Chapter 6 Asymmetric Burst Correcting Integer Codes

The contents of this chapter are based on the paper mentioned below:

- Pokhrel, N. K., Das, P. K. and Radonjic, A. Integer codes capable of correcting burst asymmetric errors. *Journal of Applied Mathematics and Computing*, 1-14, 2022, doi:10.1007/s12190-022-01770-7.

Chapter 6

Asymmetric Burst Correcting Integer Codes

6.1 Overview

The codes discussed in this chapter are similar to the ones discussed in Chapter 5. In fact, these are a generalisation of Chapter 5. The occurrence of bursts and asymmetric patterns has already been discussed in Chapter 1. We present a class of integer codes capable of correcting asymmetric burst errors of length up to l. We name these codes as integer $(B_lAEC)_b$ codes. 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 probability of erroneous decoding for different Bit Error Rates (BERs) is calculated in Section 6.3. Based on existing integer codes, the codes presented here are a generalisation of the integer codes discussed in Definition 1.21. The presented codes are also analysed from a rate-efficiency point of view in Section 6.3. The obtained results show that for many data lengths they require less check-bits than optimal burst error-correcting codes.

6.2 Construction of codes

For the construction of the codes, we continue with Definition 1.21 and introduce the syndrome set for the other case as done in Chapter 5.

6.2.1 Encoding procedure

The encoding technique is identical to Chapter 5, where the set of syndromes is separated into two categories based on the location of occurrence.

Definition 6.1. *The syndrome set of all asymmetric bursts of length up to l corrupting b-bit byte is defined by*

$$S_1 = \bigcup_{i=1}^{k+1}\Big[-C_i.\epsilon_{b,l}\Big] \pmod{2^b - 1}, \tag{6.1}$$

where $\epsilon_{b,l} = e_{b,1} \cup e_{b,2} \cup \cdots \cup e_{b,l}$ with $e_{b,t} = 2^r(1, 3, \ldots, 2^t - 1)$, $0 \leq r \leq b - t$.

Definition 6.2. *The syndrome set of all asymmetric bursts up to length l corrupting two adjoining b-bit bytes (including check byte) is defined by*

$$S_2 = \bigcup_{i=1}^{k}\Big[C_i P_r + C_{i+1} Q_s\Big] \pmod{2^b - 1}, \tag{6.2}$$

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 $max\{r + s\} = l$.

Note: While choosing the coefficient C_i's in (6.1) and (6.2), we have to ensure that the collection $C_1, C_2, \ldots, C_k, C_{k+1}$ is such that the sets $-C_1\epsilon_{b,l}$, $-C_2\epsilon_{b,l}$, $\ldots$, $-C_k\epsilon_{b,l}$, $\epsilon_{b,l}$, and $C_iP_r + C_{i+1}Q_s$ for $1 \leq i \leq k$ are mutually disjoint where $C_{k+1} = -1$. The coefficients can be obtained by using a suitable computer search result (python code attached in Appendix F).

The expressions derived so far give us the idea of theoretical construction of the $((k + 1)b, kb)$ integer $(B_lAEC)_b$ codes. Using these construction methods, theorems below give us the number of non-zero syndromes to be used in the construction of the look-up table LUT_2 for decoding purpose.

