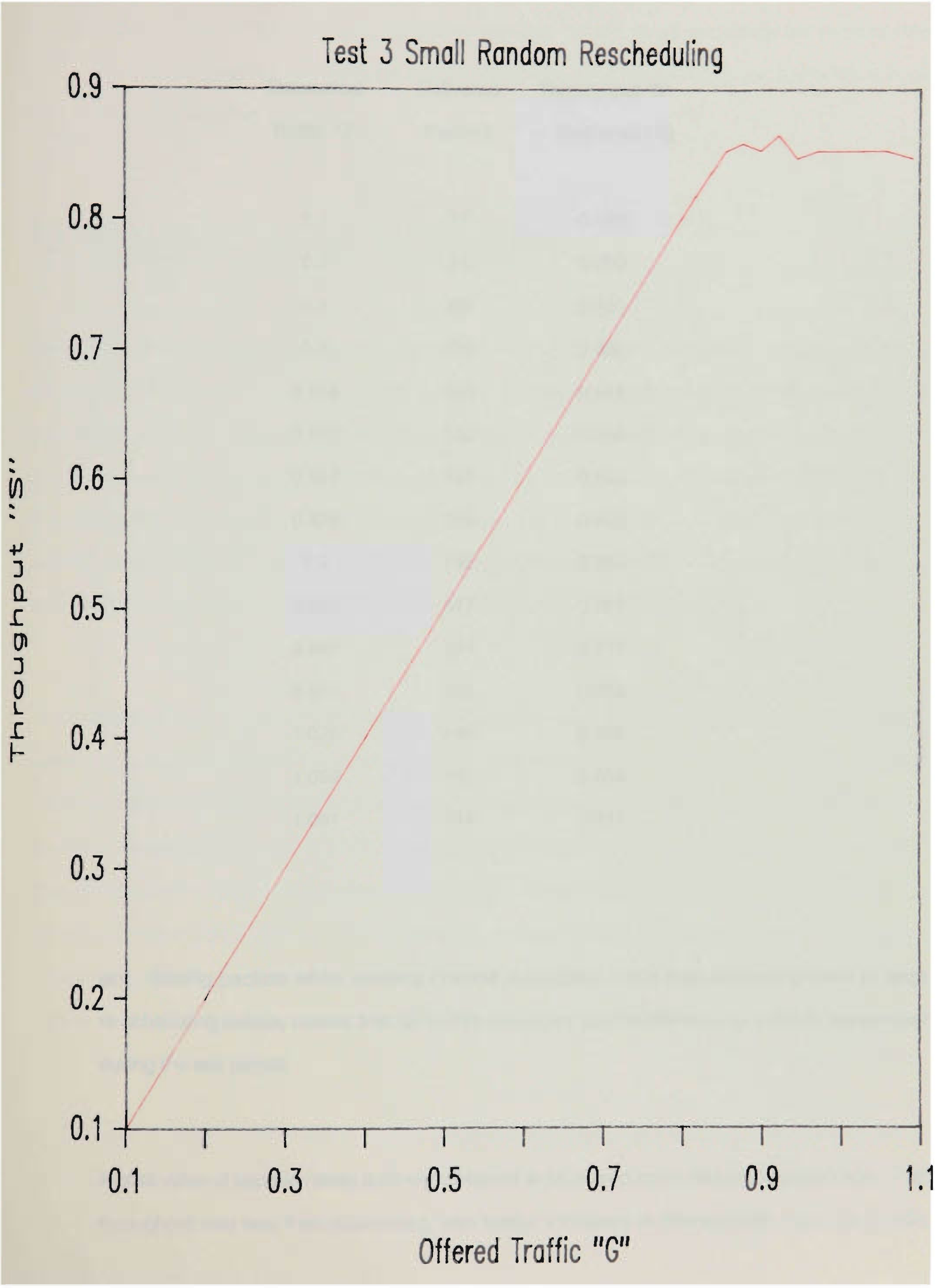


Figure 50 System Throughput - Alternative Protocol (small random rescheduling)

Table 7 Alternative Protocol Throughput - short re-scheduling delay

Theoretical Traffic "G"	Delivered Packets	Throughput "S" = Delivered/170
0.1	17	0.100
0.2	34	0.200
0.4	68	0.400
0.8	136	0.800
0.818	139	0.818
0.837	142	0.835
0.857	145	0.853
0.878	146	0.859
0.9	145	0.853
0.923	147	0.865
0.947	144	0.847
0.973	145	0.853
1.028	145	0.853
1.059	145	0.853
1.091	144	0.847

slot. Storing packets whilst awaiting channel availability, rather than subjecting them to large re-scheduling delays, means that up to this rate every packet offered was actually transmitted during the test period.

A peak value of approximately **0.86** was obtained at an offered traffic rate of the same order. This throughput rate was then maintained, with further increases in offered traffic rate, rather than suffering the decrease of the previous protocol, as illustrated in Figure 49 above. The cost of the maintenance of the throughput rate was an increase in the number of packets being held at the node awaiting transmission into the network. This means that the node requires significant storage for these "**delayed**" packets.

7.7 Analysis of System Performance

Section 7.5 described the basic system connection testing. Node **#1** transmitted to node **#2** which transmitted to node **#3** which completed the loop by transmitting to node **#1**. 100 packets were transmitted, each of which required 3 transmissions to complete the loop. Each packet was checked for errors whenever it was received, and none were found. This procedure was carried out 10 times, resulting in a total of 3000 separate transmissions taking place without error. This showed that the address code selection, modulation, demodulation and data recovery circuits were performing as expected; and that the HDLC synchronisation and clock recovery was being achieved satisfactorily.

After confirming the correct operation of the hardware the next step was to investigate the system's performance characteristics. The three measures of system throughput or traffic handling performance follow very closely the characteristics predicted in the theoretical and simulation studies carried out, and described in chapter 6. This verifies the validity of the models developed. The excellent traffic handling characteristics exhibited by the hardware (and software) confirm the suitability of the system to meet the initial design aims. ie. to provide simultaneous multiple access facilities. The flexibility of the system is also demonstrated in the fact that with the relatively simple modifications to the system software, outlined in section 7.6.2.3., an alternative protocol could be investigated.

Overall these tests have verified and quantified the system's performance characteristics. The system has performed as predicted by both the theoretical and simulation studies carried out for this purpose. These tests verify that the system, as built, actually meets all of the original design criteria.

8 CONCLUSIONS

8.1 Discussion of the Project

In the preceding chapters it has been shown that it is a practical proposition to implement a Local Area Network using Spread Spectrum techniques. The network produced has been designed with commonly available components and proven, by exhaustive testing, to be both reliable and resistant to electrical interference. This makes the network particularly useful in areas which, hitherto, have been considered unsuitable for the installation of a LAN either because it wasn't cost effective or the environment in which the LAN would be expected to work was considered too hostile.

The channel address codes, used in a Code Division Multiple Access system such as this, are of paramount importance, and 5 code allocation schemes were considered before a selection was made. The first 3 of the schemes considered were suitable for asynchronous systems, and the latter 2 for synchronised systems.

The suitability of the codes in the address allocation scheme selected was demonstrated practically. The expectations about the shape and magnitude of the correlation values across the 2 bit window were also confirmed by actual hardware measurements (Figures 35 to 41). This allowed significant modularity in hardware design and testing, thus ensuring that validated modules could be assembled to form the final system. The use of a Read Only Memory to store the synchronisation and address codes, meant that the act of synchronising automatically created other codes in the correct phases to be used directly as address codes. The creation of a discrete stepped sliding correlator meant that it was possible to use the microprocessor control circuit to exercise direct control over the delay in this correlator, so that the optimum decoding delay could be used.

The theoretical studies showed how the throughput could be expected to change with different values of vulnerable period, **a** (Figure 43). The theoretical throughput model developed in Chapter 6 shows significant throughput advantages over Aloha, Slotted Aloha and also the 1-persistent and non-persistent CSMA models. Figure 44 shows how, at a vulnerable period of **a = 0.01**, the throughput of this system compares with that of the multiple access protocols noted above. The simulation results supported those predicted by the theoretical work, and Figure 45 shows how the theoretical and simulated throughputs compare at a vulnerable period value of **a = 0.0178**. The theoretical and simulation studies were extremely valuable in that they provided a guideline against which the hardware could be compared; and these three sets of results were, in fact, used to predict the expected performance of the hardware.

The vulnerable period in the system is determined by the time taken to perform the pre-transmission scan, divided by the time taken to transmit a packet (ie. **a = 32/1800**). As explained in section 7.6.2.1, the peak channel occupancy, **90%**, is a function of the throughput test calculation methods. This peak value, illustrated in Figure 48, is, therefore, the maximum against which the throughput with interference should be measured. Having measured the performance under ideal conditions, the throughput characteristics under "normal" conditions (ie. with more than one transmitter) were measured and compared with those predicted. The measured and predicted results were again very close, as can be seen from Figure 49, which shows how the throughput varies with an increase in the offered traffic rate, when the interference increases in the same proportion. A very high level of confidence may be placed in the accuracy of the measured throughput, as it compares very well with that predicted from both the theoretical and simulation studies. This can be seen by comparing Figure 49 with Figure 45.

Having demonstrated the performance of the system, its flexibility was investigated, and Figure 50 shows the throughput when the system was re-programmed to perform a slightly different protocol; one where packets are stored for re-transmission rather than being subjected to a "very long" re-scheduling delay. This protocol attempts to make more efficient use of the channel capacity by re-initiating the transmission process, if there is a stored packet available, as soon as the previous transmission has been completed. The peak throughput obtained with this protocol is **86%**. This value compares well with the peak values of **88%**, predicted by the simulation studies, and **91%**, predicted by the theoretical studies (Figure 45).

The number of packets retained at each node for transmission, however, grew with each increase in offered traffic rate.

The points in favour of this protocol are:-

- a high channel occupancy value is achieveable

- the rate is maintainable

- packets are transmitted in the order of arrival

The major disadvantage is that large amounts of primary memory (or secondary memory such as disk) would be required to support the protocol with very high levels of offered traffic.

This implementation of an alternative protocol demonstrates the flexibility of the system. This is an additional point in favour of this **SS LAN**, in that it may quite easily be tailored to meet the specific conditions under which it may be expected to operate. The protocol developed and simulated in chapter 6, and tested in section 7.6.2.2, is probably more appropriate to systems where the anticipated traffic rate is significantly below the breakpoint of **5** packets per traffic slot. Very little extra storage is required at each node to implement this protocol. If, however, the system may be expected to be subjected to periods where the offered traffic rate exceeds the breakpoint value of **5**, then the alternative protocol, section 7.6.2.3, provides marginally higher throughput, because of the fewer collisions and re-schedulings. This is achieved, however, at the cost of the extra storage which is required at each node.

During the course of this research project a Local Area Network using Spread Spectrum techniques has been designed, built and tested. Individual module, and total system, performance characteristics have been theoretically estimated, simulated and then actually measured. Additional investigations which demonstrate the system's flexibility have also been undertaken. This research project has, therefore, been a complete success, and each of the design goals, identified in section 2.2, has been met.

8.2 Comparisons With Other Networks

The relative advantages or disadvantages of this system compared with other, more conventional, CSMA/CD type LANs depends very much on the user's requirements and the conditions under which the system will be used. The following discussion attempts to identify certain important operating conditions and compare the two systems under these conditions, each assumes a user population of 100 :-

- If one node is dominant and the other 99 nodes all communicate with this one, eg. a file server, then the conventional system is far superior. The SS LAN is designed for equi-probable multi-user communications, ie. 50 pairs of nodes communicating concurrently.

- A CSMA/CD bus shows its best performance for loads less than 0.2, the transfer delay increases rather rapidly for loads greater than 0.4 and has a vertical asymptote at about 0.6 [29]. If the load is spread evenly over 50 channels, then one would expect the overall SS load capacity to be 50 times that of a conventional LAN before delay becomes an important factor.

- The SS LAN can find and maintain synchronisation in an environment with 7% errors at the chip bit rate (section 4.4.3.1). A 7% deviation from the correlation value of 64 expected from the A/D converter, would result in a value between 56 and 70. These values are well within the range allowed (section 7.3.1), and would not cause incorrect data recovery and subsequent packet rejection. Errors in a conventional LAN could cause packet rejection.

- A very reliable method of ensuring packet integrity is the use of error detecting codes and positive acknowledgement of each packet received correctly [30]. The acknowledgement packets would, however, be competing for channel space and could exacerbate loading problems in a conventional LAN. The SS LAN already incorporates error detecting coding in the ADLC, and positive acknowledgement does not cause a loading problem in the SS LAN, since the pair of communicating nodes each have a unique receive channel.

- In a conventional LAN the vulnerable period is determined by the maximum propagation delay in the system. The SS LAN has its vulnerable period, **a**, set to a value of 0.018 (section 7.6.2.1). Using a propagation delay factor of 5 us / Km [29], a conventional LAN operating at 10 MHz would be able to serve a network upto 650 metres in length. The SS LAN's vulnerable period is determined by the channel scan time and, with a 2 bit separation between codes, this equates to a network of 488 metres (section 3.3). So for the same value of vulnerable period the conventional LAN can service a longer network.

- A cost comparison between the SS LAN and commercially available components is not absolute, since profit and a component to cover development costs and company overheads are already included in the price of bought items. Commercial LAN cards vary considerably in cost, depending on the facilities offered and the degree of inteligence in the card. The CMC ENP-66 costs $1380, and is towards the top of the price range. The Ungermann Bass PC NIU AT costs $954, and is a mid-valued item. The Micom Interlan NI5210-8 costs around $467, and is towards the economy end of the market. A SS LAN node could be made for under $200, so with profit and development cost components etc. should be competitive with the Micom Interlan product.

- Various manufacturers now offer LAN controller chips as microprocessor peripheral devices. Intel's 82586 Local Area Network Co-Processor costs $134 and AMD's AM77990 Local Area Network Controller for Ethernet (LANCE) costs $68. The SS LAN card would cost more than these individual chip controllers.

8.3 Suggestions for Future Work

During the development of this Spread Spectrum LAN, some areas have been identified which may be suitable for enhancement in future:-

- a reduction in the number of Integrated Circuits (**IC**s) in the control circuits could be achieved by using a single chip microprocessor

- an improvement in processing potential is available by using a more modern 8 bit microprocessor, such as the Motorola MC68008. An even more substantial improvement could, obviously, be achieved with a MC68020 or MC68030 device

- throughout this development, the decision to use standard commercial components has meant that custom **IC**s have not been considered. The very recent introduction of re-programmable Application Specific Integrated Circuits (**ASIC**s) means that these devices may be cost effective and, therefore, suitable for the replacement of a substantial amount of the digital logic in this system

- the possible development of **ASIC**s with analogue functions may mean that both the analogue and digital chip count could be significantly reduced by using such devices, this would also reduce printed circuit complexity and, hence, production costs

- **A/D** converters with built in multiplexers, **S/H** circuits, etc., are now available. The use of such devices would allow simultaneous monitoring of many channels to be performed

- code generation using Random Access Memories (**RAM**) in conjunction with the **ROM** already used, would allow dynamic channel address code allocation within the system. This may give wider appeal to a system of this nature.

- one aspect of system design which could be further investigated is the incorporation of adaptive transmission and reception sections. The system as it is now can be manually adapted to suit the number of users in certain ranges eg. up to 10, or 20 to 40 etc. by tuning the "unit" voltage setting and the analogue multiplier scale factor. This is static tuning and the system still has a limitation in its dynamic range. The inclusion of adaptive control units in each node to sense the number of concurrent users and dynamically adjust the "unit" and scale settings could overcome this limitation. This control unit could also adjust the scale setting to cater for possible variations in the attenuation experienced by transmissions from different nodes.

The incorporation of the above suggestions for future work into this, already working, system could only enhance its market potential by making it cheaper to assemble and possibly more reliable, whilst at the same time giving it performance improvements to increase its appeal to potential users.

Bibliography

1 SIMCOCK A.L., naRANONG C. and LEUNG P., "A Synchronised Spread Spectrum Local Area Network", The Journal of The Institution of Electronic and Radio Engineers. pp 17-21, Vol 57 No. 1 January/February 1987.

2 SCHOLTZ R.A., "The Origins of Spread Spectrum Communications", I.E.E.E. Transactions on Communications, pp. 822-854, Vol. COM-30, No 5, May 1982.

3 MAHMOOD N., "Spread Spectrum Communication System For Civil Use", Wireless World, pp. 76-80, March 1983.

4 SKAUG R. and HJELMSTAD J.F., "Spread Spectrum in Communication", Peter Peringrinus (London), 1985.

5 PICKHOLTZ R.L., SCHILLING D.L., MILSTEIN L.B., "Theory of Spread Spectrum Communications - A Tutorial", I.E.E.E. Transactions on Communications, pp. 855-884, Vol. COM-30, May 1982.

6 naRANONG C., SIMCOCK A.L. and LEUNG P. "A Spread Spectrum Manufacturing Information Network", The Third Annual Conference on Manufacturing Engineering - Technology For Manufacturing Growth pp 118- 122 Newcastle 1986

7 SIMCOCK A.L., naRANONG C. and LEUNG P., "A Spread Spectrum Manufacturing Information Network", College of Mechanical Engineering Technical Report, pp 1-5, August 1987.

8 SIMCOCK A.L., naRANONG C. and LEUNG P., "Protection, Rejection and Throughput Characteristics of a Spread Spectrum Local Area Network" Conference on Computing Systems and Information Technology 1987, pp 146- 150, Brisbane, Queensland June 1987.

9 SCHOLTZ R.A., "The Spread Spectrum Concept", I.E.E.E. Transactions on Communications, pp 748-755, Vol. COM-25, August 1977.

10 PURSLEY M.B., "Spread-Spectrum Multiple Access Communications", - chapter in "Multi-User Communications Systems", Longo G. (ED), 1981.

11 SIMCOCK A.L., naRANONG C. and LEUNG P., "A Localised Spread Spectrum Network" The Journal of Electrical and Electronic Engineers, Australia, pp 110-116, Vol. 7, No. 2, June 1987.

12 DIXON R.C., "Spread Spectrum Systems", Wiley (New York), 1976.

13 SMYTHE C.S. and SPRACKLEN C.T., "A High Speed Local Area Network Using Spread-spectrum Code-division-multiple-access Techniques", The Radio and Electronic Engineer, pp 225-226, Vol. 54, May 1984.

14 ELHAKEEM A.K., HAFEZ H.M. and MAHMOUD S.A., "Spread-Spectrum Access to Mixed Voice-Data Local Area Networks", I.E.E.E. Journal on Selected Areas in Communication, pp 1054-1069, Vol. SAC-1, No. 6, December 1983. .

15 PFIEFFER P. and MEYRUEIS P., "Fiber Optic Bus with Spread Spectrum Techniques", Proceeding S.P.I.E. Int. Soc. Opt. Eng. (USA), pp 20-23, Vol. 434, 1983.

16 HIROSAKI B., HASEGAWA S., SAWAI A., "Spread Spectrum Multiple Access Data Loop", NEC Res DEV n71 pp 48-58, October 1983.

17 SPRACKLEN. C.T., SMYTHE C., SCOTT M.A. and ISMAIL N.A., "Spreadnet - A Spread Spectrum Local Area Network", Journal of the Institution of Electronic and Radio Engineers, pp 12-16, Vol 57, No 1. January/February 1987.

18 CARLSON BRUCE A., "Communication Systems -An Introduction to Signals and Noise in Electrical Communication", McGraw-Hill, (Tokyo), 1981.

19 PETERSON W.W. and WELDON E.J., "Error Correcting Codes 2nd Edition", M.I.T. Press (Cambridge, MA), 1972.

20 SARWATE D.V. and PURSLEY M.B., "Cross Correlation Properties of Pseudo- Random and Related Sequences", Proc. I.E.E.E., pp 593-619, Vol. 69, May 1980.

21 CHORAFAS D.N., "Designing and Implementing Local Area Networks", McGraw-Hill (New York), 1984.

22 HOPPER A., "The Cambridge Ring - A Local Network", Advanced Techniques for Microprocessor Systems, Hanna F.K. (ED), pp 67-71, Perigrinus (Stevenage, Herts), 1980.

23 SIMCOCK A.L., naRANONG C. and LEUNG P., "Spread Spectrum Techniques for LAN Interconnections" A.T.E.R.B. Workshop on LANs 1987, pp 34-38 Leura, N.S.W. March 1987.

24 LINDHOLM J.H., "An Analysis of the Pseudo-Randomness Properties of Subsequences of Long m-Sequences", I.E.E.E., Transactions on Information Theory, pp 569-576, Vol. IT-14, July 1968.

25 GOLD R., "Optimal Binary Sequences For Spread Spectrum Multiplexing", I.E.E., Transactions on Information Theory, pp 619-621, Vol. IT-13, 1967.

26 Motorola MC6854 Data Sheet.

27 KLEINROCK L. and TOBAGI F.A., "Packet Switching in Radio Channels : Part 1 ", IEEE Transactions on Communications, pp 1400-1416, Vol COM-23, December 1975.

28 ROUNDS F.N., "Use Modeling Techniques to Estimate Local-Net Success", E.D.N., pp 143-150, April 1983.

29 BUX W., "Local-Area Networks: A Performance Comparison", IEEE Transactions on Communications, pp 1465-1473, Vol COM-29, No 10, October 1981.

30 TOBAGI F.A., "Multiaccess Protocols in Packet Communication Systems", IEEE Transactions on Communications, pp 468-488, Vol COM-28, No 4, April 1980.

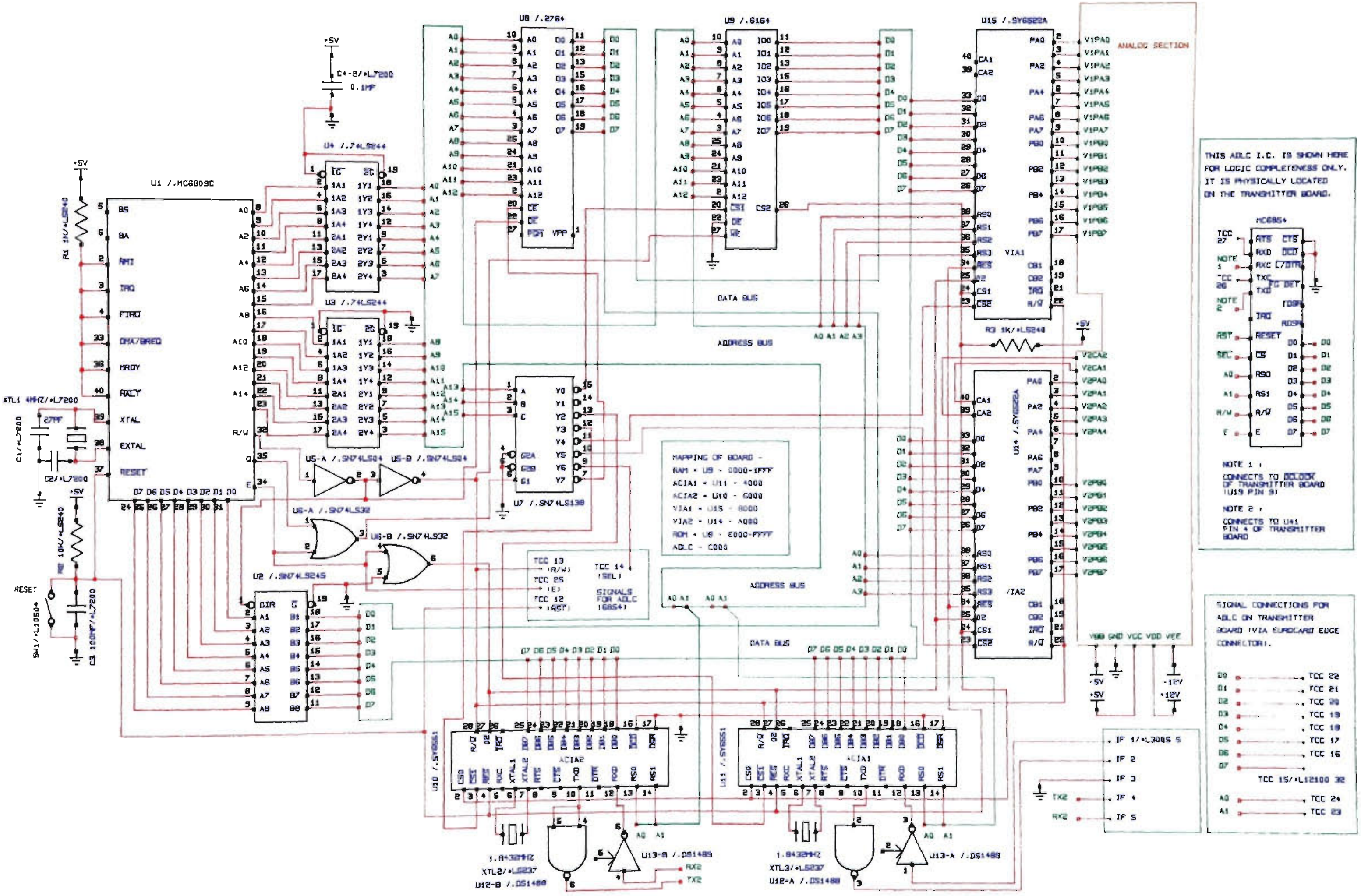
31 GRAYBEAL W. and POOCH U.W., "Simulation Principles and Methods", Little, Brown and Company, (Canada), 1980.

APPENDIX A

A CIRCUIT DIAGRAMS

A.1 Summary

This appendix contains the full circuit diagrams (Figures A1 to A5) for the Microprocessor and Peripheral boards which together constitute a Node of this Spread Spectrum Local Area Network.



MAPPING OF BOARD -
RAM = U9 - 0000-1FFF
ACIA1 = U11 - 4000
ACIA2 = U10 - 6000
VIA1 = U15 - 8000
VIA2 = U14 - A000
ROM = U8 - E000-FFFF
ADLC - C000

THIS ADLC I.C. IS SHOWN HERE FOR LOGIC COMPLETENESS ONLY. IT IS PHYSICALLY LOCATED ON THE TRANSMITTER BOARD.

NOTE 1 :
CONNECTS TO DCLOCK OF TRANSMITTER BOARD (U19 PIN 9)

NOTE 2 :
CONNECTS TO U41 PIN 4 OF TRANSMITTER BOARD

SIGNAL CONNECTIONS FOR ADLC ON TRANSMITTER BOARD (VIA EUROCARD EDGE CONNECTOR).

D0	TCC 22
D1	TCC 21
D2	TCC 20
D3	TCC 19
D4	TCC 18
D5	TCC 17
D6	TCC 16
D7	TCC 15/*L12100 32
A0	TCC 24
A1	TCC 23

FIGURE A1 MICROPROCESSOR BOARD (M) SHEET 1 OF 2

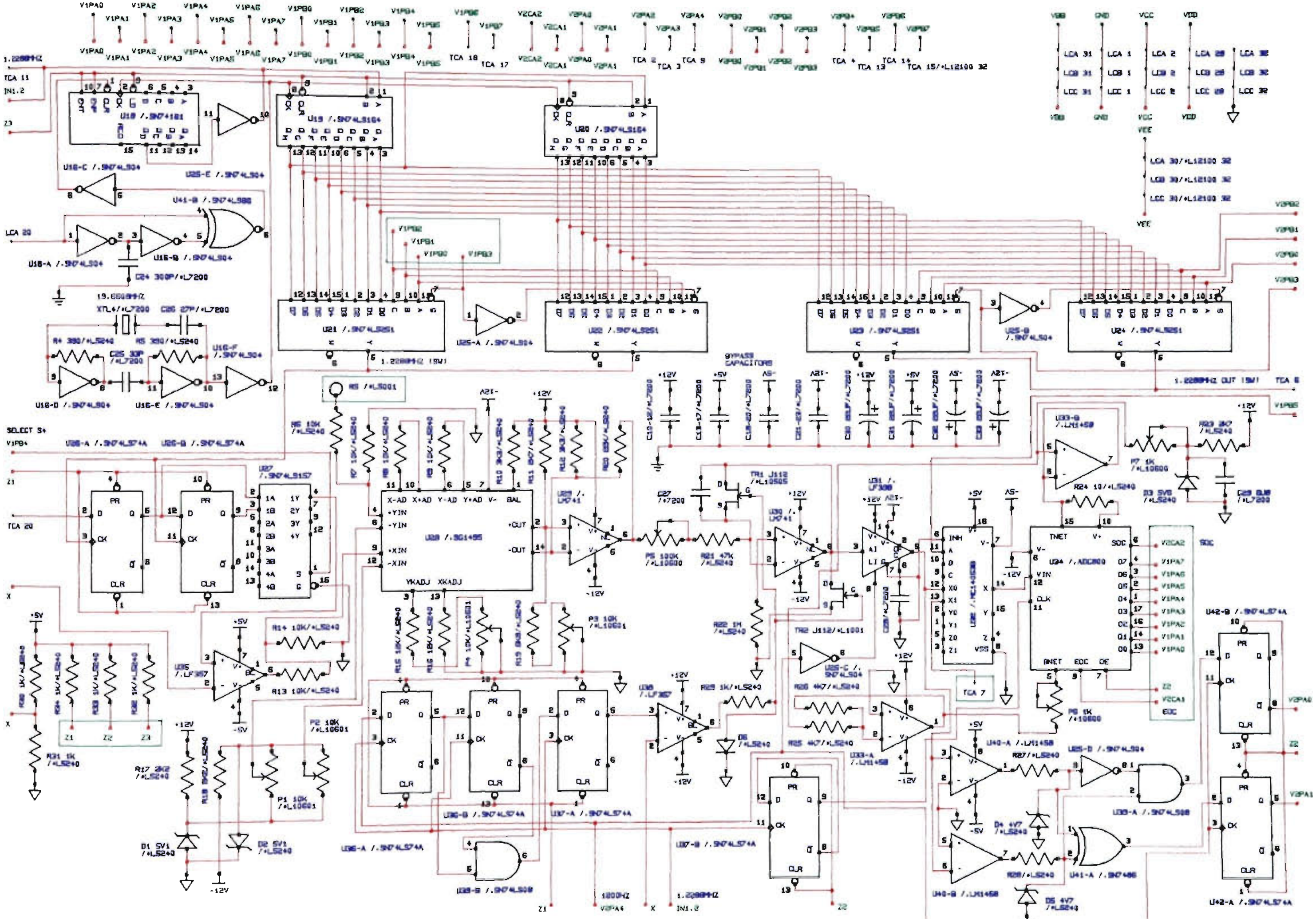


FIGURE A2 MICROPROCESSOR BOARD (M) SHEET 2 OF 2

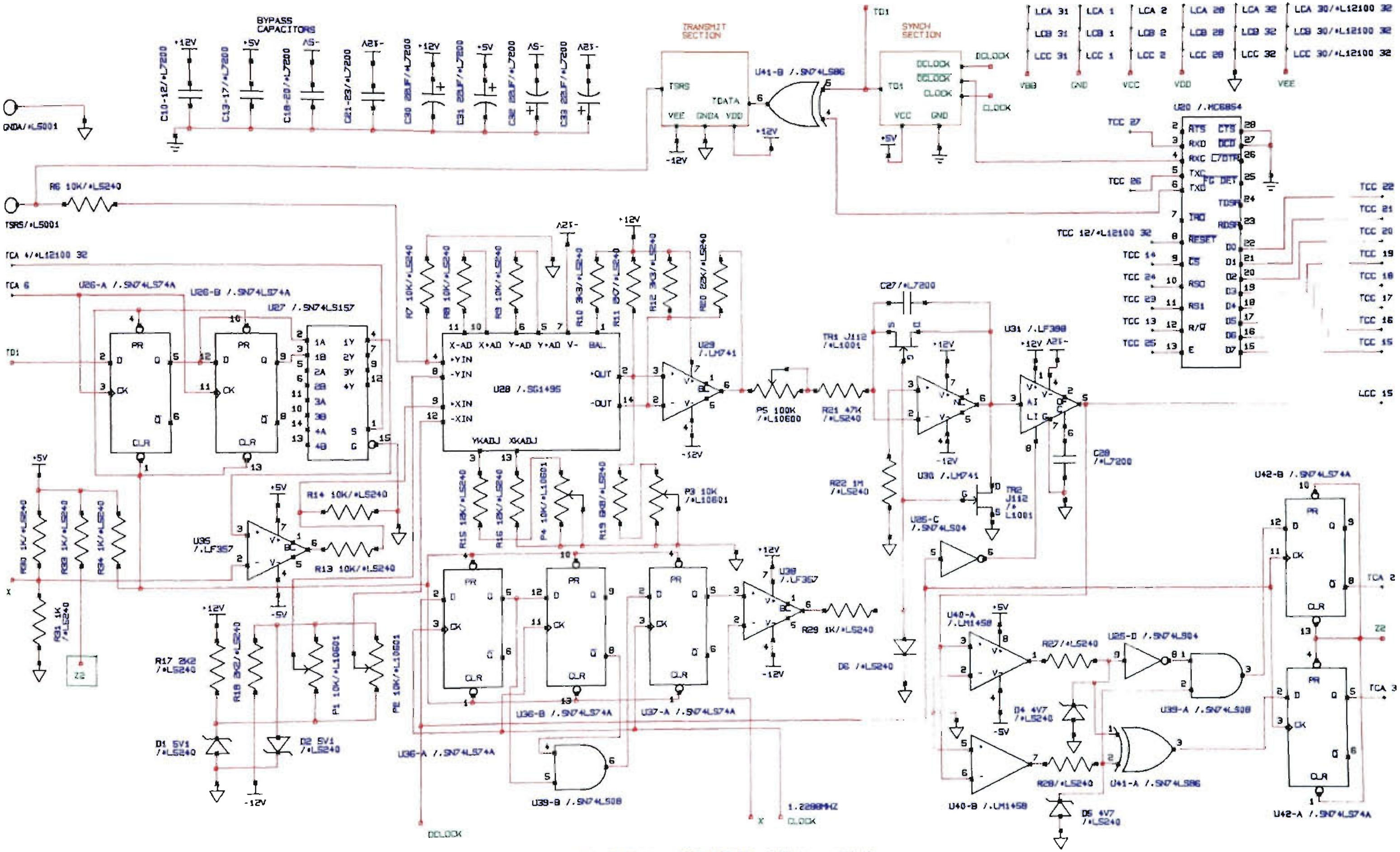


FIGURE A3 PERIPHERAL BOARD (P) SHEET 1 OF 3

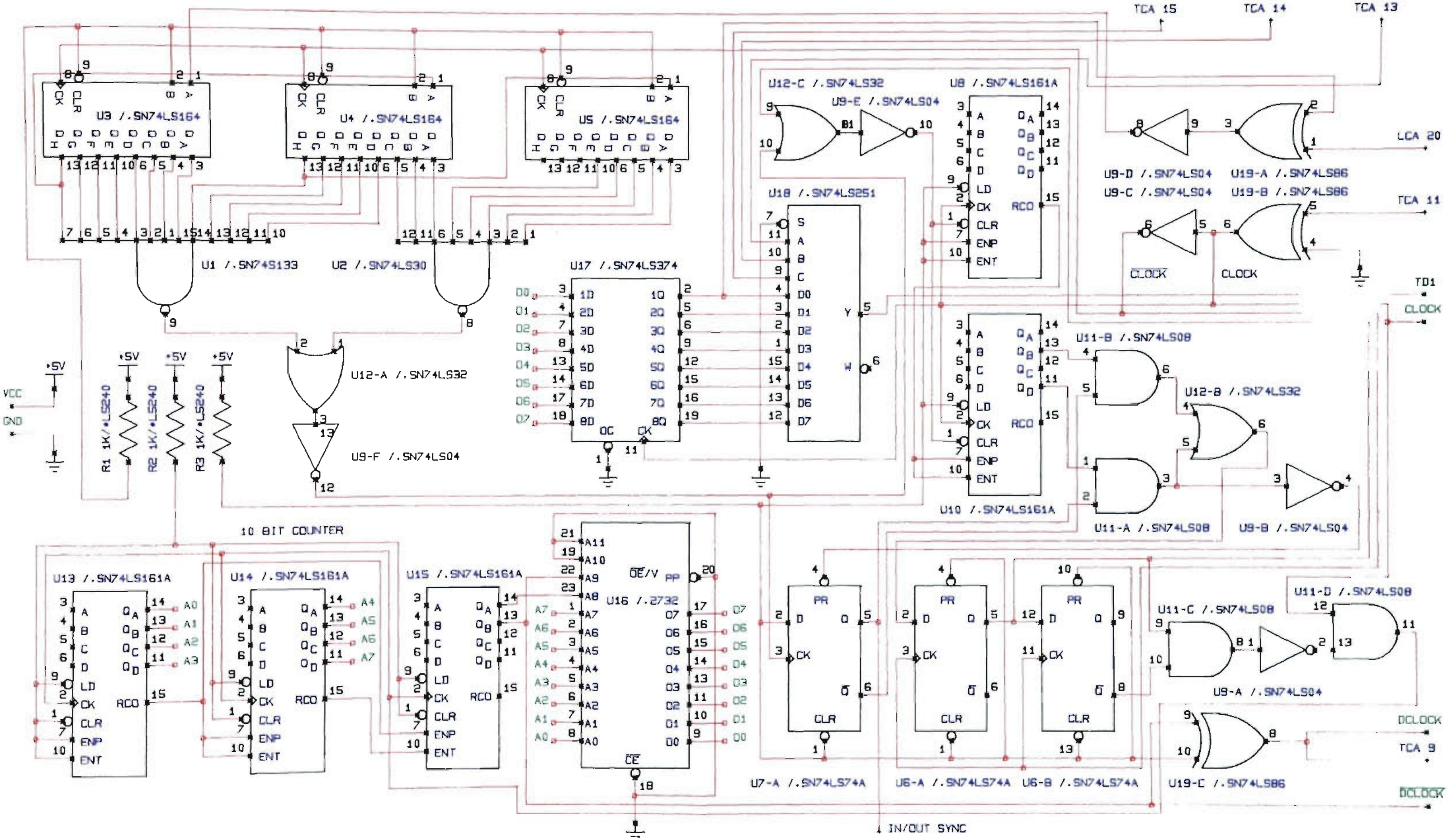
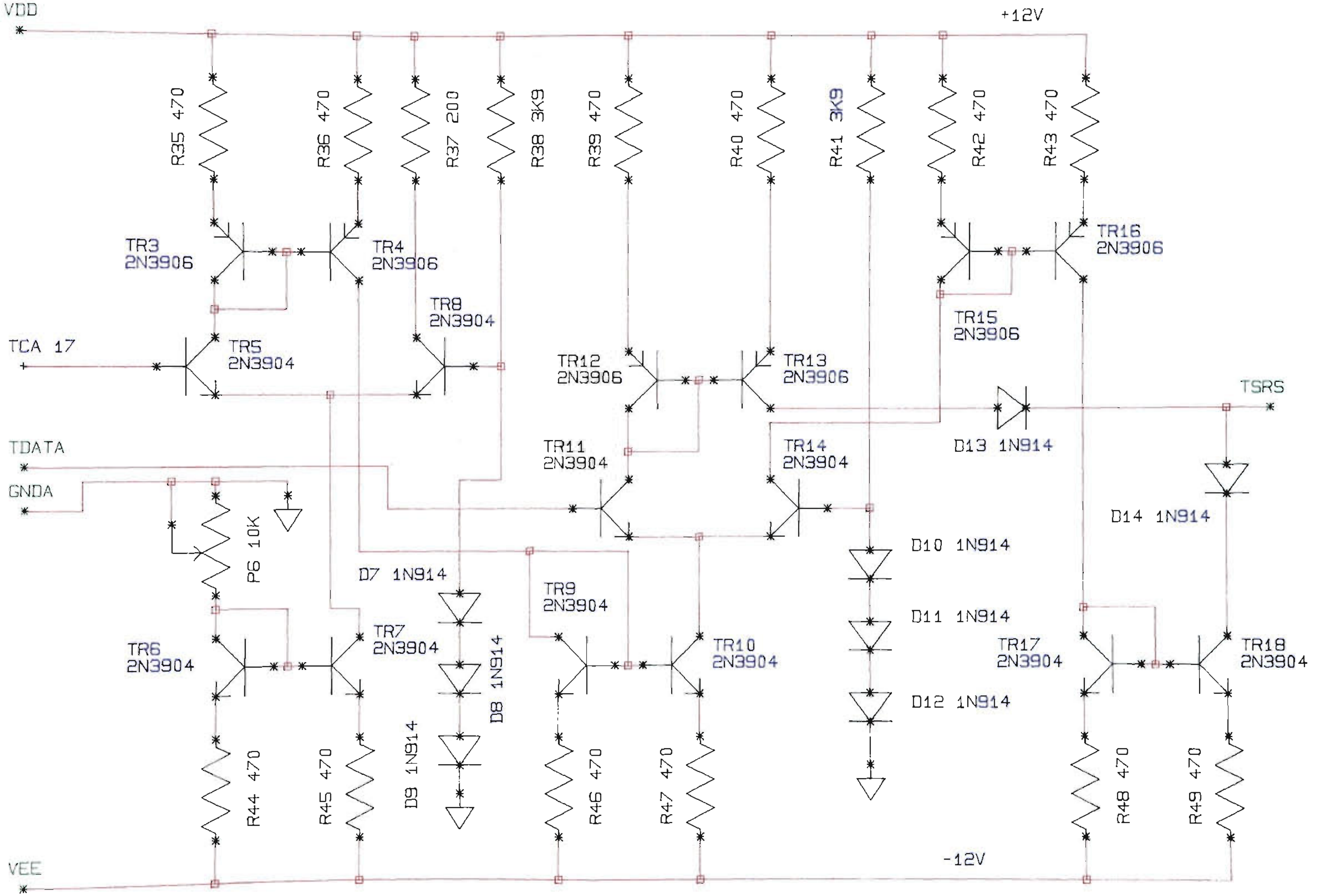


FIGURE A4 PERIPHERAL BOARD (P) SHEET 2 OF 3 (SEE ALSO FIGURE 18)



FIGURF A5 PERIPHERAL BOARD (P) SHEET 3 OF 3 (SEE ALSO FIGURE 24

APPENDIX B

B ERROR GENERATOR

B.1 Summary

This appendix describes the design and extensive testing of the programmable error generator designed especially for this project. In common with the rest of this document the description is at block diagram or functional level rather than at detailed chip implementation level. Complete circuit diagrams are, however, included for reference.

B.2 The Requirements

The question of proving the eventual design of the Spread Spectrum LAN was considered from the very early stages of development. The system had to be capable of maintaining communications even in an electrically hostile environment. It was realised, therefore, that some means must be available to create or simulate this type of environment in order to validate the system design. It would be necessary to inject errors at the chip rate (of 1.2288 MHz) in order to test the synchronisation acquisition and maintenance circuits, and also to investigate the correlation and integration processes in the receiver circuits. It would also be necessary to inject errors in the 1200 Hz data signals, to test the data validation protocols and any Automatic Request for Re-transmission or Acknowledgement features built into the network node receivers.

Standard "**white**" or "**pink**" noise generators were available as were some digital error generators but none offered the facility to inject digital errors into data signals operating in the 1.5 MHz region, nor could they be synchronised to several different data frequencies. Since no commercial equipment was available with the required facilities, it was decided to construct one as a special piece of test equipment for the project.

The design parameters for the error generator are as follows:-

- Programmable between 1 x 10^{-6} and 9 x 10^{-1}.
- Synchronised to external data sources at up to 1.5 MHz.
- Gaussian Distribution.
- Error distribution constant over range of data input frequencies.
- Seperate **Error** and **Data** + **Error** signals.
- Transparent mode for data (ie. no errors injected).

