This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

LDPC Codes for Memory Systems with Scrubbing

Seungjune Jeon, Euiseok Hwang, B. V. K. Vijaya Kumar
Data Storage Systems Center
Carnegie Mellon University
Pittsburgh, PA 15213
Email: {sjeon, euiseokh}@cmu.edu, kumar@ece.cmu.edu

Michael K. Cheng
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
Email: mkcheng@jpl.nasa.gov

***Abstract*— In space, radiation particles can introduce temporary or permanent errors in memory systems. To protect against potential memory faults, either thick shielding or error correcting codes (ECC) are used. Thick shielding translates into increased mass and conventional ECCs designed for memories are typically capable of only correcting a single error and detecting a double error. Decoding is usually performed through hard-decisions where bits are treated as either correct or flipped in polarity. In this work, we demonstrate that low-density parity-check (LDPC) codes that are already prevalent in many communication applications can also be used to protect memories in space. We develop a channel that models memory error events in a space radiation environment. We describe how to compute soft symbol reliabilities on our channel and compare the performance of soft-decision decoding LDPC codes against conventional hard-decision decoding of Reed-Solomon (RS) codes and Bose-Chaudhuri-Hoquenghem (BCH) codes for a specific memory structure.**

I. Introduction

There are two types of errors in memory systems: soft (i.e., transient) errors and hard (i.e., permanent) errors. Soft errors may be caused by energetic particles, coupling from power supply noise, or variability in device behavior. Typically, soft errors would produce only a single cell malfunction. As memory building blocks shrink into the nanometer regime and when memories are used in space applications, the frequency of multiple cell malfunction increases and these error events can range from a few bit flips to hundreds of errors. In addition to soft errors, hard errors can occur both in manufacturing due to defects or lithography contaminants and in use due to device wear-out or cosmic radiation.

For space applications, memories are either made radiation-hard by strong material shielding or made radiation-tolerant through protection of error correction codes (ECC). However, current radiation-tolerant approaches may not handle multiple bit errors elegantly or at all. For example, the Mars Exploration Rovers (MER) adopted a single error correction and double error detection code to protect memory accesses. This simple error detection and correction (EDAC) code cannot sustain an event with three or more bit errors. Moreover, the access period to selected memory modules on the MERs was about 90 nanoseconds or on the order of 11 MHz because the EDAC circuit was implemented on a radiation-hard chip and clock rates on these elements are limited. In this work, we consider modern low-density parity-check (LDPC) codes for memory systems. LDPC codes are capacity approaching codes that are increasingly being adopted in communications systems ranging from wireless routers to space communications. However, LDPC codes have yet to be considered for protecting memories because well-performing short (less than 1K bits information length) LDPC codes are hard to design and the complexity of iteratively decoding long LDPC codes is higher than the decoding of conventional codes. Decoding LDPC codes also uses soft symbol information provided by the channel but memory outputs are generally treated as hard bits and not soft information. In order to obtain the full performance of soft-decision decoding LDPC codes, we introduce a technique to generate soft symbol information using channel parameters. We show that LDPC codes can improve, when compared to conventional Reed-Solomon (RS) codes and Bose-Chaudhuri-Hoquenghem (BCH) codes, the reliability of commercial off-the-shelf (COTS) memory systems targeted for space use.

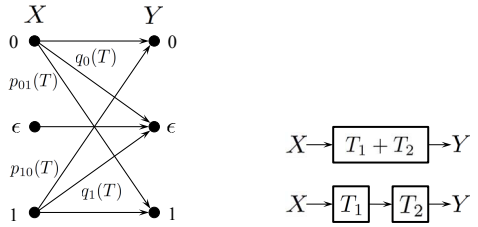
ECC for memories in space have been studied in literature. Saleh et. al [1] investigated single-error-correction and double-error detection (SECDED) codes. Shirvani et. al [2] proposed using Reed-Solomon (RS) codes to protect against data corruption in software. Cardarilli et. al [3] proposed an analytical method to calculate block error rates of RS codes for scrubbing memory systems. Kaneko et. al [4] made measurements to characterize memory failure events in a radiation environment. Recently, MacLeod et. al [5] discussed a plan for satellite test of a nonvolatile memory device. Nguyen and Irom [6] tested radiation effects on recently developed COTS NAND flash memory devices for both single level cell and multi level cell devices. The authors [7] demonstrated that 1kbit LDPC codes can outperform RS codes in the memory systems in radiation environment. More details of this work will be available in [8].

