Chapter

Integer Asymmetric Error Control Codes for Short-Range Optical Networks

Aleksandar Radonjic[*]
Institute of Technical Sciences of the Serbian Academy of Sciences and Arts,
Belgrade, Serbia
Vladimir Vujicic
Institute of Technical Sciences of the Serbian Academy of Sciences and Arts,
Belgrade, Serbia

ABSTRACT

In most communication networks, error probabilities 1 → 0 and 0 → 1 are equally likely to occur. However, in short-range optical networks (SRONs), such as local and access networks, this is not the case. In these networks, photons may fade or fail to be detected, but new photons cannot be generated. Hence, if the receiver operates correctly, only asymmetric (1 → 0) errors can occur. Motivated by this fact, the authors of this chapter have constructed four classes of integer codes capable of correcting various types of asymmetric errors. The most attractive feature of all these codes is their ability to be implemented "for free" (in software). This is achieved by using integer and lookup table operations, which are supported by all processors. The aim of this chapter is to overview four classes of integer asymmetric codes and to illustrate their potential for use in modern SRONs. Topics covered include: fundamentals in the design of integer codes, necessary and sufficient conditions for constructing integer asymmetric codes and the processor-based strategy for implementation of these codes.

Keywords: Integer codes, burst asymmetric errors, random asymmetric errors, short-range
optical networks, multicore processors, theoretical decoding throughput.

[*] Corresponding Author address
Email: sasa_radonjic@yahoo.com

1. Introduction

In optical networks (ONs) using on-off keying, the data are directly encoded into an optical signal: 1's are represented by the presence of light pulses (photons) and 0's by their absence [1]. When such a signal propagates through the fiber, it suffers from various degradition effects, such as attenuation and dispersion. These effects tend to reduce the number of photons ($1 \to 0$ errors) causing partial data loss.

In order to mitigate this problem, the engineers apply two strategies: one for long-range ONs (LRONs), and other for short-range ONs (SRONs). The strategy for LRONs is based on combined use of error control codes (ECCs) and optical amplifiers. After the transmitted signal is protected with ECCs and converted into light pulses, it is periodically strengthened with optical amplifiers. These devices regenerate the signal, but also add noise. So, if they are often used, the received signal may contain pulses at the zero time slots ($0 \to 1$ errors). Unlike in LRONs, the signals in SRONs are restored in the electrical domain. This is done using repeaters, which convert optical signals to electronic ones, amplify them, and again convert them to optical signals. Due to this reason, the regenerated signal is always identical or nearly identical to the original one.

With this in mind, in this chapter, we overview four classes of integer codes that are suitable for use in SRONs. The main advantage of these codes, compared to classical ECCs (CECCs), is their ability to exploit the high computing power of the network nodes (PCs, routers, switches, ONU units, etc.) [1]. This has been achieved by using integer and lookup table operations, which are supported by all processors. In addition, unlike CECCs, the presented codes can be interleaved without delay and without using dedicated hardware. Owing to this, they can be transformed into simple codes capable of correcting various combinations of asymmetric ($1 \to 0$) errors.

The organization of this chapter is as follows. Section 2 provides necessary and sufficient conditions for constructing four classes of integer codes capable of correcting multiple asymmetric errors. The implementation strategy and theoretical decoding throughputs for these codes are described and evaluated in Section 3, while Section 4 concludes the chapter.

2. Integer Error Control Codes

Unlike CECCs, integer ECCs (IECCs) [9], [10], [11], [12], are designed to correct errors of a given type. This means that we can choose certain types of errors and after that construct IECCs capable of correcting them. This characteristic can be deduced from their encoding and decoding procedures.

2.1. Encoding and Decoding Procedures

Let Z_{2^b-1} = {0, 1,…,2^b - 2} be the ring of integers modulo 2^b – 1 and let C_i be integers such that
$C_i \in Z_{2^b-1} \backslash \{0,1\}$, where 1 ≤ i ≤ k. Now, suppose that the data are divided into k b-bit bytes, and that B_i and $\underline{B}_i$ denote integer values of the i-th b-bit byte at the sender and receiver side, respectively. In that case, the encoder will compute the check-byte using the expression

$$C_B = [C_1 \cdot B_1 + C_2 \cdot B_2 + \cdots + C_k \cdot B_k] \,(\text{mod } 2^b - 1) = \sum_{i=1}^{k} C_i \cdot B_i \,(\text{mod } 2^b - 1)$$

(1)
At the receiver, the decoder will perform the same calculation

$$C_{\underline{B}} = [C_1 \cdot \underline{B}_1 + C_2 \cdot \underline{B}_2 + \cdots + C_k \cdot \underline{B}_k] \,(\text{mod } 2^b - 1) = \sum_{i=1}^{k} C_i \cdot \underline{B}_i \,(\text{mod } 2^b - 1)$$

(2)
after which the syndrome S will be formed

$$S = [C_{\underline{B}} - \underline{C}_B] \,(\text{mod } 2^b - 1) \equiv \sum_{i=1}^{k+1} (\underline{B}_i - B_i) \cdot C_i \,(\text{mod } 2^b - 1) = \sum_{i=1}^{k+1} e_i \cdot C_i \,(\text{mod } 2^b - 1)$$

(3)
where $C_{k+1} = -1$.
From (3) it is easy to see that the nonzero value of S indicates the presence of errors e_i. Whether they can be corrected or not depends on the values of the C_i's. In this chapter, we will see what conditions the C_i's must satisfy in order to construct codes capable of correcting burst asymmetric (BA) errors within a byte, random asymmetric (RA) errors within a byte, BA and RA errors within a byte and double asymmetric (DA) errors within a codeword.

