Some New EC/AUED Codes

Jehoshua Bruck and Mario Blaum
IBM Research Division
Almaden Research Center
San Jose, CA 95120

Abstract

We present a new construction that differs from the traditional way of constructing systematic EC/AUED codes. Most authors take a systematic t-error-correcting code and then they append a tail in such a way that the new code can detect more than t errors when they are unidirectional. In our construction, we modify the t-error-correcting code in such a way that the weight distribution of the original code is reduced. We then have to add a smaller tail. Frequently, we have less redundancy than the best available systematic t-EC/AUED codes.

1 Introduction

The problem of finding systematic error correcting/all unidirectional error detecting codes (EC/AUED) has received wide attention in recent literature [1]-[9].

The way most authors construct a t-EC/AUED code is as follows: first the information bits are encoded into a systematic t-EC (error-correcting) code. A tail is then added as further redundancy in such a way that the resulting code can detect all unidirectional errors. This tail is a function of the weight of the codeword in the t-EC code. The shorter the tail, the smaller the redundancy, so authors concentrate in obtaining a tail as short as possible. To the moment of this writing, the record is held by [1]. The construction in [1] heavily depends on the best asymmetric error-correcting codes available. The optimality of the construction is still an open problem.

In this paper, we propose a slightly different approach. For t-EC codes, we use codes that contain the all-1 vector (for instance, BCH codes and the Golay code have this property). When choosing a

codeword, we take either a codeword or its complement, according to which of the two has smaller weight. We have to pay a bit for this operation, but the weight distribution is reduced by half. We then append a tail in the way described in [1]. Overall, we will often gain in redundancy.
The construction will be described in detail in the next section. We then consider the problems of encoding and decoding. Although the new codes are not strictly systematic, they are very close to being so. We will see that encoding and decoding are nearly as simple as in the systematic case.
We also provide tables and examples.

2 Construction

As stated in the introduction, the construction in [1] depends on a tail that is appended to each codeword in a t-EC code. This tail is a function of the weight of the codeword. In fact, it comes from a so called descending tail matrix of strength s. For the sake of completeness, we give the definition.
Given two vectors $\underline{u}$ and $\underline{v}$, denote by $N(\underline{u},\underline{v})$ the number of 1$\to$0 transitions from $\underline{u}$ to $\underline{v}$. (for example, if $\underline{u} = 10101$ and $\underline{v} = 00011$, then $N(\underline{u},\underline{v}) = 2$).

Definition 2.1 A descending tail matrix of strength s is an $m \times r$ $\{0,1\}$-matrix with rows $\underline{t}_i$, $0 \leq i \leq m-1$, such that for all $0 \leq i \leq j \leq m-1$,

$$N(\underline{t}_i,\underline{t}_j) \geq min\{s, \lceil (j-i)/2 \rceil\}.$$

We denote this matrix by $T(m,r;s)$. Table 1 gives a list of parameters for the descending tail matrices obtained in [1].
The next theorem [1] is the key for constructing t-EC/AUED codes.

Theorem 2.1 Let C' be a t-EC code of length n' and let T be a descending tail matrix $T(n'+1,r;t+1)$ with rows $\underline{t}_i$, $0 \leq i \leq n'$. Then

$$C = \{(\underline{c}, \underline{t}_{w(\underline{c})}) : \underline{c} \in C'\}$$

is a t-EC/AUED code of length $n = n' + r$ ($w(\underline{c})$ denotes the Hamming weight of vector $\underline{c}$).

The next construction is our main result.