B.3 Circuit Description

Figure B1 shows the block diagram layout of this circuit. The basis of operation of the error generator is that two 24 bit binary numbers "**A**" and "**B**" are compared with each other and an error generated if "**B**" is greater than "**A**". The number "**A**" is generated by a pseudo random sequence generator, block 1 in Figure B1. In this case a 31 stage Maximal Length Sequence generator producing a range of possible values between 1 and 2^{31}-1, ie. 1 and 2,147,483,647. Of the 31 output stages,24 are actually used for comparison purposes. This gives a range of values between 1 and 16,777,215. This MLS generator is clocked by a high speed clock (block 2), the frequency of which must be at least 24 times the data rate required. With the chip sequence data rate of 1.2288 MHz, therefore, the high speed clock rate f_H, must

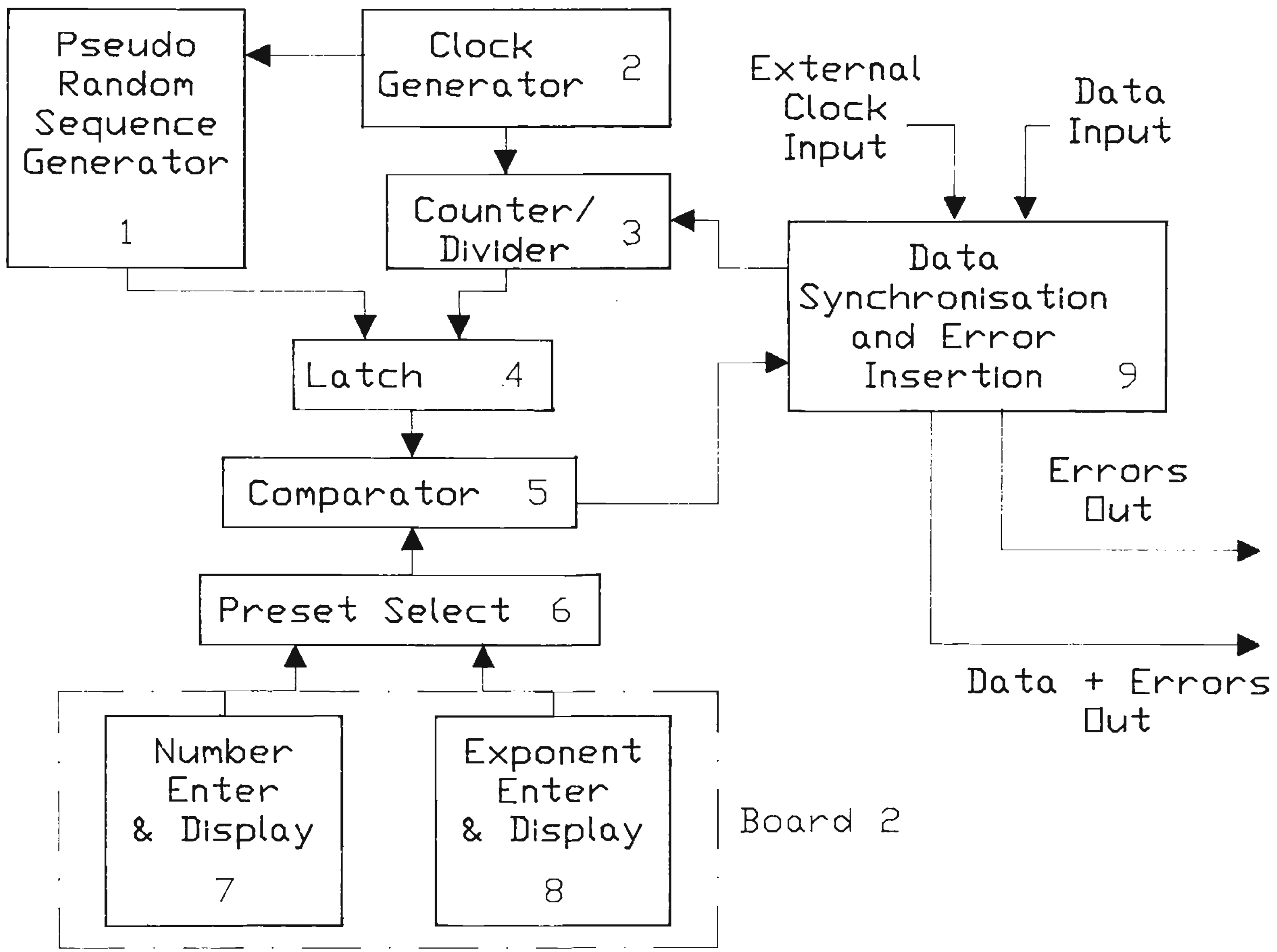


Figure B1 Block Diagram of Error Generator

be at least 29.4912 MHz. 36 MHz was, in fact, chosen for convenience giving a maximum data rate of 1.5 MHz .

The reason why the clock must be at least 24 times the data rate is that the pseudo-random sequence generator must be clocked 24 times to create a statistically independent value of "**A**" to be used in the comparison. This ensures that error bits are generated independently. This complete change of comparison number "**A**" is ensured by using a divide by 24 counter circuit (block 3) which causes a new 24 bit number to be loaded into a latch circuit (block 4) ready for comparison in the comparator circuit (block 5). The other input to the comparator number "**B**", is produced from a look-up table comprising of Read Only Memories, (block 6) which have been pre-loaded with values which correspond to equivalent error rates of between 1 in 10^{-6} and 9 in 10^{-1} when compared to the 24 bit number "**A**". The output value from the look-up table is produced by two select inputs, the number select and display (block 7) and the exponent select and display (block 8). The outputs from these circuits act as address inputs to the look-up table. The range of values available from these select circuits must, therefore, be restricted as follows:-

- Number Range 1 - 9

- Exponent Range 1 - 6

The functional block which makes the error generator so versatile is the data synchronisation and error insertion circuit (block 9). This circuit ensures that only one comparison is made per data bit and also that a new 24 bit number for "**A**" has been used. This circuit also synchronises error insertion and re-times the data after errors have been injected.

Table B1 shows the values pre-loaded into the look-up table. The table shows columns for:-

- Error rate selected
- Address used for look-up purposes
- Decimal look-up value
- Hexadecimal look-up value

Figure B2 shows the complete circuit diagram for board 1 of the error generator and Figure B3 shows board 2, ie. the preset select and display circuits. Table B2 and B3 are the Integrated Circuit component lists for these two boards and the **IC** numbers are cross referenced to Figures B2 and B3 respectively.

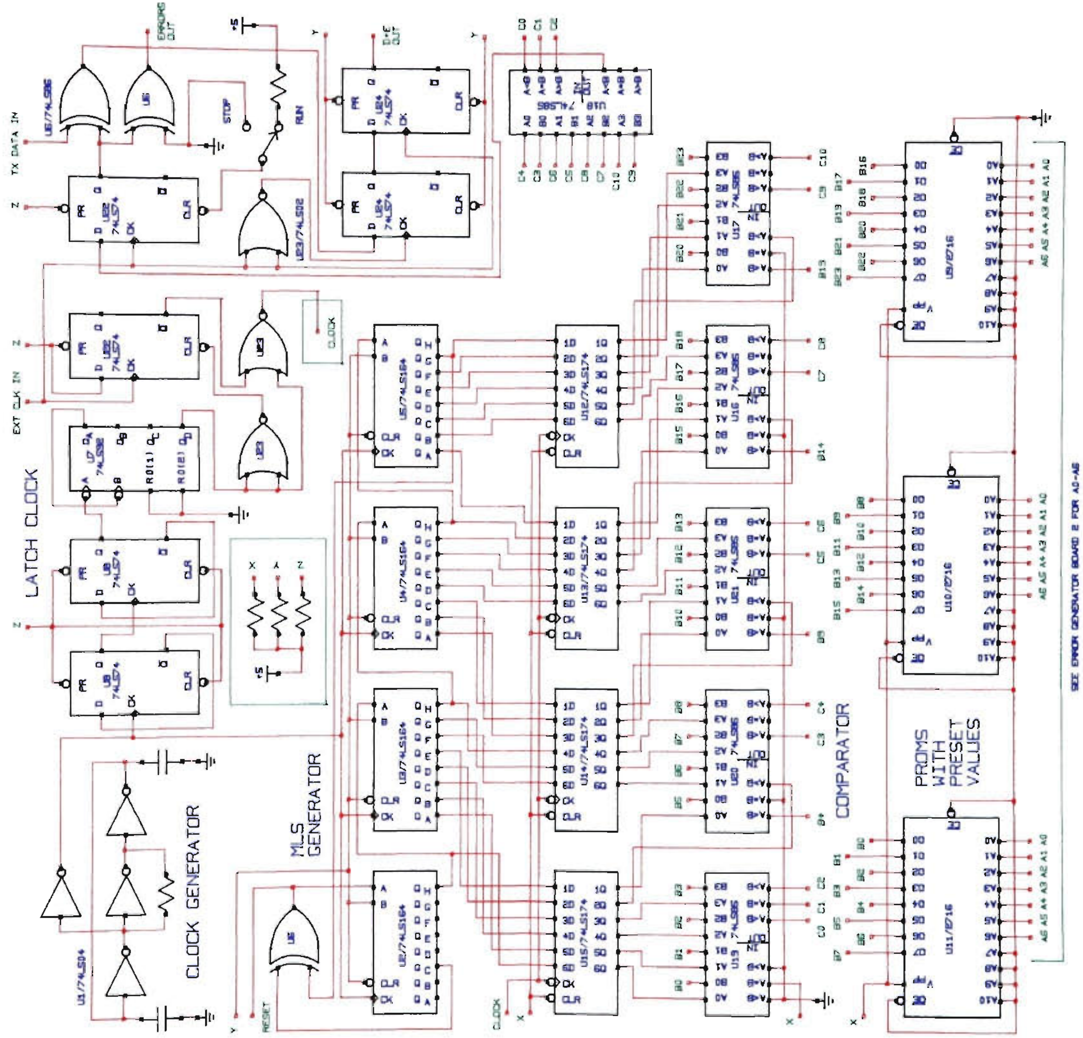


FIGURE B2 CIRCUIT DIAGRAM FOR ERROR GENERATOR BOARD 1

NOTE : ALL RESISTORS ARE 1K 1/8 WATT 5%

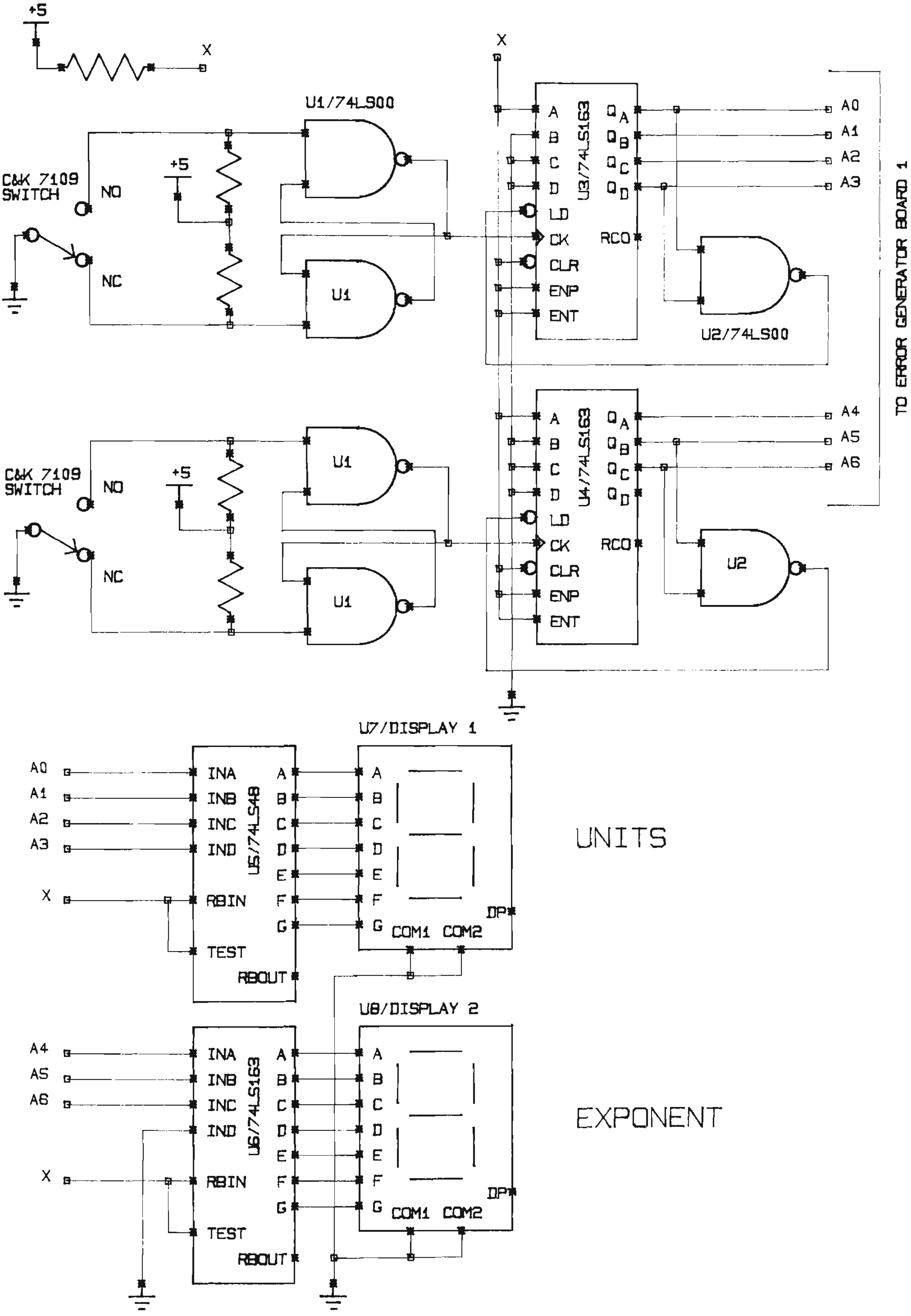


FIGURE B3 CIRCUIT DIAGRAM FOR
ERROR GENERATOR BOARD 2

Table B1 - Error Generator PROM Details

Error Rate	Hex Address	Decimal Value	Hex Value	Error Rate	Hex Address	Decimal Value	Hex Value
1x10^{-6}	61	17	11	1x10^{-3}	31	16777	4189
2	62	34	22	2	32	33554	8312
3	63	50	32	3	33	50332	C49C
4	64	67	43	4	34	67109	10625
5	65	84	54	5	35	83886	147AE
6	66	101	65	6	36	100663	18937
7	67	117	75	7	37	117441	1CAC1
8	68	134	86	8	38	134218	20C4A
9	69	151	97	9	39	150995	24DD3
1x10^{-5}	51	168	A8	1x10^{-2}	21	167772	28F5C
2	52	336	150	2	22	335544	51E68
3	53	503	1F7	3	23	503317	7AE15
4	54	671	29F	4	24	671089	A3D71
5	55	839	347	5	25	838861	CCCCD
6	56	1007	3EF	6	26	1006633	F5C29
7	57	1174	496	7	27	1174405	11EB85
8	58	1342	53E	8	28	1342177	147AE1
9	59	1510	5E6	9	29	1509949	170A3D
1x10^{-4}	41	1678	68E	1x10^{-1}	11	1677722	19999A
2	42	3355	D1B	2	12	3355443	333333
3	43	5033	13A9	3	13	5033165	4CCCCD
4	44	6711	1A37	4	14	6710886	666666
5	45	8389	20C5	5	15	8388608	800000
6	46	10066	2752	6	16	10066329	999999
7	47	11744	2DE0	7	17	11744051	B33333
8	48	13422	346E	8	18	13421772	CCCCCC
9	49	15099	3AFB	9	19	15099493	E66665

Table B2 - Error Generator Board 1 - Chip List

Number	Type	V_{cc}	GND.	No. Of Pins
1	04	14	7	14
2	164	14	7	14
3	164	14	7	14
4	164	14	7	14
5	164	14	7	14
6	86	14	7	14
7	92	10	5	14
8	74	14	7	14
9	2716	24	12	24
10	2716	24	12	24
11	2716	24	12	24
12	174	16	8	16
13	174	16	8	16
14	174	16	8	16
15	174	16	8	16
16	85	16	8	16
17	85	16	8	16
18	85	16	8	16
19	85	16	8	16
20	85	16	8	16
21	85	16	8	16
22	74	14	7	14
23	02	14	7	14
24	74	14	7	14

NOTE All **TTL** devices are "**LS**"

Table B3 - Error Generator Board 2 - Chip List

Number	Type	V_{cc}	GND.	No. Of Pins
1	00	14	7	14
2	00	14	7	14
3	163	16	8	16
4	163	16	8	16
5	48	16	8	16
6	48	16	8	16
7	Disp	--	4,12	14
8	Disp	--	4,12	14

NOTE The displays are common cathode and use 74LS48 drivers. If common cathode displays are not available it will be necessary to use common anode devices and 74LS47 drivers. All **TTL** devices are "**LS**".

B.4 Circuit Testing and Discussion of Results

In order to use this error generator, in the validation of the **SS LAN** circuits, its performance had to be verified. The performance was tested thoroughly with respect to the following parameters:-

- Accuracy of Preset Error Rate

- Distribution of Errors within a preset rate

- Consistency of Error rate with input data rate variation

- Spread of single bit and multi-bit errors

Various tests were carried out to validate these parameters. Table B4 is a list of definitions used to describe the conditions being tested. In this table zeros and ones are used to represent the presence or absence of errors, ie. a single bit error is identified by the pattern **010**, and a double bit error by the pattern **0110**. The reference letters **A** to **J** may then be used when describing the results obtained. Figures B4, B5 and B6 show the special test circuits built to create the conditions identified by the reference letters **A** to **J**

B.4.1 Test #1

- Data rate set to 1 MHz.

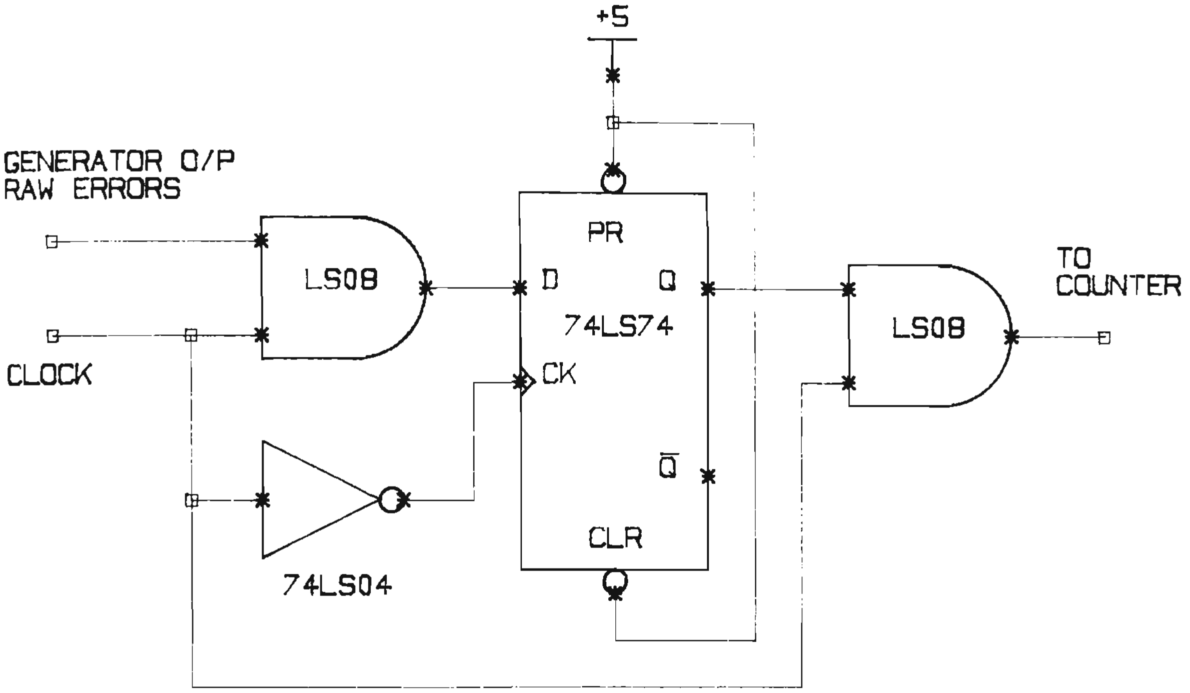
- Error rate pre-select set to 1 in 10.

- Test circuit A in Figure B4 used.

The timer counter was set to record events over a 10 second period. 100 results were collected and a histogram of the spread of the results was produced, Figure B7. This figure shows the distribution of all errors produced.

CIRCUIT FOR TEST A

RAW ERRORS = COUNT B



CIRCUIT FOR TEST C

SINGLE BIT ERRORS

0 1 0

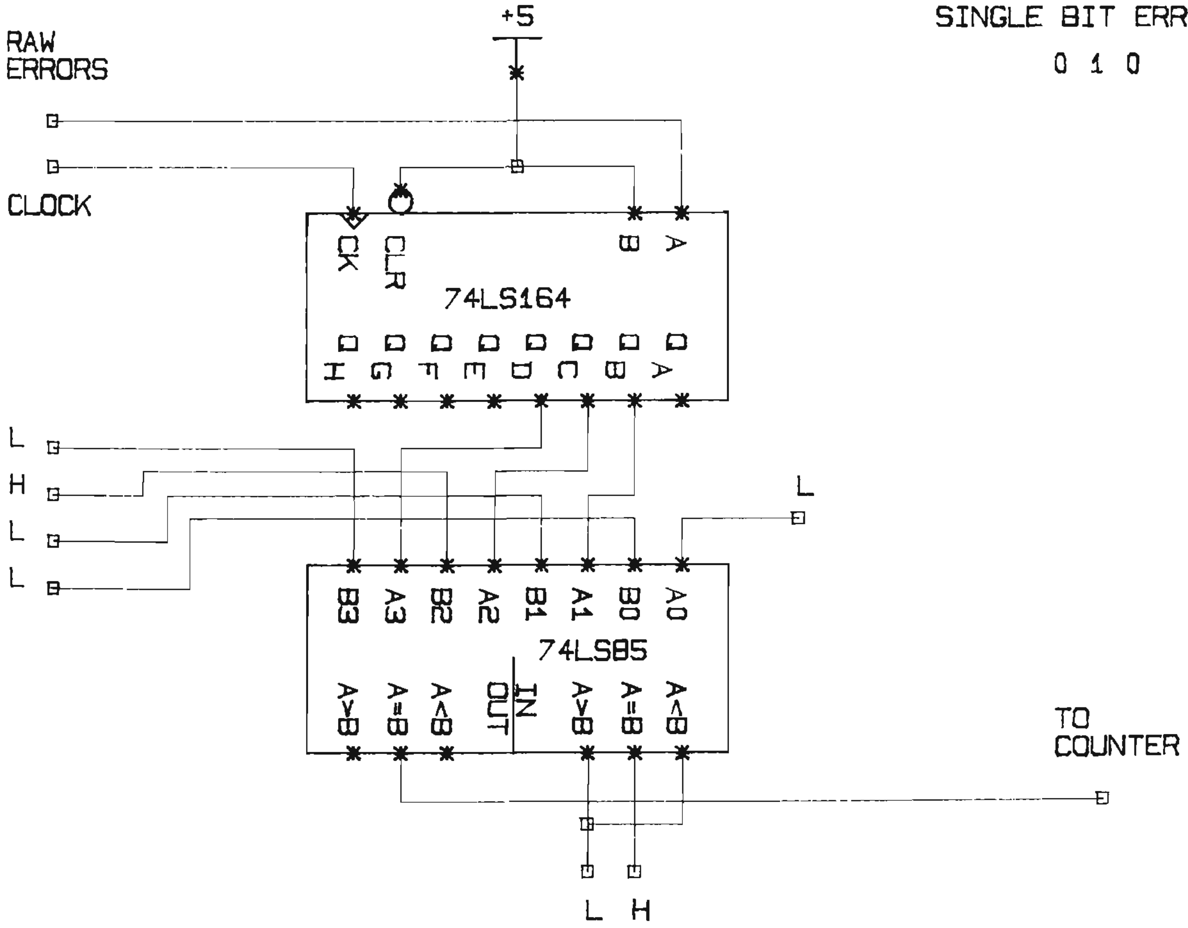
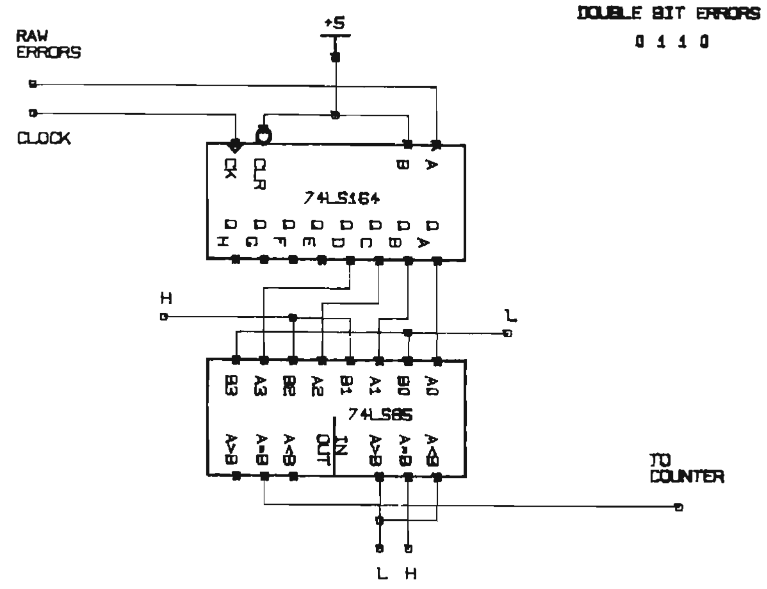
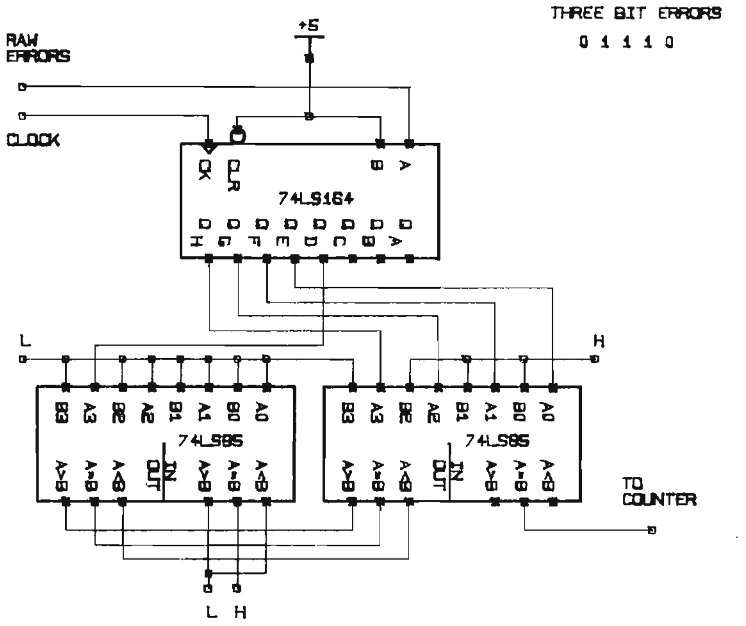


FIGURE B4 TEST CIRCUITS A AND C

CIRCUIT FOR TEST D



CIRCUIT FOR TEST E



CIRCUIT FOR TEST F

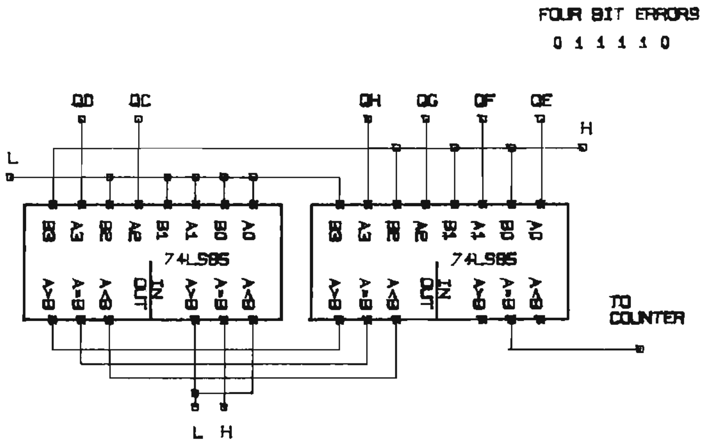
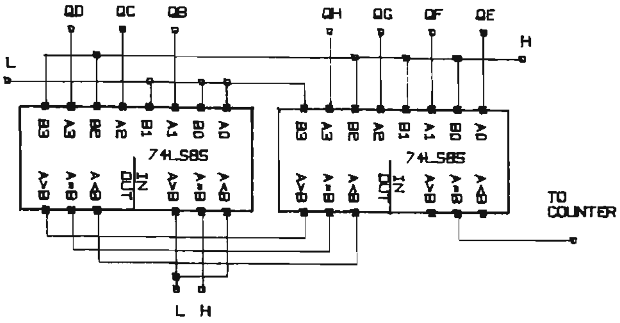


FIGURE B5 TEST CIRCUITS D,E AND F

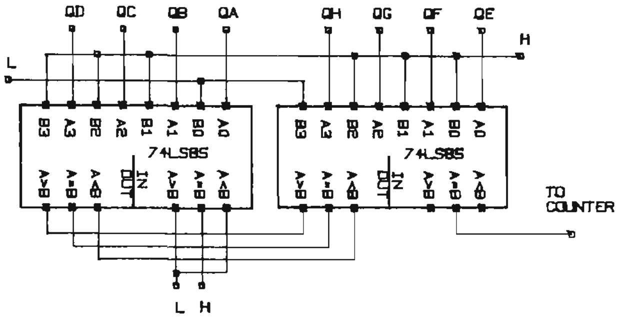
CIRCUIT FOR TEST G

FIVE BIT ERRORS
0 1 1 1 1 1 0



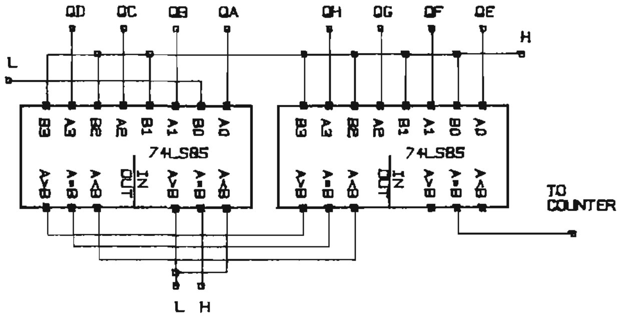
CIRCUIT FOR TEST H

SIX BIT ERRORS
0 1 1 1 1 1 1 0



CIRCUIT FOR TEST I

SEVEN (OR MORE)
BIT ERRORS
0 1 1 1 1 1 1 1



CIRCUIT FOR TEST J

EIGHT (OR MORE)
BIT ERRORS
1 1 1 1 1 1 1 1

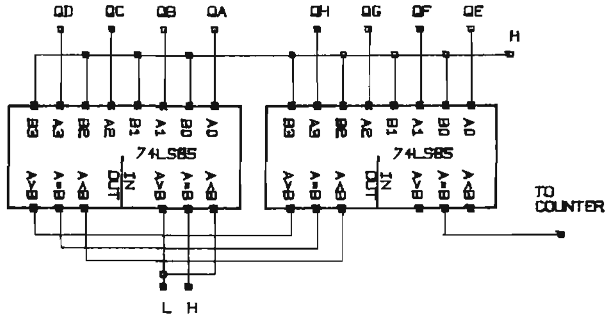


FIGURE B6 TEST CIRCUITS G,H,I AND J

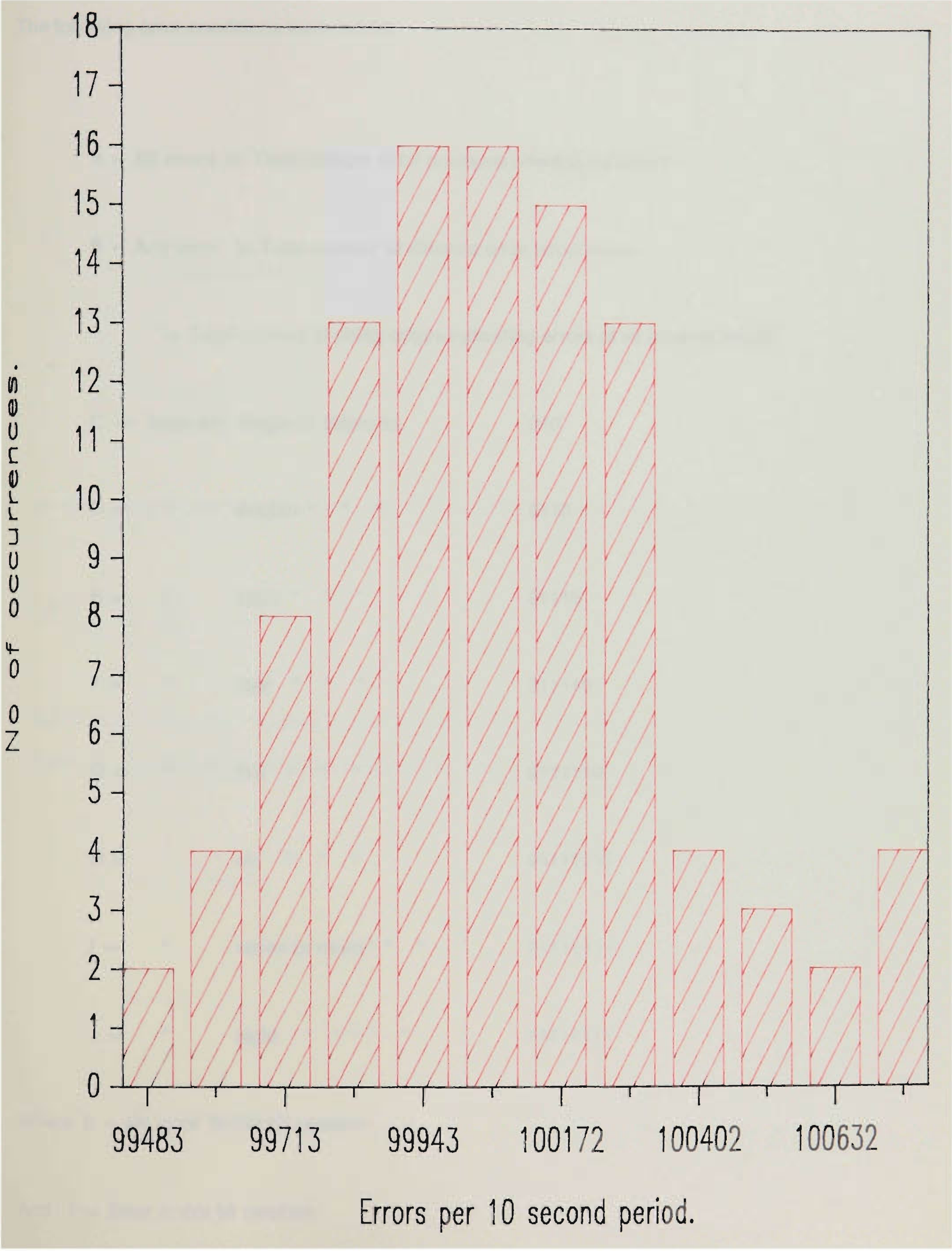


Figure B7 Distribution of All Errors (A). F = 1Mhz. R = 1 in 10

Table B.4 - Error Generator Test Circuit Definitions

The following error conditions were tested:

A = **All** errors ie. Total number of bit positions affected by errors

B = **Any** error ie. Total number of different error occurences

ie. Total number of rising edges indicating errors of all (or any) length.

C = Separate single bit errors ie. 010

D = " double " " " 0110

E = " triple " " " 01110

F = " four " " " 011110

G = " five " " " 0111110

H = " six " " " 01111110

I = " seven or more " " " 01111111

J = " eight " " " " " " 11111111

Where 0 = No error in this bit position

And 1 = Error in this bit position

The main statistics from this set of results are:-

- The sum is 10000383
- The mean is 100003.83
- The variance of the samples is 78970.143
- The standard deviation of the samples is 281.016

The mean value represents an error rate of 1.0000383 x 10^{-1}compared to the expected value of 1 x 10^{-1}.

B.4.2 Test #2

These results show the raw errors or the total number of different error occurrences in the 10 second period. The results histogram is included as Figure B8. The statistics are:-

- Sum 8410954
- Mean 84109.54
- Variance 50361.827

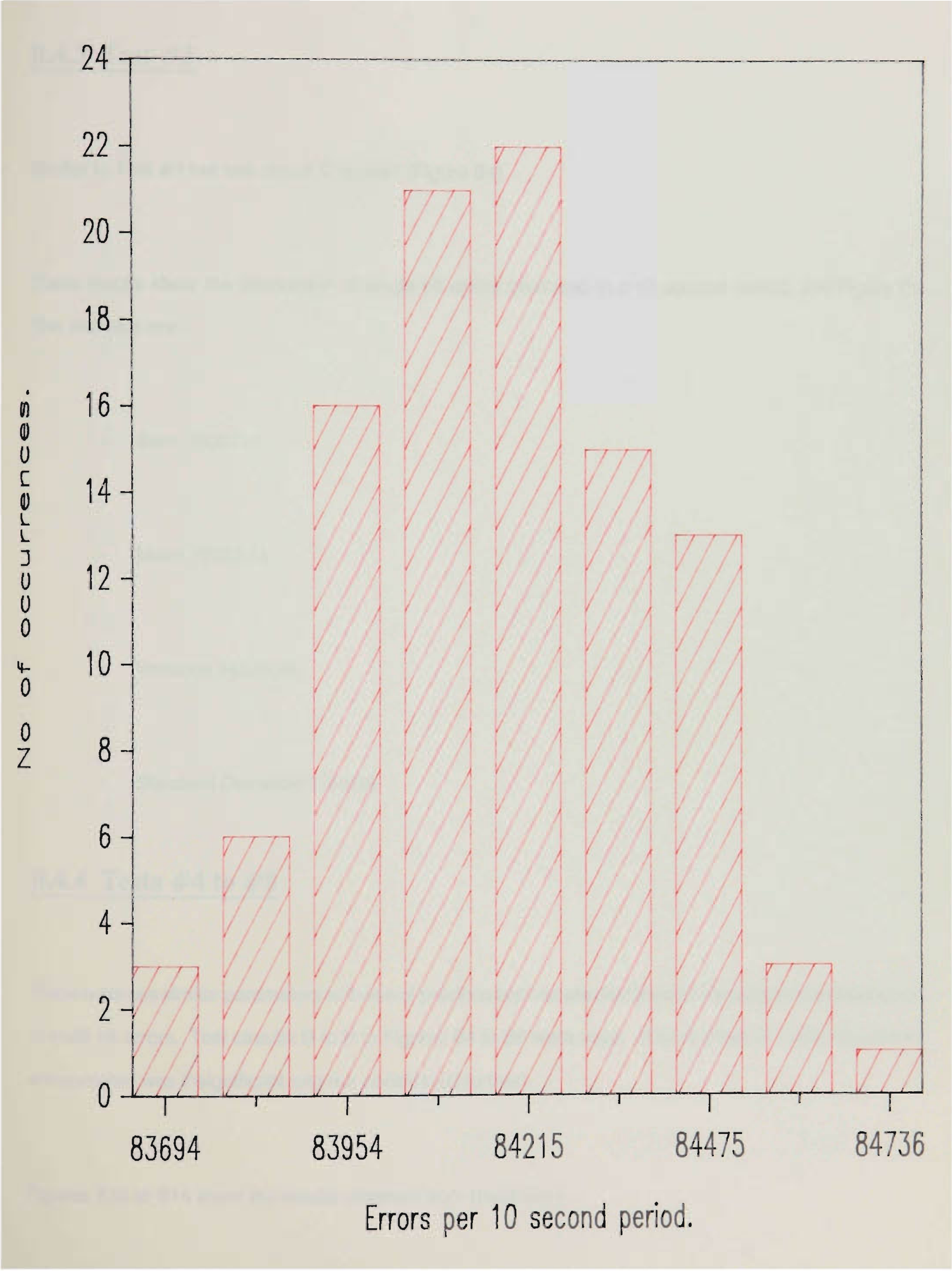


Figure B8 Distribution of Any Error (B). F = 1MHz. R = 1 in 10

- Standard Deviation 224.414

B.4.3 Test #3

Similar to Test **#1** but test circuit **C** is used (Figure B4)

These results show the distribution of single bit errors produced in a 10 second period, see Figure B9. The statistics are:-

- Sum 7030714

- Mean 70307.14

- Variance 44376.99

- Standard Deviation 210.658

B.4.4 Tests #4 to #8

These tests use similar parameters to those of previous test but are designed to investigate the distribution of multi-bit errors. Test circuits **D** to **H** in Figures B4 to B6 were used. (The number of 7 and 8 bit errors encountered was insignificant and not considered further).

Figures B10 to B14 show the results obtained from these tests.

