Chapter 4
Unidirectional Solid Burst Correcting Integer Codes

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

- Pokhrel, N. K. and Das, P. K. Unidirectional solid burst correcting integer codes. *Journal of Applied Mathematics and Computing*, 1-16, 2021, doi:10.1007/s12190-021-01662-2.

- Pokhrel, N.K. and Das, P.K. Probability of erroneous decoding for integer codes correcting burst asymmetric/unidirectional/symmetric errors within a byte and up to double asymmetric errors between two bytes. *Kuwait Journal of Science*, 2022, doi: 10.48129/kjs.online.

Chapter 4

Unidirectional Solid Burst Correcting Integer Codes

4.1 Overview

In LSI technology with memories, byte-per-card memory organisation is used rather than bit-per-card memory organization, where a byte is a cluster of b-adjacent bits. So to overcome the noise-affected errors in these memories, a proper error-correcting correcting scheme becomes important. The chances of the occurrences of unidirectional errors in these situations have been discussed in Section 1.3. These errors may lead to system failure [74]. Similarly, the occurrence of solid bursts is also seen in many storage systems, as discussed in Section 1.3 of Chapter 1.

In this chapter, we shall use Definition 1.9 for unidirectional solid bursts for constructing the integer codes capable of correcting unidirectional solid bursts. These codes will be called as integer $(U_lSB)_b$ codes. Figure 1.2 depicts the binary symmetric channel where these errors may occur with crossover probability ϵ. A brief implementation on a quad-core processor is discussed, along with its memory consumption in Section 4.3. Further, a comparison is made with linear codes and integer codes correcting similar types of errors. The codes discussed here use integers and table look ups which makes them suitable for software implementation in all processors. The construction process here identifies the repeating errors (integer values) in symmetric channels and accordingly develops the error-correcting procedure. Finally, the probability of erroneous decoding is determined.

4.2 Construction of codes

Suppose $B = (x_o, x_1, \ldots, x_{b-1})$ and $\bar{B} = (\bar{x}_0, \bar{x}_1, \ldots, \bar{x}_{b-1})$ be the sent and received messages respectively, then the set of unidirectional solid bursts of length t pertaining to the nature $1 \to 0$ occurring within a b-bit byte will be $e_{b,t} = 2^r(2^t - 1)$, where $0 \leq r \leq b-t$, here error is considered as sent-received. Similarly for the nature $0 \to 1$, set of solid bursts of length t within a b-bit byte will be $e'_{b,t} = -2^r(2^t - 1)$, where $0 \leq r \leq b-t$. By considering $E_{b,l} = e_{b,1} \cup e_{b,2} \cup \ldots \cup e_{b,l}$ and $E'_{b,l} = e'_{b,1} \cup e'_{b,2} \cup \ldots \cup e'_{b,l}$, the set of all unidirectional solid bursts up to length l will be $\epsilon_{b,l} = E_{b,l} \cup E'_{b,l}$.

Unlike the error patterns discussed earlier over the binary asymmetric channels, the error patterns here have repetitions after attaining a certain length. Therefore, to decode a sent message uniquely subject to the occurrence of a unidirectional solid burst, the lemma below will be helpful to identify the common error patterns occurring in the set.

Lemma 4.1. *Elements within the error sets $E_{b,l}$ and $E'_{b,l}$ are unique. Further in the error set $\epsilon_{b,l}$, for all $1 \leq t \leq b-1$, $2^0(2^t - 1) = -2^t(2^{b-t} - 1)$ and $2^{b-t}(2^t - 1) = -2^0(2^{b-t} - 1)$.*

Proof. Since the ring $\mathbb{Z}_{2^b-1}$ has no elements of order 2, so by proving elements in $E_{b,l}$ distinct clearly will imply the same for the set $E'_{b,l}$ and vice-versa. As discussed $E_{b,l} = e_{b,1} \cup e_{b,2} \cup \ldots \cup e_{b,l}$ and the highest possible element in the sets $e_{b,t}$: $1 \leq t \leq l < b$ can be written as $2^{b-t}(2^t - 1) = 2^{b-t}(2^{t-1} + 2^{t-2} + \ldots + 2^1 + 2^0) = 2^{b-1} + 2^{b-2} + \ldots + 2^{b-t} < 2^b - 1$. Again by using the fact that each element in $\mathbb{Z}_{2^b-1}$ has a unique binary representation, we can conclude the uniqueness among the elements within the sets $E_{b,l}$ and $E'_{b,l}$.

