# $(b_1, b_2)$ -Optimal Byte Correcting Codes

Vinod Tyagi<sup>1</sup>, Amita Sethi<sup>2</sup>

<sup>1</sup>Department of Mathematics, Syamlal College (Eve.) (Universityof Delhi), Shahdara, Delhi 110032, India 2Research Scholar, Department of Mathematics, Universityof Delhi, Delhi 110007, India

**Abstract** — In this correspondence, we show how to generate optimal binary linear codes organized in bytes which can correct a fixed burst of length  $b_1$  (fixed) in one byte and burst of length  $b_2$  (fixed) in the remaining bytes.

In an (n,k) linear code, if there are m bytes of length  $\beta$ , then  $n = m\beta$ .

**Keywords** — *Optimal codes, Byte organized memory, syndromes, burst of length b (fixed), parity check matrix..* 

## I. INTRODUCTION

In most memory and storage system, the information is stored in bytes. In such byte oriented memories, whenever an error occurs, it is generally in the form of a burst. Thus error correction in such system means correcting all errors that occur in the same byte. Therefore we consider byte correcting codes for such storage system.

Most of the studies under byte correcting codes is with respect to the usual definition of burst according to which

'A burst of length b is a vector whose all the nonzero components are confined to some b consecutive positions, the first and the last of which is nonzero'.

Tuvi Etzion [6] has defined five types of bytecorrecting codes according to different sizes of the bytes viz.

Type 1. All bytes have the same size

Type 2. One byte of size  $n_1$  and other bytes of size  $n_2$ .

Type 3. Each byte is of either size  $n_1$  or size  $n_2$ .

Type 4. The size of each byte is a power of 2.

Type 5. All the other cases.

For more information on these byte correcting codes and their applications, the reader is referred to [2], [3], [5].

There is another definition of burst due to Chien and Tang [4] with a modification due to Dass [1], named as CTD burst. According to this definition,

'A CTD burst of length b is a vector whose all the non-zero components are confined to some b-consecutive positions, the first of which is non-zero and the number of its starting positions is (n-b+1)'.

We also call such burst as a burst of length b(fixed). It may be noted that according to this definition, (1000000) will be considered as a burst of length up to 7, whereas (0001000) will be a burst of length at most 4.

In this correspondence, we develop byte-oriented codes with respect to this definition of burst. We consider optimal codes where all bytes have the same size but they can correct different CTD burst in different bytes. This situation is possible in codes where it is known that a particular type of error may occur within a specified number of bytes and if one desires to increase a byte in the block length, it is natural to expect some more errors among the additional digits.

However, the errors which are likely to occur in the additional digits need not necessarily be of the type existing in earlier bytes. So there is a need to study optimal  $(b_1, b_2)$  burst correcting codes. We would be interested to find when such optimal byte correcting codes exist and when they can not.

Tyagi and Sethi [7] proved a lower bound over the number of party check digits required for a code that corrects different fixed burst of length  $b_1, b_2$ and  $b_3$  in first  $n_1$ , next  $n_2$  and last  $n_3$  digits,  $n_1 + n_2 + n_3 = n$ . The bound is based on the fact that the number of cosets is at least as large as the number of error patterns to be corrected and is given as

$$q^{r} \ge 1 + \sum_{i=1}^{3} (n_{i} - b_{i} + 1)(q - 1)q^{b_{i} - 1}$$
(1)

For binary case and for byte oriented codes with all bytes of the same size  $\beta$  and redundancy *r*, the bound turns out to be

$$q^{r} \ge 1 + \sum_{i=1}^{3} (\beta - b_{i} + 1)q^{b_{i} - 1}$$
(2)

A linear code which satisfies (2) with equally is called optimal. This gives

$$q^{r} = 1 + \sum_{i=1}^{3} (\beta - b_{i} + 1)q^{b_{i}-1}$$
(3)

The result in (3) can be generalized to m bytes correcting m different fixed bursts of length  $b_i$ , i = 1, 2, ..., m and can be stated as

$$q^{r} = 1 + \sum_{i=1}^{3} (\beta - b_{i} + 1)2^{b_{i} - 1}$$
(4)

As we have mentioned earlier, we are interested in  $(b_1, b_2)$  -optimal burst correcting codes within bytes of size  $\beta$ .

The rest of this correspondence is organized as follows: In section 2, we construct  $(m\beta, m\beta - r)$  optimal byte oriented codes for  $b_1 = 1$  and  $b_2 = 2$  (fixed) and for each  $r \ge 3$ . Conclusion and a list of open problems are suggested in section 3.

## **II.** $(b_1, b_2)$ -Codes for byte-Oriented memories

A trivial example is a optimal (1,2)-(6,3) code which corrects all single errors in the first byte of length  $\beta = 3$  and all burst of length 2(fixed) in the second byte. Its parity check matrix may be written as

$$H = \begin{bmatrix} 110 & 100\\ 010 & 010\\ 111 & 001 \end{bmatrix}$$
(5)

Other simple example is an optimal (1,2)-(12,8) burst correcting code which corrects all burst of length  $b_1 = 1$  in the first byte of length  $\beta = 3$  and all burst of length  $b_2 = 2$  (fix) in the remaining m-1=3 bytes. Its parity check matrix may be written as

$$H = \begin{bmatrix} 110 & 100 & 001 & 010\\ 111 & 010 & 000 & 100\\ 111 & 000 & 100 & 110\\ 100 & 001 & 010 & 101 \end{bmatrix}$$
(6)

It can be verified that this code correct all single errors in the first byte and all burst of length 2(fixed) in the remaining three bytes.