Construction 2.1 Let k be the number of information bits. Assume that we want to construct a t-EC/AUED code. Then:

1. Choose an $[n', k+1, d]$ EC code ($d \geq 2t+1$) C' containing the all-1 vector with n' as small as possible.
2. Choose a $T(\lfloor n'/2 \rfloor + 1, r, t+1)$ descending tail matrix T with rows $\underline{t}_i$, $0 \leq i \leq \lfloor n'/2 \rfloor + 1$ and r as small as possible.
3. Let C be the code

$$C = \{(\underline{c}, \underline{t}_{w(\underline{c})}) : \underline{c} \in C', w(\underline{c}) \leq n'/2\}.$$

The code C obtained in the previous construction is t-EC/AUED since the subset of codewords of weight $\leq \lfloor n/2 \rfloor$ is still a t-EC. According to Theorem 2.1, the tail makes it t-EC/AUED.

Example 2.1 Assume $k = 3$ and $t = 1$. According to Construction 2.1, we consider the [7,4,3] Hamming code whose generator matrix is

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

We easily see that the codewords of weight ≤ 3 are:

$$\begin{aligned} \underline{c}_0 &= 0000000 \\ \underline{c}_1 &= 1000011 \\ \underline{c}_2 &= 0100101 \\ \underline{c}_3 &= 0010110 \\ \underline{c}_4 &= 1001100 \\ \underline{c}_5 &= 0101010 \\ \underline{c}_6 &= 0011001 \\ \underline{c}_7 &= 1110000 \end{aligned}$$

According to [1], we can use the $T(4, 2; 2)$ matrix

$$T = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

The code is then given by the following set of codewords:

$$\begin{aligned} \underline{v}_0 &= 0000000\ \ 11 \\ \underline{v}_1 &= 1000011\ \ 00 \\ \underline{v}_2 &= 0100101\ \ 00 \\ \underline{v}_3 &= 0010110\ \ 00 \\ \underline{v}_4 &= 1001100\ \ 00 \\ \underline{v}_5 &= 0101010\ \ 00 \\ \underline{v}_6 &= 0011001\ \ 00 \\ \underline{v}_7 &= 1110000\ \ 00 \end{aligned}$$

Notice that we have 3 information bits and 6 redundant bits. If we use the construction in [1], we need 7 redundant bits.

3 Encoding and Decoding

In the previous Section we described a t-EC/AUED code but we did not explain how to encode the data. This is very easily done, as we will see.
Assume we want to encode k bits into a t-EC/AUED code C. Choose an $[n', k + 1, 2t + 1]$ code C' containing the all-1 vector (with n' as small as possible) and a $T(\lfloor n'/2 \rfloor + 1, r, t + 1)$ descending tail matrix T (with r as small as possible). The symbol $\oplus$ denotes "exclusive-OR" and $\underline{1}$ denotes the all-1 vector. Then proceed as follows:

Algorithm 3.1 (Encoding Algorithm) Let $\underline{u} = (u_1, u_2, \ldots, u_k)$ be the vector of information bits. Then:

1. Encode $(\underline{u}, 0) = (u_1, u_2, \ldots, u_k, 0)$ into a vector $\underline{c}$ in C'.
2. If $w(\underline{c}) > \lfloor n'/2 \rfloor$ then $\underline{c} \leftarrow \underline{c} \oplus \underline{1}$.
3. Let $\underline{v} = (\underline{c}, \underline{t}_{w(\underline{c})})$ be the output of the encoder, where $\underline{t}_i$, $0 \leq i \leq \lfloor n'/2 \rfloor$, are the rows of T.

Observe that code C' is not required to be systematic. However, if that is the case, the t-EC/AUED code C will be practically systematic, in the sense that the first k bits in codeword $\underline{v}$ will either be the information bits or their complements.

Example 3.1 Consider code C in Example 2.1. Assume that we want to encode $\underline{u} = 010$. The first step is to encode $(\underline{u}, 0) = 0100$ into the $[7,4]$ Hamming code. This gives $\underline{c} = 0100101$. Since $w(\underline{c}) = 3$, $\underline{t}_{w(\underline{c})} = 00$. The encoded vector is then $\underline{v} = (\underline{c}, \underline{t}_3) = 010010100$.
Similarly, assume that we want to encode $\underline{u} = 110$. The encoding of $(\underline{u}, 0) = 1100$ into C' gives $\underline{c} = 1100110$. Since $w(\underline{c}) = 4 > 3 = \lfloor n'/2 \rfloor$, then $\underline{c} = 1111111 \oplus 1100110 = 0011001$. As before, the encoded vector is $\underline{v} = (\underline{c}, \underline{t}_3) = 001100100$.