2.2. Necessary and Sufficient Conditions for Constructing Asymmetric IECCs

As explained in the previous section, in modern SRONs, the number of received photons never exceeds the number of sent ones. Hence, if the receiver operates correctly, only asymmetric errors may occur [2], [3], [4]. The number and distribution of these errors depends on the nodes' distance and protocol design. Accordingly, it can be said that burst errors (up to five bits) are typical for passive networks, while random errors are dominant in local networks [5], [6], [7], [8]. Having this in mind, we start this subsection by defining the conditions for constructing IECCs capable of correcting t RA errors within a b-bit byte (integer $R_{t/b}$AEC codes). These codes are suitable for use in Ethernet networks using self-synchronous scramblers (e.g. 10GBASE-LR).

2.2.1. Integer $R_{t/b}$AEC Codes

Definition 1. An error is called a t/b RA error if within a b-bit there exist t asymmetric errors, where $1 \leq t < b$.

Table 1. Coefficients for Integer $R_{t/b}$AEC Codes with Parameters $t = 3$, $b = 32$ and $k \leq 64$.

2	15	31	71	83	89	101	119
127	139	141	143	149	157	163	167
173	177	179	181	189	191	199	203
211	223	227	229	233	239	251	253
263	269	271	277	281	283	305	307
313	317	331	339	349	353	359	361
367	373	379	383	389	395	397	401
409	421	431	433	443	463	465	467

Definition 2. The set of syndromes corresponding to t/b RA errors is defined as

$$s_1 = \left\{ -\left(2^{x_1} + 2^{x_2} + \cdots + 2^{x_m}\right) \cdot C_i \left(\bmod 2^b - 1\right) : 0 \leq x_1 < x_2 < \cdots < x_m \leq b-1, 1 \leq m \leq t, 1 \leq i \leq k+1 \right\}$$

(4)

With these definitions we can prove the following theorem.

Theorem 1. The codes defined by (1) can correct all t/b RA errors if there exist k mutually different coefficients $C_i \in Z_{2^b-1} \setminus \{0, 1\}$ such that

$$|s_1| = (k+1) \cdot \sum_{m=1}^{t} \binom{b}{m},$$

where $|s_1|$ denotes the cardinality of s_1.

Proof. From (4) it is clear that the set s_1 can be expressed as

$$s_1 = \bigcup_{u=1}^{k+1} X_u$$

where

$$X_1 = \left\{ \left[-\left(2^{x_1} + 2^{x_2} + \cdots + 2^{x_m}\right) \cdot C_1 \right] \left(\bmod 2^b - 1\right) : 0 \leq x_1 < x_2 < \cdots < x_m \leq b-1, 1 \leq m \leq t \right\}$$

$$\vdots$$

$$X_k = \left\{ \left[-\left(2^{x_1} + 2^{x_2} + \cdots + 2^{x_m}\right) \cdot C_k \right] \left(\bmod 2^b - 1\right) : 0 \leq x_1 < x_2 < \cdots < x_m \leq b-1, 1 \leq m \leq t \right\}$$

$$X_{k+1} = \left\{ 2^{x_1} + 2^{x_2} + \cdots + 2^{x_m} : 0 \leq x_1 < x_2 < \cdots < x_m \leq b-1, 1 \leq m \leq t \right\}$$

From this it is easy to see that the syndromes caused by t/b asymmetric errors will be nonzero and
mutually different only if there exist k different coefficients $C_i \in Z_{2^b-1} \setminus \{0,1\}$ such that

$$X_1 \cap \cdots \cap X_k \cap X_{k+1} = \varnothing$$

$$|X_1| = \cdots = |X_k| = |X_{k+1}|.$$

In that case, the set s_1 will have

$$|s_1| = \sum_{u=1}^{k+1} |X_u| = (k+1) \cdot \sum_{m=1}^{t} \binom{b}{m}$$

nonzero elements. □

In order to illustrate the applicability of Theorem 1, we show results of a computer-search for codes with parameters $t = 3$, $b = 32$ and $k \leq 64$ (Table 1).

2.2.2. Integer $B_{l/b}$AEC Codes

In some SRONs, such as passive optical networks (e.g. 10G PONs), the errors tend to occur in bursts. Hence, in these networks, it is preferable to use codes capable of correcting l-bit BA errors or l-bit BA errors in a b-bit byte (integer $B_{l/b}$AEC codes). The first step towards the construction of the latter codes is to define the integer values of BA errors confined to a b-bit byte. For that purpose, we will rely on the analysis from [9]. In that paper, it was shown that the integer value of a l-bit burst error within a b-bit byte is equal to $e = \pm 2^s \cdot (2n - 1)$, where $0 \le s \le b - l$, $1 \le n \le 2^{u-1}$ and $1 \le u \le l$. With this in mind, we can state the following definitions and theorem.

Definition 3. An error is called l/b BA error if within a b-bit there exists any number of asymmetric errors confined to l (< b) adjacent positions.

Definition 4. The set of syndromes corresponding to l/b BA errors is defined as

$$s_2 = \left\{ \left[-2^s \cdot (2n-1) \cdot C_i \right] \left(\bmod 2^b - 1 \right) : 0 \le s \le b - l, 1 \le n \le 2^{u-1}, 1 \le u \le l, 1 \le i \le k+1 \right\}$$

(5)

Theorem 2. The codes defined by (1) can correct all l/b BA errors if there exist k mutually different coefficients $C_i \in Z_{2^b-1} \setminus \{0,1\}$ such that $|s_2| = (k+1) \cdot \left[2^{l-1} \cdot (b - l + 2) - 1 \right]$.