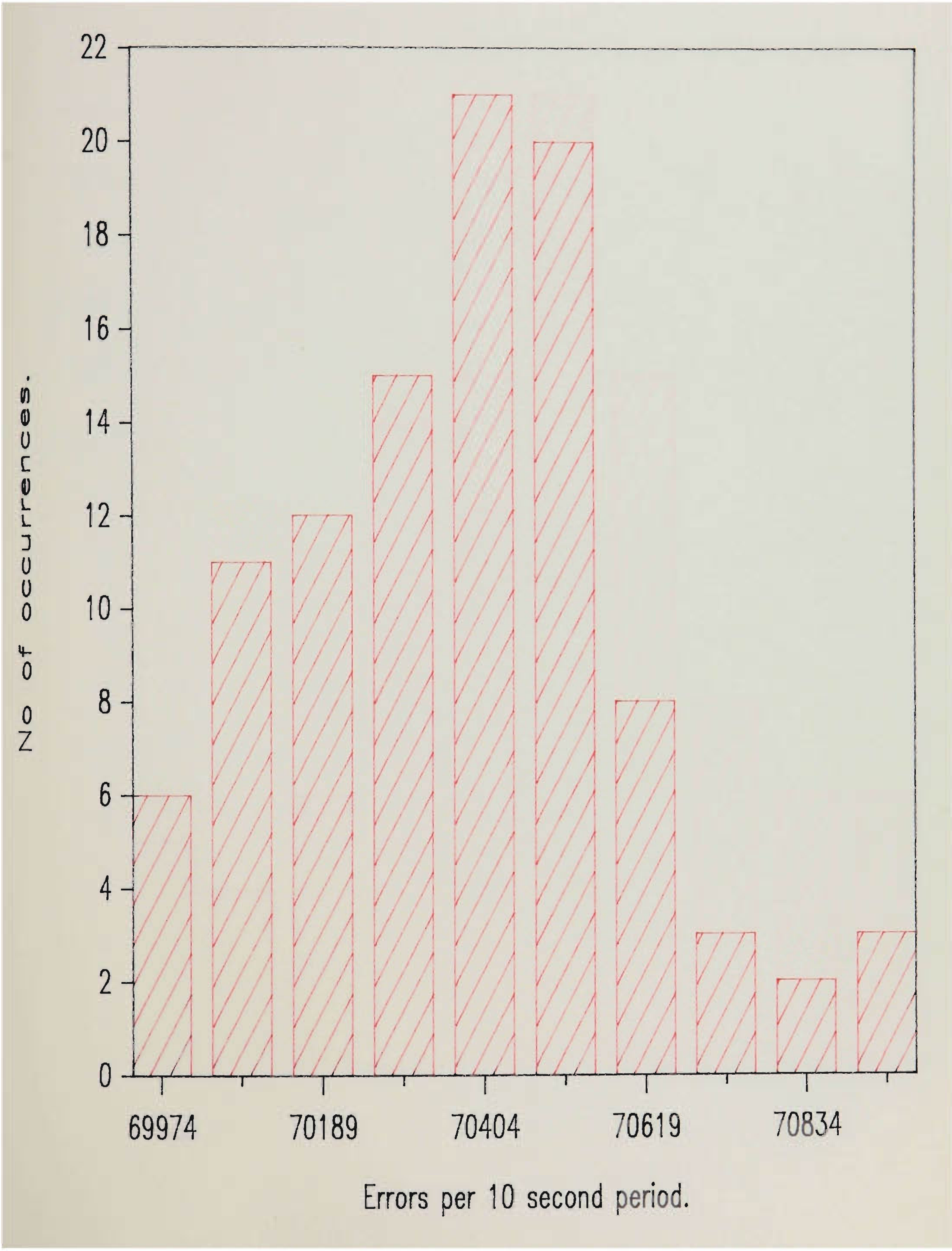


Figure B9 Distribution of 1 Bit Errors (C). F = 1MHz. R = 1 in 10

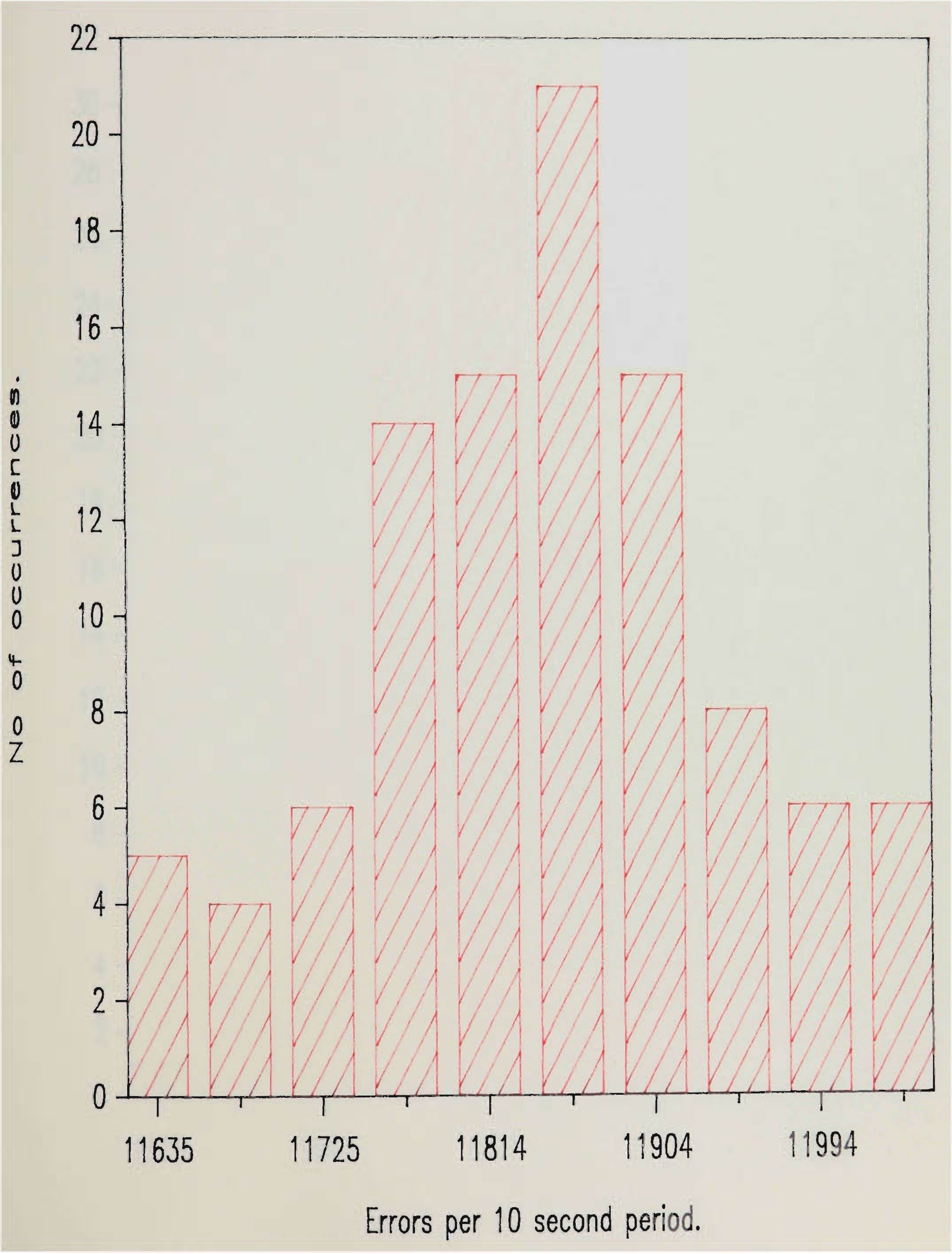


Figure B10 Distribution of 2 Bit Errors (D). F = 1MHz. R = 1 in 10

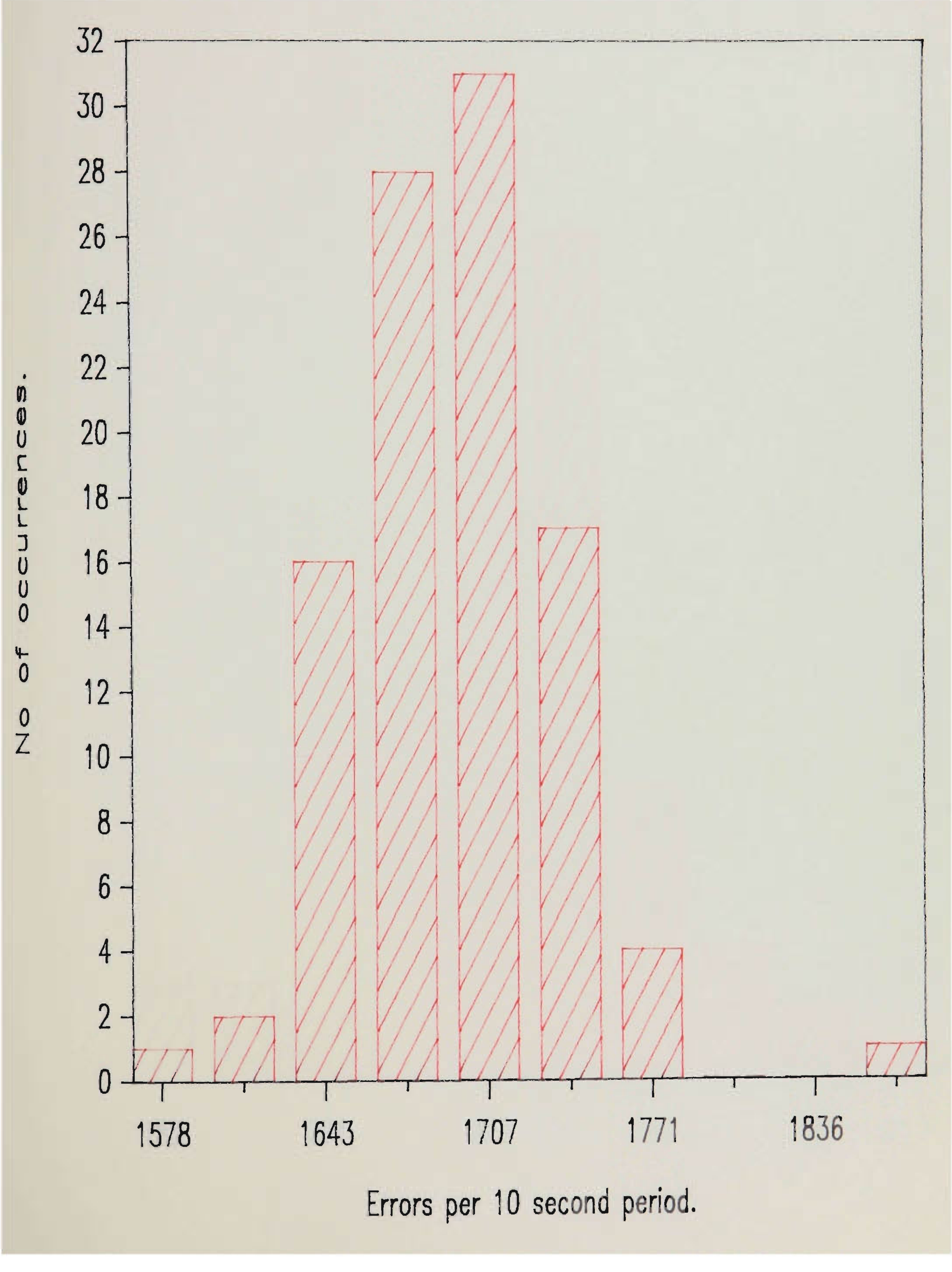


Figure B11 Distribution of 3 Bit Errors (E). F = 1MHz. R = 1 in 10

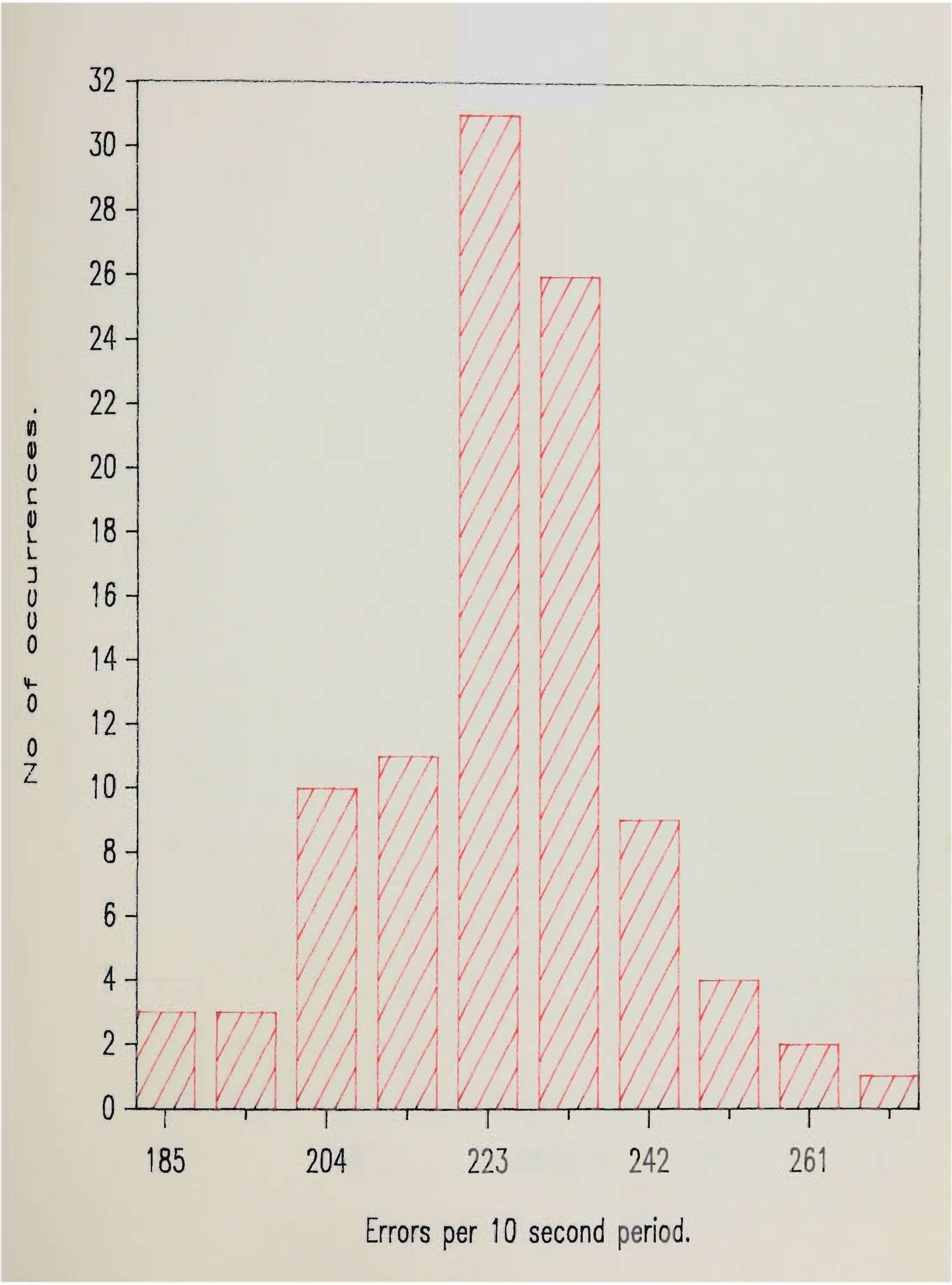


Figure B12 Distribution of 4 Bit Errors (F). F = 1MHz. R = 1 in 10

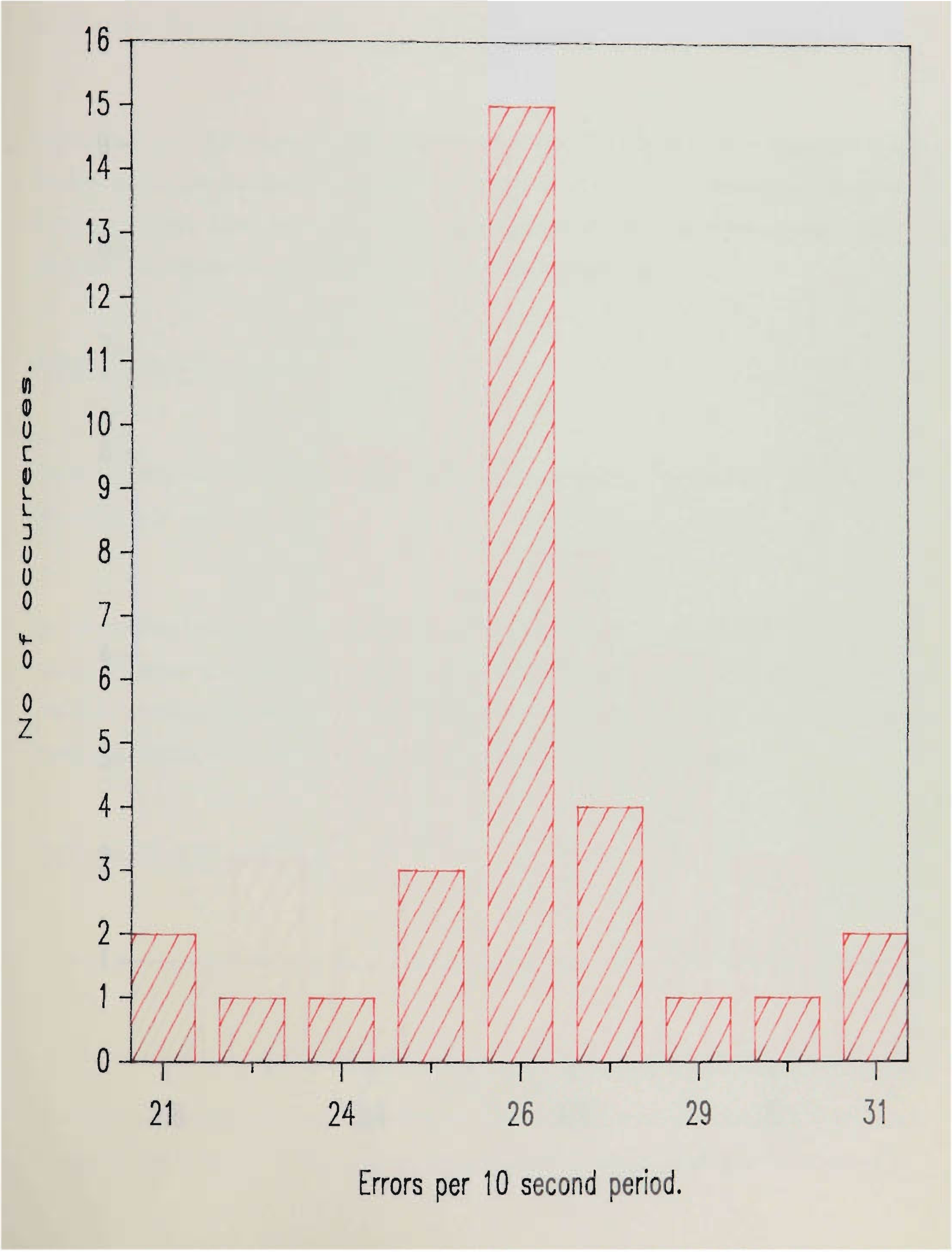


Figure B13 Distribution of 5 Bit Errors (G). F = 1MHz. R = 1 in 10

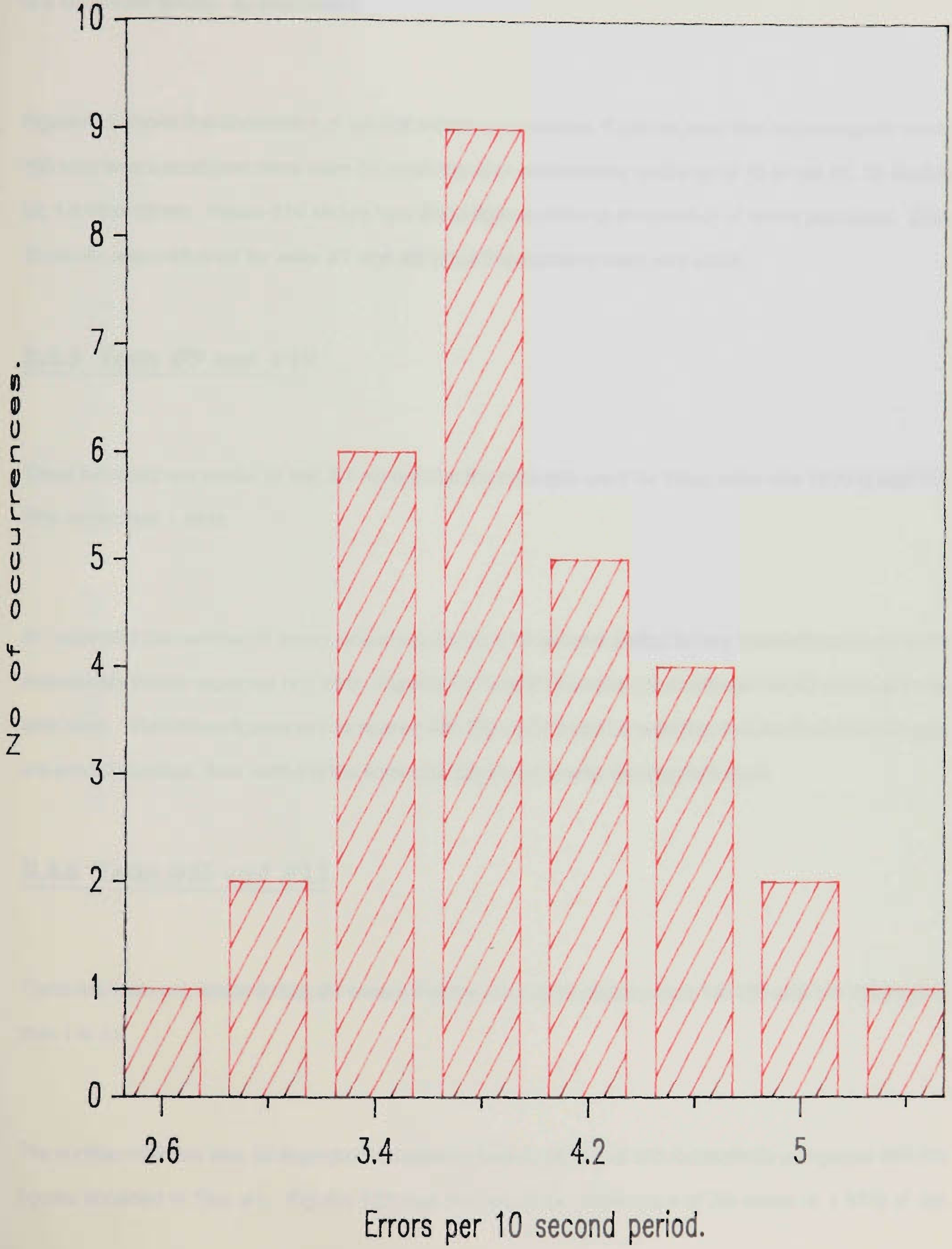


Figure B14 Distribution of 6 Bit Errors (H). F = 1MHz. R = 1 in 10

Tests **#1** to **#8** were designed to map the distribution of all errors produced. These tests were carried out at a single frequency. It can be seen from these results that the distribution of errors does approximate to a **Gaussian** spread, as anticipated.

Figure B15 shows the distribution of separate error occurrences. It can be seen that on average for every 100 total errors produced there were 84 separate error occurrences made up of 70 single bit, 10 double bit, 1.6 triple bit etc. Figure B16 shows how these figures relate to the number of errors produced. Only 30 results were obtained for tests **#7** and **#8** since the numbers were very small.

B.4.5 Tests #9 and #10

These two tests are similar to test **#1** except that the data rate used for these tests was 10 KHz and 100 KHz rather than 1 MHz.

As expected the number of errors occurring during a 10 second period is one hundreth and one tenth respectively of that expected at 1 MHz. Figures B17 and B18 shows the distribution of All errors at these data rates. When these figures are compared with Figure B7 it can be seen that the distribution envelopes are almost identical, thus verifying the error rate consistency with varying data rates.

B.4.6 Tests #11 and #12

These two tests are similar to test **#1** except that the error rates chosen were 1 in 100 and 1 in 1000 rather than 1 in 10.

The number of errors was, as expected, reduced by factors of 10 and 100 respectively compared with the figures obtained in Test **#1**. Figures B19 and B20 show the distribution of **All** errors at 1 MHz at the

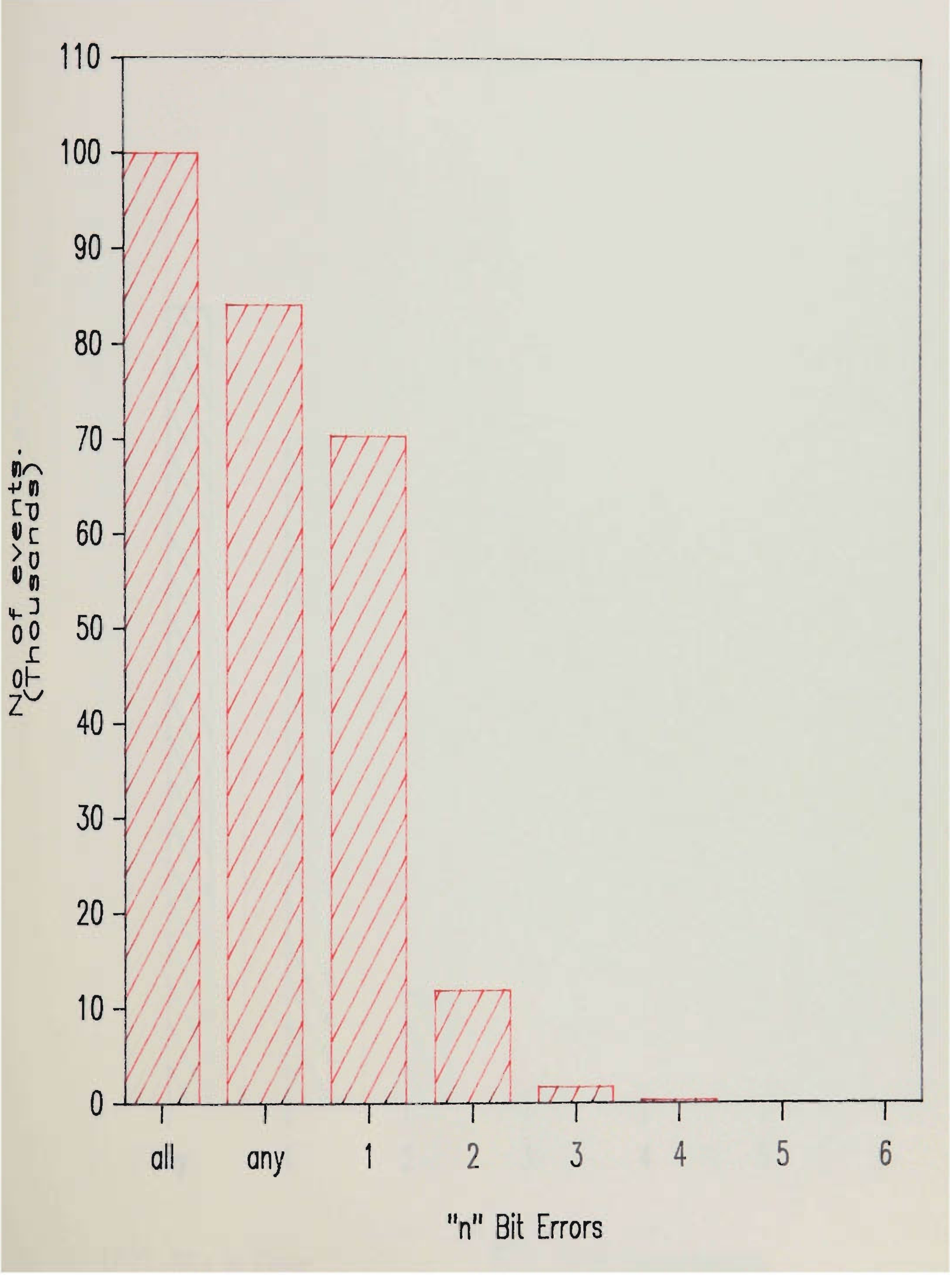


Figure B15 Distribution of Separate Error Occurrences

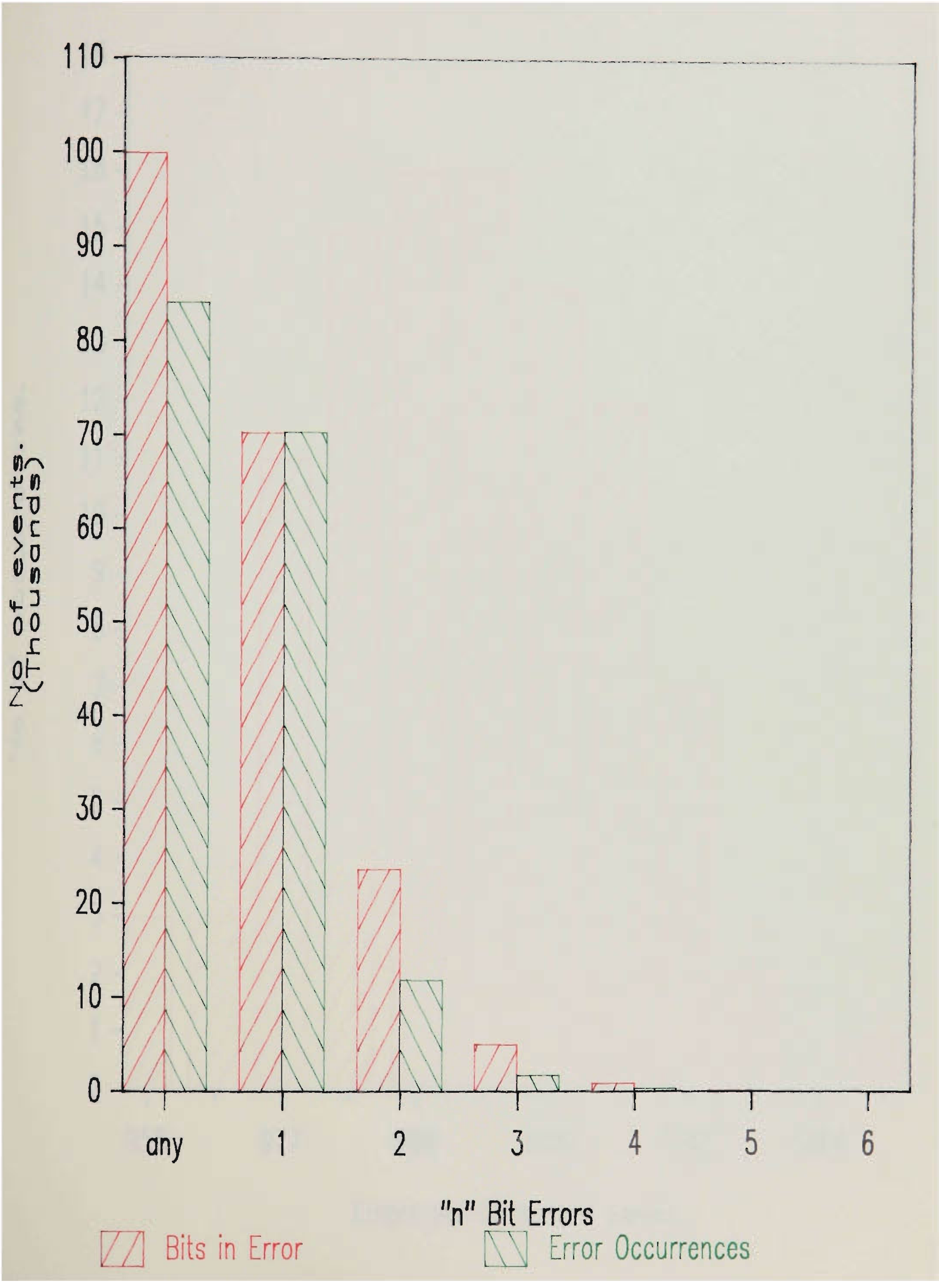


Figure B16 Multi-Bit Error Characteristics

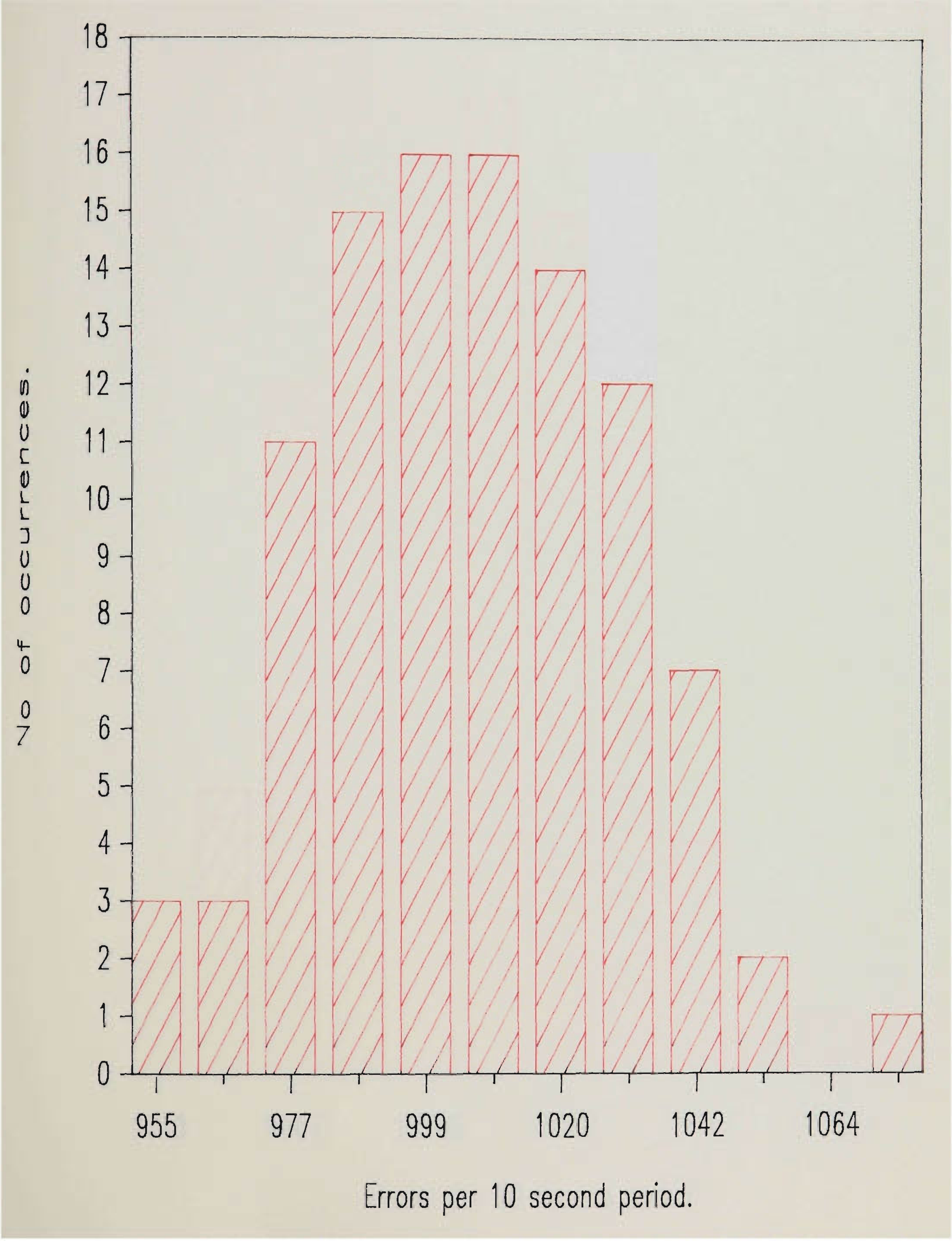


Figure B17 Distribution of All Errors (A). F = 10KHz. R = 1 in 10

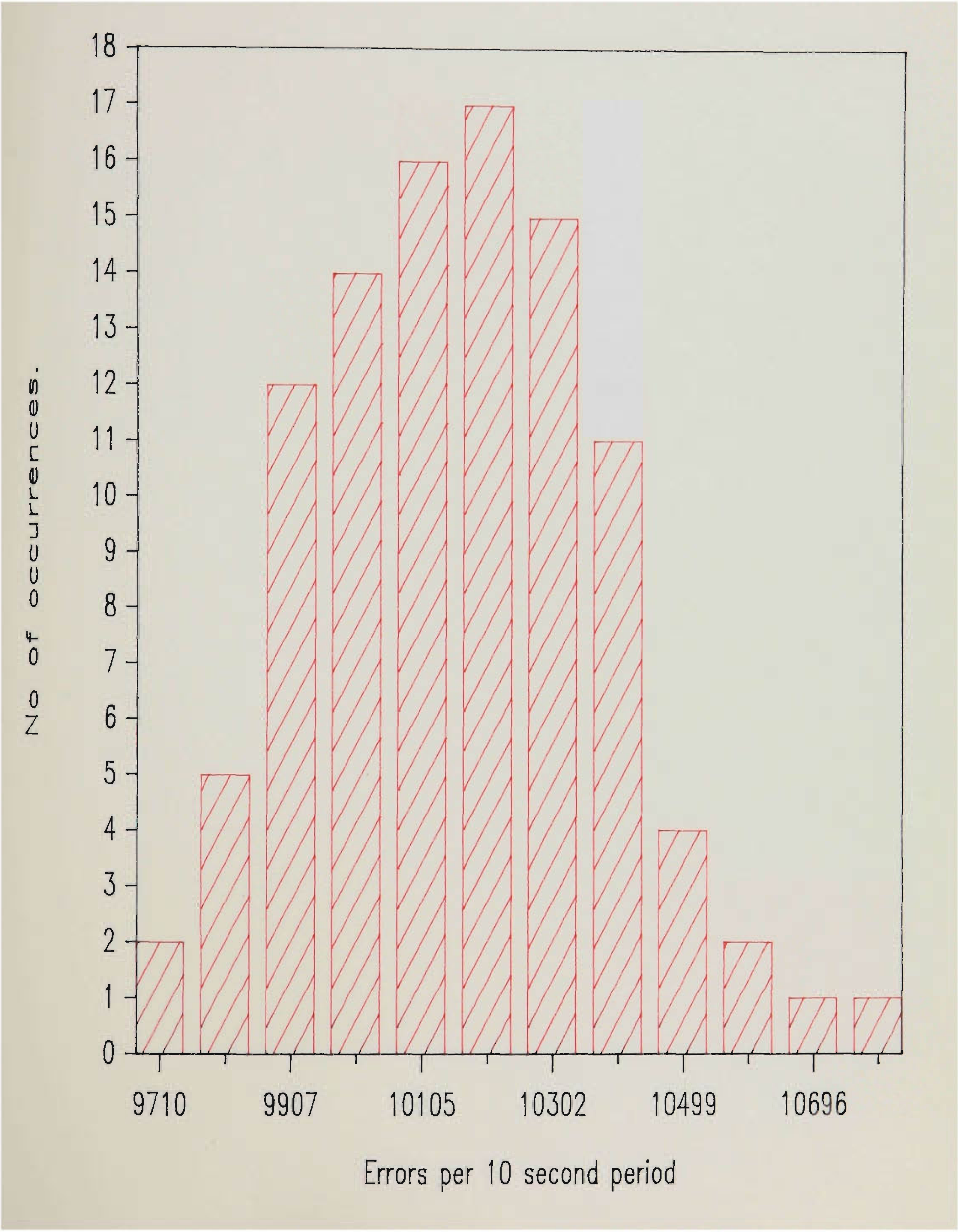


Figure B18 Distribution of All Errors (A). F = 100KHz. R = 1 in 10

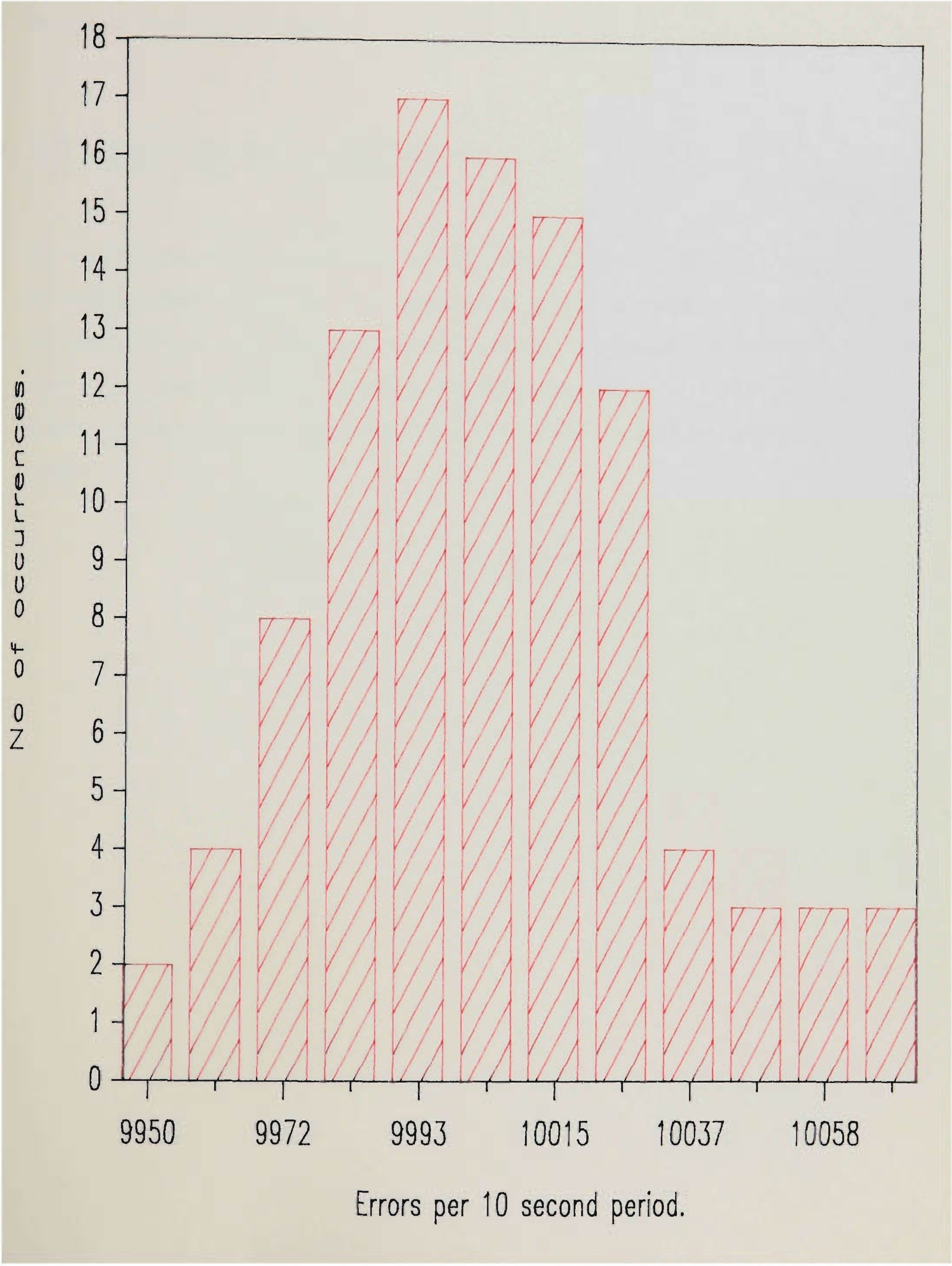


Figure B19 Distribution of All Errors (A). F = 1MHz. R = 1 in 100

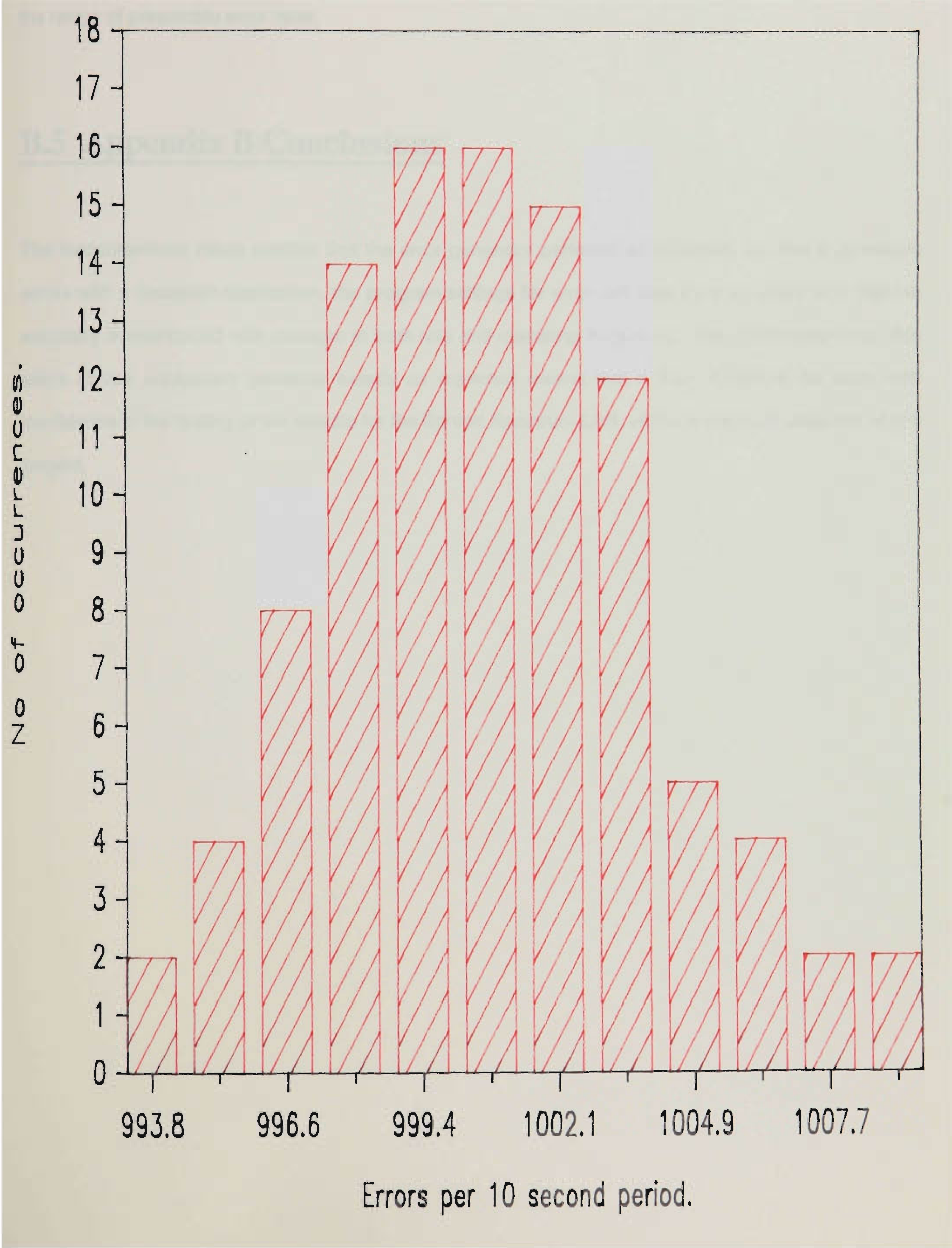


Figure B20 Distribution of All Errors (A). F = 1MHz. R = 1 in 1000

selected error rates. When these figures are compared with Figure B7 it can be seen that the distribution envelopes are very similar. This confirms the distribution properties and accuracy of rate selection across the range of presetable error rates.

B.5 Appendix B Conclusions

The measurements made confirm that the error generator performs as expected, i.e. that it generates errors with a Gaussian distribution, the program settings for error rate select are accurate, and that the accuracy is maintained with changes in both rate and operating frequency. The confirmation that this piece of test equipment performs exactly as expected means that it may, therefore, be used with confidence in the testing of the circuits for the Spread Spectrum LAN, which is the main objective of this project.

APPENDIX C

C THE ANALOGUE MULTIPLIER

C.1 Summary

This appendix includes details of the calculations made to choose the required peripheral circuit component values associated with the RS 1495 Analogue Multiplier. The method of component calculation follows very closely that illustrated in the component literature.

