In-Memory BER Estimation Using Syndromes of LDPC Codes

Yotam Gershon Yuval Cassuto
The Viterbi Dept. of Electrical and Computer Engineering, Technion - Israel Institute of Technology
Email: {yotamgr@campus, ycassuto@ee}.technion.ac.il

***Abstract*—In-memory computing architectures highly improve computation latency and power compared to von Neumann architectures suffering the memory wall. However, maintaining reliability is challenging due to the difficulty in implementing error-correction coding within memory. We propose a scheme in which strong codes can be used for in-memory computing, while avoiding their costly decoding when error rates are sufficiently small. The key idea and thrust of the paper is to complement the design of LDPC codes to also provide accurate estimation of the input bit-error rate (BER). Toward that, we derive analytical results and give code-design insights for BER estimation in two frameworks: minimizing the mean-squared error (MSE), and estimating threshold crossing as a hypothesis-testing problem.**

I. Introduction

In-memory computing is a paradigm aiming to address the emerging bottlenecks of data-transfer rates and power to and from the memory [1], termed as the memory wall [2]. This capability is achieved by enabling the memory to perform computational tasks without transferring data externally. Many approaches have been explored for such architectures, ranging from implementing logic gates within traditional charge-based memories (mainly SRAM [3] and DRAM [4]), to utilizing inherent properties of resistance-based memories (mainly RRAM [5] and MRAM [6]), with promising approaches to enable logic-in-memory, such as MAGIC [7]. However, this capability also presents challenges, one of them being maintaining reliability. On the one hand, data integrity is challenged and degraded by the dynamic nature of the data and frequent read/write accesses, and on the other hand, error mitigation becomes challenging due to resource and latency constraints limiting the ability to employ powerful error-correction coding (ECC). The traditional way in which ECC is used for memory reliability is encoding before each write and *decoding after each read* [8]. This method is problematic for in-memory computing because of the fine access granularity and limited computing resources that challenge this approach. There have been proposals to implement ECC in memory (e.g. [9]), however, modern coding techniques, such as LDPC [10], [11] and polar [12], [13] codes, are difficult to decode with in-memory logic, while more-traditional, easier-to-decode coding schemes are costly in redundancy.

In this paper we propose a middle-ground architecture that utilizes the strong capabilities of LDPC codes for error detection and correction, while allowing their operation without need to decode after each read. Since a certain fraction of errors is tolerable in many modern applications, such as approximate computing [14] and machine learning [15], allowing a bit-error rate (BER) tolerance – under which no costly decoding should be performed – is highly reasonable. Only when the BER approaches this tolerance, data is transferred to a processing unit which decodes the LDPC code, then re-writing the correct codeword to memory. The advantages of this architecture are: (1) it enables the use of efficient LDPC codes for error protection, (2) it spares the in-memory logic from complicated decoding, and (3) it reduces the required frequency of data transfer without the risk of an error rate that exceeds the correction capability of the code.

A key component in the proposed architecture is the use of the sparse check equations for *in-memory BER estimation*, providing a precise indication of when it is necessary to transfer the data outside the memory for error decoding. BER estimation from syndromes was studied in [16], and in [17] that provided a closed-form expression for regular codes and a Cramer-Rao bound analysis on the estimation mean squared error (MSE). We revisit this problem, introducing: (1) a closed-form expression for the estimator of irregular codes, based on an approximate likelihood function, (2) an analysis of MSE dependency on the degree distribution, using approximated closed-form expressions, and (3) BER threshold-crossing estimation via hypothesis testing [18], and its performance dependence on the degree distribution. Using these results, we show that in the regime of small BER, right-regular codes both minimize estimation MSE and maximize a lower bound on threshold-crossing detection performance. These analytical results and insights provide useful tools for designing reliable in-memory computing architectures.

II. Problem Formulation

A. Notations

We denote by $C^n(\Lambda, \Omega)$ the *ensemble* of *irregular* LDPC codes of length n and degree distributions $\Lambda(x) \triangleq \sum_{i=1}^{d_v} \Lambda_i x^i$ and $\Omega(x) \triangleq \sum_{i=1}^{d_c} \Omega_i x^i$, with Ω_i, Λ_i describing the relative portion of variable and check nodes of degree i, respectively. The class of *right-regular* codes, studied extensively in the literature (e.g. [19]–[22]), has $\Omega(x) = x^{d_c}$, and the ensemble of such codes is denoted $C^n(\Lambda, d_c)$. $m \triangleq n\Lambda'(1)/\Omega'(1)$ denotes the number of check nodes. The parity-check matrix of size $m \times n$ of $C \in C^n(\Lambda, \Omega)$ is denoted H. For any word $\boldsymbol{x} \in \{0,1\}^n$ the *syndrome* is defined by $\boldsymbol{s_x} \triangleq \mathrm{H}\boldsymbol{x}^T$ (with $\boldsymbol{s_x} = \boldsymbol{0}$ for codewords). The well known Hamming

Work supported by US-Israel Binational Science Foundation grant 2023627.

weight and Hamming distance are denoted $w_H(\cdot)$ and $d_H(\cdot,\cdot)$, respectively. $\mathrm{P}(z), \mathbb{E}[z]$ denote the probability function (PMF or PDF) and expectation of a random variable z. $O(\cdot)$ is the big-O notation [23], and $[x]_+ \triangleq \max\{0,x\}$. $F_{\mathrm{p}}(x;\lambda)$ denotes the CDF of a Poisson random variable with parameter λ.

