International Journal of Computer and Information Sciences, Vol. 13, No. 4, 1984

Further Remarks for the Matrix Type-B Codes

D. C. Voukalis

Received February 1983; revised June 1984

Different philosophies lie behind the detecting and correcting error patterns in a real communication channel. The sceptic points in choosing an efficient code, specifically the matrix type-B code, were pointed out in Refs. 1 and 2.

Some more points are shown here. As a result the matrix type-B code is found to be a "best choice". Some more theoretical aspects for this code are also given. These are useful for the realization and testing of an encoding-decoding algorithm with IC's used in a unique way for its implementation.

KEY WORDS: Matrix type-B code; mitroid; optimum bribing; asymptotic behavior; and implementation procedure method.

1. INTRODUCTION

As it is known from our previous research,[1,2] matrix codes have a great advantage among other codes used for error control in a real communication channel. This is why they are described as a "best choice" codes.

In general the idea behind a matrix code can be tested in two ways. By building up equipment, in otherwords, by hardware and/or by producing a software package. The problem considered here is to find a decoding algorithm for a special case of the matrix type-B code and implement it by means of hardware (with logical IC's) in a real communication channel. In addition, the encoding-decoding (codec) apparatus is of main importance to both sender and receiver.

The investigation reported in the present paper is a continuation on the fundamental matrix type-B codes.[3]

The error patterns for correction to be considered are discussed briefly in Section 2. Some theoretical aspects necessary for the discussion of the

0091-7036/84/0800-0309$03.50/0

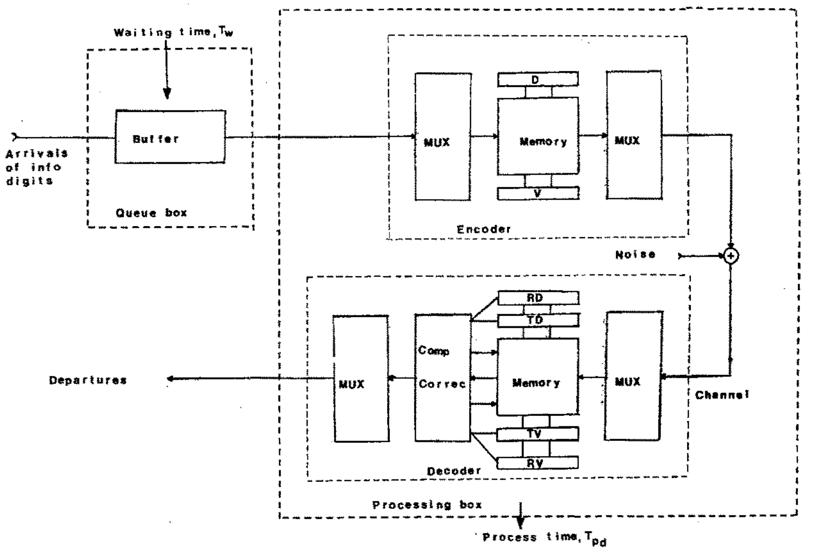

Fig. 1. The channel E-D queueing finite system.

decoding algorithm of Section 4 are given in Section 3. Then the implementation structure with IC's of the encoder-decoder is described in Section 5. In Sections 6 and 7, results are stated concerning the theoretical and the technical contents presented on all the previous sections. The work leading to these results was done independently by the author and some of this research (as i.e. the decoder, Theorem 4, comparisons between other burst correcting codes, et. al.) is new. Therefore, we can point out in view of these that the paper contributes much to the existing knowledge of the technical and theoretical matters of the matrix codes.

The best known methods of structuring elementary matrix and concatenated matrix codes, such as matrix type-B, matrix type-C, powerful

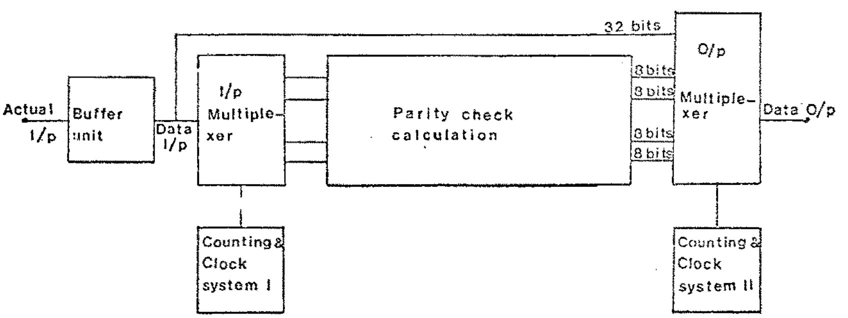

Fig. 2. Encoder.