C.2 Theory of Operation

Figure C1 shows a circuit schematic of the analogue multiplier with calculated component values and the level shifter already included. Figure C2, which shows the internal circuit schematic is taken from the component data sheet. These circuits are referred to frequently in the following text.

The multiplier operates on the principle of variable transconductance. The differential output current (seen at pins 2 and 14) is given by:

$$I_{14} - I_2 = (2 \times V_x \times V_y) / (R_x \times R_y \times I_3)$$

where V_x and V_y are the input voltages and I_3 is the scale factor current.

The scale factor is a value multiplied by the output to reduce the size. eg. For a scale factor 1/10 then $V_o = V_x \times V_y/10$. This scale factor simply prevents overdriving of the amplifier causing non linear opeation.

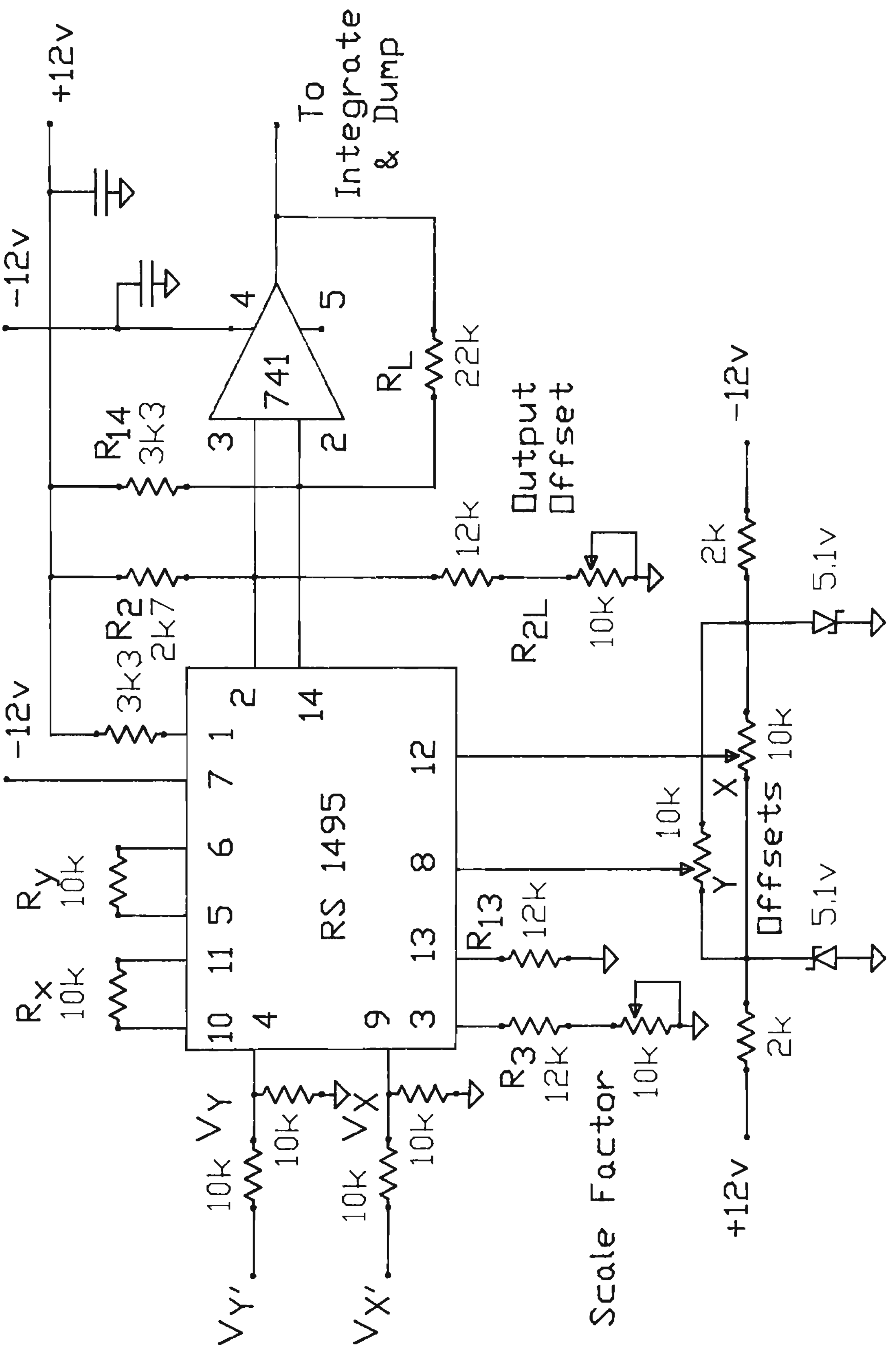


Figure C1 Circuit Schematic of Analogue Multiplier

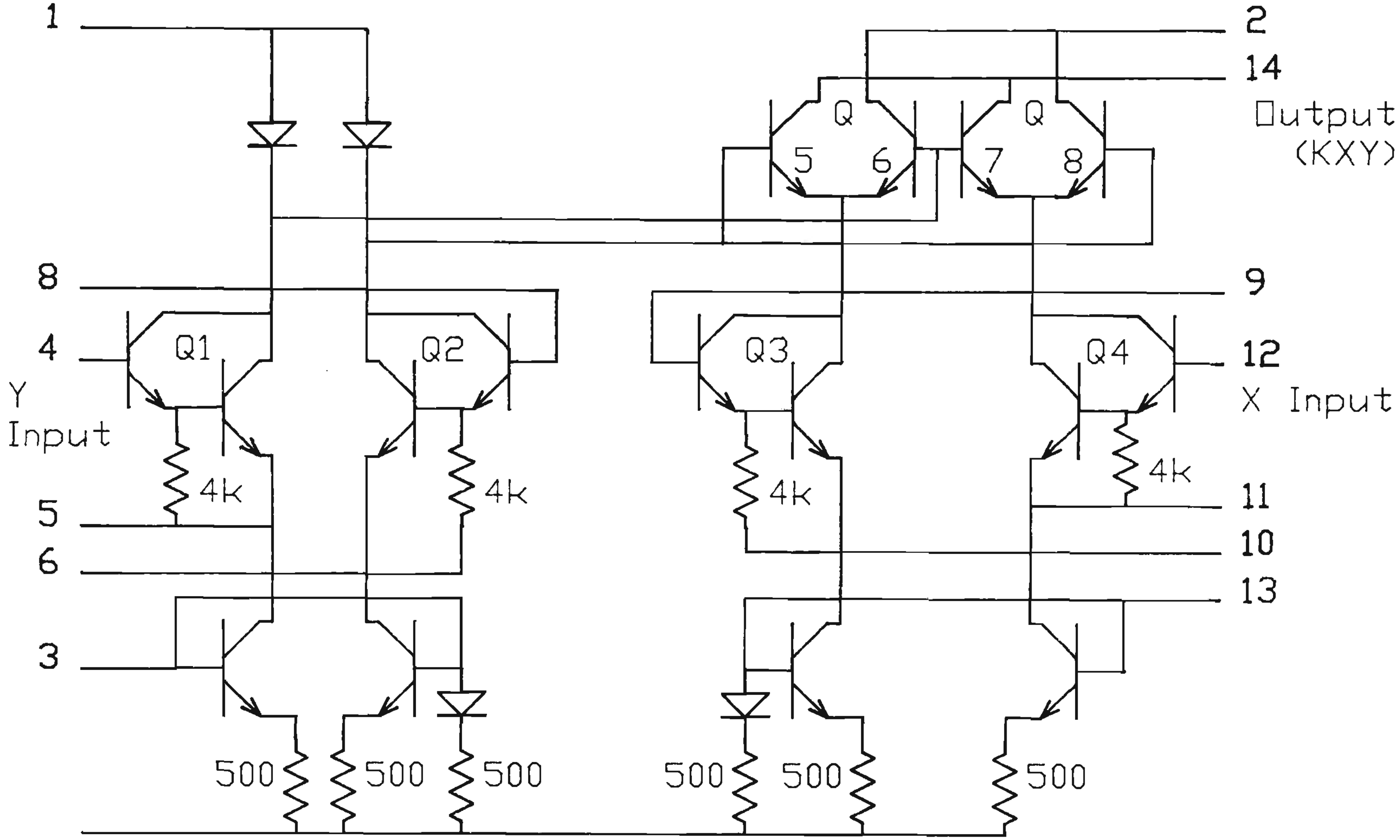


Figure C2 Internal Schematic of RS1495

The potential dividers on the inputs X and Y simply limit the maximum multiplier input to half the value of the V_x and V_y input voltages. Once again, this is done simply to prevent the occurrence of non linear operation if the inputs become too large.

The resistors R_x and R_y ensure the input transistors are always active, and I_{13} and I_3 flow through R_x and R_y respectively. To ensure the input transistors are always active then the voltages $I_{13}R_x$ and I_3R_y should be larger than the input voltages V_x and V_y, and the larger they are, with respect to V_x and V_y, the more accurate the multiplier will be.

Pins 8 and 12 provide facilities for an X and Y offset adjustment. Even when transistor base emitter junctions are matched within 1mV and resistors within 2%, errors in the output can still occur. Correction must, therefore, be allowed for. The arrangement shown in Figure C1 allows for precise adjustment of the X and Y inputs. These potentiometers combine with the output offset adjustment, connected to pin 2, give complete output control.

ie. $V_o = K(V_x \pm V_{iox} \pm V_{xoff})\ (V_y \pm V_{ioy} \pm V_{yoff}) \pm V_{oo}$

- K = scale factor (see section C.4)
- V_x = X input voltage
- V_y = Y input voltage
- V_{iox} = X input offset voltage

- V_{ioy} = Y input offset voltage

- V_{xoff} = X input offset adjust voltage

- V_{yoff} = Y input offset adjust voltage

- V_{oo} = output offset voltage

C.3 Level Shifting

In order to convert the differential output between pins 2 and 14 on the multiplier to a single ended output voltage, referenced to ground, a level shifter is required. The level shifter is designed separately to the analogue multiplier. The level shifter simply takes the voltages on pins 2 and 14 and obtains the difference, the result being used as the single ended output. ie.

$$V_o = (I_2 - I_{14}) \times R_1$$

Provided that in both cases R_1 is equal, then no errors should occur in the output. Perfect fixed resistor matching is, however, not possible and mismatches can be compensated for by using the offset adjust potentiometers as described above.

C.4 Analogue Multiplier Design Procedure

In this description reference is made to Figure C2.

The resistors at the input are chosen to limit the X and Y inputs to ± 5 volts.

This is for a $V'_x = V'_y =$ nominally ± 10 volt input

For a scale factor of 1/10 ie. $V_o = (V'_x \times V'_y) / 10$

$$= (2V_x \times 2V_y) / 10$$

$$= 4 / 10\ V_x V_y$$

$\therefore$ scale factor $K = 4/10$

Step one is to select I_3 ie. the current in pin 3. Device limitations indicate that this current cannot exceed 10mA.

$I_{13} = I_3$ and in this case they are both set to equal 1mA.

Setting I_3 and I_{13} only needs the connection of a resistor between pin 13 and ground and between pin 3 and ground.

$$R_{13} + 500 = (V(-) - 0.7) / I_{13}$$

and $$R_3 + 500 = (V(-) - 0.7) / I_3$$

$$V(-) = -12 \text{ volts} \ \& \ I_3 = I_{13} = 1\text{mA}$$

$$\therefore R_3 + 500 = 11.3 / 1\text{mA}$$

$$\therefore R_3 = 10.8\text{k}\,\Omega \approx 12\text{k}\,\Omega \text{ (Nearest Preferred Value)}$$

If R_3 is made equal to 15kΩ by using a 12kΩ resistor and a 10kΩ potentiometer then adjustment of R_3 and I_3 would be allowed for, this also allows a convenient method of making a final trim of the scale factor.

The next step is the consideration of R_x and R_y. If reference is made to the schematic, then to ensure the input transistors will always be active

$$(V_x / R_x) < I_{13} \quad \& \quad (V_y / R_y) < I_3$$

If $V_x = V_y = 5$ volts

Then make $R_x = R_y = 10k\Omega$

$R_x = R_y = 10k\Omega$ is used ,as the larger the I_3R_y and $I_{13}R_x$ products in relation to V_y and V_x respectively the more accurate the multiplier will be.

Next R_L must be determined. Using the following known values R_x, R_y and I_3 , R_L may be calculated using the relationship

$$K = 2R_L / (R_x * R_y * I_3) = 4 / 10$$

$$\therefore R_L = 20k\Omega \approx 22k\Omega \text{ (Nearest Preferred Value)}$$

Next R_1 is chosen so that the voltage at pin one is several volts higher than the maximum input voltage, hence it must be at least 6 volts as $V_x = V_y = 5$ volts, (7 volts was actually used). This is to ensure transistors Q1,Q2,Q3,Q4, are maintained in their active region.

The current in pin 1, $I_1 = 2 \times I_3$ (see schematic)

$$\therefore R_1 = (V(+) - V1) / 2 \times I_3$$

where $V_1 = 7$ volts, $V(+) = 12V$ and $I_3 = 1mA$

$\therefore R_1 = 2.5k\,\Omega \qquad \therefore$ Use $3.3k\,\Omega$

C.5 Level Shifter Design

It can be shown that $V_o = (I_2 - I_{14})\,R_L$, and from the schematic

$$I_2 - I_{14} = 2 \times I_x \times I_y / I_3$$

$$= 2 \times V_x \times V_y / (R_x \times R_y \times I_3)$$

$$\therefore \quad Vo = 2 \times R_L \times V_x \times V_y / (R_x \times R_y \times I_3)$$

Referring to the Level Shifter diagram, C3, and using the values $I_2 = I_{14} = I_3 = 1mA$, Ro can easily be found as V_2 and V_{14} are half way between V_1 and the supply rail ie. 9 volts.

From above diagram $V_{14} / R_L + I_{14} = (V(+) - V_{14}) / Ro$

$\therefore Ro = 2.6k\,\Omega$

$\therefore$ use 3.3k Ω for R_0

Using a potentiometer in the R_2 load resistor section allows flexibility. The following values have been selected

$R_2 = 2.7k\,\Omega$

$R_{14} = 3.3k\,\Omega$

$R_L = 22k\,\Omega$

$R_{2L} = 12k\,\Omega + 10k\,\Omega$ potentiometer

This completes Appendix C, the description of the selection of component values for the RS 1495 Analogue Multiplier

APPENDIX D

D PROGRAM - CSMA/CD SIMULATOR

D.1 Summary

This appendix includes the code for the program written to simulate the throughput characteristics of the CSMA/CD protocol developed theoretically in section 6.2. The use of the program and the results obtained are described in section 6.3.

D.2 Brief Outline of the Program

The program simulates the arrival of packets into a node for transmission into the network. The average arrival time is governed by the rate of traffic being offered, **G**. The actual arrival time is subject to **Poisson** distribution characteristics around the average arrival time. (The POISSONRANDOM function used for this purpose is described in Appendix F). 50 packets are offered to the node at each of 16 different offered traffic rates. The first packet to arrive will find the network free and be assumed to be transmitted **successfully** so the channel **Use** time is set to **1**, the channel **Busy** time is set to **1+a** (ie. **1** plus the channel occupancy sensing time) and the **Idle** time is set to the **average arrival time** (see also section 6.2). The arrival time for the next packet is then calculated and the procedure CSMA is used to determine:-

- If the **arrival time** is less than the **channel sensing time**, ie. the **vulnerable period**, **a**, a collision is assumed and both packets are "discarded". The COLLISION procedure is then used to reset the **Busy** time to "**3** × **a**" (the sensing time) and the **Use** time to zero.

- If the arrival time is greater than the **vulnerable period** but less than the **Busy** time then the channel is already occupied by another packet and the present packet is classified as a "**CLASH**" and is discarded.

- If the arrival time is greater than the **Busy** time then the channel is assumed free again and the **Use**, **Busy** and **Idle** times set for a successful packet arrival.

When 50 packets have arrived the CSMA statistics are calculated and the simulated throughput, **ST**, is calculated from the average **Use**, average **Busy** and average **Idle** times as shown in equation (24).

For each offered traffic rate (**G**) the Theoretical Throughput (**TS**) is calculated from equation (31) and saved on file with the Simulated Throughput (**ST**), the number of **collisions**, the number of **successes** and the number of **clashes**, as shown in Table 5 in order that graphs may be plotted.

```
program Simcsma(input,output);

const
   packetbits = 1800;
   quit = 50;


type
   keeparr = array[1..50] of real;
   intarr = array[1..50] of integer;
var
   thrput : text;
   cyclecount, j,clash : integer;
   succtrans,col : integer;
   idle,busy,use,avebusy,aveidle,aveuse : real;
   mean,offered,aveariv,arivtime : real;
   totariv,totalbusy,totaluse,totalidle,a : real;
   theothr,simthr,garr : keeparr;
   cols,clashs : intarr;

procedure Collision;
begin
   col := col + 1;
   use := 0;
   busy := arivtime + (3 * a); {time channel occupied unsuccessfully}
end; {end of Collision}

procedure Csma;
begin
   if (arivtime > a) and (totariv < busy) then
   begin
      clash := clash + 1;
   end
   else
   begin {start of outer else}
      if arivtime <= a then
      Collision
      else
      begin
         totariv := 0;
         use := 1;
         succtrans := succtrans + 1;
         busy := 1 + a; {channel occupancy plus sensing time}
      end;
      begin
         totalbusy := totalbusy + busy;
         idle := aveariv;
         totalidle := totalidle + idle;
         totaluse := totaluse + use;
      end;
   end; {end of outer else}
end; {end of Csma}
```

```
procedure Csmastats;
begin
   if succtrans <> 0 then
   begin
      avebusy := totalbusy / succtrans;
      aveuse := totaluse / succtrans;
      aveidle := totalidle / succtrans;
      simthr[j] := aveuse / (avebusy + aveidle);
   end;
end; {end of Csmastats}

procedure Dispcond;
begin
   writeln('Bits Per Packet = ',packetbits);
   writeln('Detection time = ',a:6:4,' in units of packets');
   writeln;
end; {end of Dispcond}

procedure Outmod;
var
   i : integer;

begin
   Dispcond;
   writeln('Theoretical  Simulated   Offered   Collisions');
   writeln;
   for i := 1 to j do
   begin
      writeln(theothr[i]:8:4,simthr[i]:12:4,
              garr[i]:11:4,cols[i]:11,clashs[i]:11);
      writeln(thrput,theothr[i]:8:4,simthr[i]:12:4,
              garr[i]:11:4,cols[i]:11,clashs[i]:11);
   end;
end; {end of Outmod}

procedure Theocsma;
var
   x,y,g : real;

begin
   g := garr[j];
   x := g * exp(-a*g);
   y := (3*a*g + (exp(-a*g)*(2 + g - 3*a*g - exp(-a*g))));
   theothr[j] := x / y;
end; {end of Theocsma}


procedure Calcstats;
begin
   cols[j] := col;
   clashs[j] := clash;
   Csmastats;
   garr[j] := offered;
   Theocsma;
end; {end of Calcstats}
```

```
function PoissonRandom(mean: real): integer;
var
   product: real;
   n: integer;
begin { PoissonRandom }
   product := 1;
   n := 0;
   while product > exp(-mean) do
   begin
      product := product * random;
      n := n + 1
   end;
   PoissonRandom := n - 1
end;  { PoissonRandom }



begin
   a := (32 / packetbits);
   for j := 1 to 16 do
   begin
      randomize;
      case j of
         1  : offered := 0.1;
         2  : offered := 0.2;
         3  : offered := 0.4;
         4  : offered := 0.8;
         5  : offered := 1;
         6  : offered := 2;
         7  : offered := 5;
         8  : offered := 10;
         9  : offered := 15;
         10 : offered := 20;
         11 : offered := 50;
         12 : offered := 60;
         13 : offered := 70;
         14 : offered := 80;
         15 : offered := 80;
         16 : offered := 100;
      end; {end of case}
      arivtime := 0;
      clash := 0;
      totariv := 0;
      col := 0;
      use := 0;
      idle := 0;
      totalidle := 0;
      totaluse := 0;
      totalbusy := 0;
      cyclecount := 0;
      succtrans := 0;
      busy := 0;
      aveariv := 1 / offered;
      for cyclecount := 1 to quit do
      begin
         arivtime := (aveariv * PoissonRandom(20)/20);
         if arivtime <0 then writeln('negative arrival');
         totariv := totariv + arivtime;
         Csma;
      end;
      Calcstats;
   end; {end of for j}
   assign(thrput,'simres3.prn');
   rewrite(thrput);
   Outmod;
   close(thrput);
end.
```

APPENDIX E

E MICROPROCESSOR PROGRAM

E.1 Summary

This appendix contains the commented listing for the microprocessor control program.

```
                  $paginate , default , allpublic
                  $title(The ALS Spread Spectrum Program)
                  *          THIS IS THE ALS SPREAD SPECTRUM PROGRAM
                  *          FEATURES IMPLEMENTED INCLUDE :-
                  *          RECEIVER SCAN
                  *          AUTO RANGING OF 32 BITS FROM FIRST VALID BIT AT RX
                  *          TRANSMITTER PRE TX SEARCH
                  *
                  *
                  *          THIS IS MODIFIED FOR THROUGHPUT MEASUREMENTS
                  *          IT CAN ACT AS A RECEIVER COUNTING THE NUMBER OF ARRIVALS
                  *          AND GENERATING REAL TIME SIGNALS FOR THE PC
                  *          OR AS A TRAFFIC GENERATOR TAKING ITS MESSAGE ARRIVAL
                  *          TIMES AS POISSON DISTRIBUTED EVENTS, STORES THE TIME
                  *          IN THE T2 COUNTER AND AT THE EXPIRY OF THE COUNT ATTEMPTS
                  *          TO TRANSMIT THE MESSAGE.
                  *          OR AS AN INTERFERENCE GENERATOR WORKING TO GAUSSIAN
                  *          DISTRIBUTED EVENTS.
                  *          IN EITHER CASE TXS IS THE NUMBER OF ARRIVALS.
                  *
                  *
                  *          DEFINE VARIBLES TO BE USED
                  *
                             DEFSEG    VARIABLES   START=$700
                             SEG       VARIABLES
                  *
0000& (0001)      RXCTR      DS        1
0001& (0001)      LOCKRX     DS        1
0002& (0001)      TXSEL      DS        1
0003& (0001)      TXCTR      DS        1
0004& (0001)      RXMAXC     DS        1
```

```
0005& (0001)        32        RXMAXD     DS       1
0006& (0001)        33        TORXMT     DS       1
0007& (0001)        34        INVALI     DS       1
0008& (0001)        35        FIRCOL     DS       1
0009& (0001)        36        INUSE      DS       1
000A& (0002)        37        SAVTX      DS       2
000C& (0002)        38        SAVRX      DS       2
000E& (0001)        39        FRAME      DS       1
000F& (0001)        40        TOTXMT     DS       1
0010& (0001)        41        INPROG     DS       1
0011& (0001)        42        PREGON     DS       1
0012& (0001)        43        RXFIRS     DS       1
0013& (0001)        44        VALID1     DS       1
0014& (0001)        45        SCNCLR     DS       1
0015& (0001)        46        TXORRX     DS       1
0016& (0001)        47        TXENAB     DS       1
0017& (0001)        48        TOTMIT     DS       1
0018& (0001)        49        ALREAD     DS       1
0019& (0001)        50        HEXCNT     DS       1
001A& (0002)        51        TRFR       DS       2
001C& (0001)        52        READ       DS       1
001D& (0001)        53        RND        DS       1
001E& (0001)        54        FIRST      DS       1
001F& (0001)        55        OFLOW      DS       1
0020& (0001)        56        COLLS      DS       1
0021& (0001)        57        ABORT      DS       1
0022& (0002)        58        SRX        DS       2
0024& (0002)        59        GRX        DS       2
                    60        *
                    61        *
0026& (0002)        62        TXS        DS       2
```

```
Avocet 6809 Assembler v1.13, #00500   Chip=6809                                  11/10/87 15:50:13
The ALS Spread Spectrum Program                                                              Page 3


                       63        *
                       64        *
0028& (0020)           65        RXSAVE     DS        $20
0048& (0020)           66        TXSAVE     DS        $20
                       67        *
                       68        *
                       69        *          EQUATES
                       70        *
                       71        *
      =1FFF            72        SSTACK     EQU        $1FFF
      =17FF            73        USTACK     EQU        $17FF
      =0800            74        RXSTOR     EQU        $800
      =1000            75        TXSTOR     EQU        $1000
                       76        *
                       77        *
                       78                   DEFSEG    LOOKUP    START=$600
                       79                   SEG       LOOKUP
                       80        *
                       81        *
0000& 02 02 02 02      82        DVALID     DB        2,2,2,2,
0004& 02 02 02 02                2,2,2,2
0008& 02 02 02 02      83                   DB        2,2,2,2,
000C& 02 02 02 02                2,2,2,2
0010& 02 02 02 02      84                   DB        2,2,2,2,
0014& 02 02 02 02                2,2,2,2
0018& 02 02 02 02      85                   DB        2,2,2,2,
001C& 02 02 02 02                2,2,2,2
0020& 02 02 02 02      86                   DB        2,2,2,2,
0024& 02 02 02 02                2,2,2,2
0028& 02 02 02 02      87                   DB        2,2,2,2,
002C& 02 02 02 02                2,2,2,2
```

```
0030& 00 00 00 00    88                   DB       0,0,0,0,
0034& 00 00 00 00              0,0,0,0
0038& 00 00 00 00    89                   DB       0,0,0,0,
003C& 00 00 00 00              0,0,0,0
0040& 00 00 00 00    90                   DB       0,0,0,0,
0044& 00 00 00 00              0,0,0,0
0048& 00 00 00 00    91                   DB       0,0,0,0,
004C& 00 00 00 00              0,0,0,0
0050& 00 00 00 00    92                   DB       0,0,0,0,
0054& 00 00 00 00              0,0,0,0
0058& 00 00 00 00    93                   DB       0,0,0,0,
005C& 00 00 00 00              0,0,0,0
0060& 00 00 00 00    94                   DB       0,0,0,0,
0064& 00 00 00 00              0,0,0,0
0068& 00 00 00 00    95                   DB       0,0,0,0,
006C& 00 00 00 00              0,0,0,0
0070& 01 01 01 01    96                   DB       1,1,1,1,
0074& 01 01 01 01              1,1,1,1
0078& 01 01 01 01    97                   DB       1,1,1,1,
007C& 01 01 01 01              1,1,1,1
                     98        *
                     99        *
                    100                   DEFSEG    CONSTANTS   START=$680
                    101                   SEG       CONSTANTS
                    102        *
                    103        *
0000& 20            104        DESTAD     DB       $20            ;CHANGE ADDRESSES AS NECESSARY
0001& 60            105        LOCADD     DB       $60            ;$60 IS TRAFFIC GEN ADDRESS
                    106        *                                   $40 IS INTERFERENCE GEN ADDRESS
                    107        *                                   $20 IS RECEIVER ADDRESS
0002& 00 00 00      108        INHEX      DB       0,0,0
```

```
                    109       *
                    110       *
                    111               SEG       CODE ,    START=$100
                    112       *
                    113       *
0000' 86 0D         114       SETUP   LDA       #$0D
0002' B7 A00C       115               STA       $A00C
0005' 86 6B         116               LDA       #$6B
0007' B7 4002       117               STA       $4002
000A' B7 6002       118               STA       $6002
000D' 86 3E         119               LDA       #$3E
000F' B7 001D&      120               STA       RND          ;INITIALISE RANDOM BACKOFF TIME
0012' B7 4003       121               STA       $4003
0015' B7 6003       122               STA       $6003
0018' 86 05         123               LDA       #$05
001A' 1F 8B         124               TFR       A,DP
001C' 10 CE 1FFF    125               LDS       #SSTACK
0020' CE 17FF       126               LDU       #USTACK
0023' 86 FF         127               LDA       #$FF
0025' B7 0005&      128               STA       RXMAXD
0028' B7 001C&      129               STA       READ
002B' B7 A002       130               STA       $A002
002E' 86 01         131               LDA       #$01
0030' B7 0000&      132               STA       RXCTR
0033' 86 BF         133               LDA       #$BF
0035' B7 8002       134               STA       $8002
0038' 86 20         135               LDA       #$20         ;SET UP T2 COUNTER AT 1200 HZ
003A' B7 800B       136               STA       $800B
003D' 4F            137               CLRA
003E' B7 0016&      138               STA       TXENAB
0041' B7 001F&      139               STA       OFLOW
```

```
0044' B7 0001&      140                   STA     LOCKRX
0047' B7 0010&      141                   STA     INPROG
004A' B7 0009&      142                   STA     INUSE
004D' B7 0011&      143                   STA     PREGON
0050' B7 000F&      144                   STA     TOTXMT
0053' B7 000E&      145                   STA     FRAME
0056' B7 0026&      146                   STA     TXS
0059' B7 0027&      147                   STA     TXS+1
005C' B7 0019&      148                   STA     HEXCNT
005F' B7 0012&      149                   STA     RXFIRS
0062' B7 0008&      150                   STA     FIRCOL
0065' B7 0007&      151                   STA     INVALI
0068' B7 0020&      152                   STA     COLLS
006B' B7 0006&      153                   STA     TORXMT
006E' B7 0014&      154                   STA     SCNCLR
0071' B7 0013&      155                   STA     VALID1
0074' B7 0017&      156                   STA     TOTMIT
0077' B7 0018&      157                   STA     ALREAD
007A' B7 8003       158                   STA     $8003
007D' B7 A003       159                   STA     $A003
0080' 86 C1         160                   LDA     #$C1
0082' B7 C000       161                   STA     $C000
0085' 86 1E         162                   LDA     #$1E
0087' B7 C003       163                   STA     $C003
008A' 86 00         164                   LDA     #$00
008C' B7 C001       165                   STA     $C001
008F' 86 C0         166                   LDA     #$C0
0091' B7 C000       167                   STA     $C000
0094' 86 60         168                   LDA     #$60
0096' B7 C001       169                   STA     $C001
0099' 86 00         170                   LDA     #$00
```

```
009B' B7 C001                      STA     $C001
009E' B7 C000                      STA     $C000
00A1' B7 8000                      STA     $8000
00A4' B6 6000                      LDA     $6000      ;CLEAR RX REGISTER
                       *
                       *
00A7' 8E 0000&         START       LDX     #DVALID
00AA' B6 0001&                     LDA     LOCKRX     ;START OF OPERATIONAL PROGRAM
00AD' 27 1A                        BEQ     MAIN       ;ALREADY LOCKED ON RX?
00AF' B6 A001          HALTSE      LDA     $A001
00B2' 84 10                        ANDA    #$10       ;ONLY READ IF CLOCK HIGH
00B4' 27 F9                        BEQ     HALTSE
00B6' 17 02A2                      LBSR    SEARCH
00B9' E6 86                        LDB     A,X        ;CHECK IF DATA VALID
00BB' 26 05                        BNE     NOGOOD     ;IF OK THEN LOOK AT TX
00BD' F6 0000&         SETIT       LDB     RXCTR
00C0' 20 42                        BRA     OTHER
00C2' 86 01            NOGOOD      LDA     #$01       ;OTHERWISE DEAL WITH COLLISION
00C4' B7 0007&                     STA     INVALI     ;SET FLAG TO SAY COLLISION OCCURRED
00C7' 20 F4                        BRA     SETIT
00C9' 10 8E 0028&      MAIN        LDY     #RXSAVE    ;IF NOT LOCKED ON RX THEN...
00CD' B6 A001          WAITSE      LDA     $A001
00D0' 84 10                        ANDA    #$10       ;ONLY READ IF CLOCK HIGH
00D2' 27 F9                        BEQ     WAITSE
00D4' 17 0284                      LBSR    SEARCH
00D7' F6 0000&                     LDB     RXCTR      ;GET PRESENT COUNTER VALUE
00DA' A7 A5                        STA     B,Y        ;STORE DATA . POINTED TO BY COUNTER
00DC' F6 0012&                     LDB     RXFIRS     ;ARE WE IN FIRST SCAN LOOP?
00DF' 27 61                        BEQ     PASS1      ;IF SO GO TO FIRST LOOP ROUTINES
00E1' E6 86                        LDB     A,X        ;IS DATA VALID?
00E3' 26 0E                        BNE     NOWAY
```

```
00E5' B1 0005&      202                 CMPA    RXMAXD          ;IF SO IS IT LARGEST CORRELATION YET?
00E8' 24 09         203                 BCC     NOWAY
00EA' B7 0005&      204                 STA     RXMAXD          ;IF SO SAVE FOR REFERENCE
00ED' F6 0000&      205                 LDB     RXCTR           ;SAVE COUNTER VALUE ALSO
00F0' F7 0004&      206                 STB     RXMAXC
00F3' F6 0000&      207      NOWAY      LDB     RXCTR
00F6' F1 0013&      208                 CMPB    VALID1          ;32 SAMPLES SINCE FIRST VALID DATA?
00F9' 27 1B         209                 BEQ     DONE1           ;IF SO FINISHED SCAN
00FB' C1 0F         210                 CMPB    #$0F
00FD' 27 03         211                 BEQ     P2AVO           ;AVOID $10 DELAY BECAUSE OF JITTER
00FF' 5C            212                 INCB
0100' 20 02         213                 BRA     OTHER
0102' C6 11         214      P2AVO      LDB     #$11            ;STEP OVER $10 POSITION
0104' F7 0000&      215      OTHER      STB     RXCTR           ;SETUP NEXT RECEIVER SCAN POSITION
0107' B6 0017&      216      SETNRX     LDA     TOTMIT          ;READ FROM TX CORRELATOR?
010A' 27 02         217                 BEQ     NOTRMT          ;IF SOMETHING READ FROM TX
010C' CA 20         218                 ORB     #$20            ;SET UP TO READ FROM TX CORRELATOR
010E' FA 0016&      219      NOTRMT     ORB     TXENAB          ;ENABLE TRANSMITTER?
0111' F7 8000       220                 STB     $8000
0114' 20 6B         221                 BRA     LOOKTX
                    222      *
                    223      *
0116' C6 00         224      DONE1      LDB     #$00            ;PREPARE FOR PASS1 NEXT SEATCHX
0118' F7 0012&      225                 STB     RXFIRS
011B' F6 0004&      226                 LDB     RXMAXC          ;IF DONE LOCK ON RX
011E' F7 0000&      227                 STB     RXCTR           ;KEEP MAX CORRELATION POSITION
0121' F7 0001&      228                 STB     LOCKRX          ;AT MAX CORRELATION POSITION
0124' F7 0013&      229                 STB     VALID1
0127' 86 FF         230                 LDA     #$FF
0129' B7 0005&      231                 STA     RXMAXD          ;SET FOR NEXT SCAN COMPARISON
012C' 86 40         232                 LDA     #$40            ;CLEAR RX STATUS REG
```

```
012E' B7 C000      233                STA    $C000        ;TO START FRESH
0131' 86 00        234                LDA    #$00
0133' B7 C000      235                STA    $C000
0136' 86 20        236                LDA    #$20
0138' B7 C001      237                STA    $C001
013B' 86 00        238                LDA    #$00
013D' B7 C001      239                STA    $C001
0140' 20 C2        240                BRA    OTHER
                   241       *
                   242       *
0142' E6 86        243       PASS1    LDB    A,X          ;IF FIRST PASS THEN CHECK DATA
0144' 26 19        244                BNE    NVALRX
0146' F6 0013&     245                LDB    VALID1       ;IF DATA VALID IS IT FIRST?
0149' 26 06        246                BNE    NOTFIR
014B' F6 0000&     247                LDB    RXCTR        ;IF FIRST SETUP END OF PASS2
014E' F7 0013&     248                STB    VALID1
0151' B1 0005&     249       NOTFIR   CMPA   RXMAXD       ;IS IT LARGEST CORRELATION
0154' 24 09        250                BCC    NVALRX
0156' B7 0005&     251                STA    RXMAXD       ;IF SO THEN SAVE CORRELATION VALUE
0159' F6 0000&     252                LDB    RXCTR        ;AND ADDRESS OF MAX VALUE
015C' F7 0004&     253                STB    RXMAXC
015F' F6 0000&     254       NVALRX   LDB    RXCTR
0162' C1 1F        255                CMPB   #$1F         ;END OF FIRST PASS?
0164' 27 07        256                BEQ    ENDP1
0166' C1 0F        257                CMPB   #$0F         ;AVOID $10 DELAY
0168' 27 0F        258                BEQ    AVOID
016A' 5C           259                INCB
016B' 20 97        260                BRA    OTHER
016D' F6 0013&     261       ENDP1    LDB    VALID1       ;ANYTHING VALID DURING PASS1?
0170' 27 0B        262                BEQ    STILFI       ;IF NOT SETUP FIR CONTINUING PASS1
0172' C6 01        263                LDB    #$01
```

```
0174' F7 0012&       264                STB     RXFIRS        ;IF VALID SETUP FOR PASS2
0177' 20 8B          265                BRA     OTHER
0179' C6 11          266        AVOID   LDB     #$11          ;STEP OVER $10 DELAY
017B' 20 87          267                BRA     OTHER
017D' C6 01          268        STILFI  LDB     #$01          ;SETUP FOR PASS1 AGAIN
017F' 20 83          269                BRA     OTHER
                     270        *
                     271        *
0181' B6 0017&       272        LOOKTX  LDA     TOTMIT        ;IS THERE ANYTHING TO TRANSMIT?
0184' 10 27 0093     273                LBEQ    NOTHIN        ;IF NOT SETUP FOR NEXT RX
0188' 17 01D0        274                LBSR    SEARCH        ;GET VALUE FROM ADLC
018B' F6 0018&       275                LDB     ALREAD        ;ARE WE ALREDY TRANSMITTING?
018E' 27 4F          276                BEQ     TXSCAN        ;IF NOT CHECK FOR SCAN
0190' F6 0008&       277        INTX    LDB     FIRCOL        ;ARE WE JAMMING AFTER A COLLISION?
0193' 26 24          278                BNE     MORCOL        ;IF SO KEEP ON JAMMING
0195' E6 86          279                LDB     A,X           ;IS DATA VALID?
0197' C1 02          280                CMPB    #$02          ;LOOK FOR COLLISION
0199' 27 02          281                BEQ     TXCOLL        ;IF NOT THEN REACT TO COLLISION
019B' 20 73          282                BRA     KEEPON        ;IF VALID PREPARE FOR NEXT
019D' F6 0008&       283        TXCOLL  LDB     FIRCOL        ;IS IT FIRST COLLISION?
01A0' 26 17          284                BNE     MORCOL
01A2' C6 80          285                LDB     #$80
01A4' F7 C000        286                STB     $C000         ;RESET ADLC TRANSMITTER
01A7' 5F             287                CLRB
01A8' F7 C000        288                STB     $C000         ;SET OPERATIONAL AGAIN
01AB' F7 0010&       289                STB     INPROG
01AE' 8E 1000        290                LDX     #TXSTOR
01B1' BF 000A&       291                STX     SAVTX         ;RESET TX MESSAGE IF COLLISION
01B4' C6 23          292                LDB     #35           ;IF FIRST THEN SETUP JAM NUMBER
01B6' F7 0008&       293                STB     FIRCOL
01B9' 5A             294        MORCOL  DECB                  ;DECREMENT JAM COUNT
```

```
01BA' 27 05                    BEQ     TROFF      ;IF END OF JAM THEN TURN TX OFF
01BC' F7 0008&                 STB     FIRCOL
01BF' 20 4F                    BRA     KEEPON     ;IF NOT THEN PREPARE NEXT
01C1' F7 0008&         TROFF   STB     FIRCOL     ;CLEARDOWN ACTIVE TX FLAGS
01C4' F7 0014&                 STB     SCNCLR     ;CLEAR SCAN COUNT
01C7' F7 0018&                 STB     ALREAD     ;CLEAR ALREADY IN TX
01CA' F7 0016&                 STB     TXENAB     ;DISABLE TRANSMITTER
01CD' F7 0011&                 STB     PREGON     ;CLEAR FRAME PREAMBLE COUNT
01D0' C6 01                    LDB     #$01
01D2' F7 0003&                 STB     TXCTR
01D5' 20 3B                    BRA     SETNTX     ;SETUP FOR NEXT
                       *
                       *
01D7' B6 001D&         BUSY    LDA     RND        ;SET RANDOM BACKOFF COUNT
01DA' B7 0009&                 STA     INUSE      ;IF CHANNEL BUSY SET BACKOFF COUNT
01DD' 20 31                    BRA     KEEPON
                       *
                       *
01DF' F6 0009&         TXSCAN  LDB     INUSE
01E2' 26 2C                    BNE     KEEPON     ;BACKOFF IF CHANNEL BUSY
01E4' E6 86                    LDB     A,X        ;IS THERE VALID DATA?
01E6' 27 EF                    BEQ     BUSY       ;IE. IS CHANNEL BUSY?
01E8' F6 0003&                 LDB     TXCTR
01EB' C1 0F                    CMPB    #$0F       ;IF NOT BUSY AVOID $10 DELAY
01ED' 27 07                    BEQ     TXAVO
01EF' C1 1F                    CMPB    #$1F       ;HAVE WE COMPLETED PRE TX SEARCH?
01F1' 27 07                    BEQ     DONETX
01F3' 5C                       INCB               ;IF NOT LOOK AT NEXT POSITION
01F4' 20 1C                    BRA     SETNTX
01F6' C6 11            TXAVO   LDB     #$11       ;STEP OVER DELAY $10
01F8' 20 18                    BRA     SETNTX
```

```
Avocet 6809 Assembler v1.13, #00500   Chip=6809                                  11/10/87 15:50:13
The ALS Spread Spectrum Program                                                             Page 12


01FA' F7 0014&      326       DONETX    STB     SCNCLR        ;NOTE SCAN COMPLETE
01FD' C6 41         327                 LDB     #$41          ;SET PREAMBLE COUNT
01FF' F7 0011&      328                 STB     PREGON
0202' F7 0018&      329                 STB     ALREAD        ;SET TO CHECK OWN TX NEXT TIME
0205' C6 80         330                 LDB     #$80          ;SET TRANSMITTER ENABLE
0207' F7 0016&      331                 STB     TXENAB        ;SCAN OK SO ENABLE TX
020A' F6 0000&      332                 LDB     DESTAD        ;GET DESTINATION ADDRESS
020D' F7 0002&      333                 STB     TXSEL         ;AND COLLECT ADDRESS CODE
0210' C6 04         334       KEEPON    LDB     #$04          ;SETUP FOR 0.5 BIT DELAYED TX
0212' F7 0003&      335       SETNTX    STB     TXCTR         ;NEXT TX POSITION TO LOOK AT
0215' FA 0002&      336                 ORB     TXSEL         ;SELECT TX ADDRESS CODE
0218' F7 A000       337                 STB     $A000
021B' F6 0000&      338       NOTHIN    LDB     RXCTR         ;SETUP FOR NEXT RX
021E' FA 0016&      339                 ORB     TXENAB        ;ENABLE TX IF NECESSARY
0221' C4 DF         340                 ANDB    #$DF          ;POINT ADC TO RECEIVER
0223' F7 8000       341                 STB     $8000
                    342       *
                    343       *
                    344       *                               ANY MESSAGE TO TRNSMIT
0226' B6 000F&      345       MOREGO    LDA     TOTXMT        ;ANY MORE PACKETS?
0229' 26 03         346                 BNE     EMPTY
022B' 7F 0017&      347                 CLR     TOTMIT
022E' 20 38         348       EMPTY     BRA     NONEED        ;IF NOT CHECK KEYBOARD
0230' B6 0009&      349       GOWITH    LDA     INUSE         ;IS CHANNEL BUSY
0233' 26 2C         350                 BNE     BACKOF
0235' B6 0010&      351                 LDA     INPROG        ;ALREADY IN PROGRESS?
0238' 10 26 00BE    352                 LBNE    ADLCTX        ;IF IN PROGRESS CHECH ADLC TX
023C' B6 0014&      353                 LDA     SCNCLR        ;HAVE WE COMPLEATED PRE TX SCAN?
023F' 26 03         354                 BNE     ENDSCN
0241' 16 0145       355                 LBRA    INADLC        ;IF NOT THEN CHECK INPUT FROM ADLC
0244' B6 0011&      356       ENDSCN    LDA     PREGON        ;FINISHED DATA PREAMBLE?
```

```
0247' 27 12          357                BEQ       ENABTX
0249' 4A             358                DECA                        ;IF NOT DECREMENT PREAMBLE COUNT
024A' B7 0011&       359                STA       PREGON
024D' 8E 1000        360                LDX       #TXSTOR
0250' B6 0001&       361                LDA       LOCADD
0253' A7 84          362                STA       ,X
0255' BF 000A&       363                STX       SAVTX
0258' 16 012E        364                LBRA      INADLC            ;AND CHECK RX ADLC
025B' 7C 0010&       365     ENABTX     INC       INPROG            ;SET IN PROGRESS FLAG
025E' 16 0128        366                LBRA      INADLC
                     367     *
                     368     *
0261' 4A            369     BACKOF     DECA
0262' B7 0009&       370                STA       INUSE             ;DECREMENT BACKOFF COUNT
0265' 16 0121        371                LBRA      INADLC
                     372     *
                     373     *
                     374     *        Special note
                     375     *        This section needs to be modified to transmit rather than
                     376     *        receiving timing signals (ie. from the PC which generates
                     377     *        the Poisson or Gaussian statistics.
                     378     *        The receiver generates real-time signals
                     379     *        and the two transmitter options use the PC produced outputs
                     380     *
0268' B6 4001        381     NONEED     LDA       $4001
026B' 84 08          382                ANDA      #$08              ;CHECK FOR CHAR FROM KEYBOARD
026D' 10 27 000A     383                LBEQ      PCCHK             ;IF NOT CHECK T2 TIMER ETC.
0271' B6 4000        384                LDA       $4000             ;IF YES THEN GET CHARACTER
0274' 84 7F          385                ANDA      #$7F              ;STRIP PARITY
0276' 81 51          386                CMPA      #$51              ;SEE IF QUIT CHAR
0278' 26 01          387                BNE       PCCHK
```

```
027A' 3F            388               SWI                      ;IF SO THEN HALT
027B' B6 6001       389      PCCHK    LDA      $6001
027E' 84 08         390               ANDA     #$08
0280' 10 27 0047    391               LBEQ     SENTFR
0284' B6 6000       392               LDA      $6000
0287' 8E 0002&      393               LDX      #INHEX
028A' F6 0019&      394               LDB      HEXCNT
028D' A7 85         395               STA      B,X             ;SAVE COUNT VALUES FROM PC
028F' C1 03         396               CMPB     #$03
0291' 27 07         397               BEQ      GETHEX
0293' 5C            398               INCB
0294' F7 0019&      399               STB      HEXCNT
0297' 16 0031       400               LBRA     SENTFR
029A' AD 9F FFDA    401      GETHEX   JSR      [$FFDA]         ;TRANSLATE STRING TO HEX
029E' 10 BF 001A&   402               STY      TRFR            ;PUT RESULT IN TRANSFER LOCATION
02A2' 7F 0019&      403               CLR      HEXCNT
02A5' B6 000F&      404               LDA      TOTXMT
02A8' 27 0A         405               BEQ      NEWTME          ;IF WE HAVEN'T SENT LAST
02AA' BE 0026&      406               LDX      TXS             ;MESSAGE IGNORE T2 AND JUST STACK
02AD' 30 01         407               LEAX     1,X             ;FOR TRANSMISSION ASAP
02AF' BF 0026&      408               STX      TXS
02B2' 20 30         409               BRA      TIMOUT
02B4' B6 001B&      410      NEWTME   LDA      TRFR+1
02B7' B7 8008       411               STA      $8008
02BA' B6 001A&      412               LDA      TRFR
02BD' B7 8009       413               STA      $8009           ;SET T2 COUNTER FOR MESSAGE ARRIVAL
02C0' BE 0026&      414               LDX      TXS             ;ADD 1 TO MESSAGES TO BE TRANSMITTED
02C3' 30 01         415               LEAX     1,X
02C5' BF 0026&      416               STX      TXS
02C8' 7F 001C&      417               CLR      READ
02CB' B6 001C&      418      SENTFR   LDA      READ
```

```
02CE' 10 26 FF5E    419                LBNE     GOWITH       ;IF ALREADY USED WAIT TO BE RESET
02D2' FC 8008       420                LDD      $8008
02D5' 10 83 0000    421                CMPD     #$0000       ;LOOK FOR T2 TIMEOUT
02D9' 27 09         422                BEQ      TIMOUT
02DB' 10 83 FFFF    423                CMPD     #$FFFF       ;ALTERNATIVE TIMEOUT VALUE
02DF' 27 03         424                BEQ      TIMOUT
02E1' 16 FF4C       425                LBRA     GOWITH       ;IF NOTHING THEN CONTINUE
02E4' B6 000F&      426       TIMOUT   LDA      TOTXMT       ;SEND ONE MORE MESSAGE
02E7' 4C            427                INCA
02E8' B7 001C&      428                STA      READ
02EB' B7 000F&      429                STA      TOTXMT
02EE' B7 0017&      430                STA      TOTMIT       ;SET FLAG SOMETHING TO GO
02F1' B6 0000&      431                LDA      DESTAD       ;GET DESTINATION ADDRESS
02F4' B7 0002&      432                STA      TXSEL        ;SET DEST ADDRESS CODE
02F7' 16 FF36       433                LBRA     GOWITH
                    434       *
                    435       *
02FA' B6 C000       436       ADLCTX   LDA      $C000
02FD' 84 40         437                ANDA     #$40         ;IS TX REGISTER EMPTY?
02FF' 26 0A         438                BNE      TXFRA        ;IF SO THEN CHECK RX
0301' B6 C000       439                LDA      $C000
0304' 84 20         440                ANDA     #$20
0306' 26 4B         441                BNE      UNDRUN       ;CHECK FOR TRANSMITTER UNDERRUN
0308' 16 007E       442                LBRA     INADLC       ;IF TX NOT EMPTY OR UNDERRUN
030B' B6 000E&      443       TXFRA    LDA      FRAME        ;IF EMPTY HAVE WE MORE TO TX
030E' 27 15         444                BEQ      MORTX
0310' 4F            445                CLRA                  ;IF END OF FRAME CLEAR TX FLAGS
0311' B7 0010&      446                STA      INPROG
0314' B7 0014&      447                STA      SCNCLR
0317' B7 0016&      448                STA      TXENAB
031A' B7 0018&      449                STA      ALREAD
```

```
0320' B7 C001     451              STA     $C001      ;SET TX FLAG TO TDRE
0323' 20 64       452              BRA     INADLC
0325' BE 000A&    453    MORTX     LDX     SAVTX      ;COLLECT DATA POINTER
0328' 8C 10C7     454              CMPX    #$10C7     ;TX ALL MESSAGE OF 100 BYTES?
032B' 27 0A       455              BEQ     LASTGO     ;IF SO SEND LAST DATA BYTE
032D' A6 80       456              LDA     ,X+        ;IF NOT GET NEXT BYTE & INC POINTER
032F' B7 C002     457              STA     $C002      ;STORE IN FRAME CONTINUE ADDRESS
0332' BF 000A&    458              STX     SAVTX
0335' 20 52       459              BRA     INADLC     ;CHECK RX NEXT
0337' A6 84       460    LASTGO    LDA     ,X         ;GET LAST BYTE
0339' B7 C003     461              STA     $C003      ;STORE IN FRAME TERMINATE ADDRESS
033C' 8E 1000     462              LDX     #TXSTOR    ;POINT AT BEGINNING OF MESSAGE
033F' BF 000A&    463              STX     SAVTX      ;SAVE POINTER
0342' B6 000F&    464              LDA     TOTXMT     ;MORE MESSAGES TO GO?
0345' 4A          465              DECA               ;DECREMENT NO OF MESSAGES
0346' B7 000F&    466              STA     TOTXMT
0349' 86 08       467              LDA     #$08       ;SET TX FLAG TO FRAME END
034B' B7 C001     468              STA     $C001
034E' B7 000E&    469              STA     FRAME      ;SET TO CHECK FOR FRAME END
0351' 20 36       470              BRA     INADLC     ;CHECK RX ADLC NEXT
                  471    *
0353' 86 35       472    UNDRUN    LDA     #$35       ;DEAL WITH UNDERRUN
0355' B7 0008&    473              STA     FIRCOL     ;AS A COLLISION
0358' 16 FE42     474              LBRA    TXCOLL
                  475    *
035B' 86 0F       476    SEARCH    LDA     #$0F
035D' C6 0D       477              LDB     #$0D
035F' B7 A00C     478              STA     $A00C      ;SEND START CONVERSION PULSE
0362' F7 A00C     479              STB     $A00C
0365' 86 50       480              LDA     #80
```

```
0367' B7 A006                STA     $A006        ;SET TIMER TO ENSURE RESTART
036A' 86 00                  LDA     #$00         ;IF ADC FAILS TO RESPOND
036C' B7 A005                STA     $A005
036F' B6 A00D       WAITH    LDA     $A00D        ;WAIT FOR END OF CONVERSION
0372' 84 02                  ANDA    #$02
0374' 26 07                  BNE     GOTIT
0376' B6 A004                LDA     $A004
0379' 27 E0                  BEQ     SEARCH
037B' 20 F2                  BRA     WAITH
037D' B6 A001       GOTIT    LDA     $A001        ;CLEAR EOC RESPONSE
0380' B6 8001                LDA     $8001        ;GET DATA FROM ADC
0383' 2A 01                  BPL     PLUS
0385' 43                     COMA                 ;GET MAGNITUDE VALUE OF DATA
0386' 84 7F         PLUS     ANDA    #$7F         ;REMOVE SIGN BIT
0388' 39                     RTS                  ;RETURN TO CALLING ROUTINE
                    *
                    *
0389' B6 0001&      INADLC   LDA     LOCKRX       ;ARE WE LOCKED ONTO RECEIVING
038C' 27 02                  BEQ     WAITL
038E' 20 0D                  BRA     SMTHIN
0390' 17 00A0       WAITL    LBSR    RANDOM
0393' B6 A001                LDA     $A001        ;WAIT TILL CLOCK LOW BEFORE RETURNING
0396' 84 10                  ANDA    #$10
0398' 26 F6                  BNE     WAITL        ;MIGHT NOT NEED LATER
039A' 16 FD0A                LBRA    START
039D' B6 C000       SMTHIN   LDA     $C000        ;ANYTHING TO LOOK AT?
03A0' 84 02                  ANDA    #$02
03A2' 27 EC                  BEQ     WAITL        ;IF NOT THEN START AGAIN
03A4' B6 C001                LDA     $C001        ;IF SO THEN WHAT?
03A7' 84 02                  ANDA    #$02         ;FRAME VALID?
03A9' 26 1E                  BNE     ENDFR
```

```
03AB' B6 C001       512                 LDA      $C001
03AE' 84 01         513                 ANDA     #$01         ;NEW FRAME - ADDRESS PRESENT?
03B0' 10 26 0055    514                 LBNE     NEWFR
03B4' B6 C001       515                 LDA      $C001
03B7' 84 80         516                 ANDA     #$80         ;DATA BYTE?
03B9' 10 26 005A    517                 LBNE     GOTBYT
03BD' B6 C001       518                 LDA      $C001
03C0' 84 08         519                 ANDA     #$08         ;LOOK FOR ABORTED TRANSMISSIONS
03C2' 27 CC         520                 BEQ      WAITL        ;IF NOTHING THEN JUST WAIT
03C4' 7C 0021&      521                 INC      ABORT        ;OTHERWISE INCREMENT ABORT COUNTER
03C7' 20 25         522                 BRA      RESRX        ;AND RESET RECEIVER
                    523        *
                    524        *
03C9' B6 0007&      525        ENDFR    LDA      INVALI       ;ANY PROBLEMS DURING RECEPTION?
03CC' 27 07         526                 BEQ      GOODFR
03CE' B6 0020&      527                 LDA      COLLS        ;IF SO THEN INC COLLISION COUNTER
03D1' 4C            528                 INCA
03D2' B7 0020&      529                 STA      COLLS
03D5' 8E 0800       530        GOODFR   LDX      #RXSTOR
03D8' A6 84         531                 LDA      ,X           ;GET SENDER'S ADDRESS
03DA' 81 40         532                 CMPA     #$40
03DC' 27 08         533                 BEQ      INTFME       ;FROM INTERFERENCE GENERATOR?
03DE' BE 0022&      534                 LDX      SRX
03E1' 30 01         535                 LEAX     1,X          ;IF NOT THEN INC THROUGHPUT COUNT
03E3' BF 0022&      536                 STX      SRX
03E6' BE 0024&      537        INTFME   LDX      GRX          ;INCREMENT TRAFFIG GENERATOR COUNT
03E9' 30 01         538                 LEAX     1,X
03EB' BF 0024&      539                 STX      GRX
03EE' 86 20         540        RESRX    LDA      #$20         ;THEN RESET RECEIVER STATUS
03F0' B7 C001       541                 STA      $C001
03F3' 86 00         542                 LDA      #$00
```

```
03F5' B7 C001      543               STA     $C001
03F8' B7 0001&     544               STA     LOCKRX      ;CLEAR RECEIVER FLAGS
03FB' B7 0013&     545               STA     VALID1
03FE' B7 0007&     546               STA     INVALI
0401' 86 01        547               LDA     #$01
0403' B7 0000&     548               STA     RXCTR       ;RESET RX COUNTER
0406' 16 FF87      549               LBRA    WAITL       ;AND START AGAIN
                   550     *
                   551     *
0409' 8E 0800      552     NEWFR     LDX     #RXSTOR
040C' B6 C002      553               LDA     $C002       ;LOAD FROM ADDRESS FIELD
040F' A7 80        554               STA     ,X+         ;SAVE ADDRESS VALUE
0411' BF 000C&     555               STX     SAVRX       ;SAVE POINTER
0414' 16 FF79      556               LBRA    WAITL
                   557     *
                   558     *
0417' BE 000C&     559     GOTBYT    LDX     SAVRX       ;GET POINTER
041A' B6 C002      560               LDA     $C002       ;GET DATA VALUE
041D' A7 80        561               STA     ,X+         ;SAVE DATA BYTE FROM ADLC
041F' 8C 08D0      562               CMPX    #$08D0      ;MORE BYTES THAN SENT?
0422' 27 06        563               BEQ     BADEND
0424' BF 000C&     564               STX     SAVRX
0427' 16 FF66      565               LBRA    WAITL       ;IF OK THEN DO IT AGAIN
                   566     *
                   567     *
042A' B6 001F&     568     BADEND    LDA     OFLOW       ;MESSAGE TOO LONG
042D' 4C           569               INCA
042E' B7 001F&     570               STA     OFLOW       ;COUNT OVERLONG MESSAGES
0431' 20 A2        571               BRA     GOODFR
                   572     *
                   573     *
```

```
                    574     *
0433' B6 001D&      575     RANDOM   LDA      RND
0436' 84 01         576              ANDA     #$01        ;CHECK BIT 8
0438' 27 1B         577              BEQ      SETONE
043A' 4F            578              CLRA
043B' B7 001E&      579              STA      FIRST
043E' B6 001D&      580     NEXT     LDA      RND
0441' 84 02         581              ANDA     #$02        ;CHECK BIT 7
0443' 27 17         582              BEQ      SETTWO
0445' 4F            583              CLRA
0446' B8 001E&      584     ANTHER   EORA     FIRST
0449' 27 04         585              BEQ      NOCARR
044B' 1A 01         586              ORCC     #$01
044D' 20 02         587              BRA      FIXIT
044F' 1C FE         588     NOCARR   ANDCC    #$FE
0451' 76 001D&      589     FIXIT    ROR      RND         ;ROTATE THROUGH CARRY
0454' 39            590              RTS
                    591     *
0455' 86 01         592     SETONE   LDA      #$01
0457' B7 001E&      593              STA      FIRST
045A' 20 E2         594              BRA      NEXT
045C' 86 01         595     SETTWO   LDA      #$01
045E' 20 E6         596              BRA      ANTHER
                    597     *
                    598              END
```

