Noname manuscript No.
(will be inserted by the editor)

Integer codes capable of correcting burst asymmetric errors

Nabin Kumar Pokhrel · Pankaj Kumar Das · Aleksandar Radonjic

Received: date / Accepted: date

Abstract In this paper, we present a class of integer codes capable of correcting burst asymmetric errors. The presented codes are constructed with the help of a computer and have the potential to be used in various practical systems, such as optical networks and VLSI memories. In order to evaluate the performance of the proposed codes, the paper analyzes the probability of erroneous decoding for different bit error rates. The presented codes are also analyzed from a rate-efficiency point of view. The obtained results show that for many data lengths they require less check-bits than optimal burst error correcting codes.

Keywords Integer codes · asymmetric bursts · syndromes · lookup table · error correction · probability.

Mathematics Subject Classification (2010) 94B35 · 94B60 · 94B65

1 Introduction

Binary-oriented codes have been intensively studied in the literature for the last 70 years. Unlike them, the codes over the ring of integers modulo m, $\mathbb{Z}_m$, have been scarcely studied. The reason pertaining to this could be the weak error correcting capabilities of these codes [11]. However, inspired by the work of Blake [1], [2], Varshamov and Tenengolz [13], and Levenshtein and Vinck [4], who constructed perfect (d, k)-codes capable of correcting single-peak shifts,

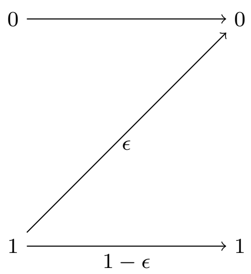
Nabin Kumar Pokhrel, Pankaj Kumar Das
Department of Mathematical Sciences, Tezpur University, Napaam, Sonitpur, Assam-784028, India.
E-mail: nabinkr@tezu.ernet.in, pankaj4thapril@yahoo.co.in, pankaj4@tezu.ernet.in

Aleksandar Radonjic
Institute of Technical Sciences of the Serbian Academy of Sciences and Arts, Belgrade, Serbia.
E-mail: sasa_radonjic@yahoo.com

the authors of [14] showed that the concept of integer coding can be very useful in systems where the messages are represented by integers (e.g. coded modulation and magnetic recording). This fact motivated the authors of [6]-[10] to construct several classes of integer codes that are suitable for use in computer-based systems.

However, none of these codes is able to correct burst asymmetric errors (BAEs). These errors occur in all systems in which binary 1s represent the presence of some particles or energy. One example of such systems is optical networks without optical amplifiers. In these networks, photons (represented by binary 1s) may fade or fail to be detected, but new photons cannot be generated. Hence, upon transmission, only asymmetric ($1 \to 0$) errors may occur [5]. The situation is same with VLSI memories in which charges (represented by binary 1s) may leak with time, but new charges cannot be added [11]. This is the reason why the mentioned systems are often modeled by the Z-channel with crossover probability ϵ (Fig. 1).

Fig. 1: Z-Channel



Motivated by this fact, in this paper, we present a class of integer codes capable of correcting BAEs up to length l. The presented codes are a generalization of codes from [10], which means that they do not have to be interleaved in order to be able to correct all BAEs. Thanks to this, they have the potential to be used in many practical systems, including those with short codewords (e.g. VLSI circuits and memories).

The organisation of this paper is as follows: Section 2 deals with the construction of integer codes capable of correcting all BAEs of length up to l. The bit error rate (BER) and the probability of erroneous dcoding for these codes are determined in Section 3. This section also compares the proposed codes with other codes of same or similar properties, while Section 4 concludes the paper.

2 Construction of Codes

In this section, we discuss the construction of integer codes correcting BAEs up to length l. We also illustrate these codes with a suitable example. We begin with the following definition.

Definition 1 (**Integer Codes**) [14] For $m, M, N \in \mathbb{N}$, pre defined $H \in {\mathbb{Z}_m}^{M\times N}$ and $d \in \mathbb{Z}_m^M$. Then, an integer code is defined by $\Big\{a \in \mathbb{Z}_m^N : aH^T = d\Big\}$, where H can be seen as the parity check matrix.

By considering $N = k+1, m = 2^b - 1, M = 1$ and $d = 0$ in Definition 1, a $((k+1)b, kb)$ integer code is defined as $\left\{(B_1B_2 \dots B_kC_B) \in \mathbb{Z}_{2^b-1}^{k+1} \mid (B_1B_2\dots B_kC_B)\begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_k \\ -1\end{pmatrix} \pmod{2^b-1} = 0\right\}$, where the C_i's are coefficients from the ring $\mathbb{Z}_{2^b-1}$, the B_i's are data bytes and the C_B is the check-byte. It is clear that each byte, including the check-byte, can be written as $B_i = (x_{b-1}, x_{b-2}, \dots, x_0) = [x_{b-1}2^{b-1} + x_{b-2}2^{b-2} + \dots + x_0 2^0] \pmod{2^b-1}$.