In this paper, we extend the above work. In Section II, we define a new channel to model both soft and hard errors in space. In Section III, we apply RS, BCH, and LDPC codes to our memory model. For decoding of LDPC codes, we derive the needed soft information from our channel model. In Section IV, we compare RS, BCH and LDPC code performance in the targeted environment. In Section V, we conclude our work.

II. Model of Memory Systems

There are two types of errors in memory systems under the effects of radiation. A *soft error* is temporary. The memory

978-1-4244-5637-6/10/$26.00 ©2010 IEEE



(a) Channel model (b) Two equivalent channel models

Fig. 1. Channel model of a one-bit memory without scrubbing. Each transition probability is function of time. Hard error is denoted as ϵ. If $X = \epsilon$, the memory bit is not usable due to the hard error.

content at the soft error location can be corrected. For example, ECCs can correct soft errors by updating the erroneous location with the correct value. Therefore, the number of soft errors can be decreased after memory contents are overwritten by the ECC decoding output.

In contrast, *hard error* is fixed such that the memory content at the hard error location cannot be changed. No new information can be written. The locations of hard errors can be detected and made known to the ECC decoder. Therefore, hard errors can be treated as erasures. The number of hard errors cannot be decreased even after memory contents are overwritten by the ECC decoding output and the number will only increase with time.

Scrubbing is an operation to rewrite the ECC decoder outputs to correct errors. The *scrubbing interval* is the time between each scrubbing. The number of soft errors can be reduced after scrubbing, whereas the number of hard errors cannot be decreased by scrubbing. The number of hard errors only increases. Frequent scrubbing or short scrubbing interval may keep error rates low over longer periods. However, frequent scrubbing will consume more power, which is unattractive in space applications.

It might be possible to avoid writing to the hard error locations and writing the corrected bits somewhere else after scrubbing. However, such a scheme requires additional memories to store the corrected bits and the addresses of those bit locations. Moreover, the new bit locations and the data of addresses can also suffer from radiation effects. Since this scheme uses more redundant bits, the effectiveness of this scheme should be evaluated by comparing ECCs of lower code rates. Analysis and evaluation of this scheme is out of the scope of this paper. We will focus on nominal scrubbing without relocating hard error bits.

In the following Subsections, we will define a channel model without scrubbing and a channel model with scrubbing.

A. Channel Model without Scrubbing

Since the number of errors caused by radiation particles depend on the time interval T of the radiation exposure, our channel model depends on T, as seen in Fig. 1(a). Often COTS memories are used in spacecrafts but COTS memories do not provide soft information per bit location during read-back. The available information to the ECC decoder is the binary information and the location of hard errors. We mark erasures as ϵ and define the transition probabilities as follows.

$$\Pr(Y = 1 \mid X = 0) = p_{01}(T) \tag{1}$$
$$\Pr(Y = 0 \mid X = 1) = p_{10}(T) \tag{2}$$
$$\Pr(Y = \epsilon \mid X = 0) = q_0(T) \tag{3}$$
$$\Pr(Y = \epsilon \mid X = 1) = q_1(T) \tag{4}$$
$$\Pr(Y = \epsilon \mid X = \epsilon) = 1 \tag{5}$$

Eq. (5) implies that a hard error always remains unchanged.