Proof. Observe that the set s_2 can be expressed as

$$s_2 = \bigcup_{u=1}^{l} Y_u$$

where

$$Y_1 = \left\{ \left[-2^s \cdot (1) \cdot C_i \right] (\bmod 2^b - 1) : 0 \le s \le b-1,\ 1 \le i \le k+1 \right\},$$

$$Y_2 = \left\{ \left[-2^s \cdot (3) \cdot C_i \right] (\bmod 2^b - 1) : 0 \le s \le b-2,\ 1 \le i \le k+1 \right\},$$

$$\vdots$$

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

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

$$Y_1 \cap Y_2 \cap \cdots \cap Y_l = \varnothing,$$

$$|Y_1| = (k+1) \cdot b,$$

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

In that case, it is easy to show that

$$|s_2| = \sum_{u=1}^{l} |Y_u| = (k+1) \cdot \left[2^{l-1} \cdot (b - l + 2) - 1 \right]. \ \square$$

The next step in constructing codes capable of correcting l/b BA errors is to find the C_i's that satisfy the condition of Theorems 2. For that purpose it is necessary to perform an exhaustive search on all possible candidates from the set $Z_{2^b-1}\backslash\{0,1\}$. In this chapter, we have restricted ourselves to the codes with

Table 2. Coefficients for Integer $B_{l/b}$AEC Codes with Parameters $l = 5$, $b = 32$ and $k \le 64$.

2	33	35	37	41	43	47	53
59	61	67	71	73	79	83	89
97	101	103	107	109	113	117	127
131	137	139	149	151	157	163	167
173	179	181	191	193	197	199	211
223	227	229	233	239	241	251	255
257	263	269	271	277	281	283	293
307	311	313	317	331	337	343	347

parameters l = 5, b = 32 and k ≤ 64 (Table 2).

2.2.3. Integer $B_{l/b}$AEC-$R_{t/b}$AEC Codes

The previous classes of IECCs are designed to correct t/b RA or l/b BA errors. For this reason, they have limited practical applicability. One solution to overcome this drawback is to design codes with $B_{l/b}$AEC-$R_{t/b}$AEC capability (integer $B_{l/b}$AEC-$R_{t/b}$AEC codes). Such codes would have the potential to be used in both passive optical networks and Ethernet networks using self-synchronous

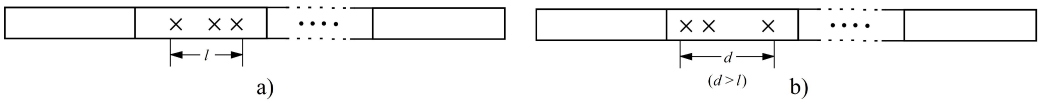

Fig. 1. Short *t/b* RA errors and (b) long *t/b* RA errors.

scramblers. However, before we move to the mathematical details, let us observe that t/b RA errors spaced less than l bits (short t/b RA errors) can be treated as l/b BA errors. Hence, in addition to l/b BA errors, we need to consider t/b RA errors that are spaced d (l < d < b) bits apart (long t/b RA errors) (Fig. 1).
Now, we can give the following definition.

Definition 5. Let 3 ≤ v < l < b and s + l ≤ z ≤ b – 1. Then, the set of syndromes corresponding to long t/b RA errors is defined as

$$s_3 = \varepsilon_1 \cup \varepsilon_2$$

(6)

where

$$\varepsilon_1 = \left\{\left[-(2^s+2^z)\cdot C_i\right]\left(\mathrm{mod}\ 2^b-1\right): 1 \le i \le k+1\right\}$$

(7)

$$\varepsilon_2 = \left\{\left[-(2^s+2^{x_1}+\cdots+2^{x_{v-2}}+2^z)\cdot C_i\right]\left(\mathrm{mod}\ 2^b-1\right): 0 \le s < x_1 < \cdots < x_{v-2} \le z, 1 \le i \le k+1\right\}$$

(8)

Although the expressions (5)-(8) provide a theoretical basis for the construction of integer $B_{l/b}AEC$-$R_{t/b}AEC$ codes, they do not give the explicit information about the number of nonzero syndromes. Hence, we need the following theorem.

Theorem 3. The codes defined by (1) can correct all l/b BA and t/b RA errors if there exist k mutually different coefficients $C_i \in Z_{2^b-1} \setminus \{0,1\}$ such that

1. $|s_2| = (k+1)\cdot\left[2^{l-1}\cdot(b-l+2)-1\right]$,

2. $|s_3| = (k+1)\cdot\sum_{i=0}^{b-l-1}\sum_{j=2}^{t}(b-l-i)\cdot\binom{l+i-1}{j-2}$,

3. $s_2 \cap s_3 = \varnothing$.

Proof. The proof for Condition 1 is the same as that given in Theorem 2. Hence, it will be omitted. As far as Condition 2 is concerned, it states that that long t/b RA errors generate $(k+1)\cdot\sum_{i=0}^{b-l-1}\sum_{j=2}^{t}(b-l-i)\cdot\binom{l+i-1}{j-2}$ syndromes that are nonzero. To prove this, note that the set s_3 can be expressed as

$$s_3 = \bigcup_{u=0}^{b-l-1} Z_u$$

where

$$Z_0 = \left\{\left[-(2^s+\underbrace{2^e+2^f+\cdots+2^p}_{t-2\text{ addends}}+2^{s+l})\cdot C_i\right](\text{mod } 2^b-1):\ 0\le s\le b-l-1,\ 1\le i\le k+1\right\}$$

$$Z_1 = \left\{\left[-(2^s+\underbrace{2^e+2^f+\cdots+2^p}_{t-2\text{ addends}}+2^{s+l+1})\cdot C_i\right](\text{mod } 2^b-1):\ 0\le s\le b-l-2,\ 1\le i\le k+1\right\}$$

$$\vdots$$

$$Z_{b-l-1} = \left\{\left[-(2^s+\underbrace{2^e+2^f+\cdots+2^p}_{t-2\text{ addends}}+2^{s+b-1})\cdot C_i\right](\text{mod } 2^b-1):\ s=0,\ 1\le i\le k+1\right\}$$

and $s < e < f < \cdots < p < s + l + u$ for any $u = 0, 1, \ldots, b-l-1$. Consequently, we can write

$$|Z_0| = (b-l)\cdot\binom{l-1}{t-2}\cdot(k+1)$$

$$|Z_1| = (b-l-1)\cdot\binom{l}{t-2}\cdot(k+1)$$

$$\vdots$$

$$|Z_{b-l-1}| = 1\cdot\binom{b-2}{t-2}\cdot(k+1)$$

wherefrom it follows that

$$|s_3| = \sum_{u=0}^{b-l-1}|Z_u| = (k+1)\cdot\sum_{i=0}^{b-l-1}\sum_{j=2}^{t}(b-l-i)\cdot\binom{l+i-1}{j-2}.$$