APPENDIX F

F PROGRAM - TRAFFIC GENERATOR

F.1 Summary

This appendix describes the random number generator which was written to produce the traffic arrival characteristics required for System Test purposes (see chapter 7). The standard "**random**" function; available in most high level programming languages, produces numbers with a uniform distribution. The need to produce simulated packet arrivals subject to **Poisson** and/or **Normal** (**Gaussian**) distribution characteristics, however, meant that the standard function could not be used directly, but had to be modified to produce the required distribution.

F.2 Poisson & Normal Distribution Characteristics

F.2.1 Normal (Gaussian)

One of the most important examples of a continuous probability is the Normal, or Gaussian, distribution. This distribution is represented, in standard form, by:-

$$Y = \frac{1}{\sqrt{2\pi}} e^{-(1/2)(z \times z)} \qquad \text{(L1)}$$

where $z = (x - \mu) / \lambda$ and x is in standard units.

In such a case we can say the **z** is normally distributed with a **mean** of **zero** and a **variance** of **one**.

The major characteristics of the Gaussian distribution are:-

- Mean = μ
- Variance = σ^2
- Standard Deviation = σ

F.2.2 Poisson

The Poisson distribution has been widely used to model arrival distributions and other seemingly random events. The distribution is named after Poisson, who discovered it in the early part of the 19th century. The discrete probability distribution is given by:

$$p(x) = \frac{\lambda^x e^{-\lambda}}{x!} \qquad \text{(L2)}$$

where e = 2.71828 and λ is a given constant.

p (x) is then called the Poisson distribution.

The major characteristics of the Poisson distribution are:-

- Mean $= \mu = \lambda$

- Variance $= \sigma^2 = \lambda$

- Standard Deviation $= \sigma = \sqrt{\lambda}$

F 3 The "Random" Programs

Many texts have been written on the applications of stochastic processes in simulation and modelling. One such text [31] describes standard procedures which will produce either Poisson or Gaussian distribution characteristics from the uniformly distributed random numbers produced as a standard language function.

F.3.1 Normal Distribution Numbers

The theorem for the production of numbers with a Gausssian distribution is called the "**Central Limit Theorem**". It states that the sum, x_n, of **n** identically distributed independent variables has an approximately normal distribution with a mean $= \mu$ **n** and a variance of $n\sigma^2$. Studies have shown that with **n = 12** this technique provides good results.

If the variables summed follow a standard uniform distribution then $\mu =$ **0.5** and $\sigma^2 =$ **1/12.** Thus summing **n** standard uniform random numbers gives an approximate Gaussian distribution with a mean of **0.5 n** and a variance of **n/12.** By dividing the sum by **6** instead of **12** a mean value of 1 rather than 0.5 can be obtained. A multiplying factor may then be applied to produce random numbers in a suitable range.

A procedure called **NORMALRANDOM** was written to produce random numbers with the characteristics of Gaussian distribution. This routine was tested to determine the characteristics of the distribution produced, see listing at the end of this appendix. The largest value of **RANGE** selectable in the program is 2730 , this is in order to avoid Integer Overflow. This can be extended to 10000, however, if a Maths co-processor is used. 10,000 samples were obtained from the procedure and the number of times each value was produced was plotted against the value. Figure **F1** is a plot of the distribution obtained, with a mean value of 50. It can be seen that this distribution does, indeed, approximate to Gaussian characteristics.

F.3.2 Poisson Number Program

Random numbers with a Poisson distribution with a mean of λ may be accomplished by mulitplying, **n**, successively generated numbers until the product is less than $\mathbf{e}^{-\lambda}$. The value **n** - 1 is then the value of the Poisson random variable **x**.

A maximum value of 88 must be placed on the value of **MEAN** used with this procedure, in order to avoid Floating Point Overflow. This value can be extended to 100 if a Maths co-processor is used. Figure F2 is a plot of the distribution produced by the procedure **POISSONRANDOM**, which is part of the Traffic program included in this appendix. 10,000 samples with a mean value of 50 were used to produce this distribution curve. It can be seen that the error distribution obtained from this procedure approximates to Poisson characteristics.

F.4 The Traffic Program

The mean valued "**TIME**" is initialised at the start of the program (see section 7.6). This program accepts an input from node **#1** of the Spread Spectrum LAN, in order to synchronise to real world timing. Two random numbers with the preset "**TIME**" value are then produced. One of the numbers is subjected to

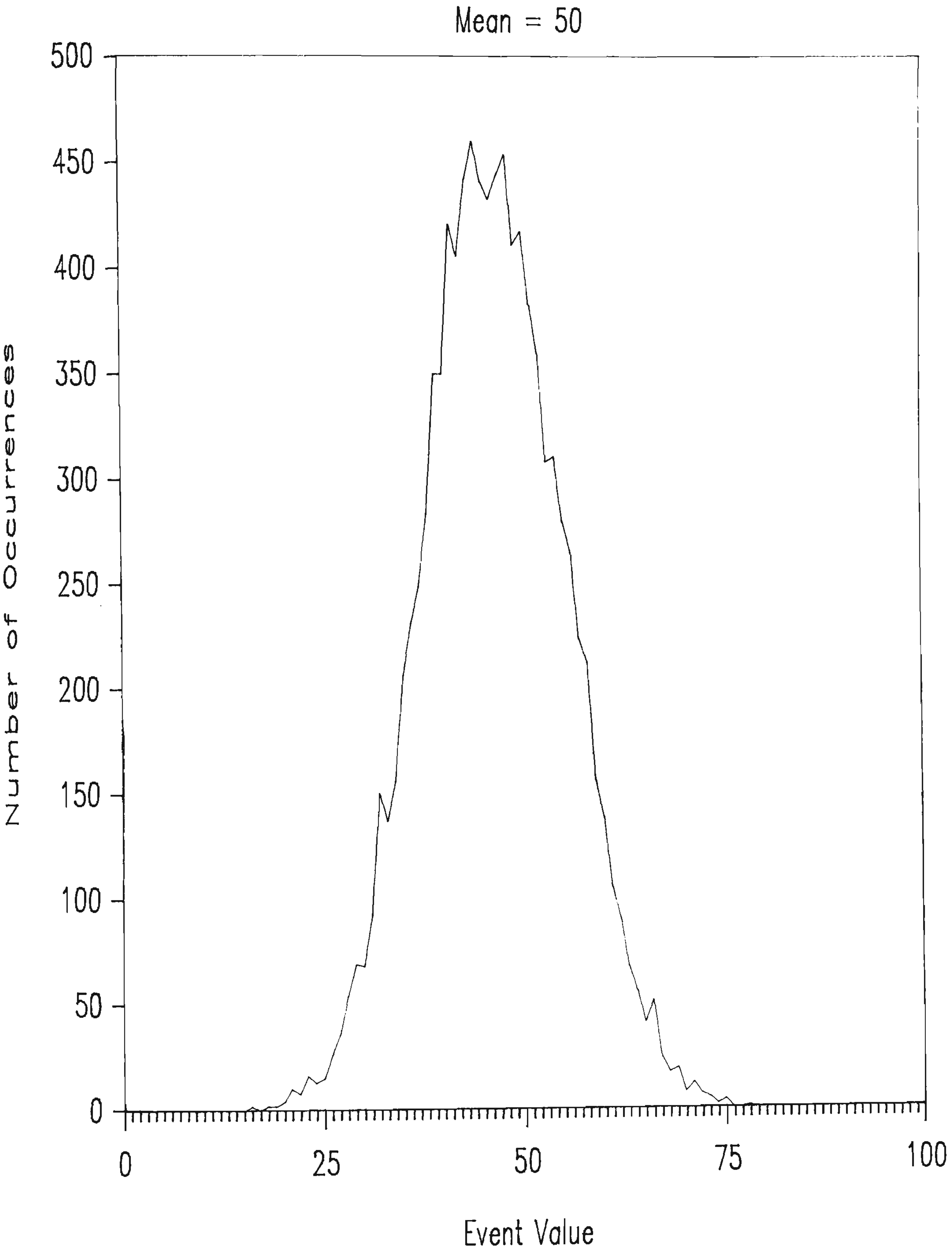


Figure F1 Probability Distribution Normal (Gaussian) Characteristics

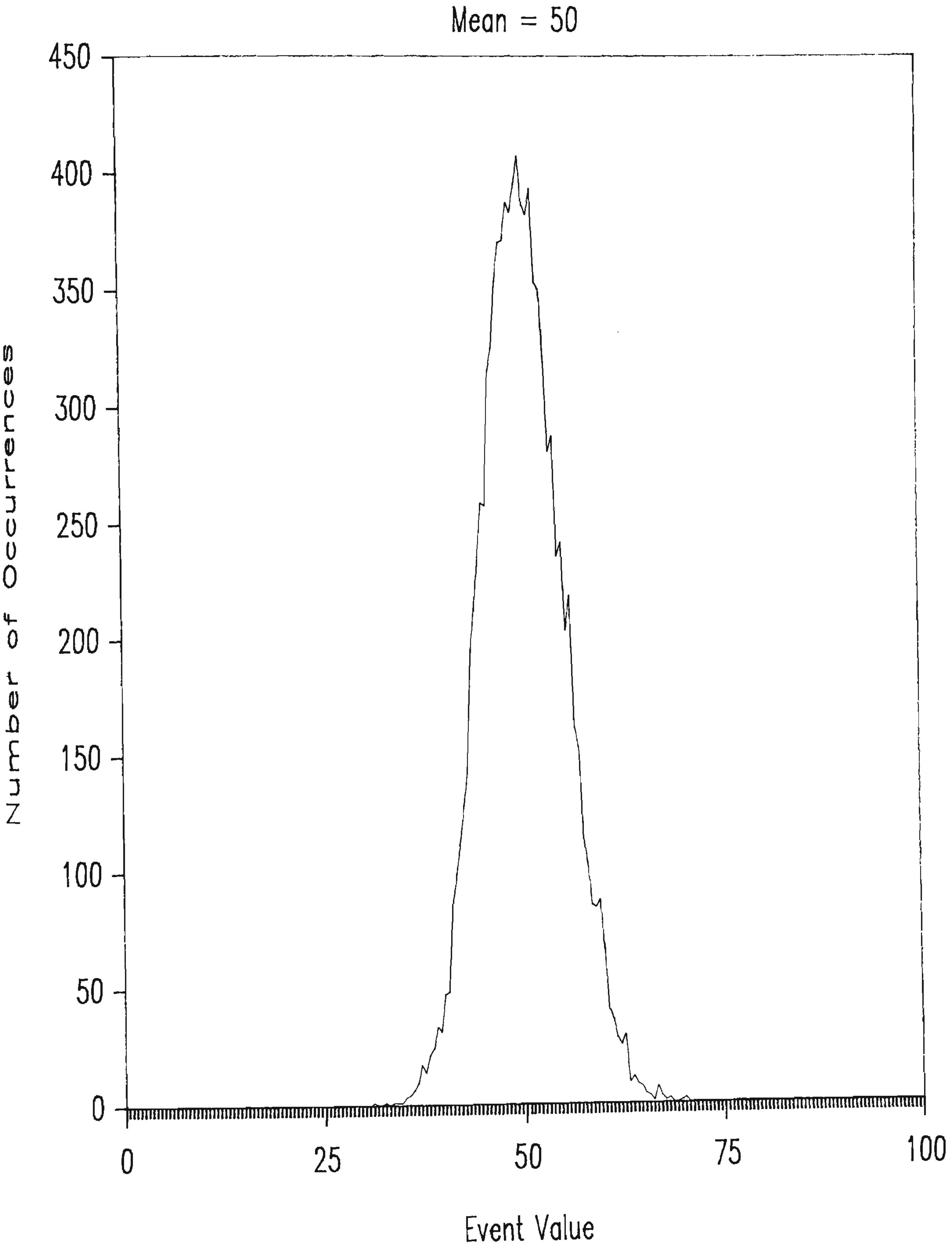


Figure F2 Probability Distribution Poisson Characteristics

Gaussian distribution and the other to Poisson characteristics from the two procedures described above. These numbers are converted to Hexadecimal ASCII characters. The Poisson number is transmitted to node **#3**, the traffic generator, and the Gaussian number is transmitted to node **#2**, the interference generator. These numbers are then used to represent packet arrival times. By altering the standard inter-arrival delay produced by node **#1**, and adjusting the program's mean value accordingly, offered traffic with Poisson and Gaussian distributions can be simulated.

```
program Traffic (input,output);
{$C-}

type
   hexstr = array[1..4] of char;

var
   time,count,temp,j,i,c,d : integer;
   b,a : real;
   intval,trafval : hexstr;
   hold : byte;


function Selval(temp : integer) :char;
begin
   case temp of
      0  : Selval := '0';
      1  : Selval := '1';
      2  : Selval := '2';
      3  : Selval := '3';
      4  : Selval := '4';
      5  : Selval := '5';
      6  : Selval := '6';
      7  : Selval := '7';
      8  : Selval := '8';
      9  : Selval := '9';
      10 : Selval := 'A';
      11 : Selval := 'B';
      12 : Selval := 'C';
      13 : Selval := 'D';
      14 : Selval := 'E';
      15 : Selval := 'F';
   end; {end of case}
end; {end of Selval}


procedure Hexout(var hexval :hexstr ; num : integer);
begin
   temp := hi(num) div 16 and $F;
   hexval[1] := Selval(temp);
   temp := hi(num) and $F;
   hexval[2] := Selval(temp);
   temp := lo(num) div 16 and $F;
   hexval[3] := Selval(temp);
   temp := lo(num) and $F;
   hexval[4] := Selval(temp);
end; {end of Hexout}
```

```
function NormalRandom(range: integer): real;
{ Generate normal distribution of random integers between 0 and range }
var
   i: integer;
   sum: real;

begin { NormalRandom }
   sum := 0;
   for i := 1 to 12 do
   sum := sum + random(range)/range;
   NormalRandom := (sum / 6)
end;  { NormalRandom }


function PoissonRandom(mean: integer): integer;
{ Generate a Poisson distribution of random integers }
var
   product: real;
   n: integer;
   res : real;

begin { PoissonRandom }
   product := 1;
   n := 0;
   res := exp(-mean);
   while product > res  do
   begin
      product := product * random;
      n := n + 1
   end;
   PoissonRandom := n - 1
end;  { PoissonRandom }


procedure Send;
begin
   repeat
   until ((port[$3fd] and $01) > 0);  {wait for prompt from rx board}
   hold := port[$3f8];                {clear flag from rx board}
   for j := 1 to 4 do  {send hex arrival times to both transmitters}
   begin
      repeat
      until ((port[$3fd] and $20) > 0);
      hold := ord(trafval[j]);    {send to traffic gen}
      port[$3f8] := hold;
      repeat
      until ((port[$2fd] and $20) > 0);
      port[$2f8] := ord(intval[j]);    {send to interference gen}
   end;
end;  {end of Send}
```

```
begin { Traffic Generator }
   count := 0;
   port[$3fb] := $9A;   {set up serial port 1}
   port[$3f8] := $0C;   {to go to traffic generator}
   port[$3f9] := $00;
   port[$3fb] := $1A;
   port[$2fb] := $9A;   {set up serial port 2}
   port[$2f8] := $0C;   {to go to interference generator}
   port[$2f9] := $00;
   port[$2fb] := $1A;
   repeat
   begin
      write('Enter mean value for this run  ..  ');
      readln(time);
      randomize;
      a := (PoissonRandom(100))/100;
      b := NormalRandom(10000);
      c := trunc(a*time);
      d := trunc(b*time);
      Hexout(trafval,c);
      Hexout(intval,d);
      Send;
      count := count + 1;
      write(count,'     ');
      for j := 1 to 4 do
      write(trafval[j]);
      write('     ');
      for j := 1 to 4 do
      write(intval[j]);
      writeln;
   end;
   until (keypressed or (count = 10000));
 end.  { Traffic Generator }
```

APPENDIX G

G PUBLICATIONS

G.1 Summary

This appendix contains copies of all of the conference and journal papers which were written as a direct result of the work carried out on this research project.

UDC 621.394.4
Indexing Terms: Data transmission, Local area networks, Spread spectrum

A synchronized spread-spectrum local area network

A. L. SIMCOCK, BSc,*
C. na RANONG, BE, PhD*
and
P. LEUNG, BE, MEngSc*

* *Department of Electrical and Electronic Engineering, Footscray Institute of Technology, Footscray, Australia 3011*

SUMMARY

As the number of nodes connected to a local area network increases, the network throughput decreases owing to channel contention. This is conventionally overcome by either upgrading the network channel bandwidth, and hence increasing the data rate, or introducing a hierarchical network structure. This paper describes an adaptation of spread-spectrum communication techniques, used by the military, to a localized network. The resulting system enables multiple simultaneous communications within a channel rather than the conventional single communication. Such a system is capable of retaining communications facilities in a heavily loaded multi-user environment.

1 Introduction

The expanding need for data communication channels has increased the pressure on the available transmission bandwidth. This has led to developments in two broad areas: the extension of the usable spectrum through the use of even higher frequency carriers and the development of complex frequency allocation algorithms. An extension to the latter has led to the adaptation of spread-spectrum techniques which were originally developed in the mid 1950s[1] for military use as a means of providing a jam resistant communication link.

Spread-spectrum communication involves a transmission system which uses an overall frequency bandwidth far in excess of the minimum required. The receiver restores the signal to its normal bandwidth and, in so doing, it enhances the desired signal while suppressing the effects of all other prevailing signals and noise. Overall, the communication channel bandwidth is not 'wasted' because, with a suitable selection of transmission scheme, several nodes may simultaneously transmit messages over the same channel with almost zero interference between transmissions.

A local area network (LAN) such as Ethernet, where multiple nodes use a single communication channel—one pair at a time, is a prime candidate for the application of spread-spectrum techniques. Being inherently jam-proof, spread-spectrum techniques have the potential of reducing the contention and allowing multi-pair communications, thus increasing the network's throughput under these conditions.

This paper deals with an implementation of a synchronized spread-spectrum LAN by the Department of Electrical and Electronic Engineering, Footscray Institute of Technology. A code division multiple access (CDMA) scheme sense is used.[2] Within the CDMA channel a carrier sense multiple access scheme with collision detection (CSMA/CD) is used to reconcile the remaining contention problems.

2 Spread Spectrum Techniques

The three primary methods of spreading the data transmission are:[3–5]

(1) Direct sequence—in which the data is directly modulated with a fast pseudo-random code sequence.
(2) Frequency hopping—in which the carrier which is modulating the data is caused to change frequency in a pseudo-random manner.
(3) Time hopping—which can be likened to the pseudo-random allocation of channels in a time division multiplexed system.

The proposed network uses the direct sequence method, as shown in Fig. 1. The data signal $d(t)$ is multiplied by a fast pseudo-random binary sequence (the direct or chipping sequence) $p(t)$ to produce a received waveform $r(t)$ which, in the presence of an interfering waveform $J(t)$, would be

$$r(t) = d(t)p(t) + J(t) \qquad (1)$$

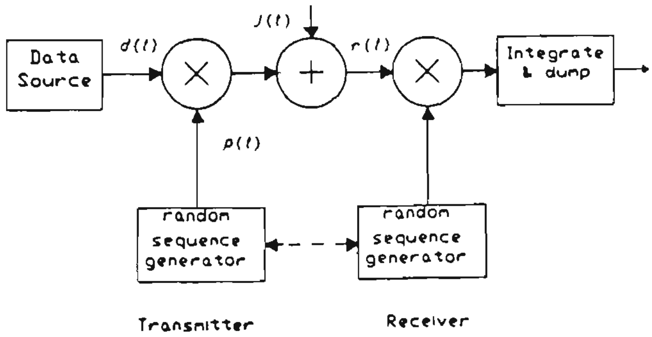


Fig. 1. Single channel direct sequence system.

© 1987 IERE

At the receiver this signal is multiplied by the same pseudo-random signal in order to recover the original data as in equation (2)

$$r(t)p(t) = d(t) + J(t)p(t) \qquad (2)$$

In equation (2) the jamming signal $J(t)$ may be external interference or it may be interference from other transmitting nodes. The rejection of the other nodes' transmissions as noise depends upon the orthogonality of the codes[6-8] used for spreading each node's data.

3 Network Considerations

Current bus and ring local area network topologies typified by Ethernet and the Cambridge Ring respectively employ some form of (asynchronous) time division multiplexing. This may be in the more or less fixed compartment format of the Cambridge Ring or the packet message type format of Ethernet. Whichever system is used, any node wishing to access the system must wait until a time when no other users are transmitting. The node will then transmit at a relatively high bit rate into the system. The overall effect is that each node emits bursts of data at high bit rates for relatively short periods and then remains dormant for a much longer period. Occasionally, two or more nodes attempt to transmit at the same time and the messages collide; back-off and retransmission algorithms would then be used to arbitrate. Thus, network throughput is dependent on the number of nodes in the network and the network bandwidth. Special chip sets capable of high speed burst transmission (e.g. Motorola 68590) have been developed for this purpose.

The application of spread-spectrum techniques to a local area network provides it with the ability to have many nodes transmitting simultaneously, thus increasing network throughput and possibly eliminating the need for very high speed transmissions. The system under development utilizes a code division multiple access scheme where each node has its own 'address' and associated code (or spreading sequence). The bus structure is shown in Fig. 2.

In the point-to-point communication system adopted in the network, if node X wishes to transmit to node Y then node X will modulate its data with the code of node Y. Hence, the transmitter of each node must be able to modulate its data with any recipient's address code. Since only one node at any instant in time may be transmitting to a particular address, a collision check for the intended receiver is still necessary, i.e. a CSMA/CD scheme is adopted within the channel. Thus, each transmitter has a built in demodulator to pre-check for collisions and continuously check while transmitting.

As a receiver, each node monitors the line, looking for data modulated with its own 'address code'. It treats data for any other intended recipient as noise and rejects it (as shown in equation (2)) but demodulates the data encoded with its own address code in order to recover the original data.

Each node has two demodulators—one for monitoring the channel for its own message and another one for collision avoidance and detection. The first demodulator works exclusively with its own address code whereas the code used in the second is switchable.

The task of choosing codes for their orthogonality can be simplified if the environment in which the system is to work, such as connection distance and hence propagation delay, attenuation, etc., can be controlled. This degree of control allows a synchronous network to be produced. Synchronization may be considered from both chip and data bit viewpoints. In a synchronous, cable-connected network such as this, with predictable delay and

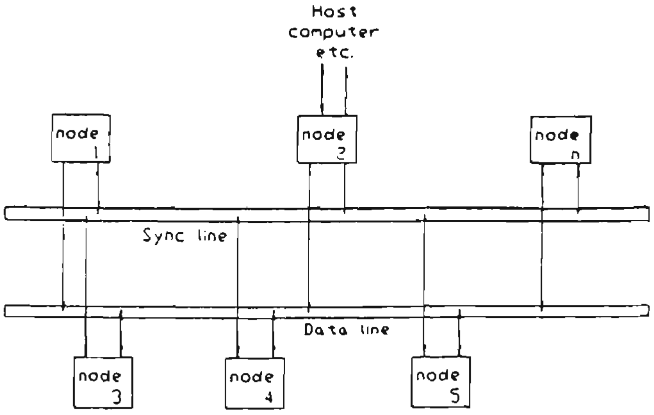


Fig. 2. Bus structured node interconnections.

attenuation characteristics, the effect of the so called odd-even cross correlation properties are also predictable and codes with multi-level cross correlation values, e.g. Gold or Kasami codes,[6-9] are not necessary. The code allocation scheme selected is based on providing each address with a code which is a phase shifted version of the master code; this master code is also the synchronization code transmitted on the sync. line indicated in Fig. 2. Thus the cross correlation between codes is the auto-correlation function of the one code. This simplifies both the code generation and decision making circuitry. The allocation of phase shifted versions of the one code to act as the individual node address codes is similar to that used by B. Hirosaki *et al.*[10] in their performance analysis of a fibre optics system. They showed that the variance $[q^i(t)]^2$ in mutual interference from node i is given by

$$[q^i(jNT_c)]^2 = 1/N^2 \qquad (3)$$

where j represents the individual bit position, N is the chipping sequence length and T_c is the chip sequence bit period.

Equation (3) above shows that the per-channel interference is inversely proportional to the square of the chip sequence length, or processing gain; and since one of the design aims was to produce a reliable and resilient system, a fairly large 'processing gain' of 1024 (approx. 30 dB) was selected. This choice of value for the processing gain produces a system capable of working in an electrically hostile environment and also allows a large number of simultaneous code transmissions.

$$G_p = \text{processing gain} = \frac{B_{SS}}{B_D} \qquad (4)$$

where

B_{SS} = bandwidth of the spread-spectrum signal, and
B_D = bandwidth of the data signal.

The concept of processing gain noted above is simply a power improvement factor which a receiver, possessing a replica of the spreading sequence, can achieve by a correlation process. For an ideal system the theoretical maximum number of users that a system could support would be equal to the processing gain G_p. This is not realistic under practical conditions and a figure of $G_p/10$ (Ref. 5) is usually accepted as being the maximum number of users which a system can accommodate.

Using maximal length sequences (MLS) of $2^{10}-1$ bits (i.e. 1023) to encode each data bit results in a residual offset effect at the output of the analogue multiplier which is dependent on the number of simultaneous transmitting stations. This results from the two level (1023, -1) autocorrelation function of an MLS. In order to eliminate

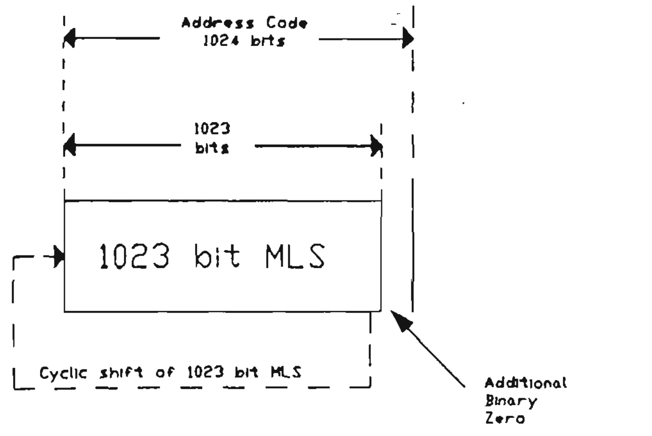


Fig. 3. Address code generation.

this residual effect the MLS sequences were modified to produce codes with (1024, 0) autocorrelation properties by the addition of an extra binary zero. The code for each station, therefore, becomes an MLS, cyclically bit shifted within its 1023 bit length, plus an additional binary zero which is transmitted as bit number 1024, as illustrated in Fig. 3. Each data bit is then encoded by the 1024 bit modified MLS.

The proposed network is based on twisted or parallel pair cable; this is to maintain simplicity and allow new nodes to be connected to, or disconnected from, the network without unduly affecting other users. Figure 4 shows a simple system layout with a number of transmitting nodes and a single receiving node. The output from each node is added onto the line to produce a received waveform $r(t)$ (Fig. 5), which is a summation of the outputs from each transmitter;

$$r(t) = s_1(t)+s_2(t)+s_3(t)+\ldots s_n(t) \tag{5}$$

$$r(t) = \sum_{n=1}^{n} s_n(t) \tag{6}$$

At the receiver the multi-level line signal $r(t)$ is multiplied by the receiver's own ± 1 level chip sequence $p_x(t)$ and from equation (2)

$$r(t)p_x(t) = d_x(t) + \sum_{n=1}^{x-1} s_n(t)p_x(t) + \\ + \sum_{n=x+1}^{n} s_n(t)p_x(t) + J(t)p_x(t) \tag{7}$$

where $1 \leqslant x \leqslant n$. The first term $d_x(t)$ is the recovered one-dimensional data signal and the last three terms are the multi-dimensional interference or jamming signals. The output from the analogue multiplier is subjected to low-pass filtering in the integrate and dump circuit in order to remove the unwanted high frequency components before being passed on to the decision circuit to determine whether this particular data bit represents a logic one or a logic zero.

In a completely asynchronous system, acquisition has two stages, with both the code and the code phase being vital to data recovery. In the completely synchronous system adopted in our network the code phase search stage is eliminated since channel address codes are in a fixed phase relationship with the synchronization channel. The phase search stage is replaced, however, by a requirement to scan the locally generated address code over a time window in order to identify possible delayed versions of the same code; this process is called tracking using the 'sliding correlator' method.[5] The maximum propagation delay in the system, therefore, determines the size of the window for tracking purposes and also the code allocation spacing since no two stations should be allocated codes which could be mistaken for delayed versions of each other within this tracking window.