Definition 2 An integer code that corrects all BAEs of length up to l is called an integer B_lAEC code.

2.1 Encoding Procedure

A codeword comprises of k b-bit data bytes and one b-bit check byte. Accordingly, if $c = B_1B_2\dots B_kC_B$ is the sent codeword and $r = \bar{B}_1\bar{B}_2\dots\bar{B}_k\bar{C}_B$ the received one corrupted by an error e such that $e = c - r$, then the **syndrome** is given by

$$\begin{aligned} S(r) &= [cH^T - eH^T] \pmod{2^b-1} \\ &= [-eH^T] \pmod{2^b-1} \\ &= -\Big[(B_1B_2\dots B_kC_B) - (\bar{B}_1\bar{B}_2\dots\bar{B}_k\bar{C}_B)\Big]\begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_k \\ -1\end{pmatrix} \pmod{2^b-1} \\ &= [C_{\bar{B}} - \bar{C}_B] \pmod{2^b-1}, \end{aligned}$$

where, $C_{\bar{B}} = [C_1\bar{B}_1 + C_2\bar{B}_2 + \ldots + C_k\bar{B}_k] \pmod{2^b-1}$. Keeping this in mind, we give below the definitions of the corresponding syndrome sets.

Definition 3 [10] The syndrome set of all BAEs of length up to l corrupting one b-bit byte is defined by

$$S_1 = \bigcup_{i=1}^{k+1}\Big[-C_i.\epsilon_{b,l}\Big] \pmod{2^b-1}, \tag{1}$$

where $\epsilon_{b,l} = e_{b,1} \cup e_{b,2} \cup \dots \cup e_{b,l}$ and $e_{b,t} = 2^r(1, 3, \dots, 2^t-1)$, $0 \le r \le b-t$.

Definition 4 The syndrome set of all BAEs up to length l corrupting two adjacent b-bit bytes is defined by

$$S_2 = \bigcup_{i=1}^{k} \Big[C_i P_r + C_{i+1} Q_s\Big] \pmod{2^b - 1}, \tag{2}$$

where $C_{k+1} = -1$, $P_r = \{-2^{r-1} + p_{r-2}2^{r-2} + \cdots + p_0\}$, $Q_s = \{q_{b-1}2^{b-1} + q_{b-2}2^{b-2} + \cdots + (-1)2^{b-s}\}$, $p_i, q_i \in \{-1, 0\}$, $1 \leq r, s < l$ and $max\{r + s\} = l$.

Using the above definitions, we can state the following theorems.

Theorem 1 *A $((k+1)b, kb)$ integer B_lAEC code can correct all BAEs up to length l if there exist k mutually distinct coefficients $C_i \in \mathbb{Z}_{2^b-1} \setminus \{0, 1\}$ such that*

1. $\mid S_1 \mid = (k+1)[2^{l-1}(b - l + 2) - 1]$.
2. $\mid S_2 \mid = k \sum_{i=2}^{l} \alpha_i$, *where* $\alpha_i = (i-1)2^{i-2}$.
3. $S_1 \cap S_2 = \phi$.

Proof Condition 1 has been proved in Theorem 1 of [10]. For Condition 2, we follow the same technique as Condition 1. For a BAE of length 2 corrupting two adjacent b-bit bytes, the syndrome element will be of the type $C_iP_1 + C_{i+1}Q_1 = C_i(-1) + C_{i+1}(-2^{b-1})$. From this it is obvious that $\alpha_2 = 1$. For a BAE of length 3 corrupting two adjacent b-bit bytes, the syndrome elements are of the type $C_iP_1 + C_{i+1}Q_2$ and $C_iP_2 + C_{i+1}Q_1$. The possibilities in this case are $C_i(-1) + C_{i+1}(q_{b-1}2^{b-1} - 2^{b-2})$ and $C_i(-2^1 + p_0) + C_{i+1}(-2^{b-1})$. Hence, it is clear that $\alpha_3 = 2 \times 2 = 4$. Further, for a BAE of length 4, the syndrome elements are of type $C_i(-1) + C_{i+1}(q_{b-1}2^{b-1} + q_{b-2}2^{b-2} - 2^{b-3})$, $C_i(-2 + p_0) + C_{i+1}(q_{b-1}2^{b-1} - 2^{b-2})$ and $C_i(-2^2 + p_1 2^1 + p_0) + C_{i+1}(-2^{b-1})$. From this it is easy to conclude that $\alpha_4 = 3 \times 2^2 = 12$. Following this pattern, we observe that the possibilities for syndrome elements corresponding to a BAE of length l corrupting two adjacent b-bit bytes are $C_iP_1 + C_{i+1}Q_{l-1}$, $C_iP_2 + C_{i+1}Q_{l-2}$, $\ldots$, $C_iP_{l-1} + C_{i+1}Q_1$. Thus, the pattern of syndrome elements in this case will be $C_i(-1) + C_{i+1}(q_{b-1}2^{b-1} + \cdots + q_{b-l-1}2^{b-l-1} - 2^{b-l})$, $C_i(-2 + p_0) + C_{i+1}(q_{b-1}2^{b-1} + \cdots + q_{b-l-2}2^{b-l-2} - 2^{b-l-1})$, $\ldots$, $C_i(-2^{b-l} + p_{b-l+1}2^{b-l+1} + \cdots + p_0) + C_{i+1}(-2^{b-1})$ and $\alpha_l = (l-1)2^{l-2}$.

