Coding with Side Information for Radiation-Tolerant Memory Devices

Euiseok Hwang,* Seungjune Jeon,* Rohit Negi,* B. V. K. Vijaya Kumar,* and Michael K. Cheng†

ABSTRACT. — Memory devices aboard spacecraft experience radiation-induced errors either in the form of temporary upsets (soft errors) or permanent defects (hard or stuck-at errors). Error-correcting codes (ECCs) are used to recover memory content from errors where defective cells are either regarded as erasures by the decoder or entire blocks containing defective cells are marked as unusable. In this article, alternative coding schemes are investigated for memory devices in space, where the encoder is provided with the locations of the defective cells, denoted by side information. This coding approach has the potential to improve the overall storage capacity of memory devices, since the information theoretic capacity of a channel where side information is only available at the encoder is the same as the capacity where side information is available at both the encoder and decoder. Spacecraft memory controllers typically scrub memory devices periodically for errors. Partial side information can be obtained during this scrubbing process by comparing the ECC decoder output with its input and thereby avoid the need for additional cell tests or storage overhead. In between scrubbings, the encoder can use this partial side information to account for permanent defects to improve reliability or to increase the storage capacity of onboard memory devices. In order to achieve performance gains for practical memory systems, several coding schemes that adaptively incorporate the codeword with the known side information are proposed in this article. The proposed coding schemes are evaluated by numerical simulations on a memory channel model characterized by soft and hard errors. Simulation results show that while coding with complete side information at the encoder offers the most performance gain compared to when coding without side information is used, coding with partial side information can close the gap between the optimal and current approach without incurring much additional overhead.

I. Introduction

Memory devices aboard spacecraft are susceptible to radiation-induced errors. Some memory cells will only experience temporary bit reversals called soft errors. But if the radiation dose is strong enough, a memory cell can become permanently damaged and remain stuck at a fixed value [1]. The radiation dosage is not uniform in space and is very heavy during

* Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania.

† Communications Architectures and Research Section.

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

periods of sunspots or in certain regions such as near Jupiter. To protect against radiation-induced errors, memory controllers aboard spacecraft periodically "scrub" memory devices for errors and update memory contents with the error-correcting code (ECC) decoder output [2,3]. By scrubbing a memory device, soft errors can be removed by rewriting the cells; however, stuck-at bits remain and accumulate over time. In legacy memory systems, the defective cells are treated simply as soft errors, which makes error correction inefficient because stuck-at bits cannot be corrected. Alternatively, an entire memory block can be marked unusable even if only a few cells in the block are defective. Although these approaches have low overhead, they do not make the best use of all available storage area. If the location of the stuck-at bits, denoted by side information, is available at the decoder, the stuck-at bits can be set as erasures to make ECC decoding most efficient [4–6].

In this article, we take a look at another coding approach that takes into account the location of the stuck-at bits during encoding. We model error behavior on a memory device as a channel with state-dependent error transitions and analyze the error performance when side information is available at the encoder. We refer to our problem setup as channel coding with side information at the encoder (CSIE). Similar problems in the area of coding with side information at the transmitter have been studied [7–9]. Figure 1 shows a block diagram of channel coding problems with side information, where S represents side information related to cell states, and S_e and S_d are side information available at the encoder and decoder, respectively. W, X, and Y represent the message, the codeword, and the received word. $\hat{W}$ denotes the message estimate. Theoretical storage capacity for CSIE ($S_e = S$ and $S_d = \emptyset$) has been shown to be equal to the capacity of a channel with side information available at both the encoder and decoder ($S_e = S_d = S$) [10,11]. Side information on the locations of the stuck-at bits can be made available to the encoder through repeated reading and writing of a constant pattern and its complement to a memory block without requiring additional storage. CSIE schemes based on binning or partitioning the codewords have been proposed [12,13]. However, the analyses were theoretical and the devised schemes were not practical for implementation. In the following sections, we propose two CSIE schemes amenable to practical implementation. To avoid the additional overhead of identifying stuck-at bits, we also propose to obtain *partial* side information by comparing the ECC decoder output with the input during every memory scrubbing. The side information extracted from the decoder is not complete because not all corrected errors are stuck-at errors and not all stuck-at errors are identified. Since stuck-at errors accumulate over time, the decoder-identified errors will be dominated by hard errors, and therefore we expect to obtain performance gain even when the side information is not complete [14]. Based on a memory channel model developed in [4] that accounts for radiation effects, we evaluate the performance of CSIE schemes when applied to memory devices in a radiation environment. Results show that CSIE can extend the lifetime of memory devices in space. Moreover, coding with decoder-provided partial side information also shows benefits when compared to coding without information while incurring little overhead. Preliminary investigation results of CSIE for memory systems were presented by the authors in [14].

The rest of the article is organized as follows. In Section II, we provide a background on channel modeling of radiation effects in space and coding with side information. In Sections III and IV, we present our CSIE scheme and evaluate the performance of various CSIE approaches. In Section V, we summarize our findings.

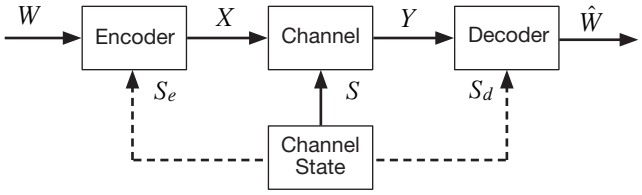

Figure 1. Coding with channel side information. S is the side information observed at the channel; S_e is the side information available at the encoder; S_d is the side information available at the decoder. A message W is encoded and mapped to a codeword X. After the channel the transmitted codeword becomes Y and is decoded to the estimated message $\hat{W}$.