Consider an example system with a tracking window of 2 bits. It would be necessary to allocate station addresses separated by more than 3 bits. Incremental tracking steps could vary from 1/4 to say 1/64 of a chip bit. The first would give rapid aquisition but only coarse tuning and a smaller 'noise margin'; the second would give much finer tuning but at the expense of aquisition time and the need to include a longer data packet synchronization preamble. A compromise figure of 1/16 bit would seem appropriate, representing the need for a preamble of 32 bits to each data packet. The practice of including a data packet preamble is common in most LAN systems. This figure of 32 tracking steps compares favourably with that which would be necessary if different codes were used for each address and all code phases had to be tested. The system in this example would work in a network up to 450 metres long at the present chip rate of 1·2288 MHz. The practicality of increasing the chip rate beyond this figure is being investigated.

In order to test phase synchronization aspects, a novel error generator which produces random errors at rates between 1×10^{-6} and 9×10^{-1} is used. This generator is capable of maintaining the error rates irrespective of the input data rate, which may vary from 1 Hz to 1·5 MHz. Tests with the prototype system have shown that the system reliably achieves phase synchronization within 40 ms, even with an error rate as high as 1×10^{-2}, and that synchronization drop-out does not become a problem until the error rate reaches 2×10^{-2}.

4 Code Investigations

Computer programs have been written to investigate the system coding aspects. These programs are designed to emulate (in non-real time) the ideal line drive characteristics of a 60 user system and have been written to investigate two areas of coding and code allocation:

(1) To ensure that transmission number x (where $1 \leqslant x \leqslant 60$, see Fig. 4) will be successfully decoded when a number of simultaneous transmissions are taking place. A program emulating 60 nodes has been developed and the correlation properties investigated. This has verified the (1024, 0) correlation properties of the codes selected.

(2) To investigate the orthogonality properties of the codes used. This part takes the form of emulating a number of simultaneous transmissions (numbers 1 to n excluding x) and attempting to extract data using code $p_x(t)$ (i.e. the code omitted). Up to 59

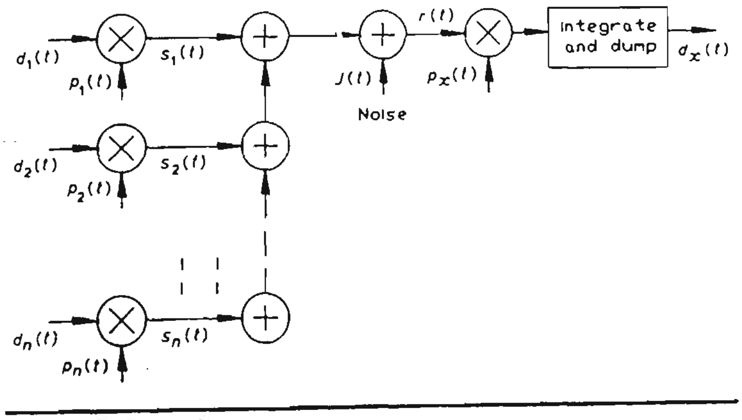


Fig. 4. Simple system layout.

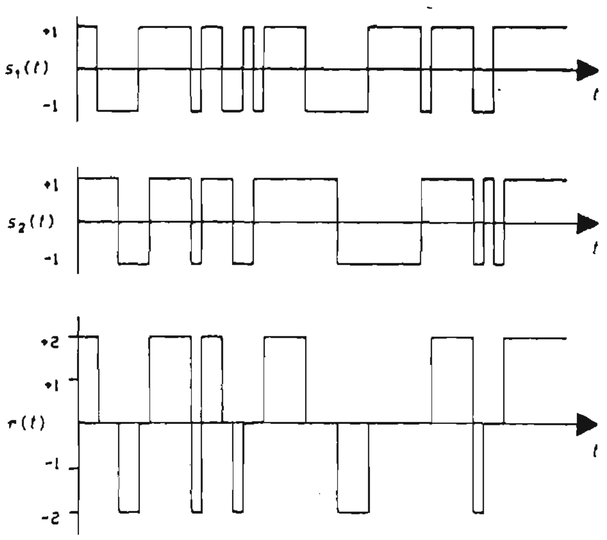


Fig. 5. Typical transmitter waveforms.

simultaneous transmissions have been demonstrated to create no mimicry problems, i.e. no combination of code transmissions from up to 59 stations looks like the code of the non-transmitting station x. These tests have been carried out with channel allocations based on code separations of 1, 5 and 10 bits, with the same results.

5 Prototype System Tests

The system was configured as shown in Fig. 4 for the purpose of the preliminary bit error rate (BER) measurements. Three transmitters and two receivers were used. The first transmitter/receiver pair was used for the BER measurements and the second pair used to verify the simultaneous transmission capability, the third transmitter acting merely as an additional source of interference. Each transmitter, $s_1(t)$ to $s_3(t)$, outputs a signal which represents the summation of ± 1 volt on to the transmission line, as illustrated in Fig. 5. In the absence of other signals, therefore, $s_1(t)$ would cause a ± 1 volt signal to appear at $r(t)$. The upper trace in Fig. 6 shows the total composite signal $r(t)$ (see Fig. 4) when $s_1(t)$, $s_2(t)$, $s_3(t)$ and $J(t)$ (added white Gaussian noise (AWGN)) are all present. $J(t)$ is band limited to 3 MHz. The signal-to-noise figures are actually signal-to-total-interference figures, S/I, and are the ratio of the signal $r(t)$ due to $s_1(t)$ alone compared to that due to all signals other than $s_1(t)$.

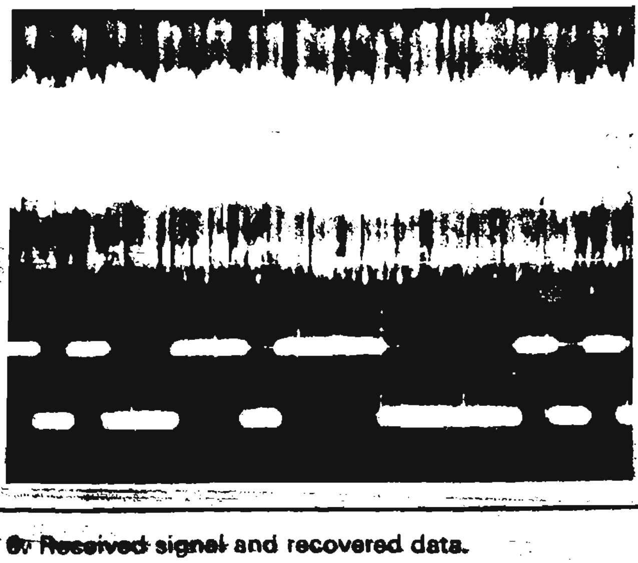


Fig. 6. Received signal and recovered data.

Signals were measured with a true r.m.s. voltmeter the input to which was band limited to 3 MHz. The upper trace in Fig. 6 shows the composite signal $r(t)$ at a S/I ratio of -10 dB, and the lower trace shows the data recovered on channel 1. Figure 7 shows the preliminary results of the BER vs S/I Ratio measurements taken on channel 1 with channels 2 and 3 both transmitting and AWGN also present. This illustrates the system's ability to recover data in the presence of simultaneous multiple transmissions and external interference.

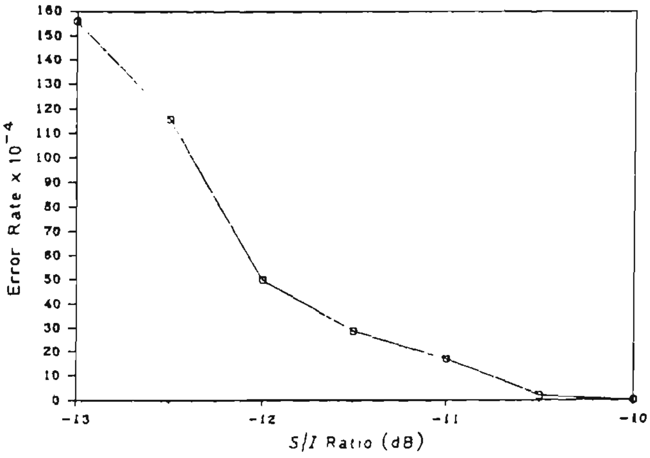


Fig. 7. BER vs S/I ratio.

6 Conclusions

By applying a CDMA spread-spectrum technique to the design of a local area network, a simple and error resistant network is being developed. Its main application will be either as a network in an electrically noisy environment or in a high throughput multiple access network. Computer studies carried out indicate that the system will be capable of supporting over 60 users transmitting simultaneously. Tests carried out on a prototype network consisting of 5 nodes (3 transmitting nodes and 2 receiver nodes) have verified the results predicted by the computer studies and demonstrated the feasibility of simultaneous multi-pair communications.

The next stages of the project will be to investigate alternative phase synchronization schemes and to determine the protocols necessary to control this type of system.

Whilst it is too early at this stage of development to make accurate predictions regarding the cost and reliability of the final network, the reliability of the low cost development modules has been encouraging.

7 Acknowledgments

This research project is being supported by a grant from the Australian Telecommunications and Electronics Research Board.

8 References

1 Scholtz, R. A., 'The origins of spread-spectrum communications', *IEEE Trans on Communications*, **COM-30**, no. 5, pp. 822–54, May 1982.

2 Smythe, C. S. and Spracken, C. T., 'A high speed local area network using spread-spectrum code-division-multiple-access techniques', *The Radio and Electronic Engineer*, **54**, no. 5, pp. 225–6, May 1984.

3 Dixon, R. C., 'Spread-spectrum Systems' (Wiley, New York, 1976).

4 Holmes, J. K., 'Coherent Spread-spectrum Systems' (Wiley, New York, 1982).

5 Pickholtz, R. L., Schilling, D. L. and Milstein, L. B., 'Theory of spread-spectrum communications—a tutorial', *IEEE Trans.*, **COM-30**, no. 5, pp. 855–84, May 1982.

6 Sarwate, D. V. and Pursley, M. B., 'Cross-correlation properties of pseudo-random and related sequences', *Proc. IEEE*, **68**, no. 5, pp. 593–619, May 1980.

7 Gold, R., 'Optimal binary sequences for spread-spectrum multiplexing', *IEEE Trans on Information Theory*, **IT-13**, no. 4, pp. 619–21, October 1967.

8 Bruce-Carlson, A., 'Communication Systems—An Introduction to Signals and Noise in Electrical Communication' (McGraw-Hill, Tokyo, 1981).

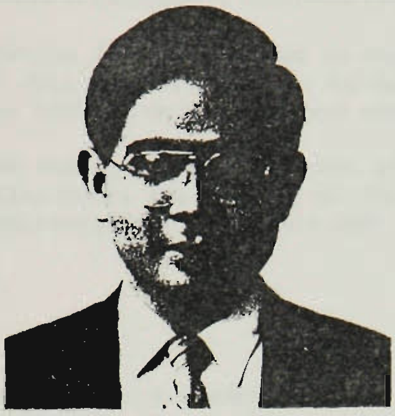
9 Pursley, M. B. and Sarawate, D. P., 'Performance evaluation for phase-coded spread-spectrum multiple-access communications—part II', *IEEE Trans*, **COM-25**, no. 8, pp. 800–3, August 1977.

10 Hirosaki, B., Hasegawa, S. and Sawai, A., 'Spread-spectrum multiple-access data loop', *NEC Res. & Dev.*, no. 71, pp. 48–58, October 1983.

Manuscript first received by the Institution on 28th October 1985 and in revised form on 13th May 1986.
Paper No. 2257/COMM423

The Authors

A. L. Simcock



C. na Ranong

P. S. K. Leung

Alec Simcock graduated from the University of Kent at Canterbury with a BSc(Hons) degree in Electronics in 1972. He worked for 12 years with the Procurement Executive of the Ministry of Defence in the UK, researching in digital circuit design, digital data switching, computer controlled communication systems, and microprocessor replacement of digital circuits. The work also involved the implementation of software project management techniques. He was appointed to a Lectureship with the Department of Electrical and Electronic Engineering, Footscray Institute of Technology in 1983.

Chula na Ranong received the BE(Hons) degree in Electrical Engineering and PhD from the University of New South Wales in 1970 and 1973 respectively. Since 1972 he has been with the Department of Electrical and Electronic Engineering, Footscray Institute of Technology, as a Lecturer and, later, Senior Lecturer in Digital Electronics and Systems. During 1977–78, he spent a year as a Visiting Professor at Cornell University, USA. Then, for the first six months of 1982, he worked as a Specialist Engineer with Compagnie Européenne de Télétransmission (Thomson-CSF) in France developing a mass storage system for a real-time digital data network supervisory control system. More recently, as a result of his interest in Automation Technology, Dr Ranong spent two months in Japan in 1984 visiting government, industrial and educational research institutions.

Patrick Leung received the degrees of BE(Elect) and MEngSci in Electrical Engineering from the University of Melbourne in 1972 and 1974 respectively. Between 1975 and 1977 he was a Lecturer in the Department of Electrical and Electronic Engineering, Footscray Institute of Technology. In 1977 he joined Telecom Australia as an Engineer in the Datel design and provisioning section, but rejoined F.I.T. as a Lecturer in Communications Engineering in 1979. In the first half of 1985 he visited the Department of Electrical Engineering, Ottawa University, as a Research Fellow, to carry out work in data communications over satellite channels. On his return to F.I.T. he was appointed Senior Lecturer in Communications. His main interests are data transmission techniques, data networks and data protocols.

Conference on Computing Systems
and Information Technology 1987
Brisbane, 17-19 June, 1987

Protection, Rejection and Throughput Characteristics Of A Spread Spectrum Local Area Network

A.L. SIMCOCK
Lecturer, Department of Electrical & Electronic Engineering, Footscray Institute of Technology
C. NA RANONG
Centre for Automation Technology, Department of Electrical and Electronic Engineering, Footscray Institute of Technology
P. LEUNG
Senior Lecturer, Department of Electrical & Electronic Engineering, Footscray Institute of Technology

SUMMARY More and more information is being stored on magnetic media, with Local Networks of computers sharing this data. Since some of this material will be valuable to the owner it is necessary to protect the data, firstly against corruption during transmission and secondly against prying or interception.

This paper describes some of the more pertinent features of a Local Area Nerwork being developed, by the Department of Electrical and Electronic Engineering of the Footscray Institute of Technology, to address these points. The throughput characteristics of this LAN, based on Spread Spectrum techniques, are also described.

1 INTRODUCTION

The advent of the microprocessor has had a dramatic impact on society. The significant reduction in the cost of computing power has resulted in the widescale use of computers for all kinds of data manipulation and storage. Magnetic storage devices such as floppy disks have, to some extent, replaced paper as a means of intra and inter company data transfer. Local Area Networks (LAN) were primarily developed to allow the sharing of expensive resources, but their development has also allowed the sharing of data between interested parties without the need to physically transfer the storage medium.

The ever increasing use of LANs has brought with it one problem with two manifestations. The problem is to protect the data. Firstly, it is necessary to protect the data against interference, be it deliberate or otherwise, in order to ensure delivery of correct information to the correct user. If this is achieved the LAN may be used with confidence in an electrically hostile environment. Secondly, as more proprietory information is appearing on data bases, it is becoming increasingly necessary to protect the information against prying or interception. Society has, indeed, become information conscious to the extent that information, the acquisition of it and the protection of it has become an industry in its own right.

Various techniques are already being used, or are under development, to address these problems. One communication technique which looks to have the potential to be of assistance in both areas is called Spread Spectrum (SS). The Department of Electrical and Electronic Engineering at Footscray Institute of Technology is developing a Local Area Network based on these Spread Spectrum techniques.

2 CHARACTERISTICS OF SPREAD SPECTRUM

One definition of Spread Spectrum which adequately reflects the major characteristics of this technique is as follows:-

> "Spread Spectrum is a means of transmission in which the signal occupies a bandwidth in excess of the minimum necessary to send the information. The band spreading is accomplished by modulating the data with a code, which is independent of the data. A synchronised version of the same code is then used at the receiver in order to despread the signal and facilitate data recovery."

There are three primary methods of achieving this bandwidth spreading:-

a) Direct Sequence - in which the data is directly modulated with a fast pseudo random code sequence.

b) Frequency Hopping - in which the carrier which is modulating the data is caused to hop from one frequency to another in a pseudo random manner.

c) Time Hopping - which may be likened to the pseudo random allocation of channels in a Time Division Multiplexed (TDM) system.

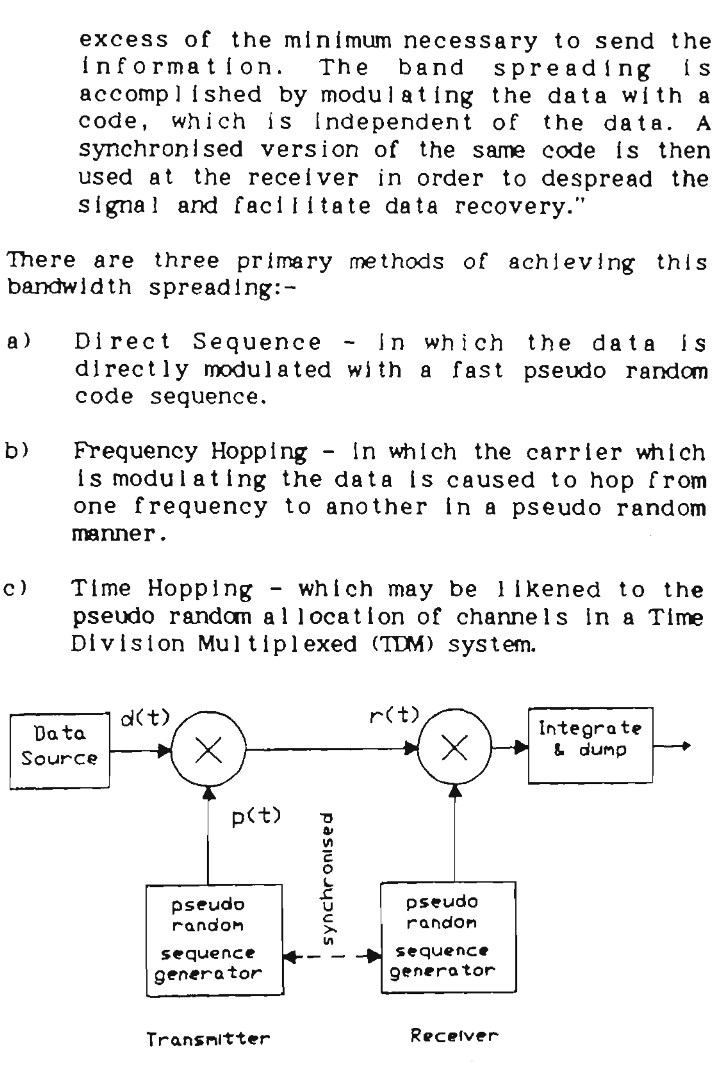


Figure 1 Components of a SS Communications System

Figure 1 illustrates a typical single channel direct sequence system. The data d(t) is multiplied by the chip sequence p(t) creating a received signal, r(t) of the following form:

$$r(t) = d(t)p(t) \tag{1}$$

This signal is re-multiplied by a synchronised version of the chip sequence in order to collapse the signal back into the data signal's bandwidth and allow data recovery

$$r(t)p(t) = d(t)p^2(t) \qquad (2)$$

And, since $p(t) = \pm 1$, $p^2(t) = 1$ therefore

$$r(t)p(t) = d(t) \qquad (3)$$

It may seem strange to devise a transmission system which intentionally uses more bandwidth than necessary, but the very nature of the Spread Spectrum signal characteristics imparts at least five important performance attributes :-

i) Low Probability of Intercept (LPI). This can be achieved with the use of a high processing gain and unpredictable carrier signals, which result in power being spread thinly and uniformly over the spectral bandwidth. This makes detection against background noise, by a surveillance receiver, very difficult. A Low Probability of Position Fix (LPPF) attribute goes one step further in including both intercept and Direction Finding (DFing) in its evaluation. Low Probability of Signal Exploitation (LPSE) or giving all stations a Universal Transmitter Fingerprint (UTF) offer additional benefits in that signal identification and exploitation possibilities are denied to the interceptor.

ii) Antijam capabilities can be gained with the use of an unpredictable carrier signal. The jammer cannot use signal observations to improve its performance, and must rely on jamming techniques which are independent of the signal being jammed.

iii) High time resolution is attained by the correlation detection of wide band signals. Differences in the Time Of Arrival (TOA), of the wide band signal, in the order of the reciprocal of the signal bandwidth, are detectable. This property can be used to suppress multipath and, at the same time, render repeater jammers ineffective or minimise the effects of reflections, caused by impedance mismatches in cable systems.

iv) Transmitter Receiver pairs using independent pseudo random carriers can operate concurrently in the same bandwidth with minimal inter-channel interference. These are called Spread Spectrum Multiple Access (SSMA) or Code Division Multiple Access) systems.

v) Cryptographic capabilities result when the data modulation cannot be distinguished from the carrier modulation, and the carrier modulation is effectively random to an unwanted observer. In this case the SS carrier modulation takes on the role of a key in a Cipher system. A system using indistinguishable data and SS carrier modulation is a form of privacy system.

These attributes may be seen to cover many applications and have been extensively exploited in the military environment, where factors such as Antijam and a Universal Transmitter Fingerprint (i and ii above) have particular significance. These are not, necessarily, the factors of interest when the question of data protection in a LAN is to be addressed, features such as interference rejection, a multi user capability and privacy (attributes iii, iv and v above) are of greater interest and will be considered further.

3 PROTECTION PROPERTIES

The protection and rejection properties inherent in SS systems come from the codes used to spread the data. For cryptography the ideal code should:-

1) Be easy to generate
2) Be random in nature
3) Have a long period
4) Be difficult to reconstruct from a short segment.

Both cryptographic and SS codes require properties 1, 2 and 3 but they have different requirements with respect to property 4. A good cryptographic code should be difficult (almost impossible) to reconstruct from a short segment, in order that data recovery by anyone other than the intended recipient should be as difficult as possible. SS systems, on the other hand, require that the transmitter and receiver have synchronised codes to facilitate the bandwidth collapsing necessary for data recovery. This means that the receiver should be able to synchronise the locally generated code to that of the transmitter, and in order to do this in a practical amount of time it should be possible to regenerate the code from a relatively short segment. From this point of view, therefore, SS and cryptographic codes have conflicting requirements. Cryptographic and SS protection are not, fortunately, mutually exclusive as SS codes do provide protection against the casual eavesdropper.

The act of spreading the data causes the frequency spectrum of the transmitted information to be changed from that shown in Figure 2 (which is the spectrum of the data, d(t)) to that shown in Figure 3 (which is the spectrum of the spread data, d(t)p(t), with the chip sequence, p(t), having a frequency of f_c). This act has benefit from two points of view, firstly the receiver must remove the spreading effect of the chip sequence, ie. decrypt the transmission, in order to obtain the original data, and secondly the actual magnitude of the signal is very much reduced; possibly into the noise threshold region. This means that the receiver must have both the code and the SS demodulation equipment to recover the data.

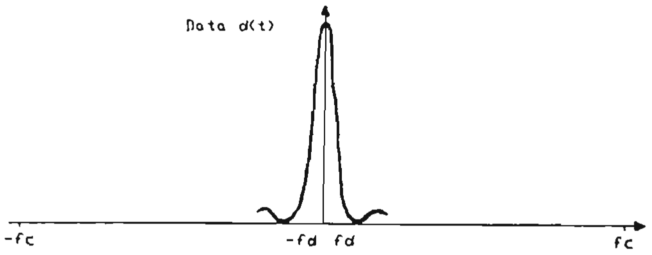


Figure 2 Spectral Occupancy of Data Signal

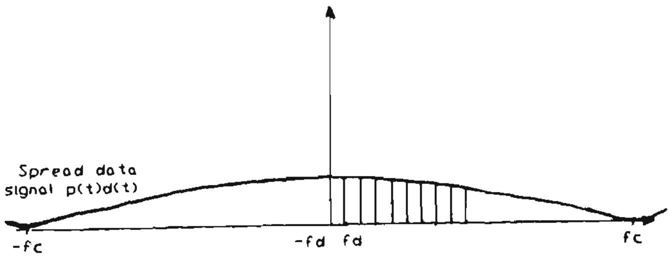


Figure 3 Spectral Occupancy of Spread Data Signal

Maximal Length Sequences (MLSs) and Gold codes are probably the most common codes for use in SS systems. MLSs certainly have the appropriate properties with respect to synchronisation etc. in SS systems. The two level autocorrelation function of the MLS makes synchronisation a relatively straightforward matter and the code is relatively long ie.the code length, N, is given by $2^n - 1$, where n is the number of shift registers in the generator function. Even if the code is relatively long, however, synchronisation can be obtained relatively quickly. If n = 100 such that the code length $N = 2^{100} - 1 \approx 10^{30}$, it would still be possible to reconstruct the entire sequence from only 198 bits by solving 198 linear equations, not a difficult task for a computer.

For multi-user applications, however, MLSs are not ideal since the cross correlation between MLSs of the same length (but generated from different polynomial expressions) can vary widely. Gold codes are generated in "families" and the cross correlation between different codes within the same family is predictable and subject to identifiable bounds. Gold code generation is illustrated in Figure 4. Figure 4a shows how a sequence may be formed from a Linear Feedback Shift Register (LFSR) of degree 10 which generates a sequence of period $2^5 - 1 = 31$, from the generator polynomial

$$F = x^{10} + x^4 + x + 1 .$$

Since there are $2^{10} - 1$ possible non-zero initial states, the number of states which result in different sequences, ie. the number of codes in the family, is $(2^{10} - 1)/(2^5 - 1) = 2^5 + 1 = 33$. Figure 4b illustrates the same Gold code family being generated from two independent 5 bit LFSRs,

$$F_1 = x^5 + x^3 + x + 1$$

and $F_2 = x^5 + x^3 + 1$

the outputs of which are modulo 2 added to form the output code. The actual code generated, out of the family of codes, is dependent on the relative phase of the two 5 bit LFSRs.

Neither Gold codes nor MLSs are particularly good cryptographic codes, but it is possible to design a SS system which uses shorter codes for synchronisation and, having achieved synchronisation, then reverts to the use of a code with better cryptographic properties for actual data spreading. Cross correlation properties in a multi-user environment would, however, still have to be considered.

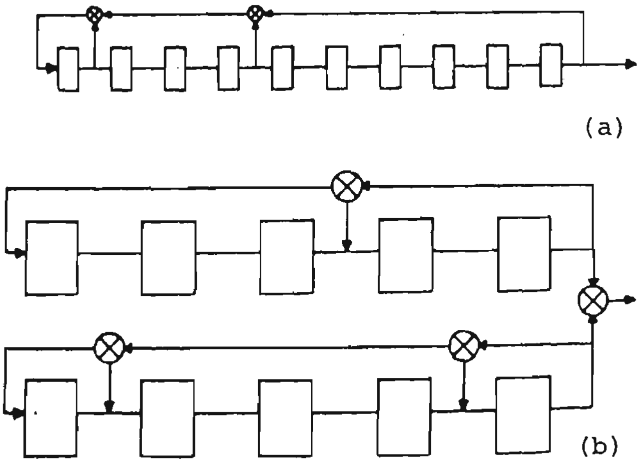


Figure 4 Gold Code Generation

4 REJECTION PROPERTIES

The circuit shown in Figure 1 represents the ideal case, Figure 5 illustrates a more realistic system. Each of the 'n' transmitters has been allocated a code for data spreading purposes. These codes should, ideally, be completely orthogonal but, even if the codes themselves are orthogonal, in reality amplitude and phase distortion will cause some interference in the demodulation process being carried out by the receiver.

External interference, J(t), may also be present, as illustrated in figure 5. Equation 3 does not indicate the full picture which, with several stations transmitting and in the presence of external interference, is more realistically represented by

$$r(t)p_x(t) = d_x(t) + \sum_{n=1}^{x-1} p_x(t)s_n(t) + \sum_{n=x+1}^{n} p_x(t)s_n(t) + J(t)p_x(t) \qquad (4)$$

Where $s_n(t) = d_n(t)p_n(t)$ and $1 \leqslant x \leqslant n$

The first term, $d_x(t)$, occupies the bandwidth of the data signal, f_d, see Figure 2. The remaining three terms occupy a much greater bandwidth appropriate to the chip sequence frequency, f_c, see Figure 3. By filtering out the high frequency components it is possible to leave only the data signal $d_x(t)$.

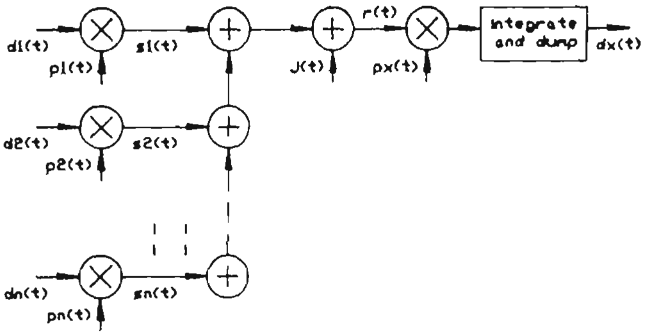


Figure 5 Realistic System Model

It can be seen, therefore, that using more chip bits to represent each data bit spreads the unwanted signals over a much wider bandwidth, with an accompanying reduction in magnitude, thus making the filtering more effective. The number of chip bits per data bit is called the Processing Gain of the system. The Processing Gain gives an indication of the interference rejection capabilities of the system, but the true figure of merit also depends on the SNR of the filter and on the codes used. For a given Processing Gain it is possible to have one long sequence modulating the data, or to have several repetitions of a shorter sequence. Since, however, the spacing of the spectral lines, Δf, is given by f_c/N, a long sequence will have many lines closely spaced, and a shorter sequence will have fewer lines further apart. This indicates that a long sequence should give better results than a short sequence.

5 THE FIT SYSTEM

The system being developed at FIT uses the Direct Sequence method of data spreading, and is connected using a Bus Topology, as shown in Figure 6. A Code Division Multiple Access (CDMA) mechanism is used to allow multi-user access. In this type of system each node is allocated a unique chip sequence as its "address code". Data intended for a node must be modulated with the "address code" for that particular node, thus if Node #1 wishes to communicate with Node #3 then its data must be modulated with the code for Node #3. Codes allocated to each node are bit shifted versions of the one modified MLS, and the whole system is synchronised by means of a separate sync channel.

The system being developed at FIT also offers a security facility by the very nature of its construction. In this synchronised system the delay time between one Node and another is fixed, and could be validated, thus any party wishing to inject mis-information into the system by posing as a valid user would have to be co-sited with that user, or emulate the inter-node delay properties exactly.

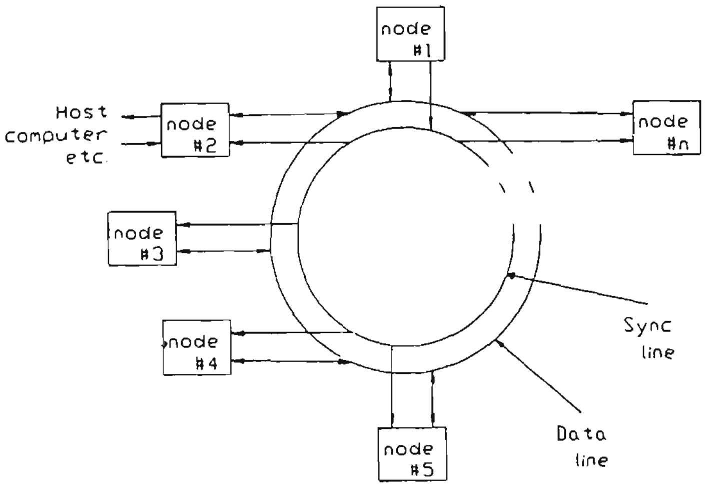


Figure 6 Bus-Connected System Topology

A prototype system similar to that illustrated in Figure 5 has been built for the purpose of preliminary Bit Error Rate (BER) against Signal to Noise (S/N) measurements; where Noise is used to represent total interference, ie external interference J(t) plus interference caused by other nodes transmitting simultaneously. Three transmitters and two receivers were used. The first transmitter/receiver pair were used for the BER vs S/N measurements. The second transmitter/receiver pair was used to verify the simultaneous transmission capability, the third transmitter acting merely as an additional source of interference.

Figure 7 shows preliminary results of the BER vs S/N Ratio measurements. The preliminary results obtained from the prototype system have verified the concept of a synchronous LAN using bit shifted modified Maximal Length Sequences as the address codes. Data recovery has been demonstrated, with the error rates indicated in Figure 7, thus illustrating the potential of such a system for use in an electrically noisy environment.

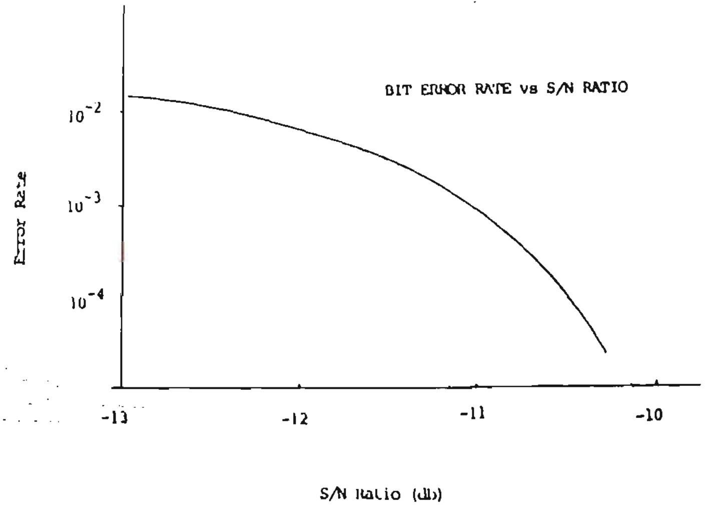


Figure 7. BER vs S/N Ratio

6 THROUGHPUT CHARACTERISTICS

The modified MLS codes were chosen to generate as little co-channel interference as possible, it is assumed that each (CDMA) channel's traffic may be regarded as independent in order to model individual channel throughput characteristics. Whilst each channel's traffic may be regarded as independent it is possible to have contention within an individual channel. This contention is handled by a mechanism similar to Carrier Sense Multiple Access with Collision Detection (CSMA/CD). The specific channel address code is the "carrier" being sensed in this case.

The techniques used in the derivation of this throughput model are based upon those used by Kleinrock and Tobagi. The full development of the model is outside the scope of this paper but the final traffic throughput, S, in terms of the offered traffic, G, and the busy sensing period, **a**, is given by

$$S = \frac{Ge^{-aG}}{(3aG + (e^{-aG}(2 + G - 3aG - e^{-aG})))} \qquad (5)$$

Figure 8 is a graph of S vs G, and several curves are included in order to represent different values of **a**. These curves show the expected characteristic shape with the throughput increasing as the offered traffic increases until a breakpoint is reached. Above this point the probability of collisions occurring is such that increasing the offered traffic actually causes a reduction in throughput as the channel capacity is progressively taken up with more retransmissions and less new traffic. It can be seen from Figure 9 that the busy sensing time is a highly important parameter in the throughput model and that reducing this sensing time causes the breakpoint in throughput to occur at higher levels of offered traffic.

The throughput characteristics of this model are compared with those of some of the more familiar multi-access protocols in Figure 9. These curves are plotted using a value of **a** = 0.01, except for the two Aloha protocols which are independent of **a**. In the Non-persistent and 1-persistent CSMA models the value of **a** represents the propagation delay in the system, rather than a busy sensing period as in this model, but in all three models **a** gives a representation of the 'vulnerable period' for transmission modelling purposes. These curves show the comparison between the FIT system model, which has been developed for a LAN application, and other protocols (most noticably Aloha and Slotted Aloha) which were not. The indication is that application specific protocols can offer throughput advantages.

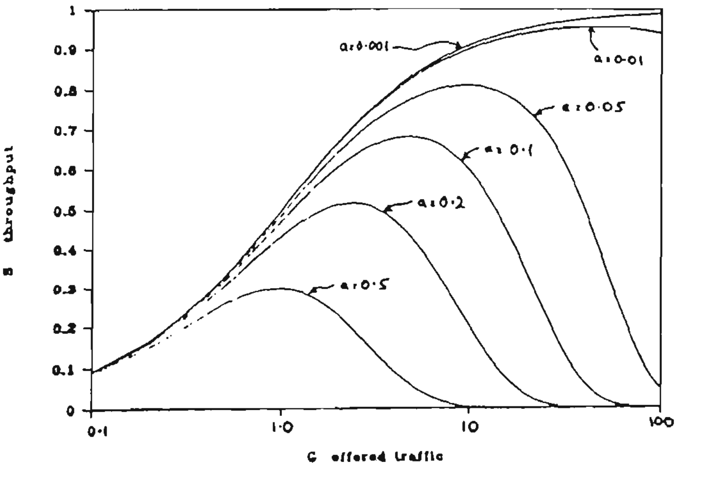


Figure 8. Throughput vs Offered Traffic

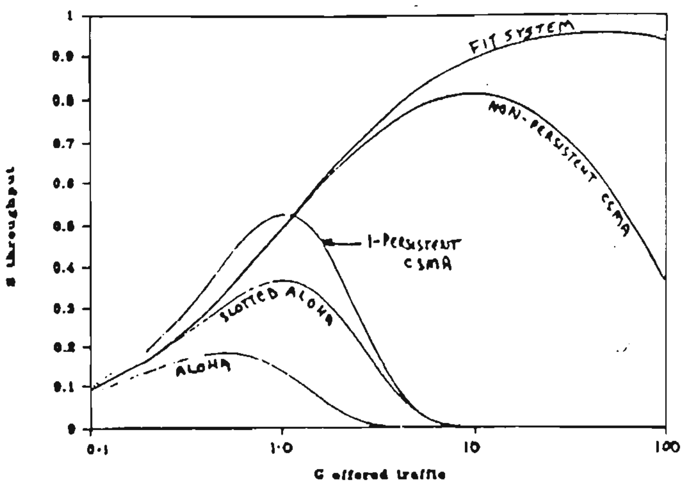


Figure 9. Comparison of Throughput Characteristics

7 CONCLUSIONS

The application of CDMA spread spectrum techniques to the design of a Local Area Network can result in a system with inherent data privacy characteristics, which is relatively simple and also has significant error rejection characteristics.

A system of this nature could, therefore, be used where noise rejection in an electrically noisy environment is of prime concern, or to provide validation and privacy for valuable information. Whatever the application the throughput model indicates that this type of system provides a "high availability" system for multi-user environments.

8 ACKNOWLEDGEMENTS

This research project is being supported by a grant from the Australian Telecommunications and Electronics Research Board.

9 REFERENCES

DIXON, R.C. (1976). "Spread Spectrum Systems. New York, Wiley.

HOLMES, J.K. (1982). Coherent Spread Spectrum Systems. New York, Wiley.

KLEINROCK, L. and TOBAGI, F.A. (1975). Packet Switching in Radio Channels : Part 1. I.E.E.E. Transactions on Communications. Vol-23, pp 1400-1416.

PICKHOLTZ, R.L., SCHILLING, D.L. and MILSTEIN, L.B. (1982) Theory of Spread Spectrum Communications - A Tutorial. I.E.E.E. Transactions on Communications. Vol. COM-30, pp. 855-884.

SCHOLTZ, R.A. (1982). The Origins of Spread Spectrum Communications. I.E.E.E. Transactions on Communications. Vol. COM-30, No 5, pp. 822-854.

SKAUG, R. and HJELMSTAD, J.F. (1985). Spread Spectrum in Communications. Peter Peregrinus.

The Institution of Engineers, Australia

MECHANICAL COLLEGE TECHNICAL REPORT

SPONSORED BY ASSCO

A spread-spectrum manufacturing information network

by A.L. Simcock, C. na Ranong and P. Leung

This paper concerns the development and implementation of a totally integrated, computerised manufacturing environment. This concept has been called by a number of buzz-words or phrases: computer integrated manufacturing (CIM), flexible manufacturing system (FMS), integrated manufacturing system (IMS). Whatever it is called most Australian manufacturers will come across it in the next few years, and the rapidity of the absorbtion of the CIM concept may well decide the future of Australia as a manufacturing country.

Introduction

Footscray Institute of Technology is developing a computer interconnection system, or local area network, the architecture of which makes it particularly suitable for use in the type of environment often found in a manufacturing environment. A description of the background and development of the concepts of CIM would, however, be useful in order to put into proper perspective the system being developed.

History of MAP

Several years ago General Motors realised the need for a multivendor local area network (LAN) for factory automation. The scale and complexity of machinery and computers needed on the factory floor was so vast that no single manufacturer could provide all the required equipment. This state of affairs led to the formation of **islands of automation**, as shown in figure 1, unable to communicate on any type of global scale. To remedy this situation GM set about defining a true multivendor LAN, basing it on existing standards and protocols where appropriate. Where gaps existed GM worked with the standards committees and other interested parties to fill these gaps. The outcome of this was GM's **manufacturing automation protocol (MAP).**

The public first became aware of the benefits at the Las Vegas National Computer Conference in July 1984, where the first demonstration of inter-vendor interoperation took place using GM's MAP. Only seven companies took part: Allen Bradley, Concord Data Systems, DEC, Gould, HP, IBM and Motorola. By the time the second public demonstration occurred, in November 1985 at the Autofact 85 exhibition in Detroit, 21 companies were demonstrating equipment operating with GM's MAP. This sparked off enormous interest in MAP and now user groups are springing up all over the US, Europe and Japan. Two "interest groups" have also been formed in Melbourne and Sydney.

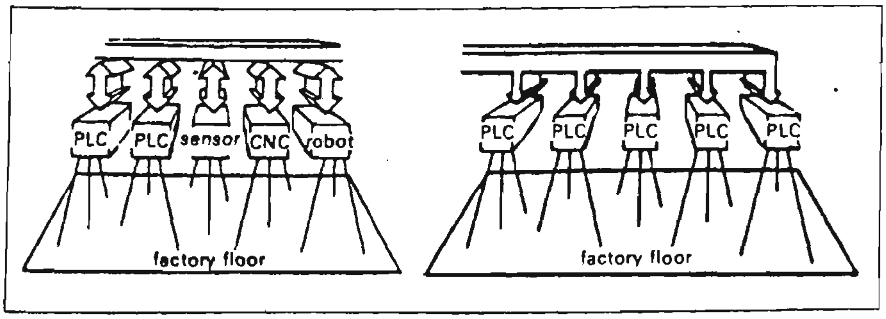


Figure 1: Islands of automation

The requirement

To achieve an integrated manufacturing system it is necessary to be able to take these **islands of automation** and connect them together in an orderly fashion, so that they are no longer isolated but form part of a coherent network, as shown in figure 2, resulting in factory-wide automation. This is not the end of it, however, as other sections within the company could benefit from being incorporated into the network. With this in mind the Society of Manufacturing Engineers produced the computer integrated manufacturing or **CIM wheel**, figure 3. This represents a company-wide networking requirement.

Many of the interactions are obvious eg machine scheduling, manpower, production rates etc. Some of the other interactions, though not as obvious, may be equally crucial to company profitability. The goods inward, stores receipts and purchasing department interaction may well mean that less capital could be tied up in raw materials and cash flow/liquidity improved by exercising better control over stock and re-order levels etc. If this sounds familiar, it should do, because this is *precisely* the type of control needed to implement **zero inventory** or **just-in-time** procedures.