Theorem 6.3. *A $((k+1)b, kb)$ integer $(B_lAEC)_b$ code can correct asymmetric bursts up to length l within a byte or occurring between two adjoining bytes 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 Result 1.22.
For asymmetric burst of length 2 occurring between two adjoining b-bit bytes, the syndrome element will be of the type $C_iP_1 + C_{i+1}Q_1 = C_i(-1) + C_{i+1}(-2^{b-1})$, which has only one possibility leading to the order $\alpha_2 = 1$. For asymmetric burst of length 3 occurring between two adjoining b-bit bytes, the syndrome elements are of the type $C_iP_1 + C_{i+1}Q_2$ and $C_iP_2 + C_{i+1}Q_1$. Possibilities in this case are $C_i(-1) + C_{i+1}(q_{b-1}2^{b-1} - 2^{b-2})$ and $C_i(-2^1 + p_0) + C_{i+1}(-2^{b-1})$ leading to the order $\alpha_3 = 2 \times 2 = 4$. Again, for the case of length 4, $C_iP_1 + C_{i+1}Q_3, C_iP_2 + C_{i+1}Q_2$ and $C_iP_3 + C_{i+1}Q_1$ are the possibilities for the syndrome which makes the syndrome element pattern as $C_i(-1) + C_{i+1}(q_{b-1}2^{b-1} + q_{b-2}2^{b-2} - 2^{b-3})$, $C_i(-2 + p_0) + C_{i+1}(q_{b-1}2^{b-1} - 2^{b-2})$ and $C_i(-2^2 + p_12^1 + p_0) + C_{i+1}(-2^{b-2})$ and the order $\alpha_4 = 3 \times 2^2 = 12$. Following this pattern, we observe that the possibilities for syndrome elements corresponding to an asymmetric burst of length l occurring between two b-bit bytes is $C_iP_1 + C_{i+1}Q_{l-1}$, $C_iP_2 + C_{i+1}Q_{l-2}, \ldots, C_iP_{l-1} + C_{i+1}Q_1$. 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 the asymmetric bursts corrupting one b-bit byte are different from those corrupting two adjacent b-bit bytes. Hence, the codes satisfying Condition $1 - 3$ are $((k+1)b, kb)$ integer $(B_lAEC)_b$ codes. □

Theorem 6.4. *Let* $\zeta_{b,l} = S_1 \cup S_2$ *be the set of syndromes for a* $((k+1)b, kb)$ *integer* $(B_lAEC)_b$ *code, then* $\mid \zeta_{b,l} \mid = (k+1)[2^{l-1}(b-l+2)-1] + k\sum_{i=2}^{l}\alpha_i$, *where* $\alpha_i = (i-1)2^{i-2}$.

Proof. From Theorem 6.3, it is clear that $\mid \zeta_{b,l} \mid = (k+1)[2^{l-1}(b-l+2)-1] + k\sum_{i=2}^{l}\alpha_i$. □

Table 6.1: **Possible coefficients for a few $((k+1)b, kb)$ integer $(B_lAEC)_b$ codes**

b	l	**Coefficients**
8	2	5, 7, 9, 25, 29
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, 33, 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

Contd...

b	***l***	**Coefficients**
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

6.2.2 Decoding procedure

The decoding procedure is similar to Chapter 5 subject to the following changes in the steps of decoding.

- For asymmetric burst of length up to l occurring within the check byte:

$$C_B = [\bar{C}_B + e_1] \pmod{2^b - 1};$$

  where syndrome $S =$ error $e_1 \in \epsilon_{b,l}$.

- For asymmetric burst of length up to l occurring within i^{th} data byte ($1 \leq i \leq k$):

$$B_i = [\bar{B}_i + e_1] \pmod{2^b - 1};$$

  where syndrome $S = -C_i \times e_1 \pmod{2^b - 1}$, with error $e_1 \in \epsilon_{b,l}$.

- For asymmetric burst of length up to l occurring between i^{th} and $(i+1)^{th}$ data byte ($1 \leq i \leq k-1$):

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

$$B_{i+1} = [\bar{B}_{i+1} + e_2] \pmod{2^b - 1},\ e_2 \in \epsilon_{b,l},\ -e_2 \in Q_s;$$

where syndrome $S = [C_i(-e_1) + C_{i+1}(-e_2)] \pmod{2^b - 1}$.