II. Channel Coding with Side Information

A. Channel Model

A state-dependent channel model for memory systems in a space radiation environment is shown in Figure 2. Hard and soft error probabilities over a time period T are defined as [4]

$$q_T = 1 - e^{-\lambda_h T} \tag{1}$$

$$p_T = \left(1 - e^{-2\lambda_s T}\right)/2 \tag{2}$$

where λ_h and λ_s are hard and soft error rates (errors/bit/day). In this model, all hard errors are assumed to be either stuck at 0 or stuck at 1 with equal probability. A cell could be in one of three states $\left(s = \{\alpha, 1, 0\}\right)$: normal, stuck at 1, or stuck at 0, and each has a probability of $1 - q_T$, $q_T/2$, and $q_T/2$, respectively. The channel output, y, of a cell depends on its input x and state s, and is defined as

$$y = \begin{cases} x + z, & s = \alpha \text{ (no hard error)} \\ s, & s \neq \alpha \text{ (hard error)} \end{cases} \tag{3}$$

where $z \in \{0, 1\}$ is a soft error with $p_T = \Pr(z = 1 \mid s = \alpha)$.

B. Theoretical Capacity of Channel Coding With Side Information

For state-dependent memory channels, the channel capacities with *no* side information and with *complete* side information at the encoder and decoder are given as $C_{\min}$ and $C_{\max}$ in [11]. $C_{\min}$ is based on the binary symmetric channel (BSC) with cross-over probability $(1 - q_T)p_T + q_T/2$, while $C_{\max}$ is based on the binary erasure and error channel (BEEC) model with erasure probability q_T and error probability $(1 - q_T)p_T$. The capacity C_{enc} of the channel where only the encoder is provided with *complete* side information $(S_e = S, S_d = \emptyset)$ was shown to be equal to $C_{\max}$ in [11].

In memory devices where the controller periodically scrubs for errors, the ECC decoder can provide *partial* side information to the encoder for use to improve the overall error performance. We can assume that the bit positions where error correction have occurred are

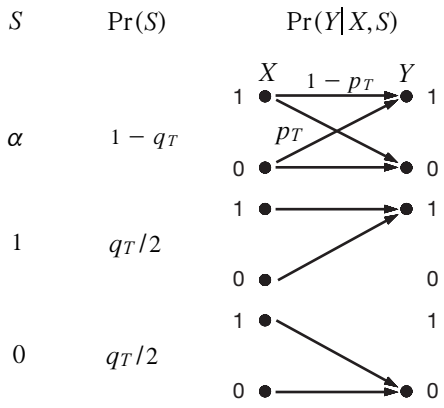

Figure 2. A state-dependent channel model for memory devices exposed to space radiation. Over a time period T, the hard (permanent) error probability is q_T and the soft (temporary) error probability is p_T.

the locations of hard errors. This assumption is not entirely correct because not all errors are hard errors and those stuck-at bits that agree with the information are not identified as hard errors. Therefore, comparing the decoder output and input only provides *partial* and not *complete* side information. However, hard errors will accumulate over time to become the dominate error source, thereby making our assumption relatively accurate. The capacity of channels with *partial* side information is unknown in general [15]; however, we can compute a tight upper bound. The sequence of obtaining partial side information from the decoder and forwarding this information to the encoder during a scrubbing period is illustrated in Figure 3.

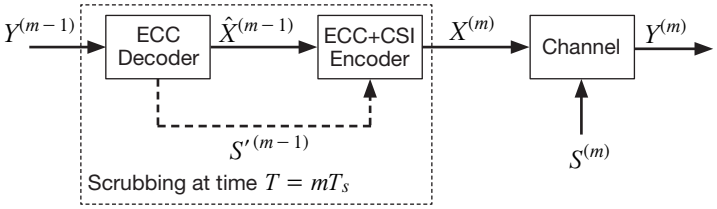

Figure 3. Memory scrubbing with partial side information at the encoder with a memory scrubbing period of duration T_s.

We can decompose this "memory scrubbing channel" into a set of binary asymmetric channels (BACs), each with transition probabilities that depend on the side information provided by comparing the decoder output to its input. We develop this modified state-dependent channel, shown in Figure 4, by defining a state descriptor S' that can take on one of four possible states $\{Y\hat{X} = 11, 10, 01, 00\}$ depending on the action of the decoder. The input to the decoder is represented by Y and output by $\hat{X}$. So $S' = \{11\}$ or $\{00\}$ indicates that a bit remained the same before and after the decoder while $S' = \{10\}$ or $\{01\}$ indicates that the bit changed. Again, we assumed that the number of errors are within the correction bound of the ECC. Two possible scenarios lead to a bit remaining constant before and after the decoder. One occurs when the bit is not stuck at a value and does not experience any error, and this event occurs with a probability of $r_T(1 - p_T)$, where $r_T = (1 - q_T)$. The other occurs when the bit is stuck at 1 or 0, and this event occurs with a probability of $q_T/2$.

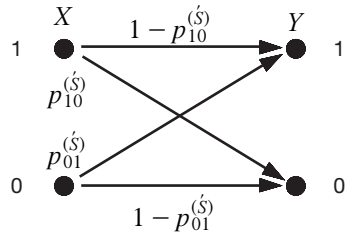

Figure 4. A binary asymmetric channel (BAC) model for the modified states $\{S'\}$. The state transition probabilities are listed in Table 1.