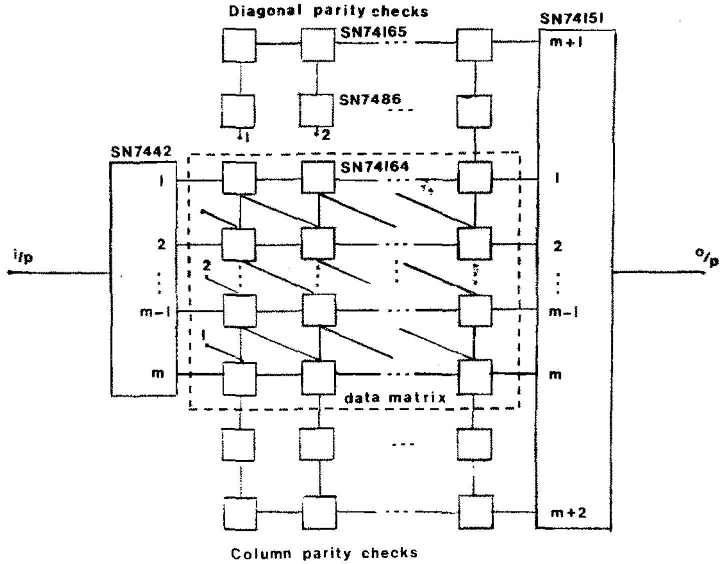

Fig. 3. Sequential circuit for matrix type-B encoder system.

product, and quadratic residue etc., are presented in Refs. 1–4. We investigated the independent behavior of the E-DS in Ref. 4, (Fig. 1.), and saw how the matrix codes link together with other communication systems under the beady eye of a communication designer engineer.[5]

In Ref. 4, we discussed the details of implementing an encoder. A block diagram of the encoder is shown in Fig. 2. and the sequential circuit encoder is shown in Fig. 3. Further on in this paper we complete the study of matrix type-B code by giving the experimental results of the matrix type-B decoder system as shown in Fig. 4.

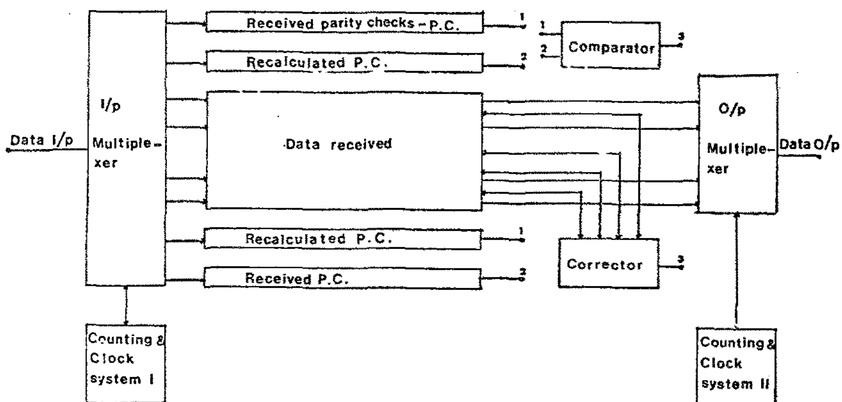

Fig. 4. Decoder.

2. THE ERROR PATTERNS

Until now in our research described in Refs. 1–3, we discussed a number of cases concerning different types of error patterns. They are:

Case 1. A single random error. For the signal S,

$$S \rightarrow \overbrace{1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ \underbrace{1}_{\sigma_e}\ 0\ 0}^{n}$$

we have the probability $P(n, \sigma_e)$, where n—the signal length and σ_e—the number of the erroneous digits. In this case $\sigma_e = \sigma_r = 1$.

Case 2. Random errors t or more, where case of $t = 1$ is the previous trivial case of this category. For the signal S,

$$S \rightarrow 1\ 0\ \underbrace{1}_{}\ 0\ \underbrace{1}_{}\ 1\ 1\ 1\ 1\ 0\ 0$$

we have probability $P(n, \sigma_t)$ with $\sigma_t > 1$.

Case 3. Multiple (dependent) errors.

A. For the signal S,

$$S \rightarrow 1\ 0\ 1\ \underbrace{0\ 1\ 1\ 1\ 1}_{\sigma_b}\ 1\ 0\ 0$$

we have a burst of errors, with probability $P(n, \sigma_b)$.

B. For the signal S,

$$S \rightarrow 1\ 0\ \underbrace{\underbrace{1\ 0}_{\sigma_{b1}}\ 1\ \underbrace{1\ 1\ 1}_{\sigma_{b2}}}_{\sigma_c}\ 1\ 0\ 0$$

we have two bursts of errors σ_{b1} and σ_{b2} which form a cluster of errors.

