AN INTERLEAVING TECHNIQUE TO IMPROVE THE BER PERFORMANCE IN A COMPLEX INTERFERENCE ENVIRONMENT

by

Savvas A. Kosmopoulos

The thesis submitted to the School of Graduate Studies and Research of the University of Ottawa in partial fulfillement of the requirements for the degree of Master of Applied Science in Electrical Engineering.

Ottawa, Canada, 1981.

ACKNOWLDGEMENTS

At the end of this adventure I would like to thank all those who helped me to make this thesis possible and therefore complete successfully a struggle which started long time ago. Namely I would like to thank, my parents for the encourangements and continuous motivations they provided me with in order to complete this work, Mr Andre Brind' Amour for his so valuable quidance and supervision and Mr Kenji Yamamoto for the time he spent on the theoretical calgulations on impulsive noise. Finally I would like to thank my dearest Fiorella and Fiorellina for their love patience and understanding, during this hard working period of our life.

ABSTRACT

An experimental error control system utilizing a single error and error-burst correcting technique has been designed, built and evaluated. In the scheme presented, a class (n_2, n_1) block interleaver has been inserted between the channel convolutional encoder and the channel. This type of interleaver reorders a sequence of transmitted symbols so that no contiguous sequence of n_2 symbols in the reordered sequence contains any symbols that were separated by fewer than n_1 symbols in the original ordering. In the receiving end a de-interleaver is inserted between the threshold comparator and the threshold decoder. The de-interleaver, which for an (n_2, n_1) interleaver is itself an (n_1, n_2) interleaver, rearranges the received symbols into their original order.

The reported experimental results clearly demonstrate that the system Bit Error Rate (BER) performance in complex interference environments has been improved by orders of magnitudes, indicating that single occurring errors and error-bursts as long as 32 symbols have been successfully controlled.

GLOSSARY OF NOTATION

Symbol	Definition
SI	Sinusoidal Interference
S/I	Signal to Interference Ratio
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
P(e)	Probability of Error
E_b/N_o	Energy per Symbol to Noise Density Ratio
NRZ	Non Zero Return
LPF	Low Pass Filter
HPF	High Pass Filter
S/N	Signal to Noise Ratio
PRBS	Pseudo Random Binary Sequence
rms	Root Mean Square
TWT	Travel Wave Tube
dB	Decibell
Hz	Hertz
kHz	KiloHertz
kb/s	Kilo bits per second
W/R	Write/Read
DS	Disable
Vcc	Power Supply Line
νT	Impulse Repetition Rate

TABLE OF CONTENTS

LIST OF FIGURES

Figure		Page

CHAPTER I

INTRODUCTION

1.1 An Introduction to Error-Burst Correcting Techniques

Digital communications has become increasingly attractive over the past decades, both because of increased demand for data communications and because digital transmission offers options and flexibility not available with analog transmission [8].

The major components and underlying factors in digital communications are shown in Figure 1.1.1. The detailed structures of the various boxes in this figure depend upon the assumed characteristics of the system in question and upon the state of engineering development at the time of design. The source symbols come either from a digital data source or a digitized analog source. The transmitter sends channel symbols to the forward channel, where interference may cause errors in the received symbols*. Once a waveform is received at the receiver's input, a decision must be made on the possible state of the received symbols; this is the function of a decision circuit (usually a threshold comparator). The ability to make a correct decision depends of course, on the level of the interference in the channel. It is a consequence

* The term of "symbol" is used in this thesis, even though the actual operation of the interleaving system is based on single bit per symbol. However, it is possible that in some coding cases more than one bit per symbol can be used [1].

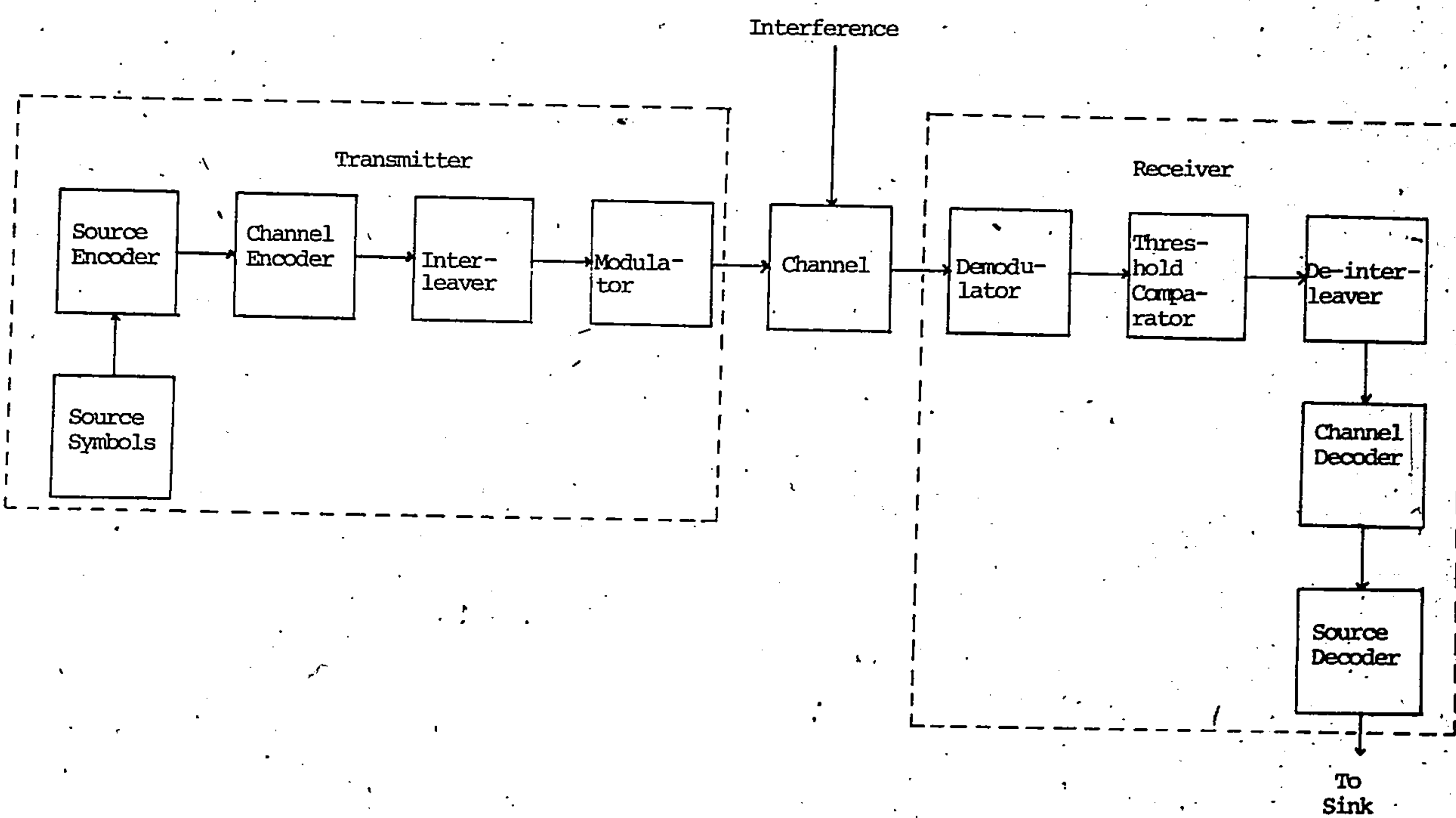


Fig. 1.1.1 - Digital Communication System

of the random nature of the channel interference effects on the received symbols, that the criteria of performance for the system must be also statistical in nature. Thus one can only ask that the probability of being in error [P(e)] in the receiving symbols, be below a certain bound.

In general, interference, present in a communication channel, may be caused by a wide variety of natural and man-made sources, such as lighting, cosmic noise, atmospheric absorption, rain fall, thermal noise, sinusoidal and square wave interference, ignition interference etc. Traditionally these interferences are divided as follows [19] [20]:

i) Gaussian-distributed random noise from receiver front ends or from antenna pick-up of natural sources, such as cosmic noise and thermal noise;

ii) impulsive noise from natural and man-made sources, such as lighting, out of band intermodulation products and ignition interference;

iii) interference from other signals, such as sinusoidal and square wave interference.

If only Gaussian distributed random noise is present, the channel is said to be memoryless, or to be a random channel. In such a channel, the probability of symbol being in error is the same for every symbol and does not depend on what has occurred with any previous symbols (assuming that there is no intersymbol interference in the channel). If in place of or in addition to Gaussian noise, impulsive noise and/or interference from other signals are present on the channel, the channel is said to have memory. The effects of impulsive noise and interference from other signals on a received digital signal, is such that many successive symbols are affected [21] [28]. Hence the errors in the received digital signal occur not only independently at random but also in single bursts or chains of bursts, separated by error free intervals.

An error-burst is defined as a collection of one or more symbols, beginning and ending with a symbol in error [5]. The total number of symbols within an error-burst (not necessary in error) including the first and last symbols of the error-burst, is called the burst length.

Two successive single error-bursts are separated by a "guard space" of G or more error-free symbols. On the other hand, a group of error-bursts that succeed each other close in time, and whose length exceeds a specific limit* determined by the constraints of the particular communication system, is called a burst-chain.

Channels with memory usually degrade the BER performance of error correcting codes, designed to operate on memoryless channels [5]. A number of coding techniques for channels with memory have been proposed and demonstrated to be

* For example, if the length of each error-burst of the chain is longer than the typical guard space.

reasonable effective in some cases [5] [13]. To control the errors effectively through such coding techniques, statistical modelling of the channel is required [1]. The greatest problem with such coding techniques is to find meaningful statistical models, since channel memory statistics are often time variable [1]. Therefore codes matched to one set of memory parameters will be much less effective for another set of parameters. In order to overcome this problem, error-burst correcting techniques, independent of the channel memory statistics variations, have been developed [13]. One of these techniques, which is described in this thesis, is based on interleaving of convolutional coded signals.

By interleaving (or interlacing) a t-random error correcting (n, k) code to degree λ, a $(\lambda n, \lambda k)$ code is obtained. If the original code corrects any single error-burst of length b symbols or less, the interleaved code will correct any single error-burst of length λb symbols or less [13]. Therefore interleaving can be used as an adjunct to any coding scheme for improving its error-burst correction capabilities. One method of achieving this interleaving process consists of inserting an interleaver between the channel encoder and the modulator (external interleaving). An interleaver is a device that redistributes the channel symbols so that the normally successive symbols become mutually separated by somewhat more than the length of a "typical" error-burst. Associated with any interleaver is a de-interleaver, which is a device that restores in the receiving end the reordered symbols to their original ordering. Thus interleaving effectively makes the channel to appear like a random-error channel to the decoder [29] [13]. In general interleavers are grouped into two categories; block and sequential interleavers. Block interleaving can be achieved by dividing the symbol sequences into blocks corresponding to a two-dimensional array and to conceptually read symbols in by rows and out by columns [29]. In some applications such as those dealing with sequential processing of symbols, it is more natural to consider synchronous (or sequential) interleavers [29]. In this type of interleaver a symbol is read out each time a symbol is read in.

Particularly efficient interleavers, implemented to break up error-bursts as long as 1024 symbols and ensure that any two symbols within this burst will appear at the decoder with a separation of at least 64 symbols, have been already incorporated in a single integrated circuit chip [1]. With such devices more readily available, coding applications will be more economically feasible in all types of digital transmission systems. Such increased use justifies an investigation of the capabilities of interleaved coded systems. Therefore with this in view, a convolutional codec along with a (32,64)/(64,32)* block interleaving system has been used in this thesis,as illustrated in Figure 1.1.1,to investigate the effectiveness of interleaving on the BER performance of digital transmission of a compound interference environment. The codec, built at the University of Ottawa [7] consists of:

i) a rate 3/4, 2 error-correcting, canonical self-orthogonal convolutional encoder developed by Robinson and Bernstein [18], and

ii) a type II majority logic threshold decoder [17].

The (32,64) interleaver at the transmitting end reorders a sequence of encoded symbols, so that no contiguous sequence of 32 symbols in the reordered sequence contains any symbols that were separated by fewer than 64 symbols (spreading symbols) in the original ordering. The de-interleaver at the receiving end, which is a (64,32) block interleaver, rearranges the received symbols in their original ordering, before interleaving.

To simulate a complex interference environment in the laboratory, the following three schemes have been used:

---

* The selection of the maximum correctable error-burst length or interleaving depth (32 symbols) and the number of spreading symbols (64 symbols) for the interleaver, is an engineering judgement, partly based on the 3/4 convolutional code error correcting capabilities. This code is guaranteed to correct up to 2 errors in any 80 consecutive channel symbols, when using threshold decoding with syndrom resetting [7]. Therefore the interleaving degree in this application is equal to 16.

i) Sinusoidal interference plus Gaussian noise,

ii) impulsive noise consisting of periodically occurring impulses, and

iii) non-Gaussian noise, obtained by non-linear processing of Gaussian noise. This non-Gaussian noise generation method can be compared to the mechanism of out-of-band intermodulation "spikey" product generation, in a real satellite or microwave communication channel [38].

1.2 Outline of the Thesis

In Chapter II of this thesis, principles of coding and a number of coding techniques are briefly discussed. In Chapter III, the concept of interleaving is presented with emphasis mostly given to interleaving convolutional codes. In Chapter IV, the implementation of the (32,64)/(64,32) block interleaving system is described. In Chapter V, performance evaluation of the 3/4 convolutional codec along with the (32,64)/(64,32) interleaving system, in a sinusoidal interference plus Gaussian noise and in an impulsive noise environment is reported. Finally, in Chapter VI, the features and limitations of possible applications of the interleaved 3/4 rate convolutional codec are discussed.

CHAPTER II

A REVIEW OF CODING TECHNIQUES

2.1 General

Following the rapid growth of digital communications and their applications in ground-to-ground communications via satellites, coding has gained more and more importance in both commercial and military systems. This is particularly true in communication channels where power limitations rather than bandwidth limitations exist [9]. In such channels the use of coding implies improvement of the communication efficiency by counteracting the effects of the various types of additive interference [20] in the received digital symbols.

In this chapter the fundamental principles of coding, with an emphasis on convolutional coding, are briefly described.

2.2 Error Control Through Coding

The use of digital information in the modern world makes the occurrence of errors undesirable in most practical applications. Therefore, the channel error rate, in some cases must be drastically reduced by some methods in order to satisfy the overall system specifications. One method of reducing the error rate is to increase the energy per symbol to noise density ratio (E_b/N_o). If $E_b >> N_o$ the probability of making an incorrect decision can be vanishingly small. However, in some practical systems, the channel may be so noisy that the E_b/N_o required to bring the error probability to a reasonable value is so high that it may not be practical. This makes it impossible to achieve very low error rates by this method.

A second method of reducing the error rate is through the use of error correcting codes. Error correcting codes do not increase the available E_b/N_o, but they permit the same data fidelity at a lower value of E_b/N_o, which is as real as a gain that is effected by increasing the transmitted energy. Shannon [10] has shown that the capacity (maximum rate at which information can be transmitted) of a linear channel with additive Gaussian noise affecting the signal is given by:

$$C = W \log_2(1+P/N_o W) \text{ symbols/sec} \tag{1}$$

where P = received signal power

N_o = single sided noise spectral density

W = noise bandwidth.

He showed that if the information rate R is less than C, then there exists a coding scheme that will make the probability of error arbitrarily small. If R = C, then

$$P/(N_o\ C) = (2^{C/W}-1)(W/C) \tag{2}$$

and for the case of no bandwidth restrictions, can be derived:

$$\lim_{W \to \infty} P/(N_o\ C) = -\ 1.6 \text{ dB} \tag{3}$$

where $P/(N_o\ C) = E_b/N_o$

This implies that for a power limited system, an arbitrary low error rate (BER) can be theoretically achieved with Eb/No = - 1.6 dB.

In Figure 2.2.1 this optimum bound is shown along with the probability of error curve of a typical digital transmission scheme with no error correcting coding. It can be seen from Figure 2.2.1 that for a BER of 10^{-5}, with

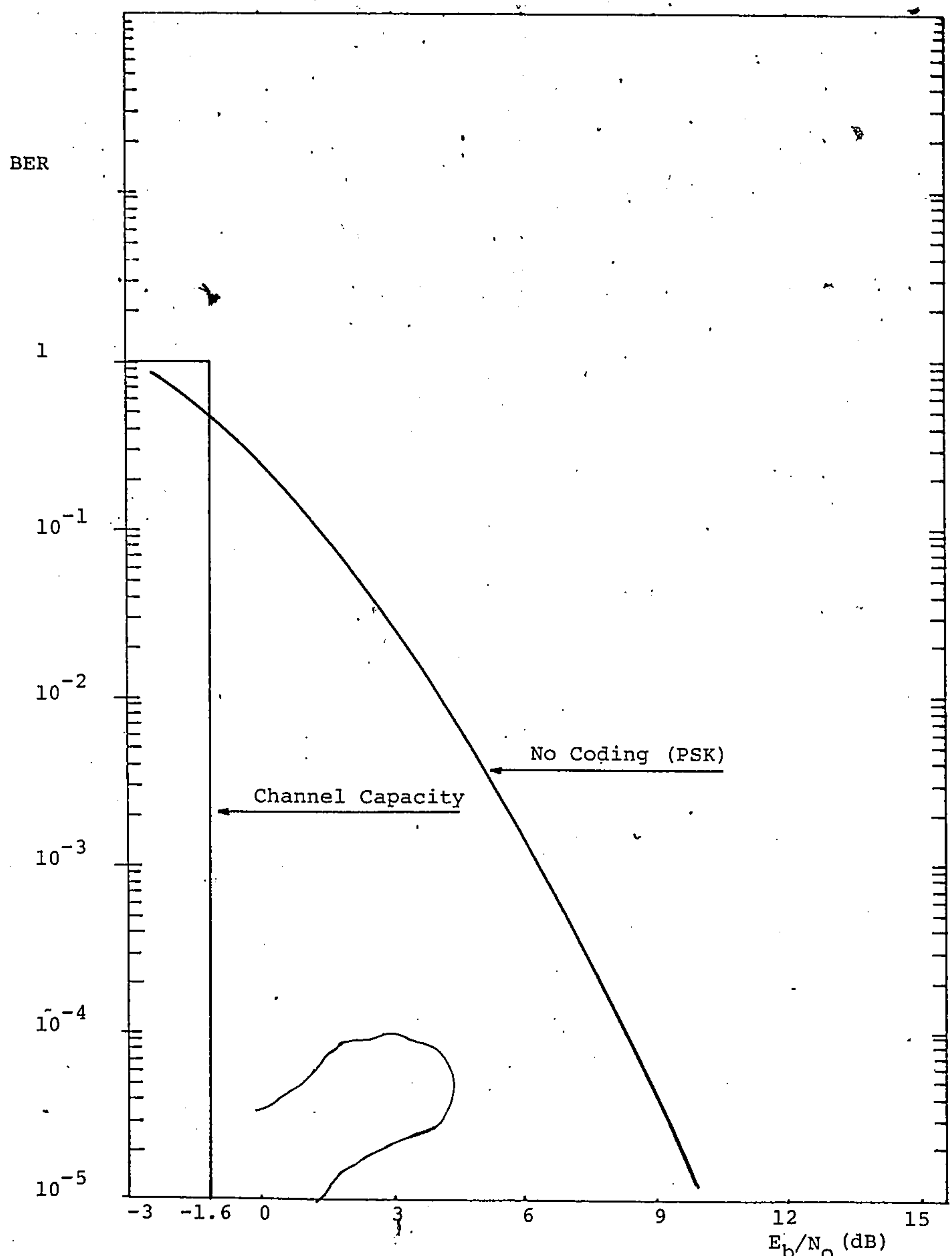


Fig. 2.2.1 - BER vs E_b/N_o for no Coding and Coding at Channel Capacity

no coding, 9.6 dB of Eb/No is required while if Shannon's limit can be achieved the same BER is possible for Eb/No = − 1.6 dB. Therefore a potential coding gain of 11.2 dB is available for a BER of 10^{-5}. Coding schemes, which can provide a substantial fraction of this potential gain, are in use in digital communication systems today [5,] [13].

E.N. Gilbert [11] in 1960 showed that the channel capacity for a communication channel with memory is given by:

$$C_m = C + \mu H_o \tag{4}$$

where C = channel capacity of a random memoryless channel

μ = a measure of the departure of the channel with memory from the memoryless channel of the same rate

H_o = the average conditional entropy* of a typical error sequence of the memoryless channel.

Equation 4 shows that memory increases the channel capacity. This provides further motivation for work on error control techniques for channels with memory.

* The concept of entropy is used here as a measure of the uncertainty of a process involving the erroneous and correct processing of data symbols through a memoryless channel [36].

2.3 Types of Codes

Error correcting codes can be classified as belonging to one or the other of two fundamentally different types [13]. In block codes, the information sequence is divided up into blocks of k digits, which are then encoded into blocks of n digits where $n > k$. Each block is encoded and decoded independently of all other blocks. The number n is called the code length, or the block length of the code. The information rate of such a code is equal to k/n.

The other type of codes are under the general name of tree codes. When such a code is used, each long, perhaps semi-infinite information sequence is encoded into a somewhat longer code sequence. The information sequence is divided into blocks of k symbols. Then the encoder outputs a section of the code sequence of length n, where the particular pattern of the n symbols depends not only on the k symbols being processed but also on some previous blocks of symbols. Such codes are called tree codes because the encoding rule is described most conveniently by a tree diagram. Convolutional codes [5] form a subset of the tree codes.

In a real communication channel the communication efficiency, promised by Shannon's coding theorem [10], can be approached by means of different schemes employing convolutional coding. Among these schemes, the most attractive schemes include [5] [13]:

i) The scheme employing convolutional coding at the input and threshold decoding [17] at the output and

ii) the scheme employing convolutional coding at the input and probabilistic (Viterbi, Sequential) decoding [13] at the output.

The following sections will deal with convolutional codes and threshold decoding, for the correction of random errors, error-bursts and simultaneously occurring error-bursts and random errors.

2.4 Convolutional Codes

Convolutional codes (or recurrent codes) were first introduced by Elias [15] in 1955. Since convolutional codes belong to the tree code category, their difference from block codes can be described as follows. In a block code, the block of n code symbols generated by the encoder in any given time unit depends only upon the block of k input message symbols, within this time unit [13]. On the other hand, in a convolutional code, the n code symbol block, which the encoder produces in a given time unit, does not depend only upon the block of k input message symbols within that time unit but also on the symbols of input message blocks of the previous N-1 time units ($N > 1$) [13]. Like block codes, convolutional codes can be designed to have independent random error correcting abilities, error-burst correcting abilities, or a combination of both.

Figure 2.4.1 shows the basic arrangement for a convolutional encoder, which will be used as an illustrative example. The span or constraint length of the encoder is termed K, which specifies the number of k-symbol blocks, over which a given information symbol can influence the encoder output. Therefore a (n, k) convolutional code of constraint length K is capable of correcting any error sequence with t or fewer errors in any span of nK consecutive symbols by a particular decoding algorithm. If k information symbols are entered at a time, the number of shift-register stages required is kK, for a span K. The k input information symbols (also called message block), to be encoded, are shifted into the shift register in a clocked sequence. The stages are binary added in n separate modulo 2 adders, each connected to different register taps.

At predetermined time units, a commutator reads out the output of the n adders in sequence forming a n symbol word (also called the code block) at this time unit. As the input information sequence is shifted in, the clocking of the commutator is repeated and another n symbol code block appears at the output. Practically, the commutator is clocked after the k symbol message block is stored into the register. Therefore, for each message block of k symbols, there is a code block of n symbols produced at the output. Then is said that the encoder operates at an output rate of

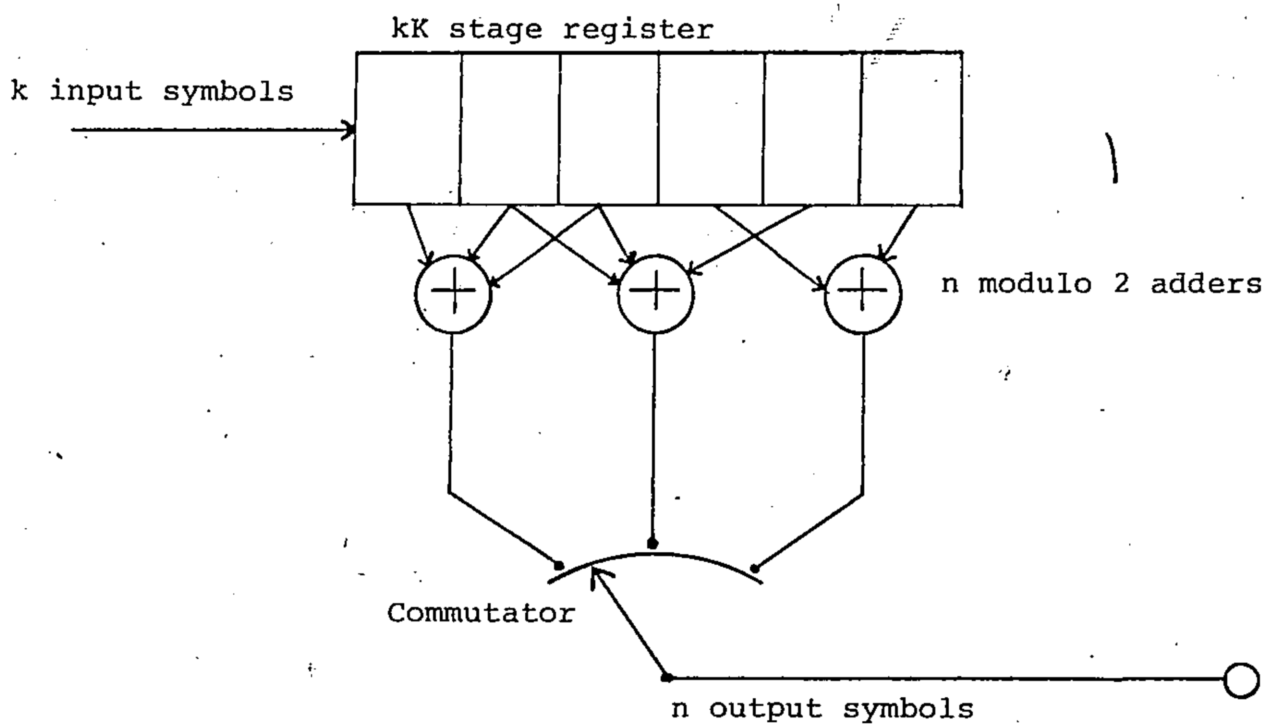


Fig. 2.4.1 - Convolutional Encoding Using Shift Registers

n/k times the input rate. The ratio k/n is called the encoder rate. When encoding is accomplished in this manner, the output sequence of the code blocks can be written as a convolution of the input sequence of the message blocks, and an encoding weighting function [5]; hence the name convolutional coding.

