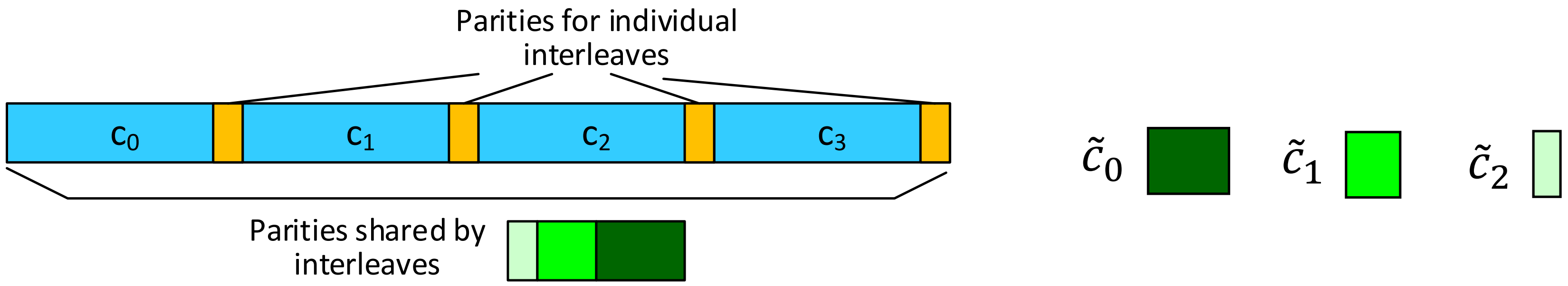
Fast Nested Decoding for Short Generalized Integrated Interleaved BCH Codes

Zhenshan Xie and Xinmiao Zhang
Dept. of Electrical and Computer Engineering
The Ohio State University

Outline

- ➢Generalized integrated interleaved (GII) codes
- ➢Short GII-BCH codes and miscorrections
- ➢Improved miscorrection detection and mitigation schemes
- ➢Latency analyses and comparisons
- ➢Conclusions

Generalized Integrated Interleaved (GII) Codes



Nesting matrix G

$$\begin{bmatrix} \tilde{c}_0 \\ \tilde{c}_1 \\ \vdots \\ \tilde{c}_{v-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{m-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{(v-1)} & \alpha^{2(v-1)} & \cdots & \alpha^{(v-1)(m-1)} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{m-1} \end{bmatrix}$$

Correction capability

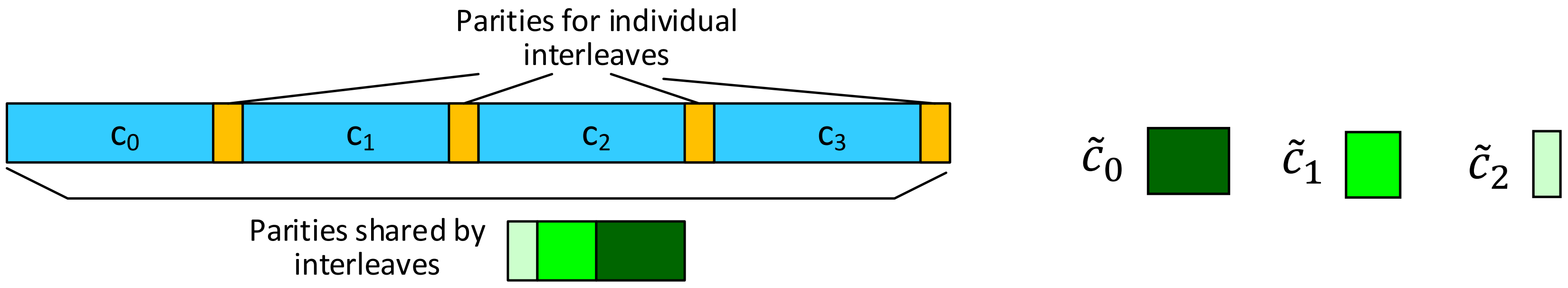
$$t_v \geq t_{v-1} \geq \cdots \geq t_1 \geq t_0$$

$$\mathcal{C}_v \subseteq \mathcal{C}_{v-1} \subseteq \cdots \subseteq \mathcal{C}_1 \subseteq \mathcal{C}_0$$

$$\tilde{c}_0 \in \mathcal{C}_v, \quad \tilde{c}_1 \in \mathcal{C}_{v-1}, \quad \tilde{c}_{v-1} \in \mathcal{C}_1, \quad c_0, c_1, \cdots, c_{m-1} \in \mathcal{C}_0$$