Case 3 is the most interesting case as we can see in the following section.

Let the code have the relative probability of error correction ability $e(n, \sigma_e)$.

Then we shall have for the channel C,

$$C \rightarrow P(n, \sigma_e)$$

and for the code $\mathbb{C}$,

$$\mathbb{C} \rightarrow e(n, \sigma_e).$$

These two probabilites then have to match in order to correct the errors. So:

$$P(n, \sigma_e) < e(n, \sigma_e) \; \forall \sigma_e, n \tag{1}$$

The probability $P(n, \sigma_e)$ for a given channel is obtained almost always experimentally.

3. SOME THEORETICAL ASPECTS OF MATRIX TYPE-B CODES

Good codes for error control in a digital communication system are those characterised by their ability, at the receiving apparatus, to be good enough to define the exact syndrome (i.e. the ratio of the correct check digits vector to the recalculating check digits vector). Such a code is the matrix type-B code with column and diagonal parity checking. The following definition gives an algebraic characterization of the matrix type-B code, which represents the matrix behavior of the code.

Definition. The matrix type-B code has the following form: $[g(x)$ – The matrix codebook information digits $= a_{ij}$ (usually a matrix $n = n_s \times m)]$.

$$g_d(x) = f[\![g(x)]\!] = \sum_{\substack{i=k \\ j=\lambda}}^{m} a_{ij} \; (\text{mod-}2) \doteq C_d$$

$$\text{is } a_{ij} = 0 \begin{cases} i > m & j < n & k \in [\![l, m]\!] \\ i > m & j < n & \lambda \in [\![l, n]\!] \end{cases}$$

where: $d = 1 \div [\![m + n - 1]\!]$, the diagonal parity checks calculated this with either left or right direction and:

$$g_\tau(x) = f[\![g(x)]\!] = \sum a_{ij} \Bigg| \begin{matrix} i = l, m \\ j = \lambda \end{matrix} \; (\text{mod-}2) \doteq C_c$$

where: $\tau = 1 \div n$, the column parity checks. Then the universal matrix type-B code, $U(x)$, becomes;

$$U(x) = \begin{array}{|c|} g_d(x) \\ \hline g(x) \\ \hline g_\tau(x) \end{array}$$

Where: n_s-codeword of $g(x)$, m-number of codewords in $g(x)$.

Example. An $[\![(m=4)(n_s=8)]\!]$ matrix type-B code is:

$$
U(x)=\left|\begin{array}{l}
0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 1 \\
\hline
0\ 1\ 1\ 1\quad 1\ 0\ 1\quad 0 \\
0\ 0\ 1\ 0\quad 0\ 0\ 1\quad 0 \\
\\
0\ 1\ 1\ 0\quad 1\ 0\ 0\quad 1 \\
0\ 0\ 0\ 0\quad 1\ 1\ 0\quad 1 \\
\hline
0\ 0\ 1\ 1\quad 1\ 1\ 0\quad 0
\end{array}\right|
\begin{array}{l}
[\![g_d(x)]\!] \\
\\
\\
g(x) \\
\\
\\
\\
[\![g_\tau(x)]\!]
\end{array}
$$

The definition and the example shows that the matrix type-B code breaks the continuous signal sequence of the information digits into sections or blocks, n_s, making a data codebook matrix and then operates on the unique matrix.

Theorem 1. A matrix type-B code with $n_s > m$ has a minimum bound burst correcting ability b, $n_s \geqslant b$ and/or a random number of digits in the guard space equal to or greater than m.

Proof. For burst errors (using orthogonal or double checking of the errors) we have a number of check space vectors, $\mathbf{F}$, as:

$$\mathbf{F} = C_\tau + C_d = 2n_s + (m-1)$$

then this holds:

$$2n_s + (m-1) \geqslant 2\sigma_e = 2b$$

which is guaranteed by,

$$2n_s \geqslant 2\mathrm{b}$$

so that finally:

$$n_s \geqslant b$$

which is the desired conclusion, with b as the burst length given by the real channel. We correct random errors m or greater in the guard space, because we have n_s actual correcting pair vector equations and some more diagonal correcting vector equations that can correct random errors accordingly. ∎

In light of this theorem we can prove the following corollaries:

Corollary 1. The burst error correction of the matrix type-B code is greater than the Reiger bound.