On the other hand, Condition 3 is a necessary condition for distinguishing l/b BA errors from long t/b RA errors. So, integer codes satisfying conditions 1 to 3 are $B_{l/b}$AEC-$R_{t/b}$AEC codes. □

Theorem 4. Let t = 3 and let $\xi_1 = s_2 \cup s_3$ be the error set for integer $B_{l/b}$AEC-$R_{t/b}$AEC codes. Then,

$$|\xi_1| = |s_2| + |s_3| = \left[2^{l-1} \cdot (b - l + 2) - 1 + \frac{(b-l)^2 + b - l}{2} \cdot b \right] \cdot (k+1) - \left[\frac{2 \cdot (b-l)^3 + 3 \cdot (b-l)^2 + b - l}{6} \right] \cdot (k+1).$$

Proof. This theorem follows directly from Theorem 4.

To illustrate the applicability of Theorem 4, we show results of a computer-search for the codes with parameters l = 5, t = 3, b = 32 and k ≤ 64 (Table 3).

2.2.4. Integer DAEC Codes

In Ethernet networks not using self-synchronous scramblers (e.g. 10GBASE-LX4), the errors are randomly distributed. For this reason, in these networks, it is desirable to protect the data with DA error correcting (DAEC) codes. These codes are able to correct three types of errors: single asymmetric errors, DA errors corrupting one b-bit byte and DA errors corrupting two b-bit bytes.

Definition 6. The set of syndromes corresponding to single asymmetric errors is defined as

$$s_4 = \left\{ \left(-2^s \cdot C_i \right) \left(\bmod 2^b - 1 \right) : 0 \le s \le b-1, 1 \le i \le k+1 \right\}$$

(9)

Definition 7. The set of syndromes corresponding to DA errors corrupting one b-bit byte is defined as

$$s_5 = \left\{ \left[(-2^r - 2^s) \cdot C_i \right] \left(\bmod 2^b - 1 \right) : 0 \le r < s \le b-1, 1 \le i \le k+1 \right\}$$

(10)

Definition 8. The set of syndromes corresponding to DA errors corrupting two b-bit bytes is defined as

$$s_6 = \left\{ \left(-2^r \cdot C_i - 2^s \cdot C_j \right) \left(\bmod 2^b - 1 \right) : 0 \le r, s \le b-1, 1 \le i < j \le k+1 \right\}$$

(11)

Now we are able to state the following theorem.

Theorem 5. The codes defined by (1) can correct all single asymmetric errors and all DA errors if there exist k mutually different coefficients $C_i \in Z_{2^b-1} \setminus \{0,1\}$ such that

Table 3. Coefficients for Integer $B_{l/b}$AEC-$R_{t/b}$AEC Codes with Parameters $l = 5$, $t = 3$, $b = 32$ and $k \le 64$.

2	45	71	83	89	101	119	127
139	141	143	149	157	159	163	167
173	177	179	181	191	199	211	223
227	229	233	237	239	251	263	269
271	277	281	283	301	395	307	313
317	331	339	349	353	361	367	373
379	383	389	397	401	409	421	431
433	443	453	463	467	479	487	499

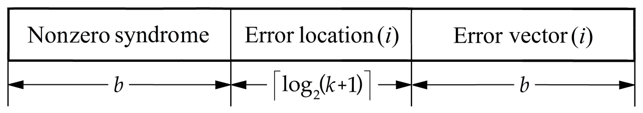

Fig. 2. Bit-width of one syndrome table entry in the case of IECCs correcting errors within one *b*-bit byte.

1. $|s_4| = (k+1) \cdot b$,

2. $|s_5| = (k+1) \cdot \frac{b \cdot (b-1)}{2}$,

3. $|s_6| = (k+1) \cdot \frac{b^2 \cdot k}{2}$,

4. $s_4 \cap s_5 \cap s_6 = \varnothing$

Proof. It can be easily proved that Conditions 1 is the necessary and sufficient condition for correcting single asymmetric errors, Condition 2 for correcting DA errors within a b-bit byte, and Condition 3 for correcting DA errors corrupting two b-bit bytes. Finally, Condition 4 ensures that the syndromes caused by single and DA errors are distinguishable. □

Theorem 6. Let $\xi_2 = s_4 \cup s_5 \cup s_6$ be the error set for integer DAEC codes. Then,

$$|\xi_2| = |s_4| + |s_5| + |s_6| = (k+1)^2 \cdot \frac{b^2}{2} + (k+1) \cdot \frac{b}{2}$$

Proof. This theorem follows from Theorem 5.

To illustrate the applicability of Theorem 5, we show results of a computer-search for the codes with parameters b = 32 and k ≤ 64 (Table 4).

2.3. Error Correction Procedure

After receiving the incoming packet, the decoder will generate the syndrome S. On the basis of its value, it will either accept the packet (S = 0) or try to recover the original data (S ≠ 0). In the latter case, the decoder will look up the syndrome

Table 4. Coefficients for Integer DAEC Codes with Parameters $b = 32$ and $k \leq 64$.

19	93	197	485	1111	1771	2247	2625
3923	4721	6459	7463	10941	13277	14329	15263
20183	21329	26267	31609	36597	40453	44993	47465
63449	65371	82467	86767	92563	95947	105581	108297
110849	131365	138807	150609	152801	177615	197791	203213
229677	246945	263135	278391	303707	305785	332895	341235
352515	388459	428269	435293	457423	478493	490611	504469
545203	571633	590825	595143	674535	715827	740695	742475

table to get the error correction data. In the case of IECCs described in Sections 2.2.1, 2.2.2 and 2.2.3, this table will have $E_1 = \{|s_1|, |s_2|, |\xi_1|\}$ entries (Theorems 1, 2 and 4), where each entry describes the relationship between the nonzero syndrome, error location and error vector (Definitions 2, 4 and 5) (Fig. 2).