But we have some pairs of elements between the sets $E_{b,l}$ and $E'_{b,l}$ having same integer values in the ring $\mathbb{Z}_{2^b-1}$. The equality $2^{r_1}(2^{l_1} - 1) = -2^{r_2}(2^{l_2} - 1)$ for $1 \leq l_1, l_2 < b$ and $0 \leq r_i \leq b - l_i$ can be further expressed as $2^{r_1}(2^0 + 2^1 + \ldots + 2^{l_1} - 1) + 2^{r_2}(2^0 + 2^1 + \ldots + 2^{l_2} - 1) = 0 \implies 2^{r_1} + 2^{r_1+1} + \ldots + 2^{r_1+l_1-1} + 2^{r_2} + 2^{r_2+1} +$

$\ldots + 2^{r_2+l_2-1} = 0$, this is possible only if

(1) **Case I:** $r_1 = 0, r_1 + 1 = 1, \ldots, r_1 + l_1 - 1 = t - 1$ and $r_2 = t, r_2 + 1 = t+1, \ldots, r_2 + l_2 - 1 = b - 1 \implies r_1 = 0, l_1 - 1 = t - 1$ and $r_2 = t, l_2 - 1 = b - 1$, thus $2^0(2^t - 1) = -2^t(2^{b-t} - 1)$ for all $1 \le t \le b - 1$.

(2) **Case II:** $r_2 = 0, r_2 + 1 = 1, \ldots, r_2 + l_2 - 1 = t - 1$ and $r_1 = t, r_1 + 1 = t+1, \ldots, r_1 + l_1 - 1 = b - 1$, thus $2^{b-t}(2^t - 1) = -2^0(2^{b-t} - 1)$ for all $1 \le t \le b-1$.

□

Note: As per the discussion in the lemma above, for $1 \le l < \lceil \frac{b}{2} \rceil$ the sets $E_{b,l}$ and $E'_{b,l}$ will not have any common element. Also, for $\lceil \frac{b}{2} \rceil \le l < b$, the common elements occur between the pairs $e_{b,i}$ and $e'_{b,b-i}$.

4.2.1 Encoding procedure

The encoding procedure, width representation of an encoded codeword, and process for choosing the coefficients are similar to the processes discussed in Section 2.2 of Chapter 2. The definitions below gives us the syndrome sets used in the error-correcting procedure.

Definition 4.2. *The set of syndromes for integer codes correcting unidirectional solid bursts is*

$$S_1 = \bigcup_{i=1}^{k+1} -[C_i \epsilon_{b,l}] \pmod{2^b - 1}, \tag{4.1}$$

where the coefficient $C_{k+1} = -1$ *and other coefficients* $C_1, C_2, \ldots, C_k$ *are chosen from* $\mathbb{Z}_{2^b-1} \setminus \{0, 1\}$ *such that the sets* $-C_1\epsilon_{b,l}, -C_2\epsilon_{b,l}, \ldots, -C_k\epsilon_{b,l}$ *and* $\epsilon_{b,l}$ *are all mutually disjoint.*

The Python programme used to find the coefficients is given in Appendix D. Theorem below gives us the number of elements in the syndrome set used for decoding a received message.

Theorem 4.3. *A* $((k+1)b, kb)$ *integer* $(U_lSB)_b$ *code can correct unidirectional solid bursts up to length* l *within a b-bit byte for*

1. $1 \leq l < \lceil \frac{b}{2} \rceil$ *if there exist* k *distinct coefficients in* $\mathbb{Z}_{2^b-1} \setminus \{0,1\}$ *such that* $|S_1| = (k+1)(2b-l+1)l$.

2. $\lceil \frac{b}{2} \rceil \leq l < b$ *if there exist* k *distinct coefficients in* $\mathbb{Z}_{2^b-1} \setminus \{0,1\}$ *such that* $|S_1| = (k+1)[(2b-l-3)l+2b-2]$ *for even* l *and* $|S_1| = (k+1)[(2b-l-3)l+4\lceil \frac{b}{2} \rceil - 4]$ *for odd* l.

Proof. We know $S_1 = (-C_1\epsilon_{b,l}) \cup (-C_2\epsilon_{b,l}) \cup \ldots \cup (-C_k\epsilon_{b,l}) \cup (\epsilon_{b,l})$ and each of the sets in the union above have same number of elements, also $\epsilon_{b,l} = E_{b,l} \cup E'_{b,l}$ and both the sets again have same number of elements, also coefficient C_i's are chosen such that the union sets in S_1 are mutually disjoint. So to prove the theorem it is sufficient to show that (1) $|\epsilon_{b,l}| = (2b-l+1)l$ for $1 \leq l < \lceil \frac{b}{2} \rceil$ and (2) $|\epsilon_{b,l}| = (2b-l-3)l+2b-2$ or $|\epsilon_{b,l}| = (2b-l-3)l+4\lceil \frac{b}{2} \rceil - 4$ for $\lceil \frac{b}{2} \rceil \leq l < b$ with even or odd l.

