Modified Extended Integrated Interleaved Codes

Yok Jye Tang and Xinmiao Zhang

***Abstract*—Extended integrated interleaved (EII) codes have hyper-speed decoding and good correction capability. They are among the best candidates for the next-generation digital communications and storage. However, in existing EII schemes, parities are allocated to correct a long burst of errors and limit the correction capability of random errors. In this paper, a modified EII code is proposed to re-allocate parities for correcting burst errors to form column-wise codes across individual sub-codewords. The overall decoding process incorporating the column-wise decoding is developed to reduce the decoding failure rate. Besides, the allocations of the column-wise parities are optimized. Miscorrections in column-wise decoding are analyzed. Compared to previous EII schemes with the same redundancy, our design achieves a good trade-off between the burst and random error correction capabilities with small overheads on the EII encoding and decoding processes. For channels that generate random errors, the proposed EII code can achieve orders of magnitude lower frame error rates than the previous EII code and its best alternative code.**

***Index Terms*—BCH codes, column-wise codes, error-correcting codes, extended integrated interleaved codes, generalized integrated interleaved codes, Reed-Solomon codes.**

I. Introduction

Next-generation digital communication and storage systems require error-correcting codes with hyper-speed decoding and good correction capability. Extended integrated interleaved (EII) codes [1], [2] and its special case, generalized integrated interleaved (GII) codes [3], [4], that nest Reed-Solomon (RS) or BCH sub-codewords to generate codewords of stronger codes are among the best candidates. The decoding of these codes consists of two stages. The first stage is traditional RS or BCH decoding on individual sub-words. The second-stage nested decoding is activated when any of the sub-words has extra errors. Hardware architectures for implementing GII encoding and decoding explored in [5]–[10] can be easily extended to the cases of EII codes to achieve hundred of gigabits throughput. Although EII codes can also be constructed as in [11], [12], their decoding process involves iterative re-encoding, which results in very long latency.

Unlike GII codes, EII codes allocate more parities to correct a long burst of errors in a sub-word. This limits the amount of redundancy that can be utilized to correct errors from other sub-words. For many practical applications, such as flash memory and optical communications, the errors are mostly random [4], [12]. For channels with random errors, the probability of a sub-word experiencing a long burst of errors is lower than that of having many sub-words with relatively fewer errors. In such channels, it turns out that EII codes with the same amount of redundancy have worse error-correcting performance compared to GII codes.

This paper proposes a modified EII code that re-allocates parities for correcting a long burst of errors to form column-wise codes across individual sub-codewords. These column-wise codes can be utilized to correct errors from sub-words such that the nested decoding process can continue to correct more errors. The overall modified EII decoding process incorporating column-wise decoding is developed to reduce the decoding failure rate. The number of column-wise codewords and their correction capabilities can vary. Different settings are analyzed to decide the optimal design. Also miscorrections in column-wise decoding are investigated and a method is proposed to identify most of the miscorrections by keeping track of whether a sub-word is corrected or not. Without increasing the overall redundancy, the proposed EII scheme achieves a good trade-off between the burst and random error correction capabilities with small complexity overheads on the encoding and decoding processes. For channels that generate random errors, the proposed modified EII code can achieve orders of magnitude lower decoding frame error rates than previous EII and GII codes with the same redundancy.

II. EII codes and decoding

A $([m, v], n)$ EII or GII codeword is divided into m sub-codewords, $c_0, c_1, \cdots, c_{m-1}$, each of which is a codeword of $\mathcal{C}_0$ Reed-Solomon (RS) or BCH code with length n. Linear combinations of the m sub-codewords generate nested codewords, which belong to stronger codes, $\mathcal{C}_v \subset \mathcal{C}_{v-1} \subset \cdots \subset \mathcal{C}_1$. Let $s_j (1 \leq j \leq v)$ denote the number of nested codewords belonging to $\mathcal{C}_j$ and $\hat{s}_j = \sum_{i=j}^{v} s_i$. Also define $\hat{s}_{v+1} = 0$. EII codes [1] can be defined as follows

$$\begin{aligned} \mathcal{C} \triangleq & \{c = [c_0, \cdots, c_{m-1}] : c_i \in \mathcal{C}_0, \\ & \tilde{c}_l = \sum_{i=0}^{m-1} \beta^{li} c_i \in \mathcal{C}_j, 1 \leq j \leq v, \hat{s}_{j+1} \leq l \leq \hat{s}_j - 1\}, \end{aligned} \tag{1}$$

where $\tilde{c}_l$ is the nested codeword and β is a primitive element of $GF(2^q)$. For EII codes, a codeword of $\mathcal{C}_v$ is an all-zero vector [1]. When $s_v = 0$, an EII code reduces to a GII code.