B. Memory Channel Model

For some word $\boldsymbol{x} \in \{0,1\}^n$ stored in memory, we assume the occurrence of bit-flips, replacing $\boldsymbol{x}$ with $\boldsymbol{x}'$, the result of passing $\boldsymbol{x}$ through a memoryless binary symmetric channel (BSC) with crossover probability $p \in [0, 0.5)$. The BER is defined by $\boldsymbol{d}_H(\boldsymbol{x},\boldsymbol{x}')/n$, where $\boldsymbol{d}_H(\boldsymbol{x},\boldsymbol{x}')$ is a Binomial random variable with parameter p.

C. In-Memory BER Estimation Mechanism

Let $C_i \in \mathcal{C}(\Lambda_i,\Omega_i)$, $i=\{1,2\}$, be two codes such that $C_1 \subseteq C_2$ (meaning H_2 is obtained by a subset of $m_2 \le m_1$ rows of H_1). C_1, C_2 will be called the *correction code* and *estimation code* respectively. Let $\boldsymbol{u} \in \{0,1\}^{n-m_1}$ be an information word to be stored. The proposed mechanism is as follows.

1) ***Encoding****:* use C_1 to encode $\boldsymbol{u}$ into $\boldsymbol{c}$. Store $\boldsymbol{c}$ in a logic-in-memory capable memory.

2) ***Estimation****:* Perform a periodical in-memory computation of the *syndrome* $\boldsymbol{s} \triangleq \boldsymbol{s}_{\boldsymbol{c}'} = \mathrm{H}_2\boldsymbol{c}'^T$, and estimate $\hat{p}(\boldsymbol{s})$ (discussed in Section III). Estimate whether p exceeds a predefined tolerance p_{tol} (discussed in Section IV), and move to 3 if it does.

3) ***Decoding****:* Read $\boldsymbol{c}'$ from the memory into a resourceful processing unit, use C_1 to decode it into $\boldsymbol{c}$, and store it again.

III. BER Estimation From Syndromes

In this section we derive a maximum-likelihood (ML) estimator $\hat{p}(\boldsymbol{s})$. For completeness, we revisit several results from [17].

A. Unsatisfied Check Equations

Lemma 1. *(From [17]) The probability $p_u(i)$ that some check node of degree i is unsatisfied is given by*

$$p_u(i) = \frac{1-(1-2p)^i}{2}. \tag{1}$$

As done in [17], and shown to be very reasonable, we will assume that the satisfaction of different check nodes can be treated as independent Bernoulli trials.

B. Maximum Likelihood Estimators

We first consider the ML-estimator for right-regular codes. Based on the independent checks assumption, the syndrome's Hamming weight $w_{\boldsymbol{s}} \triangleq w_H(\boldsymbol{s})$ is Binomially distributed.

Proposition 2. *(From [17]) For a right-regular code $C \in \mathcal{C}^n(\Lambda, d_c)$, the ML estimator for the BER is*

$$\hat{p}_{\mathrm{reg}}(w_{\boldsymbol{s}}) \triangleq \frac{1}{2}\left(1-\left([1-2w_{\boldsymbol{s}}/m]_+\right)^{1/d_c}\right). \tag{2}$$

We wish to generalize this result to irregular codes. In this case, the Bernoulli trials of different check nodes have different probabilities, turning $w_{\boldsymbol{s}}$ to a Poisson-Binomial random variable, which we approximate as a Poisson random variable, as justified by [24] when the probabilities $p_u(i)$ are small. Since the Binomial distribution is a special case of Poisson-Binomial, this approximation remains relevant for the right-regular case also.

Assumption 3. *From hereon we assume $w_{\boldsymbol{s}} \sim \mathrm{Pois}(\lambda)$, with*

$$\lambda(p) \triangleq \sum_{i=1}^{d_c} m\Omega_i p_u(i) = \frac{m}{2}\left(1-\Omega(1-2p)\right), \tag{3}$$

where $m\Omega_i$ is the number of nodes with degree i.

Numerical examination shows that this assumption holds with high accuracy, but will not be discussed further here. Since $\Omega(x)$ is monotone increasing for $x \ge 0$, it is clear that $\lambda(p)$ is also monotone increasing with p, and therefore injective. We also see that $\lambda(p) \in [0, m/2]$, and note that it equals the expectation value under the exact Poisson-Binomial distribution as well.

Proposition 4. *Under Assumption 3, for an irregular code $C \in \mathcal{C}^n(\Lambda,\Omega)$, the ML estimator for the BER is*

$$\hat{p}_{\mathrm{irr}}(w_{\boldsymbol{s}}) \triangleq \frac{1}{2}\left(1-\Omega^{-1}\left([1-2w_{\boldsymbol{s}}/m]_+\right)\right), \tag{4}$$

where $\Omega^{-1}(\cdot)$ is the inverse function of the polynomial $\Omega(x)$.