Convolutional codes are classified as systematic and non-systematic codes. If the modulo 2 additions are such that the k symbols of the message block appear as the first k symbols of the n output symbols, then the convolutional code is called systematic. The encoding constraint length of a systematic code is usually the same as the decoding constraint length [5]. The same code in a non-systematic form has a shorter encoding constraint length than the decoding constraint length, and hence fewer memory elements required in the encoder [5].

A general decoder for a systematic convolutional code is shown in Figure 2.4.2.

The operation of the circuit is as follows. With gate 1 closed and gate 2 open, k information symbols are fed into the information register. Then with gate 1 open and gate 2 closed, the received parity check symbols (n-k) and recalculated parity check symbols are compared [5]. The difference in these two sets of parity check symbols is the error syndrome which is unique for each correctable error pattern. This syndrome is fed into the syndrome register.

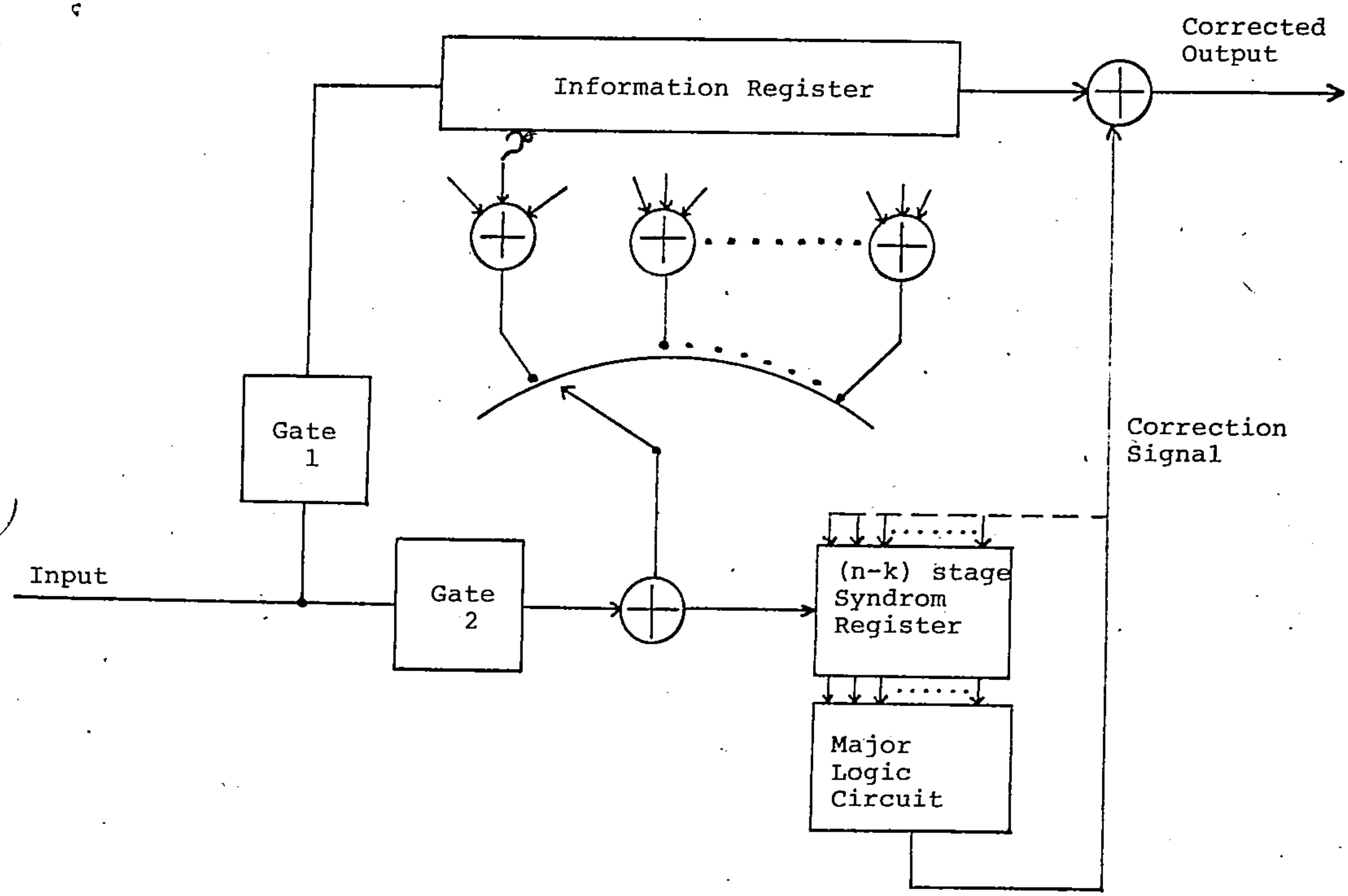


Fig. 2.4.2 - A Majority Logic Threshold Decoder for a Systematic Convolutional Code

The majority logic circuit then determines from the syndromes whether or not a correctable error pattern has occurred in the k information symbols ready to be released from the information register. If so, this block of information symbols is corrected as it leaves the decoder.

In order to make use of the full error correcting capability of the code, the effect of incorrect symbols on the syndrome must be removed as the errors are corrected. The feedback connections noted by the dotted line in Fig. 2.4.2 does this. However, if an uncorrectable error pattern occurs, the feedback connections can cause error propagation. That is, the decoder can continue to decode incorrectly even though no further channel errors occur. If the effect of the feedback symbols is always to reduce the weight of the overall syndrome register symbols, the propagation will cease after a finite and usually small number of blocks [13]. If this condition is not met, the output of the feedback register may be periodic and infinite. That is, catastrophic error propagation occurs. If no feedback is used, the code does not perform up to its capability, but error propagation does not occur.

Such a decoding scheme is called majority logic decoder and is discussed in more detail by J.L. Massey [17].

For a given m, there is a single error correcting convolutional code with parameters $n=2^m-1$ and $k=n-1$ and a double error correcting convolutional code with parameters $n=2(2^m-1)$ and $k=n-2(m+1)$. In addition, there are two classes of random error correcting convolutional codes, the self-orthogonal and the orthogonalized codes [5].

In general, a convolutional code is called self-orthogonal if its complete set equations are orthogonal on a given symbol dj. The concept of orthogonality differs from the well known one from mathematics and signal analysis, in which two vectors or functions are orthogonal to each other if and only if their inner product, appropriately defined, is equal to zero. Rather, orthogonality in coding theory is defined as follows [5]:

Given a set of equations which relate various information and check symbols through check sums, it is said that these equations are orthogonal on a symbol dj, if dj is checked by every equation and if no other symbol is checked by more than one equation. Self-orthogonal convolutional codes (Massey, 1963; Robinson and Bernstein, 1966) are majority logic decodable which makes them even more attractive. (For more details, see Section 13.4 of reference [5].)

2.5 Error-burst Correcting Convolutional Codes

When convolutional codes are used for error-burst correction, it is necessary to define both b, the burst correcting capability (or the length of a correctable error-burst) and g, the guard space (or error-free gap) required to allow the error-burst to be corrected [28]. A guard space is the number of error-free symbols that must follow an error-burst. Any error within this guard space may cause incorrect decoding [5]. The burst correcting ability of an (mn, mk) convolutional code appears bounded by [5]:

$$b \leq \frac{(m-1)(n-k)}{1+k/n} + n - 1 \tag{5}$$

and the guard space is upper bounded by:

$$g \leq mn - 1. \tag{6}$$

For practical decoders, g cannot be much less than (mn-b).

An (mn, mk) convolutional code is said to have Type-B_2 error-burst capabilities, if all error-bursts of length $b_2=rn$, which are confined to r consecutive blocks are correctable, but at least one error-burst of length $(r+1)n$ is uncorrectable. The burst correcting ability of a code with Type-B_2 error-burst correcting ability b_2 is bounded by:

$$b_2 + n - 1 \geq b \geq b_2 - n - 1 \tag{7}$$

and also

$$b_2 \leq (n-k)/2. \tag{8}$$

Convolutional codes having Type-B_2 error-burst correcting ability can be interleaved to produce codes with capabilities for correcting longer error-bursts.

It is possible to construct convolutional codes that correct both error-bursts and random errrors. One way to construct such a code is to interleave a random error correcting convolutional code. (For more details, see Chapter III.) The diffuse convolutional codes have inherent error-burst correcting ability as well as random error correcting ability. A convolutional code is said to be b-diffuse if

i) no burst of length b or less, which begins with the first information symbol to be decoded, affects more than t-1 of the syndrome symbol used to decode that symbol, and

ii) no burst of length b or less which begins elsewhere affects more than t of these symbols [5].

Such a code can correct simultaneously t random errors and an error-burst of length b.

A method called adaptive decoding [5] can be used to allow a convolutional code to correct both error-bursts and random errors. The basic code is designed so that only a small fraction of its cosets are used for random error correction. The decoder operates in a random error correcting mode until it detects an uncorrectable error pattern. Since only a small fraction of the cosets are used for random error correction, the probability of detecting patterns which are uncorrectable, before erroneous symbols have left the decoder, is high. Once the decoder detects an uncorrectable error pattern, it assumes that an error-burst has occurred and switches to an error-burst correcting mode. When a number of blocks have been decoded for which the syndrome contains all zeros, the decoder considers that the error-burst has ended and returns to the random error correcting model.

In the succeeding chapter on interleaved codes, only convolutional codes will be considered, although most of what will be said can also be applied to block codes.

CHAPTER III

INTERLEAVING AND INTERLEAVED CODES

3.1 First Approach to Interleaving

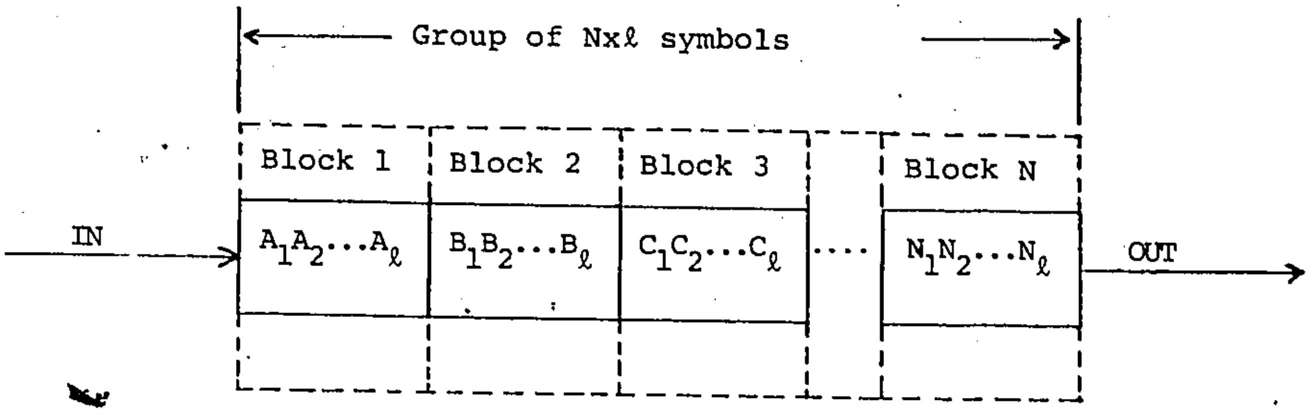
Interleaving is an adjunct to coding error-burst correcting technique, that is used to spread the effects of an error-burst among a number of codewords of a basic code. This can be accomplished as illustrated in Figure 1.1.1. In the transmitting end the source symbols are first encoded in codewords of an original code. Then the interleaver spreads in time the encoded symbols and in the spaces thus produced, symbols from other codewords are introduced. For the purpose of this thesis, the constituent codewords of an interleaved code will be referred to as component codewords.

In the receiving end the interleaved symbols are reordered to their original order by the de-interleaver whose function is identical and complementary to the function of the interleaver. Thus the error-bursts, caused to the received symbols by the channel interference are spread in time and appear like random errors to the decoder.

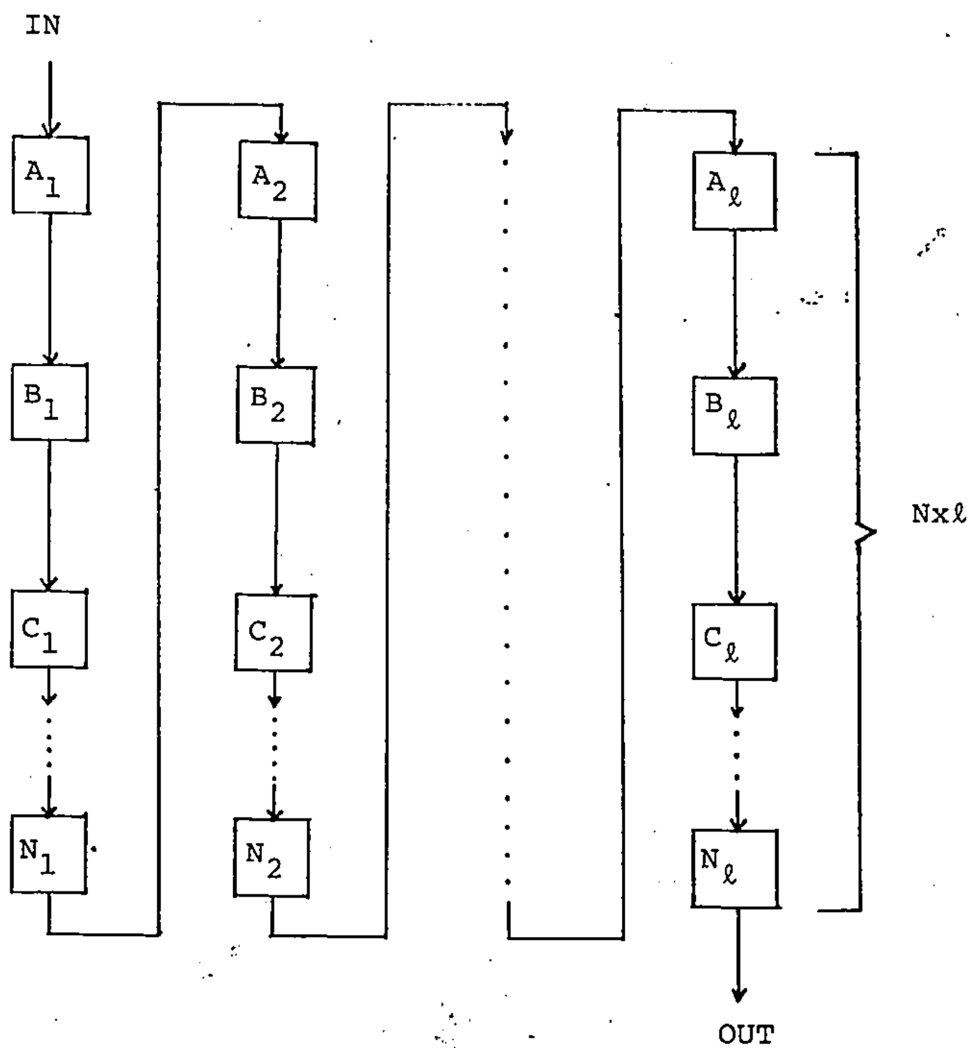
Usually there are two ways of interleaving an error correcting code. Block and symbol interleaving. If block interleaving is used, the component codewords are divided into blocks of length l, where usually l=b (the error-burst correcting ability desired for the interleaved code). Each such block of a component word is separated from other blocks of the same word by a block of every other component codeword. A common example of a block interleaving function is shown in Figure 3.1.1, where the input symbol sequences are divided into blocks corresponding to a two-dimensional array, and conceptually symbols are read in by rows and out by columns. On the other hand, Figure 3.1.2 illustrates the function of block de-interleaving.

If symbol or synchronous interleaving is employed, a symbol is read out each time a symbol is read in. Then no two symbols from any component codeword will be adjacent in the word of the interleaved code. Symbol interleaving is a more general class of interleaving than block interleaving, since any block interleaving function can be realized by a synchronous interleaver [29]. Therefore, Figure 3.1.1 can represent a synchronous interleaving function as well, providing that one symbol is read in when a symbol is read out. On the other hand, a significant difference between block and synchronous interleaving exists. That is, for continuous flow of data through a block interleaver, data bufferring is necessary (see Chapter IV).

Interleaved codes are efficient in a probabilistic sense. No elaborate decoding scheme is necessary to take advantage of this error correcting ability. They are successful because they can correct many more error patterns than their guaranteed error correcting ability would indicate, before interleaving.

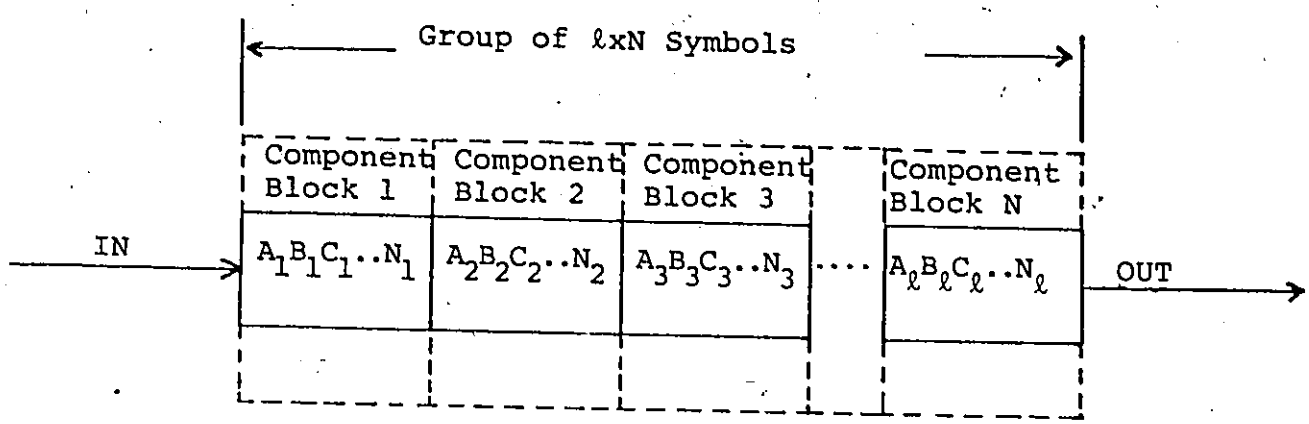


a) Symbols Read in by Rows

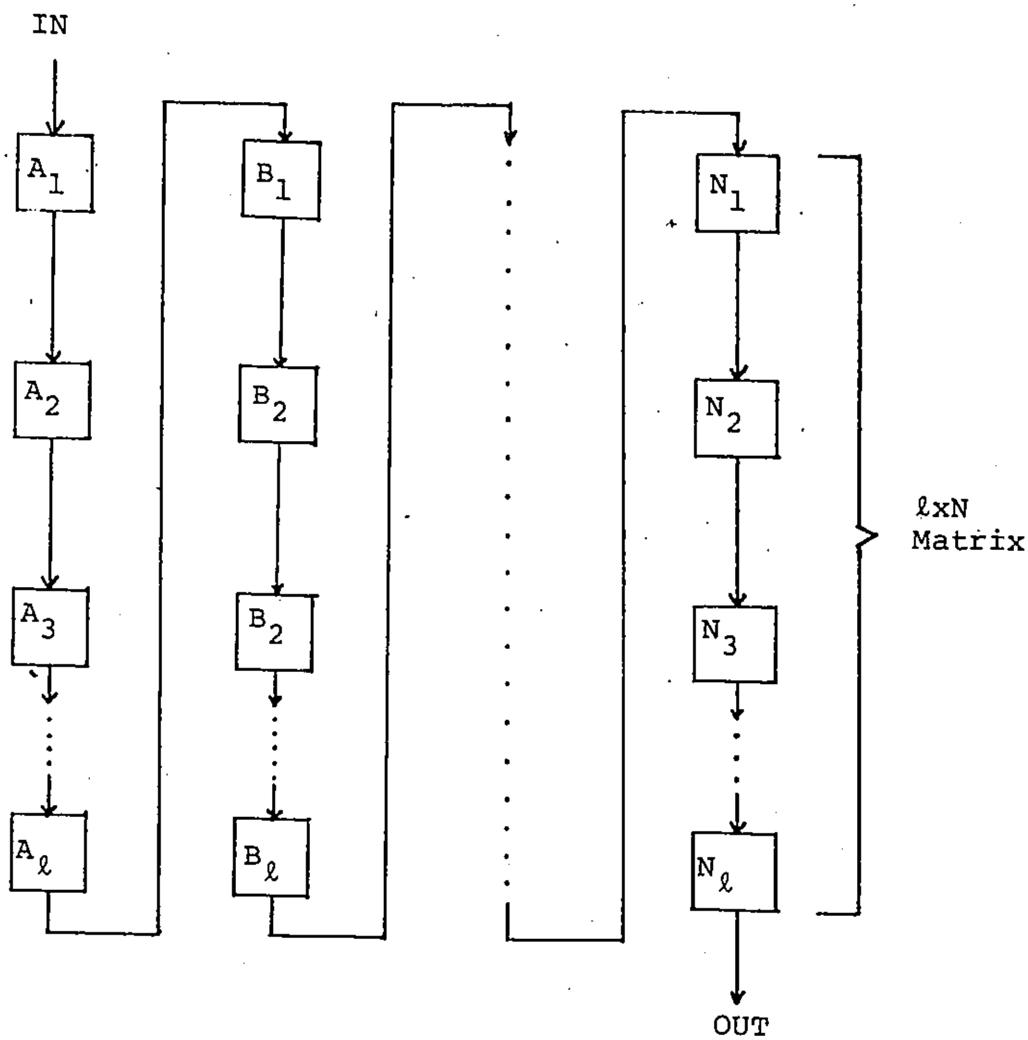


b) Symbols Read out by Columns

Fig. 3.1.1 - Block Interleaving



a) Symbols Read in by Rows



b) Symbols Read out by Columns

Fig. 3.1.2 - Block De-interleaving

3.2 Interleaving of Random Error Correcting Codes

Codes which are designed to correct random errors can be interleaved to produce codes which will correct long error-bursts in addition to random errors. Given a (n, k, t) random error correcting code, it is possible to construct a (λn, λk, t) code, i.e., a code λ times as long with λ times as much information symbols, by interleaving. The parameter λ is called the interleaving degree. The interleaved code is guaranteed to correct any single error-burst of length λt or any t random errors. However, this code will also correct at least one pattern of λ error-burst of length t and another of λt random errors [5].

Block interleaving to degree λ of a random error correcting (mn, mk, t) convolutional code, produces a longer convolutional code which can correct all error-bursts of length [t/n]λ or less, all error patterns of weight t or less, and many error patterns of weight somewhat greater than t (burst-chains) [5].

3.3 Codes Correcting Error-bursts and Random Errors

The codes presented in this section have been designed [5] to correct error patterns that have neither all errors occurring at random nor to a single error-burst. Out of the many existing techniques [5] [13], interleaving a random error and burst correcting convolutional code is a very powerful technique.

Block interleaving to degree λ a (mn, mk) convolutional code, with b error-burst correcting abilities, produces a $[mn(\lambda-1)+n, mk(\lambda-1)+k]$ code with b error-burst correcting abilities. The self-orthogonal convolutional codes have random error correcting capability t and error-burst correcting ability that typically is only slightly larger than t. Therefore by interleaving a self-orthogonal convolutional code its error-burst abilities are increased by a multiple of the interleaving degree λ. Obviously the performance of these codes improves as the interleaving degree increases, because the probability of an uncorrectable error pattern in a component word decreases [13].

As in the case of interleaved random error correcting codes, interleaved codes with random error and burst correcting abilities cannot correct only error-bursts of length b, but they can correct also many patterns of multiple error-bursts [36].

From the above discussion on interleaving and interleaved codes, it is concluded that, in general, interleaving of an error correcting code can be achieved by the following two ways:

a) <u>External interleaving</u>: In this type of interleaving the interleaved code is derived from the basic code by the connection to the encoder and decoder of an external interleaving unit (see Figure 1.1.1).

b) <u>Internal Interleaving</u>: In this type of interleaving the interleaved code is derived from the basic code by replacing each shift register single-unit of the encoder and decoder by delay-units, without changing the other encoding and decoding connections.

Usually, external interleaving is more preferred than internal interleaving, mainly because of the ease of implementation and flexibility of use (easier expansion of interleaving degree). The effectiveness of external interleaving on the performance of convolutional coding in various error-burst environments is experimentally verified in Chapter V.