A sub-codeword received by the decoder that may contain errors is called a 'sub-word' in this paper. Let the error-correcting capabilities of $\mathcal{C}_v, \cdots, \mathcal{C}_1, \mathcal{C}_0$ be $t_v > \cdots > t_1 > t_0$, respectively. The decoding process of EII codes consists of two stages: 1) sub-word decoding and 2) nested decoding. The sub-word decoding is traditional RS or BCH decoding carried out on individual sub-words to correct up to t_0 errors. The second-stage nested decoding is activated when any of the m sub-words has more than t_0 errors. The nested decoding has up to v rounds. Let $b_\eta \leq \hat{s}_\eta$ denote the number of erroneous sub-words in the beginning of round η. In the η-th $(1 \leq \eta < v)$ round,

The authors are with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: tang.1121@osu.edu and zhang.8952@osu.edu). This material is based upon work supported by the National Science Foundation under Award No. 2011785.

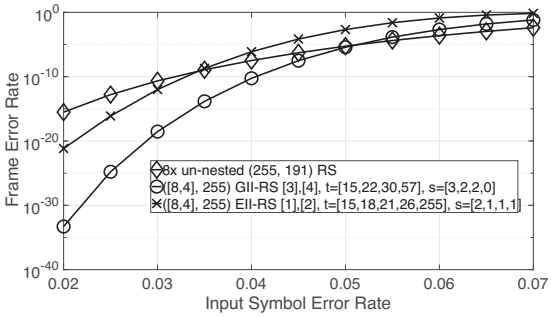

Fig. 1. FERs of un-nested $(255, 191)$ RS codes, the $([8, 4], 255)$ GII-RS code with $t = [15, 22, 30, 57]$ and $s = [3, 2, 2, 0]$, and the $([8, 4], 255)$ EII-RS code with $t = [15, 18, 21, 26, 255]$ and $s = [2, 1, 1, 1]$ over $GF(2^8)$ with random input errors.

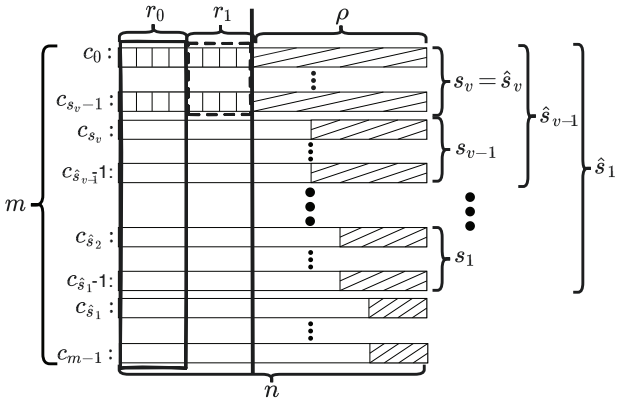

Fig. 2. The proposed $([m, v], n)$ EII code with the column-wise code.

the $2(t_\eta - t_{\eta-1})$ higher-order syndromes of the first b_η nested words are computed. Then the higher-order syndromes for those b_η sub-words with extra errors are derived by inverting the linear combinations in (1) and the t_η-error-correcting RS or BCH decoding is carried out afterward. As a result, up to t_η errors can be corrected in at most $\hat{s}_\eta$ sub-words. If some sub-words are still uncorrected, they will be sent to the next nested decoding round and a similar process is repeated.

