This is the peer reviewed version of the following article:

Radonjic, A., Vujicic, V., 2019. Integer Codes Correcting Burst Asymmetric Errors Within a Byte. IETE Journal of Research.
https://doi.org/10.1080/03772063.2019.1593056

.

Integer Codes Correcting Burst Asymmetric Errors within a Byte

Aleksandar Radonjic* and Vladimir Vujicic

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

***Abstract* - This paper presents two types of integer codes capable of correcting burst asymmetric errors within a byte. The presented codes are constructed with the help of a computer and are very efficient in terms of redundancy. The results of a computer search have shown that, for practical data lengths up to 4096 bits, the presented codes use up to two check-bits less than the best burst asymmetric error correcting codes. Besides this, it is shown that the presented codes are suitable for implementation on modern processors.**

***Keywords* - Integer codes, error correction, burst asymmetric errors, multicore processors.**

1. Introduction

Most classes of channel codes have been developed for use on binary symmetric channel, where the error probabilities $1 \rightarrow 0$ and $0 \rightarrow 1$ are equal. However, in certain systems the error statistics are different. For instance, in optical networks without optical amplifiers (e.g. access networks) photons may fade or fail to be detected, but new photons cannot be generated. Hence, if the receiver operates correctly, only asymmetric $(1 \rightarrow 0)$ errors can occur [1], [2]. On the other hand, in WOM memories, such as digital optical disks, $1 \rightarrow 0$ errors are not possible. The reason is that in these systems the l's correspond to the holes burned into the coating [3]. So, once a 1 is written in a bit position, it cannot be changed back into a 0.

With this in mind, in this paper, we present two types of integer codes capable of correcting l-bit burst asymmetric errors within a b-bit byte ($B_{l/b}A$ errors). The first type of codes (type-I integer $B_{l/b}AEC$ codes) is designed to correct "negative" $(1 \rightarrow 0)$ $B_{l/b}A$ errors, while the second type of codes (type-II integer $B_{l/b}AEC$ codes) can correct "positive" $(0 \rightarrow 1)$ $B_{l/b}A$ errors. Like all other integer codes [4]-[11], the presented ones can be interleaved without delay and without using dedicated hardware. Owing to this, they can be transformed into simple codes capable of correcting (multiple) burst asymmetric (BA) errors.

2. Codes Construction

At the beginning, let us remind the general definition of integer error control codes (IECCs).

Definition 1. [9] *Let* $Z_{2^b-1} = \{0, 1,\ldots, 2^b - 2\}$ *be the ring of integers modulo* $2^b - 1$ *and let* $B_i = \sum_{n=0}^{b-1} a_n \cdot 2^n$ *be the integer representation of a b-bit byte, where* $a_n \in \{0, 1\}$ *and* $1 \le i \le k$.

Then, the code $C(b, k, c)$, defined as

$$C(b,k,c)=\left\{(B_1,B_2,...,B_k,B_{k+1})\in Z_{2^b-1}^{k+1}:\ \sum_{i=1}^{k} C_i\cdot B_i \equiv B_{k+1}\ (\text{mod } 2^b-1)\right\} \quad (1)$$

is an $(kb + b, kb)$ IECC, where $c = (C_1, C_2, ..., C_k, 1) \in Z_{2^b-1}^{k+1}$ is the coefficient vector and $B_{k+1} \in Z_{2^b-1}$ is an integer.

To construct type-I and type-II integer $B_{l/b}$AEC codes, it is necessary to know the integer values of both types of $B_{l/b}$A errors. For that purpose, we will rely on the analysis from [4]. In that paper, it was shown that the integer value of a l-bit burst error within a b-bit byte is equal to $e = \pm 2^r\cdot(2m - 1)$, where $0 \le r \le b - l$, $1 \le m \le 2^{x-1}$ and $1 \le x \le l$. Based on this it is easy to conclude that the integer values of "negative" and "positive" $B_{l/b}$A errors are respectively equal to $e^- = -2^s\cdot(2n - 1)$ and $e^+ = 2^s\cdot(2n - 1)$, where $0 \le s \le b - l$, $1 \le n \le 2^{u-1}$ and $1 \le u \le l$. Knowing this, we are able to construct both types of integer $B_{l/b}$AEC codes.