- For asymmetric burst of length up to l occurring between the last data byte (k^{th} byte) and the check byte:

$$B_k = [\bar{B}_k + e_1] \pmod{2^b - 1},\ e_1 \in \epsilon_{b,l},\ -e_1 \in P_r;$$

$$C_B = [\bar{C}_B + e_2] \pmod{2^b - 1},\ e_2 \in \epsilon_{b,l},\ -e_2 \in Q_s;$$

where syndrome $S = [C_k(-e_1) + e_2] \pmod{2^b - 1}$.

The width representation of a syndrome table entry pertaining to the two types of bursts discussed can be found in Figure 5.1-5.2. We now illustrate the encoding and decoding methods with the help of an example for $b = 8$ and $l = 2$.

Table 6.2: $\boldsymbol{LUT_2}$ **for** $\boldsymbol{(48, 40)}$ **integer** $\boldsymbol{(B_2AEC)_8}$ **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

Contd...

Sl. No.	**Syndrome ($\zeta_{b,l}$)**	**Error Loc.**	**Error (e_1)**	**Error Loc.**	**Error (e_2)**
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

Contd...

Sl. No.	**Syndrome ($\zeta_{b,l}$)**	**Error Loc.**	**Error (e_1)**	**Error Loc.**	**Error (e_2)**
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

Contd...

Sl. No.	**Syndrome ($\zeta_{b,l}$)**	**Error Loc.**	**Error (e_1)**	**Error Loc.**	**Error (e_2)**
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

Example 6.5. *Let $b = 8$, $l = 2$ with $C_1 = 5, C_2 = 7, C_3 = 9, C_4 = 25, C_5 = 29$ and $C_6 = -1$ with $|\ \zeta_{8,2}\ |= 95$ syndromes listed in Table 6.2. Suppose we want to transmit* 5 8*-bit data bytes* 11011011 00110101 10100111 10101010 01010011*, then*

check byte will be $C_B = 191 = 10111111$*.*

Case I *(Asymmetric burst within a data byte): Suppose the decoder receives* $110\underline{00}011$ 00110101 10100111 10101010 01010011 10111111*, then syndrome* $S = [71 - 191] \pmod{255} = 135 = -5 \times 24 \pmod{255} = -5 \times [2^4 + 2^3] \pmod{255}$*. Thus the error has occurred in* B_1 *at* 4th *and* 5th *positions. So the corrected data byte* $B_1 = [195 + 24] \pmod{255} = 219 = 11011011$*.*

Case II *(Asymmetric burst within the check byte): Suppose the decoder receives* 11011011 00110101 10100111 10101010 01010011 $1011\underline{00}11$*, then syndrome* $S = [191 - 179] \pmod{255} = 12 = 2^2 + 2^3$*. Thus the error has occurred in the check byte at* 5th *and* 6th *positions. So the corrected check byte will be* $C_B = [179 + 12] \pmod{255} = 191 = 10111111$*.*

Case III *(Asymmetric burst between two data bytes): Suppose the decoder receives* 11011011 00110101 $1010011\underline{0\ 0}0101010$ 01010011 10111111*, then syndrome* $S = [42 - 191] \pmod{255} = 106 = [9(-1) + 25(-128)] \pmod{255}$*. Thus the error has occurred between* 3rd(B_3) *and* 4th(B_4) *data bytes, where error* $1 = 2^0$ *in* 8th *component of* B_3 *and error* $128 = 2^7$ *in* 1st *component of* B_4*. So the corrected data bytes will be* $B_3 = [166 + 1] \pmod{255} = 167 = 10100111$ *and* $B_4 = [42 + 128] \pmod{255} = 170 = 10101010$*.*

Case IV *(Asymmetric burst occurring between last data byte and the check byte): Suppose the decoder receives*
11011011 00110101 10100111 10101010 $0101001\underline{0\ 0}0111111$*, then syndrome* $S = [162 - 63] \pmod{255} = 99 = [29(-1) + 128] \pmod{255}$*. Thus the error has occurred between* B_5 *and* C_B*, where error* $1 = 2^0$ *in* 8th *component of* B_5 *and error* $128 = 2^7$ *in* 1st *component of* C_B*. So the corrected* 8*-bit bytes will be* $B_5 = [82 + 1] \pmod{255} = 83 = 01010011$ *and* $C_B = [63 + 128] \pmod{255} = 191 = 10111111$*.*

6.3 Comparison and probability

In this section, we discuss the comparison and the probability of erroneous decoding.