For EII codes, the last nested decoding round is different since the codewords of $\mathcal{C}_v$ are all-zero vectors. In this case, the s_v linear combinations of the i-th $(0 \le i < n)$ symbols from the m sub-words are zero. Up to s_v errors in the i-th symbols can be corrected by solving these linear combinations. As a result, up to n errors can be corrected from at most s_v sub-words in this round. EII codes can correct a long burst of errors. However, for channels with random errors, a sub-word is less likely to have a long burst of errors. Most of the time, EII decoding fails because the number of sub-words that remain uncorrected at the end of round η is larger than $\hat{s}_{\eta+1}$. Unlike the $([m, v], n)$ EII codes, $([m, v], n)$ GII codes have all parities allocated to correct errors among the sub-words. Hence, for the same amount of overall redundancy, GII codes can employ more powerful $\mathcal{C}_1, \cdots, \mathcal{C}_{v-1}$, and achieve better performance over random-error channels.

For EII and GII codes with the same codeword length, the error-correcting performance increases for smaller m, while the decoder throughput and complexity improve with larger m. Besides, larger v leads to a lower overall decoding failure rate but significantly increases the worst-case decoding latency. To achieve tradeoffs on overall decoding failure rate, decoder throughput, hardware complexity, and worst-case decoding latency, example EII and GII codes for Flash memories with 2k-byte codeword length, $m = 8$, $n = 255$, and $v = 4$ are considered in this paper.

Fig. 1 plots the frame error rate (FER) of a $([8, 4], 255)$ EII-RS code over $GF(2^8)$ with 75% code rate for channel with random errors. The code rate of the EII/GII-RS codes can be calculated as $1 - ((m - \hat{s}_1)2t_0 + s_1 2t_1 + \cdots + s_{v-1}2t_{v-1} + s_v n)/(mn)$. In our simulations, each symbol has equal probability of being erroneous. If a symbol is erroneous, one of the rest 255 symbols in $GF(2^8)$ is randomly chosen to be the corresponding received symbol. It was found by simulations that $t = [t_0, t_1, t_2, t_3, t_4] = [15, 18, 21, 26, 255]$ and $s = [s_1, s_2, s_3, s_4] = [2, 1, 1, 1]$ lead to the lowest FER compared to other EII-RS codes with the same m, v, n, and code rate but different t and s. Fig. 1 also includes the FER of the $([8, 4], 255)$ GII-RS code with $t = [15, 22, 30, 57]$ and $s = [s_1, s_2, s_3, s_4] = [3, 2, 2, 0]$. The decoder of such a GII-RS code with $t_0 = 15$ can achieve similar throughput as the EII-RS decoder. It has lower FER than other $([8, 4], 255)$ GII-RS decoders with the same code rate and t_0 but different $t_1, \cdots, t_{v-1}$, and s. With the same code rate, this GII code has 3+2+2=7 instead of 2+1+1+1=5 nested codewords as in the EII code, and it has more powerful $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$. Fig. 1 shows that this GII code has lower FER than the EII code. For comparison, the FER of eight un-nested $(255, 191)$ RS codes is plotted in Fig. 1. It has the same code rate as the EII and GII codes and can correct up to 32 errors. Compared to such un-nested RS codes, the EII and GII codes have their FERs decreasing at a faster pace with the input symbol error rate and can achieve almost twice the decoding throughput since the 15-error-correcting sub-word decoding is carried out most of the time for lower symbol error rates.

III. Modified EII codes

This section presents a modified EII code that achieves better error-correcting performance over channels with random errors. Unlike previous EII codes, the parities for the all-zero codewords are re-allocated in the proposed scheme to form column-wise codes across individual sub-codewords. This allows a trade-off between burst and random error correction. Additionally, these column-wise codes can help to reduce the number of uncorrected sub-words in nested decoding round $\eta - 1$ to $\hat{s}_\eta$ so that the nested decoding can continue to round η to correct more errors.

Fig. 2 shows the layout of a $([m, v], n)$ EII codeword. The m sub-codewords, $c_0, c_1, \cdots, c_{m-1}$, are represented by the rectangles from top to bottom. A systematic GII encoding process has been proposed in [4] and it can be directly extended for EII codes. In Fig. 2, the blank parts of the rectangles denote data and the shaded parts represent parities. In [4], the parities in the sub-words are calculated through reversing the linear combinations used to define the nested codewords as in (1). Assuming that a codeword of $\mathcal{C}_j (0 \le j \le v)$ has w_j parities, the number of parities in c_i equals w_v for $0 \le i \le s_v - 1$, $w_j (1 \le j < v)$ for $\hat{s}_{j+1} \le i \le \hat{s}_j - 1$, and w_0 for $\hat{s}_1 \le i < m$.