CHAPTER IV

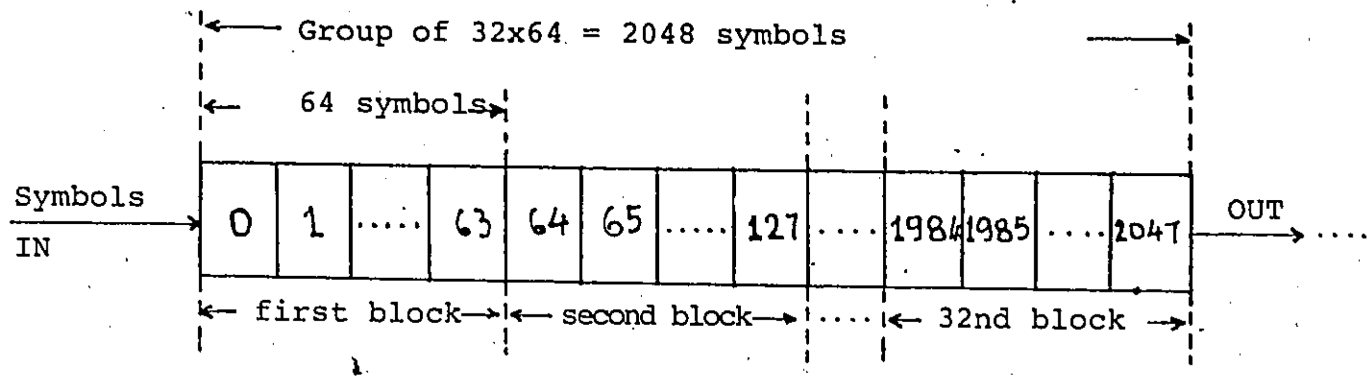
IMPLEMENTATION OF A (32,64)/(64,32) BLOCK INTERLEAVING SYSTEM

4.1 General

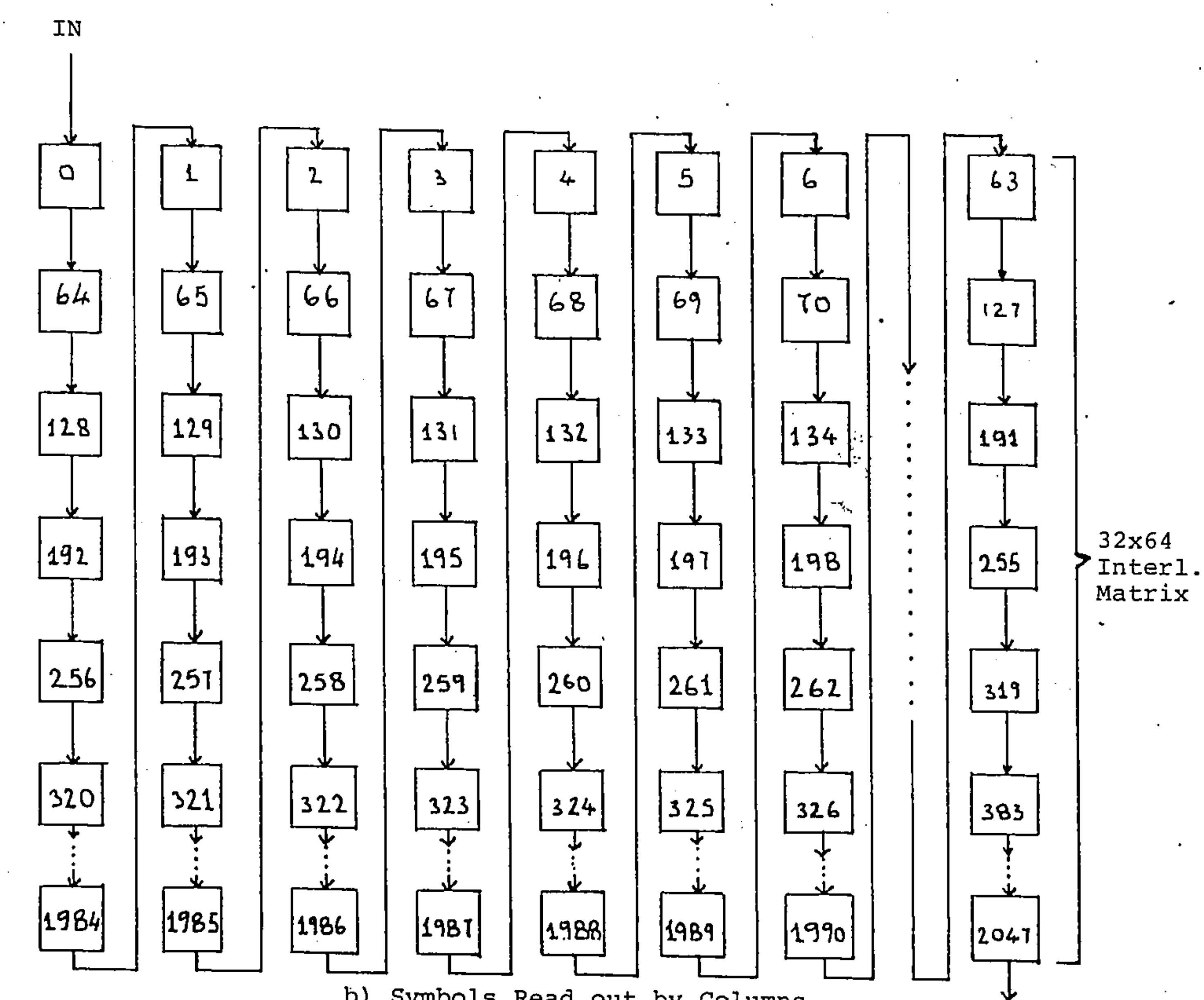
This chapter describes the hardware design of a (32,64)/(64,32) block interleaving system. The general functional description of block interleaving, was already given in Chapter III. Therefore this chapter mainly describes the specific functional and hardware characteristics of the given interleaving system, as resulted from the system error correcting requirements presented in Chapter I.

4.2 The (32,64) Block Interleaver

As already stated in Chapter I, for this interleaver, the maximum correctable error-burst length is equal to 32 symbols and the number of spreading symbols equal to 64. Then the symbol read in and read out processes of the interleaver can be described as follows. Figure 4.2.1 illustrates the functional diagram of the (32,64) block interleaver. As can be seen from this figure, the incoming information symbols are first numbered as they enter the interleaver. This position numbering of the symbols starts randomly in time, when the interleaver is connected to the power supply line. Then the interleaved symbols are organized into groups of 2048 (2^{11}) symbols. Each of these groups is divided into 32 blocks with 64 symbols per block and therefore represents an interleaving matrix of 32 rows and 64 columns. Each row of this matrix corresponds to one symbol block, with the first row corresponding to the first block. For the storage of the symbols of the interleaving matrix into the interleaver, Random Access Memories (2102A) with 2K(2048) storage positions are used. For the addressing of these storage positions and the control of the write (read in) and read (read out) operations of the interleaver memory, a 2^{11} (12 outputs) free running counter is used. Actually, the binary translation by the counter, of each information symbol position number represents the addressing sequence of the corresponding



a) Symbol Read in by Rows



b) Symbols Read out by Columns

Fig. 4.2.1 - (32,64) Block Interleaving

symbol. This counter is synchronized to the incoming information symbols by the use of the same clock. When all the symbols of an interleaving matrix are stored into the interleaver memory, the read out process starts. This is done as shown in Figure 4.2.1 b, where symbols are read out by columns rather than by rows.

Figure 4.2.2 illustrates the interleaver memory addressing for the write operation. For this mode of operation of this specific memory, the Write/Read (W/R) control line is required to be kept at low (logic 0).* This is achieved by the 12th output of the free running counter (see Figure 4.2.2 a). Since this counter output changes level from low (logic 0) to high (logic 1) and the opposite every 2048 clock pulses, it is evident that the write (or read) operation of the interleaver memory is alternative repeated for every 2048 clock pulses. The rest of the counter outputs are straightforward connected to the memory addressing lines (see Figure 4.2.2 b). Of course, this is the result of the pattern of the memory write addressing sequences (see Figure 4.2.2 a).

Figure 4.2.3 illustrates the interleaver memory addressing for the read operation. Obviously, for this mode of memory operation, the W/R control line is required to be

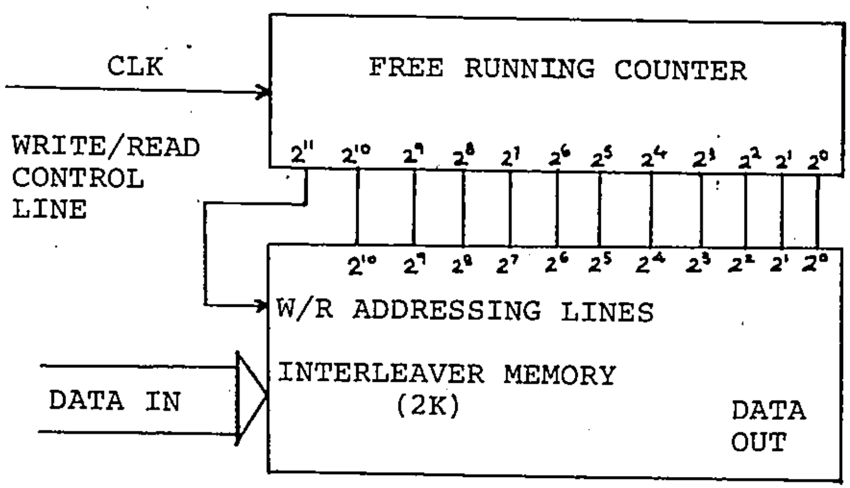
---

* For this hardware implementation logic 0 corresponds to 0 volts and logic 1 to +5 volts.

DECIMAL SYMBOL NUMBER	ADDRESSING LINES											
	W/R	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	0	0	0	0	1	1
4	0	0	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	0	0	0	1	0	1
6	0	0	0	0	0	0	0	0	0	1	1	0
7	0	0	0	0	0	0	0	0	0	1	1	1
8	0	0	0	0	0	0	0	0	1	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2047	0	1	1	1	1	1	1	1	1	1	1	1
2048	1	0	0	0	0	0	0	0	0	0	0	0
	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	COUNTER OUTPUTS											

READ

a) Write Addressing Sequences

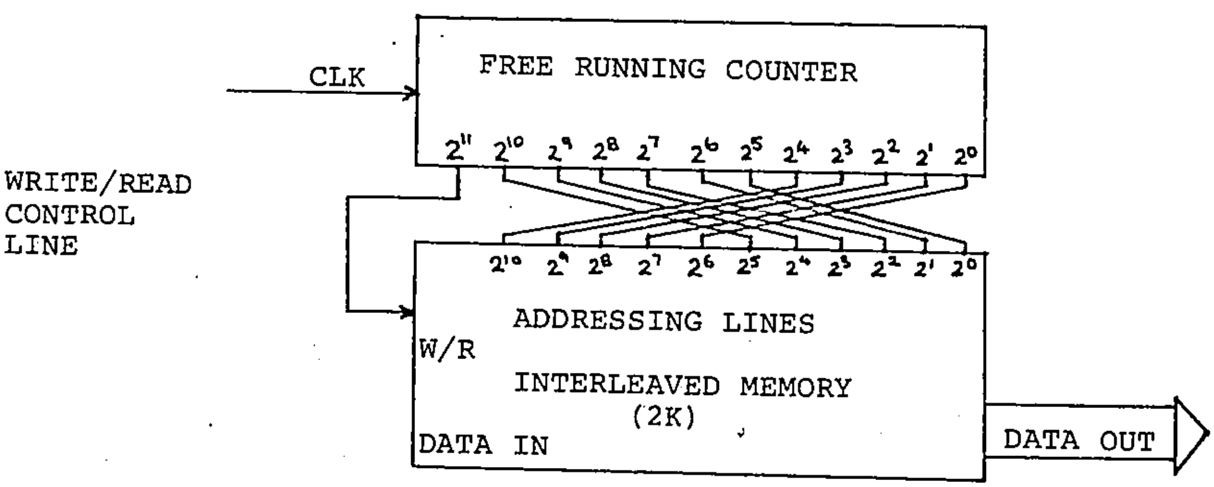


b) Memory Addressing Lines Connection for Write Operation

Fig. 4.2.2 - Memory Addressing for Write Operation

DECIMAL SYMBOL NUMBER	ADDRESSING LINES											
	W/R	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	0	0	0	0	0	0	0	0	0	0
64	1	0	0	0	0	1	0	0	0	0	0	0
128	1	0	0	0	1	0	0	0	0	0	0	0
192	1	0	0	0	1	1	0	0	0	0	0	0
256	1	0	0	1	0	0	0	0	0	0	0	0
320	1	0	0	1	0	1	0	0	0	0	0	0
384	1	0	0	1	1	0	0	0	0	0	0	0
448	1	0	0	1	1	1	0	0	0	0	0	0
512	1	0	1	0	0	0	0	0	0	0	0	0
576	1	0	1	0	0	1	0	0	0	0	0	0
640	1	0	1	0	1	0	0	0	0	0	0	0
704	1	0	1	0	1	1	0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
960	1	0	1	1	1	1	0	0	0	0	0	0
1024	1	1	0	0	0	0	0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1984	1	1	1	1	1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	1
65	1	0	0	0	0	1	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	2^{11}	2^4	2^3	2^2	2^1	2^0	2^{10}	2^9	2^8	2^7	2^6	2^5
	COUNTER OUTPUTS											

a) Read Addressing Sequences



b) Memory Addressing Lines Connection for Read Operation

Fig. 4.2.3 - Memory Addressing for Read Operation

kept at high (logic 1) by the 12th counter output. The rest of the counter outputs are connected to the memory addressing lines as shown in Figure 4.2.3 b. Again this is the result of the pattern of the memory read addressing sequences (see Figure 4.2.3 a).

From the above discussion it follows that the mode of connection of the memory addressing lines to the counter outputs, determines the symbol read in and read out functions of the block interleaver. It is also obvious that for continuous flow of the information symbols through the interleaver, a second interleaver memory has to be used as shown in Figure 4.2.4. As can be seen from this figure, when the ith interleaving matrix is writen in the interleaver memory 1, the (i-1) interleaving matrix is read out from the interleaver memory 2. Alternatively, when the (i+1) interleaving matrix is writen in the interleaving memory 2, the ith interleaving matrix is read out from the interleaver memory 1. This continuous alternation between the write and read operational modes of the two interleaver memories is realized by the W/R control line (commutator). If the W/R line of memory 1 is connected directly to the 12th output of the free running counter, the W/R control line of memory 2 is connected to the same counter output through an inverter. This is clearly shown in Figure 4.2.5, where the analytical schematic diagram of the (32,64) block interleaver is illustrated. As can be seen from this diagram, the

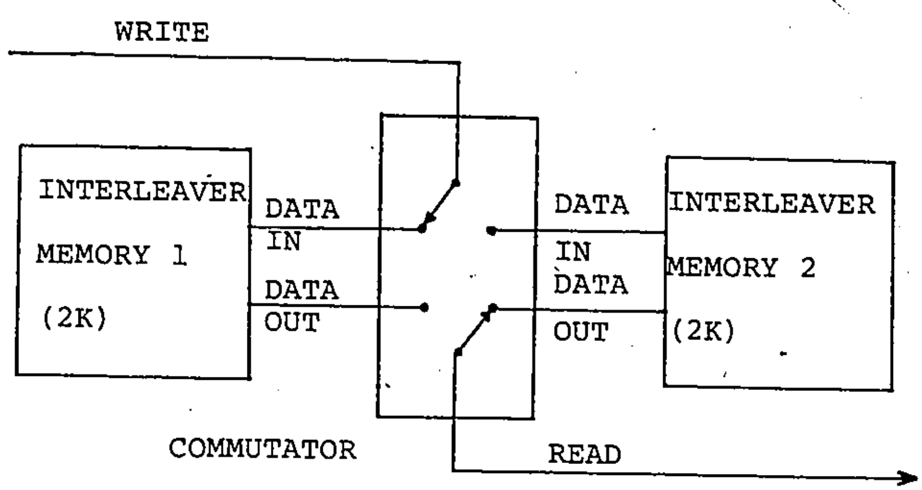


Fig. 4.2.4 - Simplified Block Diagram of the (32,64) Block Interleaver

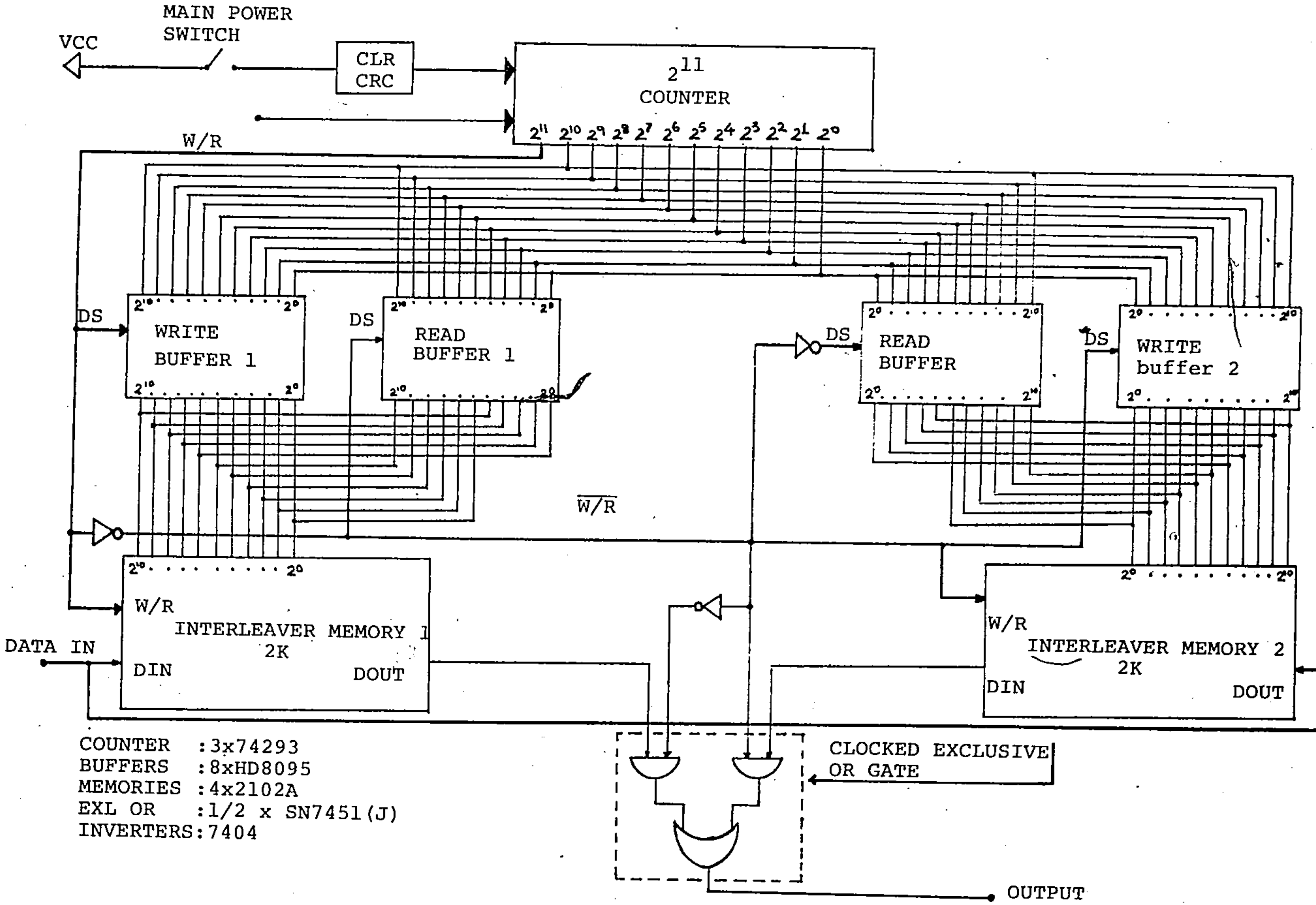


Fig. 4.2.5 - Schematic Diagram of the (32,64) Block Interleaver

addressing lines of both interleaver memories are straightforward connected to the outputs of the free running counter through write buffers. On the other hand, the addressing lines of both memories are also connected to the outputs of the counter in the multiplexing mode (shown in Figure 4.2.3 b), through read buffers. The counter generated addressing sequences for both the write and read modes, are processed through this type of buffers (HD8095), only if their disable inputs (DS) are kept at low (logic 0), by the W/R control line. When the W/R control line is at high (logic 1), the buffer outputs are shown high impedance ($Z_0 \simeq \infty$) to the memory addressing lines, which actually corresponds to disconnection of the addressing lines from the buffer outputs.

Figure 4.2.6 shows the schematic diagram of the 2^{11} free running counter. To reset the outputs of this type of counter (74293) to the low level (logic 0), an instantaneous transition from the high level state to the low level state, of the counter resetting inputs (R_0), is required. As can be seen from Figure 4.2.6, this reset operation is realized by a Schmitt Triger, whose inputs are connected through the main power switch to the power supply line (+Vcc). Therefore, whenever the main power switch is switched ON, the free running counter starts counting from the 00000000000 state (see Figure 4.2.7).

Suppose that the system's main power switch in Figure

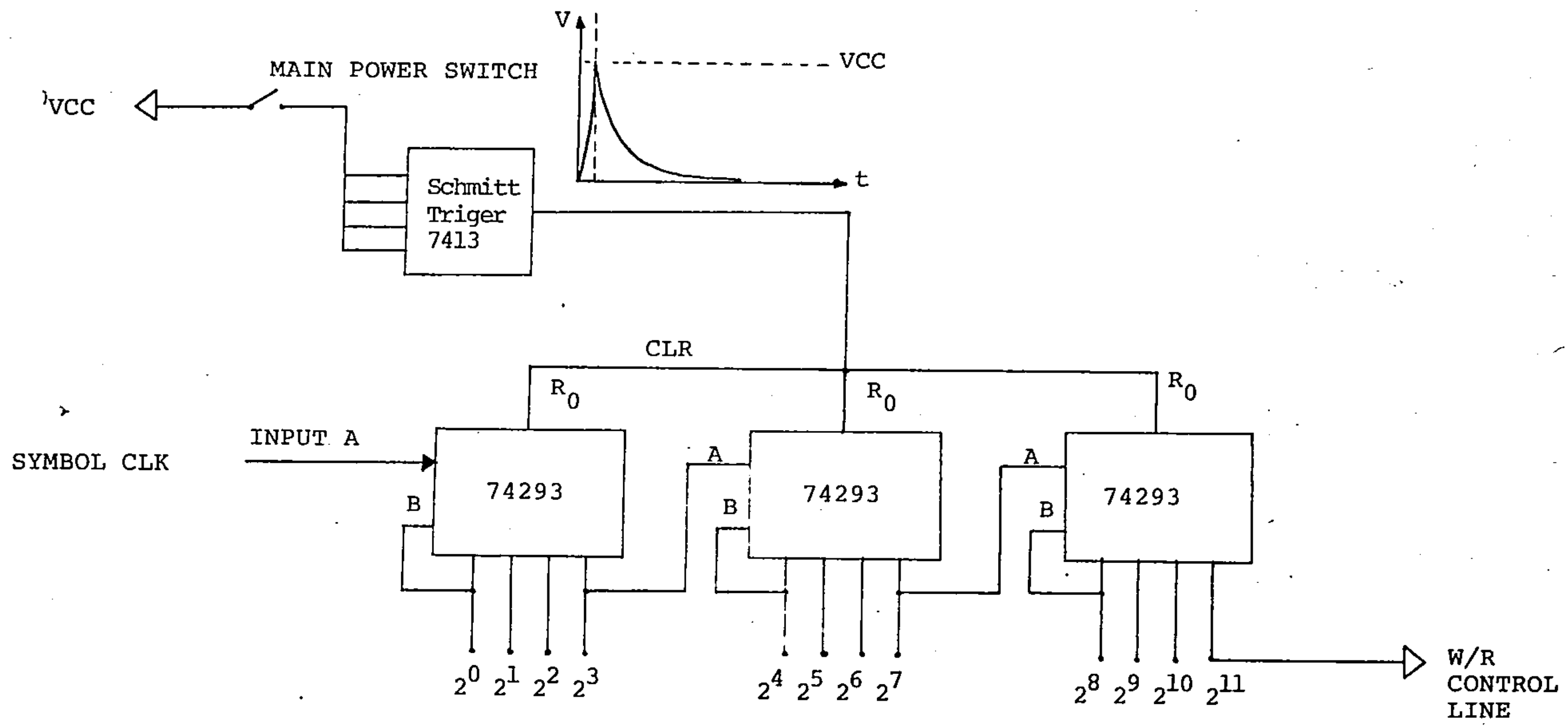


Fig. 4.2.6 - Free Running Counter

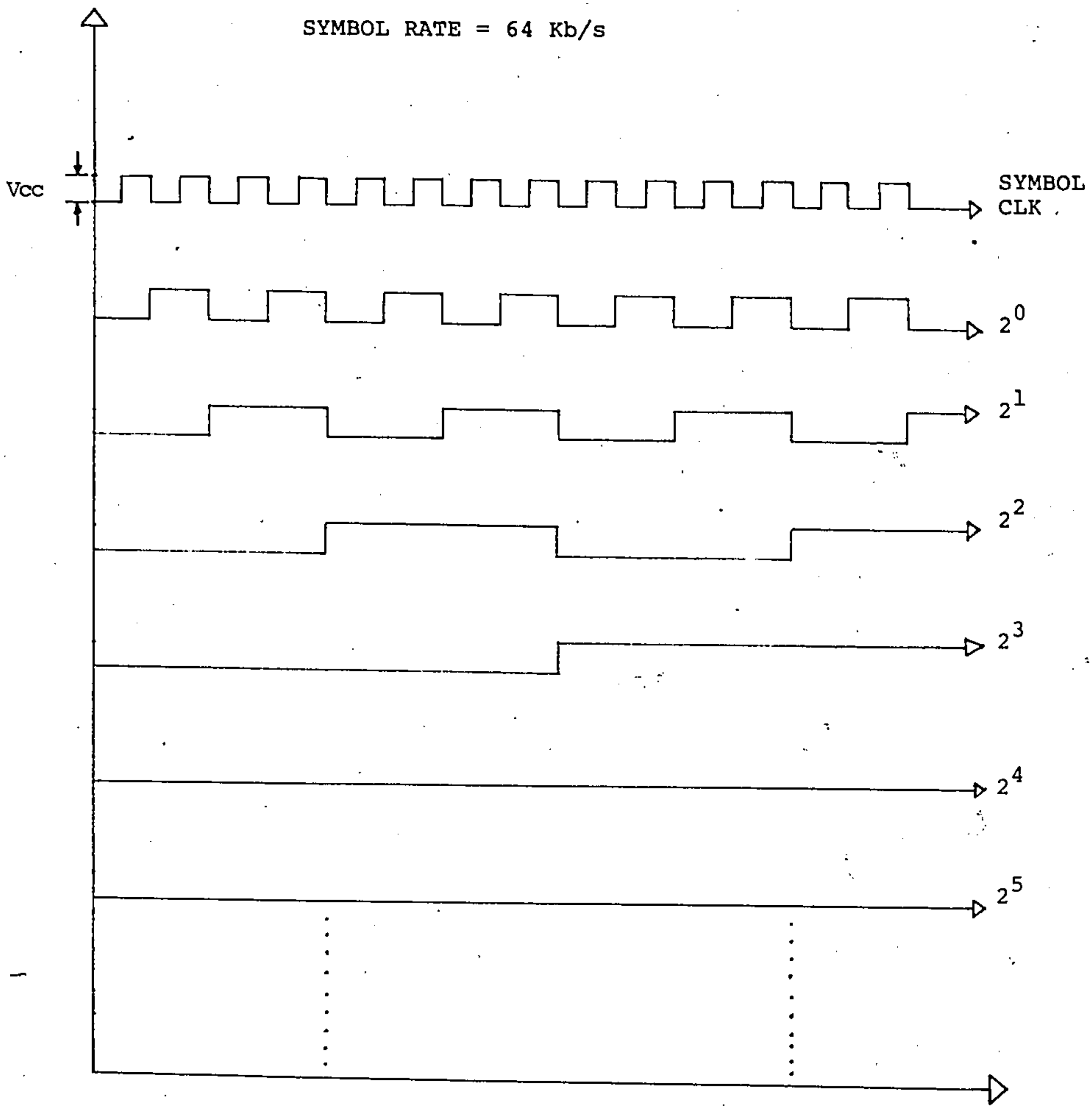


Fig. 4.2.7 - Counter Timing Diagram

4.2.5 is randomly switched ON. Then the counter starts counting the coming in the interleaver information symbols. Since for the first 2048 information symbols, the W/R control line is kept at low (because the 12th counter output is at low), memory 1 is in the write mode with its addressing lines connected to the outputs of the write buffer. On the other hand, memory 2 is in the read mode with its addressing lines connected to the outputs of the read buffer. When the W/R control line is changing state (during the next group of 2048 information symbols) memory 1 is in the read mode with its addressing lines connected to the read buffer. At the same time memory 2 is in the write mode with its addressing lines connected to the write buffer. The W/R control line controls also the operation of the output clocked exclusive OR gate. Therefore, whenever the W/R control line is at low, interleaved information symbols are read out from memory 2 and processed through the OR gate to the interleaver's output. On the contrary, whenever the W/R control line is at high, interleaved information symbols are read out from memory 1 and processed through the OR gate to the interleaver's output.