Taking account of the orders $\alpha_1, \alpha_2, \ldots, \alpha_l$ and $k+1$ distinct coefficients, we conclude that $\mid S_2 \mid = k \sum_{i=2}^{l} \alpha_i$.

Finally, Condition 3 ensures that the syndromes caused by BAEs corrupting one b-bit byte are different from those corrupting two adjacent b-bit bytes. Hence, the codes satisfying the conditions 1 to 3 are $((k+1)b, kb)$ integer B_lAEC codes.

□

Theorem 2 *Let $\zeta_{b,l} = S_1 \cup S_2$ be the set of syndromes for a $((k+1)b, kb)$ integer B_lAEC code. Then* $\mid \zeta_{b,l} \mid = (k+1)[2^{l-1}(b - l + 2) - 1] + k \sum_{i=2}^{l} \alpha_i$.

Proof From Theorem 1, it is clear that $|\ \zeta_{b,l}\ | = (k+1)[2^{l-1}(b-l+2)-1]+k\sum_{i=2}^{l}\alpha_i$. □

Note: For a $((k+1)b, kb)$ integer B_lAEC code, the sets $-C_1\epsilon_{b,l}$, $-C_2\epsilon_{b,l}$, ..., $-C_k\epsilon_{b,l}$, $-C_{k+1}\epsilon_{b,l}$ in (1) and $C_iP_r + C_{i+1}Q_s$ for $1 \le i \le k$ in (2) are required to be all mutually disjoint. The coefficients C_i (Table 1) can be obtained by using a suitable computer program (the Python code we used can be found at the following link:
https://docs.google.com/document/d/1GL01xI-9u5_e2RW4flZsIYbfwMyiE__d/edit?usp=sharing&ouid=111601884470217455043&rtpof=true&sd=true).

Table 1: The coefficients for some $((k+1)b, kb)$ integer B_lAEC codes

b	l	Coefficients
8	3	29
9	2	7, 11, 13, 23, 31, 37, 55, 61, 63, 103, 117, 119, 125
9	3	2, 19, 93
10	2	5, 7, 9, 29, 35, 41, 49, 53, 61, 63, 71, 73, 79, 89, 95, 115, 125, 127, 149, 205
10	3	2, 25, 101, 239
10	4	2, 53
11	4	19, 21, 311
12	3	2, 9, 29, 61, 97, 127, 159, 199, 245, 249, 251, 281, 447, 615, 669, 671
12	4	37, 77, 211
13	4	2, 31, 159, 269, 319, 463, 507, 675, 921, 2811
14	4	25,37, 143, 157, 269, 509, 739, 805, 829, 1627, 2495, 2797, 3581, 3949, 5983
15	4	19, 23, 41, 67, 103, 113, 131, 409, 509, 563, 599, 703, 725, 903, 1145, 1301, 1415, 1587, 1683, 1745, 1979, 2613, 3383, 4709, 6015, 6127, 6133, 7093, 7415, 7807, 7925
16	3	2, 11, 43, 61, 67, 79, 89, 101, 105, 107, 113, 117, 121, 127, 131, 139, 143, 149, 151, 153, 157, 163, 167, 169, 179, 181, 187, 191, 193, 197, 199, 207
16	4	47, 59, 61, 113, 121, 127, 169, 199, 251, 271, 323, 331, 383, 431, 437, 449, 493, 509, 551, 557, 563, 575, 577, 593, 609, 629, 647, 661, 683, 697, 701, 713
18	4	43, 71, 97, 107, 131, 151, 163, 173, 179, 181, 191, 227, 241, 269, 271, 277, 281, 283, 307, 311, 317, 323, 331, 337, 347, 349, 353, 357, 359, 361, 367, 373
20	4	31, 81, 113, 149, 167, 179, 211, 223, 227, 233, 241, 245, 257, 263, 277, 281, 283, 289, 293, 307, 311, 313, 317, 323, 331, 337, 347, 349, 353, 357, 359, 361
25	4	23, 43, 131, 137, 149, 167, 173, 197, 199, 233, 241, 269, 271, 277, 281, 283, 289, 293, 307, 311, 317, 323, 331, 337, 347, 349, 353, 357, 359, 361, 367, 373