Two possible scenarios lead to a bit being changed after decoding. One occurs when the bit is not stuck at a value but experiences an error, and this event occurs with a probability of $r_T p_T$. The other occurs when the bit becomes stuck at a value before decoding, and this event occurs with a probability of $q_T/2$. From this description, we can fill in the $\Pr(S')$ column of Table 1. Next, we considered the channel transitions. Without loss of generality, we assumed that the channel was in state $S' = \{11\}$ and a "1" was written to this bit position, but this bit became a "0" on the memory device. This scenario, marked by transition p_{10}, occurs only when the bit was not stuck at 1 and the bit experienced a soft error and occurs with probability $[(r_T(1-p_T)/2)/\Pr(S' = \{11\})]p_T$. Continuing with the same state $S' = \{11\}$, we considered the scenario where a "0" was written to this bit location but the bit became a "1" on the memory device. This event, marked by transition p_{01}, occurs when the bit was stuck at 1 or when the bit was not stuck at 1 and experienced a soft error, and occurs with probability $[(q_T/4\Pr(S' = \{11\})] + [(r_T(1-p_T)/2)/\Pr(S' = \{11\})]p_T$. By similar reasoning, we can fill out the remaining entries in Table 1.

Table 1. Transition probabilities for the state-dependent channel model illustrated in Figure 4. There are four possible states and the crossover probabilities vary depending on the channel state.

S'	$\Pr(S')$	$p_{10}^{(S')}$	$p_{01}^{(S')}$
11	$\frac{2r_T(1-p_T)+q_T}{4}$	$\frac{2r_T(1-p_T)p_T}{2r_T(1-p_T)+q_T}$	$\frac{2r_T(1-p_T)p_T+q_T}{2r_T(1-p_T)+q_T}$
10	$\frac{2r_Tp_T+q_T}{4}$	$\frac{2r_Tp_T^2}{2r_Tp_T+q_T}$	$\frac{2r_Tp_T^2+q_T}{2r_Tp_T+q_T}$
01	$\frac{2r_Tp_T+q_T}{4}$	$\frac{2r_Tp_T^2+q_T}{2r_Tp_T+q_T}$	$\frac{2r_Tp_T^2}{2r_Tp_T+q_T}$
00	$\frac{2r_T(1-p_T)+q_T}{4}$	$\frac{2r_T(1-p_T)p_T+q_T}{2r_T(1-p_T)+q_T}$	$\frac{2r_T(1-p_T)p_T}{2r_T(1-p_T)+q_T}$

When there is *no* side information available at either the encoder or the decoder, the modified channel is equivalent to the original channel of Figure 2 and the channel capacities are equivalent; i.e., $C_{\min}^{\{S'\}} = C_{\min}$. With *complete* side information available at both the encoder and the decoder, the capacity of the modified channel, denoted by $C_{\max}^{\{S'\}}$, can be computed as

$$C_{\max}^{\{S'\}} = \sum_{S'} \Pr(S')_{p(x|S')}^{\max} [I(X;Y)|S'] \tag{4}$$

The theoretical capacity of the modified channel with *partial* side information at the encoder is upper bounded by $C_{\max}^{\{S'\}}$. To compare the channel capacities for specific error rate parameters, we fixed the soft error probability at $p_T = 10^{-6}$, varied the hard error probability in the range of $10^{-4} \leq q_T \leq 10^{-1}$, and plotted the capacity curves in Figure 5. The curves $C_{\max}$ and $C_{\min}$ are the channel capacities for the original channel of Figure 2 when *complete* side information is available at the encoder and when *no* side information is available at all. The $C_{\max}^{\{S'\}}$ curve is the upper bound capacity of the modified channel with *complete* side information available at the encoder. Thus, the capacity of the original channel when only *partial* side information is available at the encoder is somewhere in between the $C_{\max}^{\{S'\}}$ and $C_{\min}$ curves.

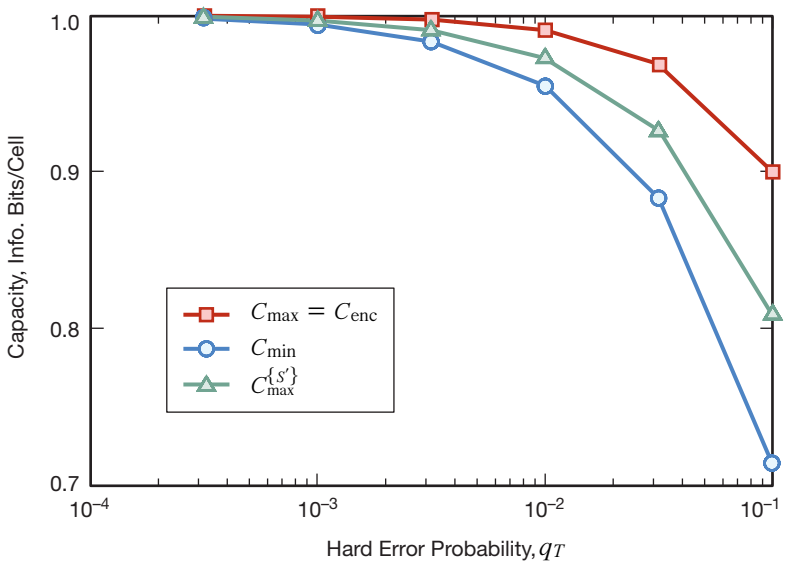

Figure 5. Channel capacities when coding with and without side information. The soft error probability p_T is fixed at 10^{-6}.

