THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 55, No. 2, February 1976
Printed in U.S.A.

A Simple Description of an Error-Correcting Code for High-Density Magnetic Tape

By N. J. A. SLOANE

(Manuscript received September 22, 1975)

Hong and Patel have described an efficient error-correcting code for magnetic tape, which has been successfully used on the IBM 6250 bits-per-inch nine-track magnetic tape units. This paper gives a simple description of the code, without using Galois fields.

I. INTRODUCTION

The latest IBM tape units use ½-in., nine-track tape with the very high density of 6250 bits per in. This is made possible, in part, by the use of an efficient error-correcting code, which can correct errors in one or two tracks.

The code was described by Patel and Hong[1] (see also Ref. 2), and is a straightforward extension of earlier IBM codes (see Refs. 3 and 4). The purpose of this paper is to give a simple description of the code and its many nice features, without using Galois fields.

II. ENCODING

A code-word consists of 72 bits arranged on the tape in a 9×8 rectangle, as shown in Fig. 1. The ninth track is simply an overall parity check on the other eight tracks, i.e., it is equal to the modulo 2 sum of the other eight tracks. The left-hand column of each code-word also consists of check bits. Thus 16 out of the 72 bits are checks and 56 are information bits. The rate or efficiency of the code is 56/72 = 0.778. Data are read on and off the tape by vertical columns (Fig. 2). The ith column consists of eight bits (denoted by B_i) together with an overall parity check bit. $B_7, B_6, \cdots, B_1$ are the information columns and are written on the tape in this order. Finally the check column B_0 is written on the tape. B_0 is chosen so that the code-word satisfies the vector equation

$$B_0 + TB_1 + T^2B_2 + \cdots + T^7B_7 = 0, \tag{1}$$

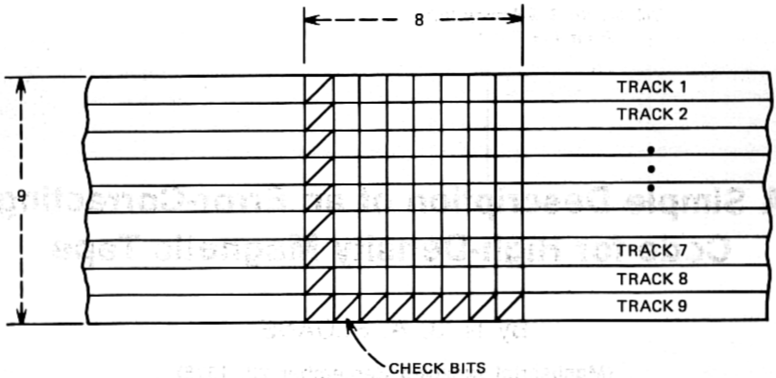

Fig. 1—A code-word is a rectangle of 9 × 8 bits.

where T is the matrix

$$T = \begin{bmatrix} & & & & & & & 1 \\ 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & 1 \\ & & & 1 & & & & 1 \\ & & & & 1 & & & 1 \\ & & & & & 1 & & \\ & & & & & & 1 & \end{bmatrix}.$$

Note the special shape of T: there are 1's only below the main diagonal and in the last column. In fact, T describes the action of the linear feedback shift register shown in Fig. 3. If the contents of the register are

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_7 \end{bmatrix},$$

then one time unit later it contains

$$\begin{bmatrix} a_7 \\ a_0 \\ a_1 \\ a_2 + a_7 \\ a_3 + a_7 \\ a_4 + a_7 \\ a_5 \\ a_6 \end{bmatrix} = T \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix}.$$

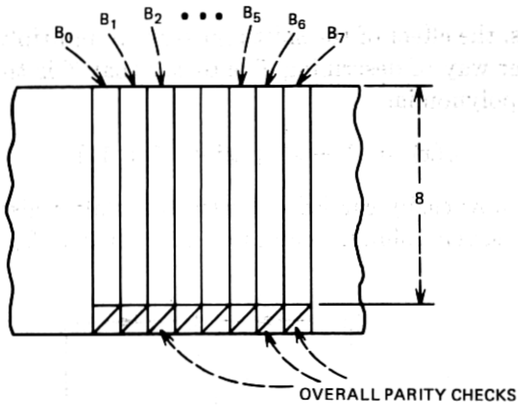

Fig. 2—A code-word divided into eight columns.

