Generalized Integrated Interleaved Codes for High-Density DRAMs

Yok Jye Tang and Xinmiao Zhang
The Ohio State University, Columbus, OH 43210, USA

***Abstract*—As the density of dynamic random access memories (DRAMs) keeps increasing, which results in higher error rates, the conventional single error correction and double error detection codes are no longer sufficient. Generalized integrated interleaved (GII) codes based on Reed-Solomon (RS) codes are among the best error-correcting codes for high-density DRAMs due to their hyper-speed decoding and good correction capability. However, the very short codeword length required by DRAMs leads to miscorrections that substantially degrade the error-correcting performance of GII-RS codes if untreated. Previous miscorrection mitigation schemes for longer GII-BCH codes lead to additional code rate loss when applied to very short GII-RS codes and hence affect the cost of DRAMs. This paper presents new miscorrection mitigation schemes with improved code rates. A small number of parity bits are allocated in an optimized manner and decoding trials are carried out to close the performance gap. Moreover, low-latency hardware implementation architectures have been developed for the proposed GII-RS decoder. For the example code considered for DRAMs, the proposed decoder reduces the worst-case latency by 45% with small area overhead while having the same average latency and critical path compared to the alternative design.**

I. Introduction

As the density of dynamic random access memories (DRAMs) keeps increasing to support high-performance systems, their error rate is becoming higher. Low-redundancy error-correcting codes with better correction capability and hyper-speed decoding are needed to replace the traditional single error correction and double error detection codes. The generalized integrated interleaved (GII) codes [1], [2] that nest short Reed-Solomon (RS) sub-codewords to form codewords of more powerful RS codes are the best potential candidates. Their decoding is divided into two stages. Hyper-speed traditional RS decoding is first carried out over individual short sub-words. Only when there are extra errors, the second-stage nested decoding is activated.

The current single symbol correction (SSC) CHIPKILL scheme for DRAMs [3] utilizes four un-nested 1-error-correcting (18,16) RS codewords over $GF(2^8)$. To keep the codeword format similar, each GII codeword consists of four 1-error-correcting RS sub-codewords with 16 data symbols. Besides, these sub-codewords are nested to generate one 3-error-correcting RS codeword to correct more errors with simple decoders. Such a GII code can achieve orders of magnitude lower decoding failure rate compared to individual (18, 16) RS codes. However, miscorrections on the 1-error-correcting sub-words lead to severe degradation on the correction performance if untreated.

For longer GII codes based on BCH codes, nested syndromes are checked and extended BCH codes are adopted to identify and mitigate miscorrections in [4], [5]. However, for very short GII-RS codes, the extra parities of extended RS codes lead to additional code rate loss. Code rate decides the capacity and is a key factor affecting the cost of DRAM devices. Hence it needs to be increased as much as possible.

In this paper, two low-redundancy miscorrection mitigation schemes are proposed for GII-RS codes with 1-error-correcting sub-codewords. Instead of using extended RS codes with extra parity symbols, our first method uses a few parity bits produced by XOR operations. The parity bits allocated to sub-words are optimized to improve the probability of identifying miscorrections by taking into account the nested syndrome checking. The performance degradation caused by miscorrections is further reduced by carrying out multiple nested decoding trials in our second method. Additionally, low-latency hardware implementation architectures are developed for each decoding step. For the example code considered for DRAMs, the proposed decoder reduces the worst-case latency by 45% with small area overhead and the same average latency and critical path compared to the alternative design.

II. Background

A $[m, v]$ GII code can be constructed using $v+1$ RS codes, $\mathcal{C}_v \subseteq \mathcal{C}_{v-1} \subseteq \cdots \mathcal{C}_1 \subset \mathcal{C}_0$, with dimension k and error-correcting capabilities $t_v \geq \cdots \geq t_1 > t_0$ as [1], [2]

$$\mathcal{C} \triangleq \{c=[c_0, \cdots, c_{m-1}] : c_i \in \mathcal{C}_0, \tilde{c}_l = \sum_{i=0}^{m-1} \beta^{il} c_i \in \mathcal{C}_{v-l}, 0 \leq l < v\}, \tag{1}$$

where β is a primitive element of $GF(2^q)$. c_i $(0 \leq i < m)$ and $\tilde{c}_l$ $(0 \leq l < v)$ are referred to as the sub-codewords and nested codewords, respectively.