The decoding is also very simple. Essentially, it works as in [1], with the extra step of taking complements when necessary.

Algorithm 3.2 (Decoding Algorithm) Let C be the EC/AUED code obtained from Construction 2.1. Let $\underline{\hat{v}}$ be the received word and $\underline{\hat{c}}$ the first n' bits of $\underline{\hat{v}}$. Then:

1. Decode $\underline{\hat{c}}$ with respect to C'. If more than t errors, declare an uncorrectable error. Else let $\underline{c}$ be the corrected word.
2. Let $\underline{v} = (\underline{c}, \underline{t}_{w(\underline{c})})$. If $d_H(\underline{\hat{v}}, \underline{v}) > t$ (d_H denotes Hamming distance), then declare an uncorrectable error.
3. Else, let $u_1, u_2, \ldots, u_{k+1}$ be the $k+1$ information bits from codeword $\underline{c} \in C'$. Then, the output of the decoder is given by the vector of length k

$$\underline{u} = (u_1 \oplus u_{k+1}, u_2 \oplus u_{k+1}, \ldots, u_k \oplus u_{k+1}).$$

Example 3.2 Again consider the code of Examples 2.1 and 3.1.

1. Assume we receive $\underline{\hat{v}} = 100101110$. According to the Decoding Algorithm, we first consider $\underline{\hat{c}} = 1001011$. The parity check matrix of C' is

$$H = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

   So, we obtain the syndrome $\underline{s} = \underline{\hat{c}}H^T = 111$ which corresponds to the fourth column of H, hence, $\underline{\hat{c}}$ is decoded as $\underline{c} = 1000011$. Now, $\underline{v} = (\underline{c}, \underline{t}_{w(\underline{c})}) = 100001100$, hence $d_H(\underline{\hat{v}}, \underline{v}) = 2 > 1 = t$. Thus, the decoder declares an uncorrectable error.

2. Assume we receive $\underline{\hat{v}} = 011011000$. As before, $\underline{\hat{c}} = 0110110$, and $\underline{s} = \underline{\hat{c}}H^T = 101$, which corresponds to the second column of H. Hence, $\underline{\hat{c}}$ is decoded as $\underline{c} = 0010110$. So, $\underline{v} = (\underline{c}, \underline{t}_{w(\underline{c})}) = 001011000$ and $d_H(\underline{\hat{c}}, \underline{c}) = 1$. Since $u_4 = 0$, the output of the decoder is $\underline{u} = 001$.

3. Assume we receive $\underline{\hat{v}} = 001110100$. Now $\underline{\hat{c}} = 0011101$, and $\underline{s} = \underline{\hat{c}}H^T = 100$, which corresponds to the fifth column of H. Hence, $\underline{\hat{c}}$ is decoded as $\underline{c} = 0011001$. So, $\underline{v} = (\underline{c}, \underline{t}_{w(\underline{c})}) = 001100100$ and $d_H(\underline{\hat{c}}, \underline{c}) = 1$. Since $u_4 = 1$, the output of the decoder is $\underline{u} = 001 \oplus 111 = 110$.

4 Tables and Comparisons

We have seen in Example 2.1 that we gained one bit with our construction with respect to [1]. In this section, we show that this is not an isolated case.

As stated in Section 2, Table 1 contains the parameters of some descending tail matrices $T(m,r;t+1)$ obtained from [1].

In Tables 2-5 we give the redundancy of t-EC/AUED codes for different values of k that were obtained using Construction 2.1, as well as the redundancy from [1]. In most of the cases, we tie the results from [1], but also quite often we improve upon them, as shown in the tables.