32	3	2, 19, 47, 61, 73, 97, 99, 103, 109, 117, 121, 127, 131, 137, 139, 143, 149, 151, 153, 157, 163, 167, 169, 171, 173, 179, 181, 187, 191, 193, 197, 199
32	4	31, 81, 113, 149, 167, 179, 211, 223, 227, 233, 241, 245, 257, 263, 269, 271, 277, 281, 283, 289, 293, 307, 311, 313, 317, 323, 331, 337, 347, 349, 353, 357

2.2 Decoding Procedure

If the value of the syndrome S is different from zero, the decoder will lookup the syndrome table (ST). It always has $|\zeta_{b,l}|$ entries (Theorem 2) and can be generated by substituting the values of l, b and C_i into (1)-(2). The purpose of each entry is to describe the relationship between the nonzero syndrome (element of the set $\zeta_{b,l}$), error locations $(i, i+1)$ and error vectors (e_1, e_2) (Fig. 2). In order to able to correct BAEs, the decoder must find the entry

Fig. 2: Bit-width of one ST entry

Element of $\zeta_{b,l}$	Error location (i)	Error value (e_1)	Error location $(i+1)$	Error value (e_2)
$\longleftarrow b \longrightarrow$	$\longleftarrow \lceil log_2(k+1)\rceil \longrightarrow$	$\longleftarrow b \longrightarrow$	$\longleftarrow \lceil log_2(k+1)\rceil \longrightarrow$	$\longleftarrow b \longrightarrow$

with the first b bits as that of the syndrome S. If the ST is sorted in increasing order (according to the elements of $\zeta_{b,l}$), this task will be completed after $\eta_{T,L}$ table lookups $(1 \le \eta_{TL} \le \lfloor log_2 \mid \zeta_{b,l} \mid \rfloor + 2))$ [10]. In the next step, using the error correction data, the decoder will perform one of the following operations:

- For BAEs of length up to l corrupting the check byte:

$$C_B = [\bar{C}_B + e_1] \pmod{2^b - 1}, e_1 \in \epsilon_{b,l}; \tag{3}$$

- For BAEs of length up to l corrupting the i^{th} data byte $(1 \le i \le k)$:

$$B_i = [\bar{B}_i + e_1] \pmod{2^b - 1}, e_1 \in \epsilon_{b,l}; \tag{4}$$

- For BAEs of length up to l corrupting the i^{th} and $(i+1)^{th}$ data byte $(1 \le i \le k-1)$:

$$\begin{aligned} B_i &= [\bar{B}_i + e_1] \pmod{2^b - 1},\ e_1 \in \epsilon_{b,l},\ -e_1 \in P_r; \\ B_{i+1} &= [\bar{B}_{i+1} + e_2] \pmod{2^b - 1},\ e_2 \in \epsilon_{b,l},\ -e_2 \in Q_s; \end{aligned} \tag{5}$$

- For BAEs of length up to l corrupting the k^{th} data byte and the check byte:

$$\begin{aligned} B_k &= [\bar{B}_k + e_1] \pmod{2^b - 1},\ e_1 \in \epsilon_{b,l},\ -e_1 \in P_r; \\ C_B &= [\bar{C}_B + e_2] \pmod{2^b - 1},\ e_2 \in \epsilon_{b,l},\ -e_2 \in Q_s; \end{aligned} \tag{6}$$

To make the above procedure more understandable, we will illustrate it on the example of the $(48, 40)$ B_2AEC code.

Example 1 Let $b = 8$, $l = 2$ with $C_1 = 5, C_2 = 7, C_3 = 9, C_4 = 25, C_5 = 29$ and $C_6 = -1$. According to Theorem 2, the ST will have $|\ \zeta_{8,2}\ | = 95$ entries (Table 2). Now, suppose we want to transmit 40 data bits,
11011011 00110101 10100111 10101010 01010011. In that case, the value of the check byte will be equal to $C_B = 191 = 10111111$.

Case 1 (BAE corrupting one data byte): Assume that the decoder receives the sequence 1100$\underline{00}$11 00110101 10100111 10101010 01010011 10111111. On the basis of the value of the syndrome, $S = [71 - 191] \pmod{255} = 135 = -5 \times 24 \pmod{255} = -5 \times [2^4 + 2^3] \pmod{255}$. The decoder will conclude that the error has corrupted the first data byte (Table 2). Hence, the corrected value of this byte will be $B_1 = [195 + 24] \pmod{255} = 219 = 11011011$.

Case 2 (BAE corrupting the check byte): Suppose the decoder receives the sequence 11011011 00110101 10100111 10101010 01010011 1011$\underline{00}$11. Based on the value of the syndrome, $S = [191 - 179] \pmod{255} = 12 = 2^2 + 2^3$. The decoder will conclude that the the check-byte is received in error (Table 2). After the error correction, its value will be $C_B = [179 + 12] \pmod{255} = 191 = 10111111$.