In previous EII codes, all the symbols in the first s_v sub-codewords are parities. To achieve a good trade-off between

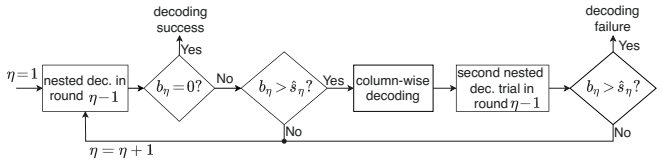

Fig. 3. The decoding process of the proposed EII code.

the burst and random error correction capabilities, the proposed scheme utilizes an RS or BCH code instead of all-zero code as $\mathcal{C}_v$. Assume that a codeword of this $\mathcal{C}_v$ has ρ parities. The rest $n-\rho$ parities from each of the first s_v sub-codewords are used to form column-wise codewords across individual sub-codewords as shown in Fig. 2. Since each column has an equal probability of containing errors, the proposed scheme selects the first $1 \leq r_0 \leq (n-\rho)$ columns to form column-wise codewords as shown in Fig. 2 to simplify the decoder. When $r_0 < n-\rho$, the parities in the rest $r_1 = n-\rho-r_0$ columns are evenly distributed as much as possible as extra parities to the r_0 column-wise codewords. The optimal choices of r_0, r_1 and t_v will be discussed in the next section.

In EII decoding, the nested decoding can not continue to round η when b_η is larger than $\hat{s}_\eta$, which is the maximum number of sub-words that can be corrected by nested decoding round η. Assuming that nested decoding round 0 is the sub-word decoding, Fig. 3 summarizes the overall decoding flow of the proposed EII codes that incorporate column-wise codes. The decoding of column-wise codes is activated when $b_\eta > \hat{s}_\eta$. If some erroneous symbols get corrected by the column-wise decoding, the second nested decoding trial in round $\eta-1$ may correct more sub-words and make $b_\eta \leq \hat{s}_\eta$, such that the next nested decoding round can continue. For the later nested decoding rounds, the decoding of those corrected columns can be skipped to reduce latency.

Since RS codes are maximum distance separable (MDS) codes, they are used as the column-wise code in our scheme. The decoding of the column-wise code is just traditional RS decoding. When $r_1 \neq r_0$, the r_0 column-wise codewords have different number of parities. Assume that the codeword for the j-th $(0 \leq j < r_0)$ column has τ_j parities including those from the r_1 columns. These parities are distributed across different sub-codewords and more than one of them may be located in the same sub-word. If the number of symbols in the b_η erroneous sub-words contributing to the j-th column-wise codeword does not exceed τ_j, they can be corrected by erasure-only decoding. Otherwise, error-correcting decoding is carried out to correct up to $\lfloor \tau_j/2 \rfloor$ errors. This may still correct some errors since not every symbol in an erroneous sub-word is incorrect.

For the encoding of the proposed EII code, the parities of the column-wise code are computed first by carrying out the traditional RS encoding. Then the rest parities can be generated by the same systematic GII encoding process in [4] by treating the parities of the column-wise codewords as the data parts of $c_0, \cdots, c_{s_v-1}$.

IV. Optimizations of Parities for Column-Wise Code

This section uses the $([8,4],255)$ EII-RS code with $t = [t_0,t_1,t_2,t_3,t_4] = [15,18,21,26,t_4]$, $s = [s_1,s_2,s_3,s_4] = [2,1,1,1]$ as an example to explain the optimizations of t_v, r_0 and r_1 values.

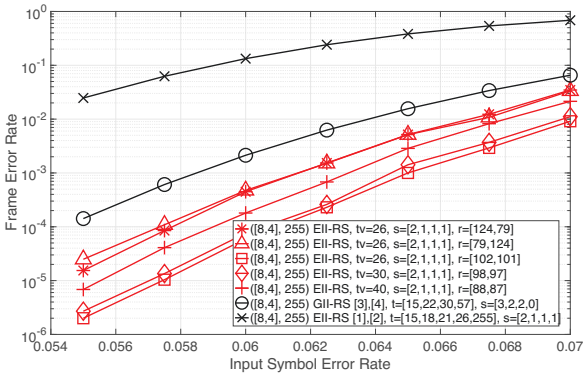

Fig. 4. FERs of the proposed $([8,4],255)$ EII-RS codes with $t = [15,18,21,26,t_4]$, $s=[2,2,1,1]$ and $r=[r_0,r_1]$ with random input errors.