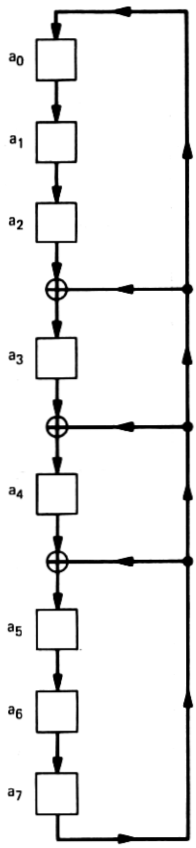

Fig. 3—A shift register which multiplies by T.

In other words, the effect of the shift register is to multiply its contents by T. [Another way of describing T is to say that T is the companion matrix of the polynomial

$$g(x) = x^8 + x^5 + x^4 + x^3 + 1.]$$

Encoding is now easily carried out with this shift register, as shown in Fig. 4. The seven column vectors of information, $B_7, B_6, \cdots, B_1,$

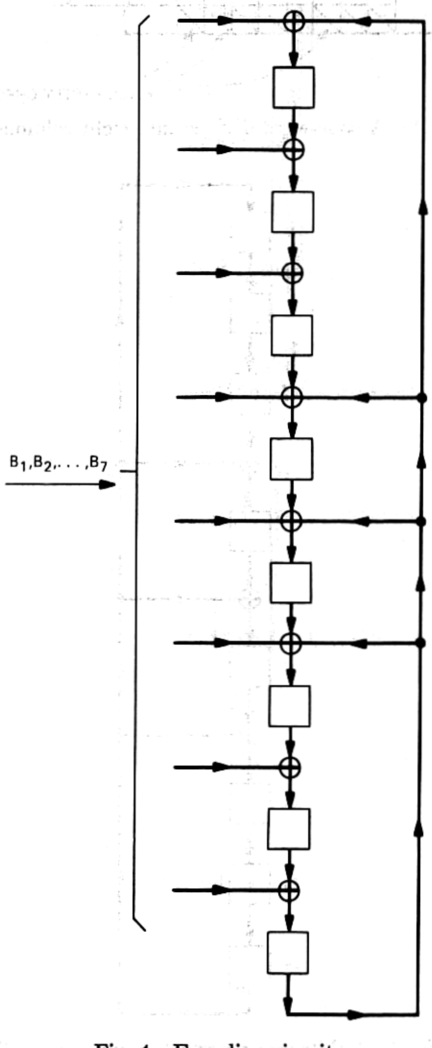

Fig. 4—Encoding circuit.

are fed into the register, which successively contains

$$B_7,$$
$$B_6 + TB_7,$$
$$B_5 + TB_6 + T^2B_7,$$
$$\cdots$$

and finally

$$TB_1 + T^2B_2 + \cdots + T^7B_7,$$

which from eq. (1) is B_0, the check column that we wanted. B_0 is then written directly on the tape (together with the overall parity check in track 9).

III. DECODING

When the code-word is read back from the tape, it may contain errors. Since the bit density in the horizontal direction is much greater than that in the vertical direction, the commonest type of error is a horizontal burst along a track. Often the erroneous tracks can be identified by a loss of signal in the tape-reading head, or by other electronic indications.

To describe the decoding process, some further notation is required. Let $Z_0, Z_1, \cdots, Z_8$ be defined as in Fig. 5. The Z_i's are the horizontal slices of the same code-word we had before. The last row Z_8 is the overall parity check row, defined by

$$Z_8 = Z_0 + \cdots + Z_7$$

or equivalently (all calculations are carried out modulo 2):

$$Z_0 + Z_1 + \cdots + Z_7 + Z_8 = 0. \tag{2}$$

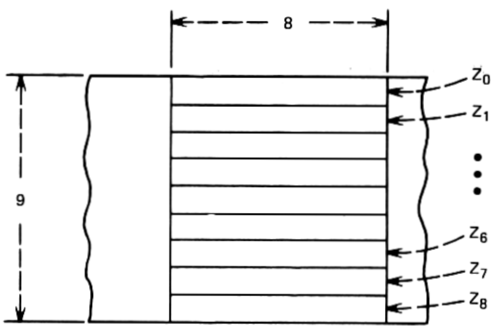

Fig. 5—A code-word divided into nine rows.

Of course the Z_i's and B_j's are related. If we write

$$Z_i = (Z_{i0}, Z_{i1}, \cdots, Z_{i7}), \qquad B_j = \begin{bmatrix} B_{0j} \\ B_{1j} \\ \vdots \\ B_{7j} \end{bmatrix},$$

then

$$Z_{ij} = B_{ij} \tag{3}$$

are both names for the bit in position (i, j), $0 \leqq i, j \leqq 7$, of the top eight rows of the code word.