We label channels with $p_{01}(T) = p_{10}(T)$ and $q_0(T) = q_1(T)$ as symmetric channels and define the probability of soft error during time T as

$$p(T) = p_{01}(T) = p_{10}(T), \tag{6}$$

the probability of hard error during time T as

$$q(T) = q_0(T) = q_1(T), \tag{7}$$

and the probability of no error during time T as $r(T)$. Thus,

$$p(T) + q(T) + r(T) = 1. \tag{8}$$

Now we will derive these probabilities by the following reasoning. If there is no scrubbing, a cascade of two channel models over successive time intervals T_1 and T_2 must be equivalent to one channel model over $T_1 + T_2$. Since we have a soft error if and only if there is only one soft error in T_1 or T_2 but no hard error, the soft error probability over the interval $T_1 + T_2$ must satisfy

$$p(T_1 + T_2) = p(T_1)r(T_2) + r(T_1)p(T_2). \tag{9}$$

Since we have a hard error if and only if there is a hard error in the first interval T_1; or no hard error in the first interval and a hard error in the second interval T_2, the hard error probability over the interval $T_1 + T_2$ must satisfy

$$q(T_1 + T_2) = q(T_1) + (1 - q(T_1))q(T_2). \tag{10}$$

We can obtain the following boundary conditions

$$\begin{aligned} p(0) &= 0, & \lim_{T\to\infty} p(T) &= 0, \\ q(0) &= 0, & \lim_{T\to\infty} q(T) &= 1, \\ r(0) &= 1, & \lim_{T\to\infty} r(T) &= 0, \end{aligned} \tag{11}$$

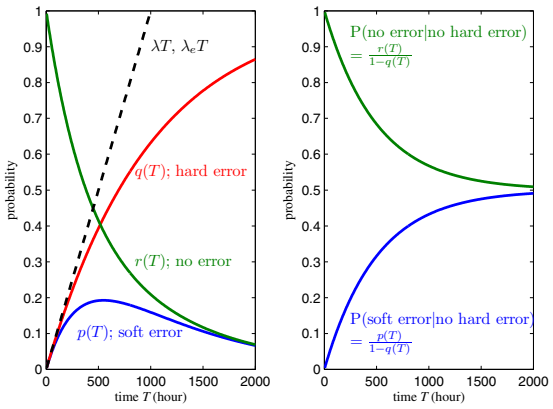
since there is no error at the beginning and all the bits will eventually become hard errors.

Now we are ready to obtain $p(T)$, $q(T)$, and $r(T)$ by solving Eqs. (8), (9), and (10). One way is to solve differential equations by differentiating the equations with respect to the time variables and using the boundary conditions in Eq. (11).

We obtained the following solutions for some nonnegative constants λ and λ_e.

$$p(T) = \frac{e^{-\lambda_e T} - e^{-(2\lambda+\lambda_e)T}}{2} \tag{12}$$
$$q(T) = 1 - e^{-\lambda_e T} \tag{13}$$
$$r(T) = \frac{e^{-\lambda_e T} + e^{-(2\lambda+\lambda_e)T}}{2} \tag{14}$$

978-1-4244-5637-6/10/$26.00 ©2010 IEEE



(a) Probabilities (b) Conditional probabilities

Fig. 2. Probabilities of soft error, hard error, and no error without scrubbing as functions of time when $\lambda = \lambda_e = 10^{-3}$ errors/bit/day

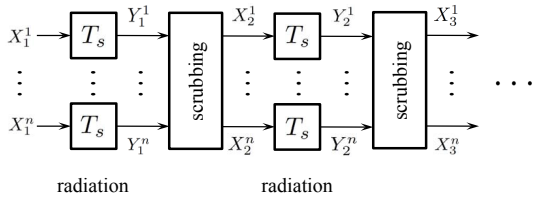


Fig. 3. Channel model with scrubbing.

For a small time interval $T \ll \frac{1}{2\lambda+\lambda_e}$, we can approximate the soft error probability to $p(T) \approx \lambda T$ and the hard error probability $q(T) \approx \lambda_e T$. We refer to λ as the *soft error rate* (the number of soft errors per bit per unit time), and λ_e as the *hard error rate* (the number of hard errors per bit per unit time). Strictly speaking, the unit for λ and λ_e is simply inverse of time. However, error/bit/day is usually used for convenience. The soft error rates and the hard error rates are a measure of the strength of the radiation effect on the memory systems.