(1) For $1 \leq l < \lceil \frac{b}{2} \rceil$, set $E_{b,l}$ consists of solid bursts up to length l within a b-bit byte pertaining to the pattern $1 \to 0$. Solid bursts up to length l beginning from 1^{st} position till $(b-l+1)^{th}$ position will be l in numbers and after $(b-l+1)^{th}$ position, we will have $l-1$ solid bursts and so on till the last position having only one choice. Also from Lemma 4.1 for $1 \leq l < \lceil \frac{b}{2} \rceil$, we have no repetition of errors when written in integer form, thus $|E_{b,l}| = (b-l+1)l+l-1+l-2+\ldots+1 = \frac{l}{2}(2b-l+1)$ consequently $|S_1| = 2(k+1)\frac{l}{2}(2b-l+1) = (k+1)(2b-l+1)l$.

(2) Once $l \geq \lceil \frac{b}{2} \rceil$, as discussed in Lemma 4.1, the integer values of the solid bursts start repeating.

- **Case I** (l *is even*): For constructing an error set $\epsilon_{b,l}$ free of repetitions, we remove two elements from the pairs $e_{b,i}$ and $e'_{b,b-i}$ with $1 \leq i \leq b-1$. Thus for $l \geq \frac{b}{2}$, we need to remove $2 + 4(l - \frac{b}{2})$ number of repeating elements. After removal by following the steps discussed in 1 above, we have $|S_1| = (k+1)[(2b-l+1)l-(2+4l-2b)] = (k+1)[(2b-l-3)+2b-2]$.
- **Case II** (l *is odd*): Similar to Case I by removing $4+4(l-\lceil \frac{b}{2} \rceil)$ repeating elements from the pairs e_i and e'_{b-i} with $1 \leq i \leq b-1$, we have $|S_1| = (k+1)[(2b-l+1)l-4+4(l-\lceil \frac{b}{2} \rceil)] = (k+1)[(2b-l-3)l+4\lceil \frac{b}{2} \rceil - 4]$.

□

4.2.2 Decoding procedure

Again, the decoding procedure is similar to Chapter 2 with the LUT_2 consuming $|S_1| \times (2b + \lceil log_2(k+1) \rceil)$ bits. Similarly, the error-correcting steps are similar to the steps discussed in Chapter 2. Example 4.4 explains the error-correction procedure for the discussed codes with $1 \leq l < \lceil \frac{b}{2} \rceil$.

Example 4.4. *Consider* $b = 8$, $l = 3$, *then* $C_1 = 11$, $C_2 = 13$, *further Table 4.1 depicts syndromes generated using Definition 4.2 with corresponding error positions and errors. Suppose a message* 11010111 01110011 *is to be transmitted. After encoding the message becomes* $B_1\, B_2\, C_B$ = 11010111 01110011 11000101. *Message received after transmission may have the following possibilities:*

Case I *(Data byte having unidirectional pattern* $1 \to 0$*): Suppose* 11010<u>000</u> 01110011 11000101 *is received, then* $C_{\bar{B}} = [11 \times 11 + 13 \times 206] \pmod{255} = 249$, *thus syndrome* $S = [249 - 163] \pmod{255} = 86 = (-11 \times 224) \pmod{255}$, *from this we conclude the data byte* B_1 *affected by the unidirectional solid burst* $e = (2^5 + 2^6 + 2^7) \pmod{255}$. *Hence the corrected data byte will be* $B_1 = [11 + 224] \pmod{255} = 235 = 11010111$.

Case II *(Data byte having unidirectional pattern* $0 \to 1$*): Suppose* 11010111 0111<u>11</u>11 11000101 *is received, then* $C_{\bar{B}} = [11 \times 235 + 13 \times 254] \pmod{255} = 22$, *thus syndrome* $S = [22 - 163] \pmod{255} = 114 = (-13 \times 207) \pmod{255}$, *from this we conclude the data byte* B_2 *affected by the unidirectional solid burst* $e = -[2^4 + 2^5] \pmod{255}$. *Therefore the corrected data byte will be* $B_2 = [254 + 207] \pmod{255} = 206 = 01110011$.