Table 6.3: **Comparison of redundancy (ρ) with linear codes**

Code length	b	l	**Redundancy for linear codes**	**Redundancy for proposed codes**
16	8	3	$\rho \geq 8$	$\rho = 8$
30	10	4	$\rho > 10$	$\rho = 10$
36	9	3	$\rho > 9$	$\rho = 9$
44	11	4	$\rho > 11$	$\rho = 11$
143	13	4	$\rho > 13$	$\rho = 13$
204	12	3	$\rho \geq 12$	$\rho = 12$
528	16	4	$\rho \geq 16$	$\rho = 16$

6.3.1 Comparison

In coding theory, codes are compared based on the code rates with the same number of information bits. Codes with lesser check bits are considered to be more efficient since the encoding can be carried out by maintaining fewer bit consumption. For this comparison, we try to bring the parameters on the same line. A codeword in [96] is constructed using m b-bit data/information bytes $X_1, X_2, \ldots, X_m$ and $b + l$ check symbols in the form of a b-bit byte X_m and another l-bit byte y. Thus the code rate R_1 for the presented codes here will be $R_1 = \frac{mb}{mb+b} = \frac{1}{1+\frac{1}{m}}$ and code rate for the codes in [96] with $b = l$ will be $R_2 = \frac{mb}{mb+2b} = \frac{1}{1+\frac{2}{m}} < R_1$. This clearly imply the efficiency of these codes turning out to be better than that of [96] in terms of its different applications.

Campopiano upper bound [24] can be considered as the least possible redundancy for linear codes capable of correcting a single burst. This bound for (n, k) linear codes defined over the field $GF(q)$ is given by $q^{n-k} > q^{2(l-1)}[(q-1)(n-2l+1)+1]$, by considering $q = 2$, the upper bound further gets reduced to $2^{n-k} > 2^{2(l-1)}[n-2l+2]$. Now, by comparing the proposed codes on the lines of a linear code having the same code length, upon the existence of the codes, we observe in many cases the proposed codes have less or equal redundancy, leading to a better code rate as highlighted