Proof. From the comparison of two Theorems: Theorem 1 in which $b=n_s$, and from the Reiger bound (given from Theorem 4.15[5]) in which $b \leqslant (n-k/2)$ (where $n=$codeword length, $k=$information digits in a codeword) follows that, at least, we have the same results or the matrix type-B code has greater correcting ability. Because, for all n_s pairs we may have the same pairs in the cyclic codes but there are some with diagonal correcting vector equations from which we can correct more errors. ▮

Corollary 2. The burst error correction of the matrix type-B code is greater than the Weyner–Ash bound.

Proof. Cyclic and convolutional codes have similar error correcting capabilities and same fundamental limitations. Convolutional codes are correcting less information digits than the cyclic codes. In Corollary 1, we prove that matrix type-B code have greater ability than cyclic codes and that the matrix type-B code has the greater ability. ▮

Later in this section one can see examples showing another way of proving Corollaries 1 and 2.

Corollary 3. Matrix type-B codes are good for synchronization recovery, because they are easily self-synchronizable codes.

Proof. If we manage the first row or vector of the matrix to be of zero value then this can be used as a self-synchronizable criterion. This can be easily achieved if the first level of the quantizer is always zero. ▮

Theorem 2. Every matrix type-B code, as well as any other matrix code, is equivalent to a systematic code.

Proof. There is a simple way (as the simple parity check codes) to find the parity check vectors for the code that follows the Hamming sense, when the data matrix a_{ij} is given. Then each data digit can be checked by two linearly independent parity check elements. ▮

Theorem 3. Every matrix type-B code has the capability to produce series of concatenated codes.

Proof. From the definition given when $g(x)=\mathbb{C}_E$, where $\mathbb{C}_E$ is an inner code of any type (cyclic, convolutional, matrix or any other) then one obtains:

$$U(x)=\begin{array}{|c|}g_d(x)\\\hline \mathbb{C}_E\\\hline g_\tau(x)\end{array}$$ ▮

This theorem implies the following corollaries:

Corollary 4. Every matrix type-B code has the ability to produce good communication protocols.

Proof. Forming the channel signal, S, one obtains:

$$S \to U(x) \doteq g_d(x)+\mathbb{C}_E+g_\tau(x)$$

$$\doteq \mathbb{C}_{EZ}+\mathbb{C}_E$$

if:

$$\mathbb{C}_{EZ}=g_d(x)+g_\tau(x)$$

is the external matrix type-B code. ▮

We can see that each block of the protocol is defined and each one can be synchronized (see Corollary 3).

Corollary 5. Simplification of the data matrix $[\![a_{ij}]\!]$ of a cyclic matrix type-B code gives an echelon canonical form from which we can use only the parity checks and one obtains a powerful cyclic productive matrix type-B code.

Proof. The combination of the two codes give a significant mathematical structure that can always be reduced to echelon canonical form and has a error probability lower than the probability of individual code types. This is the cyclic productive matrix type-B code that is more powerful than each one individually.[2] ▮

Corollary 6. Every matrix type-B code has the ability to produce good computer communication network protocols.

Proof. Forming the channel signal protocol, Pr, one obtains:

$$\mathrm{Pr} \to U(x)+\cdots \doteq g_d(x)+\mathbb{C}_E+g_\tau(x)+\cdots$$

$$\doteq \mathbb{C}_{EZ}+\mathbb{C}_E+\cdots$$

Where the dots show the computer communication pattern criteria. ▮

Also as in Corollary 4 all the blocks are defined and exactly synchronized.

For our error correction needs we shall lay down the following theorem.

Theorem 4. Noise effects giving any type of errors (any type of error pattern), in the bound $\sigma_e \leqslant n_s$, for the matrix type-B codes, can be corrected by the code.

Proof. From Theorem 1, any burst or cluster errors with $\sigma_b \leqslant n_s$, also any random error $\sigma_r = 1$ can be corrected. In general any random multiple error $\sigma_t \leqslant n_s$ can be corrected and this is obvious because of the geometrical properties of the code, which are capable of correcting any type of error pattern less or equal to n_s. ▮

Theorem 4 shows that $e(n, \sigma_e) = 1$ for $\sigma_e \leqslant n_s$ and σ_e perhaps is any type of error pattern such as random, burst, or cluster. Further on we can consider any type of errors with length n_s as a burst case as in Section 2. As we can see this result seems strange for other codes than elementary matrix (type-B and type-C) codes and as such it is open to criticism.

Now we are ready to make a new comparison between the burst error correcting codes.

In general the parity checks give a sort of picture of the burst and then the burst error correction is a relation of the parity checks and in extent a relation of the code structure (usually a geometrical property with good or no burst correcting ability).