2.1. Type-I integer $B_{l/b}$AEC codes

Definition 2. *Let $x = (B_1, B_2,\ldots, B_k, B_{k+1}) \in Z_{2^b-1}^{k+1}$, $y = (\underline{B}_1, \underline{B}_2,\ldots, \underline{B}_k, \underline{B}_{k+1}) \in Z_{2^b-1}^{k+1}$ and $e = y - x = (\underline{B}_1 - B_1, \underline{B}_2 - B_2,\ldots,\underline{B}_k - B_k, \underline{B}_{k+1} - B_{k+1}) = (e_1, e_2,..., e_k, e_{k+1}) \in Z_{2^b-1}^{k+1}$ be respectively, the sent codeword, the received codeword and the error vector. Then, an $(kb + b, kb)$ IECC is said to be type-I integer $B_{l/b}$AEC code if it can correct error vectors from the set $E = \{(e^-, 0,..., 0, 0),..., (0, 0, ..., e^-, 0), (0, 0, ..., 0, -e^-)\}$ where $e^- \in \{-2^s\cdot(2n - 1): 0 \le s \le b - l, 1 \le n \le 2^{u-1}, 1 \le u \le l\}$.*

Definition 3. *The error set for $(kb + b, kb)$ type-I integer $B_{l/b}$AEC codes is defined by*

$$\xi_{b,l,k}^{-} = s_1 \cup s_2 \quad (2)$$

where

$$s_1=\left\{\left[-2^s\cdot(2n-1)\cdot C_i\right]\left(\text{mod } 2^b-1\right):\ 0\le s\le b-l, 1\le n\le 2^{u-1}, 1\le u\le l, 1\le i\le k\right\} \quad (3)$$

$$s_2=\left\{\left[\ 2^s\cdot(2n-1)\right]\left(\text{mod } 2^b-1\right):\ 0\le s\le b-l, 1\le n\le 2^{u-1}, 1\le u\le l\right\} \quad (4)$$

From the above it is clear that type-I integer $B_{l/b}$AEC codes cannot be constructed without knowing the values of the C_i's. This fact, however, does not prevent us to state the following theorem.

Theorem 1. *The codes defined by (1) can correct all "negative" $B_{l/b}$A errors iff there exist k mutually different coefficients $C_i \in Z_{2^b-1} \setminus \{0,1\}$ such that*

$$\left|\xi_{b,l,k}^{-}\right| = \left[2^{l-1}\cdot(b-l+2)-1\right]\cdot(k+1)$$

where $|A|$ denotes the cardinality of A.

Proof. Observe that the set $\xi_{b,l,k}^{-}$ can be expressed as

$$\xi_{b,l,k}^{-} = \bigcup_{i=1}^{2l} Z_i$$

where

$$Z_1 = \left\{ \left[-2^s \cdot (1) \cdot C_i \right] (\text{mod } 2^b - 1) : 0 \le s \le b-1,\ 1 \le i \le k \right\},$$

$$Z_2 = \left\{ \left[2^s \cdot (1) \right] (\text{mod } 2^b - 1) : 0 \le s \le b-1 \right\},$$

$$Z_3 = \left\{ \left[-2^s \cdot (3) \cdot C_i \right] (\text{mod } 2^b - 1) : 0 \le s \le b-2,\ 1 \le i \le k \right\},$$

$$Z_4 = \left\{ \left[2^s \cdot (3) \right] (\text{mod } 2^b - 1) : 0 \le s \le b-2 \right\},$$

$$\vdots$$

$$Z_{2l\text{-}1} = \left\{ \left[-2^s \cdot (2^{l\text{-}1}+1, 2^{l\text{-}1}+3, ..., 2^l-1) \cdot C_i \right] (\text{mod } 2^b - 1) : 0 \le s \le b-l, 1 \le i \le k \right\},$$

$$Z_{2l} = \left\{ \left[2^s \cdot (2^{l\text{-}1}+1, 2^{l\text{-}1}+3, ..., 2^l-1) \right] (\text{mod } 2^b - 1) : 0 \le s \le b-l \right\}.$$