The B_j's must satisfy eq. (1). How does this constrain the Z_i's? The answer is a nice surprise: they must satisfy essentially the same equation, namely

$$Z'_0 + TZ'_1 + T^2Z'_2 + \cdots + T^7Z'_7 = 0, \tag{4}$$

where the prime denotes transpose. This equation is derived in the appendix.

Now suppose that errors have occurred, and the distorted vectors

$$\hat{Z}_i = Z_i + e_i, \qquad i = 0, \cdots, 8$$

have been read off the tape, where e_i is the eight-bit error vector in the ith horizontal slice (or track). We wish to find the e_i's so that we can recover the original code-word using

$$Z_i = \hat{Z}_i + e_i, \qquad i = 0, \cdots, 8.$$

The decoder begins by computing the *syndromes*

$$S_1 = \hat{Z}'_0 + \hat{Z}'_1 + \cdots + \hat{Z}'_8$$

and

$$\begin{aligned} S_2 &= \hat{Z}'_0 + T\hat{Z}'_1 + \cdots + T^7\hat{Z}'_7 \\ &= \hat{B}_0 + T\hat{B}_1 + \cdots + T^7\hat{B}_7. \end{aligned}$$

From eqs. (2) and (4) we see that S_1 and S_2 are zero if there are no errors and, in general, give the "symptoms" of the errors. In fact,

$$\begin{aligned} S_1 &= \sum_{i=0}^{8} Z'_i + \sum_{i=0}^{8} e'_i \\ &= \sum_{i=0}^{8} e'_i \qquad \text{from (2),} \end{aligned} \tag{5}$$

and similarly

$$S_2 = \sum_{i=0}^{7} T^i e'_i. \tag{6}$$

S_1 is easily found: it is simply the sum of all the rows. S_2 is obtained by feeding $\hat{B}_7, \hat{B}_6, \cdots, \hat{B}_0$ into the shift register of Fig. 4 as they are read off the tape. After $\hat{B}_0$ has been fed in, by eq. (1) the register contains $\sum_{j=0}^{7} T^j\hat{B}_j = S_2$.

After the decoder has found S_1 and S_2, there are two ways of proceeding.

Mode I. To correct an error in one track.

Suppose the ith track is in error, and e_i is the error vector in the ith track. The decoder knows [from eqs. (5) and (6)]

$$S_1 = e_i'$$

and

$$S_2 = \begin{cases} T^i e_i' & \text{if } 0 \leqq i \leqq 7 \\ 0 & \text{if } i = 8, \end{cases}$$

since S_2 only involves tracks 0 through 7. Thus S_1 tells us e_i. To find i, we use S_2. If $S_2 = 0$, $i = 8$. Otherwise S_2 is successively multiplied by T^{-1} until e_i' is reached, and i is the number of multiplications required. A circuit which multiplies by T^{-1} is simply obtained by reversing the direction of the arrows in Fig. 3 and is shown in Fig. 6. In Mode I, any error pattern which is confined to one track can be corrected, for a total of $1 + 9(2^8 - 1) = 2296$ error patterns.

Mode II. To correct errors in two tracks, if it is known which tracks are in error.

Suppose it is known (for instance, by a loss of signal in the tape-reading head) that tracks i and j are in error, where i and j are known. Assume $i < j$.

The decoder first finds

$$S_1 = e_i' + e_j'$$

and

$$S_2 = \begin{cases} T^i e_i' + T^j e_j' & \text{if } j \leqq 7 \\ T^i e_i' & \text{if } j = 8. \end{cases}$$

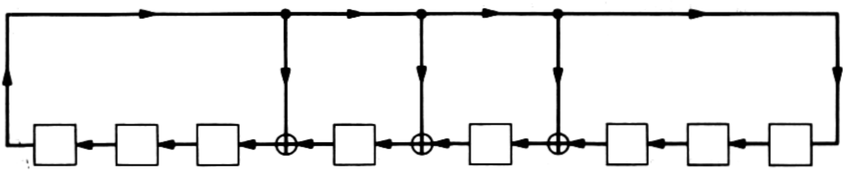

Fig. 6—A shift register which multiplies by T^{-1}.