GII decoding consists of two stages. The first stage is the traditional RS decoding that corrects up to t_0 errors in each sub-word. The second-stage nested decoding is activated when there are extra errors. It has up to v rounds and the η-th $(1 \leq \eta < v+1)$ round can correct up to t_η errors in at most $b_\eta \leq v+1-\eta$ sub-words. This is achieved by first computing $2(t_\eta - t_{\eta-1})$ higher-order syndromes of the first b_η nested codewords and then deriving the higher-order syndromes of the sub-words by inverting the linear combinations in (1).

For DRAM applications, the GII-RS $[4, 1]$ code with $k = 16$ and $[t_0, t_1] = [1, 3]$ over $GF(2^8)$ achieves good trade-off on the error-correcting capability and decoding complexity. Fig. 1 shows that this code can theoretically achieve orders of magnitude lower decoding frame error rate (FER) compared to four un-nested (18, 16) RS codes used in SSC CHIPKILL [3]. However, the actual FER of this GII-RS code taking into

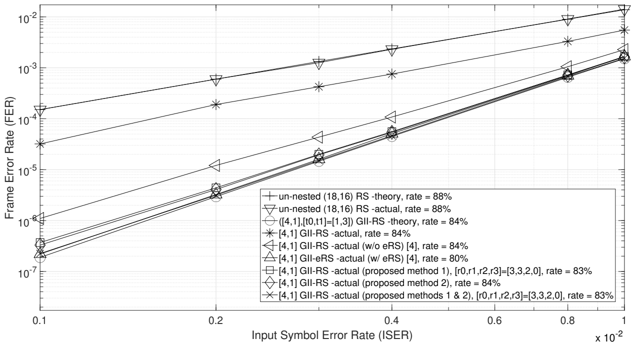

Fig. 1. FERs of [4, 1] GII-RS decoding with $[t_0, t_1] = [1, 3]$.

account the miscorrections is much higher. This is because, if the miscorrected sub-words are not identified, then they are not sent to the nested decoding to correct extra errors.

Three miscorrection mitigation schemes have been proposed for $t_0 = 3$ GII-BCH codes in [4]. First, if any higher-order nested syndrome is non-zero, some sub-words are miscorrected. To identify which sub-words are miscorrected, the second method utilizes extended BCH codes and the third scheme checks if the degree of the error-locator polynomial is higher than t in t-error-correcting decoding.

III. Miscorrection mitigation schemes for GII-RS codes with 1-error-correcting sub-codeword

In the case of GII-RS codes, extended RS (eRS) codes can be employed for miscorrection mitigation. Combined with the other two schemes in [4], the actual FER of the $[4, 1]$ GII-RS code becomes closer to the theoretical FER as shown in Fig. 1. However, using eRS codes reduces the code rate by 4%. On the other hand, the FER increases significantly without them. Utilizing doubly eRS codes as proposed in [5] reduces the code rate by an additional 4% and is not considered. This section presents two low-redundancy miscorrection mitigation schemes to close the performance gap of short GII-RS codes.

Instead of using parity symbols, parity bits generated by XOR operations can be adopted to detect miscorrections with reduced redundancy. Assume c_i has n_i symbols over $GF(2^q)$ and $r_i (r_i < q)$ parity bits are used for miscorrection detection. Each parity is the XOR result of nq/r_i bits of c_i. The same parity is calculated after the decoding and miscorrection is declared if the parities do not match. The probability that a miscorrected sub-word is not identified from these parities is around $1/2^{r_i}$. Although eRS codes can detect miscorrections with higher probability, they require at least q parity bits for each sub-word and hence significantly decrease the code rate for short GII-RS codes.