So, when $S \neq 0$, the decoder's task is to find the entry with the first b bits as that of the syndrome S. If the syndrome table is sorted in increasing order (according to the values of S), this task will be completed after n_{TL} table lookups $(1 \le n_{\mathrm{TL}} \le \lfloor log_2 E_1 \rfloor + 2)$ [13]. In the next step, using the error correction data, the decoder will execute the operation

$B_i = \underline{B}_i + e_i \ (\mathrm{mod}\ 2^b - 1)$

(12)

where $1 \le i \le k + 1$ and where $e_i \in s_1$, $e_i \in s_2$ or $e_i \in \xi_1$. The similar procedure applies for IECCs described in Sections 2.2.4. In their case, the syndrome table has $E_2 = |\xi_2|$ entries (Theorem 6), where each entry occupies $3 \cdot b + 2 \cdot \lceil log_2(k+1) \rceil$ bits (Definitions 6, 7 and 8) (Fig. 3).

Accordingly, if the syndrome table is sorted in increasing order (according to the values of S), the decoder will perform n_{TL} table lookups $(1 \le n_{\mathrm{TL}} \le \lfloor log_2 E_2 \rfloor + 2)$ to find the entry where the first b bits matches that of the syndrome S. After that, it

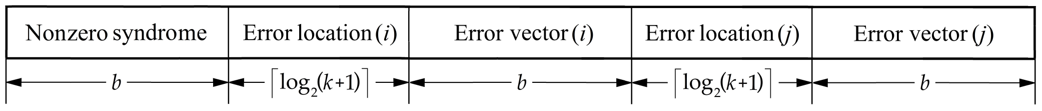

Fig. 3. Bit-width of one syndrome table entry in the case of IECCs correcting errors within two *b*-bit bytes.

will execute the operations

$B_i = \underline{B}_i + e_i \ (\mathrm{mod}\ 2^b - 1)$

(13)

$B_j = \underline{B}_j + e_j \ (\mathrm{mod}\ 2^b - 1)$

(14)

where $1 \le i < j \le k + 1$ and where 1) $e_i, e_j \in \xi_2$, 2) $e_i \in \xi_2$, $e_j = 0$, or 3) $e_i = 0$ and $e_j \in \xi_2$.

3. Evaluation and Implementation Strategy

From coding theory [14] it is known that practical implementation of CECCs is extremely expensive. For that purpose it is necessary to equip each node (PC, router, switch, OLT/ONU unit, etc.) with hardware that performs encoding and decoding operations. The software implementation is not feasible, since CECCs use Galois field arithmetic. This type of arithmetic entirely differs from the integer arithmetic of modern processors, which results in the fact that the software encoding and decoding of CECCs requires complex and time-consuming instructions.

Unlike CECCs, IECCs use integer and lookup table operations. Owing to this, they have the potential to be implemented "for free", i.e. without any hardware assistance. To illustrate this, in this section, we will consider two scenarios: the first one, where all nodes are equipped with quad-core processors and the second, where they are equipped with eight-core processors.

3.1. Scenario 1: Network nodes are equipped with quad-core processors

Suppose that all network nodes are equipped with quad-core processors (Fig. 4) having the following specifications [15], [16]:

1) clock rate: $C_{R4} = 3.5 \cdot 10^9$ Hz,

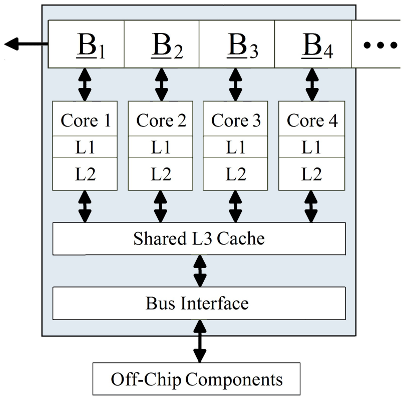

Fig. 4. Block diagram of quad-core processor.

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 (32 KB per core): 4 cycles latency,
8) access to the L2 cache (256 KB per core): 11 cycles latency,
9) access to the L3 cache (16 MB shared): 28 cycles latency.

In addition, suppose that the coefficients C_i are stored in each of the four L1 caches and that the syndrome table is placed into the L3 cache. In that case, instead of one, the decoder will (in parallel) generate four check-bytes:

- Core 1

$$C_{\underline{B}1} = [C_1 \cdot \underline{B}_1 + C_2 \cdot \underline{B}_5 + \cdots + C_k \cdot \underline{B}_{4\cdot(k-1)+1}] \pmod{2^{32}-1} = \sum_{i=1}^{k} C_i \cdot \underline{B}_{4\cdot(i-1)+1} \pmod{2^{32}-1} \quad (15)$$

- Core 2

$$C_{\underline{B}2} = [C_1 \cdot \underline{B}_2 + C_2 \cdot \underline{B}_6 + \cdots + C_k \cdot \underline{B}_{4\cdot(k-1)+2}] \pmod{2^{32}-1} = \sum_{i=1}^{k} C_i \cdot \underline{B}_{4\cdot(i-1)+2} \pmod{2^{32}-1}$$

(16)

- Core 3

$$C_{\underline{B}3} = [C_1 \cdot \underline{B}_3 + C_2 \cdot \underline{B}_7 + \cdots + C_k \cdot \underline{B}_{4\cdot(k-1)+3}] \ (\mathrm{mod}\ 2^{32}-1) = \sum_{i=1}^{k} C_i \cdot \underline{B}_{4\cdot(i-1)+3} \ (\mathrm{mod}\ 2^{32}-1)$$