In Fig. 2(a), we see that the probability of hard error $q(T)$ increases monotonically, the probability of no error $r(T)$ decreases monotonically. It might not be immediately clear why the probability of soft error $p(T)$ increases to the maximum and then decreases to zero. Fig. 2(b) is used to understand this behavior. The non-stuck bits (or bits yet to become hard errors) will experience soft errors with increasing probability (seen in Fig. 2(b)) over time and eventually deteriorate to become stuck bits or hard errors. Therefore, the probability of hard error increases monotonically and the probability of soft errors increases initially until reaching a peak then decreases to zero as seen in Fig. 2(a). Note that even if there are no hard errors at all, memory will eventually not be able to store any information at infinite time because $p(T)$ approaches 1/2 over time. This behavior exhibits the same property as an infinite cascade of binary symmetric channels.

B. Channel Model with Scrubbing

We now consider a channel model that includes scrubbing and denote the scrubbing interval as T_s. We illustrate the channel model in in Fig. 3. We assume that soft errors and hard errors in different bit locations are independent, and an error correcting code of an n-bit codeword is used for scrubbing. We also assume that the scrubbing operation is not affected by radiation. That is, the decoder itself is free from any errors caused by the radiation effect. The processing time for each scrubbing is assumed to be negligible so that the overall elapsed time is simply the scrubbing interval times the number of scrubbing intervals.

III. Coding

A. Review of BCH Codes and RS Codes

Suppose that we have an (n, k) BCH code that can correct up to t bit errors per codeword if there are no bit erasures. The minimum distance of the BCH is $d_{\min} = 2t+1$. As long as the number of bit errors (soft errors) e and the number of bit erasures (hard errors) f in a received word are bounded by

$$2e + f \leq 2t, \tag{15}$$

the correct BCH codeword can be found by the decoder.

For pseudo-decoding of BCH codes, if Eq. (15) is satisfied for a received word, we assume that all the errors and erasures are corrected and the corrected codeword is rewritten into memory during scrubbing. Otherwise, we declare a decoding failure and no scrubbing is performed. Note that once the number of accumulated hard errors (bit erasures) reach $d_{\min} - 1$ in a codeword, no further errors (hard or soft) can be corrected.

Eq. (15) also can be used for the pseudo-decoding of an (n, k) RS code over a Galois Field of size 2^m where we map k information symbols into n codeword symbols and each field element is represented by m bits. This RS code can correct up to t symbol errors if there is no symbol erasures and the minimum distance of the RS code is $d_{\min} = n-k+1 = 2t+1$.

An RS symbol is considered erased if one or more bits that comprise the symbol is erased. An RS symbol is considered erroneous when one or more bits that comprise the symbol is in error and no erasures occur in the symbol. Therefore we can obtain the probability of hard symbol error (symbol erasure) during a scrubbing interval as

$$q_s(T_s) = 1 - (1 - q(T_s))^m \tag{16}$$

and the probability of soft symbol error (erroneous symbol) as

$$p_s(T_s) = (1 - q(T_s))^m - (1 - p(T_s) - q(T_s))^m. \tag{17}$$

By using the method in [3], The block error rates of RS codes can be calculated analytically if the probabilities of soft symbol error and hard symbol error are given. We also modified the method in order to calculate the block error rates of BCH codes analytically.

978-1-4244-5637-6/10/$26.00 ©2010 IEEE

B. LDPC Codes

Soft information can be generated using the radiation parameters and the scrubbing interval. The log-likelihood ratio (LLR) $L_{\text{ch}}(y)$ for a memory output y can be expressed in terms of the parameters in the channel model as follows.

$$L_{\text{ch}}(y) \triangleq \log \frac{\Pr(Y=y|X=0)}{\Pr(Y=y|X=1)} \tag{18}$$

$$= \begin{cases} \log \dfrac{p_{01}(T_s)}{1-p_{10}(T_s)-q_1(T_s)}, & \text{if } y=1, \\ \log \dfrac{1-p_{01}(T_s)-q_0(T_s)}{p_{10}(T_s)}, & \text{if } y=0, \\ \log \dfrac{q_0(T_s)}{q_1(T_s)}, & \text{if } y=\epsilon \end{cases} \tag{19}$$

For symmetric channels, the LLR from the memory output can be simplified as

$$L_{\text{ch}}(y) = \begin{cases} \pm \log \dfrac{p(T_s)}{r(T_s)}, & \text{if } y=1(+),\ y=0(-), \\ 0, & \text{if } y=\epsilon. \end{cases} \tag{20}$$

The derived LLRs are applicable not just to LDPC decoding but to all soft decision decoding algorithms. Bit decisions are made based on the LLRs and the corrected bits can be used to update the soft errors but not the hard errors.