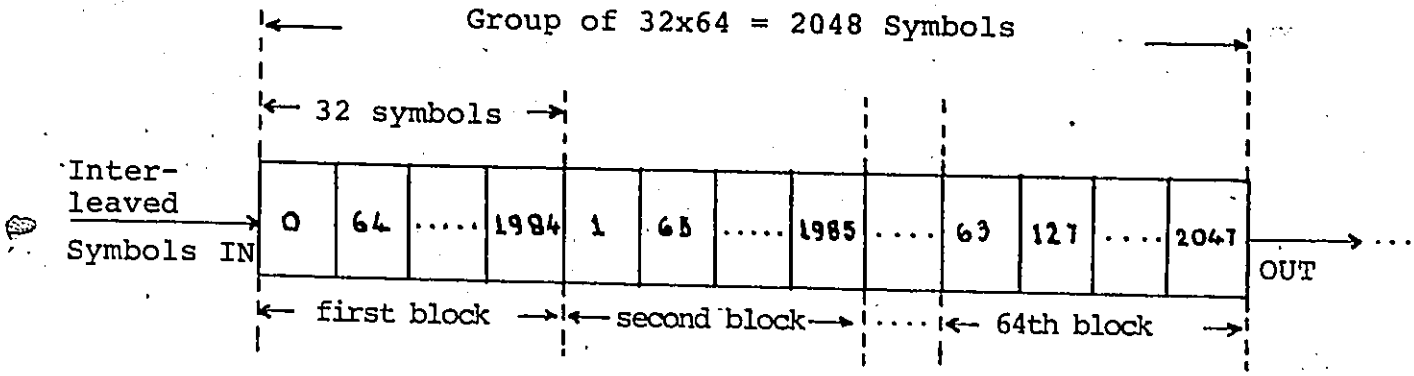
4.3 The (64,32) Block De-interleaver

As already stated in Chapter I, the de-interleaver at the receiving end is a (64,32) block interleaver. For the implementation of this interleaver, an almost similar configuration like the one described in Section 4.2 was used.

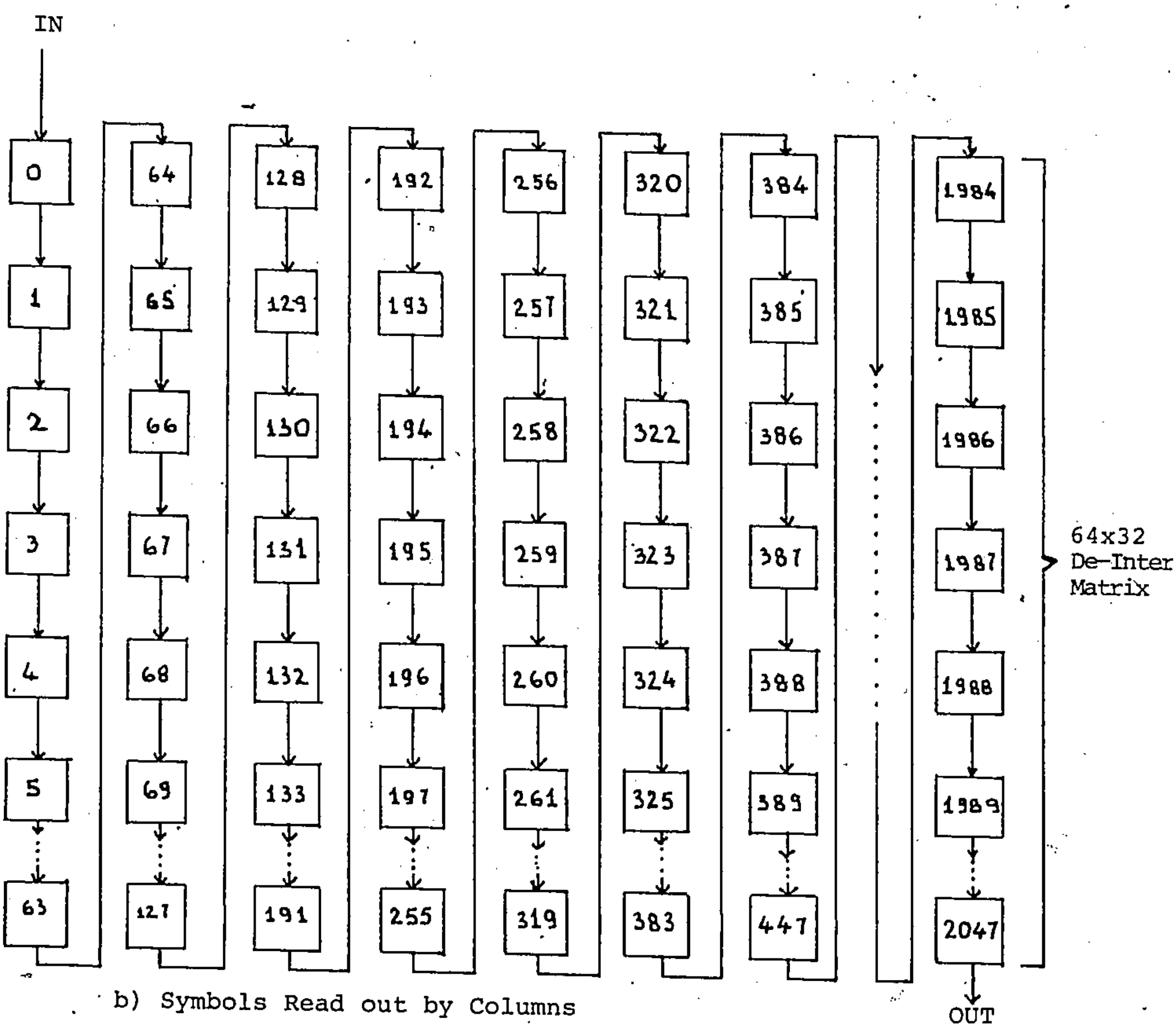
The few differences between the two configurations are resulted from the de-interleaver functional diagram given in Figure 4.3.1. According to this figure, the coming in the de-interleaver interleaved symbols, are first divided into groups of 2048 symbols. These groups are consisted of 64 blocks of 32 symbols per block. Again, these groups form the de-interleaving matrices of 64 rows and 32 columns. Then the symbol read out process starts, by reading symbols out by columns rather than by rows. In this way the recceived symbols are taken apart by 64 symbols and reordered in their original positions.

From the above, it follows that the symbols of any possible error-burst, caused to the received symbols by the channel disturbances, are appearing like random errors to the following de-interleaver decoder. The statistical occurrance of these errors (2 errors in a block of 128 information symbols) is well above the error correcting capabilities (2 errors in a block of 80 information symbols [7]) of the threshold decoder used for the system performance evaluation (see Chapter V). Therefore, error-bursts of length up to 32 symbols, are expected to be corrected by the interleaved codec, considering that the decoder does not introduce random errors to the decoded symbol sequences [7].

From Figure 4.3.1 it follows that the symbol read in (write) and read out (read) operations are done by the de-interleaver memories as shown in Figure 4.3.2. Therefore



a) Symbols Read in by Rows



b) Symbols Read out by Columns

Fig. 4.3.1 - (64,32) Block De-interleaving

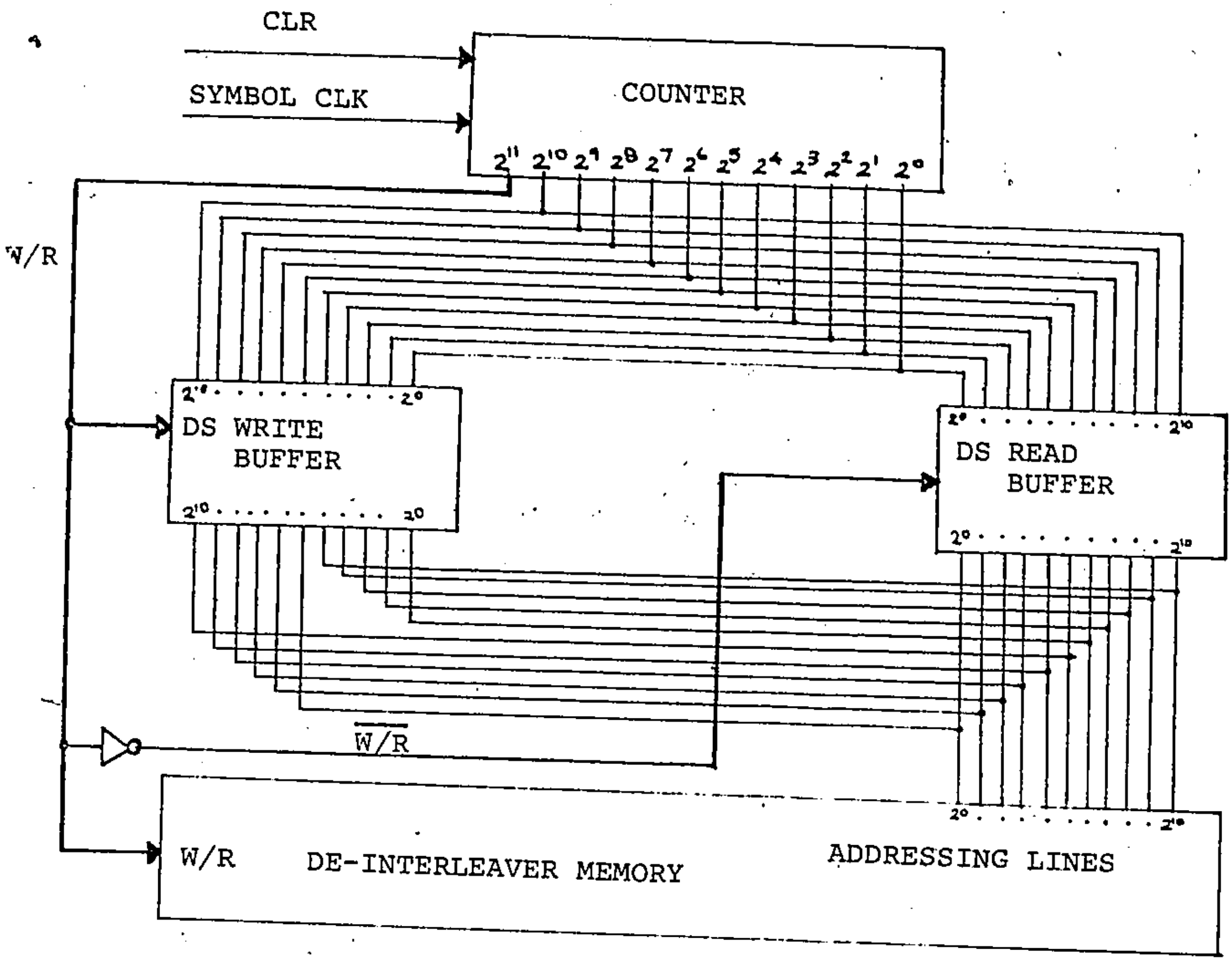


Fig. 4.3.2 - De-interleaver Memory Addressing Line Connections for Read in and Read out Operations

the schematic diagram of Figure 4.2.5 can also represent the schematic diagram of the (64,32) block de-interleaver, after the modifications on the addressing line connections in both memories, as shown in Figure 4.3.2.

For the synchronous operation of the interleaver and de-interleaver, the de-interleaver counter was enslaved to the interleaver counter. This was realized by the configuration shown in Figure 4.3.3. As can be seen from this figure and the timing diagram of Figure 4.3.4, the de-interleaver counter outputs are set at low (CLEARED) whenever all the interleaver counter outputs are set at high (logic 1).

The FF_1 JK flip-flop is incorporated in the synchronization configuration in order to anticipate the delay caused to the information symbols by the flip-flop of the threshold comparator (see Figure 1.1.1) [30]. In this way, information symbol correspondence between the interleaver and de-interleaver was maintained.
This configuration was implemented only for the specific experimental requirements of this thesis. In a real application of this interleaving system, a more complicated synchronization process is required [39]. The description of such a synchronization process is beyond the goals of this thesis.

In conclusion, both the (32,64) block interleaver and (64,32) block de-interleaver were buffered to a 50 output

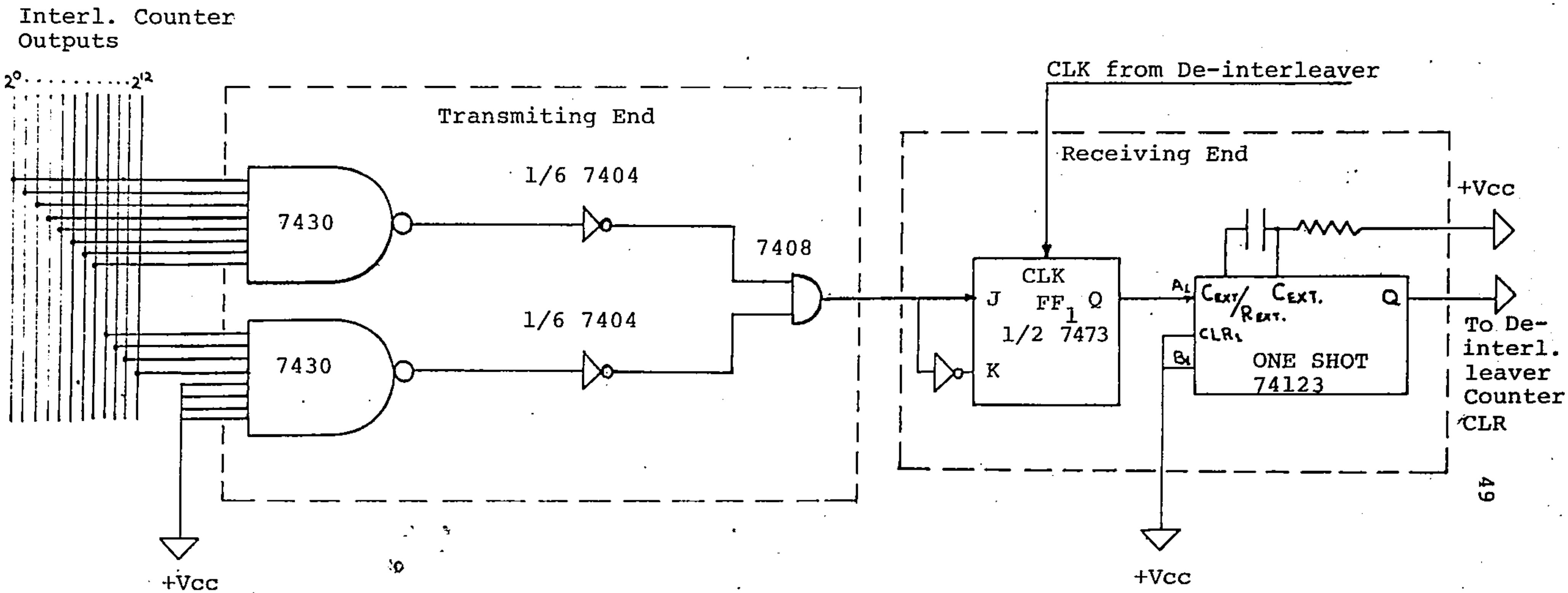


Fig. 4.3.3 - De-interleaver Counter Synchronization Configuration

	t_1	t_2
Interl. Counter	111111111111	000000000000
De-Inter. Counter		000000000000

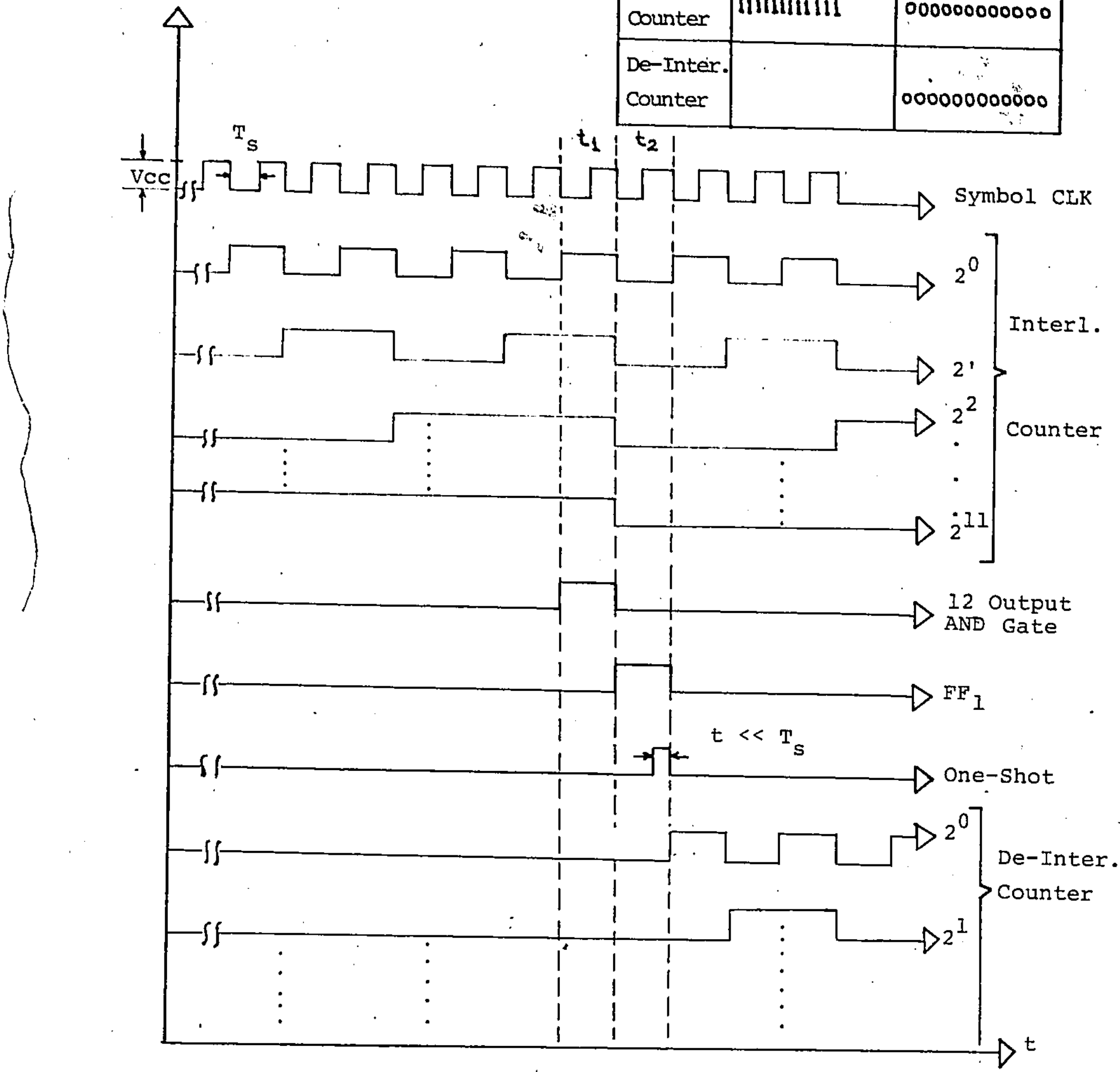


Fig. 4.3.4 - Synchronization Timing Diagram

impedance. In this way all the impedance missmatching effects were eliminated, when the interleaving system was interfaced with other configurations and measurement instruments as described next in Chapter V.

CHAPTER V

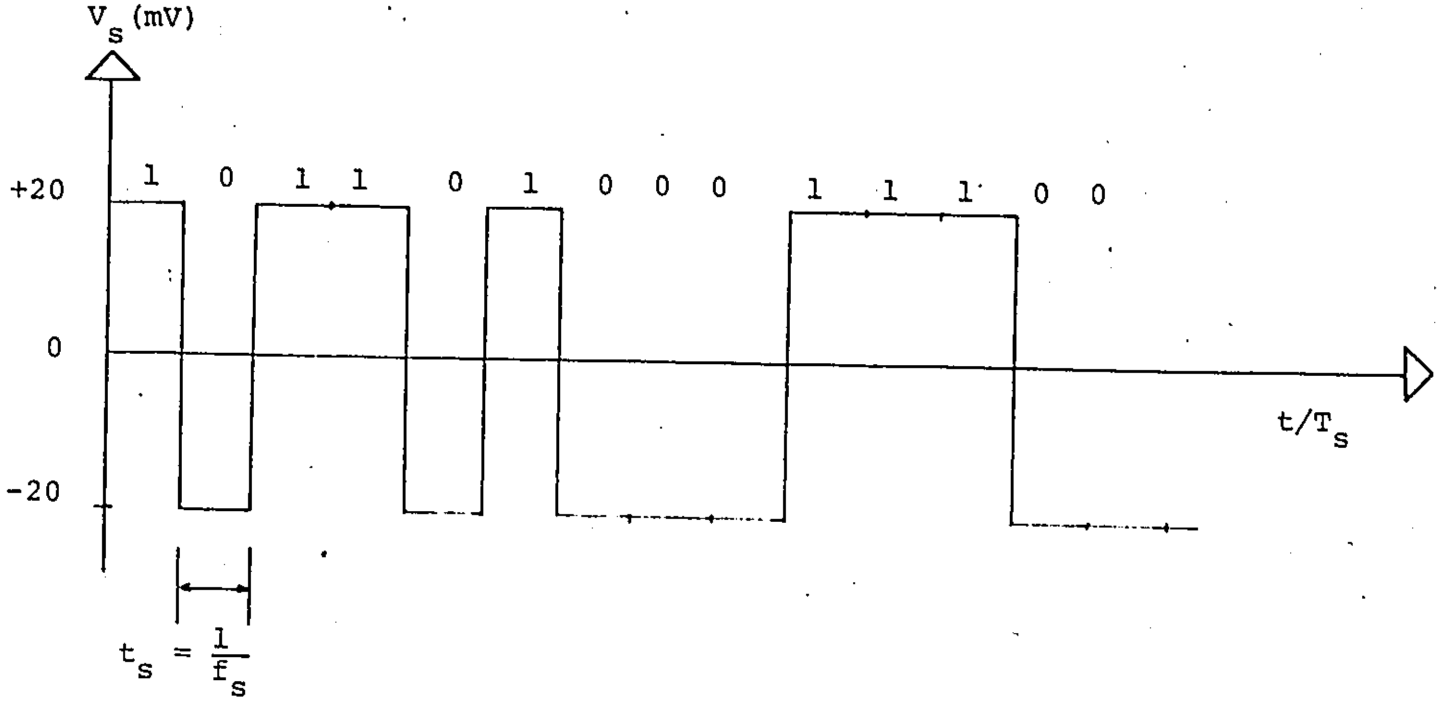
PERFORMANCE EVALUATION OF THE INTERLEAVED 3/4 RATE CONVOLUTIONAL CODEC

5.1 General

This chapter is divided into two major sections. The first section reports the BER performance evaluation of the 3/4 convolutional codec along with the (32,64)/(64,32) interleaving system, in a sinusoidal interference (SI) environment and in a SI plus AWGN environment. In both environments the system BER performance is considerably improved due to interleaving.

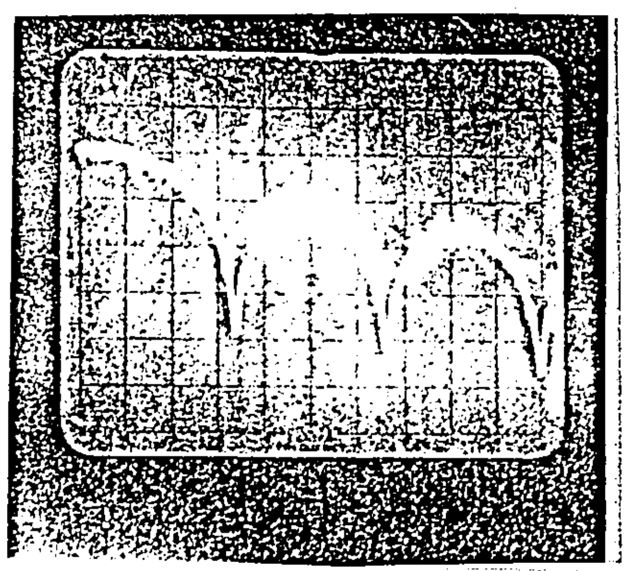
The second section of this chapter deals with the BER performance evaluation of the same system in an impulsive noise environment. Again significant improvements of the system BER performance were obtained, due to ~~the error~~-burst correcting abilities of the interleaved 3/4 convolutional codec.

In all the measurements done for the purpose of performance evaluation in the various interference environments mentioned above, a Pseudo Random Binary Sequence (PRBS) of length $2^{15}-1$ was used as a Non-Return to Zero (NRZ) baseband signal (see Figure 5.1.1) [49]. In the measurements where uncoded signals were used, the symbol rate of this NRZ was set to $f_s = 64$ kb/s. On the other hand when coded and interleaved signals were used, the NRZ source symbol rate was



a) Time Domain Representation

(Power Spectral Density)



Vertical: 10 dB/div.
Horizontal: 20 kHz/div.

b) Frequency Domain Representation

Fig. 5.1.1 - Time and Frequency Domain Representation of the NRZ Baseband Signal (unfiltered)

set of f_s = 48 kb/s and therefore the resulting encoded rate was always equal to 64 kb/s [7].

The filtering in all the experimental set-ups given in this chapter was done by 4th order Butterworth filters with a steepness factor of 80 dB/decade (4 x 20 dB/decade)* [51]. For all RMS voltage measurements a true RMS voltmeter (HP3400) was used, while for the Bit Error Rate (BER) measurements an error detector (HP3761A) was used. Finally, note that for all the measurements included in this chapter, an appropriate ± 5% approximation has to be considered (mostly due to observation inaccuracies) even though each final measured value was averaged out of three readings.