Proof: (sketch) Similar to the proof of Proposition 2 in [17], using the Poisson rather than Binomial distribution and calculating $\lambda^* = \arg\max_{\lambda\in[0,m/2]} \mathrm{P}(w_{\boldsymbol{s}}=j|\lambda)$. ■

Finding the inverse $\Omega^{-1}(\cdot)$ as an algebraic expression is generally not possible, but the inversion can be performed numerically for values in $[0,1]$.

Observation 5. *For a right-regular code $C \in \mathcal{C}^n(\Lambda, d_c)$, for which $\Omega(x) = x^{d_c}$, we have $\Omega^{-1}(x) = x^{1/d_c}$. Substituting to* (4) *we get* (2)*, showing that the right-regular estimator* (2) *is a special case of the irregular estimator* (4)*.*

C. Estimation Performance

We examine the estimation MSE, defined by

$$\mathbf{mse}(p) \triangleq \mathbb{E}\left[(\hat{p}-p)^2\right] = \mathbf{var}(p) + \mathbf{bias}(p)^2,$$

where $\mathbf{var}(p)$ and $\mathbf{bias}(p)$ are the estimation variance and bias respectively. We denote $\zeta \triangleq 1-2p$ (but use both ζ and p as convenient), and then

$$\tau(\zeta) \triangleq \mathrm{P}\left(w_{\boldsymbol{s}} \le m/2\right) = F_{\mathrm{p}}\left(m/2; \lambda\left((1-\zeta)/2\right)\right), \tag{5}$$

so that $1-\tau$ is the estimator's *truncation probability*. The following theorem describes the dependence of MSE on $\Omega(x)$, for the case of small truncation probability.

Theorem 6. *For $(1-\tau) \ll \tau$, the MSE is given by*

$$\boldsymbol{mse}(p;\Omega) = \frac{\tau(\zeta)}{4}\frac{1-\Omega(\zeta^2)}{m\Omega'(\zeta)^2} + (1-\tau(\zeta))\left(\frac{\zeta}{2}\right)^2 + \eta(p), \tag{6}$$

where $\eta(p)$ represents terms that are empirically small.

Proof: See Appendix A for a proof sketch. ■

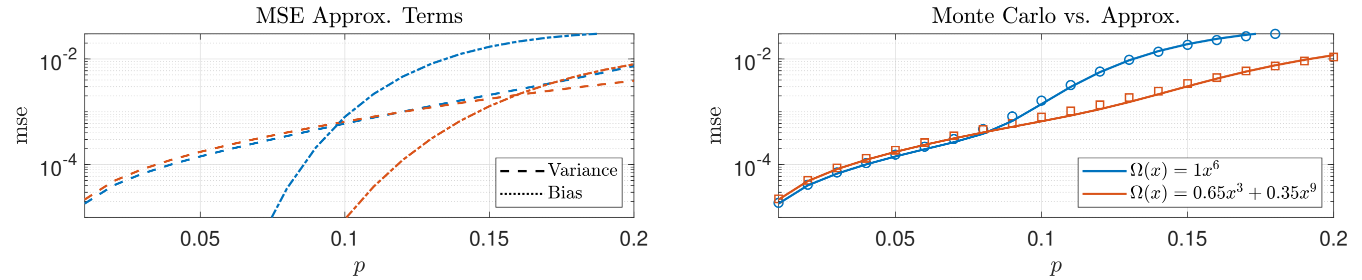

Fig. 1: **Left:** MSE approximation terms (first and second terms in (6) - dashes and dash-dots, respectively) for different $\Omega(x)$ (colors), **right:** MSE Monte-Carlo simulations (circles and squares), assuming independent checks, vs. approximation (solid).

Discussion. *The MSE arises from two distinct error mechanisms. The first term stems from the estimation variance, dominating in the non-truncated region. The second term results from estimation bias, dominating in the truncated region, given by* $\left(\frac{1}{2} - p\right) = \frac{\zeta}{2}$ *and the variance is zero.*

Fig. 1 illustrates the leftmost and middle terms in Eq. (6) (left), and demonstrates that the approximation using these terms fits tightly to corresponding Monte Carlo simulations (right). The simulations were performed based on the assumption of independent checks, by directly drawing random weights from the Poisson-Binomial distribution corresponding to $\Omega(x)$.

D. Degree Effect on MSE

We examine the approximation provided by Eq. (6) to explore the main trends of MSE dependence on $\Omega(x)$. As discussed after Theorem 6, the MSE is dominated by the first or the second term in different regions, depending on the truncation probability $1 - \tau$. Moreover, it is clear from Eq. (5) that as ζ decreases from 1 (p increases), this truncation probability increases. We use the following definition for further exploiting these properties.