Now, suppose that the coefficients $C_i \in Z_{2^b\text{-}1} \setminus \{0,1\}$ have values such that

$$\bigcap_{i=1}^{2l} Z_i = \varnothing,$$

$$|Z_1| = k \cdot b,$$

$$|Z_2| = b,$$

$$|Z_{2h\text{-}1}| = k \cdot 2^{h\text{-}2} \cdot (b-h+1),\ 2 \le h \le l,$$

$$|Z_{2h}| = 2^{h\text{-}2} \cdot (b-h+1),\ 2 \le h \le l.$$

In that case, it is easy to show that

$$\left|\xi^{-}_{b,l,k}\right| = \sum_{i=1}^{2l} |Z_i| = \left[2^{l\text{-}1} \cdot (b-l+2) - 1 \right] \cdot (k+1).$$

Conversely, if the codes satisfy the above condition, then we correct all "negative" $B_{l/b}$A errors. Therefore, these codes are (kb + b, kb) type-I integer $B_{l/b}$AECcodes. □

2.2. Type-II integer $B_{l/b}$AEC codes

Using the same method as above, we can construct type-II integer $B_{l/b}$AECcodes.

Definition 4. *Let* $x = (B_1, B_2, \ldots, B_k, B_{k+1}) \in Z^{k+1}_{2^b-1}$, $y = (\underline{B}_1, \underline{B}_2, \ldots, \underline{B}_k, \underline{B}_{k+1}) \in Z^{k+1}_{2^b-1}$ *and* $e = y - x = (\underline{B}_1 - B_1, \underline{B}_2 - B_2, \ldots, \underline{B}_k - B_k, \underline{B}_{k+1} - B_{k+1}) = (e_1, e_2, ..., e_k, e_{k+1}) \in Z^{k+1}_{2^b-1}$ *be respectively, the sent codeword, the received codeword and the error vector. Then, an* (kb + b, kb) *IECC is said to be type-II integer $B_{l/b}$AEC code if it can correct error vectors from the set* $E = \{(e^+, 0, ..., 0, 0), ..., (0, 0, ..., e^+, 0), (0, 0, ..., 0, -e^+)\}$ *where* $e^+ \in \{2^s \cdot (2n-1) : 0 \le s \le b-l,\ 1 \le n \le 2^{u\text{-}1},\ 1 \le u \le l\}$.

Definition 5. *The error set for* (kb + b, kb) *type-II integer $B_{l/b}$AEC codes is defined by*

$$\xi^{+}_{b,l,k} = s_3 \cup s_4 \tag{5}$$

where

$$s_3 = \left\{ \left[2^s \cdot (2n-1) \cdot C_i \right] \left(\text{mod } 2^b - 1 \right) : 0 \le s \le b-l, 1 \le n \le 2^{u-1}, 1 \le u \le l, 1 \le i \le k \right\} \tag{6}$$

$$s_4 = \left\{ \left[-2^s \cdot (2n-1) \right] \left(\text{mod } 2^b - 1 \right) : 0 \le s \le b-l, 1 \le n \le 2^{u-1}, 1 \le u \le l \right\} \tag{7}$$

As in the previous section, we can state the following.

Theorem 2. *The codes defined by (1) can correct all "positive" $B_{l/b}A$ errors iff there exist k mutually different coefficients* $C_i \in Z_{2^b-1} \setminus \{0,1\}$ *such that*

$$\left|\xi^{+}_{b,l,k}\right| = \left[2^{l-1}\cdot(b-l+2)-1\right]\cdot(k+1).$$

Proof. The proof is basically the same as in Theorem 1. Hence, it is omitted. □

Since the sets $\xi^{-}_{b,l,k}$ and $\xi^{+}_{b,l,k}$ have the same cardinality, we can state the theorem that relates to both types of codes.