(17)

- Core 4

$$C_{\underline{B}4} = [C_1 \cdot \underline{B}_4 + C_2 \cdot \underline{B}_8 + \cdots + C_k \cdot \underline{B}_{4\cdot(k-1)+4}] \ (\mathrm{mod}\ 2^{32}-1) = \sum_{i=1}^{k} C_i \cdot \underline{B}_{4\cdot(i-1)+4} \ (\mathrm{mod}\ 2^{32}-1)$$

(18)

If we add to this K/128 = 4·k·b/128 = k shift operations (K - the number of data bits) we easily calculate that each processing core requires T_1 = 9·k clock cycles (k accesses to the L1 cache, k integer multiplications, k - 1 integer additions, 1 modulo reduction and k shift operations) to compute the value of the check-byte. After finishing this task, each core will take T_2 = 2 clock cycles (1 integer subtraction and 1 modulo reduction) to perform the following operations:

- Core 1

$$S_1 = [C_{\underline{B}1} - \underline{C}_{B1}] \ (\mathrm{mod}\ 2^{32}-1)$$

(19)

- Core 2

$$S_2 = [C_{\underline{B}2} - \underline{C}_{B2}] \ (\mathrm{mod}\ 2^{32}-1)$$

(20)

- Core 3

$$S_3 = [C_{\underline{B}3} - \underline{C}_{B3}] \ (\mathrm{mod}\ 2^{32}-1)$$

(21)

- Core 4

$$S_4 = [C_{\underline{B}4} - \underline{C}_{B4}] \ (\mathrm{mod}\ 2^{32}-1)$$

(22)

As explained in Section 2.3, if the data are received in error, the decoder will additionally perform n_{TL} table lookups, n_{TL} comparisons, 1 integer addition (or two integer additions in parallel) and 1 modulo reduction (or two modulo reductions in parallel). In our case, four such operations will be executed in parallel in T_3 = 29·n_{TL} + 2 clock cycles. Hence, if we sum up all processing times, we come to the conclusion that the processor requires

$$\mathrm{T}_4 = \mathrm{T}_1 + \mathrm{T}_2 + \mathrm{T}_3 = 9 \cdot k + 29 \cdot n_{\mathrm{TL}} + 4$$

(23)

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

$$G_4 = \frac{C_{\mathrm{R}4}}{\mathrm{T}_4/K} = \frac{(3.5 \cdot 10^9) \cdot 128 \cdot k}{9 \cdot k + 29 \cdot n_{\mathrm{TL}} + 4}$$

(24)

data bits. By substituting the values of k and n_{TL} in (24) we obtain that G_{min} = 15.91 Gbps and G_{max} = 27.46 Gbps. In other words, it can be concluded that all codes from Table 5, except (2080, 2048) DAEC code, have the potential to be used in 10G networks [1]. In addition, from (15)-(22) we see that the analyzed codes can correct asymmetric errors affecting four adjacent 32-bit bytes. This feature is very useful, since in practice channel errors often corrupt two adjacent bytes [5], [6], [7].

3.2. Scenario 2: Network nodes are equipped with eight-core processors

Suppose now that the network nodes are equipped with eight-core processors (Fig. 5) having the following specifications [17], [18]:

1) clock rate: $C_{R8} = 3.5\cdot 10^9$ Hz,
2) integer addition/subtraction/multiplication operation: 2 cycles latency,
3) 128-bit shift operation: 1 cycle latency,
4) modulo reduction operation: 1 cycle latency,
5) comparison operation: 1 cycle latency,
6) access to the L1 cache (64 KB per core): 3 cycles latency,

Table 5. Memory Requirements and Theoretical Decoding Throughputs for Some Four-Byte Interleaved Integer Codes.

Integer Codes	k	Memory Requirements for Storing the Coefficients C_i	Memory Requirements for Storing the Syndrome Table	Number of Table Lookups	Minimum Theoretical Decoding Throughput
(1056, 1024) $B_{5/32}$AEC Code	32	4 x 128 B	0.14 MB	$1 \le n_{TL} \le 15$	19.72 Gbps
(1056, 1024) $R_{3/32}$AEC Code	32	4 x 128 B	1.58 MB	$1 \le n_{TL} \le 19$	17.01 Gbps
(1056, 1032) $B_{5/32}$AEC-$R_{3/32}$AEC Code	32	4 x 128 B	1.63 MB	$1 \le n_{TL} \le 19$	17.01 Gbps
(1056, 1032) DAEC Code	32	4 x 128 B	7.53 MB	$1 \le n_{TL} \le 21$	15.91 Gbps
(2080, 2048) $B_{5/32}$AEC Code	64	4 x 256 B	0.27 MB	$1 \le n_{TL} \le 16$	27.46 Gbps
(2080, 2048) $R_{3/32}$AEC Code	64	4 x 256 B	3.17 MB	$1 \le n_{TL} \le 20$	24.72 Gbps
(2080, 2048) $B_{5/32}$AEC-$R_{3/32}$AEC Code	64	4 x 256 B	3.25 MB	$1 \le n_{TL} \le 20$	24.72 Gbps
(2080, 2048) DAEC Code	64	4 x 256 B	29.76 MB	$1 \le n_{TL} \le 23$	--------[1]

[1]The size of the syndrome table exceeds the capacity of the cache.

7) access to the L2 cache (256 KB per core): 8 cycles latency,
8) access to the L3 cache (32 MB shared): 25 cycles latency.