Case III *(Check byte having unidirectional pattern* $1 \to 0$*): Suppose* 11010111 01110011 <u>00</u>000101 *is received, then clearly* $C_{\bar{B}} = C_B$ *and* $\bar{C}_B = 163$, *so syndrome* $S = [163 - 160] \pmod{255} = 3$, *thus check byte* C_B *is affected by the unidirectional solid burst* $e = 3 = [2^0 + 2^1] \pmod{255}$, *hence the corrected check byte will be* $[160 + 3] \pmod{255} = 163 = 11000101$.

Case IV *(Check byte having unidirectional pattern* $0 \to 1$*): Suppose* 11010111 01110011 11<u>111</u>101 *is received, then* $C_{\bar{B}} = C_B$ *and* $\bar{C}_B = 191$, *so syndrome* $S =$

$[163-191] \pmod{255} = 227$, *thus check byte is affected by the unidirectional solid burst* $e = 227 = -[2^2+2^3+2^4] \pmod{255}$, *hence the corrected check byte will be* $[191+227] \pmod{255} = 11000101$.

Note: If the syndrome obtained for a received message is not in the look up table, then it is beyond the scope of the decoder. However, this is possible only when the occurred error is not unidirectional solid burst or has length exceeding l. Example 4.5 illustrates the capability of these codes dealing with longer burst lengths.

Example 4.5. *Consider* $b=8$ *and* $l=6$, *then* $C_1 = 11$, *after removing* 10 *repeating integer values, viz.,* $3, 7, 15, 31, 63, 192, 224, 240, 248, 252$, *we have* $|\epsilon_{8,6}| = 56$, *consequently* $|S_1| = 112$. *Encoding and decoding procedure for non-repeating elements is similar to Example 4.4, we shall elaborate the decoding of two error patterns having a common integer value. Suppose* 11111110 *is encoded as* 11111110 01011110 *and after transmission it is received as* <u>00</u>011110 01011110, *then syndrome* $S = [C_{\bar{B}} - \bar{C}_B] \pmod{2^b-1} = [45-122] \pmod{255} = 178 = [-11\times 7] \pmod{255}$. *Thus we have error* $e=7$ *but* 7 *pertains to two error patterns, viz.,* $[2^0+2^1+2^2] \pmod{255}$ *and* $[-2^3-2^4-2^5-2^6-2^7] \pmod{255}$. *For error* $e = [2^0+2^1+2^2] \pmod{255}$, *the sent message is decoded as* $B_1 = [\bar{B}_1 + e] \pmod{2^b-1} = [120+7] \pmod{255} = 127 = 11111110$, *similarly for* $e = 7 = [-2^3-2^4-2^5-2^6-2^7] \pmod{255}$, *we get the same result. But by matching the received message* 00011110 *with error* 11100000, *we observe unidirectional solid burst of nature* $1 \to 0$ *occurring from* 1^{st} *to* 3^{rd} *position during the transmission whereas matching error* $000-1-1-1-1-1$ *with the received message* 00011110 *doesn't give us a solid burst of unidirectional nature. So we consider error* $e = 11100000$ *and ascertain the sent message as* 11111110.

Again let the message 01100000 *be encoded and transmitted as* 01100000 01000010. *Suppose it is received as* 011<u>11111</u> 01000010, *then syndrome* $S = [244-66] \pmod{255} = 178 = [-11\times 7] \pmod{255}$, *again we have error* $7 = [-2^3-2^4-2^5-2^6-2^7] \pmod{255}$ *and* $[2^0+2^1+2^2] \pmod{255}$. *But in this case error* $000-1-1-1-1-1$ *matches with the received message* 01100000 *and allows us to ascertain a solid burst occurring from* 4^{th} *to* 8^{th} *position with nature* $0 \to 1$. *Whereas the error*

11100000 *doesn't match with the received message to ascertain the occurrence of a solid burst of unidirectional type, hence the corrected data byte* $B_1 = [254 + 7] \pmod{255} = 6 = 01100000$.

Same approach is applicable in case of repeating integer values for error detection and correction in the check byte or other data bytes (in case of existence).