Now, we first give a necessary condition for the existence of such (1,2)-optimal burst correcting codes with bytes of length  $\beta$ . The number of different bursts of length 2(fixed) within a byte is  $2\beta-2$  and  $\beta$  is the number of single errors in a byte. The total number of nonzero vectors of length r is  $2^r - 1$ . Hence the total number of bytes in such a code with redundancy r is

$$\left\lceil \frac{2^r - 1}{2\beta - 2} \right\rceil + 1$$

if the remainder *R* for  $\frac{2^r - 1}{2\beta - 2}$  is equal to  $\beta$ . In the example given above for (12,8) burst correcting code, on applying condition (5), for r = 4 and  $\beta = 3$  we have

$$m = \left\lceil \frac{15}{4} \right\rceil + 1 = 4 \qquad (\text{remainder} = 3)$$

It can be verified that in this case, for any other size of  $\beta > 3$ , (1,2) optimal byte oriented codes do not exist. So condition (5) is also sufficient.

Another interesting example is with redundancy r = 5. Applying necessary condition in this case, we note that the ratio  $\frac{2^5 - 1}{2\beta - 2} = \frac{31}{2\beta - 2}$  has remainder equal to 5 only in two cases. One when  $\beta = 3$  and other when  $\beta = 7$ . This shows the existence of (24,9) and (21,16) byte oriented codes with number of bytes m = 8 and m = 3. The parity-check matrix of these codes may be written as

## When $\beta = 3$ , m = 8

Note that two or more different form of byte oriented codes can exist for any  $(m\beta, m\beta - r)$  optimal code.

Let us assume that  $\left\lceil \frac{2^r - 1}{2\beta - 2} \right\rceil + 1 = m$  and there

exists (1,2) burst correcting code of length  $m\beta$  with m bytes of size  $\beta$  and redundancy r. We will describe now how to construct the parity check matrix of such (1,2)-  $(m\beta, m\beta - r)$  optimal code with m bytes of size  $\beta$  and redundancy r.

Let *H* be the parity check matrix of (1,2) optimal  $(m\beta, m\beta - r)$  burst correcting code with bytes of size  $\beta$ . The columns of  $H = [h_1, h_2, ..., h_n]$  are the elements of  $GF(2^r)$ . There will always exist a vector  $j = (a_1, a_2, ..., a_n)$ ,  $a_i \in GF(2^r)$  such that the sums of at most two adjacent elements within m-1 bytes of size  $\beta$ , whose number is  $(2\beta - 2) (m-1)$ , along with all single elements in one byte of size  $\beta$  give each of the nonzero element of  $GF(2^r)$  exactly once. Alternatively, the *j*th column  $h_j$  can be added to the parity check matrix *H* if

- 1.  $h_j$  is different from all preceding j-1 columns in the first byte.
- 2.  $h_j$  is different from all preceding j-1 columns and also different from the sum of any two adjacent columns.

The sum of 1 and 2 comes out to be  $(2\beta-2)(m-1)+\beta$ . Thus the *j*th column  $h_j$  can always be added provided  $2^r$  is greater than this sum.

Thus we obtain the following result:

**Theorem.** There will always exist an  $(m\beta, m\beta - r)$  linear byte oriented code that corrects all bursts of length  $b_1 = 1$  in the first byte and all bursts of length  $b_2 = 2$  (fix) in the remaining n-1 bytes of size  $\beta$ ,  $m\beta = n$ , satisfying the inequality

 $2^r \ge 3 + 2m\beta - 2m - \beta$ 

#### **III.** CONCLUSION

In this paper, we gave a construction for optimal  $(m\beta, m\beta - r)$  (1,2) burst correcting by oriented codes for  $r \ge 3$ . We have also given the existence of (12,8) byte correcting (1,2) optimal code with bytes of size  $\beta = 3$  and redundancy r = 4; (24,19), (21,16) byte correcting (1,2) optimal code with bytes of size  $\beta = 3$  and 7 and redundancy r = 5. We conjecture that there exist an infinite class of (1,2) optimal byte oriented  $(m\beta, m\beta - r)$  codes for redundancy  $r \ge 3$ . Some other interesting questions are related to the

existence of different categories of  $(b_1, b_2)$  -byte oriented optimal codes viz.

- 1. If there exist other  $(b_1, b_2)$  byte oriented codes for  $b_1 \neq 1$  and  $b_2 \neq 2$  for  $r \ge 3$ ; and
- 2. If there exist such  $(b_1, b_2)$  byte oriented nonbinary codes also.

#### ACKNOWLEDGEMENT

The authors are thankful to Prof. B.K.Dass for his helpful suggestions in the preparation of this work.

#### REFERENCES

- B.K. Dass (1980), On a burst error correcting code, J. Inf. Optimiz. Sci., Vol. 1, No. 3, pp. 291-295.
- [2] C.L. Chen (1983), Error correcting codes with byte error detection capabilities, IEEE Trans. Comput., Vol. C-32, pp. 615-621.
- [3] C.L. Chen (1986), Byte oriented error correcting codes with semiconductor memory systems, IEEE Trans. Comput., Vol. C-35, pp. 646-648.
- [4] R.T. Chien and D.T. Tang (July 1965), On definition of a burst, IBM J. Res. Develop., Vol. 9, pp. 292-293.
- [5] T.R.N. Rao and E. Fujiwara (1989), Error Control Coding for computer System, Englewood Cliff, NJ: Prentice Hall.
- [6] Tuvi Etzion(1998), Perfect byte correcting codes, IEEE Trans. Inform. Theory, Vol. 44, pp. 3140-3146.
- [7] Vinod Tyagi and Amita Sethi (2007), (n<sub>1b<sub>1</sub></sub>, n<sub>2b<sub>2</sub></sub>, n<sub>3b<sub>3</sub></sub>) Linear Codes over GF(2), Beykent University Journal of Science and Technology, 3(2), 301-319.