In the proposed modified EII code, r_0 decides the number of columns that are protected by the column-wise code, and r_1 is the number of columns that store the extra parities as shown in Fig. 2. Fig. 4 shows the simulation results of the proposed $([8,4],255)$ EII code. The FERs of the code with $r=[102,101]$, $[79,124]$, and $[124,79]$, but the same $t_v=26$ are plotted in this figure. Simulations with other values of r have also been carried out. It was found that the FER is the lowest when $r_0 \approx r_1$.

In our scheme, t_v needs to be at least t_{v-1}, since the nested decoding rounds are calculating increasingly more higher-order syndromes and hence should not have decreasing correction capabilities. Besides, a smaller t_v allows more parities to be allocated to protect more columns and/or increase the correction capabilities of the column-wise code. Therefore, $t_v = t_{v-1}$ is used in our proposed design. In this case, the last nested decoding round is skipped to reduce the decoding latency and decoding failure is declared in the $(v-1)$-th round if $b_v > 0$. Fig. 4 also shows the FERs of the proposed EII-RS code with three different t_v values: 26, 30 and 40. For each of these values, r_0 and r_1 are set to be about the same. It can be observed that the error-correcting performance of the proposed scheme decreases with larger t_v. Overall, the proposed $([8,4],255)$ EII-RS code with $t_v = 26$ and $r = [102,101]$ achieves lower FER and decoding latency compared to the settings with different t_v and/or r.

For comparisons, the error-correcting performance of the previous EII-RS code in [1], [2] with $t_v = 255$ that sacrifices the correction capability of random errors to correct a long burst of errors is included in Fig. 4. It can be observed that the proposed $([8,4],255)$ EII-RS code with $t_v = 26$ and $r = [102,101]$ can reduce the FER by several orders of magnitude over channels with random errors. Additionally, the probability that a column-wise word has a large number of errors is small. Therefore, the proposed scheme with column-wise codes can effectively correct some errors. As a result, the nested decoding can continue for the next round to correct more errors. Fig. 4 shows that the proposed code achieves significant coding gain over the $([8,4],255)$ GII code [3], [4] with $t=[15,22,30,57]$ and $s=[s_1,s_2,s_3,s_4]=[3,2,2,0]$ that has the same code rate.

TABLE I
WORST-CASE LATENCIES OF THE $([8,4],255)$ EII-RS AND GII-RS DECODERS

nested dec. round	0	1	2	3	4	total (# of clks)
GII-RS [3], [4], $s=[3,2,2,0]$ $t=[15,22,30,57]$	90	485	505	359	-	1439 (1.03)
EII-RS [1], [2], $s=[2,1,1,1]$ $t=[15,18,21,26,255]$	90	355	498	447	1	1391(1.00)
prop. EII-RS, $s=[2,1,1,1]$ $t=[15,18,21,26,26]$,$r=[102,101]$	213	443	435	343	-	1434(1.03)

V. LATENCY AND COMPLEXITY ANALYSES

This section analyzes and compares the latencies and complexities of the proposed decoder and prior designs by using an $([8,4],255)$ EII-RS code with $t = [15,18,21,26,26]$, $s=[2,1,1,1]$ and $r=[102,101]$ as an example.