- Each sub-codeword $(c_0, c_1, \cdots)$ is a short BCH or Reed-Solomon (RS) codeword capable of correcting t_0 errors
- The nested codewords $(\tilde{c}_0, \tilde{c}_1, \cdots)$ belong to more powerful BCH or RS codes
- The extra correction power of the nested codewords are manifested as parities shared by the sub-codewords
- GII codes can achieve hyper speed decoding with good correction capability and low redundancy

Decoding of GII Codes



Sub-codeword syndromes

nested syndromes

$$\begin{bmatrix} S_j^{(l_1)} \\ S_j^{(l_2)} \\ \vdots \\ S_j^{(l_b)} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha^{l_1} & \alpha^{l_2} & \cdots & \alpha^{l_b} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{(b-1)l_1} & \alpha^{(b-1)l_2} & \cdots & \alpha^{(b-1)l_b} \end{bmatrix}^{-1} \begin{bmatrix} \tilde{S}_j^{(0)} \\ \tilde{S}_j^{(1)} \\ \vdots \\ \tilde{S}_j^{(b-1)} \end{bmatrix}$$

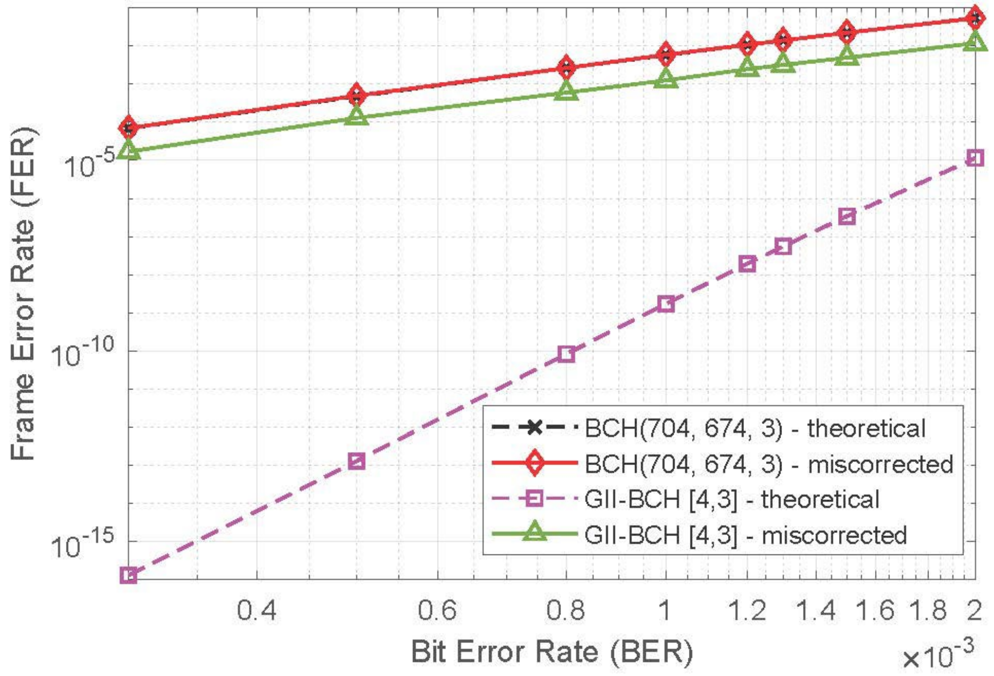
Syndrome conversion matrix

$l_1, l_2, \cdots, l_b$: indices of sub-words with more than t_0 erasures

- Two decoding stages: i) individual sub-word decoding; ii) nested decoding
- Nested decoding has up to v rounds
  - Compute higher-order syndromes of the nested words
  - Convert them to higher-order syndromes of the sub-words
  - Correct more errors in the sub-words

Performance of Short GII-BCH Codes

- Short codes are required for storage class memories (SCMs)
  - Short length, e.g., several thousand bits
  - High code rate, e.g., 90%
  - Example GII-BCH code
    - 256 parity bits to protect 2560 data bits
    - $m = 4$ sub-codewords, each has 704 bits
    - $v = 3$ nested codewords
    - $[t_0, t_1, t_2, t_3] = [3,5,6,11]$