How it is achieved

Computers around thctory may be combined encompassing both the manufacturing automation protocol (courtesy of GM), coupled in with the **technical office protocol** (TOP), which was pioneered by Boeing. The combination of the two makes factory and administration internetworking

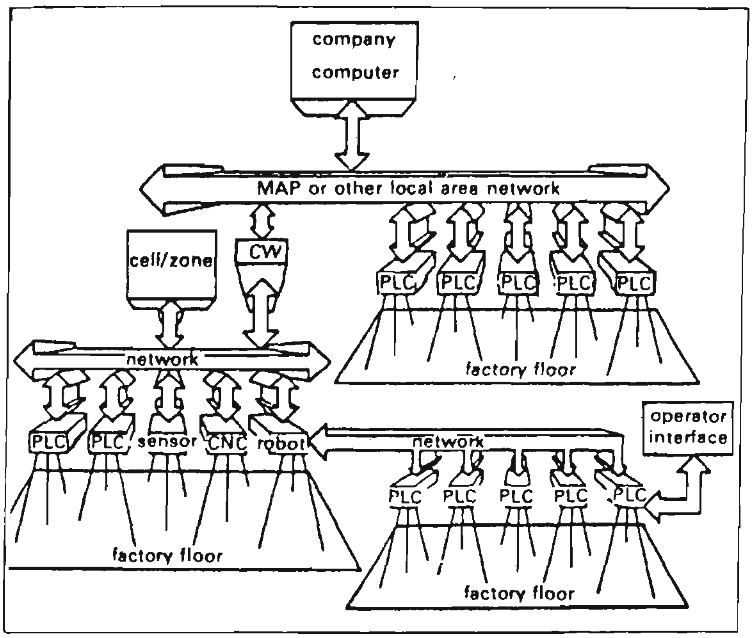


Figure 2: An integrated network

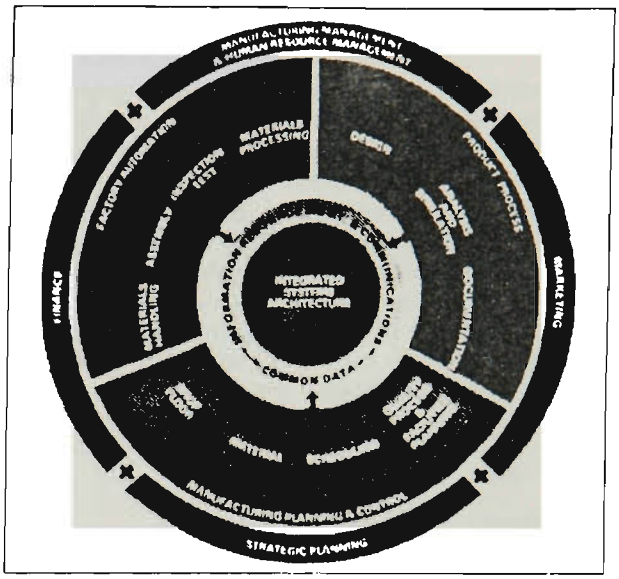


Figure 3: The Society of Manufacturing Engineers' CIM wheel –

Copyright 1985 CASA/SME, 2nd edition, revised Nov 5 1985

Selecting a CAD system from ASSCO is more than a casual affair

At ASSCO, we believe in long-term relationships. We know that buying a CAD system is only half the story.

ASSCO has always specialised in solving the data manipulation, computation and drafting needs of professional engineers.

And, engineers need the right advice from people who understand the business from both sides. We take the time to train you and your staff properly so that productivity is maximised right from the start.

Then, when you are ready, we install the CAD system quickly and efficiently. After that, we keep in close touch from then on. Our after sales service is second to none and we stock all spares and consumables.

You can rely on ASSCO to keep your CAD system performing at peak efficiency for a long, long time.

Contact ASSCO now. But only if you are interested in a long-term relationship.

Long-term productivity and performance for professionals.

ASSCO

17 Heatherdale Road, Ringwood 3134
Phone (03)873 2266

Sextant 154

Circle 1 on reader service coupon

possible this creating company-wide integration. In order to establish data transfer, control and communication the computers are connected together into a network. This can be accomplished in several ways. If there were only two computers A and B (commonly called stations or nodes) they could be connected in what is called the point-to-point configuration (or topology), as in figure 4a. This is impractical with more than two nodes and other topologies must be considered. Each topology has its problems, both the ring and the star, figures 4b and 4c, are subject to significant disruption in the event of even a single point failure. The bus, figure 4d, is not affected in the same way but information could be lost if two or more nodes attempt to transmit simultaneously; there are, however, protocols to minimise this problem. MAP and TOP use two different protocols to handle the contention problem.

MAP and TOP specifications

MAP and TOP have both been recognised by the IEEE as worthy of support in the form of standards publication and classification. This means that any supplier can obtain a copy of the standard and produce equipment satisfying the requirements. Table 1 lists some of the more important features of both MAP and TOP.

Both MAP and TOP are designed to implement the International Standards Organisation open systems interconnection seven-layer model for a communications network, shown in figure 5. The lower 3 layers have been fully defined to the point where most, if not all, features have been agreed. Layer 4 is almost at the same general acceptance level. These lower layers are designed to provide reliable end-to-end system communications. The upper 3 levels, however, have not been defined to anywhere near the same degree, but are intended to provide for the encoding rules, data formats etc necessary to ensure usable information transfer between application processes. GM has been concentrating on level 7, in order that user program interfaces may be agreed, thus allowing independent software producers a fixed point of reference when writing packages designed for MAP implementation.

The FIT LAN system

Having illustrated a little of the background, and introduced MAP and TOP, it is possible to put the LAN being developed at Footscray Institute of Technology in true perspective. The FIT LAN system should not be seen as a universal replacement for MAP, rather it should be considered more in the light of augmenting MAP by allowing networks to be implemented in areas which might otherwise be considered too hostile. By hostile I refer to electrical hostility such as noise or interference. A great deal of work has gone into the analysis of electrical noise and many factories could be classified as very noisy. Prime culprits are heavy rotating machinery, arc welders etc, the effects of which could

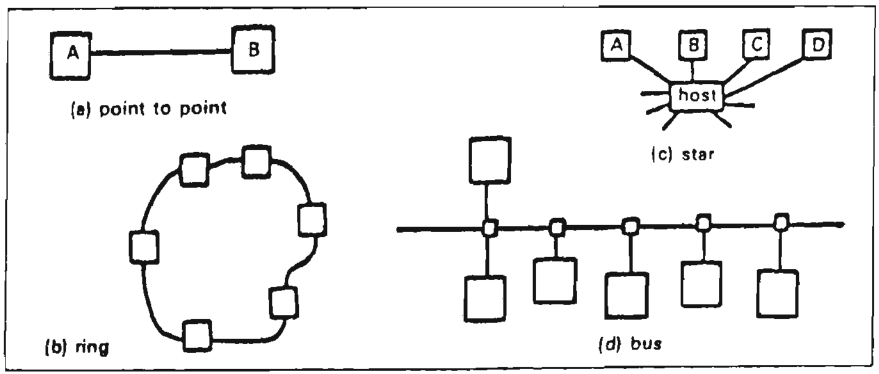


Figure 4: Network topologies

	MAP	TOP
contention handling	token passing	CSMA/CD
topology	bus	bus
bandwidth	broadband	baseband
transmission medium	co-axial cable	co-axial cable
IEEE Standard	802.4	802.3

Table 1: Major MAP and TOP features

Alex Simcock graduated from the University of Kent at Canterbury with the BSc(Hons) (electronics) in 1972. He worked for 12 years with Procurement Executive of the Ministry of Defence in Britain, researching in the following areas: digital circuit design, digital data switching, computer-controlled communication systems and microprocessor replacement of digital circuits. The work also involved the implementation of software project management techniques. He was appointed to a lectureship with the department of electrical and electronic engineering, Footscray Institute of Technology, in 1983.

Chula na Ranong received the BE(Hons) degree in electrical engineering and PhD from the University of New South Wales in 1970 and 1973 respectively. Since 1972 he has been with the department of electrical and electronic engineering, Footscray Institute of Technology, as a lecturer and later, senior lecturer in digital electronics and systems. During 1977-78, he spent a year as a visiting professor, school of electrical engineering, Cornell University, Ithaca, New York, US. Then, for the first six months of 1982, he worked as a specialist engineer with Compagnie Europeene de Teletransmission (Thomson-CSF), Chatou, France, developing a mass storage system for a real-time digital data network supervisory control system. More recently, as a result of his interest in automation technology, Chula undertook a two-month study visit to Japanese government, industrial and educational research institutions in 1984.

Patrick Leung received the BE(Elect) and MEngSc in electrical engineering from the University of Melbourne in 1972 and 1974 respectively. Between 1975 and 1977 he was a lecturer in the department of electrical and electronic engineering, Footscray Institute of Technology. In 1977 he joined Telecom Australia as an engineer in the Datel design and provisioning section. In 1979 he rejoined FIT as a lecturer in communications engineering. In the first half of 1985 he visited the department of electrical engineering, Ottawa University, as a research fellow, to carry out work in data communications over satellite channels. On his return to FIT he was appointed senior lecturer in communications. His main interests are data transmission techniques, data networks and data protocols.

	User program	Application programs (not part of the OSI model)
Interconnection	Application	Manages lower-layer services, including application programs
	Presentation	Restructures data to and from standardised format used within the network
	Session	Name, address, translation access security, and synchronises & manages data
Interoperation	Transport	Provides transparent, reliable data transfer from end device to end device
	Network	Establishes connections between equipment on the network
	Data link	Establishes, maintains and releases data links
	Physical	Encodes and physically transfers messages between adjacent devices

Figure 5: Open systems interconnection (OSI) seven-layer model

create havoc with computer communications. A whole sub-culture has grown up around the design of equipment and transmission systems which are resistant to electromagnetic interference (EMI), and/or producing equipment which transmits a minimum of interfering signals, in other words designing equipment for electromagnetic compatibility (EMC).

Most of the detailed EMC research has been carried out in the military field and it is this same environment which spawned the techniques upon which our system is based. These techniques, called **spread spectrum** have been defined as "a communication system which transmits data using a much greater bandwidth than the minimum necessary". This might seem a little strange at first but it works like this. Each data bit, see figure 6a, (at a frequency f_d) is spread across a much wider spectral bandwidth by multiplying it with a fast pseudo-random (commonly called the chipping sequence) running at a frequency of f_c, figure 6b. At the receiver the resultant signal is re-multiplied by the same chipping sequence thus recovering the original data by collapsing the signal back into the bandwidth of f_d, ie that of the data signal. This act of multiplication at the receiver has the effect of spreading any interference signal over the chip sequence bandwidth, thus allowing the data to be recovered by a filtering process.

Figure 7 illustrates the concept of noise rejection in a spread-spectrum system. If the data was transmitted in its original form then a noise spike could completely swamp the data signal, as shown in figure 7a, thus causing an error. If this same noise spike were to interfere with the spread-data signal, however, then only a small portion of the information content would be lost, as shown in figure 7b, and it would be possible to recover the original data signal without error. In this manner the act of spreading the data for transmission imparts inherent interference rejection capabilities to the transmission. This type of system is, therefore, particularly suitable when error-free communication in a hostile environment is the primary objective.

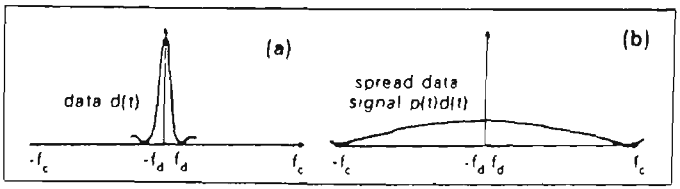


Figure 6: Data and "spread data" frequency characteristics

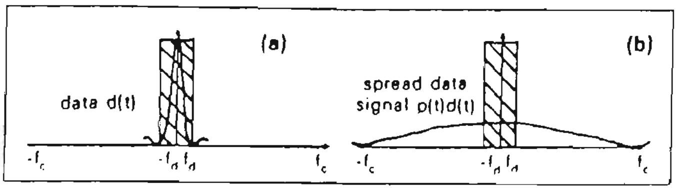


Figure 7: Noise rejection illustration

Multi-user system

The FIT system is connected using a bus topology, as illustrated in figure 8. Each node in the network is allocated a unique code for its chipping sequence, which acts as an address for that node. Thus if node 2 wishes to transmit to node 5 then it should encode its data for transmission with the chipping sequence for node 5. By carefully selecting the codes which are used for the chipping sequences it is possible to have many messages being transmitted simultaneously, with each transmission causing little or no interference to the others. This system is called code division multiple access (CDMA).

The more traditional methods of multi-access ie time division multiple access (TDMA) and frequency division multiple access (FDMA) are, it is true, more efficient in their use of bandwidth. These mechanisms operate with fixed time or frequency slots, however, and do not offer the same **random access addressing capabilities** that are offered by CDMA, neither do they have CDMA's inherent interference rejection capabilities.

Prototype system tests

A simple prototype system has already been built, with three transmitters and two receivers. With this prototype system we have been able to demonstrate the viability of simultaneous multiple transmissions. Tests have also been carried out to prove the error rejection capabilities, and preliminary results indicate that it is possible to recover data from the transmission medium even when the signal transmitted is subjected to interference by sources of 20 times the power. These preliminary results clearly demonstrate the potential which the FIT system has to carry computerised manufacturing information in a hostile environment.

Conclusions

The MAP concept is not fully mature, and the present position may well define only a subset of future systems. This is precisely where MAP illustrates one of its major features; even if MAP is expanded, present systems will not be obsolete as they will be **fully compatible** subsets of the expanded system. Thus users need not have a daunting initial capital outlay in order to use a MAP system. They can buy for their present needs and rest assured that they will be able to update as their needs develop, and that they will be able to select MAP components from many manufacturers, thereby excluding obsolescence and allowing the selection of the most suitable item for the job. All of

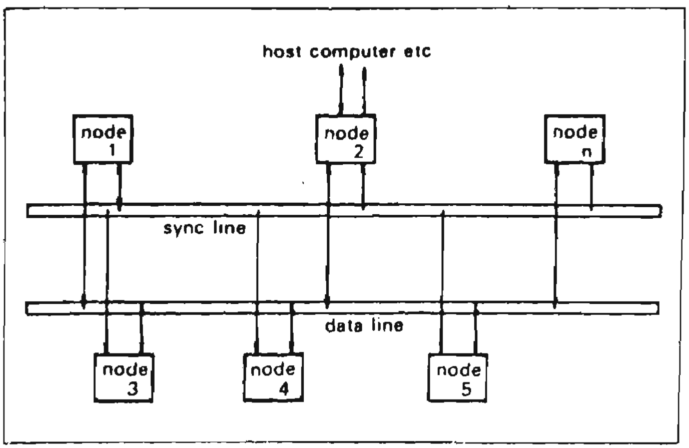


Figure 8: Bus-connected system

these points guarantee that MAP will stay and develop in importance. More and more will be written about computer integrated manufacturing and MAP in the near future, in fact it could be said that this **Is** the future for Australian industry. This includes both components of industry, the system users such as car manufacturers etc, since it won't be too long before almost all manufacturing equipment purchased will have a MAP interface, and also the machine tool manufacturers, since it will be impossible to sell such tools unless they are supplied with a MAP interface. In fact Boeing has already issued a policy statement indicating that preference will be given to suppliers who have TOP systems. The future is not all that far away as a target window of 1987 has been identified. Before this date producing MAP-compatible equipment may prove too costly, but leaving it too long after this date could mean that Australian manufacturers may well miss the boat.

We believe the system which we are designing could complement the MAP system, thus ensuring hostile environment operation, and in doing so may prove a valuable tool in generating a manufacturing-led recovery to extract Australia from the current depression.

Acknowledgements

This research project is being supported by a grant from the Australian Telecommunications and Electronics Research Board.

References

Gold R. (1967), "Optimal binary sequences for spread spectrum multiplexing", IEE Transactions on Information Theory, Vol IT-13, pp619-621.

Hirosaki B., Hasegawa S., Sawai A. (1983), "Spread spectrum multiple access data loop", NEC Res Dev n71, October pp48-58.

Holmes J.K. (1982), "Coherent spread spectrum systems", Wiley (New York).

Pickholtz R.L., Schilling D.L., Milstein L.B. (1982), "Theory of spread spectrum communications – a tutorial", IEEE Transactions on Communications, Vol Com-30, May.

Scholtz R.A. (1982), "The origins of spread spectrum communications", IEEE Transactions on Communications, Vol Com-30 May.

Watt D.G. (1985), "Networking software support for FMS", Manufacturing Engineering, September, pp103-106.

Weber D.M. (1985), "Technology update: manufacturing – more versatile tools speed factory automation", Electronics, October, pp52-55.

There's not one single, new announcement ASSCO can make about improving engineering productivity.

There's two.

Hewlett Packard's latest drafting plotters – Draft Master I and II.

These exciting new HP drafting plotters offer 50% more throughput than the highly respected HP7580. And, if that's not improved productivity, what is? HP Draft Master has full AO capacity, is variable through A4, yet can be yours for only three quarters of the HP7585 price.

To reserve yours, call now.

Hewlett Packard's latest Engineering Calculator – HP28C.

More advanced, more functions and more useful than ever before. The new HP28C offers algebraic and RPN solutions, full-word variables together with integration and differentiation, and much more. Available ex-stock. Mail orders most welcome.

Long-term productivity and performance for professionals.

ASSCO

17 Heatherdale Road, Ringwood 3134
Phone (03)873 2266
Fax (03)874 1393

Sextant 165

Circle 2 on reader service coupon

Spread Spectrum Techniques For LAN Interconnections

A. L. Simcock, C. na Ranong & P. Leung

1. Introduction

This is an age of austerity, at least as far as the allocation of transmission bandwidth is concerned. Ever increasing demands on an already overcrowded electromagnetic spectrum have led to developments in two areas:

a) Extending the usable spectrum through the use of even higher frequency carriers.

b) The design of complex frequency allocation algorithms.

Spread spectrum (SS) techniques were originally developed in the mid 1950s for military applications , but it is only relatively recently that the techniques have begun to be exploited in a commercial environment. Areas of commercial interest include mobile radio, radio telephony, amateur radio and satellite communications.

SS systems may be defined as those which transmit using a frequency allocation in excess of the minimum required, and as such do not rely on developments in either of the above fields, rather on the use and re-use of the same bandwidth. One of the properties of a SS system is that several nodes may simultaneously transmit messages, occupying the same bandwidth, with minimal interference between transmissions.

The Department of Electrical and Electronic Engineering at Footscray Institute of Technology is currently investigating SS techniques and their application in Local Area Networks.

2. Basic Spread Spectrum Fundamentals

The three primary methods of Spreading the data transmission are :

a) Direct sequence - in which the data is directly modulated with a fast pseudo random code sequence.

b) Frequency hopping - in which the carrier which is modulating the data is caused to hop from one frequency to another in a pseudo random manner.

c) Time hopping - which can be likened to the pseudo random allocation of channels in a time division multiplexed system.

The Direct Sequence method appears to be the most applicable in the LAN environment, and has been the one selected for further investigation. The direct spreading sequence is sometimes also called the "CHIP SEQUENCE" .

Figure 1 illustrates a simple single channel direct sequence system.

4. System Under Investigation

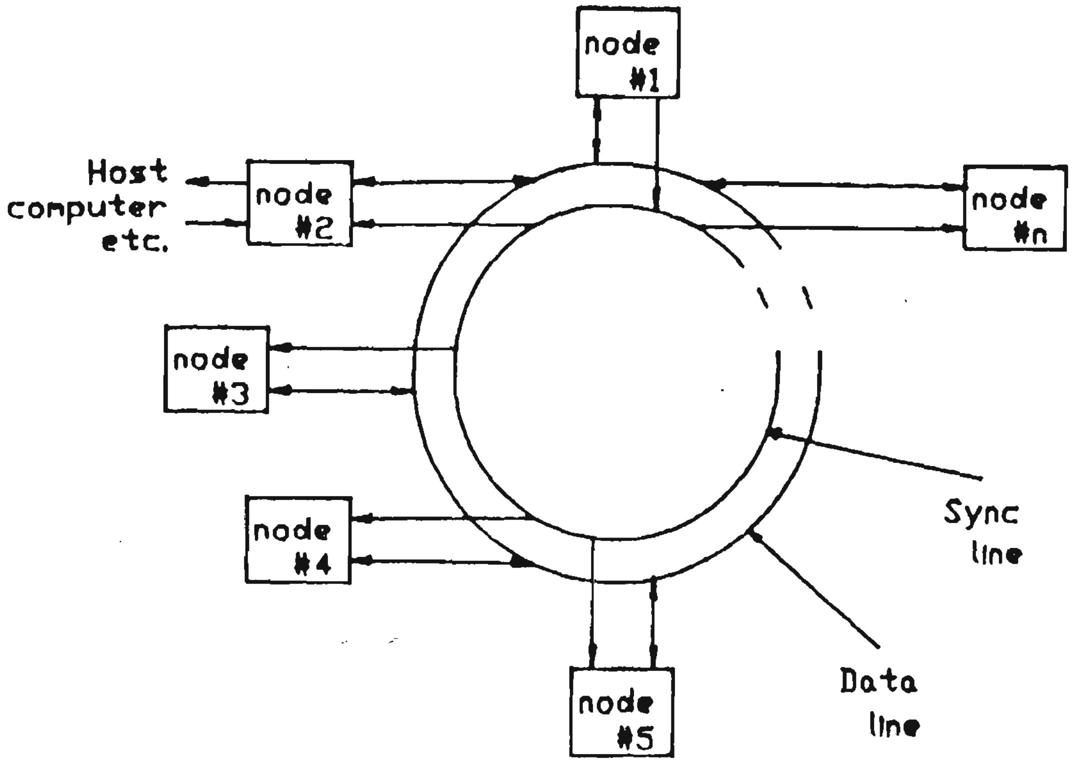


Figure 2 Bus Structured Node Interconnections

(NOTE The seperate Sync line is included to allow synchronisation and clock recovery.)

Figure 2 illustrates the bus structured LAN topology currently being investigated. Being a bus structured system each node can transmit and receive independently of the others. This is achieved by using a Code Division Multiple Access (CDMA) system where each node has its own 'address' and associated code (or chip sequence). Thus, with the point to point communications adopted in our system, if node #4 wishes to transmit to node #2 then node #4 will modulate its data with the code of node #2. The code allocated to each node is a delayed version of the synchronosation code. The delay between allocated codes being such that no code plus propagation delay (upto the maximum propagation delay in the system) can immitate any other allocated code.

It is obvious, however, that only one node at any instant in time may be transmitting to a particular address. This necessitates a collision check, as in Ethernet, but only for transmissions to the same receiver. Each node has two demodulators - one to monitor the system for its own data and one to pre-check for collisions and continuously check whilst transmitting. The first demodulator works exclusively with its own address code whereas the code used in the second is switchable. Each node is also equipped with one transmitter, which must be able to modulate the data with any recipient's address code.

The pre-transmission monitoring requires each node to check over a time window, equal to the maximum propagation delay, to ensure that no transmission is currently in progress to the desired recipient. Figure 3 illustrates this scanning, across a delay window equal to two chip bit periods, to find a transmission. Continuous monitoring for collisions is also carried out throughout the transmission, and Figure 4 illustrates the effects of simultaneous transmissions by two nodes to the same recipient with various propagation delays between the interfereing signals.

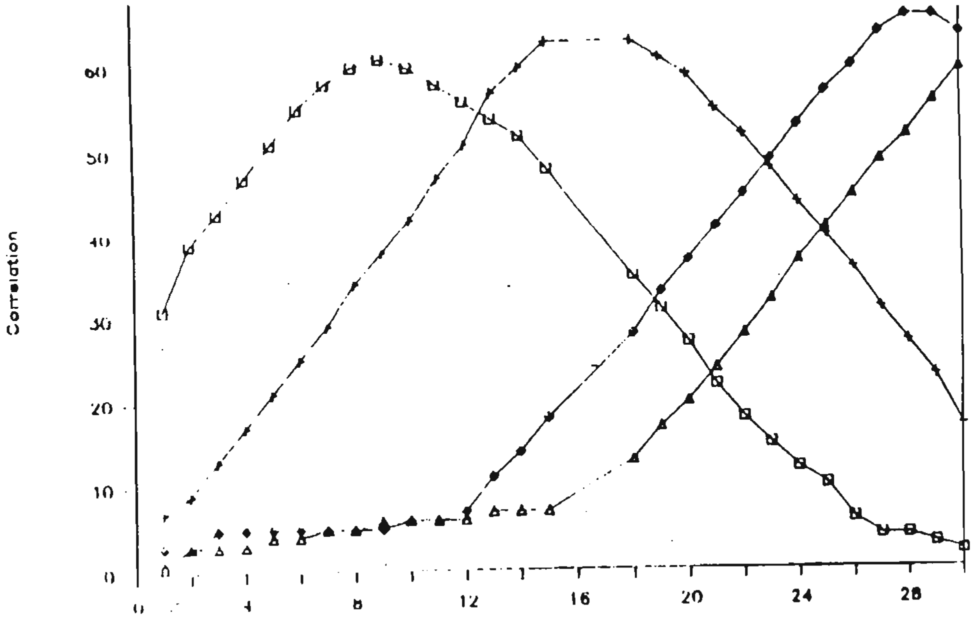


Figure 3 Pre-Transmission Scan

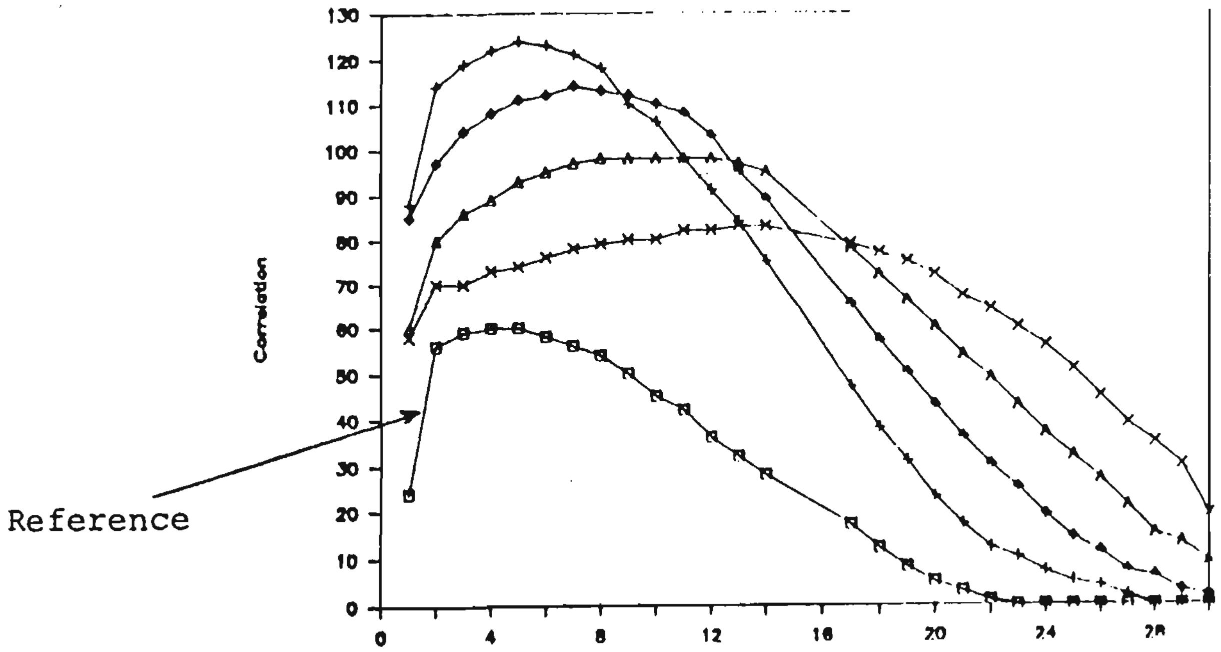


Figure 4 Correlation Output With Colliding Signals

5. Conclusions

The SS LAN being investigated at FIT has been developed to the point where the code type, code allocation, synchronisation and collision detection aspects have all been considered. The future workplan involves the investigation protocols suitable for the control of a system of this nature.

6. Acknowledgements

This research projeet is being supported by a grant from the Australian Telecommunications and Electronics Research Board.

References

1. ROBERT A. SCHOLTZ, "The Origins of Spread Spectrum Communications", I.E.E.E. Transactions on Communications, Vol. Com-30, May 1982, pp 822-850.

2. C. S. SMYTHE and C.T. SPRACKLEN, "A High Speed Local Area Network Using Spread-spectrum Code-division-multiple-access Techniques", The Radio and Electronic Engineer, Vol. 54, May 1984, pp.225-226.

3. R. C. DIXON, "Spread Spectrum Systems", Wiley (New York), 1976.

4. J. K. HOLMES, "Coherent Spread Spectrum Systems", Wiley (New York) 1982.

5. R. L. PICKHOLTZ, D. L. SCHILLING, L. B. MILSTEIN, "Theory of Spread Spectrum Communications - A Tutorial", I.E.E.E. Transactions on Communications, Vol. COM-30, May 1982, pp 855-884.

6. D.V.SARWATE and M. B. PURSLEY, "Cross Correlation Properties of Pseudo-Random and Related Sequences", Proc. I.E.E.E. Vol. 69, May 1980, pp.593-619.

7. R. GOLD, "Optimal Binary Sequences For Spread Spectrum Multiplexing", .I.E.E.,Transactions on Information Theory,Vol.IT-13, 1967, pp.619-621.

8. A. BRUCE CARLSON, "Communication Systems -An Introduction to Signals and Noise in Electrical Communication", McGraw-Hill, (Tokyo), 1981.

9. M. B. PURSLEY and D. P. SARAWATE, "Performance Evaluation for Phase Coded Spread Spectrum Multiple Access Communications - Part II", I.E.E.E. Transactions on Communications Vol. COM-25, August 1977, pp.800-803.

10. B. HIROSAKI, S. HASEGAWA, A. SAWAI, "Spread Spectrum Multiple Access Data Loop", NEC Res DEV n71, October 1983, pp.48-58.

A Spread Spectrum Manufacturing Information Network

C. NA RANONG
Centre for Automation Technology, Department of Electrical and Electronic Engineering, Footscray Institute of Technology

A.L. SIMCOCK
Lecturer, Department of Electrical & Electronic Engineering, Footscray Institute of Technology

P. LEUNG
Senior Lecturer, Department of Electrical & Electronic Engineering, Footscray Institute of Technology

SUMMARY An error free and reliable data network is vital for real time control of an automated manufacturing process. Special shielding or optical fibre can be used to protect the network against a plant's hostile electrical environment.

This paper proposes the adaptation of Spread Spectrum techniques, which were originally developed for military use as a means of providing a jam resistant communication link, to a manufacturing process network, to provide an error resistant, low cost network. An increased network throughput is also achieveable, through the maintenance of network availability in a muiltiple access high noise environment.

1. INTRODUCTION

In a Flexible Manufacturing System, FMS, parts production and material handling are achieved by automated workcells. The cells receive their work orders either from a supervisory console or from a central computer. Parts programs may also be received from remote Computer Aided Design, CAD, systems. The results of each work order are returned to the requester when the work is completed.

Ultimately the goal of factory automation is to tie a number of different systems, or islands of automation, into a single manufacturing process. To manufacture a random parts mixture in a cell, the parts programs that are to be used by each device controller must be downloaded to each of them for the particular part or parts to be made. This implies communication between the control system and the device controllers. Such communication interfaces must deal with a variety of machine tools, robots, vision and measuring systems etc. controlled by a variety of CNCs, PLCs and robot controllers. This leads to the need to specify standards that device controllers can use to communicate with each other and with the supervisory computers. The leading proponent being the General Motors Corporation's Manufacturing Automation Protocol, MAP.

Compliance with GM's MAP specifications, means that machine tools, robots, programmable controllers and computers must be able to communicate through an IEEE-802.4 token passing protocol linked with an Ethernet broadband Local Area Network (LAN). Data transmission occurs at 10 M bits per second which is adequate both for real time control at the machine tool level and for detailed data collection at the supervisory level.

This paper proposes the adaptation of Spread Spectrum techniques as a means of implementing a low cost, noise resistant communication network for a factory automation communication link such as MAP.

Spread spectrum (SS) techniques were originally developed in the mid 1950s for military applications as a means of providing a jam resistant communication link. The major attributes of spread spectrum signalling are its high noise immunity characteristics and the ability to sustain multiple or simultaneous access within its link. These properties have only relatively recently begun to be exploited in a commercial environment such as mobile radio, radio telephony, amateur radio and satellite communications.

The Centre of Automation Technology, Department of Electrical and Electronic Engineering of the Footscray Institute of Technology is in the process of developing a Local Area Network based on spread spectrum techniques to implement point to point communications. The primary objective of the research is to develop a system which is simple to implement, reliable, resilient and flexible.

2. MANUFACTURING PROCESS COMMUNICATION LINK

The definition of what comprises a manufacturing cell varies widely across the industry. Here, a cell is assumed to be a collection of 2 to 20 machine tools, a transportation system and possibly inspection systems (including vision). The operation of the cell is controlled by the cell management software and the machine dependent device controller. Cell management software performs the task such as part program management, production control, resource management, equipment monitoring etc., while the device controller includes the hardware and software required to control and monitor the individual device in the cell.

The architecture of the communication link dictates the flexibility that the manufacturing systems may achieve (Watt, 1985; Weber, 1985). There are three major approaches: a centralised system, a network and a totally distributed system; with the intelligence being progressively driven down to the lowest level. In the case of a centralised system, the hardware communication connections required may be complex and diverse. In the factory, there is often electrical noise which makes the use of long RS232C or digital input/output (I/O) lines undesirable and special cabling or optical fibre are used. With suitable software, there is a high degree of flexibility in the centralised system; however, changes or additions of hardware configuration may involve costly retrofit of communication ports. The last two approaches, which provide higher overall system flexibility, utilise the development of LANs which provide high speed, error free transmission and microprocessors to distribute the control system. The protocol section of the machine dependent software is contained in Network Interface Modules, which connect to the network. The point of connection is known as a Node and a network interface module is required for each node. Owing to the bewildering array of communication interfaces from the different vendors there is a growing pressure for standardisation in this area as the present factory automation system requires customised software and hardware to cope with the problem of non standard interfaces.

General Motors Corporation is promoting its Manufacturing Automation Protocol as a standard. (A prototype system was exhibited in Detroit last November.) In the proposed MAP local area network system, machine tools, robots and programmable controllers will be able to communicate with each other and with the supervisory computers through an IEEE 802.4 token passing protocol linked with a 10 M bits per second Ethernet broadband local area network. The protocol is based on the International Organisation for Standardisation (ISO) seven layer Open Systems Interconnection (OSI) model shown in Figure 1.

Level	7	Application	(Highest)
	6	Presentation	
	5	Session	
	4	Transport	
	3	Network	
	2	Data Link	
	1	Physical	(Lowest)

Figure 1 : ISO OSI Reference Model.

In brief, a device controller interfaces to the communication network via the Physical and Data Link layers. (The network communication medium is also part of the Physical layer.) The Network layer is required for interconnections between more than one LAN. The next two levels, Transport and Session layers, maintain and identify all the nodes (or processes) connected to the network and facilitate the data communication necessary for the control and monitoring function of a manufacturing system. The upper two levels, Presentation and Application (or User), represent the methods by which the control and monitoring functions are achieved. These include message formats and their contents, graphic displays, encryption etc.

The proposed spread spectrum implementation deals with the two lowest level : the Physical and Data Link layers.

3. SPREAD SPECTRUM COMMUNICATIONS

Spread spectrum (SS) techniques were originally developed in the mid 1950s for military applications (Scholtz, 1982), but it is only relatively recently that the techniques have begun to be exploited in a commercial environment. Spread Spectrum systems may be defined as those which transmit using a frequency allocation in excess of the minimum required and as such, have the capabilities of using and re-using the same bandwidth. The three primary methods of Spreading the data transmission are :

1. Direct sequence - in which the data is directly modulated with a fast pseudo random code sequence.

2. Frequency hopping - in which the carrier which is modulating the data is caused to shift frequency in a pseudo random manner.

3. Time hopping - which can be likened to the pseudo random allocation of channels in a time division multiplexed system.

The proposed system uses the Direct Sequence method (Holmes, 1982; Pickholtz et al, 1982) and therefore discussion will be restricted to this technique. Conceptually, each data bit is transmitted as a series of (narrower) data bits and as a consequence, the required transmission bandwidth increases. An appropriate selection of the data bit coding sequences gives rise to the interference proof properties of spread spectrum systems. The direct spreading sequence is sometimes also called the CHIP SEQUENCE.

Figures 2 and 3 illustrate a simple single channel direct sequence system and typical waveforms.

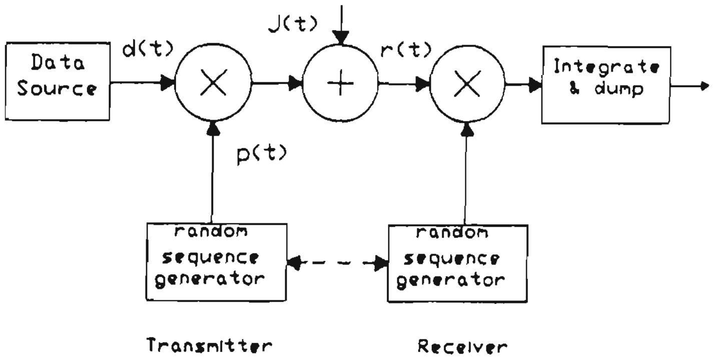


Figure 2 : Single channel direct sequence system.

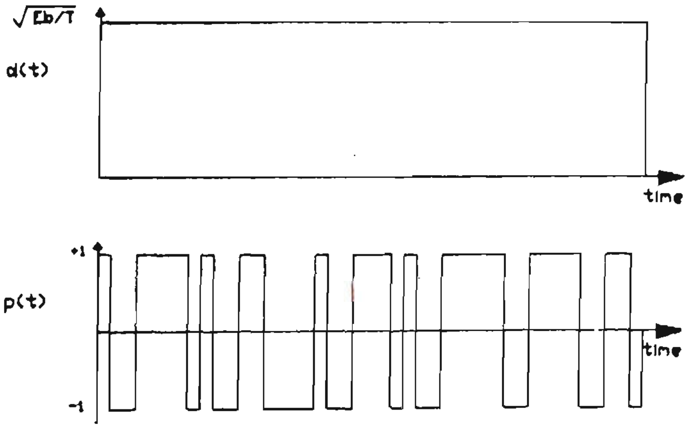


Figure 3 : Data and chip waveforms

Each data bit d(t), of energy E_b and duration T may be represented by

$$d(t) = \pm\sqrt{\frac{E_b}{T}}$$

This one dimensional signal is multiplied by a binary ±1 chipping sequence p(t) running at a frequency of f_c chips/sec, ie. a total of f_cT chip bits per data bit. The resultant sequence d(t)p(t) is then a $n = f_cT$ dimensional signal. If this signal is transmitted in the presence of a jamming signal J(t), the received signal would be

$$r(t) = d(t)p(t) + J(t) \quad 0 \leqslant t \leqslant T \qquad (1)$$

At the receiver this signal is multiplied by the same chipping sequence p(t) that was used to encode the transmitted data. The output of the multiplier is integrated over a period T to produce a decision variable U, where

$$U = \sqrt{\frac{E_b}{T}} \int_0^T r(t)p(t)\,dt \qquad (2)$$

from which the decision can be made as to whether $+\sqrt{E_b/T}$ or $-\sqrt{E_b/T}$ was sent depending on whether Uis greater than or less than zero.

The integrand in equation (2) can be expanded as follows

$$r(t)p(t) = d(t)p^2(t) + J(t)\ p(t) \qquad (3)$$

Since the chipping sequence p(t) is a binary ± 1 signal the square of this signal $p^2(t) = 1$ and therefore equation (3) may then be rewritten

$$r(t)p(t) = d(t) + J(t)\ p(t) \qquad (4)$$

It can be seen from this that the receiver recovers the one dimensional data signal and that this recovery process actually spreads the jamming signal energy over a much wider bandwidth, thus reducing the energy per Hz of the jamming signal.

The signal J(t) in equation (4) may be deliberate jamming or it may be interference from other transmitting nodes or electrical noise. The rejection of the other nodes' transmissions as noise depends upon the orthogonality of the codes used for spreading each node's data. That is, an appropriate selection of a set of chipping sequence enables several nodes to simultaneously transmit messages in a spread spectrum system, with almost zero interference between transmissions.

The primary aims of the project were to produce a system which is simple, reliable and resilient. The task of choosing codes for their orthogonality was simplified after realising that it was possible to control the environment in which the system is to work, with respect to connection distance and hence propagation delay, attenuation etc., that it should be possible to produce a synchronous network. The choice of a synchronous network would then mean that the so called odd-even cross correlation properties could be avoided, and that it should not be necessary to resort to codes with multi-level cross correlation properties, e.g. Gold or Kasami codes (Gold, 1967).

This being the case it was decided to allocate codes to each node such that each address code would be a phase shifted version of the same code, phase synchronisation and clock recovery being facilitated by the provision of a separate synchronisation line (Hirosaki et al, 1983), thus the cross correlation between codes is in fact the autocorrelation function of the one code.

3.1 Channel Re-use Capability

Figure 4 illustrates a possible bus structured topology employed to interconnect system nodes. Being a bus structured system each node can transmit and receive independently of the others.

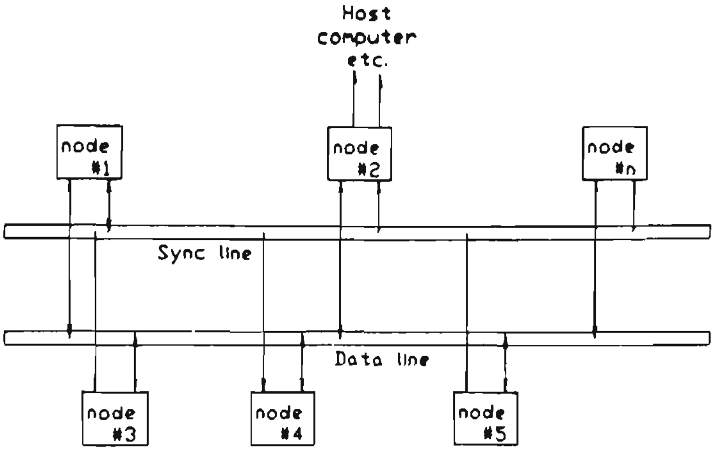


Figure 4 : Bus Structured Node Interconnections

The proposed communication link uses a Code Division Multiple Access system where each node has its own 'address' and associated code (or spreading sequence). Thus with the point to point communications adopted in our system if node #4 wishes transmit to to node #2 then node #4 will modulate its data with the code of node #2. Each node monitors the line looking for data modulated with its own 'address code', and should recognise data for any other intended recipient as noise and reject it, as shown in Equation (4), but should be able to demodulate the data encoded with its own address code in order to recover the original data.

It is obvious, however, that only one node at any instant in time may be transmitting to a particular address. This necessitates a collision check, as in Ethernet, but only for transmissions to the same receiver. Each node has two demodulators - one to monitor the system for its own data and one to pre-check for collisions and continuously check whilst transmitting. The first demodulator works exclusively with its own address code whereas the code used in the second is switchable. Each node is also equipped with one transmitter, which must be able to modulate the data with any recipient's address code.

Selection of the number of 'chip' bits to represent a data waveform determines the Processing Gain of the system, and hence its performance in presence of noise; and this in turns determines the maximum number of simultaneous communications which the system can sustain before its performance (eg. error rate) deteriorates rapidly. However, because of the channel re-use capability of the spread spectrum system, the actual number of channels available to the overall system is greater than the number of simultaneous channels that the system can sustain. For example, in the presence of severe jamming or electrical noise, a 32 channel spread spectrum system may be able to carry 10 channels' simultaneous transmissions. Having more signalling channels than could be simultaneously carried by the spread spectrum system has negligible effect on the system throughput performance since each data transmission tends to be short and very rarely would all the 32 channels be used at the same time.

3.2 Noise Resistance

Figure 5 illustrates a simple system layout, with a number of transmitting nodes and a single receiving node shown. The output from each node is added onto the line to produce a received waveform r(t) which is a summation of the outputs from each transmitter, as shown in Figure 6.