Case 3 (BAE corrupting two data bytes): Assume that the decoder receives the sequence 11011011 00110101 1010011$\underline{0\ 0}$0101010 01010011 10111111. On the basis of the value of the syndrome, $S = [42 - 191] \pmod{255} = 106 = [9(-1) + 25(-128)] \pmod{255}$. The decoder will conclude that the error has corrupted the third and fourth data byte (Table 2). Hence, the corrected values of these bytes will be $B_3 = [166 + 1] \pmod{255} = 167 = 10100111$ and $B_4 = [42 + 128] \pmod{255} = 170 = 10101010$.

Case 4 (BAE corrupting the last data byte and check byte): Suppose that the decoder receives the sequence
11011011 00110101 10100111 10101010 0101001$\underline{0\ 0}$0111111. Based on the value of the syndrome, $S = [162 - 63] \pmod{255} = 99 = [29(-1) + 128] \pmod{255}$. The decoder will conclude that the last data byte and check-byte are received in error (Table 2). After the error correction, their values will be $B_5 = [82 + 1] \pmod{255} = 83 = 01010011$ and $C_B = [63 + 128] \pmod{255} = 191 = 10111111$.

Table 2: The ST for the $(48, 40)$ B_2AEC code

Sl.No.	Syndrome ($\zeta_{b,l}$)	Error Loc.	Error (e_1)	Error Loc.	Error (e_2)
1	1	6	1	0	0
2	2	6	2	0	0
3	3	6	3	0	0
4	4	6	4	0	0

5	6	6	6	0	0
6	8	6	8	0	0
7	12	6	12	0	0
8	15	1	48	0	0
9	16	6	16	0	0
10	21	5	96	0	0
11	23	5	8	0	0
12	24	6	24	0	0
13	30	1	96	0	0
14	31	2	32	0	0
15	32	6	32	0	0
16	39	3	24	0	0
17	42	5	192	0	0
18	45	4	192	0	0
19	46	5	16	0	0
20	48	6	48	0	0
21	55	4	8	0	0
22	57	3	192	0	0
23	60	1	192	0	0
24	62	2	64	0	0
25	64	6	64	0	0
26	69	5	24	0	0
27	75	4	48	0	0
28	78	3	48	0	0
29	81	5	6	0	0
30	87	2	24	0	0
31	88	4	1	5	128
32	92	5	32	0	0
33	93	2	96	0	0
34	95	1	32	0	0
35	96	6	96	0	0
36	99	5	1	6	128
37	105	4	6	0	0
38	106	3	1	4	128
39	110	4	16	0	0
40	111	3	16	0	0
41	113	5	128	0	0
42	115	4	128	0	0
43	116	2	1	3	128
44	119	1	1	2	128
45	123	3	120	0	0
46	124	2	128	0	0
47	125	1	128	0	0
48	128	6	128	0	0
49	135	1	24	0	0
50	138	5	48	0	0

51	139	5	4	0	0
52	143	2	16	0	0
53	147	3	12	0	0
54	150	4	96	0	0
55	155	4	4	0	0
56	156	3	96	0	0
57	162	5	12	0	0
58	165	4	24	0	0
59	168	5	3	0	0
60	171	2	12	0	0
61	174	2	48	0	0
62	175	1	16	0	0
63	180	4	3	0	0
64	183	3	8	0	0
65	184	5	64	0	0
66	185	4	64	0	0
67	186	2	192	0	0
68	189	3	64	0	0
69	190	1	64	0	0
70	192	6	192	0	0
71	195	1	12	0	0
72	197	5	2	0	0
73	199	2	8	0	0
74	201	3	6	0	0
75	205	4	2	0	0
76	210	4	12	0	0
77	213	2	6	0	0
78	215	1	8	0	0
79	219	3	4	0	0
80	220	4	32	0	0
81	222	3	32	0	0
82	225	1	6	0	0
83	226	5	1	0	0
84	227	2	4	0	0
85	228	3	3	0	0
86	230	4	1	0	0
87	234	2	3	0	0
88	235	1	4	0	0
89	237	3	2	0	0
90	240	1	3	0	0
91	241	2	2	0	0
92	245	1	2	0	0
93	246	3	1	0	0
94	248	2	1	0	0
95	250	1	1	0	0

3 Evaluation and Comparisons

In this section, we will first analyze the probability of erroneous decoding for various BERs. After that, we will analyze the efficiency of the proposed codes in terms of redundancy.

3.1 The BER and Error Probability

The BER is the ratio of the total number of bits received in error to the total number of bits transmitted. The errors discussed here vary from 1 to l bits. So, for determining the BER, we shall consider the average.