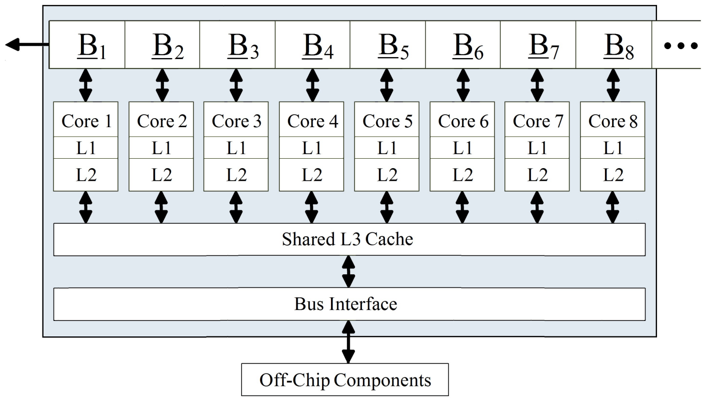

Fig. 5. Block diagram of eight-core processor.

Given this, suppose that the data word has K = 8·b·k = 256·k bits and that the coefficients C_i are stored in each of the eight L1 caches. In addition, assume that that the syndrome table is placed into the L3 cache. In that case, instead of one, we will have eight check-bytes. Their values are calculated as follows:

- Core 1

$$C_{\underline{B}1} = [C_1\cdot\underline{B}_1 + C_2\cdot\underline{B}_9 + \cdots + C_k\cdot\underline{B}_{8\cdot(k-1)+1}] \pmod{2^{32}-1} = \sum_{i=1}^{k} C_i\cdot\underline{B}_{8\cdot(i-1)+1} \pmod{2^{32}-1}$$

(25)

- Core 2

$$C_{\underline{B}2} = [C_1\cdot\underline{B}_2 + C_2\cdot\underline{B}_{10} + \cdots + C_k\cdot\underline{B}_{8\cdot(k-1)+2}] \pmod{2^{32}-1} = \sum_{i=1}^{k} C_i\cdot\underline{B}_{8\cdot(i-1)+2} \pmod{2^{32}-1}$$

(26)

- Core 3

$$C_{\underline{B}3} = [C_1\cdot\underline{B}_3 + C_2\cdot\underline{B}_{11} + \cdots + C_k\cdot\underline{B}_{8\cdot(k-1)+3}] \pmod{2^{32}-1} = \sum_{i=1}^{k} C_i\cdot\underline{B}_{8\cdot(i-1)+3} \pmod{2^{32}-1}$$

(27)

- Core 4

$$C_{\underline{B}4} = [C_1\cdot\underline{B}_4 + C_2\cdot\underline{B}_{12} + \cdots + C_k\cdot\underline{B}_{8\cdot(k-1)+4}] \pmod{2^{32}-1} = \sum_{i=1}^{k} C_i\cdot\underline{B}_{8\cdot(i-1)+4} \pmod{2^{32}-1}$$

(28)

- Core 5

$$C_{\underline{B}5} = [C_1\cdot\underline{B}_5 + C_2\cdot\underline{B}_{13} + \cdots + C_k\cdot\underline{B}_{8\cdot(k-1)+5}] \pmod{2^{32}-1} = \sum_{i=1}^{k} C_i\cdot\underline{B}_{8\cdot(i-1)+5} \pmod{2^{32}-1}$$

(29)

- Core 6

$$C_{\underline{B}6} = [C_1\cdot\underline{B}_6 + C_2\cdot\underline{B}_{14} + \cdots + C_k\cdot\underline{B}_{8\cdot(k-1)+6}] \pmod{2^{32}-1} = \sum_{i=1}^{k} C_i\cdot\underline{B}_{8\cdot(i-1)+6} \pmod{2^{32}-1}$$

(30)

- Core 7

$$C_{\underline{B}7} = [C_1 \cdot \underline{B}_7 + C_2 \cdot \underline{B}_{15} + \cdots + C_k \cdot \underline{B}_{8\cdot(k-1)+7}] \pmod{2^{32}-1} = \sum_{i=1}^{k} C_i \cdot \underline{B}_{8\cdot(i-1)+7} \pmod{2^{32}-1}$$

(31)

- Core 8

$$C_{\underline{B}8} = [C_1 \cdot \underline{B}_8 + C_2 \cdot \underline{B}_{16} + \cdots + C_k \cdot \underline{B}_{8\cdot(k-1)+8}] \pmod{2^{32}-1} = \sum_{i=1}^{k} C_i \cdot \underline{B}_{8\cdot(i-1)+8} \pmod{2^{32}-1}$$

(32)

If we add to this K/128 = 2·k shift operations, we calculate that each processor requires T_5 = 9·k clock cycles (k accesses to the L1 cache, k integer multiplications, k - 1 integer additions, 1 modulo reduction and 2·k shift operations) to compute the check-byte. After finishing this task, the processor will take T_6 = 3 clock cycles (1 integer subtraction and 1 modulo reduction) to generate the syndromes

- Core 1

$$S_1 = [C_{\underline{B}1} - \underline{C}_{B1}] \pmod{2^{32}-1}$$

(33)

- Core 2

$$S_2 = [C_{\underline{B}2} - \underline{C}_{B2}] \pmod{2^{32}-1}$$

(34)

- Core 3

$$S_3 = [C_{\underline{B}3} - \underline{C}_{B3}] \pmod{2^{32}-1}$$

(35)

- Core 4

$$S_4 = [C_{\underline{B}4} - \underline{C}_{B4}] \pmod{2^{32}-1}$$

(36)

- Core 5

$$S_5 = [C_{\underline{B}5} - \underline{C}_{B5}] \pmod{2^{32}-1}$$

(37)

- Core 6

$$S_6 = [C_{\underline{B}6} - \underline{C}_{B6}] \pmod{2^{32}-1}$$

(38)

- Core 7

$$S_7 = [C_{\underline{B}7} - \underline{C}_{B7}] \pmod{2^{32}-1}$$

(39)

- Core 8

$$S_8 = [C_{\underline{B}8} - \underline{C}_{B8}] \pmod{2^{32}-1}$$

(40)