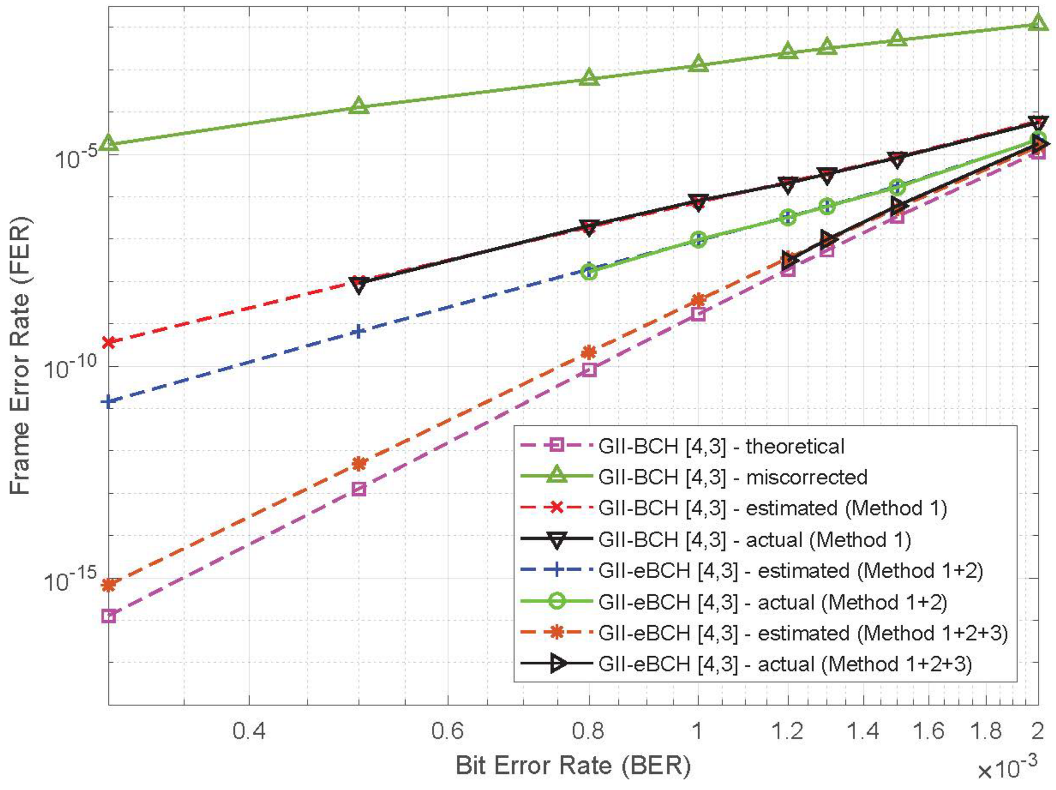


- GII-BCH codes theoretically achieve much better error-correcting performance than traditional BCH codes that have similar complexity
- Miscorrections on the sub-words cause severe performance degradation for short GII codes

Miscorrections and Previous Mitigation Schemes

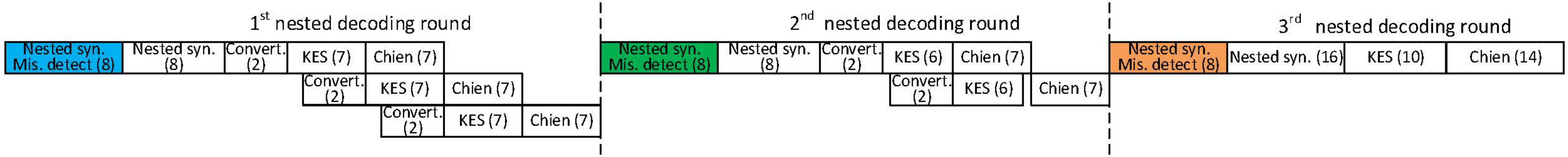
- Miscorrections in t-error-correcting BCH decoding: a received word with $> t$ errors is decoded to another codeword
- Miscorrections happen more often for smaller t
- If a mis-corrected sub-word is not detected, the more powerful nested decoding is not activated

- Prior miscorrection detection/mitigation schemes
  - Method 1: check higher-order nested syndromes
  - Method 2: test if the error locator polynomial degree is higher than t
  - Method 3: utilize extended BCH codes



[1] 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.