Theorem 3 *The BER for a* $((k+1)b, kb)$ *integer* B_lAEC *code is given by*

$$\frac{1}{l(k+1)b}\Big[\frac{l^2+5l-2}{4}\Big].$$

Proof For a BAE of length $1, 2, 3,$ and 4, BER of a $((k+1)b, kb)$ integer B_lAEC code will be $\frac{1}{(k+1)b}$, $\frac{2}{(k+1)b}$, $\frac{2+3}{2(k+1)b}$, and $\frac{2+3+4}{3(k+1)b}$ respectively. Continuing this, for a burst of length l, BER will be $\frac{2+3+\ldots+l}{(l-1)(k+1)b}$. Thus BER for the proposed code will be

$$\begin{aligned}
&\frac{1}{l(k+1)b}\Big[1+2+\frac{2+3}{2}+\ldots+\frac{2+3+4+\ldots+l}{l-1}\Big]\\
=&\frac{1}{l(k+1)b}\Big[1+\sum_{j=2}^{l}\sum_{i=2}^{j}\frac{i}{j-1}\Big]\\
=&\frac{1}{l(k+1)b}\Big[1+\sum_{j=2}^{l}(\frac{2}{j-1}+\frac{3}{j-1}+\ldots+\frac{j}{j-1})\Big]\\
=&\frac{1}{l(k+1)b}\Big[1+\sum_{j=2}^{l}\frac{1}{j-1}(2+3+\ldots+j)\Big]\\
=&\frac{1}{l(k+1)b}\Big[1+\sum_{j=2}^{l}\frac{2+j}{2}\Big]\\
=&\frac{1}{l(k+1)b}\Big[1+\frac{1}{2}\sum_{j=2}^{l}2+\frac{1}{2}\sum_{j=2}^{l}j\Big]\\
=&\frac{1}{l(k+1)b}\Big[\frac{4l+(l-1)(l+2)}{4}\Big]\\
=&\frac{1}{l(k+1)b}\Big[\frac{l^2+5l-2}{4}\Big].
\end{aligned}$$

□

Theorem 4 *For transition probability ϵ of the occurrence $1 \to 0$, the probability of erroneous decoding $P_d(AB)$ for a $((k+1)b, kb)$ integer B_lAEC code will be* $(k+1)b\epsilon^1(1-\epsilon)^{(k+1)b-1}+\epsilon^2(1-\epsilon)^{(k+1)b-2}\left[\left(\left(\frac{1}{1-\epsilon}\right)^{l-1}-1\right)\left(\frac{1-\epsilon}{\epsilon}\right)\left((k+1)b + \left(\frac{1-\epsilon}{\epsilon}\right)\right) - (l-1)\left(\frac{1-\epsilon}{\epsilon}\right)\left(\frac{1}{1-\epsilon}\right)^{l-1}\right]$.

Proof For $l = 1$, as discussed earlier, 1 bit will be erroneous and the remaining $(k+1)b-1$ bits will be non-erroneous. Thus the probability will be $(k+1)b[\epsilon^1(1-\epsilon)^{(k+1)-1}]$. Similarly for $l = 2$, the probability will be $((k+1)b-1)[\epsilon^2(1-\epsilon)^{(k+1)b-2}]$. For $l=3$, there are two BAEs having non-zero components. These errors may occur at $(k+1)b-2$ different positions. Hence, the probability will be $((k+1)b-2)[\binom{1}{0}\epsilon^2(1-\epsilon)^{(k+1)b-2}+\binom{1}{1}\epsilon^3(1-\epsilon)^{(k+1)b-3}]$. Continuing this, for $(k+1)b-l+1$ number of positions for BAEs of length l, we have the probability equal to $((k+1)b-l+1)[\sum_{i=0}^{l-2}\binom{l-2}{i}\epsilon^{i+2}(1-\epsilon)^{(k+1)b-i-2}]$. Finally by adding up the probabilities up to length l, we get

$$P_d(AB) = (k+1)b\epsilon^1(1-\epsilon)^{(k+1)b-1}+((k+1)b-1)\sum_{i=0}^{0}\binom{0}{i}\epsilon^{i+2}(1-\epsilon)^{(k+1)b-i-2}+((k+1)b-2)\sum_{i=0}^{1}\binom{1}{i}\epsilon^{i+2}(1-\epsilon)^{(k+1)b-i-2}+\ldots+((k+1)b-l+1)\sum_{i=0}^{l-2}\binom{l-2}{i}\epsilon^{i+2}(1-\epsilon)^{(k+1)b-i-2}$$

$$= (k+1)b\epsilon^1(1-\epsilon)^{(k+1)b-1} + \sum_{j=1}^{l-1}\sum_{i=0}^{j-1}((k+1)b-j)\binom{j-1}{i}\epsilon^{i+2}(1-\epsilon)^{(k+1)b-i-2}$$