If we substitute $p(T_s)$ and $r(T_s)$ from Eq. (12) for those in Eq. (20), we obtain

$$L_{\text{ch}}(y) = \begin{cases} \pm \log \tanh(\lambda T_s), & \text{if } y=1(+),\ y=0(-), \\ 0, & \text{if } y=\epsilon. \end{cases} \tag{21}$$

It is interesting to note that the hard error rate λ_e is canceled out so that the LLR is independent of λ_e for symmetric channels. In fact, it should not be a surprise since $p(T_s)/r(T_s) = \tanh(\lambda T_s)$ is the ratio between the points on the two curves in Fig. 2(b) at $T = T_s$.

For LDPC decoding, either sum-product decoding or min-sum decoding can be used. The two methods are summarized in [9]. The min-sum decoding is a low-complexity approximation of sum-product decoding and is less sensitive to the channel parameters in general.

IV. Results

Fig. 4 shows the block error rates of equivalent RS, BCH, and LDPC codes that can contain 4096 bits of message at code rate 8/9. This figure shows that LDPC code with sum-product decoding provides significant advantage over not only conventional RS or BCH codes but also the same LDPC code with min-sum decoding. Before we discuss the results further, we will describe the setup of the simulation.

The horizontal axis represents elapsed time and its unit is the number of scrubbing intervals. For example, 100 scrubbing intervals correspond to 100 hours when each scrubbing interval is one hour long. For simulation, we used the probability of soft error and hard error as 4.167×10^{-5}. These soft and hard

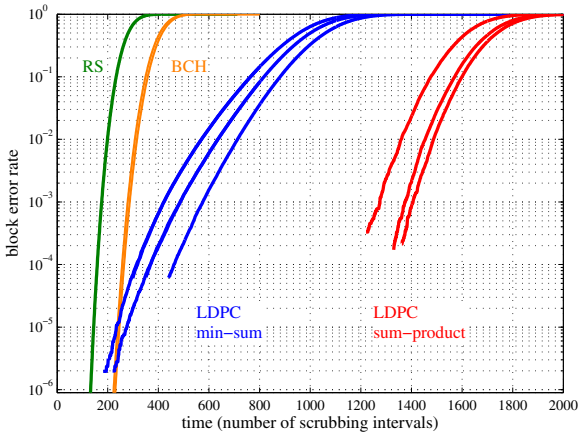


Fig. 4. Performance of a (4608, 4096) LDPC code. $p(T_s) = q(T_s) = 4.17 \times 10^{-5}$. $\lambda = \lambda_e = 10^{-3}$ error/bit/day and $T_s = 1/24$ day. The number of maximum iterations of LDPC decoder per scrubbing were 10, 20, and 40.

error probabilities do not change as long as the products λT_s and $\lambda_e T_s$ remain the same. One combination is $T_s = 1$ hour and $\lambda = \lambda_e = 10^{-3}$ errors/bit/day.

Note that this radiation parameter translates into a harsher condition than that measured in space. Typically λ and λ_e are in the range of 10^{-7} to 10^{-8} in space [4]–[6]. For $\lambda = \lambda_e = 10^{-7}$, the results in Fig. 4 correspond to the case of $T_s = 10^4$ hours.

The vertical axis in Fig. 4 denote block error rates. A block error is claimed when we obtain either a decoding failure or an incorrect codeword estimate at the decoder output. We do not show results corresponding to fewer than 5 block errors.

The RS code is a shortened $(462, 410)$ code over GF(2^{10}) that can correct up to 26 soft symbol errors ($t = 26$ symbols, $d_{\min} = 53$ symbols). Two BCH codes with code rates slightly above 8/9 and below 8/9 are plotted in Fig. 4 since the BCH code with code rate exactly 8/9 and the message length 4096 bits does not exist. The lines for the two codes are too close to each other to be distinguishable. One code is a $(4603, 4096)$ code shortened from the $(8191, 7684)$ BCH code that can correct up to 39 soft errors in a codeword ($t = 39$ bits, $d_{\min} = 79$ bits). Another code is a $(4616, 4096)$ code shortened from $(8191, 7671)$ code that can correct up to 40 soft errors in a codeword ($t = 40$ bits, $d_{\min} = 81$ bits). The block error rates obtained by the analytical method [3] for the RS code and our modification for the BCH codes were identical to our simulation results.