Theorem 3. *For any* ($kb + b$, kb) *integer $B_{l/b}AEC$ code it holds that*

$$k \le \left\lfloor \frac{2^b-2}{2^{l-1}\cdot(b-l+2)-1} - 1 \right\rfloor.$$

Proof. From Definition 1 we know that the total number of nonzero syndromes is equal to $2^b - 2$. On the other hand, from Theorems 1 and 2 we know that the sets $\xi^{-}_{b,l,k}$ and $\xi^{+}_{b,l,k}$ have $\left[2^{l-1}\cdot(b-l+2)-1\right]\cdot(k+1)$ nonzero elements. Hence, we obtain the inequality

$$\left[2^{l-1}\cdot(b-l+2)-1\right]\cdot(k+1) \le 2^b-2$$

wherefrom it follows that

$$k \le \left\lfloor \frac{2^b-2}{2^{l-1}\cdot(b-l+2)-1} - 1 \right\rfloor. \ \square$$

The last step in constructing both types of codes is to find the C_i's that satisfy the conditions of Theorems 1 and 2. For that purpose it is necessary to perform an exhaustive search on all possible candidates from the set $Z_{2^b-1} \setminus \{0,1\}$. In this paper, we have restricted ourselves to the codes with parameters $3 \le l \le 5$ and $6 \le b \le 16$. The obtained results are shown in Tables 1-3.

Table 1. Number of coefficients for some integer $B_{l/b}$AEC codes obtained via computer search.

		$b = 6$	$b = 7$	$b = 8$	$b = 9$	$b = 10$	$b = 11$	$b = 12$	$b = 13$	$b = 14$	$b = 15$	$b = 16$
$l = 3$	Theoretical bound	2	4	8	15	28	51	94	173	320	594	1109
	Type-I codes	0	1	4	7	12	25	36	98	172	297	601
	Type-II codes	0	1	4	7	12	25	37	98	174	297	601
$l = 4$	Theoretical bound	0	1	2	8	15	27	50	93	171	317	589
	Type-I codes	0	0	0	1	3	10	12	38	68	129	226
	Type-II codes	0	0	0	2	4	9	12	36	67	126	225
$l = 5$	Theoretical bound	0	1	2	2	4	11	27	50	92	170	315
	Type-I codes	0	0	0	0	1	1	4	10	20	41	76
	Type-II codes	0	0	0	0	1	3	5	11	19	41	77

Table 2. Coefficients for type-I integer $B_{l/b}$AEC codes with parameters $3 \le l \le 5$, $b = 16$ and $k \le 128$.

$l = 3$															
2	9	11	13	17	19	23	25	29	31	37	41	43	47	49	53
59	61	67	71	73	79	81	83	89	97	99	101	103	105	107	109
113	117	121	127	131	137	139	143	149	151	153	157	163	167	169	173
179	181	187	191	193	197	199	207	209	211	221	223	225	227	229	233
239	241	247	251	253	261	263	271	275	277	279	281	283	285	289	307
311	313	317	319	323	325	331	337	341	347	349	353	359	361	367	369
373	377	379	383	387	389	391	401	403	407	409	419	421	423	425	431
433	437	441	443	449	451	457	463	467	473	477	479	481	499	503	509
$l = 4$															
2	17	19	21	23	25	29	31	37	41	43	47	53	59	61	67
71	73	79	81	83	89	97	101	103	107	109	113	121	127	131	149
151	157	163	167	169	173	179	181	191	199	211	223	227	229	233	239
241	245	251	269	271	277	283	289	307	311	317	323	331	337	349	353
357	359	361	383	391	409	419	429	431	433	437	449	467	483	493	499
509	521	551	557	563	575	577	579	593	601	609	629	647	653	661	673
683	697	701	713	727	733	743	761	773	787	809	817	883	887	893	899
901	907	929	983	989	999	1009	1013	1019	1049	1051	1061	1069	1073	1087	1091
$l = 5$															
2	33	35	37	41	43	47	53	59	61	67	71	73	79	83	97
101	107	113	117	127	137	149	157	163	179	227	233	251	271	283	289
311	313	347	349	383	449	453	545	557	563	593	631	651	859	877	905
911	941	969	1009	1011	1061	1235	1249	1259	1613	1787	1889	2019	2187	2317	2489
3071	3571	4651	4903	7577	8051	10751	10867	11677	15103	24431	24567				