For the bounds of the three main categories[1,2] of the burst error correcting codes we shall limit ourselves to some exellent illustrative examples:

a. For the cyclic codes given from the Reiger bound: $b \leqslant (n - k/2)$ (see Theorem 4.15 in Ref. 5):
For $n = 15$, $k = 9$ in a $(n, k) = (15, 9)$ cyclic code we have:

$$b \leqslant \frac{n-k}{2} \leqslant \frac{15-9}{2} \leqslant 3 \text{ burst error digits}$$

b. For the convolutional codes given from the Wyner–Ash bound: $b \leqslant n_0$ (See Theorem 4.18 in Ref. 5):
For $L = 9$, $u = 6$, $k = 10$ is $n = (L + k)\,u = 114$ for a $(nn_0, mk_0) = (3.9, 3.3)$ convolutional code we have:

$$b \leqslant n_0 \leqslant 3 \text{ burst error digits}$$

Where L is length of a convolutional code (cc), n_0 is length of the internal (basic parity check) matrix, h is codeword for the cc, m_c is number of codewords of the internal matrix in a cc, k_0 is number of codewords of the internal matrix in a cc, and u is parity checks added for a cc.

c. For the matrix codes given from Theorem 1: $b \leqslant n_s$:
For $k=9, n=18, n_s=9$ for a $(n, k)=(18, 9)$ matrix type-B code we have:

$$b \leqslant n_s \leqslant 9 \text{ burst error digits}$$

Then from these examples and from Refs. 1 and 2 the matrix type-B code is advantageous over the other burst correcting codes. These examples and the examples given in Ref. 1–3 are useful in connection with the following theorem.

Theorem 5. When comparing the three main categories of burst error correcting codes (cyclic, convolutional, and matrix) only one parameter can be varied at a time. Then we say that they have asymptotic behavior.

Proof. We investigate the asymptotic behavior in the examples precited and in the examples given in Ref. 1 and 2. We can also see from these examples that only one parameter at a time can be examined.

Another way to say this is that since each code lies on different philosophy and uses a number of different parameters for the geometrical consideration of the exact error correction but when compared they have one common (in size) parameter at a time. ▮

4. ON DECODING MATRIX TYPE-B CODES

The decoding algorithm for a matrix type-B code is to link together equipment, like the matrix memory of data and the parity check memory, which usually placed in a chip or adjacent chips. However, the various ways of going about networking are numerous and complex. An ultimate solution, at present, anyway, which aims to solve such a problem is discussed in this and the next sections.

A. The Procedure of the Decoding Algorithm

1. Assume that each time $g(x)=a_{ij}=$ data matrix in columns or rows. This is choosen to be distinct and dependent by means of a mixture of standards to produce the parity check digits in specially designed hardware. Then a deterministic Marcov process is taking place.

2. Attempt at standardization correction procedure:
   a. One of the main points to remember is that a symbol at any position in the data matrix is subject to error.
   b. Only $\sigma_e \leqslant n_s$ continuous and of any form errors are accepted for the procedure.
3. Consider an array of many numbers as a single object that we denote by a single symbol. Relationships between variables can then be expressed in a clear and concise way.

The combinatorial logic network is designed to compare the received and recalculated parity checks, to find the erroneous parity checks and from them to find the form and the error positions of the data symbols. Then the matrix logical circuit indicates and corrects each time the values of the error symbols. We note that in our example the errors must not occur one under the other.

The reason is that each error vector is defined in a fixed way. So each pair of error parity check digits exactly defines the position of an error in the mitroid (see the next numerical example).

It is easy for the reader to see the difficult problem of how to choose the set of parity check calculations and permutations. In practice we made this choice by an interesting way to be shown next. This is elucidated more by the following numerical example.

B. Numerical Example

If we sent the word "veto" and instead the word "visa" is received, it is obvious that the meaning of the message is quite different. Let us examine the meanings of these labels before proceeding in the encoder. With the presentation of the remaining steps in detail by the error correction procedure using the matrix type-B $[\![(n_s = 8) \cdot (m = 4)]\!]$ code, we can see how easily we can take the correct word sent by the encoder.

We have

	0	0	0	0	0	1	0	0	0	0	1	$\longrightarrow$ PRd
/	0	0	1	1	1	1	1	0	1	0	0	$\longrightarrow$ Prsd
v	0	0	0	1	0	1	0	1				v
e	0	1	1	0	1	0	1	0				i
t	0	1	1	1	1	0	1	0				s
o	0	0	0	0	1	0	0	0				a
	0	0	0	0	1	1	0	1				$\longrightarrow$ Prsc
	0	1	1	1	1	0	1	0				$\longrightarrow$ PRc

Using EOR gates for comparison one obtains the erroneous parity checks for each pair Prsc, PRc, PRd, and Prsd. Here we need criteria to help us calculate the syndrome or decide which errors (or error) are in the message.