Definition 7. *For a given* $\Omega(x)$ *we define the* cutoff ζ *as*

$$\zeta_{\text{cut}}(\Omega) \triangleq \zeta \;\; \text{for which} \;\; \frac{\tau(\zeta)}{4} \frac{1 - \Omega(\zeta^2)}{m\Omega'(\zeta)^2} = (1 - \tau(\zeta)) \left(\frac{\zeta}{2}\right)^2 .$$

This cutoff, seen empirically to be unique, partitions the MSE into the different regions, and is important since it defines the region in which the first MSE term is dominant ($\zeta > \zeta_{\text{cut}}$), which we aim to work in. Fig. 2-left demonstrates this partitioning by showing the first term alone for $\zeta \geq \zeta_{\text{cut}}$ (recall $p = (1 - \zeta)/2$), and the second term alone for $\zeta < \zeta_{\text{cut}}$, compared to the Monte-Carlo simulations from previous section. In the region where $\zeta > \zeta_{\text{cut}}$, we can treat $\tau(\zeta)$ as 1, and focus on the first term. We denote $a_{\text{R}} \triangleq \Omega'(1)$ as the average check degree, and use the following expansions

$$\Omega(\zeta^2) = 1 - 2a_{\text{R}}(1 - \zeta) + \left(a_{\text{R}} + 2\Omega''(1)\right)(1 - \zeta)^2 + \epsilon_1,$$
$$\Omega'(\zeta)^{-2} = a_{\text{R}}^{-2} + 2a_{\text{R}}^{-3}\Omega''(1)(1 - \zeta) + \epsilon_2,$$

where $\epsilon_1 = O\left((1 - \zeta)^3\right)$, $\epsilon_2 = O\left((1 - \zeta)^2\right)$. Through algebraic manipulation we get that the MSE is dominated by

$$\mathbf{mse}_1(\Omega) = \frac{1}{4m}\left[\frac{1 - \zeta^2}{a_{\text{R}}} + \frac{2(1 - \zeta)^2}{a_{\text{R}}^2}\Omega''(1)\right]. \quad (7)$$

Two important dependencies now become apparent:

i For right-regular codes with $d_c = a_{\text{R}}$, $\Omega(x) = x^{a_{\text{R}}}$, $\Omega''(1) = a_{\text{R}}(a_{\text{R}} - 1)$, we have

$$\mathbf{mse}_1(\Omega) = \frac{1}{4m}\left[\frac{3(1 - \zeta)(\zeta - 1/3)}{a_{\text{R}}} + 2(1 - \zeta)^2\right],$$

which is clearly monotone decreasing with a_{R} (note the caveat that ζ_{cut} also depends on a_{R}, so a_{R} cannot be increased without bound.)

ii For codes with a fixed average degree a_{R}, $\mathbf{mse}_1(\Omega)$ is monotone increasing with $\Omega''(1)$. We denote $\mathbf{var}(\Omega) \triangleq \sum_{i=1}^{d_c} i^2\Omega_i - a_{\text{R}}^2$ as the variance of the check distribution. We notice that $\Omega''(1) = \mathbf{var}(\Omega) + a_{\text{R}}(a_{\text{R}} - 1)$ and conclude that $\mathbf{mse}_1(\Omega)$ is monotone increasing with $\mathbf{var}(\Omega)$, making *right-regular codes optimal* for any given a_{R}.

These dependencies are illustrated in Fig. 3, recalling that $p = (1 - \zeta)/2$. The cutoff $p_{\text{cut}} = (1 - \zeta_{\text{cut}})/2$ can be observed as the inflection points after which the MSE increases steeply. Numerical analysis shows that for right-regular codes, $\zeta_{\text{cut}}(x^{a_{\text{R}}})$ is monotone increasing with a_{R} (also apparent in Fig. 3-left), limiting the maximal a_{R} allowing effective estimation for some target value of p.

IV. Estimating BER-Threshold Crossing

The decoding stage in the mechanism from Section II-C is based on detecting when the BER exceeds p_{tol}. In this section we examine such detection based on the ML estimators.

A. Hypothesis Testing for BER Threshold

Using the framework of *hypothesis testing*, and using $\hat{p}$ as a statistic for p, the detection problem is formalized to

$$H_0 \;:\; p \leq p_{\text{tol}}, \text{ BER is within tolerance}$$
$$H_1 \;:\; p > p_{\text{tol}}, \text{ BER exceeds tolerance.}$$

We define $\psi(x) \triangleq \frac{m}{2}(1 - \Omega(1 - 2x))$, and recall from Eqs. (3), (4) that $\psi(p) = \lambda$ and $\psi(\hat{p}) = w_s$. Since $\Omega(x)$ is monotone increasing, so is $\psi(x)$. We can thus reformulate the setting by testing whether $\lambda \leq \psi(p_{\text{tol}})$ or not, using w_s as our statistic. Following Assumption 3, w_s is a Poisson random