Table 6.4: **Check-bit lengths of various codes correcting asymmetric bursts**

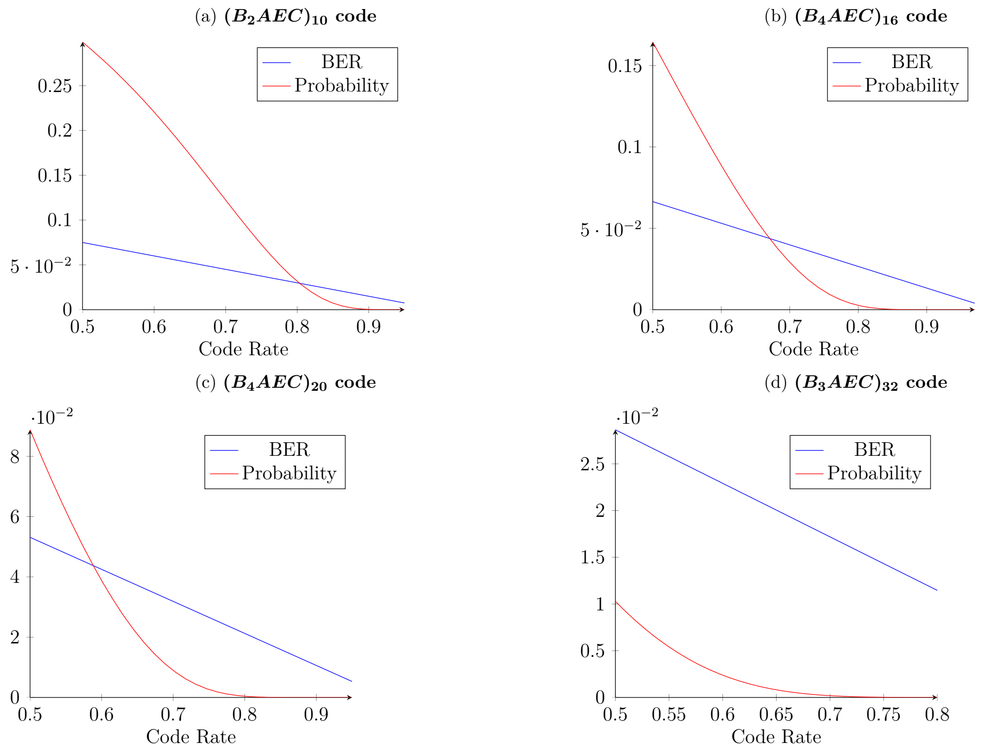
Data word length(bits)	**Proposed codes**			**Codes from** [97]			**Codes from Definition** 1.21		
	$\boldsymbol{l} = 2$	$\boldsymbol{l} = 3$	$\boldsymbol{l} = 4$	$\boldsymbol{l} = 2$	$\boldsymbol{l} = 3$	$\boldsymbol{l} = 4$	$\boldsymbol{l} = 2$	$\boldsymbol{l} = 3$	$\boldsymbol{l} = 4$
K = 128	10	12	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

in Table 6.3. By checking the existence of linear codes using Campopiano bound and existence of the proposed codes using the computer search result for same code length l, we can show that the proposed codes require less number of bits in the syndrome table to perform the error correcting process. For this, consider $q = 2$ and divide code length $(k+1)b$ of a linear code capable of correcting bursts up to length l into $k+1$ b-bit bytes. Consider X to be the number of asymmetric bursts occurring within the divided b-bit blocks and Y to be the number of asymmetric bursts occurring between two adjoining b-bit blocks.

From Table 6.4, 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 [97]. Of course, the proposed codes cannot be more rate-efficient than those discussed in Definition 1.21, but in turn they correct all asymmetric bursts (regardless of their position). Since linear codes correcting bursts also use syndrome table, so we now compare the bits consumed by the syndrome table for both linear and integer codes.

- In case of the proposed integer code, each syndrome entry corresponding to an asymmetric burst occurring within a b-bit byte has $2b + \lceil log_2(k+1) \rceil$ bits, whereas in case of a linear code, this number is $kb + 2b$, clearly $X \times (2b + \lceil log_2(k+1) \rceil) < X \times (kb + 2b)$.

Figure 6.1: **The dependence of the BER and the $P_d(AB)$ on the code rate of integer $(B_l AEC)_b$ codes ($\epsilon = 0.1$)**



- Similarly, one syndrome entry from integer code corresponding to asymmetric burst occurring between two adjoining b-bit bytes has $b+2(l-1)+\lceil log_2(k) \rceil$ bits, whereas the syndrome entry for linear codes in this case has $kb+2b$ bits. Since $l < \frac{b}{2}$ for existence of the code in both cases, so $Y \times (b+2(l-1)+\lceil log_2(k) \rceil) < Y \times (kb+2b)$.

Hence, we are able to highlight a few positive points of the proposed codes over the traditional linear codes.

6.3.2 Probability of erroneous decoding and BER

For BER, we get the following result, which is similar to what was discussed in the previous chapters.