As in the previous scenario, if the data are received in error, the decoder will perform n_{TL} table lookups, n_{TL} comparisons, 1 integer addition (or two integer additions in parallel) and 1 modulo reduction (or two modulo reductions in parallel). In this particular case, eight such operations will be executed in parallel $T_7 = 26 \cdot n_{TL} + 6$ clock cycles. So, if we sum up all processing times, we come to the conclusion that the processor requires

$$T_8 = T_5 + T_6 + T_7 = 9 \cdot k + 26 \cdot n_{TL} + 9$$

(41)

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

$$G_8 = \frac{C_{R8}}{T_8/K} = \frac{(3.5 \cdot 10^9) \cdot 256 \cdot k}{9 \cdot k + 26 \cdot n_{TL} + 9}$$

(42)

data bits. By substituting the values of k and n_{TL} in (42) we easily calculate that G_{min} = 34.01 Gbps and $G_{max.}$ = 57.29 Gbps (Table 6). In this scenario, unlike the previous one, all considered codes (Table 6) have the potential to be used in 10G networks. Moreover, we see that most of them could be candidates for use in 40G networks. In addition, unlike the previous scenario, in this one, the data bytes are interleaved to a depth of eight. This additionally reduces the probability that some errors may be undetected or miscorrected.

4. Conclusion

In this chapter, we have reviewed four classes of integer codes capable of correcting multiple asymmetric errors. We have shown that these codes have two important characteristics: first, they use processor-friendly operations, and second, they can be interleaved without delay and without using dedicated hardware. Thanks to these, they have potential to be implemented "for free" in short-range optical networks. To illustrate this, we have shown that the four-byte and eight-byte interleaved codes, implemented on the four-core and eigth-core processor, respectively, achieve theoretical throughputs of several tens of Gbps. In the future, we plan to extend our approach to codes capable of correcting symmetric errors. Unlike the presented ones, these codes would have the potential to be used in long-range optical networks.

Table 6. Memory Requirements and Theoretical Decoding Throughputs for Some Eight-Byte Interleaved Integer Codes.

Integer Codes	k	Memory Requirements for Storing the Coefficients C_i	Memory Requirements for Storing the Syndrome Table	Number of Table Lookups	Minimum Theoretical Decoding Throughput
(1056, 1024) $B_{5/32}$AEC Code	32	8 x 128 B	0.14 MB	$1 \le n_{TL} \le 15$	41.74 Gbps
(1056, 1024) $R_{3/32}$AEC Code	32	8 x 128 B	1.58 MB	$1 \le n_{TL} \le 19$	36.25 Gbps
(1056, 1032) $B_{5/32}$AEC-$R_{3/32}$AEC Code	32	8 x 128 B	1.63 MB	$1 \le n_{TL} \le 19$	36.25 Gbps
(1056, 1032) DAEC Code	32	8 x 128 B	7.53 MB	$1 \le n_{TL} \le 21$	34.01 Gbps
(2080, 2048) $B_{5/32}$AEC Code	64	8 x 256 B	0.27 MB	$1 \le n_{TL} \le 16$	57.29 Gbps
(2080, 2048) $R_{3/32}$AEC Code	64	8 x 256 B	3.17 MB	$1 \le n_{TL} \le 20$	51.90 Gbps
(2080, 2048) $B_{5/32}$AEC-$R_{3/32}$AEC Code	64	8 x 256 B	3.25 MB	$1 \le n_{TL} \le 20$	51.90 Gbps
(2080, 2048) DAEC Code	64	8 x 256 B	29.76 MB	$1 \le n_{TL} \le 23$	48.47 Gbps

.

REFERENCES

[1] R. Ramaswani, K. Sivarajan and G. Sasaki, Optical Networks: A Practical Perspective, 3rd ed., Elsevier, Inc., 2010.

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

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

[4] S. Elmougy, "Some Contributions to Asymmetric Error Control Codes," PhD thesis, Oregon State Univ., 2005.

[5] D. Mello, E. Offer and J. Reichert, "Error Arrival Statistics for FEC Design in Four-Wave Mixing Limited Systems," Proc. Optical Fiber Communications Conference, pp. 529-530, Mar. 2003.

[6] L. James, "Error Behaviour in Optical Networks," PhD thesis, Univ. Cambridge, 2005.

[7] P. Anslow and O. Ishida, "Error Distribution in Optical Links," IEEE 802.3 HSSG Interim Meeting, Nov. 2007.

[8] K. Cho et al., "Performance of Forward-Error Correction Code in 10-Gb/s RSOA-Based WDM PON," IEEE Photon. Technology Letters, vol. 22, no. 1, pp. 57-59, Jan. 2010.

[9] A. Radonjic and V. Vujicic, "Integer Codes Correcting Burst Errors within a Byte," IEEE Trans. Computers, vol. 62, no. 2, pp. 411-415, Feb. 2013.

[10] A. Radonjic et al.,"Integer Codes Correcting Double Asymmetric Errors," IET Commun., vol. 10, no. 14, pp. 1691-1696, Sep. 2016.

[11] A. Radonjic and V. Vujicic, "Integer Codes Correcting Spotty Byte Asymmetric Errors," IEEE Commun. Lett., vol. 20, no. 12, pp. 2338-2341, Dec. 2016.

[12] 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.

[13] K. Mehlhornand and P. Sanders, Algorithms and Data Structures: The Basic Toolbox, Springer, 2008.

[14] E. Fujiwara, Code Design for Dependable Systems: Theory and Practical Application, John Wiley & Sons, Inc., 2006.

[15] J. Hennessy and D. Patterson, Computer Architecture: A Quantitative Approach, 5th ed., Elsevier, Inc., 2012.

[16] A. Fog, "The Microarchitecture of Intel, AMD and VIA CPUs: An Optimization Guide for Assembly Programmers and Compiler Makers," Tech. Univ. Denmark, Feb. 2017.

[17] L. Johnsson, "Introduction to HPC architecture," Dept. Computer Sciences, Univ. Houston, Jan. 2014.

[18] http://www.7-cpu.com/cpu/Power7.html