Table 3. Coefficients for type-II integer $B_{l/b}$AEC codes with parameters $3 \le l \le 5$, $b = 16$ and $k \le 128$.

$l = 3$															
9	11	13	17	19	23	25	29	31	37	41	43	47	49	53	59
61	67	71	73	79	81	83	89	97	99	101	103	105	107	109	11
117	121	127	131	137	139	143	149	151	153	157	163	167	169	173	179
181	187	191	193	197	199	207	209	211	221	223	225	227	229	233	239
241	247	251	253	261	263	271	275	277	279	281	283	285	289	307	311
313	317	319	323	325	331	337	341	347	349	353	359	361	367	369	373
377	379	383	387	389	391	401	403	407	409	419	421	423	425	431	433
437	441	443	449	451	457	463	467	473	477	479	481	499	503	509	517
$l = 4$															
17	19	21	23	25	29	31	37	41	43	47	53	59	61	67	71
73	79	81	83	89	97	101	103	107	109	113	121	127	131	149	151
157	163	167	169	173	179	181	191	199	211	223	227	229	233	239	241
245	251	269	271	277	283	289	307	311	317	323	331	337	349	353	357
359	361	383	391	409	419	429	431	433	437	449	467	483	493	499	509
521	551	557	563	575	577	579	593	601	609	629	647	653	661	673	683
697	701	713	727	733	743	761	773	787	809	817	819	883	887	893	899
901	907	929	983	989	999	1009	1013	1019	1049	1051	1061	1069	1073	1087	1091
$l = 5$															
33	35	37	41	43	47	53	59	61	67	71	73	79	83	89	97
101	107	113	117	127	137	149	157	163	179	227	233	251	271	283	311
347	349	357	383	449	453	521	545	557	563	593	723	739	743	837	859
877	905	911	967	1009	1045	1061	1289	1559	1613	1787	1889	2021	2027	2321	2387
2489	3677	3821	4093	4693	5299	6143	6653	6971	10069	11677	23551	24503			

3. Error Correction Procedure

From Definition 1 it is easy to conclude that there exists only one syndrome. It is generated using the expression

$$S = \underline{B}_{k+1} - B_{k+1} \pmod{2^b - 1} \tag{8}$$

after which the decoder will either accept the recieved codeword ($S = 0$) or try to recover the original one ($S \neq 0$). In the latter case, the decoder will lookup the syndrome table to get the error correction data. From Theorems 1-2, we see that the syndrome table has $\left|\xi^{+}_{b,l,k}\right| = \left|\xi^{+}_{b,l,k}\right| = \left|\xi_{b,l,k}\right| = \left[2^{l\text{-}1} \cdot (b - l + 2) - 1\right] \cdot (k+1)$ entries, where each entry describes a unique relationship between the syndrome (element of the set $\xi_{b,l,k}$), error location (i) and error vector (e) (Fig. 1).

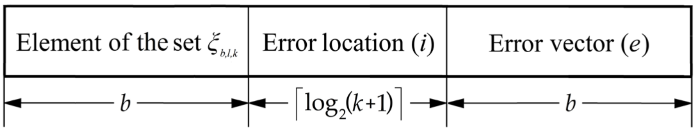

Fig. 1. Bit-width of one syndrome table entry.