So for the Prsc and PRc pair using EOR one obtains:

$$\begin{array}{ll}
\text{Prsc} & \rightarrow 0\ 0\ 0\ 0\ 1\ 1\ 0\ 1 \\
\text{PRc} & \rightarrow 0\ 1\ 1\ 1\ 1\ 0\ 1\ 0 \\
\text{Iv, EOR} & \rightarrow 0\ 1\ 1\ 1\ 0\ 1\ 1\ 1
\end{array}$$

and for the PRd and Prsd one obtains:

$$\begin{array}{ll}
\text{PRd} & \rightarrow 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1 \\
\text{Prsd} & \rightarrow 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 0 \\
\text{Id, EOR} & \rightarrow 0\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1
\end{array}$$

Using Theorem 1 as one criterion and AND gates logic constructed in a mitroid structure as a second criterion one obtains the exact form and the positions of the errors from the noise effected digits:

$$\begin{array}{ll}
\text{Iv} & \rightarrow 1\ 1\ 1\ 1\ 1\ 1 \\
\text{Id} & \rightarrow 1\ 1\ 1\ 1\ 1\ 1 \\
\text{Rw, AND} & \rightarrow 1\ 1\ 1\ 1\ 1\ 1
\end{array}$$

error pattern detected, exact positions of errors are given with 1, all others are 0.

Then using each received digit, $a_{mn}^{\varnothing}$ (correct or not), and its check, Rw, as input to a EOR one always obtains in the output the correct digit, a_{mn}, as:

$$\begin{array}{ll}
a_{mn}^{\varnothing} & \rightarrow 0\ 1\ 0\ 1 \\
\text{Rw} & \rightarrow 0\ 0\ 1\ 1 \\
 & \quad\ \ \hookrightarrow \text{erroneous case} \\
a_{mn}, \text{EOR} & \rightarrow 0\ 1\ 1\ 0
\end{array}$$

The key of the error correction ability lies in the mitroid circuit. For the output of these gates gives the correct digit to be used, i.e. the digit the encoder sent.

Where PRd represents recalculated diagonal parity checks, Prsc is received column parity checks, Prsd is received diagonal parity checks, PRc is recalculated column parity checks, $I_{v,\mathrm{EOR}}$ is output of a EOR for comparison for a Prsc and PRc position, and $I_{d,\mathrm{EOR}}$ is output of a EOR for comparison for a Prsd and PRd position. $R_{w,\mathrm{AND}}$ represents output of an AND for a check position in the mitroid structure that gives a detected error and its position, $a^{\varnothing}_{mn,\mathrm{EOR}}$ is a correct or not information digit in the receiver apparatus, and $a_{mn,\mathrm{EOR}}$ is an information digit after correction.

It is in fact, understandable from this given example, taking into consideration the matrix type-B code, that we use the simple and straightforward Euclidean decoding algorithm described as calculation of syndrome, detection of error pattern, and correction of the detected error pattern. This decoding is simple to understand and it is also comparable in speed with other methods, so, it is the method we prefer in practice.

5. IMPLEMENTATION STRUCTURE OF THE DECODER

Several key theorems were presented in Section 3. With the help of these theorems and the application of the encoding-decoding system[3,6] given in Fig. 1., the reader should be able to gain further insight into the structure of the decoding. Using the simple and easily instrumented decoding algorithm given in the previous section we attempt to provide some proofs that lead to the simplest implementations.

Theorem 6. All memory devices are assumed to contain zeros initially.

Proof. Because the code information digits are always dependent on the state of the system and the process is a deterministic Markov process, then it is right at the start where the state condition to be $S_{\mathrm{con}} = 0$. ∎

It is also quite clear that the following theorem holds:

Theorem 7. If a number of errors exist in the process the intersection of $Iv \cap Id$ is never empty.

Proof. There exists an integer of N row data digits such that $\cap n_s \leqslant N$ is not empty. ∎

If this theorem does not hold (if there are some errors and the intersection of $Iv \cap Id$ is empty) it is meaningless to talk about decoding. But when the theorem holds, then as a consequence the behavior of the decoding circuitry is governed by the matrix code (Theorem 1) behavior. Furthermore an interesting result which we shall state here without proof, is that the behavior of the optimum bribing (packaging) procedure and the cost function of the code apparatus depends upon the bribing policy and the customers will.

According to this result, which we have just proved (theorem 7), it is desirable to have this algorithm constructed for a data flow in networks that achieve the maximum possible burst error correction. With this in mind we prove the following:

Theorem 8. Every matrix type-B code should always satisfy its parity check equations during the processing.