Remark 4.6. *For two error patterns having a common integer value, we match the patterns with the received message so as to decide exactly which unidirectional solid burst may have occurred during transmission. However in case of symmetric bursts (refer Definition 1.18) this approach may not always work.*

Table 4.1: $\boldsymbol{LUT_2}$ **for** $\boldsymbol{(24, 16)}$ **integer** $\boldsymbol{(U_3SB)_8}$ **code**

Sl. No.	**Syndrome (S_1)**	**Error Loc.**	**Error (e)**	**Sl. No.**	**Syndrome (S_1)**	**Error Loc.**	**Error (e)**
1	1	3	1	64	128	3	128
2	2	3	2	65	132	1	243
3	3	3	3	66	133	1	127
4	4	3	4	67	134	2	127
5	6	3	6	68	141	2	48
6	7	3	7	69	143	3	143
7	8	3	8	70	146	2	28
8	9	1	231	71	148	2	224
9	11	1	254	72	149	1	56
10	12	3	12	73	151	2	8
11	13	2	254	74	154	1	241
12	14	3	14	75	156	2	243
13	16	3	16	76	158	1	32
14	18	1	207	77	159	3	159
15	22	1	253	78	161	2	223
16	24	3	24	79	164	2	7
17	26	2	253	80	167	1	8

Contd...

Sl. No.	Syndrome (S_1)	Error Loc.	Error (e)	Sl. No.	Syndrome (S_1)	Error Loc.	Error (e)
18	27	2	96	81	169	1	31
19	28	3	28	82	176	1	239
20	31	3	31	83	177	2	6
21	32	3	32	84	178	1	7
22	33	1	252	85	181	2	143
23	36	1	159	86	182	2	241
24	37	2	56	87	183	1	192
25	39	2	252	88	188	2	64
26	43	1	112	89	189	1	6
27	44	1	251	90	191	3	191
28	47	2	16	91	192	3	192
29	48	3	48	92	194	1	191
30	52	2	251	93	198	2	24
31	53	1	227	94	199	3	199
32	54	2	192	95	201	2	63
33	56	3	56	96	202	1	28
34	57	2	231	97	203	2	4
35	61	1	64	98	207	3	207
36	63	3	63	99	208	2	239
37	64	3	64	100	211	1	4
38	66	1	249	101	212	1	143
39	67	2	191	102	216	2	3
40	72	1	63	103	218	2	199
41	73	2	14	104	219	1	96
42	74	2	112	105	222	1	3
43	77	1	248	106	223	3	223
44	78	2	249	107	224	3	224
45	79	1	16	108	227	3	227

Contd...

Sl. No.	Syndrome (S_1)	Error Loc.	Error (e)	Sl. No.	Syndrome (S_1)	Error Loc.	Error (e)
46	86	1	224	109	228	2	159
47	88	1	247	110	229	2	2
48	91	2	248	111	231	3	231
49	94	2	32	112	233	1	2
50	96	3	96	113	237	1	48
51	97	1	223	114	239	3	239
52	99	2	12	115	241	3	241
53	101	1	14	116	242	2	1
54	104	2	247	117	243	3	243
55	106	1	199	118	244	1	1
56	107	2	31	119	246	1	24
57	109	2	227	120	247	3	247
58	112	3	112	121	248	3	248
59	114	2	207	122	249	3	249
60	121	2	128	123	251	3	251
61	122	1	128	124	252	3	252
62	123	1	12	125	253	3	253
63	127	3	127	126	254	3	254

4.3 Evaluation and comparison

In this section, we discuss the implementation, comparison, and probability.

4.3.1 Implementation and comparison

For implementation of these codes in a quad-core processor we refer to Chapter 2. Table 4.3 depicts the memory consumed by the look up tables to store the coefficients and syndromes. The codes are constructed upon the availability of coefficients (Table

4.2) determined using the computer search results (Appendix D).

Table 4.2: **Coefficients for some integer $(U_lSB)_b$ codes**

b	l	**First possible 32 coefficients**
6	2	Not available
6	3	Not available
7	2	5, 9, 13
7	3	11
7	4	11
7	5	11
8	2	7, 9
8	3	11, 13
8	4	11
8	5	11
8	6	11
8	7	11
9	2	5, 7, 9, 11, 13, 19, 23, 25, 29, 35, 51
9	3	5, 11, 19
9	4	11, 19, 45
9	5	11, 19, 45
9	6	11, 19, 45
9	7	11, 19, 45
9	8	11, 19, 45
10	3	5, 13, 17, 19, 23, 25, 29, 43, 47, 49, 83, 101, 103, 107
10	4	13, 19, 23, 35, 49
10	5	13, 27
10	9	13, 27
16	4	9, 11, 13, 19, 23, 25, 29, 31, 35, 37, 41, 43, 47, 49, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 99, 101, 103, 107, 109, 113, 117, 121

Contd...