III. Coding Schemes Using Side Information at the Encoder

Binning schemes have been used to incorporate side information and protect data against stuck-at errors [13,16,17]. In this approach, codewords in a code are randomly assigned into equal-sized bins and each bin is uniquely associated with a message. To encode a message, the encoder selects from the bin associated with the message a codeword $\mathbf{x}$ such that $\| \mathbf{x} \circ \mathbf{s} - \mathbf{x} \|$ is minimized, where $\mathbf{s}$ is a channel state vector and $\| \cdot \|$ is the Hamming weight of the vector. We define $\mathbf{x} \circ \mathbf{s} = \mathbf{u}$ such that

$$u_i = \begin{cases} x_i, & s = \alpha \text{ (no hard error)} \\ s_i, & s \neq \alpha \text{ (hard error)} \end{cases} \tag{5}$$

Random binning methods provide capacity-approaching performance for channels with an arbitrarily long codeword [9,11]; however, they are not practical in real application. We propose low-complexity alternatives to code with side information, where a codeword is reprocessed by predefined rules to avoid errors from stuck-at defects. For a message $\mathbf{w}$ and a channel state vector $\mathbf{s}$, the encoding function finds a codeword $\mathbf{x}_j = f_e(\mathbf{w}, \mathbf{s})$ such that

$$\mathbf{x}_j \circ \mathbf{s} = \mathbf{x}_j \tag{6}$$

where a reprocessing index j specifies a unique mapping from $\mathbf{w}$ to $\mathbf{x}_j$. The decoder undoes the reprocessing to obtain an index $\hat{j}$ and recover an estimate of the message, $\hat{\mathbf{w}} = f_d(\mathbf{y}, \hat{j})$, where $\mathbf{y}$ is a vector of the channel output in Equation (3). We introduce two CSIE schemes: one using linear feedback shift registers (LFSRs) in Section III.A, and the other using XOR functions in Section III.B. To simplify our analysis, we initially set the soft error rates to 0. In Section III.C, we combine CSIE with Bose-Chaudhuri-Hocquenghem (BCH) codes to account for both hard and soft errors.

A. An LFSR Scheme

The LFSR scheme shifts a codeword through a predefined linear feedback shift register until the shifted codeword agrees with the side information by satisfying Equation (6). The number of shifts $v \in \{1, ..., \infty\}$ corresponds to a reprocessing index and is recorded together with the shifted codeword. For a given number of hard errors l_s, the number of shifts required to find a matching codeword is a geometric random variable with parameter $1/2^{l_s}$. Thus, the mean number of shifts, $E[v \mid l_s]$, is 2^{l_s} and requires l_s bits to represent. If we limit the maximum number of shifts to $v_{\max}$, there may not exist a shifted codeword among the possible shifts that will satisfy Equation (6). When this occurs, the encoder finds a minimum distance codeword. Assuming that no soft error occurs in the number of shifts recorded, the probability of an encoding failure can be approximated as $P_{e,L} = \left(1 - 1/2^{l_s}\right)^{v_{\max}}$. The overhead for recording the number of shifts is $\log_2(v_{\max})$. The code rate is $R_L = 1 - \lceil \log_2(v_{\max}) \rceil / n$, where n is the length of a codeword.

B. An Exclusive-or (XOR) Coding Scheme

The codeword that agrees with the side information can also be found by the XOR addition of the message and a vector taken from a predefined set of patterns. The index associated with the XORed pattern is represented in binary and recorded in the parity bits of the codeword. Selective XORing with an all-one vector [18] has been proposed as a pre-storage protection for memory systems, where the recorded data are retrieved and selectively rewritten with a complemented codeword that accounts for detected errors. Partitioned linear block codes (PLBC) [13] also use the XOR scheme with a vector minimizing $\| \mathbf{x} \circ \mathbf{s} - \mathbf{x} \|$ by searching for a candidate vector formed by taking linear combinations of selected rows from a generator matrix. Our XOR scheme uses less overhead than PLBC and can be combined with an outer ECC to recover from temporary errors.

The encoding method of the XOR coding scheme finds a vector $\mathbf{a}_j$ such that $\mathbf{x}_j = \mathbf{w} + \mathbf{a}_j$ satisfies Equation (6). The pattern vector $\mathbf{a}_j$ is selected from a set of predefined vectors $\mathbf{A}$. The code rate is $R_X = 1 - \lceil \log_2 |\mathbf{A}| \rceil / n$, where $|\mathbf{A}|$ is the size of $\mathbf{A}$. Let $\mathbf{A}_l$ be the set of vec-

tors that can encode up to as many as l stuck-at errors. For any stuck-at pattern $\mathbf{s}_{\le l}$ with no more than l stuck-at errors, there exists a vector $\mathbf{a}_j \in \mathbf{A}_l$ such that

$$(\mathbf{w} + \mathbf{a}_j) \circ \mathbf{s}_{\le l} = \mathbf{w} + \mathbf{a}_j \tag{7}$$

for all $\mathbf{w}$ and $\mathbf{s}_{\le l}$. In order to achieve efficient encoding, we need to find a compact $\mathbf{A}_l$ that satisfies Equation (7). For $l = 1$, $\mathbf{A}_1 = \{\mathbf{0}, \mathbf{1}\}$ and the XOR coding scheme is equivalent to the prestorage protection scheme. For $l \ge 2$, $\mathbf{A}_l$ can be obtained from the parity check matrix $\mathbf{H}$ of a (n,k,d) linear block code (LBC) with minimum distance $d = l + 1$. $\mathbf{H}$ by definition has l linearly independent columns and any l or fewer distinct columns taken from $\mathbf{H}$ cannot sum to $\mathbf{0}^T$. Since row rank equals column rank, the submatrix with l columns of $\mathbf{H}$, denoted by $\mathbf{H}_l$, has at least l independent rows. Let $\mathbf{R}_l$ be a submatrix of $\mathbf{H}_l$ with l independent rows. $\mathbf{R}_l$ forms an $l \times l$ full-rank matrix and spans the vector space $\mathbf{V}_l$ of all the binary l-tuples. Likewise, the set of vectors generated by summing any i rows of $\mathbf{H}_l$ where $i \in \{1, ..., l\}$ will contain all the binary l-tuples across the columns. Therefore, $\mathbf{A}_l$ can be constructed by summing all possible combinations of i rows of $\mathbf{H}$ where $i \le l$ because doing so will generate all the binary l-tuples for any l columns, and we have