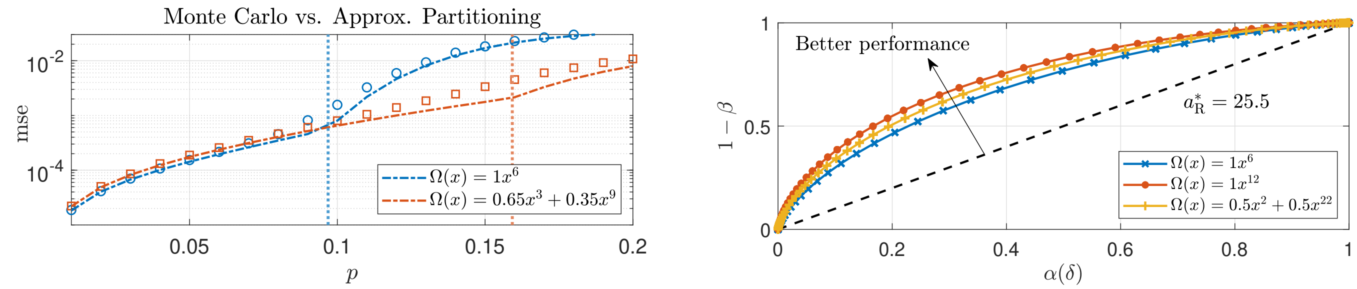

Fig. 2: **Left:** Monte-Carlo (circles and squares) vs. approximation (dashed line), where only the first and second term are used from left and right of $(1-\zeta_{\text{cut}})/2$ (vertical dotted lines), **Right:** Hypothesis testing performance, $\delta = 10^{-3}$, $p_{\text{tol}} = 10^{-2}$

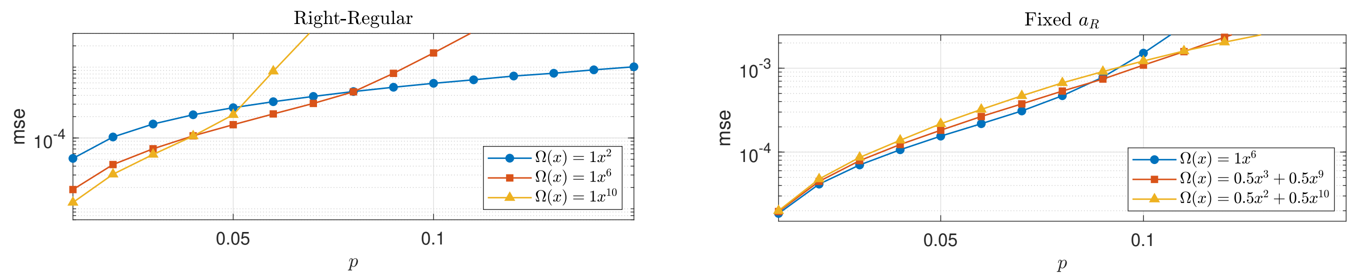

Fig. 3: **Left:** MSE for right-regular codes with different degrees, **Right:** MSE for fixed a_{R} with different distributions, $m = 100$.

variable with a CDF $F_{\text{p}}(w_{\boldsymbol{s}};\lambda)$, that meets the condition of *Karlin-Rubin theorem* [18]. Therefore, the test that rejects H_0 if and only if $w_{\boldsymbol{s}} > w_{\text{T}} \triangleq \psi(p_{\text{T}})$ is *uniformly most powerful.*

This means that, following conventional notations

$$\alpha \triangleq \text{P(reject } H_0|H_0) = \text{P}\left(w_{\boldsymbol{s}} > \psi(p_{\text{T}})|\lambda \le \psi(p_{\text{tol}})\right),$$
$$\beta \triangleq \text{P(accept } H_0|H_1) = \text{P}\left(w_{\boldsymbol{s}} \le \psi(p_{\text{T}})|\lambda > \psi(p_{\text{tol}})\right),$$

for every given significance level α, the *power* $1-\beta$ is maximal with respect to any other test. We modify the type-I error into

$$\alpha(\delta) = \text{P}\left(w_{\boldsymbol{s}} > \psi(p_{\text{T}})|\lambda \le \psi(p_{\text{tol}} - \delta)\right),$$

that is, to include an *estimation gap* of width δ where both hypotheses are "equally acceptable", and neither counts as error. The gap δ can have different practical justifications, one of which is that decoding close to the threshold has a benefit of improving data quality. Reasonably, it is desired to set a maximum allowed value for β since it represents the probability of deciding not to decode when needed, which may cause irreparable data corruption. Naturally, we will set $p_{\text{T}} < p_{\text{tol}}$, implying that

$$\beta < \text{P}\left(w_{\boldsymbol{s}} \le \psi(p_{\text{T}})|\lambda = \psi(p_{\text{tol}})\right) = F_{\text{p}}\left(\psi(p_{\text{T}});\psi(p_{\text{tol}})\right), \quad (8)$$

where the inequality is due to $F_{\text{p}}(x;\lambda)$ being monotone decreasing with λ. Thus, to meet an upper bound β^* we require

$$F_{\text{p}}\left(\psi(p_{\text{T}});\psi(p_{\text{tol}})\right) \triangleq \beta_{\psi}(p_{\text{T}}, p_{\text{tol}}) \le \beta^*. \quad (9)$$

We also have that

$$\alpha(\delta) \le 1 - F_{\text{p}}\left(\psi(p_{\text{T}});\psi(p_{\text{tol}} - \delta)\right) \triangleq \alpha_{\psi}(\delta, p_{\text{T}}, p_{\text{tol}}). \quad (10)$$