* When a 4th order Butterworth filter is used as receiving filter in communication system, the receiving noise bandwidth is given by [51]:

Noise Bandwidth = 1.05 x Signal Bandwidth

As a consequence throughout all the measurements, reported in this chapter, the noise bandwidth of the receiving filter, is considered equal to the signal bandwidth of the filter (-3 dB bandwidth), involving an acceptable measurement approximation.

5.2 Performance in a SI and in an AWGN Plus SI Environment

a) A General Overview of SI

In general, a single-frequency sinusoidal interference causes error-bursts in a Non Zero Return (NRZ) signal, if at the sampling instant the instantaneous value of the interference is greater than the peak value of the NRZ signal and opposite in sign.

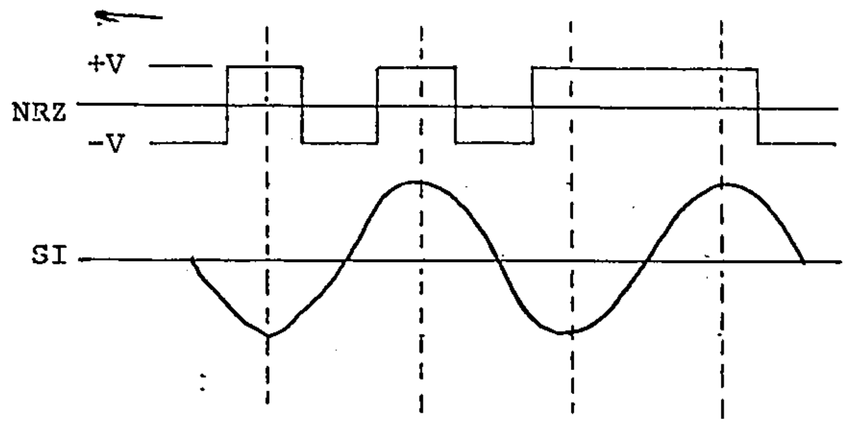


Fig. 5.2.1 - SI Corrupting NRZ Signal

The probability of error of a NRZ signal disturbed by a single-frequency SI is given in equation (9) [7] [21]:

$$P(e)_{SI} = \begin{cases} 1/2 - \frac{1}{\pi} \sin^{-1}\left(\frac{1}{\sqrt{2}} \frac{S}{I}\right) & \text{for } 0 \leq \frac{S}{I} \leq \sqrt{2} \\ 0 & \text{for } \frac{S}{I} > \sqrt{2} = 3 \text{ dB} \end{cases} \tag{9}$$

where S is the peak value of the NRZ signal and I is the peak value of the SI.

Equation (9) indicates that the P(e) function is independent of the frequency of the interfering sinusoidal wave. Of course the NRZ signal will be disturbed by the SI only if the SI frequency is in the frequency band of the receiving LPF (see Figure 1.1.1).

In the case of simultaneous disturbances of the NRZ signal by many sinusoidal waves,* the combined instantaneous vector of the SI has no longer a constant amplitude. Instead its amplitude varies with the number of the interfering sinusoidal vectors and their relative amplitudes and phases. The vector diagram given below, illustrates the case of multi-SI composition.

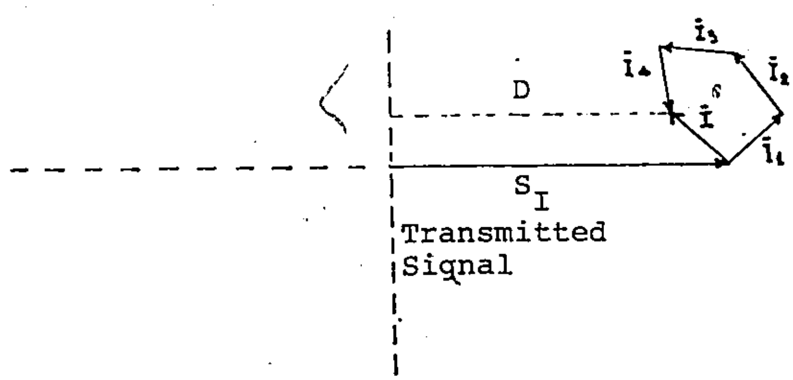


Fig. 5.2.2 - Composition of Multi-SI

In Figure 5.2.2, vector I represents the resultant of a sum of constant vectors. It is assumed that the i-th interference is statistically independent

* In general, SI can be interpreted in a practical communication system as an unmodulated carrier frequency interference (CW).

from all the other interferences. It is also assumed that the phase of the i-th interference is uniformly distributed between 0 and 2π.

The signal-to-interference ratio (S/I) in a multi-SI environment, is defined as the ratio of the signal power to the average power total of all the interfering waves.

b) Performance Evaluation

The experimental set-up of the system BER performance evaluation, in the SI and in the SI plus AWGN environments is given in Figure 5.2.3. The measured eye diagrams, obtained with and without the presence of SI at the output of the receiving LPF, are shown in Figure 5.2.4. Since the encoded symbol rate is 64 kb/s, the receive LPF bandwidth (40 kHz) is greater than the Nyquist filtering limit [6] and therefore Intersymbol Interference (ISI) problems were avoided.

The performance of the interleaved 3/4 convolutional codec in a single-frequency SI and in a multi-frequency SI environment, as compared to the uninterleaved case, is given in Figure 5.2.5. As observed, the uninterleaved codec is unable to perform properly in these environments. On the other hand the performance of the interleaved codec has been improved by orders of magnitude, in comparison with the uninterleaved case. Since interleaving does not add any redundancy to the transmitted symbols [5], it does

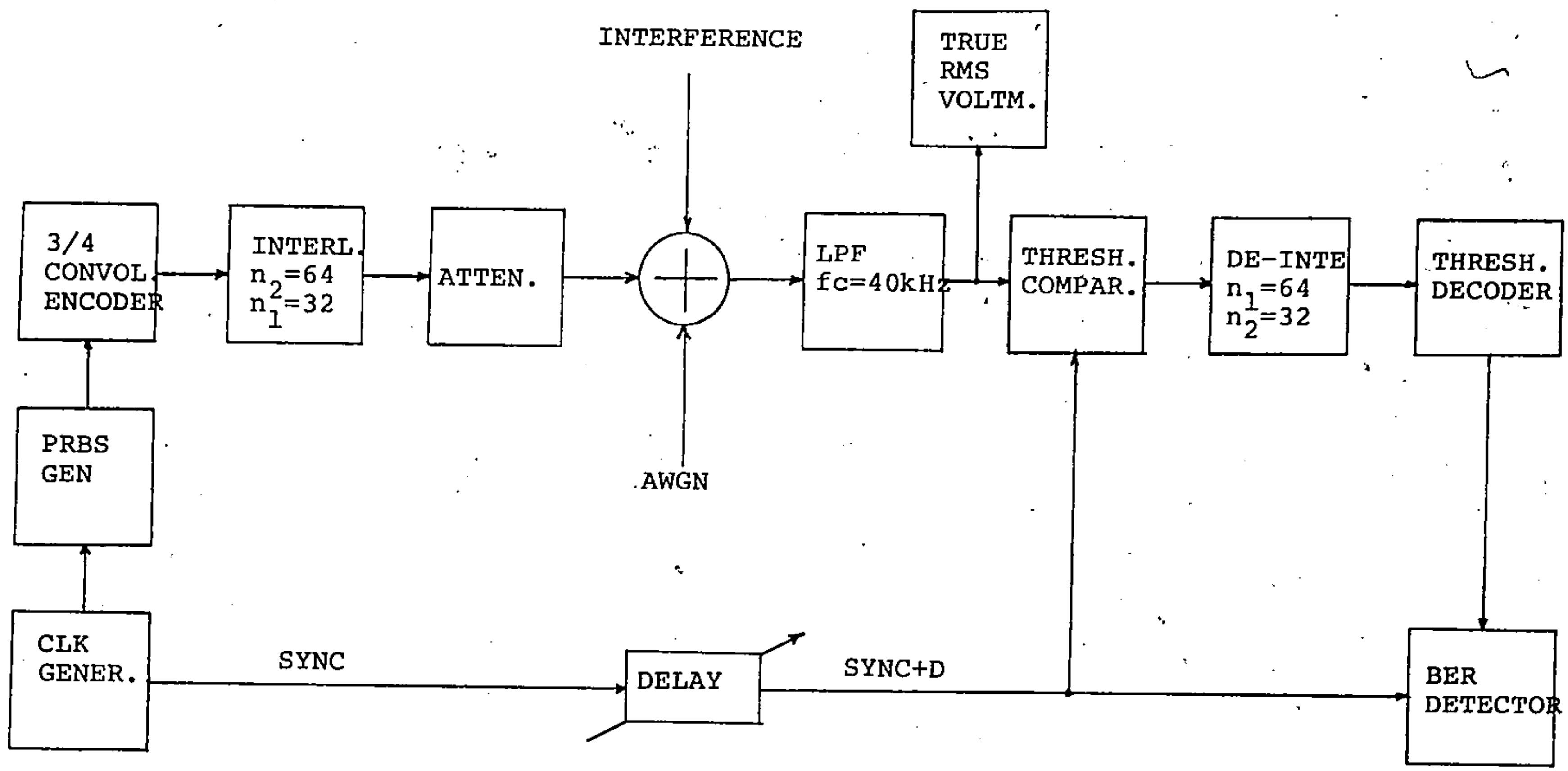
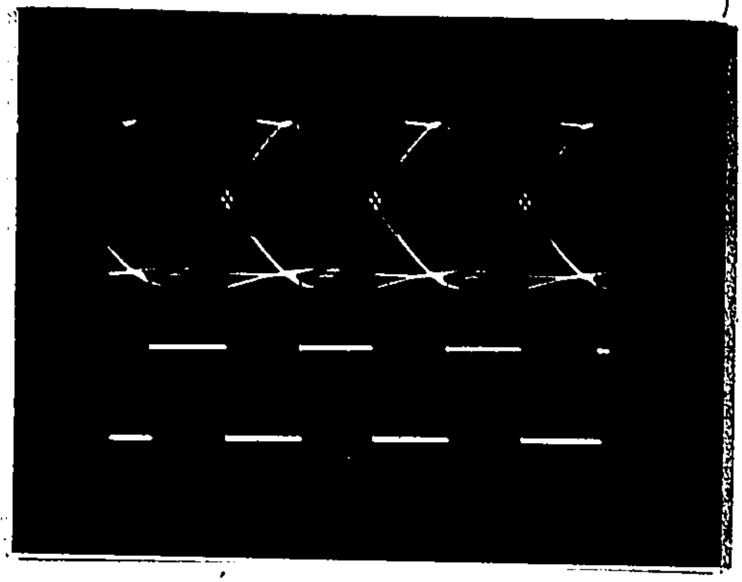
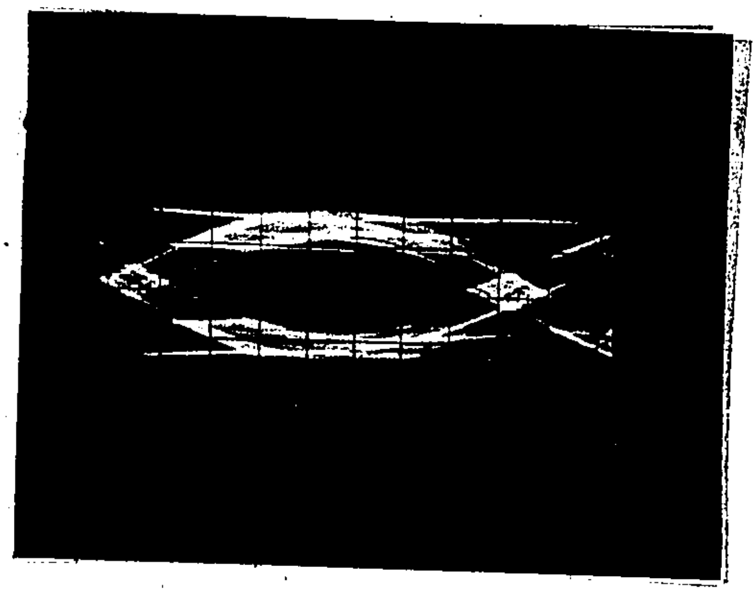


Fig. 5.2.3 - BER Performance Evaluation Set-up for Baseband Measurements



Horizontal: 5μsec/div.

a) Trace A: Interference free eye diagram
Vertical: 10 mV/div
Trace B: 64 Kb/s clock
Vertical: 1 V/div



b) Eye diagram of signal plus SI (f_{SI} = 1 kHz; S/I = 3.5 dB).

Fig. 5.2.4 - System Eye Diagrams With and Without the Presence of SI.

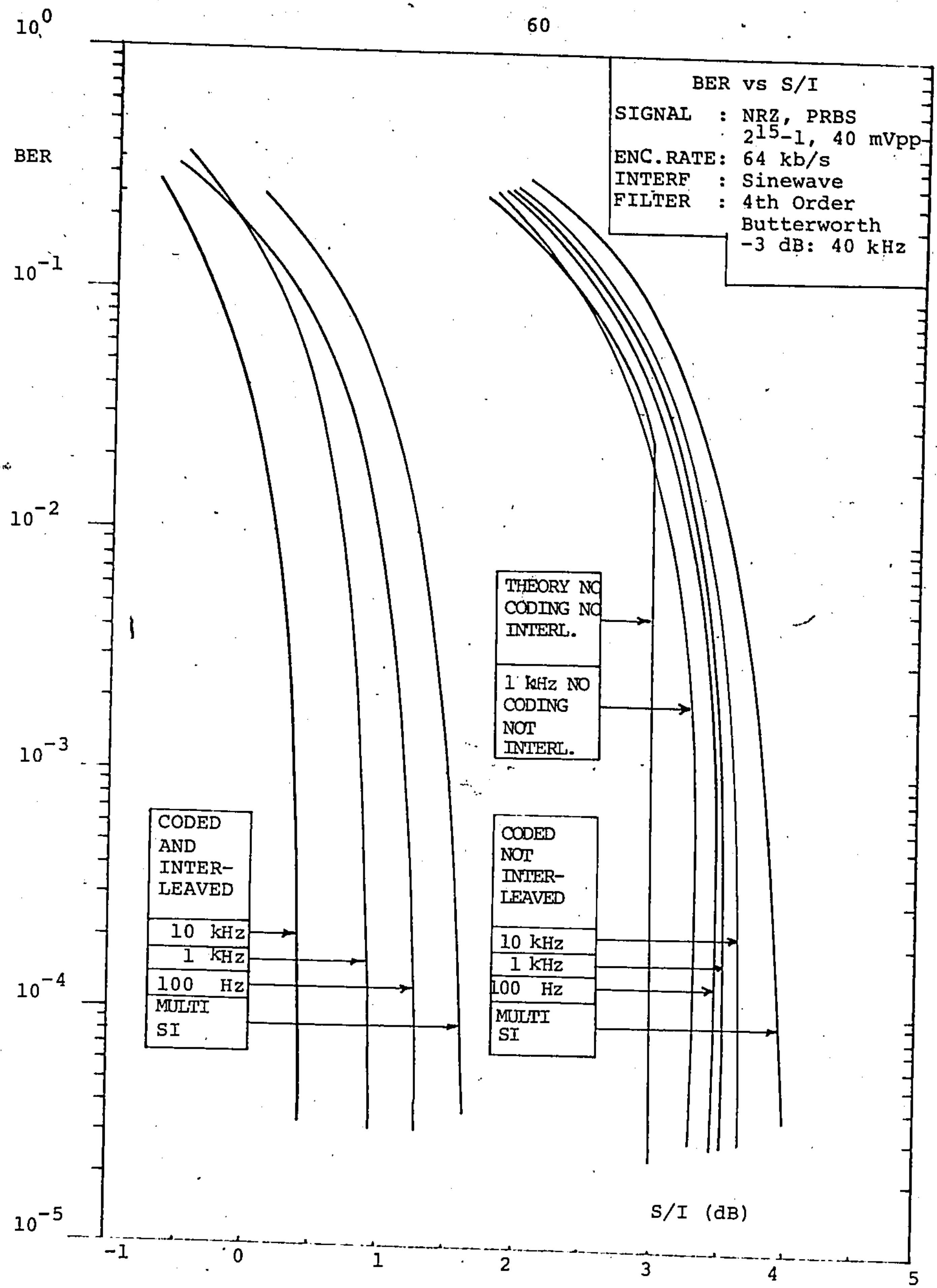
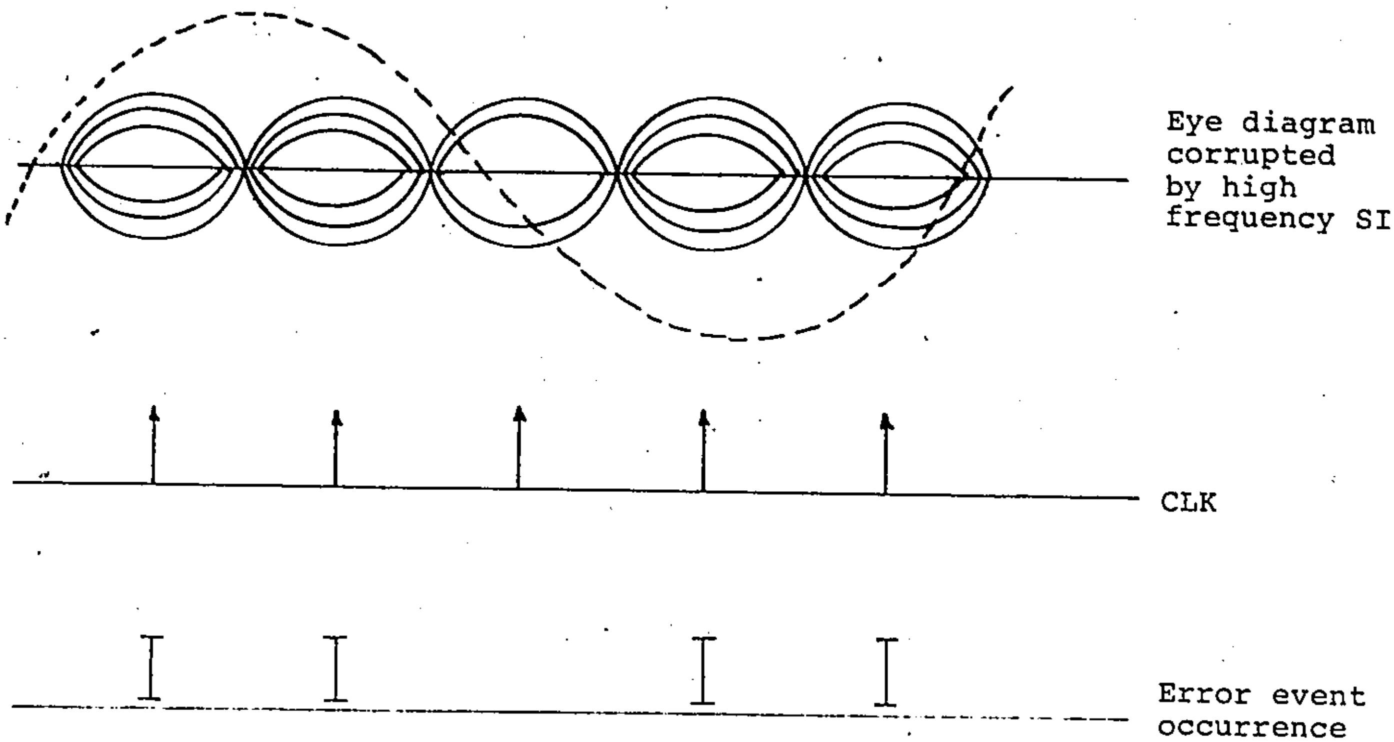


Fig. 5.2.5 - Measures BER Performance in SI Environment

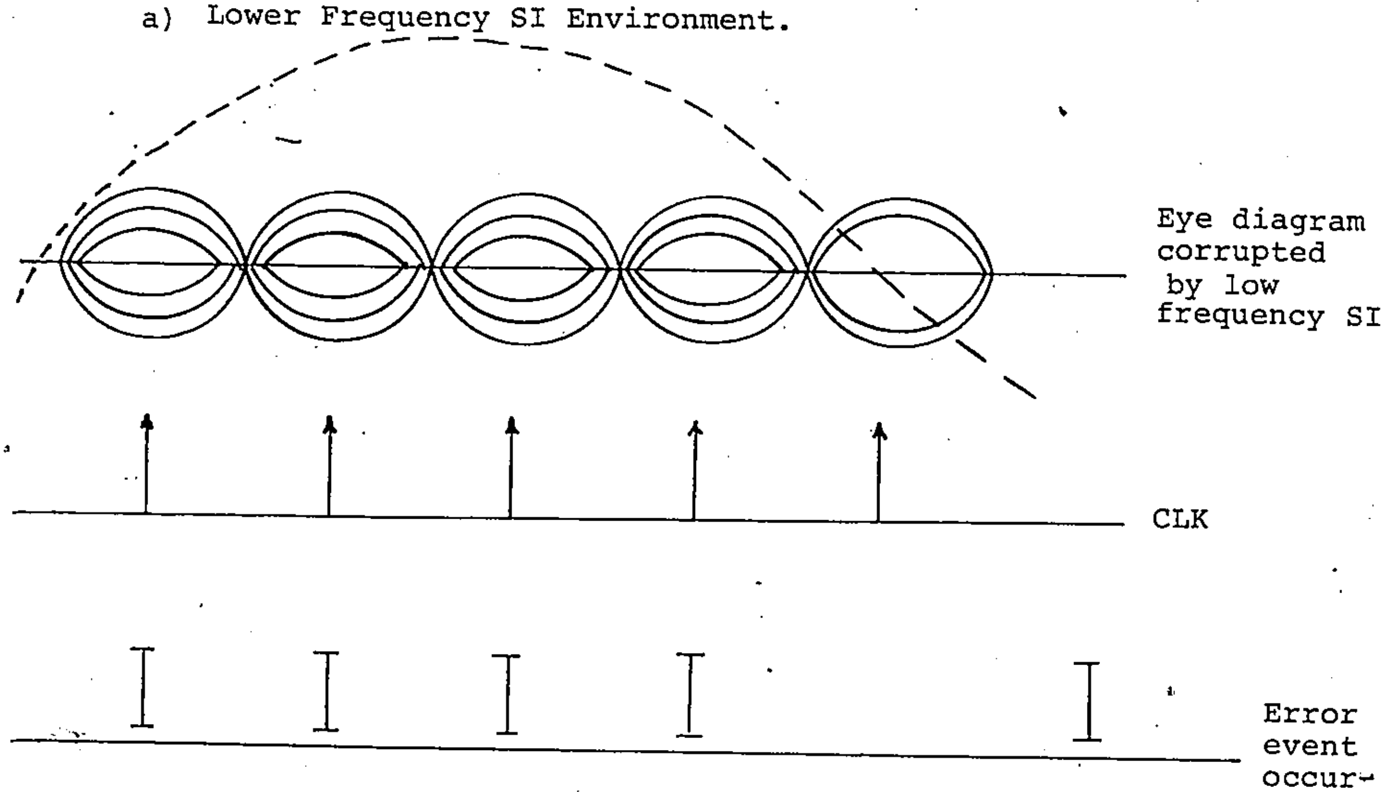
not process any inherent error correcting capability. Thus the significant improvement obtained when interleaving is used in conjunction with random error correcting coding, indicates that the error occurring on the channel have a burst type characteristics. In addition, as can be seen from Figure 5.2.5 the interleaved codec performance is better in the case of 10 kHz SI environment, than in the case of 100 Hz SI environment. This experimental result can be explained as follows.

Figure 5.2.6 illustrates the error-burst generation mechanism, caused by two different frequency sinusoidal waves. As observed, in the case of the higher frequency SI often occurring short error-bursts are caused to the information symbol stream. On the other hand, in the case of lower frequency SI longer error-bursts are caused to the information symbol stream. This difference in the length of error-bursts caused to the information symbol stream by the different frequency SI environment, is probably the main reason of the better performance of the interleaved 3/4 rate convolutional codec in the higher frequency SI environment. Actual measurements on these error event statistics were not undertaken, since they would have required a fairly sophisticated data acquisition system.

Finally, from Figure 5.2.5 is resulted that both the interleaved and not interleaved system BER performance in a combined multi-SI environment (100 Hz + 1kHz + 10 kHz),



a) Lower Frequency SI Environment.



b) Higher Frequency SI Environment.

Fig. 5.2.6 - Error Event Occurrence due to SI

indicates a small degradation in comparison with the corresponding system BER performance in a single frequency SI environment. This degradation can be explained by the variations of the amplitude of the composed multi-SI, due to the different amplitudes and phases of the component interfering sinusoidal vectors (see Figure 5.2.2).

Figure 5.2.7 illustrates the BER performance evaluation of the interleaved 3/4 rate convolutional codec in a SI plus AWGN environment. As observed, for S/I $\geq$ 3 dB, the interleaved and coded BER performance is identical to that of the coded BER performance [7]. On the other hand, for S/I $\leq$ 3 dB the interleaved codec BER performance is improved by 5.7 dB in the area of $P(e) = 10^{-4}$, with respect to the coded BER performance.

These experimental results are actually in agreement with the results of the derivation given in reference [52]. In this reference the derivation of the equation relating the P(e) to S/N of an uncoded NRZ signal corrupted by SI plus AWGN is mostly based in equation (9) [21] given on page 55 of this thesis. According to equation (9), SI is effectively disturbing NRZ signals as long as the S/I is kept below 3 dB.