The worst-case latencies of the proposed design in each nested decoding round are listed in Table I. In the sub-word decoding, which is referred to as nested decoding round 0, $m=8$ conventional RS decoders are employed to decode all the received sub-words in parallel. For $t_0=15$, the iterative key equation solver (KES) step of RS decoding needs $2t_0=30$ clock cycles. The parallelisms of other two steps, syndrome computation and Chien search, are adjusted so that they are finished in 30 clock cycles each in order to maximize the hardware utilization efficiency. As shown in Fig. 3, when $b_\eta > \hat{s}_\eta$, the proposed scheme carries out the column-wise decoding. If any columns get corrected, a second trial nested decoding for round 0 is done. For $r=[102,101]$, column 0 through 100 can correct 1 error or 2 erasures, and column 101 can only correct 1 erasure. Either 1-error-correcting or 2-erasure-correcting RS decoding can be completed in 3 clock cycles. By using 10 copies of each decoder, the overall column-wise decoding can be done in $\lceil 102/10 \rceil \times 3 = 33$ clock cycles. Besides, the second trial nested decoding for round 0 takes another 90 clock cycles by re-using the same sub-word decoder. Hence, in the worst case, the sub-word decoding of the proposed design takes $90+33+90=213$ clock cycles.

Architectures for nested decoding have been developed in [7]–[10]. Since the later nested decoding rounds are activated with low probability, hardware units are reused for round 1 through v. To make up for the longer worst-case sub-word decoding latency and achieve similar overall worst-case decoding latency as the previous EII scheme in [1], [2], higher parallelisms are used for the syndrome computation and Chien search architectures for the nested decoding in the proposed design. As mentioned in the previous section, the proposed EII-RS code has $t_v = t_{v-1}$ and the last nested decoding round is skipped. The worst-case number of clock cycles needed for each nested decoding round is listed in Table I. The same hardware units are used to decode the sub-words one by one for the nested decoding. Using the reconfigurable Chien search architecture in [7], the Chien search latency is longer in later nested decoding round, and hence the number of clock cycles needed by the nested decoding round does not reduce linearly with the number of sub-words to correct. Compared to the previous $[8,4]$ EII code [1], [2] and the $[8,4]$ GII-RS code [3], [4], the proposed design can achieve similar worst-case latency and much lower FER as shown in Fig. 4.

TABLE II
AVERAGE LATENCIES OF THE $([8,4],255)$ EII-RS AND GII-RS DECODERS OVER 3×10^{-2} INPUT SYMBOL ERROR RATE

nested dec. round	0	1	2	3	4	total (# of clks)
$([8,4],255)$ GII-RS [3], [4], $t=[15,22,30,57]$, $s=[3,2,2,0]$						
avg. # of clks in nested dec.	90	90.1	145.0	127.0	-	
activation prob. of nested dec.	1	4×10^{-2}	3×10^{-5}	$<10^{-6}$	-	
avg. # of clks aggr. with act. prob.	90	3.6	0	0	-	93.6 (0.99)
$([8,4],255)$ EII-RS [1], [2], $t=[15,18,21,26,255]$, $s=[2,1,1,1]$						
avg. # of clks in nested dec.	90	84.2	176.2	233.0	1	
activation prob. of nested dec.	1	4×10^{-2}	2×10^{-3}	9×10^{-5}	$<10^{-6}$	
avg. # of clks aggr. with act. prob.	90	3.4	0.4	0	0	93.8 (1.00)
prop. EII-RS, $t=[15,18,21,26,26]$, $s=[2,1,1,1]$, $r=[102,101]$						
avg. # of clks in first nested dec. trial	90	53.6	81.1	99.0	-	
activation prob. of first nested dec. trial	1	4×10^{-2}	2×10^{-3}	9×10^{-5}	-	
avg. # of clks in col.-wise dec. & sec. nested dec. trial	123.0	200.5	234.0	120.0	-	
activation prob. of col.-wise dec. & sec. nested dec. trial	$<10^{-7}$	$<10^{-7}$	$<10^{-7}$	$<10^{-8}$	-	
avg. # of clks aggr. with act. prob.	90	2.1	0.2	0	-	92.3(0.98)

TABLE III
HARDWARE COMPLEXITIES OF THE DECODERS FOR THE $([8,4],255)$ EII-RS AND GII-RS CODES OVER $GF(2^8)$

	sub-word decoder	nested decoder	col.-wise decoder	total (# XORs)	crit. path (# gates)
$[8,4]$ GII-RS [3], [4] $s=[3,2,2,0]$ $t=[15,22,30,57]$	289088	104392	0	393480 (1.15)	11
$[8,4]$ EII-RS [1], [2] $s=[2,1,1,1]$ $t=[15,18,21,26,255]$	289088	54372	0	343460 (1.00)	11
prop. EII-RS, $s=[2,1,1,1]$ $t=[15,18,21,26,26]$ $r=[102,101]$	289088	58775	18640	366503 (1.06)	11