$$\mathbf{A}_l = \{\{\mathbf{h}_{i_0}\}, \{\mathbf{h}_{i_0} + \mathbf{h}_{i_1}\}, ..., \{\mathbf{h}_{i_0} + ... + \mathbf{h}_{i_{l-1}}\}\} \tag{8}$$

for all $i_j \in \{0, 1, ..., n-k-1\}$, where $i_j < i_{j+1}$ and $\mathbf{h}_{i_j}$ is the i_j^{th} row vector of $\mathbf{H}$. Notice that the construction of $\mathbf{A}_l$ in Equation (8) leads to redundant rows, since all possible combinations of rows of $\mathbf{H}_l$ are not necessarily distinct. The sums of r rows of $\mathbf{H}_l$ are a subset of the sums of $r + 2$ rows of $\mathbf{H}_l$. Moreover, sums of up to $l - 1$ rows of $\mathbf{R}_l$ provide $2^l - 1$ l-tuples, and the remaining l-tuple can be obtained from one of their complements. Therefore, a more compact $\mathbf{A}_l$ can be constructed as

$$\begin{aligned} \mathbf{A}_l = \{&\{\mathbf{h}_{i_0} + ... + \mathbf{h}_{i_{l-3}}\}, \{\mathbf{h}_{i_0} + ... + \mathbf{h}_{i_{l-2}}\}, \\ &\{\overline{\mathbf{h}_{i_0} + ... + \mathbf{h}_{i_{l-3}}}\}, \{\overline{\mathbf{h}_{i_0} + ... + \mathbf{h}_{i_{l-2}}}\}\} \end{aligned} \tag{9}$$

where $\overline{\mathbf{h}}$ denotes the complement of $\mathbf{h}$. Then, the size of $\mathbf{A}_l$ is

$$|A_l| \le 2\left(\binom{n-k}{l-2} + \binom{n-k}{l-1}\right) \tag{10}$$

and the inequality is due to the possibility of repeating vectors in Equation (9). If the LBC used to generate $\mathbf{A}_l$ has a minimum distance that achieves the Singleton bound, $d = n - k + 1$, the LBC is a maximum distance separable (MDS) code and the derived set $\mathbf{A}$ is compact. The encoding failure rate of XOR coding with $\mathbf{A}_l$ given l_s stuck-at errors requires combinatorial computations except when $l = 1$ and can be obtained by numerical simulation. In the case of $l = 1$, we have $P_{e,X,l=1} = 1 - 1/2^{l_s - 1}$. We can generate an example $\mathbf{A}_l$ based on the parity check matrix of an extended Hamming code $(n = 1024, k = 1013, d = 4)$. The XOR candidates can be obtained with $|A_1| = 2$, $|A_2| = 22$, and $|A_3| = 110$, and would require 1, 5, and 7 bits, respectively, to represent the reprocessing information index j. For comparison, the LFSR coding scheme with similar parameters can be realized with the feedback polynomial $X^{1024} + X^{1015} + X^{1002} + X^{1001} + 1$ and $v_{\max,l} = |\mathbf{A}_l|$.

C. Concatenation with BCH to Correct Soft Errors

In Sections III.A and III.B, we simplified the analysis of CSIE without considering the possibility of soft errors. In this section, we combine CSIE with an outer ECC to account for both permanent and temporary errors. We use binary BCH codes as outer codes [19] because BCH code designs can easily be adapted to different code rates. A t-error-correcting BCH code can be shortened by b bits in order to record in those b bits either the number of shifts used in the LFSR scheme or the index of the XORed vector from the XOR scheme. For example, a (1023,983) BCH code with $t = 4$ can be shortened to (1023,982), (1023,978), or (1023,976) and be combined with CSIE to handle 1, 2, or 3 stuck-at errors, respectively. In order to evaluate the concatenated performance, the probability of the number l_e of residual errors after CSIE, denoted by $\Pr(l_e \mid l_s)$, needed to be simulated for all l_s stuck-at errors. Then, the block error rate for the concatenated code could be computed as

$$P_{\text{blker}\mid \text{l}_\text{s}} = \sum_{l_r} \sum_{l_e > t - l_r} \Pr(l_r \mid l_s) \Pr(l_e \mid l_s) \tag{11}$$

where l_r is the number of soft errors in a codeword, and soft errors in reprocessing the information were ignored.

IV. Numerical Simulations

We evaluated the performance of CSIE schemes on a channel model that included both temporary and permanent errors [4]. Although the difference between $C_{\max}$ and $C_{\min}$ shown in Figure 5 was less than 0.01 information bits per cell at $q_T = 10^{-3}$, the actual performance gap for finite-length codewords may be larger. To illustrate this, we ran simulations using the parameters $n = 1024$, $q_T = 10^{-3}$, and $p_T = 0$. Figure 6 shows the block error rates of BCH codes with and without using side information at the encoder. CSIE, together with random binning, either led to a $\times 10^{-8}$ lower block error rate at a coding overhead of $\eta_l \sim 0.01$ or was able to store 0.07 bits more information per cell at a block error rate of 10^{-9} compared to when no side information was used. Shannon limits of capacity when no side information is used ($1 - C_{\min}$) and for CSIE ($1 - C_{enc} = 1 - C_{\max}$) are also shown as vertical dash-dot and solid lines, respectively. Since it is impractical to obtain all of the coding gains offered by random binning, we evaluated the performance of both LFSR- and XOR-based CSIE schemes in Section IV.A when complete side information was available at the encoder, and in Section IV.B when only partial side information was produced by an ECC decoder during scrubbing.