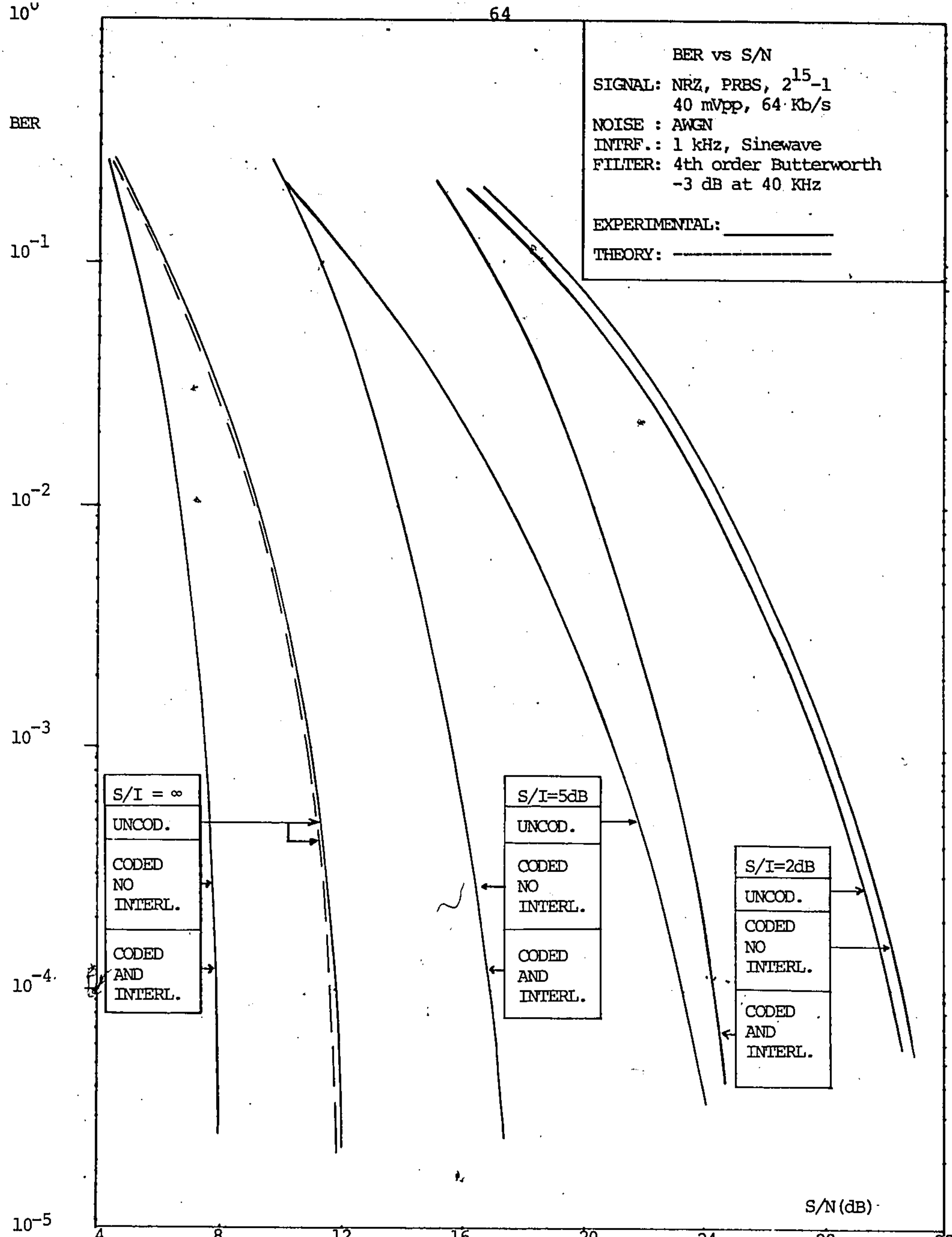


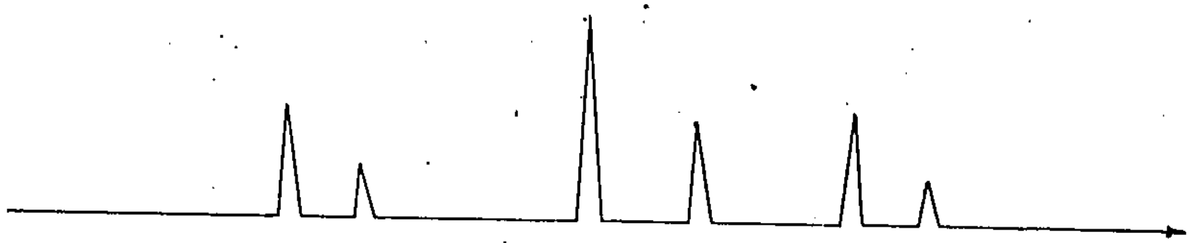
Fig. 5.2.7 - Measured BER Performance in SI + AWGN Environment.

5.3 Performance in an Impulsive Noise Environment

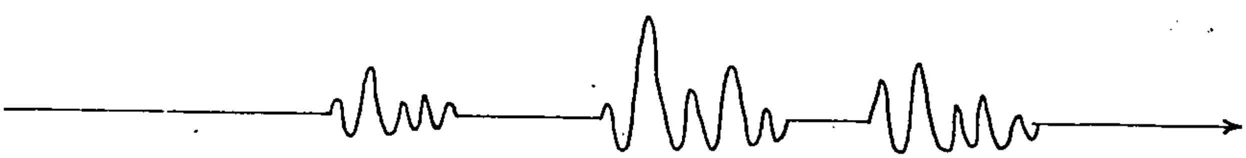
a) An Introductory Overview of Impulsive Noise

In many cases the distribution of the additive disturbances in a communication channel tend to be "impulsive noise" nature. In general, an impulse may be considered as a transient that contains an uniform spectrum over the frequency band for which it is defined [22]; a uniform spectrum requires that all frequencies are present of equal amplitudes over the frequency band. Of course, this is the ideal case of representing an impulse. In general, impulsive noise is the combination of successive impulses, which may have random amplitudes and random time spacing [22]. Usually, in a real impulsive noise environment, the separation of these successive impulses is not possible. Figure 5.3.1 illustrates different successive impulses and the response to these impulses of a typical receiving communication system.

The mechanism by which impulsive noise is produced can usually be traced back to a simple physical source, which produces sequences of usually overlapping impulses. Since the impulses, generated by the noise source, deliver energy in bursts (see Figure 5.3.1) [6], the errors caused to a received digital signal also appear in bursts. In this case, the maximum and average length of the error-bursts and the duration of the error-free intervals, depend upon the duration of the burst energy entering the receiver (after filtering) [6] [48].



(a)



(b)

Fig. 5.3.1 - (a) Impulsive Noise and
(b) System Response to Impulsive Noise

A parameter giving information about the instantaneous peak values of impulsive noise and its average energy, delivered in bursts by the noise source, is the peak factor. The peak factor is defined as follows [2].

$$\text{Peak Factor} = 20 \log_{10} \frac{\text{Vpeak}}{\text{Vrms}} \text{ (dB)} \tag{10}$$

where Vpeak = the peak value of impulsive noise,

Vrms = the rms value of impulsive noise.

From equation (10) it follows that the higher the value of the peak factor, the longer the error-bursts created to a received digital signal, exposed to an impulsive noise environment [48].

A mathematical derivation to fit such considerations can become very complicated, especially in the case of impulsive noise consisting of overlapping impulses [28]. However an illustrative mathematical model calculating the probability of error of a baseband NRZ signal exposed in a simplified impulsive noise environment, is given in reference [53].

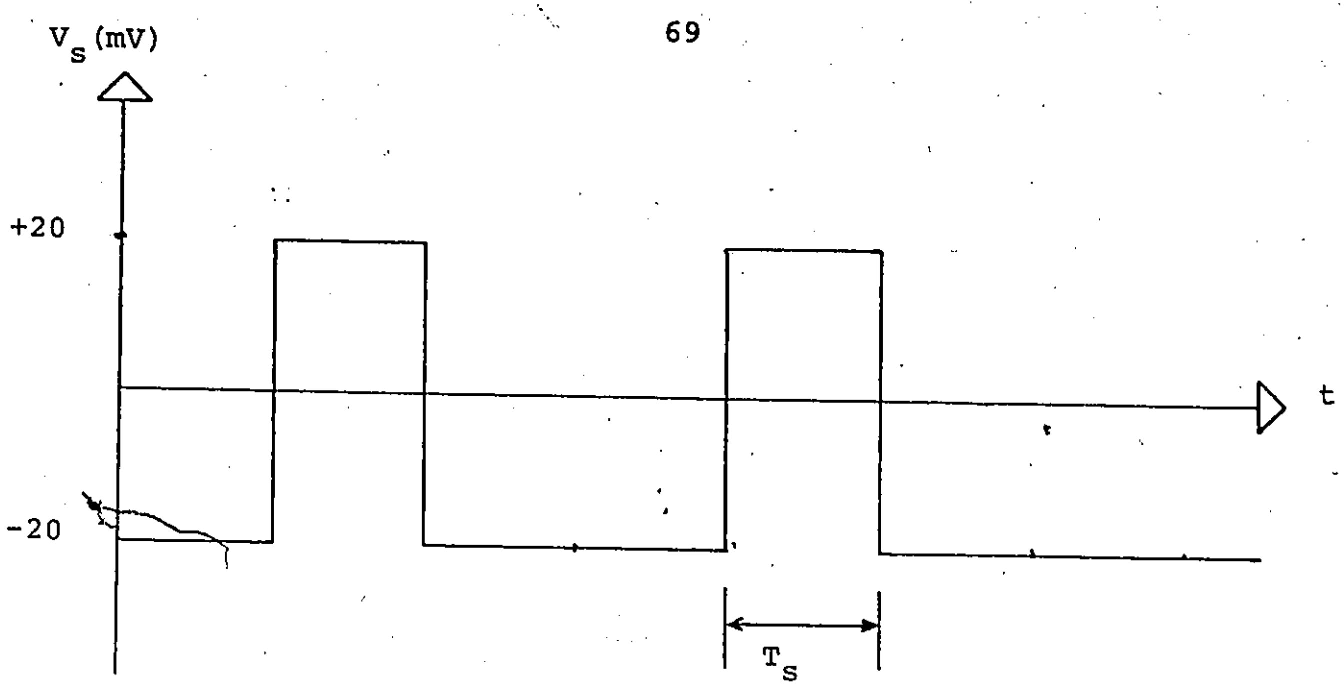
In this section the performance evaluation of the interleaved 3/4 rate convolutional codec in an impulsive noise environment such as the one considered in reference [53], is first given. Next, for a BER performance evaluation in a more realistic impulsive noise environment, the interleaved codec performance is evaluated in a non-Gaussian "spikey" noise environment, realized in the laboratory by an out-of-band intermodulation products model [38].

b) Performance Evaluation in a Simplified Impulsive Noise Environment

The experimental set-up used for the system BER performance evaluation, in this environment, is the same as the one shown in Figure 5.2.3. However, in this case, impulsive noise was considered as the only source of interference.

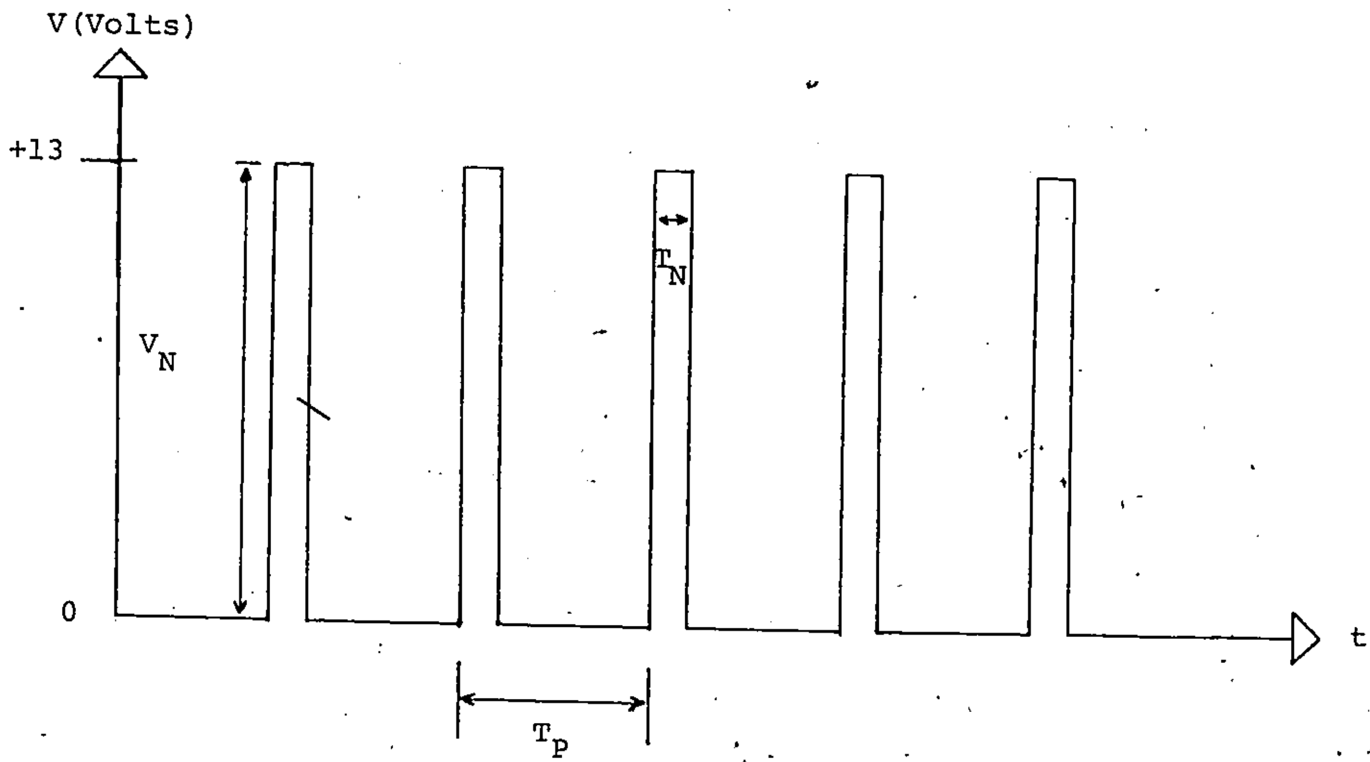
The Impulsive noise used in this scheme, consists of a "train" of periodic, positive, narrow pulses having the same very high amplitude in comparison to the NRZ signal peak-to-peak amplitude, as shown in Figure 5.3.2. As also can be seen from this figure, the duration of the pulses was fixed to T_N = 1 μsec, which is much smaller than the symbol duration (15.6 μsec) of the NRZ signal. Therefore this "train" of periodic pulses, can be easily visualized to be a "train" of disturbing impulses. Also, the term of impulse rate of repetition (νT) per information symbol, defined by equation (11) below [48] [53], will be used.

$$\nu T = \frac{\text{Frequency of Occurrence of Impulses}}{\text{Symbol Rate}} \qquad (11)$$



$$T_s = \frac{1}{f_s} = \frac{1}{64 \times 10^3} = 15.6 \ \mu\text{sec}$$

a) NRZ Signal



$V_N >> V_s$ $\quad$ $T_N = 1 \ \mu\text{sec}$ $\quad$ $V_N(\text{RMS}) = V_N(\text{RMS}) = V_N x T_N$

b) Simplified Impulsive Noise

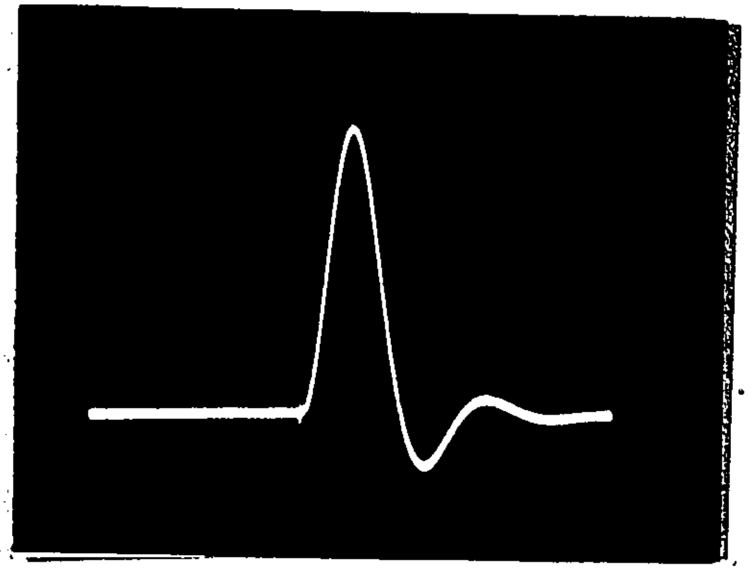
Fig: 5.3.2 - Impulsive Noise Consisting of Periodic Impulses as Compared to NRZ Signal

For the symbol rate of 64 kb/s, Table I gives three different values of νT for three different values of frequency of impulse occurrence. Figure 5.3.3 shows the time and frequency domain representations of the receiving LPF (see Figure 5.2.3) impulse response.

Table I

Frequency of Impulse Occurrence (Hz)	Information Symbols per Impulse Period	νT Impulses per Infor. Symbol
64	1000	10^{-3}
640	100	10^{-2}
6400	10	10^{-1}

The main lobe of the time domain representation of this LPF impulse response, is causing independent errors to the NRZ signal for high S/N. For low S/N the tail of the time domain representation of the LPF impulse response is also causing errors to the NRZ signal (providing that the noise power is kept constant and that no other type of interference is present) and therefore, the errors are more likely to appear in bursts. These considerations on error-burst generation are in agreement with the theoretical considerations of reference [48], since the impulsive noise power and therefore the LPF impulse response depends upon the amplitude of the periodic impulses (see Figure 5.3.2b).

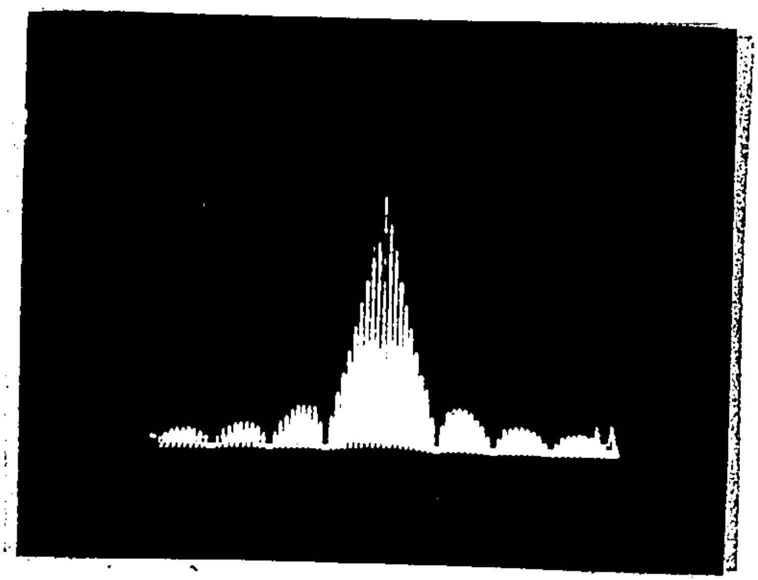


Horizontal
50 μsec/div

Vertical
20 mV/div

a) Time Domain Representation

(Power Spectral Density)

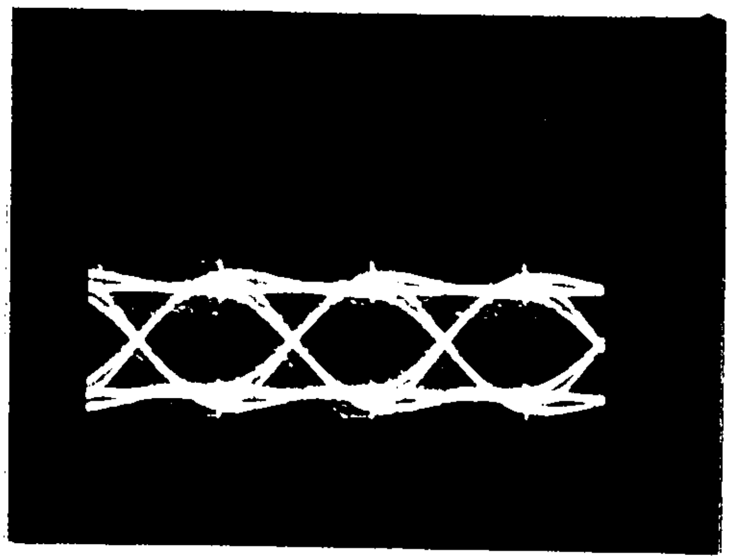


Horizontal
50 KHz/div.

Vertical
10 dB/div.

b) Frequency Domain Representation

Fig. 5.3.3 - LPF Impulse Response (4th Order Butterworth; - 3 dB at 40 KHz)



Horizontal: 5 μsec/div

Vertical : 10 mV/div

Impulse Repetition Rate

$VT = 10^{-1}$

Fig. 5.3.4 - Eye Diagram of Disturbed NRZ Signal by Impulsive Noise

Figure 5.3.5 shows the experimentally obtained BER performance evaluation for a 64 kb/s uncoded baseband NRZ signal, exposed to the impulsive noise environment described above. Of course, to obtain this BER performance the interleaved codec was by-passed from the set-up shown in Figure 5.2.3. As observed from Figure 5.3.5, for the same signal to noise ratio (S/N), higher BER values were obtained in the case of impulsive noise of higher impulse repetition rates than in the case of lower impulse repetition rates. These experimental results are in agreement (within an accuracy of ± 20%; worst case) with the theoretical obtained at the University of Ottawa [53] BER performance evaluation results, shown in Figure 5.3.5 with the dashed line. For these BER performance calculations the same set-up under the same signal and noise conditions was considered, as in the present experimental evaluation.

Figure 5.3.6 shows the experimental BER performance evaluations in the same impulsive noise environment, of the coded and uninterleaved and the coded and interleaved baseband signals, as they compared to the uncoded baseband signal case. For the BER performance evaluation shown in this figure, the impulse repetition rate of the disturbing impulsive noise was fixed at $\nu T = 10^{-1}$ (1 impulse per 10 information symbols). As can be seen, the coded and interleaved system BER performance has been improved by

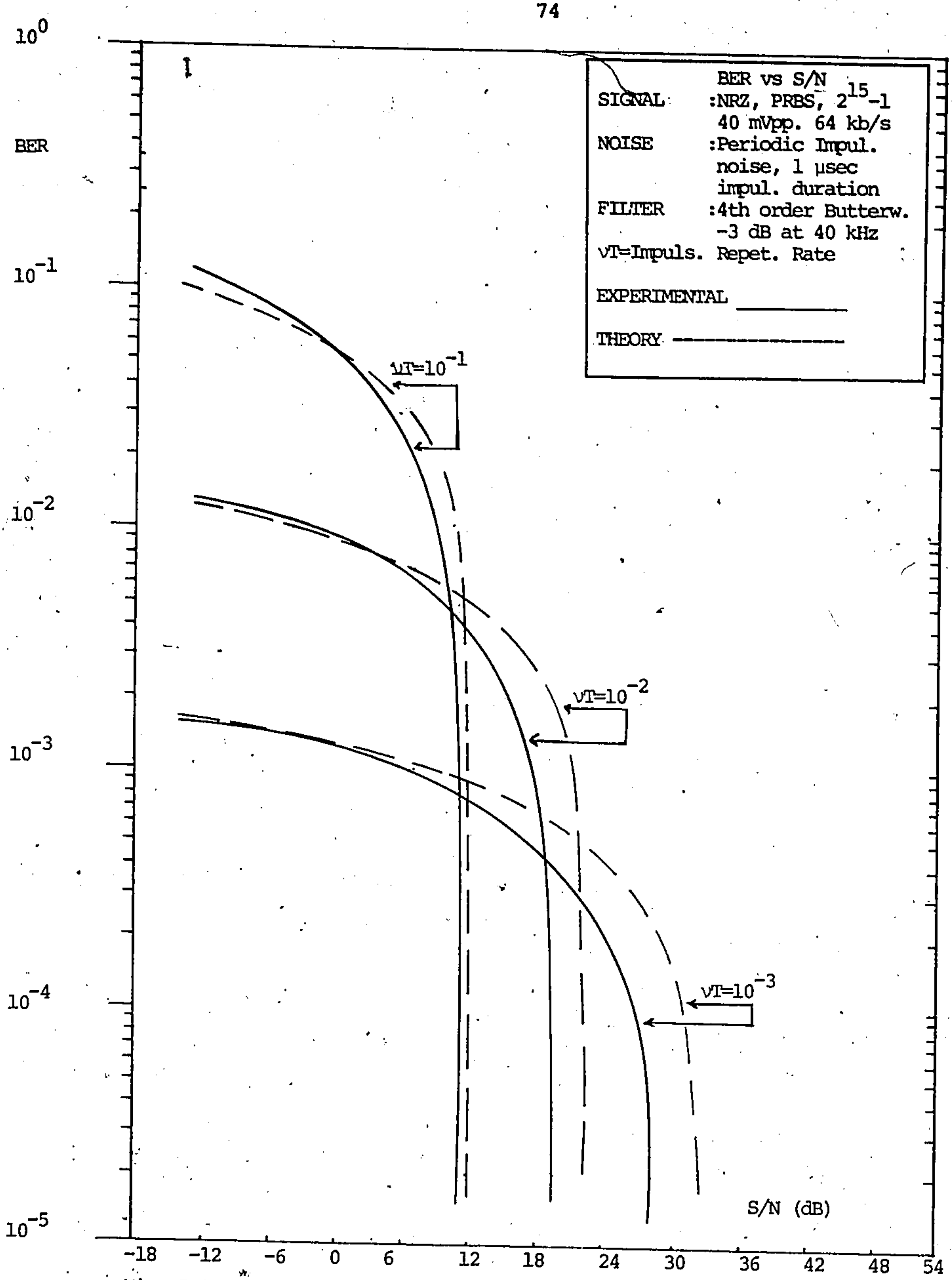


Fig. 5.3.5 – Experimental and Theoretical BER Performance in a Simplified Impulsive Noise Environment

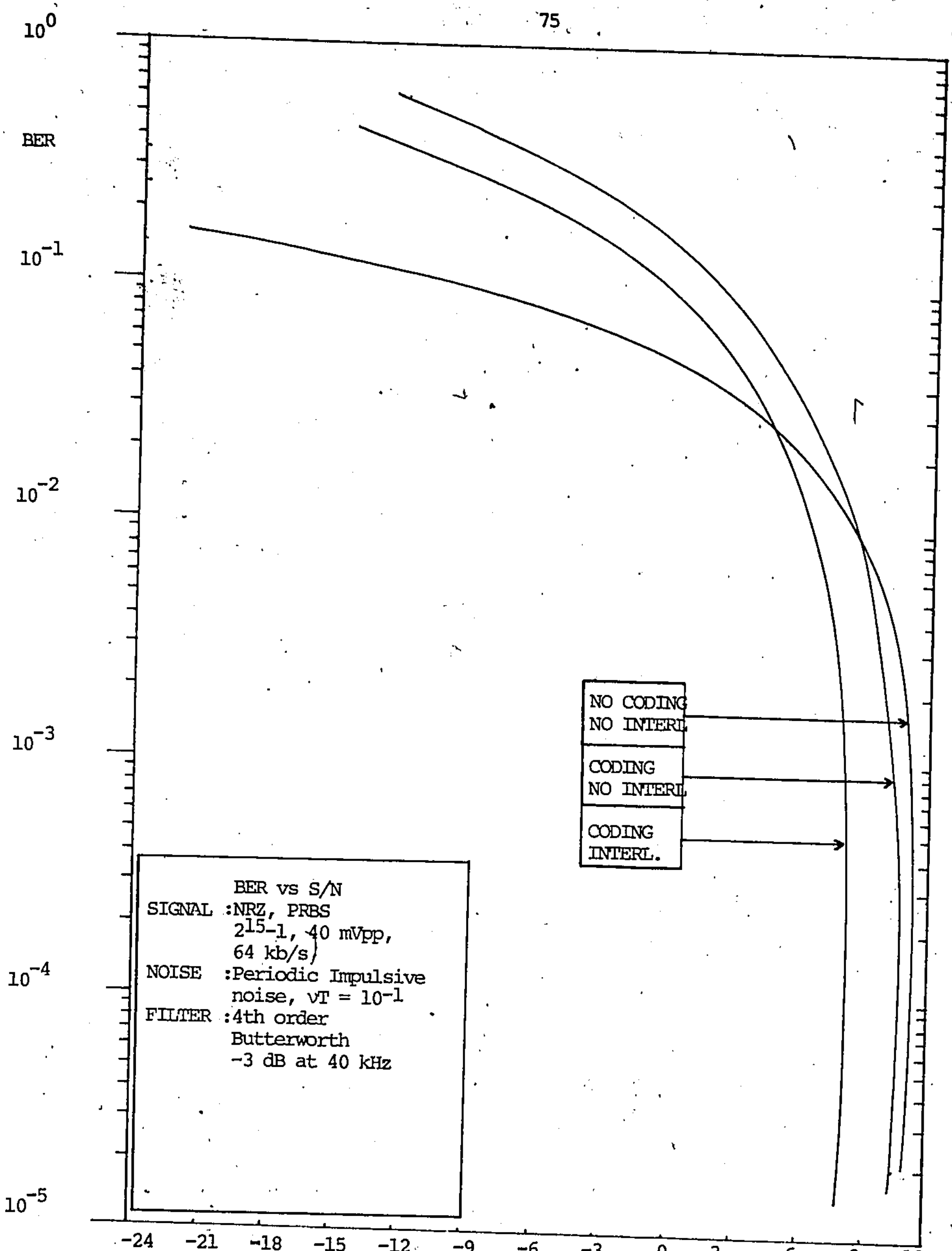


Fig. 5.3.6 - Measured BER Performance in an Impulsive Noise ($\nu T=10^{-1}$) Environment.

almost 3 dB in the area of BER=10^{-4}, as compared to the uncoded and uninterleaved case. Actually this improvement can be explained by the consideration that, for S/N > 4 dB the occurring due to impulsive noise error-bursts have average length less than the system interleaving depth of 32 symbols. On the other hand, for S/N < 4 dB the average length of the occurring error-bursts starts to become comparable to the interleaving depth of 32 symbols. As a consequence the threshold decoder at the receiving end, is unable to correct so many random errors, which are the result of the error-bursts spreading by the interleaving process. Therefore, in this area of S/N, the coded system BER performance shows a degradation with respect to the uncoded case, while the coded and interleaved BER performance shows a small improvement with respect to the coded BER performance, and an increased degradation, as the S/N decreases, with respect to the uncoded BER performance.