Overheads of Previous Miscorrection Mitigation Schemes



- The three miscorrection mitigation schemes have negligible silicon area overhead
- Using extended BCH codes and check error locator polynomial degree do not bring latency overhead
- Computing nested syndromes before each nested decoding round bring significant latency overhead

Proposed Improved Miscorrection Mitigation Schemes

- Skip nested syndrome checking when miscorrections are less likely to happen
  - When the degree of error locator polynomial, $\deg(\Lambda(x))$, is small

$\deg(\Lambda(x))$	Probability of miscorrection
3	5.4×10^{-2}
2	2.5×10^{-4}
1	6×10^{-7}

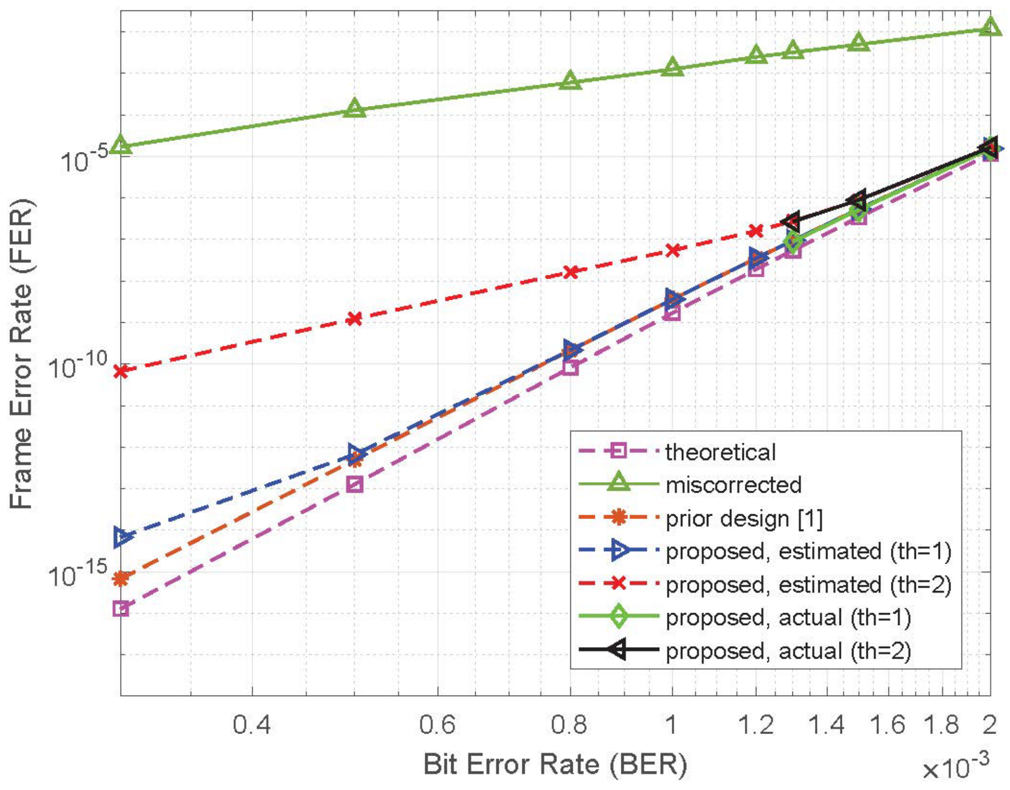
n =704 3-error-correcting BCH sub-codeword corrupted with 6 errors

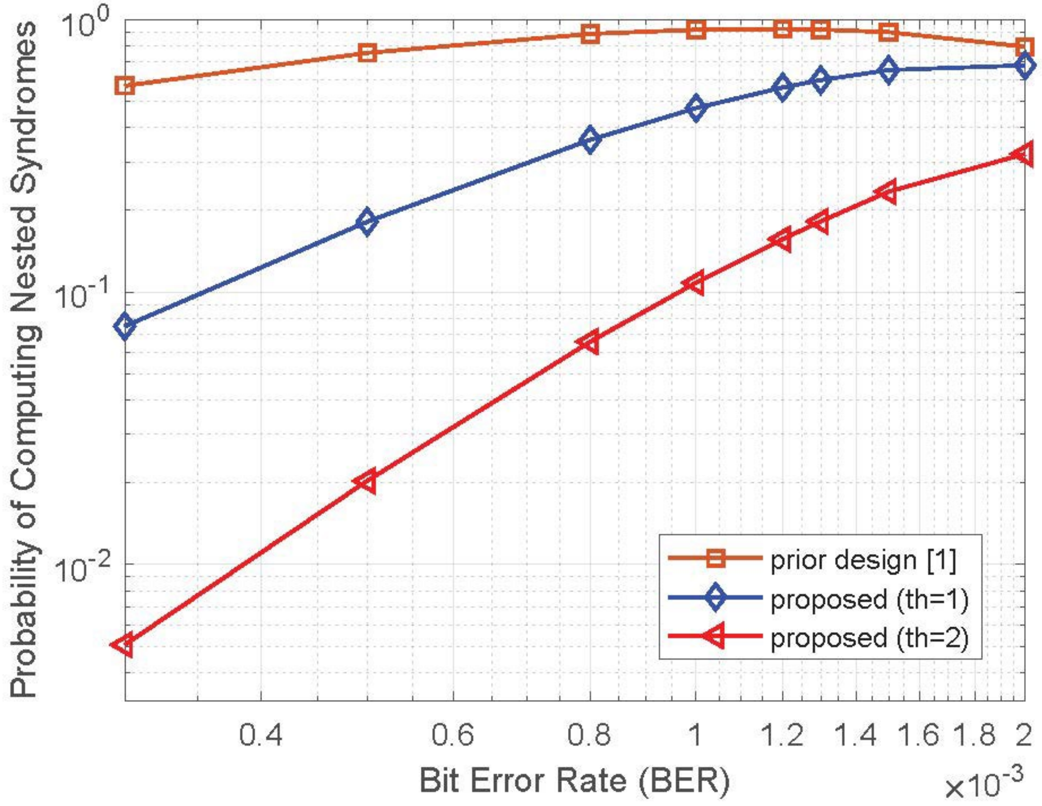
- Estimated frame error rate (FER) degradation caused by skipping the nested syndrome checking if $\deg(\Lambda(\mathrm{x})) \leq th$ in nested decoding round i