The tables have seven columns. The first column contains the number of information bits k. The second column gives the length n' of the EC-code. Column 3 contains $\lfloor n'/2 \rfloor$. Column 4 gives the number of extra bits r that we have to add to the EC-code in order to obtain a t-EC/AUED code (Construction 2.1). Column 5 gives the total redundancy $n-k=n'-k+r$ used in the Construction. Column 6 gives the redundancy obtained using the codes in [1]. Finally, column 7 indicates the EC-code used (containing the all-1 vector). The subscript "s" indicates a shortened code.

Notice that we use only BCH codes and the Golay code, which are easy to decode, while in [1] the best codes of [10] have been chosen. Sometimes no efficient decoder is known for the best possible code.

In order to shorten a code containing the all-1 vector in such a way that the shortened code also contains the all-1 vector, we use the following lemma:

Lemma 4.1 Let C be an $[n,k,d]$ EC code with parity-check matrix H. Assume that the all-1 vector is in C. Let $\underline{c}$ be a codeword in C such that its nonzero components are $i_1, i_2, \ldots, i_w$, $1 \leq i_1 < i_2 < \ldots < i_w \leq n$. Let $\tilde{H}$ be the matrix obtained by deleting columns $i_1, i_2, \ldots, i_w$ from H. Let $\tilde{C}$ be the $[n-w, k-w, d]$ code whose parity check matrix is $\tilde{H}$. Then the all-1 vector is in $\tilde{C}$.

Proof: The all-1 vector is in $\tilde{C}$ if and only if the sum (modulo 2) of all the columns in $\tilde{H}$ gives the zero column.

Since the all-1 vector $\underline{1}$ is in C, then $\underline{1} \oplus \underline{c}$ is also in C. This vector has zero components $i_1, i_2, \ldots, i_w$. Summing the columns corresponding to the nonzero components, we obtain the zero column. But these columns correspond to the columns in $\tilde{H}$. □

Example 4.1 Consider the [7,4] Hamming code of Example 2.1. Take codeword $\underline{c} = 1110000$. In order to obtain matrix $\tilde{H}$ according to Lemma 4.1, we have to delete the first three columns of matrix H of Example 3.2. This gives

$$\tilde{H} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

The shortened Hamming code has length 4 and dimension 1. The all-1 vector is in the shortened code.

We use this procedure to shorten several of the codes presented in Tables 2-5.

5 Conclusions

A new method for constructing t-EC/AUED codes has been presented. The information bits are encoded first into a t-EC code containing the all-1 vector. The key idea in the construction is reducing the weight distribution of the t-EC code used. Our codes have frequently less redundancy than the best EC/AUED codes previously known. The encoding and decoding procedures are as simple as those of known codes.

References

[1] M. Blaum and H. van Tilborg, "On t-Error Correcting/All Unidirectional Error Detecting Codes," IBM Research Report 5566 (56685), March 1987, to appear in IEEE Trans. on Computers.

[2] B. Bose, "On Systematic SEC/MUED Codes," Proc. FTCS, vol. 11, pp. 265-267, June 1981.

[3] B. Bose and D. K. Pradhan, "Optimal Unidirectional Error Detecting/Correcting Codes," IEEE Trans. on Computers, vol. C-31, pp. 521-530, June 1982.

[4] B. Bose and T. R. N. Rao, "On the Theory of Unidirectional Error Correcting/Detecting Codes," IEEE Trans. on Computers, Vol. C-31, pp. 521-530, June 1982.

[5] S. Kundu, "Design of Testable CMOS Circuits for TSC Systems," Ph. D. Thesis, University of Iowa, May 1988.

[6] D. Nikolos, N. Gaitanis and G. Philokyprou, "t-Error Correcting All Unidirectional Error Detecting Codes Starting from Cyclic AN Codes," Proc. Int. Conf. on Fault-Tolerant Computing, Kissimmee, FL, pp. 318-323, 1984.