B. Degree Effect on Detection Performance

Since a minimal $\alpha(\delta)$ is desired for any power $1-\beta$, we define $D_{\psi}(\delta, p_{\text{T}}, p_{\text{tol}}) \triangleq 1-\beta-\alpha$ as a detection performance measure we aim to maximize. We now describe the main result of this section. Denote $\Delta \triangleq \psi(p_{\text{tol}})-\psi(p_{\text{tol}}-\delta) > 0$, and define

$$D^*(\Delta) \triangleq \sum_{j=0}^{\lfloor \psi(p_{\text{T}}) \rfloor} \frac{e^{-\psi(p_{\text{tol}})}}{j!}\left[e^{\Delta}\left(\psi(p_{\text{tol}}) - \Delta\right)^j - \psi(p_{\text{tol}})^j\right].$$

Theorem 8. *For every* $(\delta, p_{\text{T}}, p_{\text{tol}})$ *such that* $0 < \delta < p_{\text{tol}} - p_{\text{T}}$,

1) *it holds that* $D_{\psi}(\delta, p_{\text{T}}, p_{\text{tol}}) > D^*(\Delta)$,
2) *for fixed* $\lfloor \psi(p_{\text{T}}) \rfloor$, $D^*(\Delta)$ *is monotone increasing with* Δ.

Proof: See Appendix B. ∎

Lemma 9. *It holds that*

$$\Delta = m\delta\left[a_{\text{R}} - 2p_{\text{tol}}\Omega^{''}(1)\right] + O(mp_{\text{tol}}^2). \quad (11)$$

Proof: The result is obtained by expanding Δ and $\Omega^{'}(1-2p_{\text{tol}})$ to a first-order Taylor series. ∎

The importance of Lemma 9 is that given a tolerance parameter δ, it shows that maximizing the term $\left[a_{\text{R}} - 2p_{\text{tol}}\Omega^{''}(1)\right]$ in (11) maximizes a lower bound on the detection performance, because $D^*(\Delta)$ is monotone increasing in Δ, and $1-\beta-\alpha \ge D^*$. Since the exact dependence of $1-\beta-\alpha$ on $\Omega(\cdot)$ is complicated, this lower-bound maximization becomes an efficient and useful design tool. From Theorem 8 and Lemma 9 we learn that the detection performance bound depends on $\Omega(x)$ as follows:

i For right-regular codes an optimal performance bound is obtained by $a_{\mathrm{R}}{}^{*} = \arg\max_{a_{\mathrm{R}}} \{a_{\mathrm{R}} - 2p_{\mathrm{tol}} a_{\mathrm{R}}(a_{\mathrm{R}}-1)\} = (1+2p_{\mathrm{tol}})/(4p_{\mathrm{tol}})$.

ii For codes with a fixed average degree a_{R}, Δ is monotone decreasing with $\mathbf{var}(\Omega)$, and optimal performance bound is obtained for right-regular codes, as in the case of $\mathbf{mse}_1$ from Section III.

Fig. 2-right demonstrates these results.

V. In-Memory BER Estimation

Based on the theoretical results of the preceding sections, we now propose a scheme for employing BER estimation for in-memory computing.

A. Architecture

The estimation stage is based on embedding logic circuitry in the memory, allowing to perform m_2 parity computations with at most $d_c - 1$ XORs per computation. Additional *simple* logic is embedded to extract the weight $w_{\boldsymbol{s}}$ (simple sum), compare $w_{\boldsymbol{s}}$ to the threshold $\psi(p_{\mathrm{T}})$, and translate $w_{\boldsymbol{s}}$ into $\hat{p}(\boldsymbol{s})$. The latter is needed only if we want an actual BER estimate, while for some applications detection of threshold crossing is sufficient. An illustration for this mechanism is given in Fig. 4.

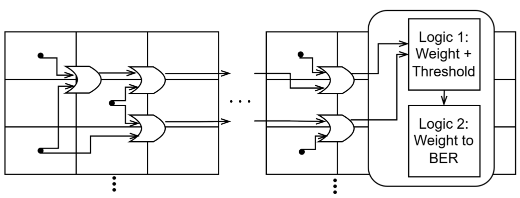

Fig. 4: Illustration of in-memory BER estimation architecture.

Many works discuss efficient in-memory processing and data placement (e.g [7], [9]). Exploring these considerations is outside the scope of this work and left for future research.

The decoding stage is based on setting p_{tol} smaller than the *correction capability* [25] of C_1, such that traditional message-passing decoding [26] can reproduce $\boldsymbol{c}$ with high probability.

B. Estimation Code Design

This section leverages the estimation and hypothesis testing performance results from previous sections to design C_1 and decision threshold p_{T}. We focus on the regime where p is sufficiently small, ensuring the accuracy of all approximations. To best serve the proposed architecture, our strategy is to optimize detection performance, while still working at the non-truncated regime of MSE (below the inflection point).

We consider p_{tol}, δ and β^* as predefined system-level parameters. The design procedure is as follows.