b	*l*	**First possible 32 coefficients**
16	6	11, 13, 19, 23, 25, 27, 29, 35, 37, 41, 43, 45, 47, 49, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 121, 131, 137
16	7	11, 13, 19, 23, 25, 27, 29, 35, 37, 41, 43, 45, 47, 49, 53, 59, 61, 67, 71, 79, 83, 89, 97, 101, 107, 109, 113, 121, 131, 143, 149, 151
16	8	11, 13, 19, 23, 25, 27, 37, 43, 45, 59, 97, 113, 121, 145, 209
16	10	11, 13, 19, 23, 25, 27, 37, 43, 45, 59, 97, 113, 121, 145, 209
16	13	11, 13, 19, 23, 25, 27, 37, 43, 45, 59, 97, 113, 121, 145, 209
16	15	11, 13, 19, 23, 25, 27, 37, 43, 45, 59, 97, 113, 121, 145, 209
32	6	11, 13, 17, 19, 23, 25, 29, 35, 37, 41, 43, 45, 47, 49, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 121, 127, 131
32	8	11, 13, 19, 23, 25, 27, 29, 35, 37, 41, 43, 45, 47, 49, 51, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97,101, 103, 107, 109, 113, 121, 131
32	12	11, 13, 19, 23, 25, 27, 29, 35, 37, 41, 43, 45, 49, 51, 53, 61, 67, 71, 79, 83, 89, 97, 101, 103, 107, 109, 113, 121, 131, 137, 139, 143
32	15	11, 13, 19, 23, 25, 27, 29, 35, 37, 41, 43, 45, 47, 49, 51, 53, 59, 61, 67, 71, 79, 83, 89, 97, 101, 103, 107, 109, 113, 121, 131, 137
32	16	11, 13, 19, 23, 25, 27, 29, 35, 37, 41, 43, 45, 47, 49, 51, 53, 59, 61, 67, 71, 79, 83, 89, 97, 101, 103, 107, 109, 113, 121, 131, 137
32	20	11, 13, 19, 23, 25, 27, 29, 35, 37, 41, 43, 45, 47, 49, 51, 53, 59, 61, 67, 71, 79, 83, 89, 97, 101, 103, 107, 109, 113, 121, 131, 137

Bose and Al-Bassam [19] have discussed byte unidirectional error detecting and

Table 4.3: **Memory requirement for some integer $(U_lSB)_b$ codes**

Codes	b	l	LUT_1 **size**	LUT_2 **size**	**Number of table look ups**
(272,256)	16	6	4× 32B	12.74 KB	$1 \leq \eta_{TL} \leq 13$
(256,240)	16	8	4× 30B	14.26 KB	$1 \leq \eta_{TL} \leq 13$
(256,240)	16	10	4× 30B	15.84KB	$1 \leq \eta_{TL} \leq 13$
(544,512)	32	12	4× 64B	93.25 KB	$1 \leq \eta_{TL} \leq 15$
(1056,1024)	32	16	4× 128B	0.23 MB	$1 \leq \eta_{TL} \leq 16$
(1056,1024)	32	20	4× 128B	0.25 MB	$1 \leq \eta_{TL} \leq 16$

correcting codes, where a codeword is of the form $IB_1\ IB_2 \ldots IB_k\ PB\ ARC$. IB_i are k information b-bit bytes, whereas PB and ARC (parity byte and arithmetic residue check) are two check bytes respectively of b and s bits obtained using some XOR operations upon the information bits. The value of s depends upon the smallest prime p such that $\frac{p-1}{2} \geq k$ and $p > b$ or $p > max\,(2k,\, b)$, clearly $1 < s < b$. Now, by using Result 1.50 for redundancy, we have R_2(rate of this code) $= \frac{bk}{(b+1)k+s} < \frac{bk}{(b+1)k} = R_1$ (rate for the proposed code). Since table look ups are not used in [19], so we abstain from comparing.

Similarly by matching the proposed codes on the lines of the binary codes introduced by Shiva and Sheng [100] with bk information bits correcting solid bursts of length l, we observe the code rate R_3 (for [100]) $= \frac{bk}{bk(2l+1)+2l}$. Since $bk + 2bkl + 2l > bk + b$, hence $R_3 < \frac{bk}{bk+b} = R_1$. Although the existence of the codes in [100] depend on the availability of at least $2l + 1$ runs of 1's unlike the existence determined by computer search results for the proposed codes.