Proof. This is an immediate consequence of Theorems 1, 6, and 7. ▮

Looking upon circuitry, we have the *a priori* knowledge of the emitter process given by the circuit of Fig. 3. This circuit has one matrix level with two transformations.

1st Transformation:

- Data entrance and parity checks calculation 2nd Transformation:
- Data and parity checks transfer to the channel process

In searching for an optimal decision at any stage of the process, one needs only to look for a decision that will optimize the activities in the subsequent stages of the decoder given by the Fig. 4. It follows that a step-by-step process can be carried out in the process period. Further on for the implemendation using the logic and consequently the truth tables given in "Texas Instruments Semicoductor Components Data Book Two. Digital Integrated Circuits" July 1971, as we used for the emitter, we obtained the following detail circuit (Fig. 5) for the decoder apparatus.

For this circuit we have three matrix level actions with two matrix transformations as: (see Fig. 5)

1st matrix level

- Data and parity checks storage, recalculation of parity checks and comparison to see the erroneous parity checks and error warning.
- 1st transformation of data and transmitted parity checks (MUX) with the main clock.

2nd matrix level

- Positions and forms of data errors, ready for error correction and error warning. Error correction or retransmission request.
- Error correction with a small period of the main clock.

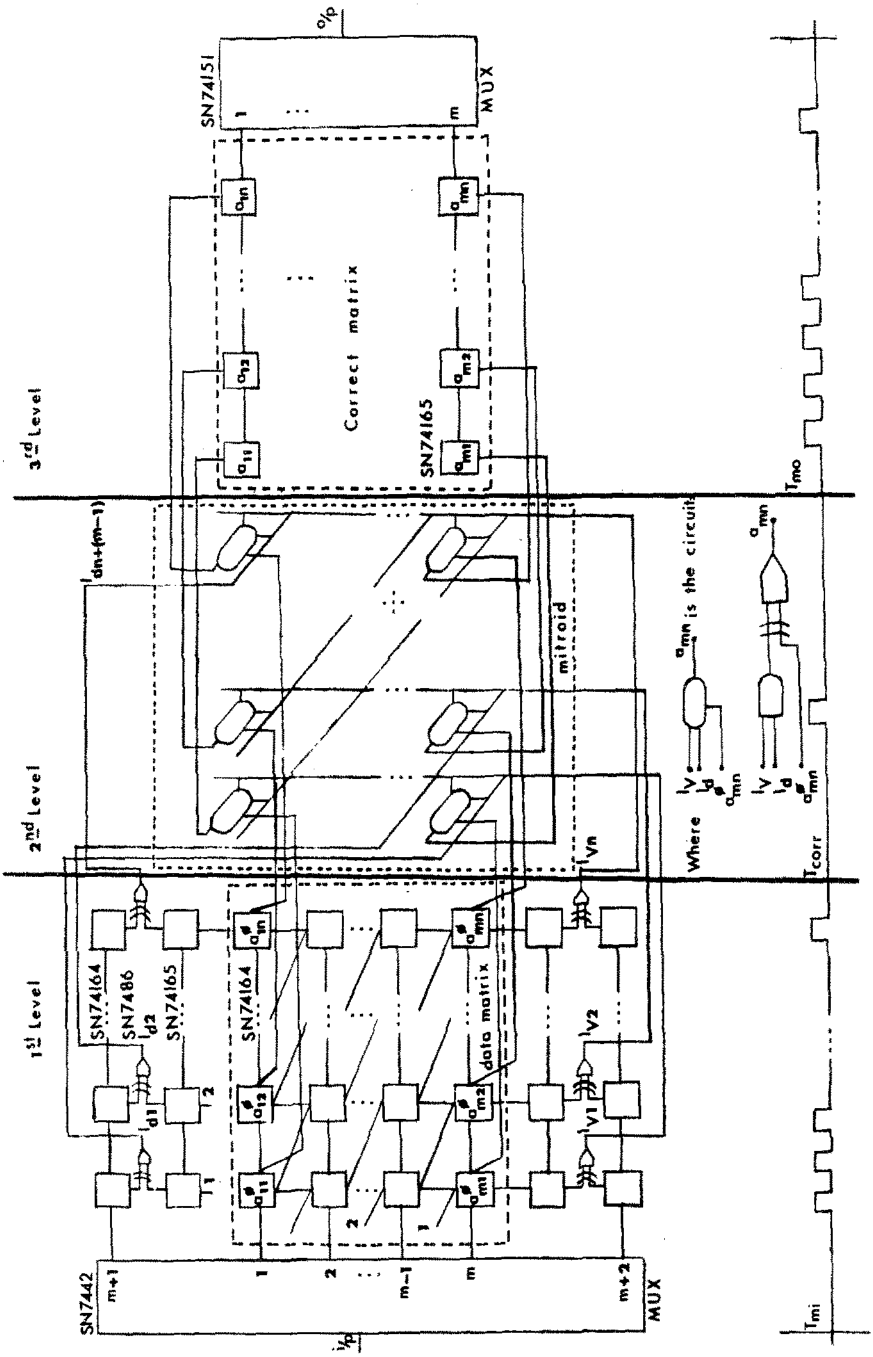

Fig. 5. Sequential circuit for matrix type-B code decoder system.

3rd matrix level

- Final correct data matrix.
- 2nd transformation of data (MUX) with the main clock.

Furthermore, we have the switching-clock system and the clock driver system logic for the decoder system designed experimentally, which are not shown here.

Instead of giving a proof of the validity of these rules of the actions and transformations, we remind the reader that such transformations and actions are just a systematic way of carrying out the substitution steps illustrated earlier for the burst error corrections.

In our experimental decoder apparatus for a $[\![(n_s=8)\cdot(m=4)]\!]$ code the regular working data clock rate was:

$$R_c = 0.557^{-1} \times 10^3 \text{ bits}/s$$

At this rate we had a good performance of error correction. This performance was maintained for data rate experimentally varied between the bounds:

$$(0.25 \times 0.5)^{-1} \times 10^3 < R_c < (0.35 \times 0.1)^{-1} \times 10^6 \text{ bits}/s$$

6. COMMENTS AND DISCUSSIONS

Theorem 4 has some rather surprising consequences in comparison with the other correcting code philosophies. It states that regardless of how a matrix type-B code is used or how noisy is the channel, the error correction probability is proportional to n_s and independent of type of error pattern. This fits very well to most real channels that are affected by bursts of noise and loss of synchronization. That is, $P(n, \sigma_e)$, is determined primarily by the number of parity check digits and the way of their calculation. These would act as giant data thorax for the network, which give the interesting geometrical property (locus) that the error positions have to be on one or more of the intermediate parallel data vector lines.