A. Coding with *Complete* Side Information at the Encoder

We plotted in Figure 7 the performance of both the XOR scheme based on an extended $(n = 1024, k = 1013, d = 4)$ Hamming code and the LFSR scheme when complete channel side information was available at the encoder in the presence of only hard errors ($q_T = 10^{-3}$ and $p_T = 0$). The y-axis indicates the block error rate and the x-axis represents the coding overhead $\eta_l = \lceil \log_2 \lfloor A_l \rfloor \rceil / n$. In our simulations, we included the possibility of stuck-at errors occurring at the indexing bits. By using CSIE and a small overhead $\sim 7 \times 10^{-3}$, the block error rate was reduced to $\sim 10^{-3}$ compared to the rate when no side information was used.

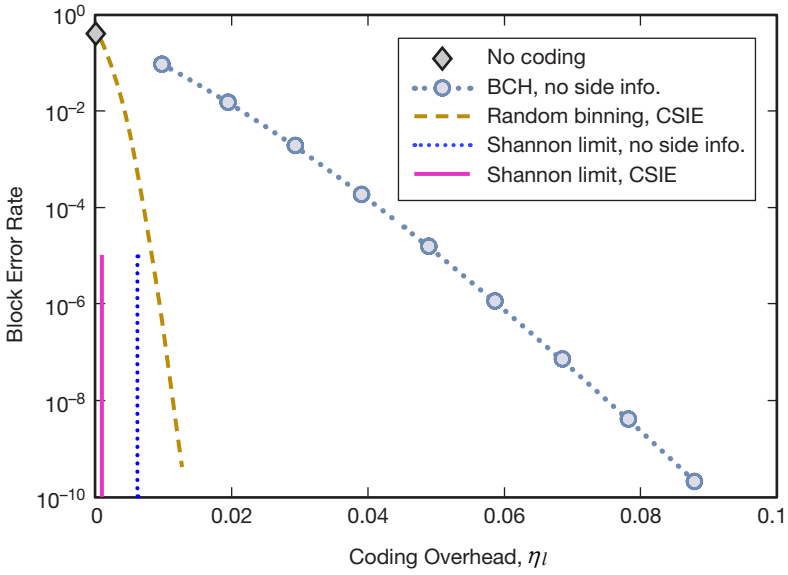

Figure 6. Block error rates of coding with and without side information using finite-length codeword, $n = 1024$. Hard and soft error probabilities are fixed at $q_T = 10^{-3}$ and $p_T = 0$, respectively.

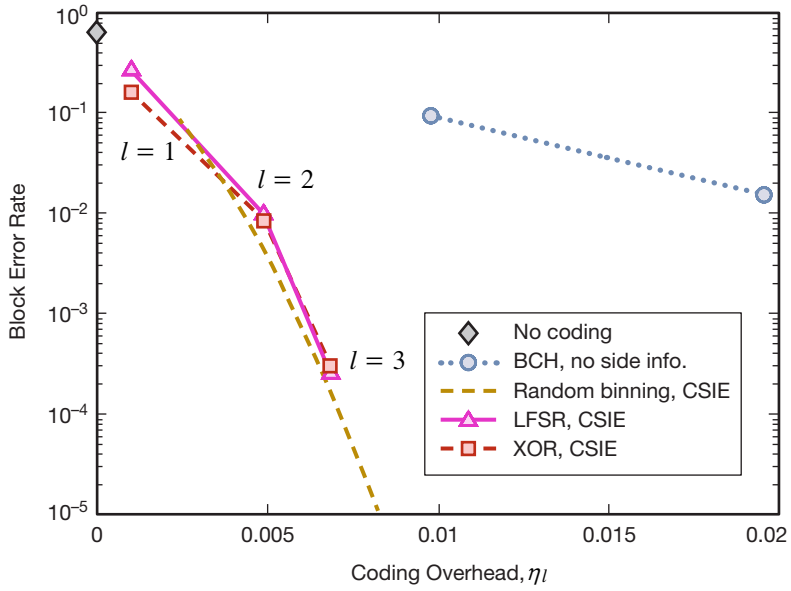

Figure 7. Simulation of CSIE in a channel with only hard errors, $q_T = 10^{-3}$ and $p_T = 0$. Block error rates are evaluated as a function of coding overhead with the length of codeword selected to be $n = 1024$.

We proceeded to evaluate the concatenation of CSIE and an outer $t = 4$ error-correcting (1023,983) BCH code in the presence of both hard and soft errors ($q_T = 10^{-3}$). To simplify analysis, we ignored the possibility of soft errors occurring in the reprocessing index and plotted the error rate performance in Figure 8. For comparison, the error rates obtained by using the (1023,983) BCH code and the $t = 5$ error-correcting (1023,973) BCH code, assuming that the stuck-at bits were treated as erasures, are also plotted. CSIE outperforms even a lower-rate BCH code with erasure decoding (solid ×) that required either additional memory space for tracking erasures or latency-intensive memory cell tests during decoding.