[7] D. Nikolos, N. Gaitanis and G. Philokyprou, "Systematic t-Error Correcting/All Unidirectional Error Detecting Codes," IEEE Trans. on Computers, vol. C-35, pp. 394-402, May 1986.

[8] D. K. Pradhan, "A New Class of Error-Correcting/Detecting Codes for Fault-Tolerant Computer Applications," IEEE Trans. on Computers, vol. C-29, pp. 471-481, June 1980.

[9] D. L. Tao, C. R. P. Hartmann and P. K. Lala, "An Efficient Class of Unidirectional Error Detecting/Correcting Codes," IEEE Trans. on Computers, vol. C-37, pp. 879-882, July 1988.

[10] T. Verhoeff, "An Updated Table of Minimum-Distance Bounds for Binary Linear Codes," IEEE Trans. on Information Theory, vol. IT-33, pp. 665-680, Sept. 1987.

t	r	m	t	r	m	t	r	m	t	r	m
1	2	4	2	3	6	3	4	8	4	5	10
1	3	6	2	4	8	3	5	10	4	6	12
1	4	8	2	5	10	3	6	12	4	7	14
1	5	12	2	6	12	3	7	14	4	8	16
1	6	16	2	7	16	3	8	16	4	9	18
1	7	24	2	8	20	3	9	20	4	10	20
1	8	48	2	9	24	3	10	24	4	11	24
1	9	72	2	10	32	3	11	28	4	12	28
1	10	144	2	11	48	3	12	32	4	13	32
1	11	248	2	12	72	3	13	40	4	14	36
1	12	432	2	13	120	3	14	48	4	15	40
			2	14	216	3	15	72	4	16	48
			2	15	392	3	16	120	4	17	56
						3	17	180	4	18	72
						3	18	264	4	19	104
						3	19	488	4	20	156
									4	21	216
									4	22	368

Table 1: Parameters of Some Descending Tail Matrices $T(m, r; t+1)$

k	n'	$\lfloor n'/2 \rfloor$	r	$n-k$	$n-k$ from [1]	EC-Code
3	7	3	2	6	7	Hamming
10	15	7	4	9	10	Hamming
22	28	14	6	12	13	Hamming_s
25	31	15	6	12	13	Hamming
87	95	47	8	16	17	Hamming_s
246	255	127	10	19	20	Hamming
277	287	143	10	20	21	Hamming_s

Table 2: Parameters of some 1-EC/AUED codes

k	n'	$\lfloor n'/2 \rfloor$	r	$n-k$	$n-k$ from [1]	EC-Code
6	15	7	4	13	15	BCH
15	26	13	7	18	20	BCH_s
20	31	15	7	18	20	BCH
45	58	29	10	23	24	BCH_s
50	63	31	10	23	24	BCH
107	122	61	12	27	28	BCH_s
112	127	63	12	27	28	BCH
222	239	119	13	30	31	BCH_s

Table 3: Parameters of some 2-EC/AUED codes

k	n'	$\lfloor n'/2 \rfloor$	r	$n-k$	$n-k$ from [1]	EC-Code
4	15	7	4	15	18	BCH
11	23	11	6	18	21	Golay
15	31	15	8	24	26	BCH
37	56	28	12	31	32	BCH_s
44	63	31	12	31	32	BCH
105	127	63	15	37	38	BCH
214	239	119	16	41	42	BCH_s
483	511	255	18	46	47	BCH

Table 4: Parameters of some 3-EC/AUED codes

k	n'	$\lfloor n'/2 \rfloor$	r	$n-k$	$n-k$ from [1]	EC-Code
38	63	31	13	38	42	BCH
98	127	63	18	47	48	BCH
222	255	127	20	53	54	BCH

Table 5: Parameters of some 4-EC/AUED codes