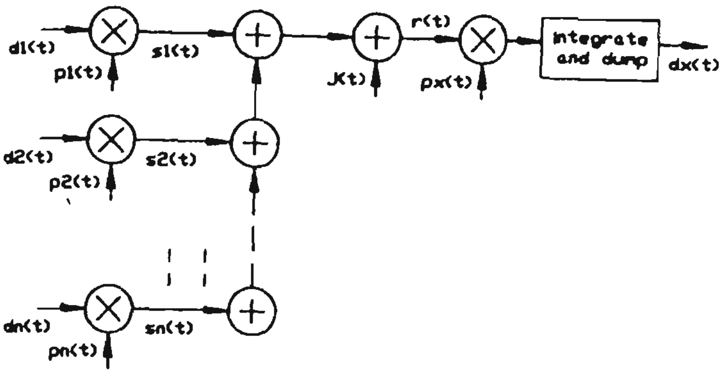


Figure 5. Simple System Layout

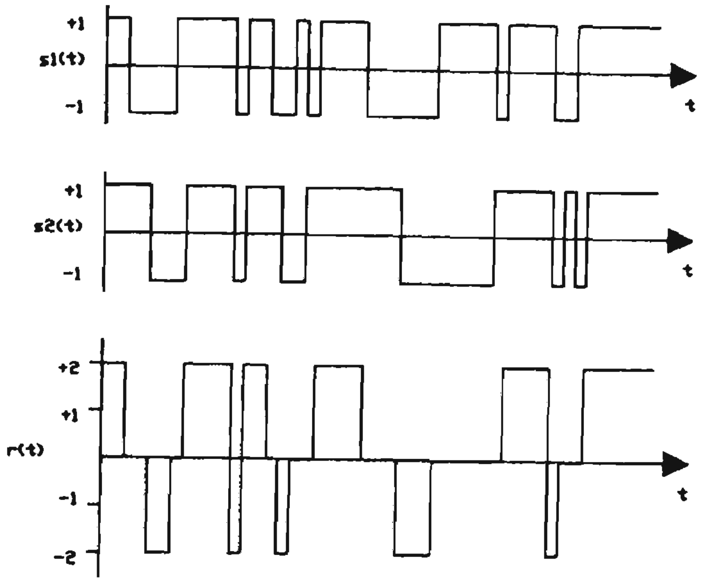


Figure 6. Typical transmitter waveforms

Figure 5 shows a number (n) of transmitting nodes and a single receiver (node #x), where $1 \leqslant x \leqslant n$ The output of each transmitter $s_n(t)$ is given by

$$s_n(t) = p_n(t)\ d_n(t) \qquad (5)$$

And, in the absence of external interference ie. J(t) = 0, the received signal is given by

$$r(t) = s_1(t) + s_2(t) + s_3(t) + \ldots\ldots s_n(t) \qquad (6)$$

$$= \sum_{n=1}^{n} s_n(t) \qquad (7)$$

The demodulation is carried out and from equation (4)

$$r(t)\ p_x(t) = d_x(t) + s_n(t)p_x(t) + s_n(t)p_x(t) \qquad (8)$$

where $1 < x < n$

In the presence of an external noise input J(t) the received signal becomes

$$r(t) = s_n(t) + J(t) \qquad (9)$$

which is demodulated to

$$r(t)p_x(t) = d_x(t) + \sum_{n=1}^{x-1} s_n(t)p_x(t) + \sum_{n=x+1}^{n} s_n(t)p_x(t) + J(t)p_x(t) \qquad (10)$$

The upper trace in Figure 7 shows of a portion of received waveform with three transmitters and Added White Gaussian Noise, the lower trace shows the data recovered from this signal. A plot of Bit Error Rate against Signal to Noise ratio is given in Figure 8.

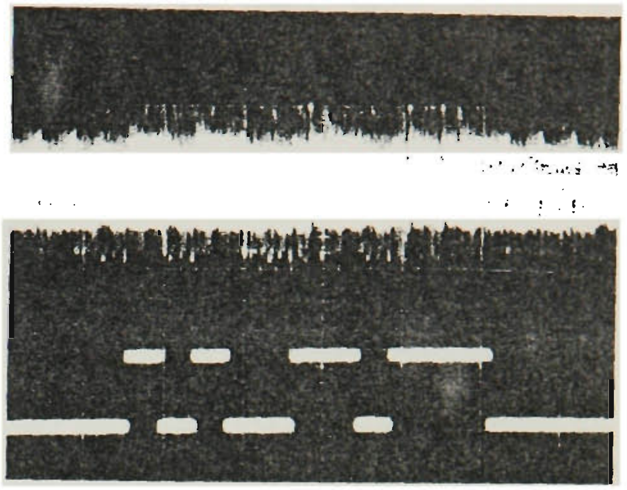


Figure 7 : Rx waveform and recovered data.

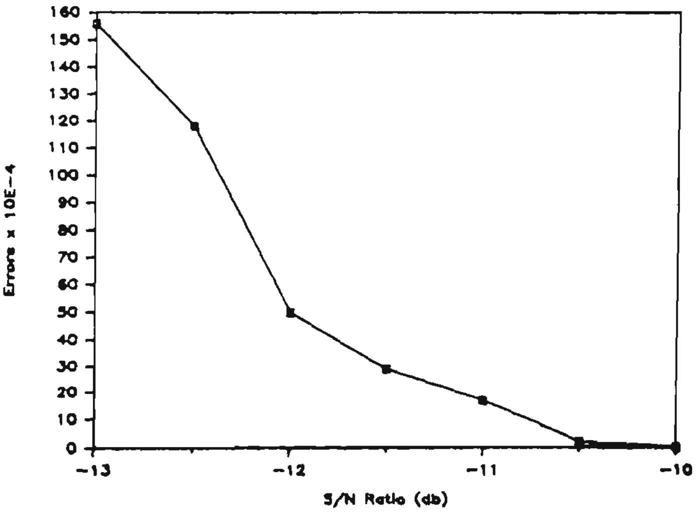


Figure 8 : BER vs. S/N Ratio

4. APPLICATION IN A MANUFACTURING NETWORK

The Manufacturing Automation Protocol revision 2.1 which is an emerging factory networking scheme promoted by General Motors Corporation, proposes a broadband factory wide communication network which allows modulated signals to be send using numerous channels or carrier frequencies i.e. the total frequency spectrum of the network is subdivided into channel by frequency allocation scheme.

By using direct sequence code allocation scheme outlined above, a spread spectrum system could be substituted for the broadband network, with the advantages of channel re-use capability and high resistance to electrical noise interference. The application of spread spectrum signalling is not restricted to a network system such as MAP, but it can be used in any communication link where the available link bandwidth is larger than the data communication bandwidth.

5. CONCLUSIONS

By applying Spread Spectrum techniques to the design of a manufacturing information network, a simple error resistant network is being developed.

when the interference levels are many times greater than the signal levels. This feature together with the ease of connection (via twisted pair cable) makes this type of system eminently suitable for carrying computer communications within manufacturing plants.

6. ACKNOWLEDGEMENTS

This research project is being supported by a grant from the Australian Telecommunications and Electronics Research Board.

7. REFERENCES

GOLD R.(1967), "Optimal Binary Sequences For Spread Spectrum Multiplexing", I.E.E., Transactions on Information Theory, Vol. IT-13, pp.619-621

HIROSAKI B., HASEGAWA S., SAWAI A.(1983), "Spread Spectrum Multiple Access Data Loop", NEC Res DEV n71, October. pp.48-58

HOLMES J.K. (1982), "Coherent Spread Spectrum Systems", Wiley (New York)

PICKHOLTZ R.L., SCHILLING D.L., MILSTEIN L.B. (1982), "Theory of Spread Spectrum Communications - A Tutorial", I.E.E.E. Transactions on Communications, Vol. COM-30, May

SCHOLTZ R.A. (1982), "The Origins of Spread Spectrum Communications", I.E.E.E. Transactions on Communications, Vol. Com-30 May

WATT D.G. (1985), "Networking Software Support for FMS" , Manufacturing Engineering, September, pp. 103-106

WEBER D.M. (1985), "Technology Update : Manufacturing - More Versatile Tools Speed Factory Automation" , Electronics, October, pp. 52-55

A Localised Spread Spectrum Network

A.L. Simock*, C. na Ranong* and P. Leung*

SUMMARY The need for many users to be able to communicate within a localised environment gave rise to the Local Area Network (LAN). This paper proposes the adaptation of Spread Spectrum techniques, which were originally developed for military use as a means of providing a jam resistant communication link, to a LAN, the objective being to develop a system capable of maintaining communications in a muiltiple access high noise environment.

1 INTRODUCTION

The expanding need for data communication channels in Local Area Networks (LANs) has increased the pressure on the available transmission bandwidth. This has led to developments in two broad areas : the extension of the usable spectrum through the use of even higher frequency carriers and the development of complex networking and multiple access algorithms. An extension to the latter has led to the adaptation of Spread Spectrum CDMA techniques, which were originally developed in the mid 1950s [1] for military use as a means of providing a jam resistant communication link.

Spread Spectrum communication involves a transmission system which uses an overall frequency bandwidth far in excess of the minimum required. Through a correlation process the receiver restores the signal to its normal bandwidth and in doing so, it enhances the desired signal while substantially reducing the effects of interference and noise. Overall, the communication channel capacity is not 'wasted' because with a suitable selection of spreading codes (having low cross correlation properties) several nodes may simultaneously transmit messages over the same channel, with low, predictable, interference between nodes.

A Local Area Network such as Ethernet, where multiple nodes use a single communication channel - one pair at a time, is a prime candidate for the application of Spread Spectrum techniques. Being inherently jam-resistant, Spread Spectrum techniques have the potential of reducing the contention and allowing multi-pair communications, thus increasing the network's throughput under these conditions. CDMA is not as efficient as Frequency Division Multiple Access (FDMA) or Time Division Multiple Access (TDMA) in terms of throughput per unit bandwidth, but it does have the advantage that random access may be achieved.

This paper deals with the development of A Localised Spread Spectrum Network, ALSSNET, by the Department of Electrical and Electronic Engineering, Footscray Institute of Technology. Code Division Multiple Access (CDMA) [2] is used to allow 'multi channel' access, and Carrier Sense Multiple Access with Collision Detection (CSMA/CD) is used to resolve any remaining contention problems within channels.

2 SPREAD SPECTRUM ASPECTS

The three primary methods of Spreading the data transmission are [3], [4], [5] :

a. Direct sequence - in which the data is directly modulated with a fast pseudo random code sequence.

b. Frequency hopping - in which the modulating carrier is caused to hop from one frequency to another in a pseudo random manner.

c. Time hopping - which can be likened to the pseudo random allocation of channels in a time division multiplexed system.

ALSSNET uses the Direct Sequence method, as illustrated in Figure 1. The pseudo random binary sequence is also known as the direct or chip sequence.

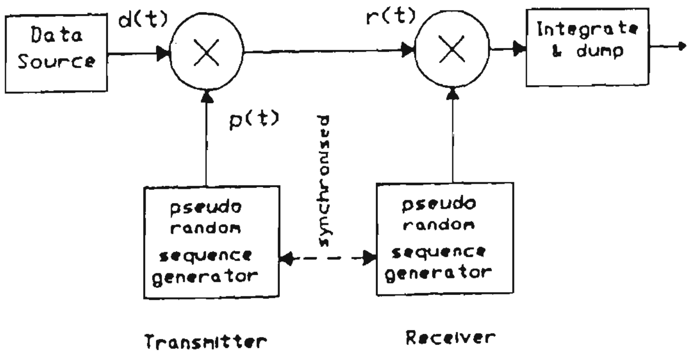


Figure 1 Single channel direct sequence system

The relationship between the data and chip signals is shown in Figure 2

Each data bit d(t), of energy E_b and duration T, may be represented by

$$d(t) = \pm\sqrt{E_b/T} \qquad (1)$$

This one dimensional data bit is multiplied by a binary ± 1 chip sequence, p(t), running at a

A.L. Simcock, C. na Ranong & P. Leung are all members of the Department of Electrical & Electronic Engineering, at Footscray Institute of Technology, and hold the positions of lecturer, senior lecturer and senior lecturer respectively.
(Paper E1279 submitted in May 1986)

frequency of f_c chips/sec, ie. a total of f_cT chips per data bit. The resultant sequence $d(t)p(t)$ is then a f_cT dimensional signal.

The ratio of the dimensionality of the chip sequence to the data is called the 'Processing Gain', and is defined as

$$\text{Processing Gain} = \frac{B_{SS}}{B_D} \qquad (2)$$

Where B_{SS} = Bandwidth of the Spread Spectrum Signal
and B_D = Bandwidth of the Data Signal

The numerical value of the processing gain is an indication of a power improvement factor, which a receiver, posessing a copy of the transmitter's chip sequence, can achieve by a process of correlation. This figure gives an indication of the interference rejection capability.

If the 'chipped data' signal, $d(t)p(t)$, is transmitted in the presence of an interfering signal $J(t)$, the received signal would be

$$r(t) = d(t)p(t) + J(t) \qquad 0 \leqslant t \leqslant T \qquad (3)$$

At the receiver the correlation process is carried out by multiplying the received signal with the same chip sequence used to encode the transmitted data, and then integrating the result over a period T to produce a decision variable U, where

$$U = \sqrt{E_b/T} \int_0^T r(t)p(t) \quad dt \qquad (4)$$

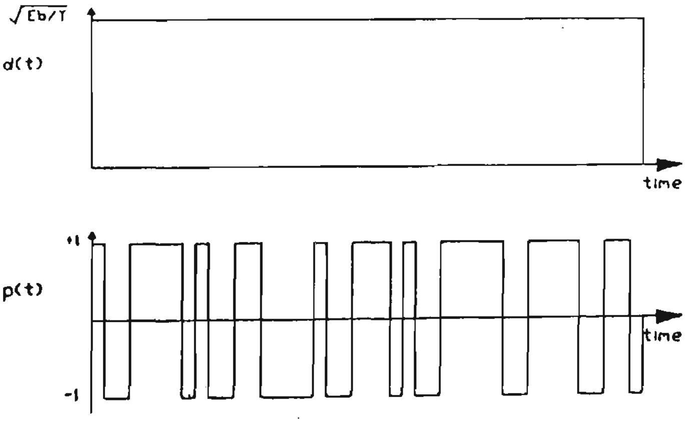


Figure 2 Typical Chip and Data Waveforms

From equation (4) it can be determined whether $\sqrt{E_b/T}$ or $-\sqrt{E_b/T}$ was was sent, depending on whether U is positive or negative. The integrand in may then be expanded as follows

$$r(t)p(t) = d(t)p^2(t) + J(t)p(t) \qquad (5)$$

And, since $p^2(t) = 1$, this may be rewritten as

$$r(t)p(t) = d(t) + J(t)p(t) \qquad (6)$$

thus the receiver recovers the original one dimensional data signal d(t).

In equation (6) the jamming signal $J(t)$ may be an external interference signal, eg. nearby high voltage equipment, or it may be interference from other transmitting nodes. The rejection of the other nodes transmissions as noise depends upon the orthogonality of the codes [6], [7], [8] used for spreading each node's data.

The task of choosing codes for their orthogonality can be simplified if the environment in which the system is to work, such as connection distance and hence propagation delay, attenuation etc. can be controlled. This degree of control allows a synchronous network to be produced. Synchronisation may be considered from both chip and data bit viewpoints. In a synchronous, cable connected, network such as this, with predictable delay and attenuation characteristics, the effect of the so called odd-even cross correlation properties are minimal and codes with multi-level cross correlation values, e.g. Gold or Kasami codes [6], [7], [8], [9] are not necessary. The code allocation scheme selected is based on providing each address a code which is a bit shifted version of the master code, this master code also being the synchronisation code indicated in Figure 3. Thus the cross correlation between codes is the autocorrelation function of the one code. This simplifies both the code generation and decision making circuitry. The allocation of phase shifted versions of the one code to act as the individual node address codes is similar to that used by B. Hirosaki et al [10] in their performance analysis of a fibre optics system. They showed that the variance $[q^1(t)]^2$ in mutual interference from node #1 is given by

$$[q^{(1)}(jNT_c)]^2 = 1/N^2 \qquad (7)$$

where j represents the individual bit position,
N the chip sequence length,
T_c the chip sequence bit period

The equation (7) above, shows that the per-channel interference is inversely proportional to the square of the chip sequence repetition length, or Processing Gain; and since one of the design aims was to produce a reliable and robust system , a fairly large processing gain of 1024 (approx. 30dB) was selected. Evidence indicates [5] that the maximum number of simultaneous users is approximately one tenth of the processing gain. This choice of a high value for the processing gain produces a system capable of working in an electrically hostile environment and allows a large number of simultaneous code transmissions.

Using Maximal Length Sequences (MLS) of $2^{10} - 1$ bits (ie. 1023) to encode each data bit results in a residual offset effect at the output of the analogue multiplier which is dependent on the number of stations transmitting simultaneously. This results from the two level (1023, -1) autocorrelation function of a MLS. In order to eliminate this residual effect the ML sequences were modified to produce codes with (1024, 0) autocorrelation properties, by the addition of an extra binary zero. Each station code is then made up from a cyclically bit shifted MLS plus an extra binary zero which is transmitted as chip bit number 1024. Each data bit is then encoded by the 1024 bit modified MLS. The station code allocation scheme is not affected, each station being allocated a phase shifted version of the same modified MLS.

3 LAN ASPECTS

Current bus and ring Local Area Network topologies, typified by Ethernet and Cambridge Ring respectively, employ some form of (asynchronous) time division multiplexing. This may be in the more or less fixed compartment format of Cambridge Ring or the Packet Message type format of Ethernet. Whichever system is used, any node wishing to access the system must wait until a time when no other users are transmitting. The node will then transmit at a relatively high bit rate into the system. The overall effect is that each node emits bursts of data at high bit rates for relatively short periods, and then remains dormant for a much longer period. Occasionally, two or more nodes attempt to transmit at the same time and the messages collide; back off and retransmission algorithms would then be used to 'arbitrate'. Thus network throughput is dependent on the number nodes in the network and the network bandwidth. Special chip sets capable of high speed burst transmission (eg. Motorola 68590) have been developed for this purpose.

Simultaneous multiple access, offered by TDMA, FDMA and CDMA, in a Local Area Network can increase overall network throughput and possibly eliminate the need for very high speed transmissions. ALSSNET capitalises on the random access, discrete addressing capabilities of CDMA and uses a scheme where each node has its own 'address' and associated code (or spreading sequence). The bus structure is shown in Figure 2.

In the point to point communication system adopted in the network, if node 1 wishes to transmit to node 2 then node 1 will modulate its data with the code of node 2. Hence, the transmitter of each node must be able to modulate its data with any recipient's address code. Since only one node at any instant in time may be transmitting to a particular address, a collision check for the intended receiver is still necessary, and a CSMA/CD protocol is used to resolve contentions within channels. Each transmitter has a built in demodulator to pre-check for traffic in the desired channel and continuously check for collisions while transmitting. Thus each node has two demodulators - one for monitoring the channel for its own message and another one for collision avoidance and detection. The first demodulator works exclusively with its own address code whereas the code used in the second is switchable.

The receiver section of each node monitors the line looking for data modulated with its own 'address code', it demodulates the data encoded with this code, in order to recover the original data, but treats data for any other intended recipient as noise and rejects it, as shown in equation (6). As each node has a unique address only signals modulated with this address code will correlate in its receiver, and be despread, all other codes will not correlate and will be spread further across the band.

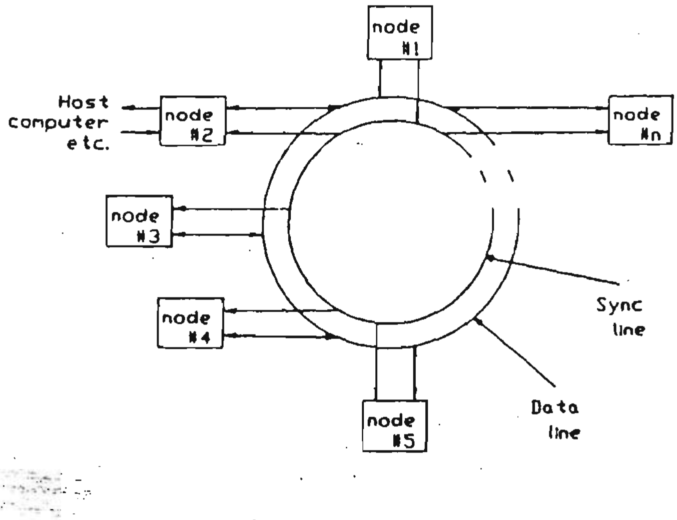


Figure 3 System Bus Structure

ALSSNET is based on twisted or parallel pair cable; this is to maintain simplicity and allow new nodes to be connected to, or disconnected from, the network without unduly affecting other users. The cable is terminated with an appropriate resistance at both ends in order to minimise mismatches and reflections, and each node outputs onto the transmission medium via a high impedence current driver. Figure 4 shows a simple system layout, illustrating the miltiple access concept, with a number of transmitting nodes and a single receiving node. The output from each node is added onto the line to produce a received waveform r(t), Figure 5, which is a summation of the outputs from each transmitter,

$$r(t) = s_1(t) + s_2(t) + \ldots . s_n(t) \qquad (8)$$

$$= \sum_1^n s_n(t) \qquad (9)$$

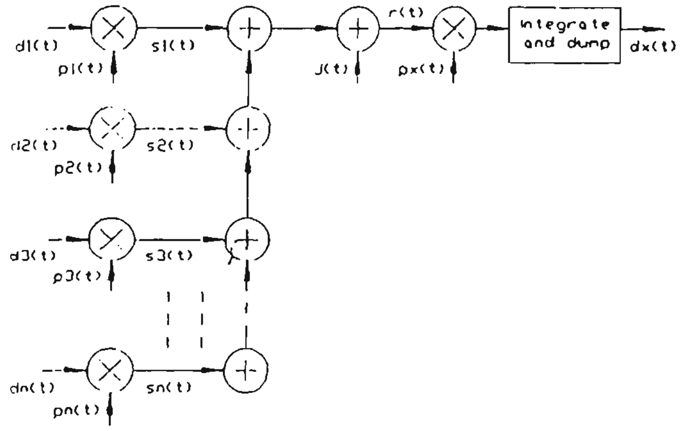


Figure 4 Multiple Access Concept

At the receiver the multi-level line signal r(t) is multiplied by the receiver's own ± 1 level chip sequence $p_x(t)$ and from equation (6)

$$r(t)p_x(t) = d_x(t) + \sum_{n=1}^{n=x-1} s_n(t)p_x(t) + \sum_{n=x+1}^{n} s_n(t)p_x(t) + J(t)p_x(t) \qquad (10)$$

Where $1 \leqslant x \leqslant n$

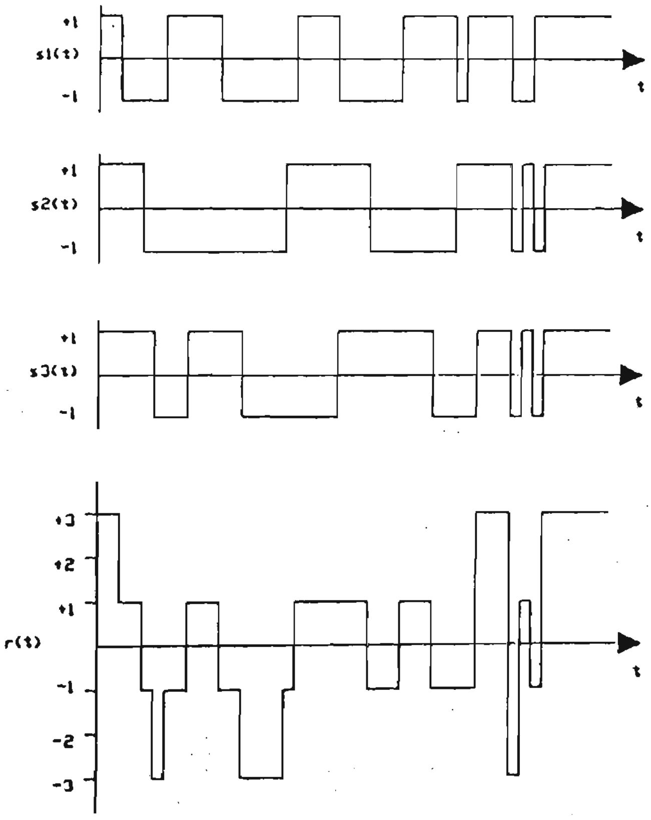


Figure 5 Typical transmitter waveforms

The first term $d_x(t)$, in equation (10), is the recovered one dimensional data signal and the last three terms are the multi dimensional interference signals. The output from the analogue multiplier is subjected to low pass filtering in the integrate and dump circuit, in order to remove the unwanted high frequency components, before being passed on to the decision circuit to determine whether this particular data bit represents a logic one or a logic zero.

In order to test code recognition and phase synchronisation, a novel error generator which produces random-errors at rates between 1×10^{-6} and 9×10^{-1} is used. This generator is capable of maintaining the error rates irrespective of the input data rate which may vary from 1 to 1.5×10^6 bits per second. Tests with the prototype system have showed that the system reliably achieves phase synchronisation within 40 ms, even with an error rate as high as 1×10^{-2}, and that synchronisation dropout does not become a problem until the error rate reaches 2×10^{-2}.

4 COMPUTER STUDY

A computer study of the system coding aspects is being developed. Ideal line drive characteristics of a 60 user system have been investigated and the performance evaluated from two aspects:

i) To ensure that transmission #x (see Figure 4; where $1 \leqslant x \leqslant n$) will be successfully decoded when a number of simultaneous transmissions are taking place. A simulation containing 60 nodes has been developed and the correlation properties investigated. This simulation has verified the 1024, 0 correlation properties of the codes selected.

ii) To investigate the orthogonality properties of the codes used. This part takes the form of simulating a number of simultaneous transmissions (#1 to #n excluding #x) and attempting to extract data using code $p_x(t)$ (ie. the code omitted). Upto 59 simultaneous transmissions have been demonstrated to create no mimicry problems, ie. no combination of code transmissions from upto 59 stations looks like the code of the non-transmitting station #x.

5 PROTOTYPE SYSTEM TESTS

The multiple access concept is illustrated in Figure 4. Three transmitters and two receivers were used for the purpose of preliminary Bit Error Rate (BER) measurements. The first transmitter/receiver pair were used for the BER vs S/N measurements.# The second transmitter/receiver pair was used to verify the simultaneous transmission capability, the third transmitter acting merely as an additional source of interference. The upper trace in Figure 6 shows the received signal r(t) with three transmitters, $s_1(t)$ to $s_3(t)$, and Added White Gaussian Noise all present. In this trace the S/N ratio is -10 db. The lower trace shows the data recovered from this signal.

Figure 7 shows preliminary results of the BER vs S/N Ratio measurements. The preliminary results obtained from the prototype system have verified the concept of a synchronous LAN using bit shifted modified Maximal Length Sequences as the address codes. Data recovery has been demonstrated, with the error rates indicated in Figure 7, thus illustrating the potential of such a system for use in an electrically noisy environment.

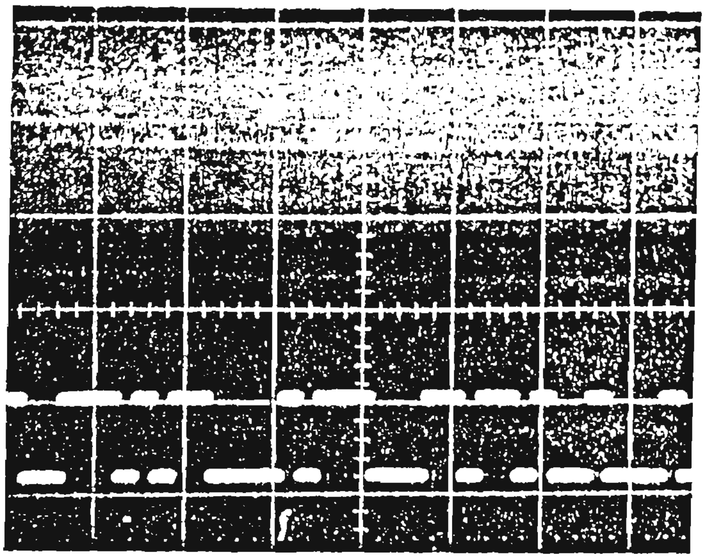


Figure 6 Received Signal and Recovered Data

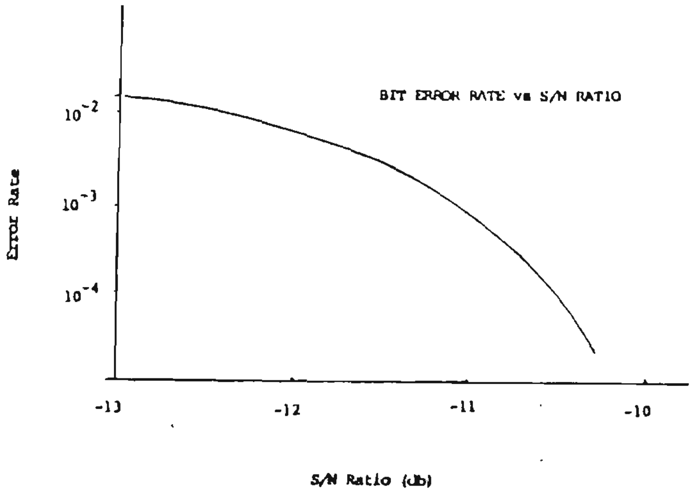


Figure 7 BER vs S/N Ratio

6 THROUGHPUT MODEL

Since user address codes have been chosen for their orthogonality, a point demonstrated by the simple computer study outlined in section 4, it is assumed that each (CDMA) channel's traffic may be regarded as independent in order to model individual channel throughput characteristics.

The techniques used in the derivation of this throughput model are based upon those used by Kleinrock and Tobagi [11]. In the development of this model the following assumptions are made:

a) Traffic sources, on a system and per channel basis, are regarded as Poisson sources with a mean packet generation rate of λ packets per second.

b) Packets are of constant length of T seconds duration, such that the average number of packets generated in this T second period is S = λT. Thus the input rate is normalised to the transmission time T. S may then be called the **Channel Throughput** or **Channel Utilisation**.

c) The offered channel traffic is denoted by G (where G > S). G is equal to the new traffic plus the traffic requiring retransmission owing to some earlier collision.

# NOTE the word Noise is used to represent total interference, ie external interference J(t) plus interference caused by other nodes transmitting simultaneously.

d) $\bar{X}$ is the mean retransmission, or back-off, time and is assumed large compared to T.

e) In this model non-persistent CSMA is used where, upon busy sensing and recognition the transmitter will immediately back off an arbitrary time period $\bar{X}$, and not continue to sense, waiting for the channel to become free.

f) No positive acknowledgement scheme is incorporated, collisions being detected (in a finite time τ) by each station involved in the transmission of the colliding packets. After collision detection packet transmission will be continued for a further period τ to jam the channel in order to to ensure that the other station(s) involved in the collision will detect the collision. The ratio of the collision detection time τ to the transmission time T is given by a where $a = \tau/T$.

g) Busy sensing is achieved in a period equal to the collision detection time a.

h) Propagation delays are negligable compared to T and are set to zero.

i) The interpacket arrival times are independent and exponentially distributed.

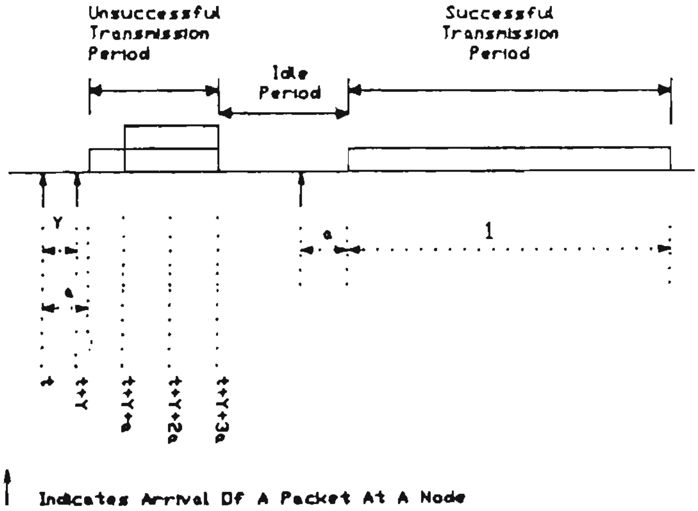

Unsuccessful Transmission Period
Idle Period
Successful Transmission Period
Y
a
a
1
t
t+Y
t+Y+a
t+Y+2a
t+Y+3a
Indicates Arrival Of A Packet At A Node

Figure 8 Channel Transmission Characteristics

Let t be the arrival time, at a node, of a packet to be transmitted. The station will sense the channel for a seconds and, sensing the channel free, will transmit at time (t+a), as shown in Figure 8. # Another station obtaining a packet between t and (t+a) will sense the channel free and also transmit, thus causing a collision. This period of a seconds is classified as the 'vulnerable period', and if no other station receives a packet for transmission in this period then the first's packet will be successfully transmitted.

NOTE Upon commencement of transmission the busy sensing circuit is reset and re-activated as a collision detection circuit.

If (t+Y+a) is the time of transmission of the last packet then at time (t+Y+2a) stations transmitting should have recognised the collision. Each station will continue to transmit for a period of a seconds after collision detection, thus at time (t+Y+3a) all stations will have stopped transmitting. Since no transmissions are occurring during the first a seconds of this period, while the first station is sensing the channel, the average duration of an unsuccessful transmission is $(\bar{Y}+2a)$, and the duration of a successful transmission is equal to 1. The time when transmissions are occurring is defined as that time when traffic is actually flowing and excludes the initial sensing period. This transmission time, whether for a successful or an unsuccessful transmission, is called the Transmission Period (TP) or Busy Period. Any period between two successive TPs is called an Idle Period, and a Busy Period followed by an Idle Period is called a Cycle. If $\bar{B}$ is the expected duration of a Busy Period and $\bar{I}$ the expected duration of an Idle period then the duration of a cycle would be $\bar{B}+\bar{I}$. Using renewal theory the average Channel Utilisation S may then be defined as

$$S = \frac{\bar{U}}{\bar{B}+\bar{I}} \tag{11}$$

where $\bar{U}$ is the average time during a cycle that a channel is being used without conflict. $\bar{U}$ is given by the probability that no other transmission occurs between t and (t+a) and therefore

$$\bar{U} = e^{-aG} \tag{12}$$

The average duration of an Idle Period, $\bar{I}$, is simply $1/G$, and the average duration of the Busy Period $\bar{B}$ is given by

$\bar{B}$ = (Average unsuccessful time)(Probability of an unsuccessful transmission) + (Average successful time)(Probability of a successful transmission)

The probability that the transmission will be successful (Ps) is equal to the probability that no other transmissions occur between t and (t+a).

$$Ps = e^{-aG}$$

Thus $\quad \bar{B} = (1 - Ps)(Y + 2a) + (Ps)(1) \qquad (13)$

The average value of Y, from [11], is given by

$$\bar{Y} = a - \frac{(1 - e^{-aG})}{G} \tag{14}$$

Substituting into equation (13) gives

$$\bar{B} = 3a - \frac{(1 - e^{-aG})}{G} - 3ae^{-aG} + e^{-aG} + \frac{e^{-aG}(1 - e^{-aG})}{G} \tag{15}$$

Using the values for I, U and B and substituting into equation (11) gives

$$S = \frac{\bar{U}}{\bar{B}+\bar{I}} = \frac{Ge^{-aG}}{(3aG + (e^{-aG}(2 + G - 3aG - e^{-aG})))} \tag{16}$$

Equation (16) above represents the throughput, S, in terms of the offered traffic, G, and the busy sensing period, a. Figure 9 is a graph of S vs G, and several curves are included in order to represent different values of a. These curves show the expected characteristic shape with the throughput increasing as the offered traffic increases until a breakpoint is reached. Above this point the probability of collisions occurring is such that increasing the offered traffic actually causes a reduction in throughput as the channel capacity is progressively taken up with more retransmissions and less new traffic. It can be seen from Figure 9 that the busy sensing time is a highly important parameter in the throughput model and that reducing this sensing time causes the breakpoint in throughput to occur at higher levels of offered traffic.

The throughput characteristics of this model are compared with those of some of the more familiar multi-access protocols in Figure 10. These curves are plotted using a value of a = 0.01, except for the two Aloha protocols which are independent of a. In the Non-persistent and 1-persistent CSMA models the value of a represents the propagation delay in the system, rather than a busy sensing period as in this model, but in all three models a gives a representation of the 'vulnerable period' for transmission modelling purposes. These curves show the comparison between the ALSSNET model, which has been developed for a LAN application, and other protocols (most noticably Aloha and Slotted Aloha) which were not. The indication is that application specific protocols can offer throughput advantages.

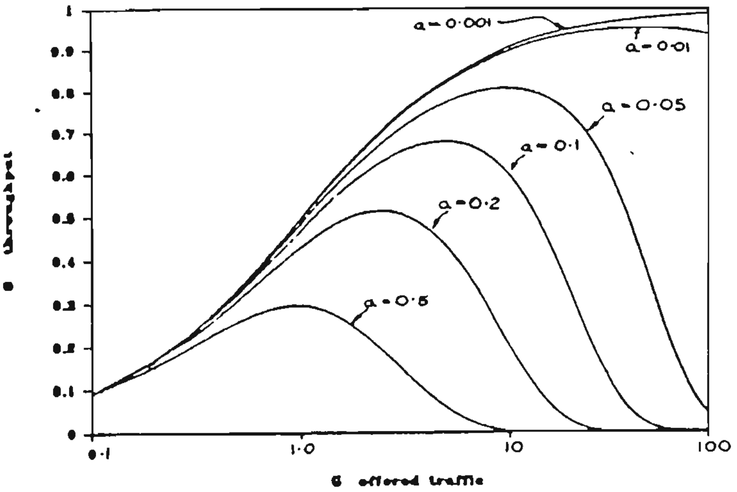


Figure 9 Throughput vs Offered Traffic

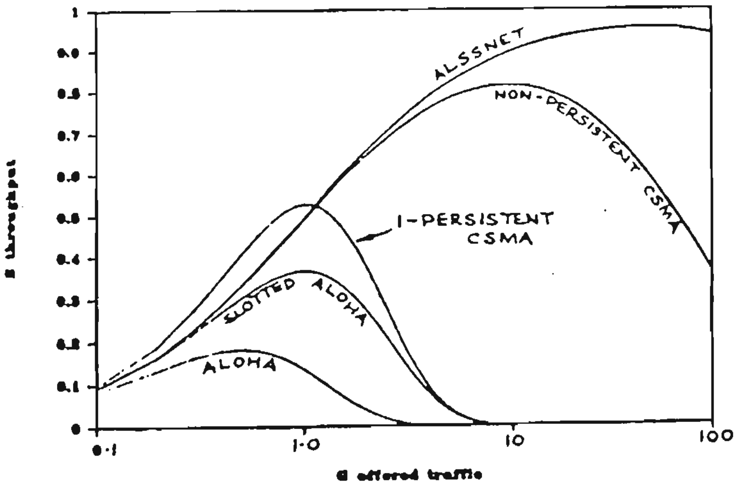


Figure 10 Comparison of Throughput Characteristics

7 CONCLUSIONS

By applying a CDMA spread spectrum techniques to the design of a Local Area Network, a simple and error resistant network is being developed. Its main application will be either as network in an electrically noisy environment or in a high throughput multiple access network. Computer simulations carried out indicate that the system will be capable of supporting over 60 users transmitting simultaneously. Tests carried out on a prototype network consisting of 5 nodes : 3 transmitting nodes and 2 receiver nodes, have verified the results predicted by the computer simulation and demonstrated the feasibility of simultaneous multi-pair communications.

The next stages of the project will be to investigate alternative phase synchronisation schemes and determine the whether the model of the protocol, outlined above, is optimal for control of a system of this nature. The computer studies are also to be enhanced in order to investigate the line monitoring and collision detection functions assumed in the model developed. Investigation of code properties under more realistic conditions is also proposed.

Whilst it is too early at this stage of development to make accurate predictions regarding the cost and reliability of the final network, the reliability of the low cost development modules has been encouraging.

8 ACKNOWLEDGEMENTS

This research project is being supported by a grant from the Australian Telecommunications and Electronics Research Board.

9 REFERENCES

1. SCHOLTZ R.A., "The Origins of Spread Spectrum Communications", I.E.E.E. Transactions on Communications, Vol. COM-30, No 5, May 1982, pp. 822-854.

2. SMYTHE C.S. and SPRACKLEN C.T., "A High Speed Local Area Network Using Spread-spectrum Code-division-multiple-access Techniques", The Radio and Electronic Engineer, Vol. 54, May 1984, pp.225-226.

3. DIXON R.C., "Spread Spectrum Systems", Wiley (New York), 1976.

4. HOLMES J.K., "Coherent Spread Spectrum Systems", Wiley (New York) 1982.

5. PICKHOLTZ R.L., SCHILLING D.L., MILSTEIN L.B., "Theory of Spread Spectrum Communications - A Tutorial", I.E.E.E. Transactions on Communications, Vol. COM-30, May 1982, pp. 855-884.

6. SARWATE D.V. and PURSLEY M.B., "Cross Correlation Properties of Pseudo-Random and Related Sequences", Proc. I.E.E.E. Vol. 69, May 1980, pp.593-619.

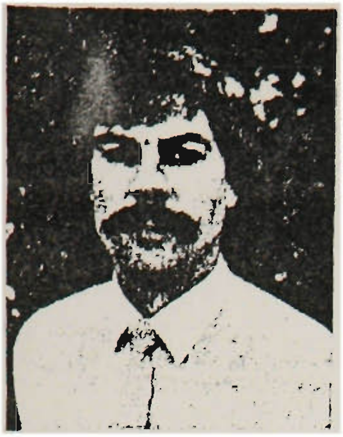
7. GOLD R., "Optimal Binary Sequences For Spread Spectrum Multiplexing", I.E.E., Transactions on Information Theory, Vol. IT-13, 1967, pp.619-621.

8. CARLSON BRUCE A., "Communication Systems -An Introduction to Signals and Noise in Electrical Communication", McGraw-Hill, (Tokyo), 1981.

9. PURSLEY M. B. and SARWATE D. V., "Performance Evaluation for Phase Coded Spread Spectrum Multiple Access Communications - Part II", I.E.E.E. Transactions on Communications Vol. COM-25, August 1977, pp.800-803.

10. HIROSAKI B., HASEGAWA S., SAWAI A., "Spread Spectrum Multiple Access Data Loop", NEC Res DEV n71, October 1983, pp.48-58.

11. KLEINROCK L. and TOBAGI F.A., "Packet Switching in Radio Channels : Part 1", IEEE Transactions on Communications Vol-23, December 1975, pp 1400-1416



A L SIMCOCK

Alec Simcock graduated from the University of Kent at Canterbury with the B.Sc. (Hons) (Electronics) in 1972. He worked for 12 years with Procurement Executive of the Ministry of Defence in UK, researching in the following areas: Digital Circuit design, Digital Data switching, computer controlled Communication systems and Microprocessor replacement of Digital Circuits. The work also involved the implementation of Software Project Management techniques. He was appointed to a Lectureship with the Department of Electrical and Electronic Engineering, Footscray Institute of Technology in 1983.

C na RANONG

Chula na Ranong received the B.E.(Hons) degree in Electrical Engineering and Ph.D. from the University of New South Wales in 1970 and 1973 respectively. Since 1972 he has been with the Department of Electrical and Electronic Engineering, Footscray Institute of Technology as a Lecturer and later, Senior Lecturer in Digital Electronics and Systems. During 1977-78, he spent a year as a Visiting Professor, School of Electrical Engineering, Cornell University, Ithaca, New York, USA. Then, for the first six months of 1982, he worked as a Specialist Engineer with Compagnie Europeene de Teletransmission (Thomson-CSF), Chatou, France developing a mass storage system for a real time digital data network supervisory control system. More recently, as a result of his interest in Automation Technology, Chula undertook a two months study visits to Japanese government, industrial and educational research institutions in 1984.

P S K LEUNG

Patrick Leung received the B.E. (Elect) and M.Eng.Sc. in Electrical Engineering from the University of Melbourne in 1972 and 1974 respectively. Between 1975 and 1977 he was a lecturer in the Department of Electrical and Electronic Engineering, Footscray Institute of Technology. In 1977 he joined Telecom Australia as an engineer in the Datel design and provisioning section. In 1979 he rejoined F.I.T. as a lecturer in communications engineering. In the first half of 1985 he visited the Department of Electrical Engineering, Ottawa Univeristy, as a research fellow, to carry out work in data communications over satellite channels. On his return to F.I.T. he was appointed senior lecturer in communications. His main interests are data transmission techniques, data networks and data protocols.