$$F_1^{(i)} = \binom{v-i}{1} \left(\sum_{w=t_i+1}^{t_v} \phi_w G_w^{(i)} \right) \left(\sum_{w=0}^{th} \phi_w \right)^{m-1}$$

  - $G_w^{(i)}$: probability of a w-error-corrupted sub-word miscorrected with $\deg(\Lambda(x)) \leq th$
  - $\phi_w = \binom{n}{w} p_b^w (1-p_b)^{n-w}$: probability of a n-bit sub-word corrupted with w errors
  - p_b: input bit error rate (BER)

Performance with Syndrome Checking Skipped when $\deg(\Lambda(x)) \leq th$





- Slight FER degradation when $th = 1$

- Syndrome checking for miscorrection detection is needed much less frequently
- Reduce the average nested decoding latency

Skip Syndrome Checking at Later Nested Decoding Rounds

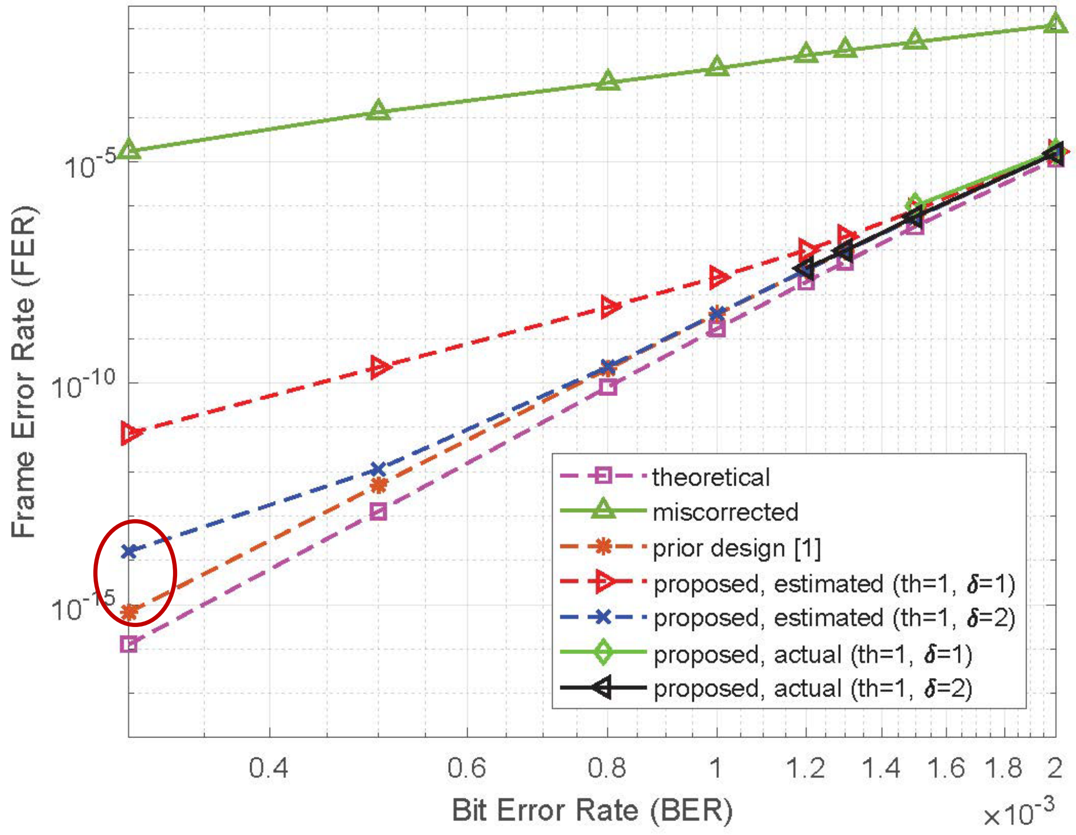
- Skip nested syndrome checking when miscorrections are less likely to happen
  - After later nested decoding rounds that have larger t_i