Sharma and Dass [99] have discussed solid burst correcting binary codes (perfect), here table operations are done for error detection/correction purpose. A look up table here has three columns, viz., sl. no., error pattern and the corresponding syndromes. Upon existence of a linear code in [99] and by drawing the parameters on the lines with the proposed $((k+1)b, kb)$ codes correcting solid bursts of length l, the required memory for codes in [99] is $X \times [2b + kb]$ whereas memory required for the

Table 4.4: **Various integer codes with information rate $\frac{512}{544}$**

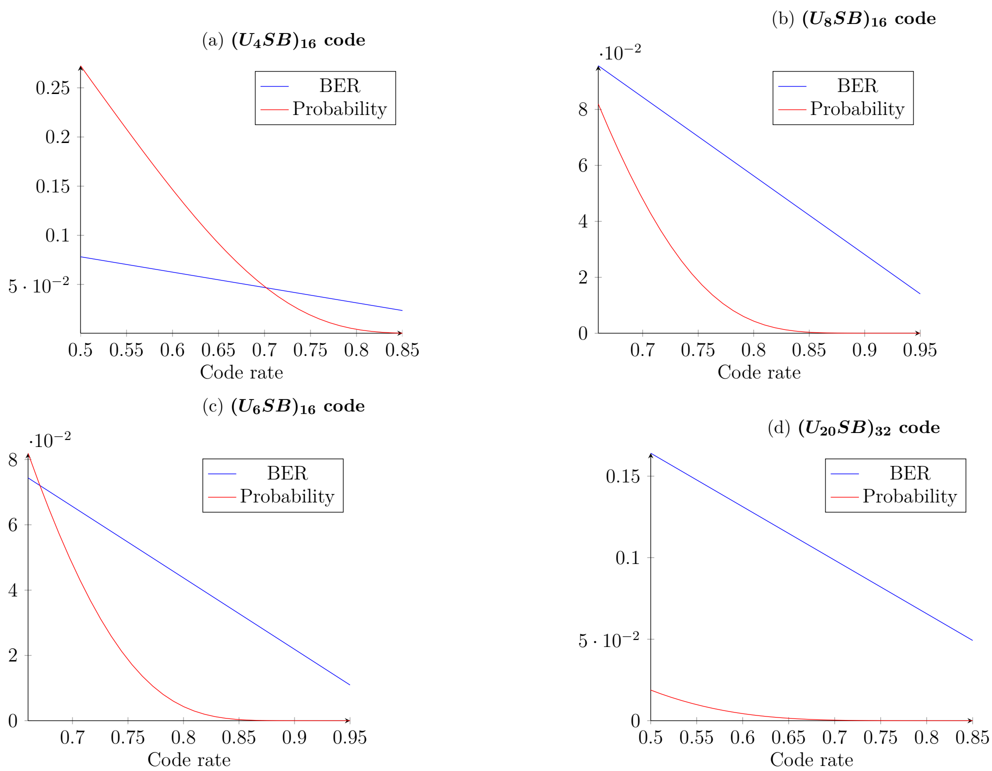
Codes	**Error type**	b	l	LUT_2 **size**	**Maximum # of table look ups**
Proposed	Unidirectional solid bursts	32	3	27.28 KB	13
Result 1.31	Double and triple adjacent	32	3	7.81 MB	21
Proposed	Unidirectional solid bursts	32	9	73.9 KB	15
Result 1.36	Asymmetric burst and double asymmetric	32	9	3.35 MB	19
Result 1.19	Symmetric bursts	32	9	15.01 MB	19
Result 1.24	Double asymmetric	32	NA	1.96 MB	19

proposed codes is $X \times [2b + \lceil log_2(k+1) \rceil]$, here X denotes the number of possible error patterns of the discussed type. We observe that $X \times [2b+kb] > X \times [2b + \lceil log_2(k+1) \rceil]$, implying the proposed codes consume less memory. The codes discussed here have a few more benefits than the integer $(DEC - (TAEC)_b)$ codes (refer [78]), which already have many additional benefits compared to other adjacent error-correcting codes defined over finite fields. A few major benefits can be highlighted as follows:

- Capability to correct adjacent bursts (rather than confined up to 3 adjacent) with maximum possible length within a b-bit byte.
- Less consumption of memory with fewer table look ups, etc.

Table 4.4 compares some integer codes capable of correcting different types of errors in terms of LUT_2 sizes and # of table look ups.

Figure 4.1: **Change in probability and BER for different code rates**



4.3.2 Probability and BER

We consider the probability of $1 \to 0$ and $0 \to 1$ equal to ϵ because the codes are investigated over a binary symmetric channel (refer Figure 1.2). The next theorem determines the probability of erroneous decoding for integer $(U_lSB)_b$ codes.