We allow $e_j = 0$, to include the case where only one track is in error. Solving these equations, we have

$$e_i' = S_1 + e_j'$$

$$e_j' = \begin{cases} (T^i + T^j)^{-1}(T^i S_1 + S_2) & \text{if } j \leqq 7 \\ T^{-i}(T^i S_1 + S_2) & \text{if } j = 8 \end{cases}$$

$$= \begin{cases} (I + T^{j-i})^{-1}(S_1 + T^{-i}S_2) & \text{if } j \leqq 7 \\ S_1 + T^{-i}S_2 & \text{if } j = 8 \end{cases}$$

$$= M_{i,j}(S_1 + T^{-i}S_2),$$

where

$$M_{i,j} = \begin{cases} (I + T^{j-i})^{-1} & \text{if } j \leqq 7 \\ I & \text{if } j = 8 \end{cases}$$

is a matrix which can be calculated in advance (and is written out in Ref. 1). Note that, if $j \leqq 7$, $M_{i,j}$ only depends on $j - i$, so only eight different M's are required.

Decoding in Mode II proceeds as follows. First, S_1 and S_2 are found. Then S_2 is multiplied i times by T^{-1}, added to S_1, and the sum is fed into a circuit which multiplies by $M_{i,j}$ to produce e_j'. Then $e_i' = S_1 + e_j'$. Finally the errors are corrected by adding e_i to track i and e_j to track j.

Observe that in this mode the number of error patterns corrected is $2^8 \cdot 2^8 = 2^{16}$ (for there are 2^8 possibilities each for e_i and for e_j). On the other hand, there are *exactly* $2^8 \cdot 2^8 = 2^{16}$ distinct syndromes. Therefore this code is optimal.

IV. REMARKS

(*i*) This description has neglected certain details of how the data are actually written on the tape—see Ref. 1 for further information.

(*ii*) A similar code exists for n-track tape, for any value of n. The code-words contain $n(n-1)$ bits, arranged in an $n \times (n-1)$ rectangle. There are $n^2 - 3n + 2$ information bits and $2n - 2$ check bits in each code word, for an efficiency of $(n-2)/n$. For example, the efficiency drops to 0.6 if $n = 5$. The code is constructed in exactly the same way, the only difference being in the matrix T. For n-track tape, T should be chosen to be the companion matrix of an irreducible polynomial $g(x)$ of degree $n - 1$. The code will still correct $(n-1)$-bit error patterns in one or two tracks. (There will be several different irreducible polynomials $g(x)$ to choose from. Patel and Hong chose one which was symmetrical about the middle, had the lowest possible exponent, and contained the fewest terms.)

V. SUMMARY

This paper describes the Patel-Hong code for nine-track tape. A code-word contains 72 bits, arranged in a 9×8 rectangle, with 56 information bits and 16 check bits. Encoding and decoding can be done using fairly simple circuitry. There are two modes of decoding. In Mode I, any eight-bit error in any one track can be corrected (even if it is not known which track is in error). In Mode II, eight-bit errors in any two tracks can be corrected, provided it is known which tracks are in error. The generalization of this code to n-track tape is briefly described.

APPENDIX

Proof of Eq. (4)

Define the column vector $s^{(0)} = (1, 0, 0, 0, 0, 0, 0, 0)'$, and let $s^{(i)} = T^i s^{(0)}$. Then the jth column of T^i is $s^{(i+j-1)}$. Equation (1) can be written as

$$[I, T, T^2, \cdots, T^7] \begin{bmatrix} B_0 \\ B_1 \\ \vdots \\ B_7 \end{bmatrix} = 0.$$

Taking the columns of the left-hand matrix in the order 1, 9, 17, $\cdots$, 57; 2, 10, 18, $\cdots$ and remembering eq. (3), we can rewrite the last equation as

$$[I, T, T^2, \cdots, T^7] \begin{bmatrix} Z_0' \\ Z_1' \\ \vdots \\ Z_7' \end{bmatrix} = 0.$$

This is eq. (4). Q.E.D.

REFERENCES

1. A. M. Patel and S. J. Hong, "Optimal Rectangular Code for High Density Magnetic Tapes," IBM J. Research Devel., *18* (1974), pp. 579–588.
2. S. J. Hong and A. M. Patel, "A General Class of Maximal Codes for Computer Applications," IEEE Trans. Comput., *C-21* (1972), pp. 1322–1331.
3. D. T. Brown and F. F. Sellers, Jr., "Error Correction for IBM 800-bit-per-inch Magnetic Tape," IBM J. Research Devel., *14* (1970), pp. 384–389.
4. R. T. Chien, "Memory Error Control: Beyond Parity," IEEE Spectrum, *10* (July 1973), pp. 18–23.