- Estimated (FER) degradation caused by skipping the nested syndrome checking after nested decoding rounds δ

$$F_2^{(\delta)} = \binom{v-\delta}{1}\left(\sum_{w=t_\delta+1}^{t_v} \phi_w G_w'^{(\delta)}\right)\left(\sum_{w=0}^{t_\delta} \phi_w\right)^{m-1}$$

  - $G_w'^{(\delta)}$: probability of a w-error-corrupted sub-word miscorrected with $\deg\big(\Lambda(x)\big) \le t_\delta$ and not detected by 1-bit extended BCH code

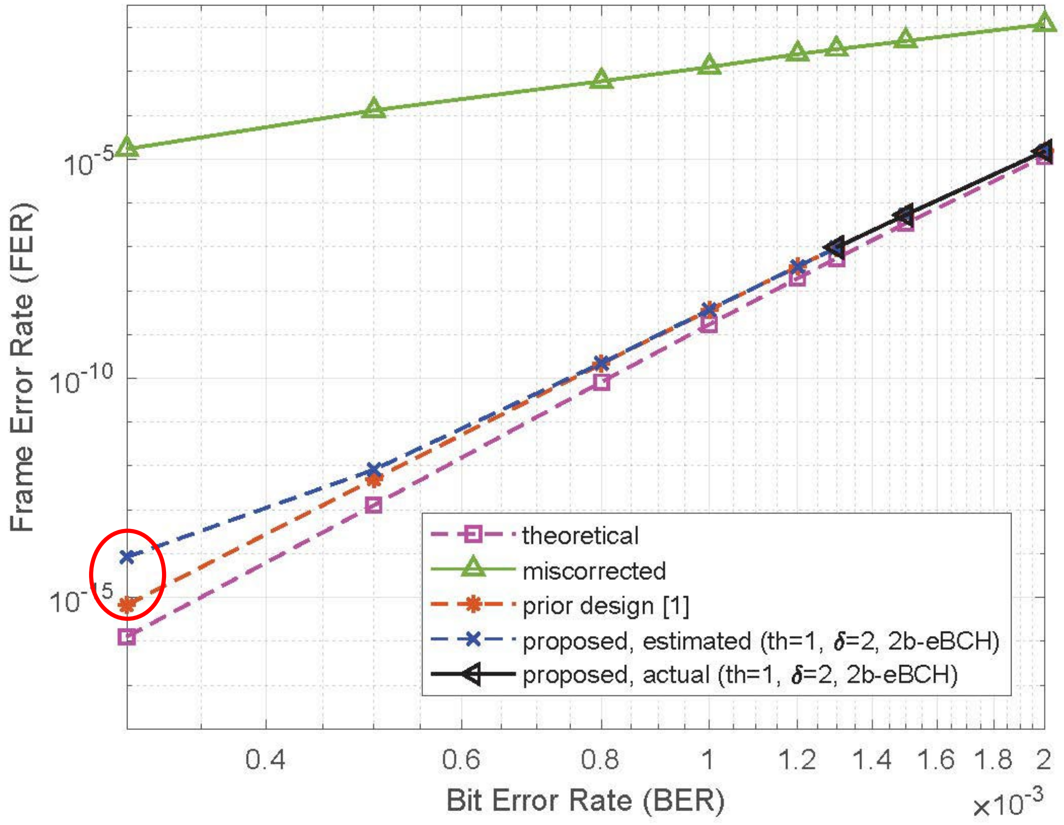
Performance with Skipped Syndrome Checking



- Small FER degradation when $th = 1$ and $\delta = 2$
- Reduce both the average and worst-case nested decoding latency