Table II shows the average latencies of the proposed EII-RS decoder when the input symbol error rate is 3×10^{-2}. The average number of clock cycles needed for each nested decoding round is dependent on the average number of erroneous sub-words to correct. Each nested decoding iteration is activated with certain probability. Besides, the column-wise decoding and the second nested decoding trial are only carried out when $b_\eta > \hat{s}_\eta$. Simulations have been carried out over 10^8 samples to find these values as listed in Table II. Then the average latency of the proposed design is computed by aggregating the average clock cycle numbers with the corresponding activation probabilities. The nested decoding has decreasing activation rates in later rounds. For the proposed design, the column-wise decoding is activated with a very low probability in each round. As a result, the average latency of the proposed design is similar to that of the previous EII and GII decoders.

The hardware complexities of the EII and GII decoders in terms of the equivalent numbers of XOR gates needed are summarized in Table III. The sub-word decoder of each design consists of $m=8$ t_0-error-correcting RS decoders. Although our proposed design has higher parallelisms for the syndrome computation and Chien search architectures in the nested decoder and requires column-wise decoders, the sub-

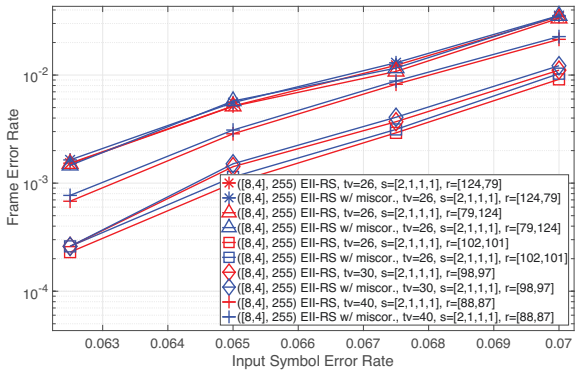

Fig. 5. Error-correcting performance of the proposed $([8,4],255)$ EII-RS codes with miscorrections with random input errors.

word decoder accounts for the majority portion of the overall decoder. Hence the total complexity of our design is slightly larger than that of the previous $[8,4]$ EII decoder [1], [2]. Since the $[8,4]$ GII-RS decoder [3], [4] has much higher correction capabilities in the nested decoding, it requires more gates to implement. Additionally, all three decoders have the same critical path and hence can achieve the same clock frequency.

VI. Discussions

In the proposed EII scheme, miscorrections may happen in the column-wise error-correcting decoding since the correction capability is low. For the example EII code with $t_v = 26$ and $r = [102, 101]$, the first 101 columns have 2 parities and their error-correcting capabilities are only 1. For a column, if miscorrection happens and any errors from the decoding results are located in corrected sub-words, then the miscorrection is identified and its decoding results are ignored. Even if all errors from the miscorrection are located in uncorrected sub-words and hence the miscorrection is unidentifiable, some of the errors in the uncorrected sub-words may still be corrected by decoding other columns. Therefore, miscorrections on the column-wise words do not bring much performance degradation on the overall EII decoding. Fig. 5 shows the FERs of the proposed $([8,4],255)$ EII-RS codes with miscorrections collected by carrying out actual EII decoding. The simulations take a long time and can not be done for lower input symbol error rates. On the other hand, the FER of the decoding without considering miscorrections can be derived by checking if the numbers of errors in the sub-words exceed the correction capability. As it can be observed from Fig. 5, the FERs of the proposed EII codes degraded by miscorrections are small.

For EII codes constructed using $\mathcal{C}_0, \mathcal{C}_1, \cdots, \mathcal{C}_v$ that are BCH codes, the proposed scheme can still be applied. RS codes are still used as the column-wise code since they are MDS. This can be done by dividing the bits in each sub-word into groups of q bits for RS codes over $GF(2^q)$. The order of the finite field for constructing the column-wise code can be different from that for forming $\mathcal{C}_j (0 \le j \le v)$ codes. Using a lower-order finite field to construct the column-wise code will divide the sub-codewords into more columns and the correction capability of the column-wise code does not change. A lower-order finite field reduces the encoding and decoding complexity but leads to more miscorrections in the column-wise decoding.