The probability of successfully detecting a miscorrection increases with the number of parity bits. Since the nested syndromes tell if there are miscorrections among the m sub-words and only one sub-word can be sent to the nested decoding process for the [4,1] GII-RS code, the available parity bits can be allocated to the first $m - 1$ sub-words to improve miscorrection detection. If some nested syndromes are nonzero and miscorrections are not detected for the first $m - 1$ sub-words, then the last sub-word is miscorrected and is sent to the nested decoding. The total number of parity bits should be a multiple of q to simplify the storage. For the $[4, 1]$ GII-RS code, allocating $[r_0, r_1, r_2, r_3] = [3, 3, 2, 0]$ parity bits can substantially reduce the performance gap as shown in Fig. 1 with only 1% code rate loss.

TABLE I

Complexity and latency comparisons of $[4, 1]$ GII-RS decoders with $[t_0, t_1] = [1, 3]$ and 83% code rate over $GF(2^8)$ using the two proposed miscorrection mitigation schemes

	total complex. (# XORs)	critical path (# gates)	worst-case latency(# clks)	avg. latency (# clks)
alternat. design [7]	24.98k	10	29	4
prop. GII-RS dec.	26.52k	10	16	4

If a miscorrected sub-word is not identified and accordingly sent to the nested decoding, the higher-order syndromes are incorrect and, as a result, the nested decoding will most likely fail. Considering this, if miscorrections are detected from the nested syndromes, our second proposed scheme carries out nested decoding on each sub-word until the decoding is successful. Besides, the nested decoding is skipped on those sub-words whose lowest $2t_0 = 2$ syndromes are both zero since they are correct sub-codeword with high probability. For the $[4, 1]$ GII-RS code, the actual FER with miscorrection is significantly reduced by adopting our second scheme as shown in Fig. 1. Additionally, combining the two proposed schemes can achieve the same FER as GII codes with eRS codes that have a much lower code rate.

Table I shows the complexity of the proposed $[4, 1]$ GII-RS decoder architecture with $[t_0, t_1] = [1, 3]$ over $GF(2^8)$ that employs the two proposed miscorrection mitigation schemes. In our designs, the key equation solver (KES) is implemented more efficiently by using the fully-pipelinable architectures in [6] such that each nested decoding trial requires one additional clock cycle. For comparison, an alternative GII decoder design that utilizes the nested KES architecture in [7] is considered. Since the proposed design only needs one additional clock cycle for each nested decoding trial, it reduces the worst-case latency by $1-(16/29)\times 100 = 45\%$ with small area overhead and same critical path. Besides, since the error rate of DRAMs is very low, the probability of activating the nested decoding is very small. Accordingly, each design in Table I has similar average decoding latency, which is decided by the sub-word decoding and nested syndrome checking.

References

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

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

[3] Yeleswarapu, Ravikiran and Somani, Arun K., "Addressing multiple bit/symbol errors in DRAM subsystem," *PeerJ Comp. Science*, Feb. 2019.

[4] Z. Xie and X. Zhang, "Miscorrection mitigation for generalized integrated interleaved BCH codes," *IEEE Commun. Letters*, vol. 25, no. 7, pp. 2118-2122, Apr. 2021.

[5] Z. Xie and X. Zhang, "Improved miscorrection detection for generalized integrated interleaved BCH codes," *Proc. IEEE Int. Conf. Commun.*, 2022.

[6] Z. Yan, J. Lin and Z. Wang, "A low-complexity RS decoder for triple-error-correcting RS codes," *IEEE Computer Society Annual Symp. on VLSI*, 2019, pp. 489-494.

[7] W. Li, J. Lin, and Z. Wang, "A 124-Gb/s decoder for generalized integrated interleaved codes," *IEEE Trans. Circuits and Syst.-I*, vol. 66, no. 8, pp. 3174-3187, Aug. 2019.