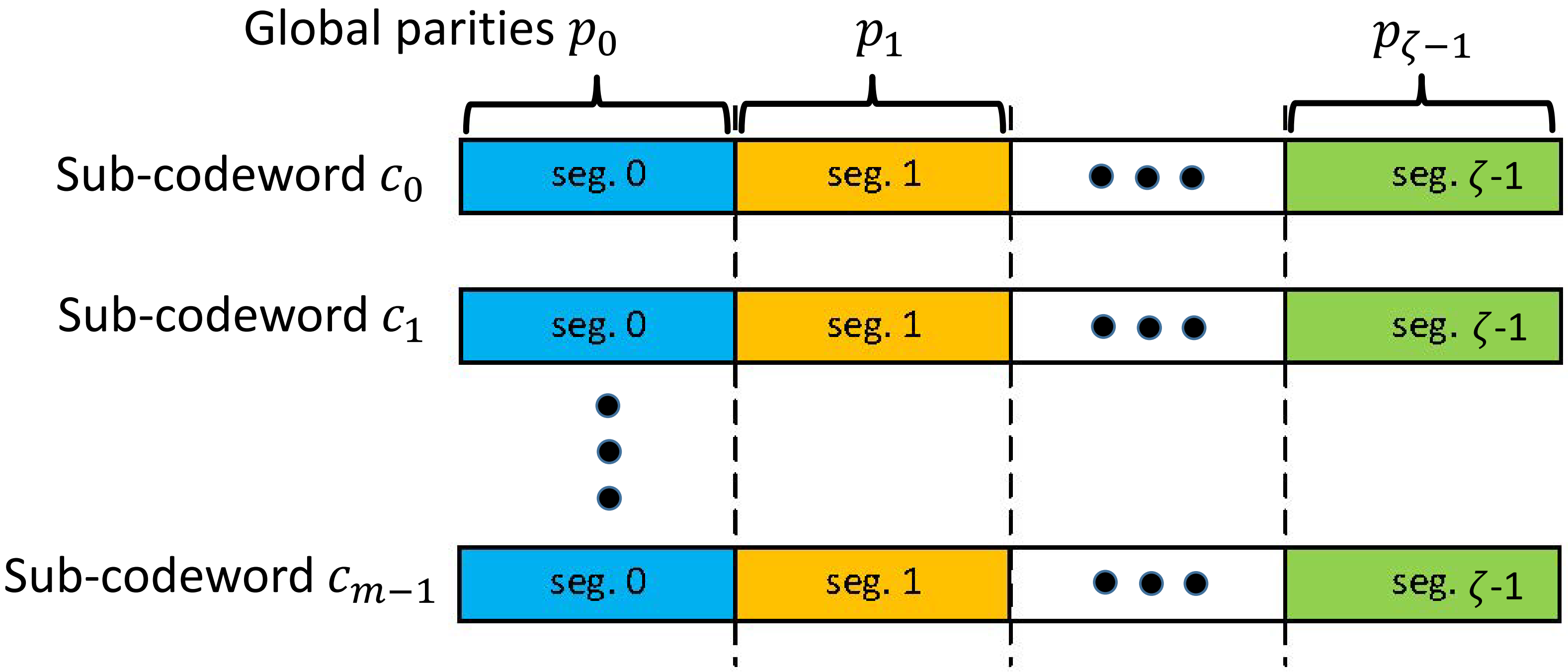
2-Bit Extended BCH Codes for Miscorrection Detection

- Utilize 2-bit extended BCH codes for each sub-codeword
  - Multiplying $(x^2 + 1)$ to all generator polynomials
  - Hamming weight is even on all odd-index or even-index bits
  - Undetected error patterns reduced by half
  - FER degradation reduced by half