So, if the elements of $\xi_{b,l,k}$ are sorted in increasing order, the decoder will find the appropriate entry after n_{TL} table lookups and n_{TL} comparisons $(1 \le n_{\mathrm{TL}} \le \left\lfloor log_2 \left|\xi_{b,l,k}\right| \right\rfloor + 2)$ [12]. In the next step, using the error correction data, the decoder will execute the operation

$$B_i = \underline{B}_i - e \pmod{2^b - 1},\ 1 \le i \le k+1; \tag{9}$$

where $e = e^{-}$ or $e = e^{+}$.

4. Evaluation and Implementation Strategy

By analyzing the data from Table 1 we note that both types of codes protect approximately the same number of data bits. More precisely, for values b = 6, 7, 8, 11 and 15 type-I codes are slightly more rate-efficient than type-II codes, while for values b = 9, 10 and 12 the situation is reversed. In all other cases (b = 13, 14 and 16), the mentioned codes are equally effecient in terms of code rate.

In addition to the above, Table 1 shows the theoretical bounds on the number of the C_i's. Although these bounds may indicate that the proposed codes are rate-inefficient, the truth is quite the opposite. This confirms the results of the comparison of the proposed codes with the best burst asymmetric error correcting codes [11]. Unlike the proposed codes, these codes use $l + log_2 K + (1/2) \cdot log_2\, log_2 K$ check bits, where K is the number of data bits. From this it is easy to show that, for practical data lengths up to 4096 bits, the proposed codes require one or two check-bits less than the codes from [11] (Table 4). The similar applies when comparing the proposed codes with integer codes capable of correcting l-bit burst errors within a b-bit byte [4].

Table 4. Check-bit lengths of the proposed codes and the codes from [4] and [11].

Data word length (in bits)	Type-I Integer $B_{l/b}$AEC Codes			Type-II Integer $B_{l/b}$AEC Codes			Codes from [4]			Codes from [11]		
	$l = 3$	$l = 4$	$l = 5$	$l = 3$	$l = 4$	$l = 5$	$l = 3$	$l = 4$	$l = 5$	$l = 3$	$l = 4$	$l = 5$
$K = 128$	11	12	13	11	12	13	12	13	14	12	13	14
$K = 256$	11	13	14	11	13	14	13	14	15	13	14	15
$K = 512$	13	14	15	13	14	15	13	15	16	14	15	16
$K = 1024$	13	15	16	13	15	16	14	16	17	15	16	17
$K = 2048$	14	16	17	14	16	17	15	17	18	16	17	18
$K = 4096$	15	17	18	15	17	18	16	18	19	17	18	19

In this case, for all values of l and K, except $l = 3$ and $K = 512$, the proposed codes require one or two check-bits less than the codes from [4].

Besides being rate-efficient, the proposed codes are extremely suitable for implementation on modern processors. To illustrate this, suppose that the decoder implemented on a ten-core processor (Fig. 2) with the following specifications [13], [14]:

1) clock rate: $C_R = 3.1 \cdot 10^9$ Hz,
2) integer addition/subtraction operation: 1 cycle latency,
3) integer multiplication operation: 3 cycles latency,
4) 128-bit shift operation: 1 cycle latency,
5) modulo reduction operation: 1 cycle latency,
6) comparison operation: 1 cycle latency,
7) access to the L1 cache (64 KB per core): 4 cycles latency,
8) access to the L2 cache (256 KB per core): 12 cycles latency,
9) access to the L3 cache (25 MB shared): 34 cycles latency.