Theorem 4.7. *The probability of erroneous decoding $P_d(UB)$ for a $((k+1)b, kb)$ integer $(U_lSB)_b$ code is*

$$2(k+1)\left[b\left\{\frac{\epsilon(1-\epsilon)^{(k+1)b}\left(1-\left(\frac{\epsilon}{1-\epsilon}\right)^l\right)}{1-2\epsilon}\right\}-\left\{\frac{\epsilon(1-\epsilon)^{(k+1)b+1}\left((l-1)\left(\frac{\epsilon}{1-\epsilon}\right)^{l+1}-l\left(\frac{\epsilon}{1-\epsilon}\right)^l+\left(\frac{\epsilon}{1-\epsilon}\right)\right)}{(1-2\epsilon)^2}\right\}\right],$$ *where ϵ is the crossover probability.*

Proof. Similar to the theorem earlier, an encoded codeword from the $((k+1)b, kb)$ integer $(U_lSB)_b$ code has $(k+1)$ b-bit blocks, and the code is capable to correct a unidirectional solid burst up to length l occurring within a b-bit byte at a time. Once the length of the unidirectional solid burst becomes greater than $\lceil \frac{l}{2} \rceil - 1$, we begin to observe common integer values (refer Lemma 4.1 and Theorem 4.3) even though the patterns are different. But, for the purpose of determining probability, it makes no difference because we calculate probability based on error patterns rather than integer values. Since there are $2b$ number of unidirectional solid bursts of length 1 within a b-bit byte, thus the probability in this case will be $2b\epsilon(1-\epsilon)^{(k+1)b-1}$. Similarly for unidirectional solid bursts of length 2, the probability will be $2(b-1)\epsilon^2(1-\epsilon)^{(k+1)b-2}$. Continuing this, the probability of erroneous decoding for a unidirectional solid burst of length l will be $2(b-l+1)\epsilon^l(1-\epsilon)^{(k+1)b-l}$. Thus the probability of erroneous decoding for unidirectional solid bursts up to length l occurring within a b-bit byte will be

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

Since $\epsilon << 0.5$, so

$$\sum_{i=1}^{l-1}\left(\frac{\epsilon}{1-\epsilon}\right)^i=\frac{\left(\frac{\epsilon}{1-\epsilon}\right)\left(1-\left(\frac{\epsilon}{1-\epsilon}\right)^{l-1}\right)}{\left(1-\frac{\epsilon}{1-\epsilon}\right)}.$$

Now by differentiating both sides with respect to $\left(\frac{\epsilon}{1-\epsilon}\right)$, we have

$$\sum_{i=1}^{l-1}i\left(\frac{\epsilon}{1-\epsilon}\right)^{i-1}=\frac{l\left(\frac{\epsilon}{1-\epsilon}\right)^l-l\left(\frac{\epsilon}{1-\epsilon}\right)^{l-1}-\left(\frac{\epsilon}{1-\epsilon}\right)^l+1}{\left(\frac{1-2\epsilon}{1-\epsilon}\right)^2}.$$

Again by multiplying both sides with $\frac{\epsilon}{1-\epsilon}$, we have

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

Now by substituting this value in the probability of erroneous decoding within a b-bit byte, we obtain the probability as

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

Since the integer $(U_lSB)_b$ codes are capable of correcting one unidirectional solid burst at a time within a b-bit byte and there are $k+1$ such b-bit bytes, thus the total probability will be

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

□

As discussed in Subsection 3.3.2, we determine

$$\begin{aligned} BER &= \frac{1}{l}\left[\frac{1}{(k+1)b}+\frac{2}{(k+1)b}+\ldots+\frac{l}{(k+1)b}\right] \\ &= \frac{1}{(k+1)bl}\left[\frac{l(l+1)}{2}\right] \\ &= \frac{l+1}{2(k+1)b}. \end{aligned}$$

In Figure 4.1, the change in BER and probability is shown with a changing code rate for different integer $(U_lSB)_b$ codes. It is evident from Table 4.2 and Appendix D that the codes are capable of correcting the discussed bursts up to maximum possible length. Therefore, while implementing these codes in a channel where this type of bursts are prevalent, the chances of unidirectional solid bursts going undetected become negligible. So we refrain from discussing the conditions of such errors going undetected.

4.4 Conclusion

In this chapter, we have presented a class of integer codes capable of correcting unidirectional solid bursts occurring within a b-bit byte. Also, it is compared with some similar error-correcting codes and found suitable in many ways. We discovered that the probability and BER of these codes decrease as the code rate increases. Similar error-correcting procedures can be analysed for symmetric channels having different crossover probabilities for $1 \to 0$ and $0 \to 1$.