$$= (k+1)b\epsilon^1(1-\epsilon)^{(k+1)b-1}+\epsilon^2(1-\epsilon)^{(k+1)b-2}\sum_{j=1}^{l-1}\sum_{i=0}^{j-1}((k+1)b-j)\left[\binom{j-1}{i}\left(\frac{\epsilon}{1-\epsilon}\right)^i\right]$$

$$= (k+1)b\epsilon^1(1-\epsilon)^{(k+1)b-1}+\epsilon^2(1-\epsilon)^{(k+1)b-2}\sum_{j=1}^{l-1}((k+1)b-j)\left(\frac{1}{1-\epsilon}\right)^{j-1}$$

$$= (k+1)b\epsilon^1(1-\epsilon)^{(k+1)b-1}+\epsilon^2(1-\epsilon)^{(k+1)b-2}\left[\left(\left(\frac{1}{1-\epsilon}\right)^{l-1}-1\right)\left(\frac{1-\epsilon}{\epsilon}\right)\left((k+1)b+\left(\frac{1-\epsilon}{\epsilon}\right)\right)-(l-1)\left(\frac{1-\epsilon}{\epsilon}\right)\left(\frac{1}{1-\epsilon}\right)^{l-1}\right].$$

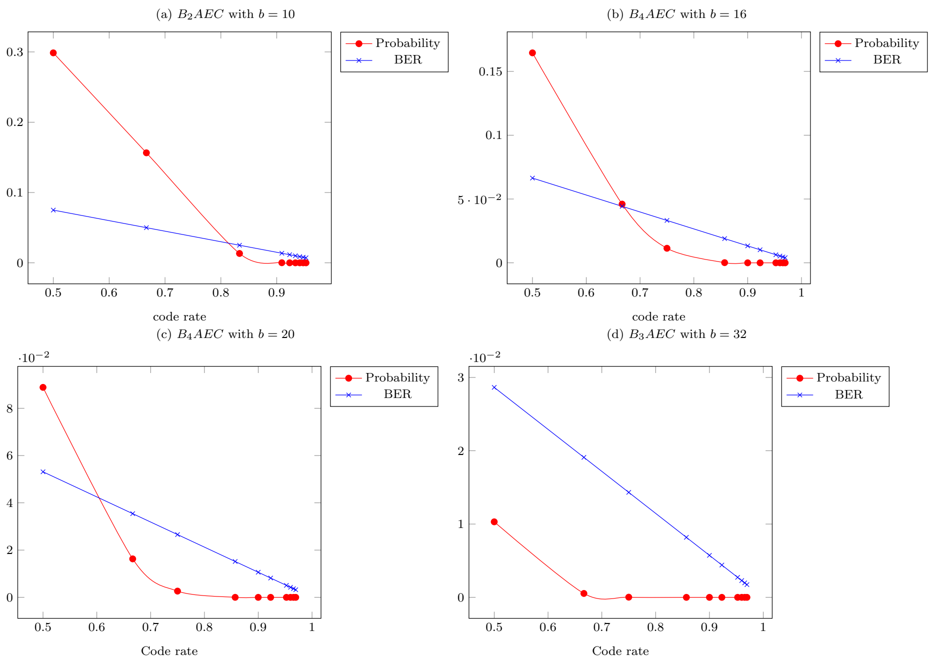
□

A few graphs in Fig. 3 show the changes in the BER and the $P_d(AB)$ for some integer B_lAEC codes. We notice that, in all cases, the BER and the $P_d(AB)$ decrease with increasing code rate. In other words, the higher the value of k, the lower the BER and the $P_d(AB)$. We also observe that as the code rate increases, the $P_d(AB)$ decreases much faster than the BER.

3.2 Comparisons

In addition to having low probability of incorrect decoding, the proposed codes are very efficient in terms of redundancy. This conclusion follows from the

Fig. 3: The dependence of the BER and the $P_d(AB)$ on the code rate of integer B_lAEC codes ($\epsilon = 0.1$)



results of comparing the proposed codes with other BAE correcting codes [10], [12]. Namely, from Table 3 we see that, for the values of $l = 2, 3, 4$, the proposed codes use up to three check bits less than Saitoh-Imai codes [12]. Of course, our codes cannot be more rate-efficient than those from [10], but in turn they correct all BAEs (regardless of their position).

Table 3: Check-bit lengths of various codes correcting BAEs

Data word length (bits)	Proposed codes			Codes from [12]			Codes from [10]		
	$l = 2$	$l = 3$	$l = 4$	$l = 2$	$l = 3$	$l = 4$	$l = 2$	$l = 3$	$l = 4$
K = 128	10	12	13	12	13	14	9	11	12
K = 256	11	13	15	13	14	15	10	11	13
K = 512	12	14	16	15	16	17	11	13	14
K = 1024	13	15	17	16	17	18	12	13	15
K = 2048	14	16	18	17	18	19	13	14	16
K = 4096	15	17	19	18	19	20	14	15	17

Another evidence that confirms the rate-efficiency of the proposed codes is the Campopiano upper bound for (N, K) linear burst error correcting (BEC) codes [3]. This bound is, in the general case, given by $q^{N-K} > q^{2(l-1)}[(q-1)(N-2l+1)+1]$, where q is the field size. By taking $q = 2$, the upper bound reduces to $2^{N-K} > 2^{2(l-1)}[N - 2l + 2]$. Using this inequality, it is easy to show that, in many cases, the proposed codes use less check-bits than the best possible linear BEC codes (Table 4).