Procedure 1. (estimation code design)

1) Define $\tilde{p}_{\mathrm{tol}} \triangleq \max_{k\in\mathbb{N}}\{2/(2k+1)\}$ s.t. $2/(2k+1) \le p_{\mathrm{tol}}$
2) Set $d_c = (1+2\tilde{p}_{\mathrm{tol}})/(4\tilde{p}_{\mathrm{tol}})$
3) Set $\Omega(x) = x^{d_c}$
4) Calculate $p_{\mathrm{cut}} \triangleq \left(1 - \zeta_{\mathrm{cut}}\left(x^{d_c}\right)\right)/2$
5) Set $\mu = \min\{p_{\mathrm{tol}} - \delta, p_{\mathrm{cut}}\}$
6) Set $p_{\mathrm{T}} = \max_{\rho\in[0,\mu]}\{\rho\}$ such that $\beta_\psi(\rho, p_{\mathrm{tol}}) \le \beta^*$

Discussion. *In Eq.* (7), (11) *we saw that* right-regular *codes have optimal performance, provided that* $a_{\mathrm{R}}{}^{*} = (1+2p_{\mathrm{tol}})/(4p_{\mathrm{tol}})$ *is an integer and can serve as the actual degree. We can take a lower tolerance* $\tilde{p}_{\mathrm{tol}}$ *that ensures an integer* $d_c = a_{\mathrm{R}}{}^{*}$*, and ensure optimal detection performance in terms of* $D^*(\Delta)$*, while only slightly underestimating the code's correction capability if* $\tilde{p}_{\mathrm{tol}} \approx p_{\mathrm{tol}}$*. We then set* p_{T}*. Since* $\alpha(\delta)$ *decreases as* p_{T} *increases, we aim to maximize it, but need to maintain* $\Delta > 0$ *(requiring* $p_{\mathrm{T}} < p_{\mathrm{tol}} - \delta$*),* $p_{\mathrm{T}} < p_{\mathrm{cut}}$ *so that the MSE is not in the truncated region, and* $\beta < \beta^*$ *as required.*

Appendix A
Proof Sketch of Theorem 6

Proof: We denote $g(x) \triangleq 1 - 2x/m$, $\mathbb{E}_c[X] = \mathbb{E}[X|w_{\boldsymbol{s}} \le m/2]$, and $\mathbf{var}_c(X)$ as the variance using $\mathbb{E}_c[X]$. η_1, η_2 will denote empirically small terms. Using Taylor's expansion for $\Omega^{-1}(g(w_{\boldsymbol{s}}))$ we get that $\mathbb{E}_c\left[\Omega^{-1}(g(w_{\boldsymbol{s}}))\right] = \zeta + \eta_1(p)$. It can be then shown that

$$\mathbf{bias}(p)^2 = \left(\frac{\zeta}{2}\right)^2 - \frac{\zeta}{2}\tau\mathbb{E}_c\left[\Omega^{-1}(g(w_{\boldsymbol{s}}))\right] + \frac{\tau^2}{4}\mathbb{E}_c\left[\Omega^{-1}(g(w_{\boldsymbol{s}}))\right]^2,$$

$$\mathbf{var}(p) = \frac{\tau}{4}\mathbf{var}_c\left(\Omega^{-1}(g(w_{\boldsymbol{s}}))\right) + \frac{\tau-\tau^2}{4}\mathbb{E}_c\left[\Omega^{-1}(g(w_{\boldsymbol{s}}))\right]^2.$$

Finally, it can be shown that

$$\mathbf{var}_c\left(\Omega^{-1}(g(w_{\boldsymbol{s}}))\right) = \frac{4\mathbf{var}(w_{\boldsymbol{s}})}{\left[m\Omega'\left(\Omega^{-1}(g(\lambda))\right)\right]^2} + \eta_2(p),$$
$$\mathbf{var}(w_{\boldsymbol{s}}) = m/4\left[1 - \Omega(\zeta^2)\right].$$

Taking $\mathbf{mse}(p) = \mathbf{bias}(p)^2 + \mathbf{var}(p)$ completes the proof. ∎

Appendix B
Proof of Theorem 8

From Eq. (8), (9), (10), we have

$$\begin{aligned} D_\psi(\delta, p_{\mathrm{T}}, p_{\mathrm{tol}}) &> 1 - \beta_\psi(p_{\mathrm{T}}, p_{\mathrm{tol}}) - \alpha_\psi(\delta, p_{\mathrm{T}}, p_{\mathrm{tol}}) \\ &= F_{\mathrm{p}}\left(\psi(p_{\mathrm{T}}); \psi(p_{\mathrm{tol}} - \delta)\right) - F_{\mathrm{p}}\left(\psi(p_{\mathrm{T}}); \psi(p_{\mathrm{tol}})\right) \\ &= \sum_{j=0}^{\lfloor\psi(p_{\mathrm{T}})\rfloor} e^{-\psi(p_{\mathrm{tol}}-\delta)}\frac{\psi(p_{\mathrm{tol}}-\delta)^j}{j!} - e^{-\psi(p_{\mathrm{tol}})}\frac{\psi(p_{\mathrm{tol}})^j}{j!} \\ &= \sum_{j=0}^{\lfloor\psi(p_{\mathrm{T}})\rfloor} e^{-\psi(p_{\mathrm{tol}})-\Delta}\frac{(\psi(p_{\mathrm{tol}})-\Delta)^j}{j!} - e^{-\psi(p_{\mathrm{tol}})}\frac{\psi(p_{\mathrm{tol}})^j}{j!}, \end{aligned}$$