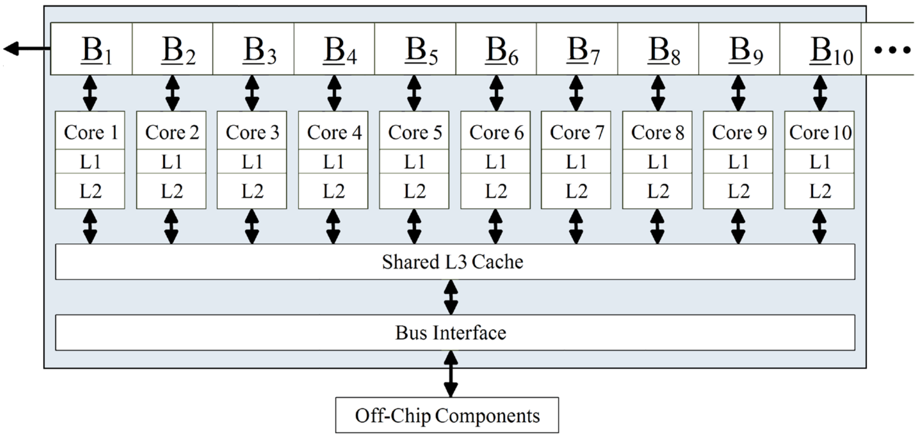

Fig. 2. Block diagram of ten-core processor.

In addition, let us suppose that the data word has $K = 10 \cdot b \cdot k = 160 \cdot k$ bits, that the coefficients C_i (Tables 2 and 3) are stored in each of the ten L1 caches and that the syndrome table is placed in each of the ten L2 caches. In that case, the decoder will perform the following operations:

- *Core* 1

$$\underline{B}_{k+1}^{(1)} = \sum_{i=1}^{k} C_i \cdot \underline{B}_{10\cdot(i-1)+1} \pmod{2^{16}-1} \tag{10}$$

- *Core* 2

$$\underline{B}_{k+1}^{(2)} = \sum_{i=1}^{k} C_i \cdot \underline{B}_{10\cdot(i-1)+2} \pmod{2^{16}-1} \tag{11}$$

$\vdots$

- *Core* 10

$$\underline{B}_{k+1}^{(10)} = \sum_{i=1}^{k} C_i \cdot \underline{B}_{10\cdot(i-1)+10} \pmod{2^{16}-1} \tag{12}$$

If we add to this $K/128$ shift operations, we conclude that the decoder requires $T_1 = 8\cdot k + K/128$ clock cycles (k accesses to the L1 cache, k integer multiplications, k – 1 integer additions, $K/128$ shift operations and 1 modulo reduction) to compute all check-bytes. After finishing this task, the decoder will take $T_2 = 2$ clock cycles (1 integer subtraction and 1 modulo reduction) to calculate the values:

- *Core* 1

$$S^{(1)} = [\underline{B}_{k+1}^{(1)} - B_{k+1}^{(1)}] \pmod{2^{16}-1} \tag{13}$$

- *Core* 2

$$S^{(2)} = [\underline{B}_{k+1}^{(2)} - B_{k+1}^{(2)}] \pmod{2^{16}-1} \tag{14}$$

$\vdots$

- *Core* 10

$$S^{(10)} = [\underline{B}_{k+1}^{(10)} - B_{k+1}^{(10)}] \pmod{2^{16}-1} \tag{15}$$

As explained in the previsous section, if the data are received in error, the decoder will perform n_{TL} table lookups, n_{TL} comparisons, 2 integer additions and 1 modulo reduction. In our case, ten such operations will be executed in parallel in $T_3 = 13\cdot n_{TL} + 3$ clock cycles. So, if we sum up all the processing times, we come to the conclusion that the decoder requires

$$T_{total} = T_1 + T_2 + T_3 = 8\cdot k + K/128 + 13\cdot n_{TL} + 5 \tag{16}$$

clock cycles to process K data bits, i.e. one second to decode

$$G = \frac{C_R}{T_{total}/K} = \frac{(3.1\cdot 10^9)\cdot 160\cdot k}{8\cdot k + 160\cdot k/128 + 13\cdot n_{TL} + 5} \tag{17}$$

data bits. By substituting the values of k and n_{TLmax} (Table 5) in (17) we obtain that $G_{min} = 40.08$ Gbps and $G_{max} = 49.70$ Gbps. This means that all considered codes have the potential to be used in various real-time systems (e.g. 10G and 40G networks). In addition, from (10)-(15) we observe that all analyzed codes are interleaved at the byte level. Thanks to this, they are able to correct (mulitple) BA errors up to l bits.

Table 5. Memory Requirements and Theoretical Decoding Throughputs for Some Ten-Byte Interleaved Integer $B_{l/16}$AEC Codes.