Finally, Figure 5.3.7 shows the system BER performance evaluation, obtained in an impulsive noise environment of impulse repetition rate $\nu T = 10^{-2}$ (1 impulse per 100 information symbols). In this case for S/N > 12 dB the occurring error-bursts have average length much smaller than the interleaving depth of 32 symbols. As a result in this area of S/N the

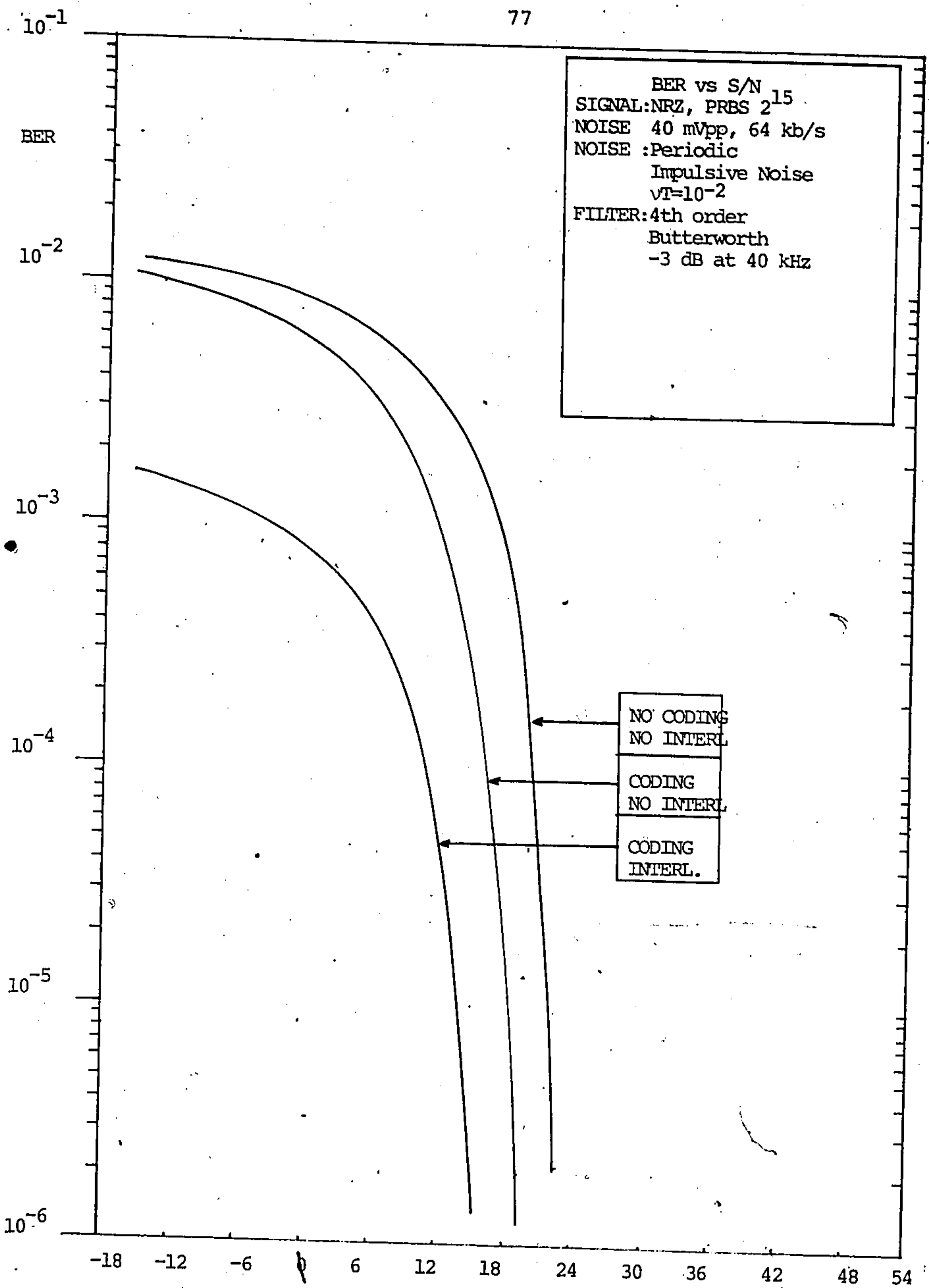


Fig. 5.3.7 - Measures BER Performance in an Impulsive Noise ($\nu T=10^{-2}$) Environment

interleaved and coded system BER performance, as compared to the uncoded case, shows an improvement of the order of 2 dB, which is almost equal to the improvement shown by the coded but uninterleaved case, as compared to the uncoded case. On the contrary, for S/N < 12 dB the error-burst length is likely to be longer than the case of S/N > 12 dB. As a consequence, the BER performance improvement due to interleaved codec is much greater than for the uninterleaved codec case.

c) Realization of a Non-Gaussian Noise Source in the Laboratory

Figure 5.3.8 shows the experimental set-up used to simulate a more realistic non-Gaussian noise source in the laboratory for the purpose of BER performance evaluation of the interleaved 3/4 rate convolutional codec. This non-Gaussian noise generation method can be easily compared to the out-of-band intermodulation "spikey" products generation, in a real satellite or microwave communication channel when signals interfered by Gaussian noise are processed through transponder nonlinearities [38].

White Gaussian noise is passed through a bandlimiting LPF, which has a cut-off frequency of 200 kHz. The output of the LPF is still Gaussian since the LPF is a linear device [21]. The bandlimited noise is then passed through a

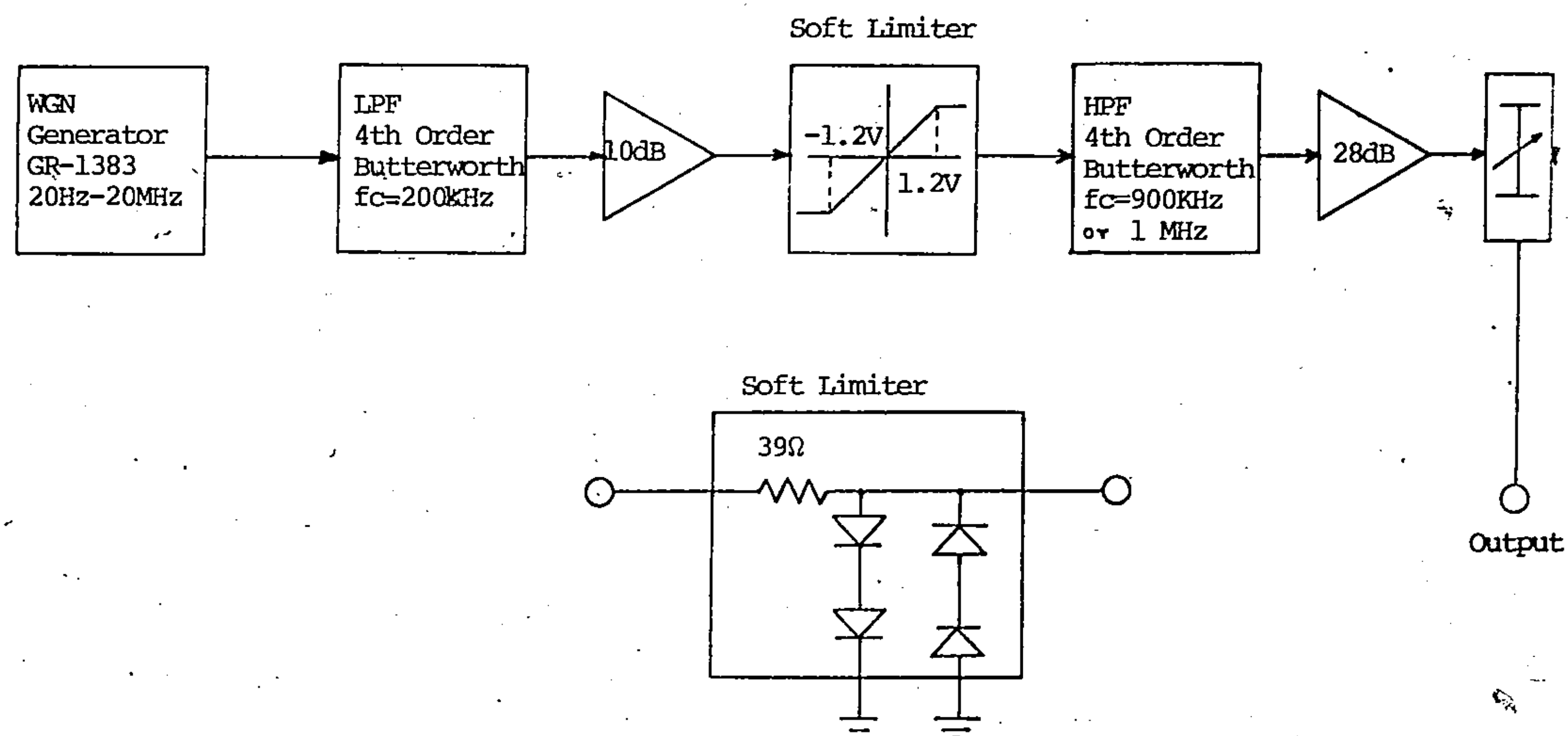


Fig. 5.3.8 - Set-up Used to Simulate Non-Gaussian Noise Source in the Laboratory

soft limiter nonlinearity. The waveform at the output of the soft limiter is shown in Figure 5.3.9.

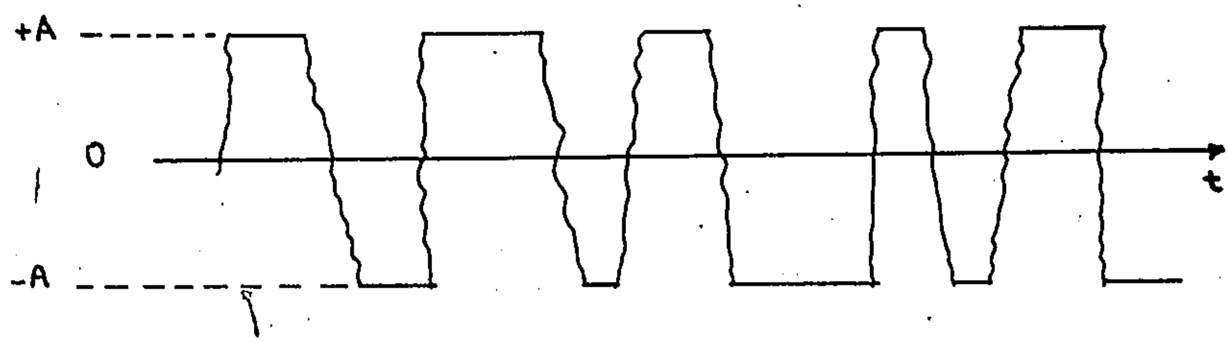


Fig. 5.3.9 - Output of the Soft Limiter

This limited waveform is then passed through a HPF with a variable cut-off frequency [51]. The HPF acts as a differentiator and therefore generates overlapping "impulses" [21], as shown in Figure 5.3.10. In order to get higher peak values for these impulses, a 28 dB amplifier is cascaded to the HPF.

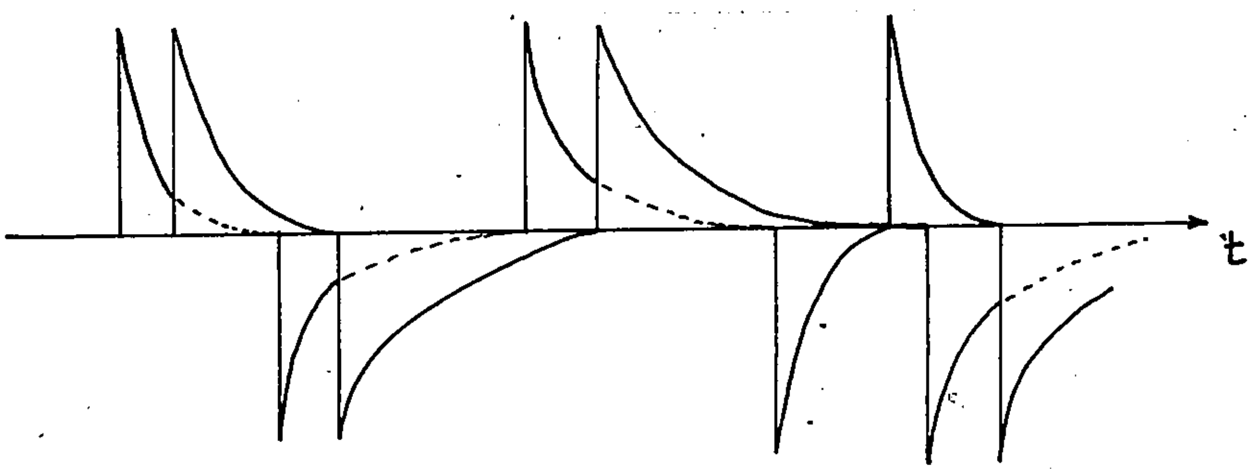


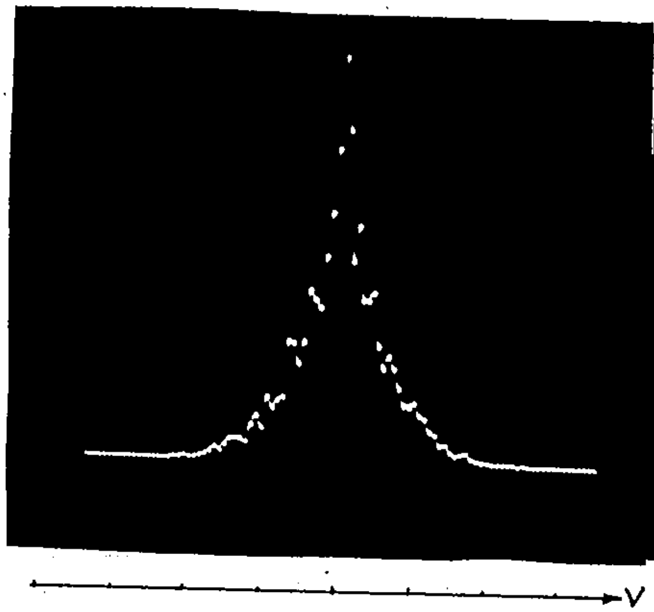
Fig. 5.3.10 - Output of the HPF

As is observed from Figure 5.3.10 the amplitude of the resulting impulses is much higher than the limiting level of the soft limiter shown in Figure 5.3.9. On the other hand, the duration t of the pulses shown in Figure 5.3.10 is not a constant. As a result the time available for each pulse to "die" completely is different, which modifies the pdf of these impulses.

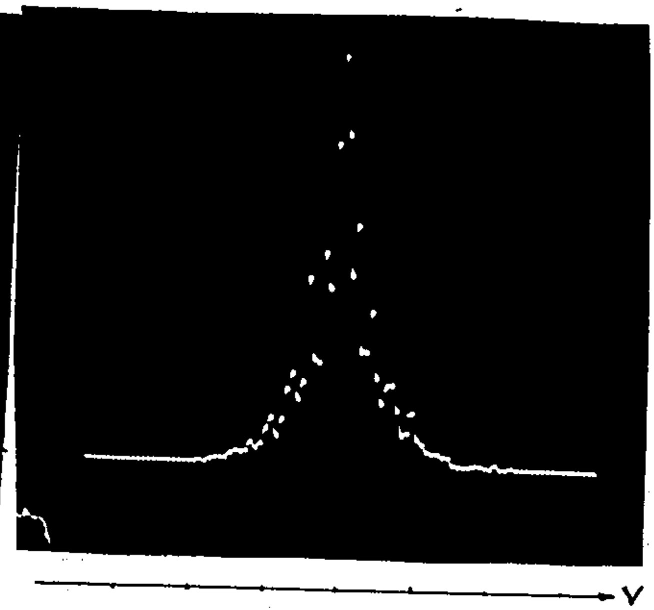
From the above discussion, it follows that the pdf of the overlapping impulses at the output of the non-Gaussian source, depends on amplitude and frequency range of the output of the Gaussian noise source, the cut-off frequencies of the LPF and HPF and also the amplitude characteristics of these two filters [38]. Experimentally it was verified that, by varying the cut-off frequency of the HPF, various pdfs were obtained (see Figure 5.3.11). As the cut-off frequency (=1/RC) of the HPF increases, the time constant t(=RC) decreases. This reduces the rms value of the output non-Gaussian noise. Because of the different pdfs and therefore different rms and peak values of the noise, different peak factors* experimentally were obtained at the output of the non-Gaussian noise source, for various HPF cut-off frequencies (see Table II). For the analytical derivation of these

---

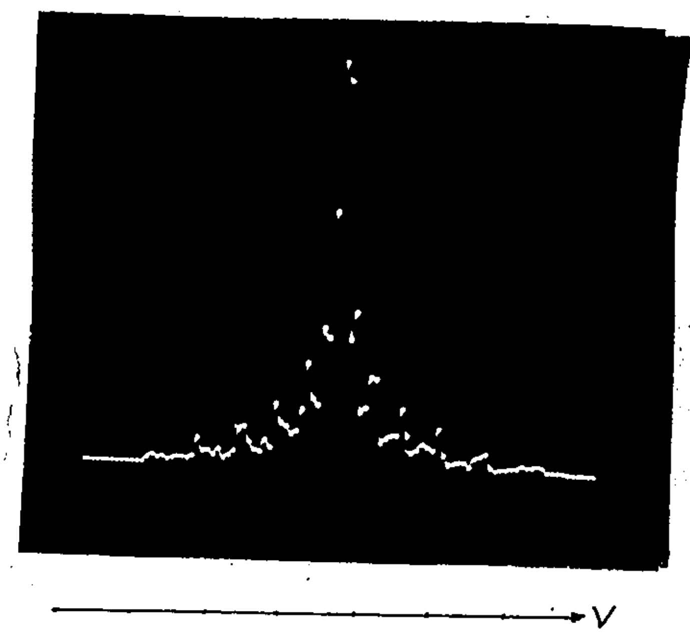
* The peak factor of a practical non-Gaussian "spikey" noise source has to be at least 7 dB higher than the "typical" peak factor (13 dB) of a practical Gaussian noise source [30] [46].



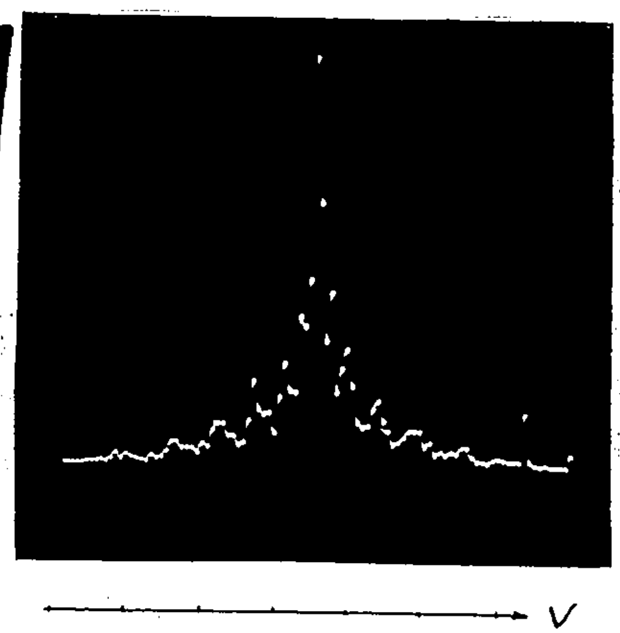
a) 700 KHz, 0.5 V/div



b). 900 KHz, 0.5 V/div



c) 1 MHz, 0.5 V/div



d) 1.2 MHz, 0.5 V/div

Fig. 5.3.11 - Experimentally Obtained pdfs for Different HPF Cut-off Frequencies of the Non-Gaussian Noise Source

pdfs, a complicated mathematical approach is required [28] [34] [48], which is out of the goals of this thesis.

Table II

HPF Cut-off Frequency	Non-Gaussian Noise Source Peak Factor
700 KHz	19.2 dB
900 KHz	25.1 dB
1 MHz	30.88 dB
1.2 MHz	35.63 dB

For more accurate estimation of the peak factor values, given in Table II, the configuration shown in Figure 5.3.12 was used to measure the noise instantaneous amplitude values at the output of the non-Gaussian noise source [30]. In this configuration any time that $V_A \neq V_B$ a positive pulse (0 to +5V) is produced by the flip-flop, counted by the following counter. Therefore, by adjusting the 10 kohm potentiometer and the attenuator, a zero count can be achieved. This means that $V_A = V_B$ and therefore, the voltage comparator output is at the 0 V level. Then by measuring V_B, V_A can be estimated within an accuracy of ±5%, due to hardware limitations. Finally, for the purpose of BER performance evaluation given in the next subsection, the 19.2 dB value of the peak factor was not used, since it is not high enough to discriminate the non-Gaussian noise source

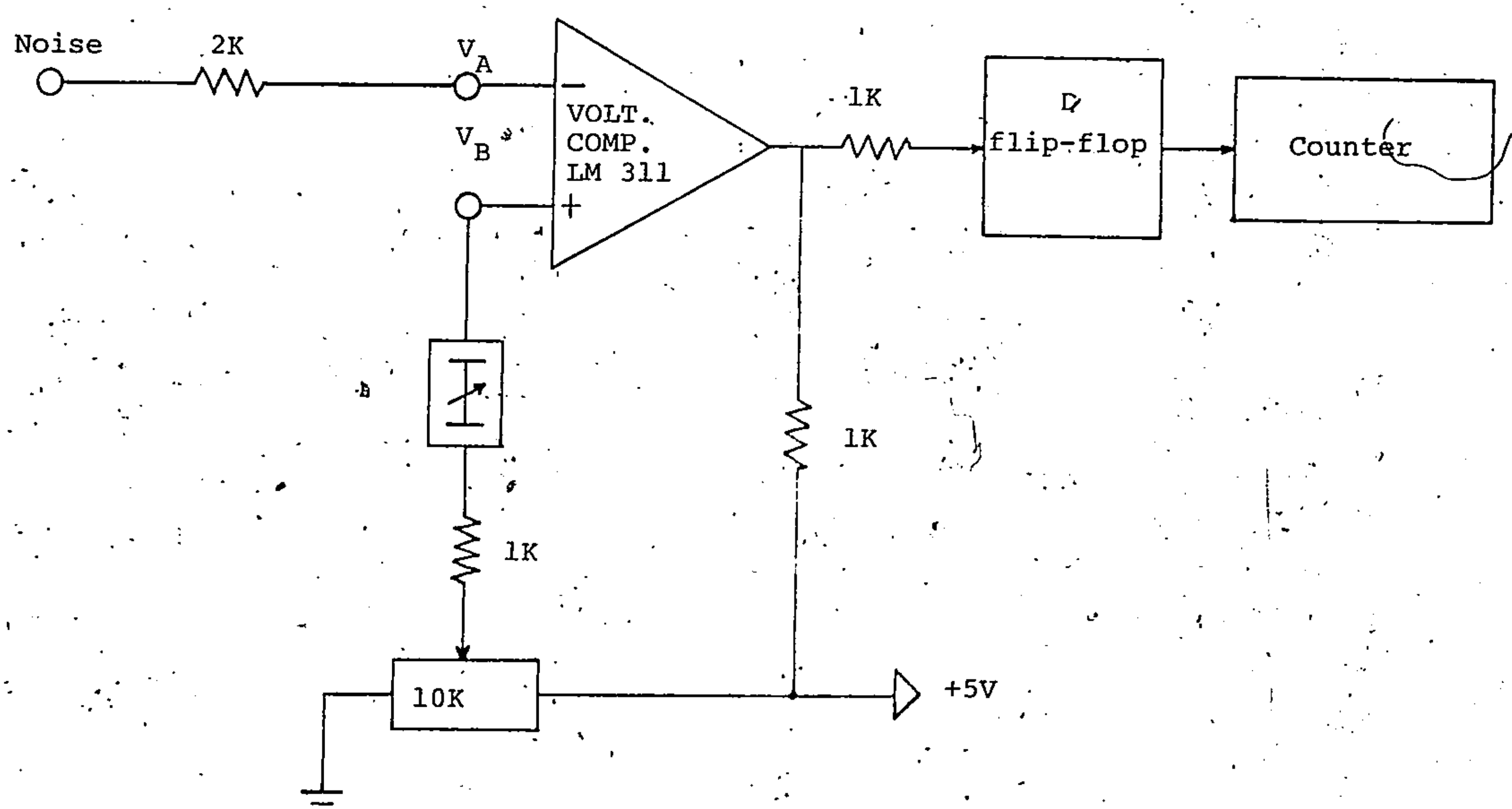


Fig. 5.3.12 - Noise Peak Value Measurements Set-up

from a Gaussian noise source. Similarly the 35.6 dB peak factor value was not used too, since it is close to the maximum operating peak factor (40 dB) of the true rms voltmeter (HP 3400), used for the evaluation measurements throughout this thesis (saturation problems).

d) Performance Evaluation in a Non-Gaussian "Spikey" Noise Environment

Figure 5.3.13 shows the experimental set-up used for the BER performance evaluation of the interleaved 3/4 rate convolutional codec in the non-Gaussian noise environment described in the previous sub-section 5.3c. Because of the operating frequency band of the non-Gaussian noise source, the interleaved codec was incorporated into a Binary Phase Shift Keying (BPSK) modulation scheme, with the carrier frequency set at 600 kHz [30].