which is exactly $D^*(\Delta)$. Taking the derivative while keeping $\lfloor\psi(p_{\mathrm{T}})\rfloor$ fixed gives

$$\frac{dD^*}{d\Delta} = \sum_{j=0}^{\lfloor\psi(p_{\mathrm{T}})\rfloor} \frac{e^{-\psi(p_{\mathrm{tol}})}}{j!}\left[e^{\Delta}\left(\psi(p_{\mathrm{tol}})-\Delta\right)^j\left(\psi(p_{\mathrm{tol}})-\Delta-j\right)\right],$$

we notice that for every $\delta < p_{\mathrm{tol}} - p_{\mathrm{T}}$ it holds that $j \le \psi(p_{\mathrm{T}}) < \psi(p_{\mathrm{tol}} - \delta) < \psi(p_{\mathrm{tol}}) - \Delta$, ensuring all terms remain positive.

REFERENCES

[1] A. Sebastian, M. Le-Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nature Nanotechnology, Vol. 15*, 2020.

[2] S. A. McKee, "Reflections on the memory wall," *Conference on Computing Frontiers*, 2004.

[3] S. Aga, S. Jeloka, A. Subramaniyan, S. Narayanasamy, D. Blaauw, and R. Das, "Compute caches," *IEEE International Symposium on High Performance Computer Architecture*, 2017.

[4] S. Li, D. Niu, K. T. Malladi, H. Zheng, B. Brennan, and Y. Xie, "Drisa: A dram-based reconfigurable in-situ accelerator," *IEEE/ACM International Symposium on Microarchitecture*, 2017.

[5] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, "'memristive' switches enable 'stateful' logic operations via material implication," *Nature, Vol. 464*, 2010.

[6] H. Mahmoudi, T. Windbacher, V. Sverdlov, and S. Selberherr, "Implication logic gates using spin-transfer-torque-operated magnetic tunnel junctions for intrinsic logic-in-memory," *Solid-State Electronics, Vol. 84*, 2013.

[7] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Magic—memristor-aided logic," *IEEE Trans. on Circuits and Systems II: Express Briefs, Vol. 61, No. 11*, 2014.

[8] C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductor memory applications: A state-of-the-art review," *IBM Journal of Research and development, Vol. 28*, 1984.

[9] O. Leitersdorf, B. Perach, R. Ronen, and S. Kvatinsky, "Efficient error-correcting-code mechanism for high-throughput memristive processing-in-memory," *IEEE Press*, 2021.

[10] R. Gallager, "Low-density parity-check codes," *IRE Trans. on Information Theory, Vol. 8, No. 1*, 1962.

[11] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. on Information Theory, Vol. 47, No. 2*, 2001.

[12] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. on Information Theory, Vol. 55, No. 7*, 2009.

[13] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Trans. on Information Theory, Vol. 59, No. 10*, 2013.

[14] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," *IEEE European Test Symposium*, 2013.

[15] G. Zervakis, H. Saadat, H. Amrouch, A. Gerstlauer, S. Parameswaran, and J. Henkel, "Approximate computing for ML: State-of-the-art, challenges and visions," *Asia and South Pacific Design Automation Conference*, 2021.

[16] E. Sharon and A. Bazarsky, "Dynamic memory error model estimation for read and ECC adaptations," *Proc. Non Volatile Memories Workshop, San Diego, USA*, 2017.

[17] G. Lechner and C. Pacher, "Estimating channel parameters from the syndrome of a linear code," *IEEE Communications Letters, Vol. 17, No. 17*, 2013.

[18] G. Casella and R. L. Berger, *Statistical Inference*. Thomson Learning, 2002.

[19] M. A. Shokrollahi, "New sequences of linear time erasure codes approaching the channel capacity," *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, 1999.

[20] O. Barak, D. Burshtein, and M. Feder, "Bounds on achievable rates of LDPC codes used over the binary erasure channel," *IEEE Trans. on Information Theory, Vol. 50, No. 10*, 2004.

[21] E. Paolini and M. Chiani, "On the threshold of right regular LDPC codes for the erasure channel," *IEEE 61st Vehicular Technology Conference*, 2005.

[22] P. Oswald and A. Shokrollahi, "Capacity-achieving sequences for the erasure channel," *IEEE Trans. on Information Theory, Vol. 48, No. 12*, 2002.

[23] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, fourth edition*. MIT Press, 2022.

[24] J. L. J. Hodges and L. Le-Cam, "The poisson approximation to the poisson binomial distribution," *Ann. Math. Statist. 31(3): 737-740*, 1960.

[25] T. J. Richardson and R. L. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2007.

[26] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. on Information Theory, Vol. 47, No. 2*, 2001.