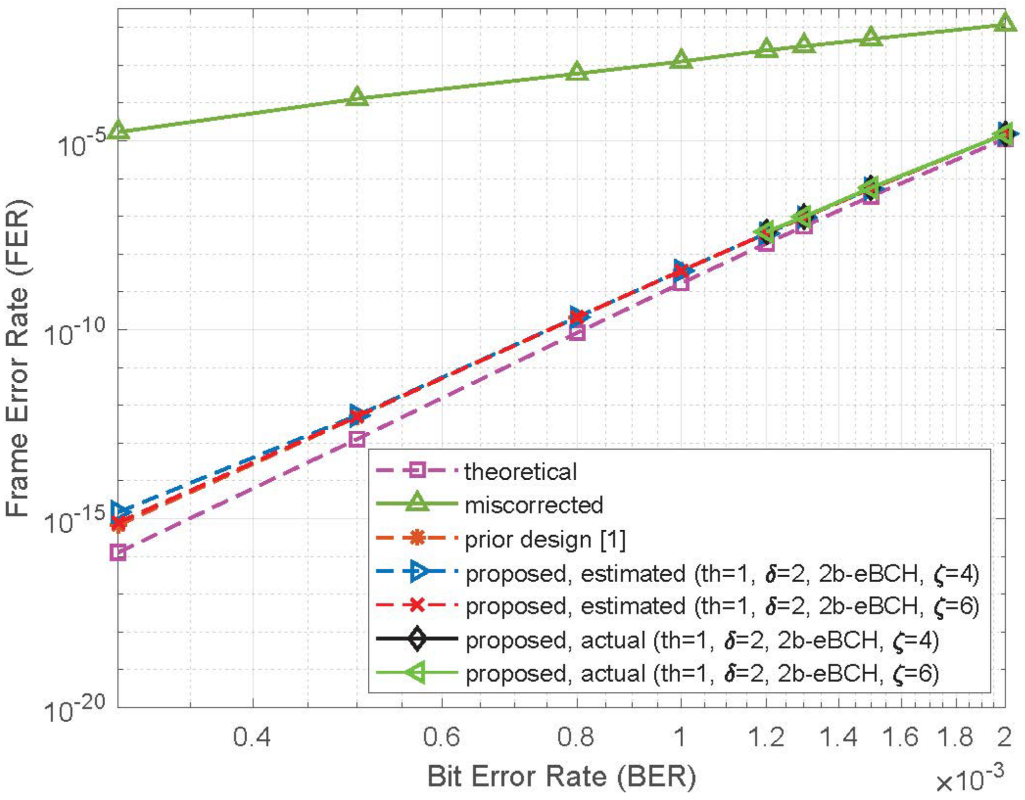


Global Parities for Miscorrection Detection

- Only one sub-word is miscorrected in most cases
- XOR result of all sub-words can detect miscorrections
  - Partition each sub-codeword into ζ segments as evenly as possible
  - The i-th global parity protects all the bits in the i-th segments of all sub-codewords

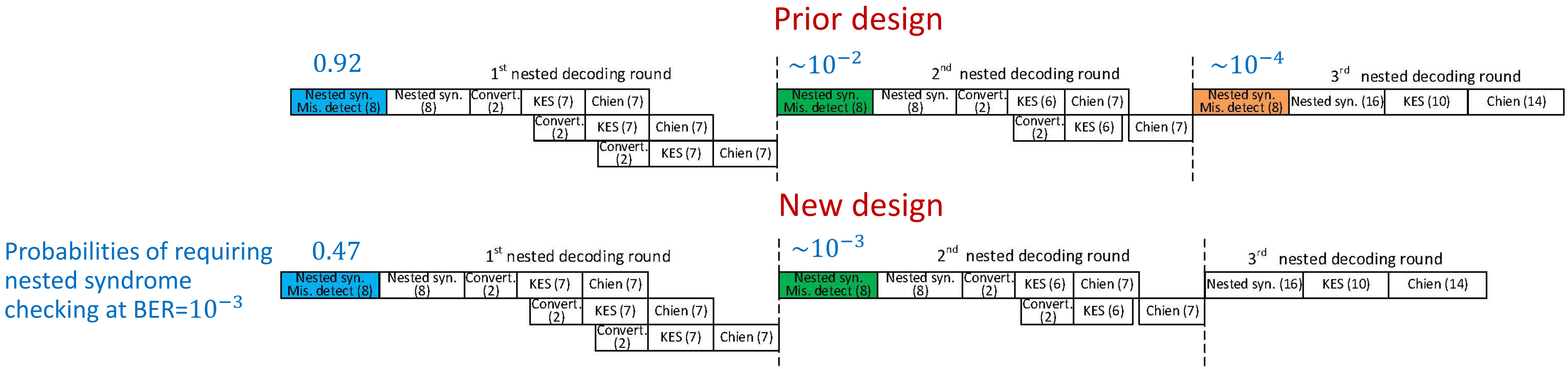


Performance with All Proposed Miscorrection Detection Schemes



- FER loss becomes negligible compared to the prior design [1]
- The 2-bit extended BCH codes and ζ =6 global parities only lead to (4+6)/704/4=0.35% code rate loss

Latency Analyses and Comparisons



	Average nested decoding latency	Worst-case nested decoding latency
Prior design [1]	8.3 clks	132 clks
Proposed design	4.7 clks	124 clks

43% average nested decoding latency reduction with almost the same FER, code rate, and silicon area!

Conclusions

- Optimize miscorrection mitigation schemes are developed for short GII-BCH codes
- The nested syndrome checking is skipped when miscorrections are less likely to happen
- 2-bit extended BCH codes and global parities are utilized to close the performance gap
- Formulas are provided to estimate the achievable FERs
- Proposed schemes lead to substantial latency reduction with almost the same error-correcting performance, code rate, and silicon area requirement