To get reliable C/N measurements the receiving LPF cut-off frequency was adjusted to 1.5 MHz. Then from the measured values of C/N, the values of E_b/N_o were calculated using the procedure indicated below [38].

In general

$$C = E_b \times BR$$

$$N = N_o \times BW$$

or

$$\frac{E_b}{N_o} = \frac{C}{N} \times \frac{BW}{BR}$$

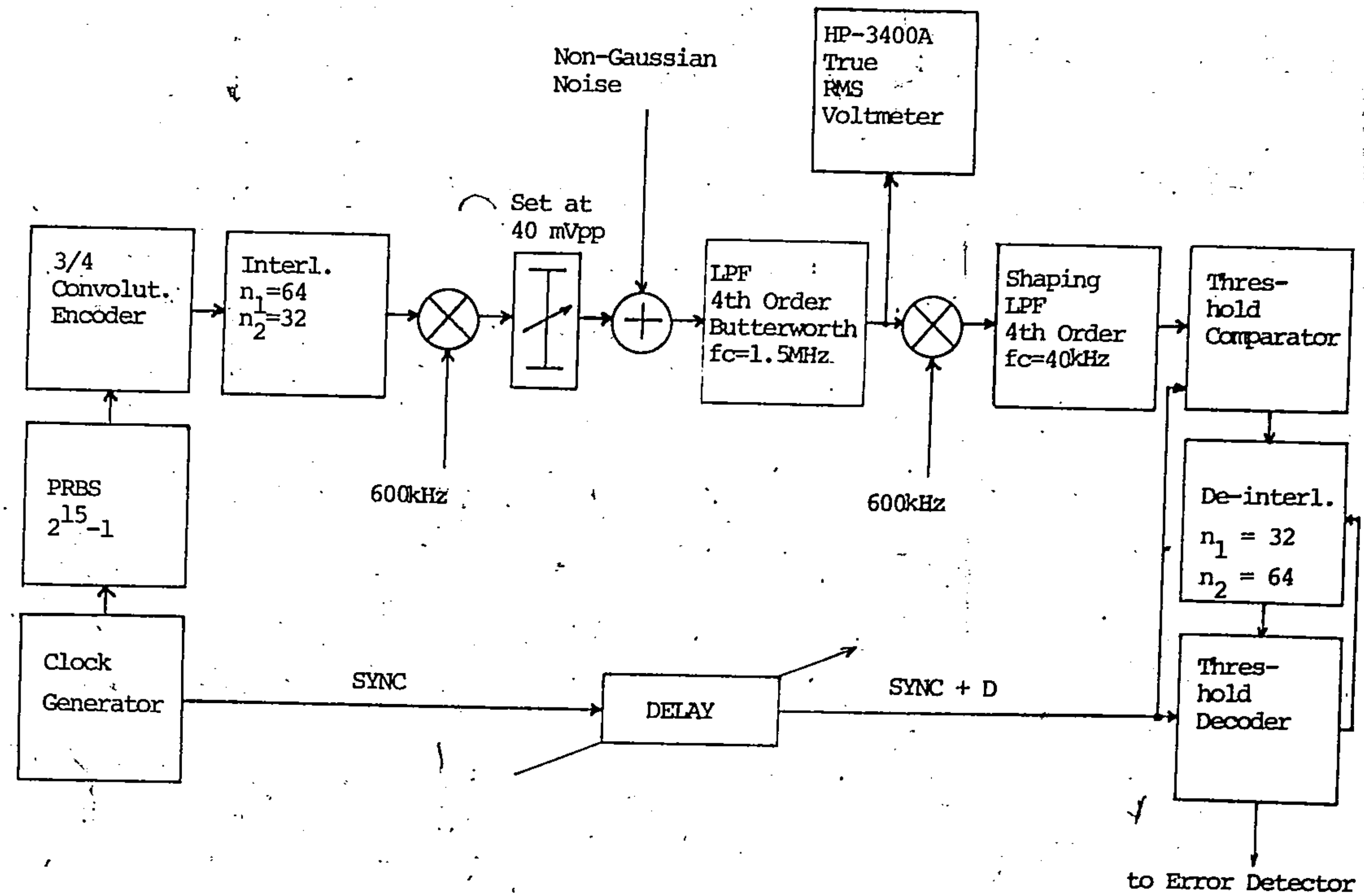


Fig. 5.3.13 - Set-up for the BER Performance in a Non-Gaussian Noise Environment

$$\text{or} \quad \frac{E_b}{N_o} = \frac{C}{N} + \frac{BW}{BR} \text{ (dB)} \qquad (12)$$

where C = carrier power

E_b = energy per bit

BR = symbol rate = 64 Kb/s for the given system

BW = receiving bandwidth = 1.5 MHz for the given system

N = mean noise power measured in the 1.5 MHz bandwidth

N_o = noise power in one Hz bandwidth

Therefore, for the given system equation (12) becomes:

$$\frac{E_b}{N_o} = \frac{C}{N} + 10 \log \frac{1.5 \times 10^6}{64 \times 10^3} \text{ (dB)}$$

$$\text{or} \quad \frac{E_b}{N_o} = \frac{C}{N} + 13.69 \text{ (dB)} \qquad (13)$$

Figure 5.3.14 shows the BER performance evaluation of the interleaved 3/4 rate convolutional codec in the non-Gaussian noise environment. As can be seen, for both values of the peak factor, the coded and uninterleaved BER performances (curves B_1 and C_1) are significantly degraded with respect to the uncoded BPSK BER performance in a Gaussian noise environment (curve A). This can be explained by the codec's inability to correct error-bursts of length longer than 2 errors per burst. The difference in degradation between curves B_1 and C_1 can be explained by the different

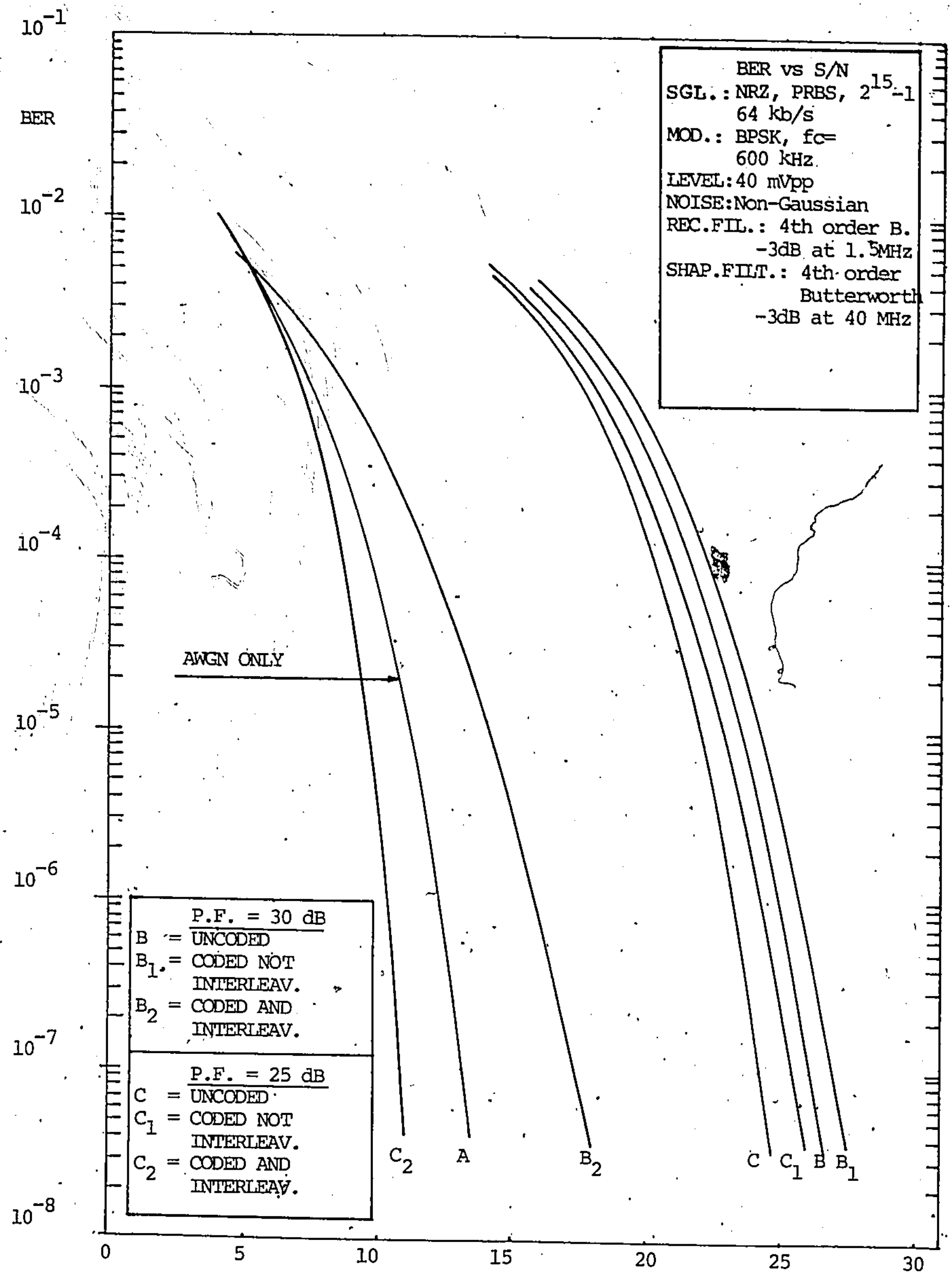


Fig. 5.3.14 - Measured BER Performance in a Non-Gaussian "Spikey" Noise Environment

error-burst lengths, caused to the BPSK signal by the different peak factors of non-Gaussian noise [48]. For the same reason, different improvements, due to interleaving, of the system BER performance were obtained, for different peak factors of the noise source. Therefore for non-Gaussian noise of peak factor of 30 dB, interleaving was not able to make the codec to correct so many random errors and consequently to "shift" the system BER performance to the left of curve A. However, the interleaved system BER performance (curve B_2) shows a significant improvement with respect to coded and uninterleaved system BER performance (curve B).

On the other hand, for non-Gaussian noise of peak factor of 25 dB, the interleaved and coded system BER performance (curve C_2) shows an improvement of 1.25 dB with respect to curve A and a total improvement of 12 dB with respect to curve C_1 (coded but uninterleaved BER performance) in the area of $P(e) = 10^{-5}$. In this case, the average length of the error-bursts, which due to interleaving appear like random errors to the threshold decoder, is shorter than in the case of peak factor of 30 dB. However, no actual measurements to verify the average error-burst length were attempted in this performance evaluation set-up also, since for these measurements, excessive computer work is required, mostly due to the complexity of the non-Gaussian "spikey" noise, generated as described in sub-section 5.3.c.

CHAPTER VI

FEATURES AND LIMITATIONS OF INTERLEAVED CODES. APPLICATIONS IN REAL COMMUNICATION CHANNELS

The data communication system designer, who must weigh the advantages of error control against its costs, will form a decision based on the nature and quality of the channel and of the other terminal equipment already present. Yet with the dramatic improvement in error control techniques and greater reliance on satellite communication systems (including also terrestial microwave links) for broadband data and voice transmission, decisions in favour of error control are becoming even more frequent. Satellite channels have been generally considered to be adequately represented by independent random error models (memoryless channel) [38]. In such channels the most effective forward error correction techniques can reduce the received signal-to-noise ratio, required for a given desired bit error rate, by 5 to 6 dB or more, compared to a system without error control [7]. This translates directly into an equivalent reduction in required satellite effective radiated power, with consequently reduced satellite weight and potentially remarkable reductions in satellite booster costs. For a satellite system with many ground stations even greater cost savings may be possible, by reducing the receiving antenna area by a factor of 4 and compensating by savings of 6 dB made possible by use of error control.

The cost of error control is two-fold: the equipment which, as just noted, may be more than compensated by reduction in other terminal equipment costs; and the redundancy required by the error control code [5] [13]. This redundancy need not, however, to reduce throughout if additional bandwidth is available in the channel. Satellite channels, in particular, are often not as much bandwidth limited as they are power limited. Still, even small bandwidth increases can produce considerable power savings. For example, an error control technique which employs a rate 1/2 code (100% redundancy) will require double the bandwidth of an uncoded system; on the other hand if a rate 3/4 code is used, the redundancy is 33% and the bandwidth expansion only 4/3. Yet the coding gain for a rate 3/4 code is only about 1 dB less than that of a rate 1/2 code of the same complexity.

However, in complex satellite systems involving multichannel communication and multiple access requirements, problems like adjacent channel interference, co-channel interference, intermodulation distortion (TWT non-linearities), loss of synchronisation, satellite tracking errors etc., may introduce error-bursts. In addition, in the satellite ground segment [38] [47] (such as microwave links, HF and tropospheric propagation links) error-bursts may be introduced. Usually in these channels, which are subject of fading and multipath phenomena, the error-bursts tend to corrupt long sequences of information symbols [38] [44]. Without precautions these error-bursts will make virtually any random-error correcting technique ineffective. One of the most direct preventive measure is to interleave (or scramble) the information symbols after encoding but prior to transmission and to perform the inverse operation at the receiver prior to decoding. These operations can be realized by devices called interleavers. An interleaver can be associated with any coding technique as an external device. Another approach is to incorporate the interleaving into the encoding and decoding procedure (internal interleaving).

Interleavers are characterized by their encoding delay D, which is the maximum delay encountered by any symbol before it is inserted into the output sequence. Actually an interleaver is optimum if it achieves the minimum possible encoding delay. It is this encoding delay which usually makes interleaving applications in packet transmission and retransmission systems in terrestial microwave communication channels problematic [38] [48]. This is so, because in these systems the information symbol propagation delay is of the order of the encoding delay usually introduced by inteleavers [29] [47]. Therefore extra circuitry (data buffers) is required to anticipate the extra propagation delay (usually in the case of data transmission), which highly increases the overall system cost [47].

On the other hand data and voice transmission via satellite will experience a propagation delay, which is usually greater than 0.25 sec [38]. Therefore, the encoding delay due to interleaving (usually optimized in the area of microseconds), will not significantly influence the overall symbol propagation. From the above discussion it is obvious that interleaved codes are widely applicable for error-burst and random error correction, in data or voice transmission via satellites [38] & [47].

In order to provide the literature with more information about the BER performance of interleaved codes in complex interference environments as appear in satellite channels [38], an (32,64)/(64,32) interleaving system was designed and built to accompany a 3/4 convolutional codec [7].

Specifically, the BER performance of the interleaved codec was first evaluated in an unmodulated carrier SI plus AWGN environment. The source of this type of interference in a satellite channel, could be a cross polarized signal maned from the same satellite, a sidelobe interference from a ground station transmitting to a nearby satellite, or a signal from a ground microwave link operating in the same band as the receive earth terminal. [38].

However, SI is not the only source of BER performance degradation of the received signals, in a satellite communication channel. Satellite transponder or earth terminal power amplifier amplitude nonlinearities can also generate input signal performance degradations [38]. As a consequence, the BER performance of the interleaved codec was evaluated in the two impulsive noise environments described in Section 5.3 of this thesis.

Finally, in all the measurements schemes included in Chapter V of this thesis, it was evident that an error event data acquisition measurement system was required for the exact error-burst length estimation. However, because of the complexity of the interfacing of the communication measurement instruments with a minicomputer system, the implementation of this measurement system was left as future work in another project.

CONCLUSION

The experimental performance of a rate 3/4 interleaved convolutional codec has been reported with respect to various transmission environments encountered in many practical communication systems. The reported results confirmed that the particular interleaving system, chosen to complement the rate 3/4 convolutional codec, eliminates indeed the channel memory effects on the received digital symbols. The interleaved codec performance was effective in complex environments such as sinusoidal interference (SI), AWGN plus SI in a simplified impulsive noise environment and in an out-of-band intermodulation "spikey" products environment. More specific in the SI plus AWGN environment, the system BER performance was improved due to interleaving by 5.7 dB in the area of $P(e) = 10^{-4}$, as long as S/I was kept below 3 dB. In the case of a simplified model of impulsive noise, consisting of periodic narrow impulses, the system BER performance was improved due to interleaving by almost 3 dB in the area of $P(e) = 10^{-4}$ for impulse repetition rate $\nu T = 10^{-1}$ impulses per symbol and by almost 4 dB for $\nu T = 10^{-2}$ in the area of $P(e) = 10^{-4}$. Finally, in the non-Gaussian "spikey" noise (intermodulation product) environment, the system BER performance was improved due to interleaving by 10 dB in the area of $P(e) = 10^{-5}$ for noise of peak factor of 30 dB. On the other hand, for noise of peak factor of 25 dB the system

BER performance was improved due to interleaving by almost 13 dB in the area of $P(e) = 10^{-5}$.

REFERENCES

[1] A.J. Viterbi & J.K. Omura. "Principles of Data Communication and Coding", McGraw Hill, New York 1979.

[2] R. Esposito. "A New Method of Calculating Probabilities of Errors Due to Impulsive Noise", IEEE Trans. on Com. Techn., Vol. COM-17, No. 3, June 1969.

[3] P.A. Bello & R. Esposito. "Error Probabilities Due to Impulsive Noise in Linear and Hard-limited DPSK System", IEEE Trans. on Com. Techn., Vol. COM-19, No. 1, February 1971.

[4] H.O. Burton & D.B. Sullivan. "Errors and Error Control", Proceedings of the IEEE, Vol. 60, No. 11, November 1972.

[5] W.W. Peterson & E.J. Weldon. "Error Correcting Codes", MIT Press, Cambridge, Mass., 1972.

[6] R. Lucky, J. Saltz & E.J. Weldon, Jr. "Principles of Data Communication", McGraw Hill, N.Y., 1968.

[7] A. Brind Amour & K. Feher. "Design and Evaluation of a Convolutional CODEC in an AWGN, Sinusoidal Interference and Intersymbol Interference Environment", IEEE Trans. of Communications, COM-28, No. 3, March 1980, pp. 391-394.

[8] K. Feher. "Digital Modulation Techniques in an Interference Environment", Don White Consultants, Inc. book, Gainesville, Virginia, 1977.

[9] J. Conan, D. Haccoun & H.H. Hoc. "Error Control Techniques for Data Transmission Over Satellites", Ecole Polytechnique, Université de Montréal, November 1975.

[10] C.E. Shannon. "A Mathematical Theory of Communication", The Bell System Technical Journal, Vol. 27, July and October 1948.

[11] E.N. Gilbert. "Capacity of a Burst-Noise Channel", The Bell System Technical Journal, September 1960, pp. 1253-1265.

[12] ____________. "Clustering in Discrete Stochastic Process with Applications to Channels Having Memory", Tech. Rep. Grant AFOSR 60-1390, Dept. of Electrical Engineering, Le High University, August 1960.

[13] S. Lin. "An Introduction to Error Correcting Codes", Prentice-Hall Inc., N.J., 1970.

[14] A.J. Viterbi. "Convolutional Systems", IEEE Trans. of Com. Techn., Vol. 19, October 1971, pp. 751-772.

[15] P. Elias. "Coding for Noisy Channels", ERE Convention Record, Part 4, 1955, pp. 37-47.

[16] J.M. Wozencraft. "Sequential Decoding for Reliable Communications", 1957 National IRE Convention Record 5, Part 2, pp. 11-25; also MIT Research Laboratory of Electronic Technical Report 325, Cambridge, Mass. 1957.

[17] J.L. Massey. "Treshold Decoding", MIT Press, 1963.

[18] J.P. Robinson & A.J. Bernstein. "A Class of Binary Recurrent Codes with Limited Error Propagation", IEEE Trans. of Information Theory, Vol. IT-13, April 1967, pp. 106-113.

[19] B. Golberg (ed.). "Communication Channels Characterization and Behavior", IEEE Press, N.Y., 1967.

[20] W.R. Bennett & J.R. Davey. "Data Transmission", McGraw Hill Book Company, N.Y., 1965.

[21] A. Papoulis. "Probability, Random Variables and Stochastic Process," McGraw Hill Book Company, N.Y. 1965.

[22] J.D. Parson & A.V. Sheikh. "The Characterization of Impulsive Noise and Consideration for a Noise-Measuring Receiver", The Radio and Electronic Engineer, Vol. 49, No. 9, September 1979, pp. 467-476.

[23] E.N. Skomal. "Man-Made Radio Noise", Van Nostrand-Renhold, N.Y., 1978.

[24] P. Mertz. "Model of Impulsive Noise for Data Transmission", IRE Int. Conv. Record, Part 5, March 1960, pp. 247-260.

[25] W.J. Richter & I.S. Talivaldis. "Signal Design and Error Rate of an Impulsive Noise Channel", IEEE Trans. of Com., COM-19, December 1971, pp. 446-458.

[26] S. Tsai & P.S. Schmied. "Interleaving and Error-burst Distribution", IEEE Trans. on Com., Vol. COM-20, June 1972, pp. 291-296.

[27] W. Feller. "An Introduction to Probability Theory and its Applications", Vol. 1, John Wiley and Sons, Chapter VI, N.Y., 1957.

[28] P. Mertz. "Impulsive Noise andError Performance in Data Transmission", USAF Memorandum, April 1965.

[29] J.L. Ramsey. "Realization of Optimum Interleavers", IEEE Trans. of Inf. Theory, Vol. IT-16, No. 3, May 1970, pp. 338-345.

[30] K. Feher. "Digital Communications: Microwave Applications", Prentice-Hall Inc. Book, Englewood Cliffs, N.J., December 1980.

[31] E. Hopner. "An Experimental Modulation-Demodulation Scheme for High-Speed Data Transmission", IBM Journal of Research and Development, Vol. 3, January 1959, pp. 74-84.

[32] G.A. Cambell. "Collected Papers", American Telephone and Telegraph Co., 1937; also, "Probability Curves, Showing Poisson Exponential Summation", Bell System TEchnical Journal, Vol. 2, 1923, pp. 95-115.

[33] ____________. "Unifilar Sources and the Source Approximation Problem", Ericson Tech., Vol. 28, 1972, pp. 175-190.

[34] E.O. Elliot. "Estimates of Error Rates for Codes on Burst-Noise Channels", Bell System Technical Journal, Vol. 42, September 1963, pp. 1977-1997.

[35] R.H. McCullough. "The Binary Regenerative Channel", Bell System Technical Journal, Vol. 47, October 1968, pp. 1713-1735.

[36] L.N. Kanal. "Models for Channels with Memory and Their Applications to Error Control", Proceedings of IEEE, Vol. 66, No. 7, July 1978.

[37] J.R. Juroshek & G.E. Wasson. "Measurements of Digital Systems in Gaussian Additive Noise and Interference", U.S. Department of Commerce, September 1976.

[38] J.J. Spilker, Jr. "Digital Communications by Satellite", Prentice-Hall, Inc., Englewood Cliffs, N.J., 1977.

[39] G.C. Clark, Jr. & R.C. Davis. "Two Recent Applications of Error-Correction Coding to Communication System Design", IEEE Trans. on Commun. Tech., Vol. COM-19, No. 5, September 1971.

[40] J. Conan, D. Haccoun & H.H. Hoang. "Performance of ARQ and Hybrid ARQ/FEC Error Control Schemes for High Speed Data Transmission on Satellite Channels",

Proceedings 1976, Canadian Conference on Communications and Power, Montreal, November 1976.

[41] H. Taub & D.L. Schilling. "Principles of Communication Systems", McGraw-Hill Book Company, N.Y., 1974.

[42] J.M. Wozencraft & I.M. Jackobs. "Principles of Communication Engineering", John Wiley and Sons, Inc., N.Y., 1965.

[43] H.T. Hsu, T. Kasami & R.T. Chien. "Error-Correcting Codes for a Compound Channel", IEEE Trans. on Information Theory, Vol. IT-14, Vol. 1, January 1968.

[44] J.R. Tucker. "Error Behavior of VHF Channels", IEEE Int. Conf. Commun. 1972, pp. 15-25 - 15-30.

[45] ____________. "1645 A DATA ERROR ANALYZER" Operating Information, Hewlett Packard, November 1976.

[46] ____________. "TYPE 1383 RANDOM-NOISE GENERATOR", Instruction Manual, General Radio, 1969.

[47] J. Martin. "Communications Satellite Systems", Prentice-Hall, Inc., Englewood Cliffs, N.J., 1978.

[48] S. Oshita, & K. Feher. "Burst Error Phenomena: Analysis and Design of Digital Systems Operating in an Impulse Noise Environment." Report, University of Ottawa, Canada, March 1981.

[49] ______________________. "3760 A Data Generator", Operating Information, Hewelett Packard.

[50] ______________________. "3761 An Error Detector", Operating Information, Hewelett Packard.

[51] ______________________. "3202 Filter", Operating Informations, Krohn-Hite.

[52] Andre Brind Amour. "Probability of Error Estimation of a Baseband Signal in the Presence of SI Plus AWGN", Report, University of Ottawa, Ottawa, Canada, 1979.

[53] K. Yamamoto & S. Oshita & K. Feher. "BER Performance Calculations of a Baseband Signal, Exposed in a Simplified Impulsive Noise Environment", Report, University of Ottawa, Ottawa, Canada, March 1981.