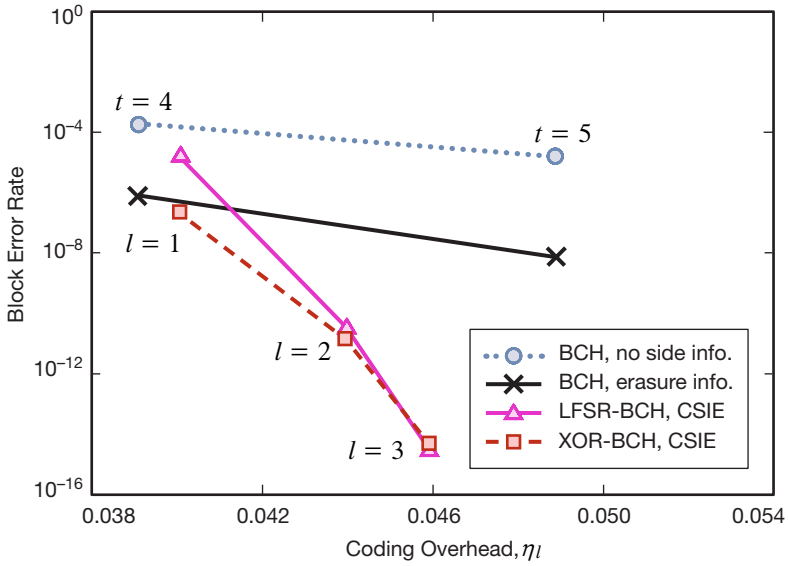

Figure 8. Simulation of ECC concatenated with CSIE in a channel with hard (q_T = 10^{-3}) and soft (p_T = 10^{-6}) errors. Performances of a t = 4 error-correcting (1023,983) BCH code and a t = 5 error-correcting (1023,973) BCH code without CSIE but with erasure of the hard errors are provided for comparisons.

C. Coding with *Partial* Side Information at the Encoder in Memory-Scrubbing Channels

Comparing the input and the output of an ECC decoder after a memory scrub produces *partial* side information that still could be beneficial in CSIE. The information generated by the decoder is not complete because not every corrected symbol is a stuck-at error, and not every stuck-at error is found. However, the impact of unidentified hard errors is dominant and coding with *partial* side information correcting more than one stuck-at error becomes inefficient. We therefore only analyzed the performance of the XOR CSIE scheme with *partial* side information for the case of one stuck-at bit. In this case, the block error rate is computed as

$$P_{\text{blker}|1_s} = \sum_{l_r}\sum_{l_{si}} \Pr(l_r \mid l_s)\Pr(l_{si} \mid l_s)\left(\sum_{l_{ei}}\sum_{l_{eu} > t - l_{ei} - l_r} \Pr(l_{ei} \mid l_{si})\Pr(l_{eu} \mid l_s - l_{si})\right) \tag{12}$$

where l_{si} is the number of stuck-at errors identified by the ECC decoder and l_{ei} and l_{eu} are the numbers of residual errors from identified l_{si} and unidentified $l_s - l_{si}$ stuck-at errors, respectively. For each scrubbing interval with duration T_s, the ECC decoder provides *partial*

side information back to the encoder if $l_{ei} + l_{eu} + l_r$ at the previous interval is within the correction bound of the ECC. The error probabilities of the m^{th} scrubbing interval can then be derived from Equations (1) and (2) as $q_m = 1 - e^{-\lambda_h m T_s}$ and $p_m = 1 - (e^{-\lambda_h T_s} + e^{-2\lambda_s T_s})/2$, where hard errors that occurred in the m^{th} scrubbing interval are treated as soft errors with probability 1/2. The block error rate of the concatenated coding at m^{th} scrubbing can be represented as

$$P_{\text{blker}}^{(m)} = P_{\text{blker}}^{(m-1)} + \left(1 - P_{\text{blker}}^{(m-1)}\right) P_{\text{blker}\,|\,l_s^{(m)}, l_r^{(m)}} \tag{13}$$

where $l_s^{(m)}$ and $l_r^{(m)}$ are the numbers of stuck-at and soft errors given by error probabilities q_m and p_m, respectively, and $P_{\text{blker}}^{(0)} = 0$. We plotted the results of memory-scrubbing simulations using BCH codes and coding with *partial* side information in Figure 9. Each XOR and LFSR CSIE scheme was combined with a $t = 4$ error-correcting (1024,983) BCH code and evaluated with the channel parameters set as $\lambda_h = \lambda_s = 10^{-6}$ errors/bit/day and a scrubbing interval of $T_s = 2$ hr. We allowed no more than one stuck-at error. We also included in the figure for comparison the BCH-only performance on the channel model developed in [5] with and without erasing all of the stuck-at errors. CSIE schemes with *partial* side information provide improved block error rate performance over the BCH code without side information and show favorable performance trade-offs compared to the BCH code with erasure decoding that requires a priori knowledge of the stuck-at error locations.

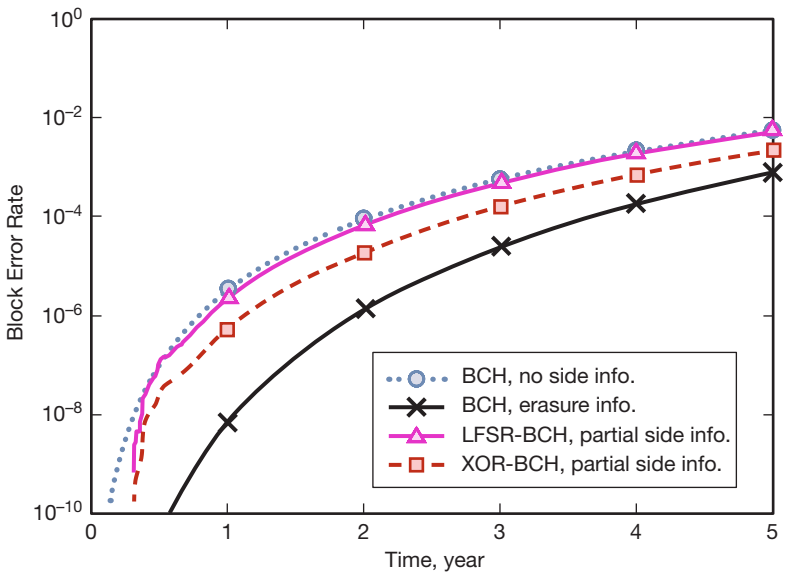

Figure 9. Simulation of a (1024,983) BCH code concatenated with coding with *partial* side information. The *partial* side information is derived from the BCH decoder. The hard and soft error rates are $\lambda_h = \lambda_S = 10^{-6}$ and the scrubbing interval is 2 hr.