Theorem 6.6. *Bit Error Rate for a $((k+1)b, kb)$ integer $(B_l AEC)_b$ code is given* *by*

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

Proof. For a burst of length $1, 2, 3,$ and 4, BER of a $((k+1)b, kb)$ integer $(B_lAEC)_b$ 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

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

□

In our study, since the codes are considered over a Z-channel (refer Figure 1.1), we obtain the probability of erroneous decoding as follows:

Theorem 6.7. *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 $(B_lAEC)_b$ code will be* $(k+1)b\epsilon^1(1-\epsilon)^{(k+1)b-1}+\epsilon^2(1-\epsilon)^{(k+1)b-2}\Bigg[\left(\left(\frac{1}{1-\epsilon}\right)^{l-1}-1\right)\left(\frac{1-\epsilon}{\epsilon}\right)\left((k+1)b+\left(\frac{1-\epsilon}{\epsilon}\right)\right)$ $-(l-1)\left(\frac{1-\epsilon}{\epsilon}\right)\left(\frac{1}{1-\epsilon}\right)^{l-1}\Bigg].$

Proof. The proposed code deals with asymmetric bursts up to length l where the bursts can occur anywhere throughout the $(k+1)$ b-bit bytes, so we shall proceed by determining the probabilities for each length $1, 2, \ldots, l$. For an asymmetric burst of length 1, as discussed earlier, 1 bit will be erroneous and the remaining $(k+1)b-1$ bits will be non-erroneous. Thus the probability in this case will be $\epsilon^1(1-\epsilon)^{(k+1)b-1}$, as there are $(k+1)b$ number positions for such bursts, therefore the probability will be $(k+1)b[\epsilon^1(1-\epsilon)^{(k+1)-1}]$. Similarly for asymmetric bursts of length 2, the probability will be $((k+1)b-1)\left[\epsilon^2(1-\epsilon)^{(k+1)b-2}\right]$, for length 3, there are asymmetric bursts having non-zero components at 2 or 3 places, as there are $(k+1)b-2$ number of positions for these bursts to occur, so the probability will be $((k+1)b-2)\left[\binom{1}{0}\epsilon^2(1-\epsilon)^{(k+1)b-2}+\binom{1}{1}\epsilon^3(1-\epsilon)^{(k+1)b-3}\right]$. Continuing this, for $(k+1)b-l+1$ number of positions for asymmetric bursts of length l, we have the probability equal to $((k+1)b-l+1)\left[\sum_{i=0}^{l-2}\binom{l-2}{i}\epsilon^{i+2}(1-\epsilon)^{(k+1)b-i-2}\right]$. Finally by adding up the probabilities up to length l, we get

$$
\begin{aligned}
P_d(AB) = {} & (k+1)b\epsilon^1(1-\epsilon)^{(k+1)b-1} + ((k+1)b-1)\sum_{i=0}^{0}\binom{0}{i}\epsilon^{i+2}(1-\epsilon)^{(k+1)b-i-2} \\
& + ((k+1)b-2)\sum_{i=0}^{1}\binom{1}{i}\epsilon^{i+2}(1-\epsilon)^{(k+1)b-i-2} + \dots \\
& \dots + ((k+1)b-l+1)\sum_{i=0}^{l-2}\binom{l-2}{i}\epsilon^{i+2}(1-\epsilon)^{(k+1)b-i-2} \\
= {} & (k+1)b\epsilon^1(1-\epsilon)^{(k+1)b-1} + \sum_{j=1}^{l-1}\sum_{i=0}^{j-1}((k+1)b-j)\binom{j-1}{i}\epsilon^{i+2}(1-\epsilon)^{(k+1)b-i-2} \\
= {} & (k+1)b\epsilon^1(1-\epsilon)^{(k+1)b-1} \\
& + \epsilon^2(1-\epsilon)^{(k+1)b-2}\sum_{j=1}^{l-1}\sum_{i=0}^{j-1}((k+1)b-j)\left[\binom{j-1}{i}\left(\frac{\epsilon}{1-\epsilon}\right)^i\right] \\
= {} & (k+1)b\epsilon^1(1-\epsilon)^{(k+1)b-1} \\
& + \epsilon^2(1-\epsilon)^{(k+1)b-2}\sum_{j=1}^{l-1}((k+1)b-j)\left[\binom{j-1}{0}\left(\frac{\epsilon}{1-\epsilon}\right)^0 + \binom{j-1}{1}\left(\frac{\epsilon}{1-\epsilon}\right)^1 + \dots\right. \\
& \left.\dots + \binom{j-1}{j-1}\left(\frac{\epsilon}{1-\epsilon}\right)^{j-1}\right] \\
= {} & (k+1)b\epsilon^1(1-\epsilon)^{(k+1)b-1} + \epsilon^2(1-\epsilon)^{(k+1)b-2}\sum_{j=1}^{l-1}((k+1)b-j)\left(1+\frac{\epsilon}{1-\epsilon}\right)^{j-1} \\
= {} & (k+1)b\epsilon^1(1-\epsilon)^{(k+1)b-1} + \epsilon^2(1-\epsilon)^{(k+1)b-2}\sum_{j=1}^{l-1}((k+1)b-j)\left(\frac{1}{1-\epsilon}\right)^{j-1} \\
= {} & (k+1)b\epsilon^1(1-\epsilon)^{(k+1)b-1}
\end{aligned}
$$