7. APPLICATIONS IN COMPUTER SYSTEMS

As computer controlled information, computer programs, and computer communication networks become an increasingly important part of the real world today, the protection and error control within the computers and in digital communications becomes increasingly vital to any information system.

A matrix type-B code can be used as a protector and error control cipher security method to any text editor. Major applications are within systems where an efficient code is needed due to the importance of high quality information transmission as are military applications and practical commercial networks. They can also be used for forward error correction (in an optimum way) in modems or network ports for character and block error protection in order to provide important time savings in the transmission of the communicating messages. Further on, the algorithms for correction used by the matrix codes can be implemented as part of a software package to achieve the previous results.

To our knowledge, however, there are no commercially available systems using the matrix type-B, or more complicated matrix type correction codes.

8. CONCLUSION

The most elementary description of matrix codes, the mathematical construction of matrix type-B code and an easy decoding algorithm of the mentioned code has been illustrated.

The matrix type-B code implemented with IC's for one simple representative case has been solved and tested with data. Further, it can be easily checked that if we now apply the results of Section 5 to any n_s to any matrix type-B (or type-C) code, then the extension to this implementation procedure method is obvious. Also this method is general and any matrix code is valid to provide a deterministic process of an encoding-decoding model. This is sufficient because the coding rule has its domain sequences of block length n_s and for such deterministic processes it can be obviously extended.

ACKNOWLEDGMENTS

It is with pleasure that the author acknowledges his debt to Dr. Chr. Koutsogeorgis and to Mr. D. Nisidis who made this work worthwhile.

REFERENCES

1. D. C. Voulalis, A comparison between burst error correcting codes, *Int. J. Electron.* **49**(4):319–325 (1980).
2. D. C. Voukalis, A comparison of known high-rate concatenated codes, *Int. J. Electron.* **52**(2):177–187 (1982).
3. D. C. Voukalis, A new series of concatenated codes subject to matrix type-B code, *IEEE Trans. Inform. Theory* **28**(3):522–530 (May 1982).

4. D. C. Voukalis, Error statistics and protocol construction using matrix burst correcting codes, *Int. J. Electron.* **48**(4):337–340 (1980).
5. W. W. Peterson and E. J. Weldon, Error correcting codes, The Massachusetts Institute of Technology Press, (1972).
6. N. E. Head, A high-speed data transmission system, *General Electric C° J.* **30**:1–22 (1963).

Printed in Belgium