V. Summary

For memory devices targeted toward space applications, we refreshed the approach to handling both temporary errors induced by single-event upsets and stuck-at errors caused by device latch-ups. We were better able to capture the radiation-error behavior using a state-dependent channel model, and through capacity analysis showed the benefits of coding with CSIE on this channel. However, existing CSIE schemes were previously developed as information-theoretic concepts and are impractical. We proceeded to devise two CSIE schemes that can be readily implemented on a memory controller. We validated our approach through simulations and demonstrated that CSIE can provide improved reliability or higher storage capacity on memory devices in an extreme radiation environment. Since obtaining complete side information requires additional overhead in storage or computation, we also proposed a low-overhead alternative that uses *partial* information extracted from the decoder during memory scrubs and showed that using CSIE with *partial* side information can be beneficial as well.

References

[1] C. Poivey, J. Barth, G. Gee, H. Safren, and K. LeBel, "Lessons Learned from Radiation-Induced Effects on Solid-State Recorders (SSR) and Memories," NASA/GSFC Radiation Effects Analysis, 2002.
http://radhome.gsfc.nasa.gov/radhome/papers/SPWG02_Poivey.pdf

[2] A. M. Saleh, J. J. Serrano, and J. H. Patel, "Reliability of Scrubbing Recovery-Techniques for Memory Systems," *IEEE Transactions on Reliability*, vol. 39, no. pp. 114–122, April 1990.

[3] G. C. Yang, "Reliability of Semiconductor RAMs with Soft Error Scrubbing Techniques," *IEE Proceedings Computers and Digital Techniques*, vol. 142, issue 5, pp. 337–344, September 1995.

[4] S. Jeon, B. V. K. Vijaya Kumar, E. Hwang, and M. K. Cheng, "Evaluation of Error-Correcting Codes for Radiation-Tolerant Memory," *The Interplanetary Network Progress Report*, vol. 42-181, Jet Propulsion Laboratory, Pasadena, California, pp. 1–18, May 15, 2010.
http://ipnpr.jpl.nasa.gov/progress_report/42-181/181A.pdf

[5] S. Jeon, E. Hwang, B. V. K. Vijaya Kumar, and M. K. Cheng, "LDPC Codes for Memory Systems with Scrubbing," in *Proceedings of IEEE Global Telecommunications Conference 2010 (IEEE GLOBECOM 2010)*, Miami, Florida, December 2010.

[6] S. Jeon, E. Hwang, B. V. K. Vijaya Kumar, and M. K. Cheng, "Investigation of Memory Protection Using Low-Density Parity-Check (LDPC) Codes," *10th Annual Non-Volatile Memory Technology Symposium (NVMTS'09)*, Portland, Oregon, October 2009.

[7] C. E. Shannon, "Channels with Side Information at the Transmitter," *IBM Journal of Research and Development*, vol. 2, no. 4, pp. 289–293, 1958.

[8] M. Costa, "Writing on Dirty Paper," *IEEE Transactions on Information Theory*, vol. 29, no. 8, pp. 439–441, May 1983.

[9] G. Keshet, Y. Steinberg, and N. Merhav, "Channel Coding in the Presence of Side Information," *Foundations and Trends in Communications Information Theory*, vol. 4, no. 6, pp. 445–586, 2007.

[10] S. I. Gelfand and M. S. Pinsker, "Coding for Channel with Random Parameters," *Problems of Control Theory*, vol. 9, no. 1, pp. 19–31, 1980.

[11] C. Heegard and A. E. Gamal, "On the Capacity of Computer Memory with Defects," *IEEE Transactions on Information Theory*, vol. IT-29, no. 5, pp. 731–739, September 1983.

[12] A. V. Kuznetsov, T. Kasami, and S. Yamamura, "An Error-Correcting Scheme for Defective Memory," *IEEE Transactions on Information Theory*, vol. IT-24, no. 6, pp. 712–718, November 1978.

[13] C. Heegard, "Partitioned Linear Block Codes for Computer Memory with 'Stuck-at' Defects," *IEEE Transactions on Information Theory*, vol. 29, no. 6, pp. 831–842, November 1983.

[14] E. Hwang, S. Jeon, B. V. K. Vijaya Kumar, and M. K. Cheng, "Scrubbing with Partial Side Information for Radiation-Tolerant Memory," *Proceedings of IEEE Global Communications Conference 2010 (GLOBECOM'10) Workshop*, Miami, Florida, December 2010.

[15] R. Tandon and S. Ulukus, "On the Rate-Limited Gelfand-Pinsker Problem," *Proceedings of IEEE International Symposium on Information Theory, 2009 (IEEE ISIT'09)*, pp. 1963–1967, Seoul, Korea, June 2009.

[16] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, New York: Wiley, 1991.

[17] R. Zamir, S. Shamai, and U. Erez, "Nested Linear/Lattice Codes for Structured Multiterminal Binning," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1250–1276, June 2002.

[18] K. Chakraborty and P. Mazumder, *Fault-Tolerance and Reliability Techniques for High-Density Random-Access Memories*: Pearson Prentice Hall, 2002.

[19] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, Englewood Cliffs, New Jersey: Prentice Hall, 1983.