$$
\begin{aligned}
&+\epsilon^2(1-\epsilon)^{(k+1)b-2}\left[(k+1)b\sum_{j=1}^{l-1}\left(\frac{1}{1-\epsilon}\right)^{j-1}-\sum_{j=1}^{l-1}j\left(\frac{1}{1-\epsilon}\right)^{j-1}\right]\\
=&(k+1)b\epsilon^1(1-\epsilon)^{(k+1)b-1}\\
&+\epsilon^2(1-\epsilon)^{(k+1)b-2}\left[(k+1)b\frac{\left(\frac{1}{1-\epsilon}\right)^{l-1}-1}{\left(\frac{1}{1-\epsilon}\right)-1}-\left\{1\left(\frac{1}{1-\epsilon}\right)^0+2\left(\frac{1}{1-\epsilon}\right)^1\right.\right.\\
&\left.\left.+3\left(\frac{1}{1-\epsilon}\right)^2+\ldots+(l-1)\left(\frac{1}{1-\epsilon}\right)^{l-2}\right\}\right] \qquad \left(\because\left|\frac{1}{1-\epsilon}\right|>1\right)\\
=&(k+1)b\epsilon^1(1-\epsilon)^{(k+1)b-1}\\
&+\epsilon^2(1-\epsilon)^{(k+1)b-2}\left[\frac{(k+1)b(1-\epsilon)}{\epsilon}\left(\left(\frac{1}{1-\epsilon}\right)^{l-1}-1\right)\right.\\
&\left.+\left(\frac{1-\epsilon}{\epsilon}\right)^2\left(\left(\frac{1}{1-\epsilon}\right)^{l-1}-1\right)-(l-1)\left(\frac{1-\epsilon}{\epsilon}\right)\left(\frac{1}{1-\epsilon}\right)^{l-1}\right]\\
=&(k+1)b\epsilon^1(1-\epsilon)^{(k+1)b-1}\\
&+\epsilon^2(1-\epsilon)^{(k+1)b-2}\left[\left(\left(\frac{1}{1-\epsilon}\right)^{l-1}-1\right)\left(\frac{1-\epsilon}{\epsilon}\right)\left((k+1)b+\left(\frac{1-\epsilon}{\epsilon}\right)\right)\right.\\
&\left.-(l-1)\left(\frac{1-\epsilon}{\epsilon}\right)\left(\frac{1}{1-\epsilon}\right)^{l-1}\right].
\end{aligned}
$$

□

A few graphs in Figure 6.1 show the changes in the BER and the $P_d(AB)$ for some integer $(B_lAEC)_b$ 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.

6.4 Conclusion

In this chapter, 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.