Table 4: Number of check-bits for the proposed and the best possible linear BEC codes

Codeword length (bits)	l	Best possible linear BEC codes	Proposed codes
$N = 16$	3	≥ 8	8
$N = 30$	4	> 10	10
$N = 36$	3	> 9	9
$N = 44$	4	> 11	11
$N = 143$	4	> 13	13
$N = 204$	3	≥ 12	12
$N = 528$	4	≥ 16	16

Besides this, it should be noted that the STs for the proposed codes occupy less memory than those used by linear BEC codes. In particular, in Section 2 we have seen that one ST entry is $3b + 2\lceil \log_2(k+1) \rceil$ bits wide (Fig. 2). On the other hand, one ST entry for linear BEC codes has $2b + kb$ bits. From this it is easy to prove that the inequality $2\lceil \log_2(k+1) \rceil \leq k(b-1)$ holds for all $b \geq 3$ (note that both codes exist only if $l < \frac{b}{2}$).

4 Conclusion

In this paper, we have presented a class of integer codes capable of correcting asymmetric burst errors. We have shown that the presented codes are very efficient in terms of redundancy. More precisely, it has been shown that they are more rate-efficient not only than their linear counterparts, but also than the optimal burst error correcting codes. In addition, the proposed codes use operations that are supported by all processors, which makes them attractive for use in systems that display asymmetric errors. The best-known examples of such systems are optical networks and VLSI memories.

5 Acknowledgement

All authors are thankful to Abhisek Upadhyaya (IIT Kharagpur) for his valuable suggestions to find the coefficients for different values of b and l. The first author is thankful to UGC (India) for granting CSIR-UGC Research Fellowship (Ref. No: 1112/(CSIR-UGC NET JUNE 2017)) to carry out this research work. The third author is thankful to the Ministry of Education, Science and Technological Development of the Republic of Serbia for financial support (Grant No. 451-03-68/2022-14/200175).

References

1. Blake, I. F.: Codes over certain rings. Information and Control, **20**(4), 396-404 (1972)
2. Blake, I. F.: Codes over integer residue rings. Information and Control, **29**(4), 295-300 (1975)
3. Campopiano, C.: Bounds on burst-error-correcting codes (corresp.). IRE Transactions on Information Theory, **8**(3), 257-259 (1962)
4. Levenshtein, V. I., Vinck, A. H.: Perfect (d, k)-codes capable of correcting single peak-shifts. IEEE transactions on information theory, **39**(2), 656-662 (1993)
5. Oprisan, P., Bose, B.: ARQ in optical networks. Proceedings of IEEE International Symposium Pacific Rim Dependable Computing, 251-257 (2001)
6. Pokhrel, N. K., Das, P. K.: Unidirectional solid burst correcting integer codes. Journal of Applied Mathematics and Computing, https://doi.org/10.1007/s12190-021-01662-2 ,1-16 (2021)
7. Pokhrel, N. K., Das, P. K.: Low-density and high-density asymmetric CT-burst correcting integer codes. Advances in Mathematics of Communications, doi: 10.3934/amc.2022030, 1-15 (2022)
8. Radonjic, A., Vujicic, V.: Integer codes correcting burst and random asymmetric errors within a byte. Journal of the Franklin Institute, **355**(2), 981-996 (2018)
9. Radonjic, A., Vujicic, V.: Integer codes correcting burst asymmetric errors within a byte and double asymmetric errors. Cryptography and Communications, **12**(2), 221-230 (2020)
10. Radonjic, A., Vujicic, V.: Integer codes correcting burst asymmetric errors within a byte. IETE Journal of Research, **68**(1), 176-182 (2022)
11. Rao, T. R. N., Chawla, A. S.: Asymmetric error codes for some LSI semiconductor memories. Proceedings of the Annual Southeastern Symposium on System Theory, 170-171 (1975)
12. Saitoh, Y., Imai, H.: Some classes of burst asymmetric or unidirectional error correcting codes. Electronics Letters, **26**(5), 286-287 (1990)
13. Varshamov, R. R., Tenengolz, G. M.: One asymmetrical error correcting codes. Avtomatika i Telemehanika, **26**(2), 288-392 (1965)
14. Vinck, A.J.H., Morita, H.: Codes over the ring of integer modulo m. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, **81-A**(10), 2013-2018 (1998)