Code	k	l	Memory Requirements for Storing the Coefficients C_i	Memory Requirements for Storing the Syndrome Table	Number of Table Lookups	Minimum Theoretical Decoding Throughput
(1040, 1024)	64	3	10 x 128 B	18.70 KB	$1 \le n_{TL} \le 13$	41.43 Gbps
(1040, 1024)	64	4	10 x 128 B	35.17 KB	$1 \le n_{TL} \le 14$	40.75 Gbps
(1040, 1024)	64	5	10 x 128 B	65.59 KB	$1 \le n_{TL} \le 15$	40.08 Gbps
(2064, 2048)	128	3	10 x 256 B	38.06 KB	$1 \le n_{TL} \le 14$	49.70 Gbps
(2064, 2048)	128	4	10 x 256 B	71.60 KB	$1 \le n_{TL} \le 15$	49.46 Gbps
(2064, 2048)	128	5	10 x 256 B	133.52 KB	$1 \le n_{TL} \le 16$	49.20 Gbps

5. Conclusion

This paper proposed two types of integer codes capable of correcting burst asymmetric errors within a byte. The proposed codes are constructed with the help of a computer and are very efficient in terms of redundancy. The results of an exhaustive search have shown that, for practical data lengths up to 4096 bits, the proposed codes use up to two check-bit less than the corresponding codes of similar properties. Besides this, the proposed codes have the ability to be interleaved without delay and without using additional hardware. In this way, it is possible to construct simple codes capable of correcting (multiple) burst asymmetric errors. Such codes could be applied to various practical channels, especially to those that display asymmetric errors.

References

[1] J. R. Pierce, "Optical Channels: Practical Limits with Photon Counting," *IEEE Trans. Communications*, vol. 26, pp. 1819-1821, Dec. 1978.

[1] P. Oprisan and B. Bose, "ARQ in Optical Networks," *Proc. IEEE Int'l Symp. Pacific Rim Dependable Computing*, pp. 251-257, Dec. 2001

[2] E. L. Leiss, "Data integrity in digital optical disks," *IEEE Trans. Comput.*, vol. 33, no. 9, pp. 818-827, Sept. 1984.

[3] A. Radonjic and V. Vujicic, "Integer codes correcting burst errors within a byte," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 411-415, Feb. 2013.

[4] A. Radonjic *et al.*,"Integer codes correcting double asymmetric errors," *IET Commun.*, vol. 10, no. 14, pp. 1691-1696, Sep. 2016.

[5] A. Radonjic and V. Vujicic, "Integer codes correcting spotty byte asymmetric errors," *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2338-2341, Dec. 2016.

[6] A. Radonjic and V. Vujicic, "Integer codes correcting high-density byte asymmetric errors," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 694-697, Apr. 2017.

[7] A. Radonjic and V. Vujicic, "Integer codes correcting single errors and burst asymmetric errors within a byte," *Inform. Process. Lett.*, vol. 121, pp. 45-50, May 2017.

[8] A. Radonjic, "(Perfect) integer codes correcting single errors," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 17-20, Jan. 2018.

[9] A. Radonjic and V. Vujicic, "Integer codes correcting burst and random asymmetric errors within a byte," *J. Franklin Inst.*, vol. 355, no. 2, pp. 981-996, Jan. 2018.

[10] Y. Saitoh and H. Imai, "Some classes of burst asymmetric or unidirectional error correcting codes," *Elect. Lett.*, vo1. 26, no. 5, pp. 286-287, Mar. 1990.
[11] K. Mehlhorn and P. Sanders, *Algorithms and Data Structures: The Basic Toolbox*, Springer, 2008.
[12] https://ark.intel.com/products/81909/Intel-Xeon-Processor-E5-2687W-v3-25M-Cache-3_10-GHz.
[13] A. Fog, "The Microarchitecture of Intel, AMD and VIA CPUs," Technical University of Denmark, May 2017. Available from: http://www.agner.org/optimize/microarchitecture.pdf