For codes with higher code rates but the same m and n, the numbers of parities allocated to $\mathcal{C}_0, \mathcal{C}_1, \cdots, \mathcal{C}_{v-1}$ and hence their correction capabilities are reduced. The chance that the nested decoding can not continue to the next round in previous EII and GII schemes becomes higher. Due to the column-wise codes, the proposed EII scheme can achieve more significant reduction on the FER. Besides, when the input error rate becomes lower, the number of errors in a column is reduced and more column-wise words can be corrected. As a result, the proposed design can achieve further performance gain over previous EII and GII codes for lower input error rates.

For practical channels, even if there are burst errors, they are unlikely to be length n. In our design, the column-wise decoding can correct some errors, after which a burst of t_v errors can be further corrected in one sub-word. Hence, our design can achieve similar decoding failure rate as the previous EII schemes for channels with burst errors of practical length.

VII. Conclusions

This paper proposes a modified EII code that incorporates the column-wise code across individual sub-codewords. Parities for correcting long bursts of errors in the previous EII schemes are re-allocated to correct random errors through the column-wise code. Our design achieves a good trade-off between the burst and random error correction capabilities. For channels with random errors, the proposed EII code can achieve significant improvement on the error-correcting performance compared to previous EII and GII codes. Future works will explore other variations of EII codes that can further improve the correction capability.

References

[1] M. Blaum and S. R. Hetzler, "Extended Product and Integrated Interleaved Codes," *IEEE Trans. on Info. Theory*, vol. 64, no. 3, pp. 1497-1513, Mar. 2018.

[2] M. Blaum, "Multiple-layer integrated interleaved codes: a class of hierarchical locally recoverable codes,"*IEEE Trans. on Info. Theory*, vol. 68, no. 8, pp. 5098-5111, Aug. 2022.

[3] X. Tang and R. Koetter, "A novel method for combining algebraic decoding and iterative processing," *IEEE Int. Symp. on Info. Theory*, Seattle, WA, USA, 2006, pp. 474-478.

[4] Y. Wu, "Generalized integrated interleaved codes," *IEEE Trans. on Info. Theory*, vol. 63, no. 2, pp. 1102-1119, Feb. 2017.

[5] X. Zhang, "Systematic encoder of generalized three-layer integrated interleaved codes," *IEEE Int. Conf. on Commun.*, Shanghai, China, 2019.

[6] Y. J. Tang and X. Zhang, "Low-complexity resource-shareable parallel generalized integrated interleaved encoder," *IEEE Trans. on Circuits and Syst.-I*, vol. 69, no. 2, pp. 694-706, Feb. 2022.

[7] X. Zhang and Z. Xie, "Efficient architectures for generalized integrated interleaved decoder," *IEEE Trans. on Circuits and Syst.-I*, vol. 66, no. 10, pp. 4018-4031, Oct. 2019.

[8] Z. Xie and X. Zhang, "Reduced-complexity key equation solvers for generalized integrated interleaved BCH decoders," *IEEE Trans. on Circuits and Syst.-I*, vol. 67, no. 12, pp. 5520-5529, Dec. 2020.

[9] Z. Xie and X. Zhang, "Fast nested key equation solvers for generalized integrated interleaved decoder," *IEEE Trans. on Circuits and Syst-I*, vol. 68, no. 1, pp. 483-495, Jan. 2021.

[10] Y. J. Tang and X. Zhang, "Fast and low-complexity soft-decision generalized integrated interleaved decoder," *IEEE Journal on Select. Areas in Info. Theory*, vol. 4, pp. 174-189, 2023.

[11] M. Blaum, "Extended integrated interleaved codes over any field with applications to locally recoverable codes," *IEEE Trans. on Info. Theory*, vol. 66, no. 2, pp. 936-956, Feb. 2020.

[12] K. Deng, X. Qiao, Y. Chen, S. Song and Z. Wang, "Performance analysis of extended integrated interleaved codes," *Asia Pacific Conf. on Commun.*, Jeju Island, Korea, Republic of, 2022, pp. 498-503.