The LDPC code is a $(4608, 4096)$ code whose parity check matrix was generated by the progressive edge growth (PEG) method [10], [11]. The column weight of the parity check matrix is 5 and the girth of the bipartite graph is 6. The minimum distance of the LDPC code is unknown, which is usual for LDPC codes. In general, computing the minimum distance of a binary linear code is an NP-hard problem [12]. RS codes and BCH codes are easy to analyze because their minimum distances are defined by the code design. For each of min-sum decoding and sum-product decoding, the number

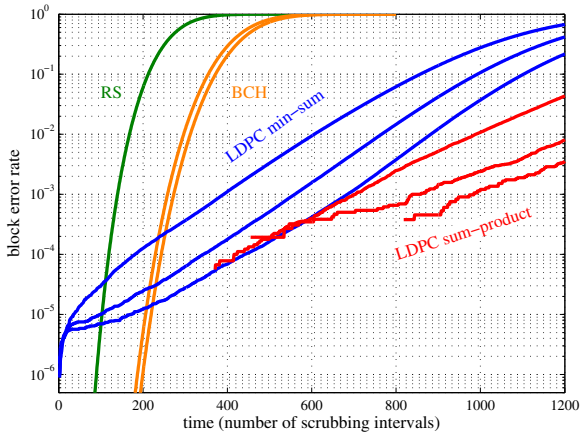


Fig. 5. Performance of a (2304, 2048) LDPC code. The number of maximum iterations of LDPC decoder per scrubbing were 10, 20, and 40.

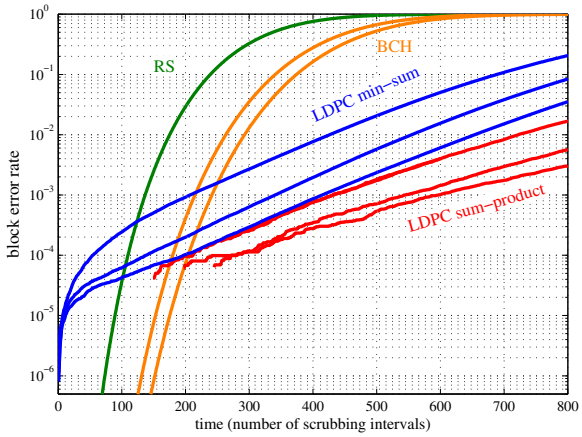


Fig. 6. Performance of a (1152, 1024) LDPC code. The number of maximum iterations of LDPC decoder per scrubbing were 10, 20, and 40.

of maximum iterations for the three lines is 10, 20, and 40 from top to bottom.

We see that LDPC code with sum-product decoding provides significant gains over RS or BCH codes. Meanwhile, the advantage of LDPC code with min-sum decoding over RS and BCH codes decreases as target block error rates decrease and the advantage even disappears when the curves cross each other in low block error rates. The sum-product curves could cross the RS or BCH curves in the low block error rate region. However, if we assume that the slopes of the curves are maintained, the crossing point is expected to be at extremely low block error rates so that the LDPC code with sum-product decoding provides advantage over RS or BCH codes for practical applications.

The performance gain of sum-product decoding over min-sum decoding can be explained as follows. In additive white Gaussian noise (AWGN) channels, sum-product decoding for the LDPC code in Fig. 4 offers a 0.4-dB coding gain over min-sum decoding at block error rate 10^{-6}, which is comparable for other LDPC codes as well [13, p. 233]. For the 0.4-dB difference, at the SNR where the block error rate for sum-product decoding is 10^{-6}, the block error rate for min-sum decoding is about 10^{-2}. The ratio between the two is about 10^{-4}. Assuming no hard errors for simple analysis, the block error rate after α scrubbing intervals is $1-(1-\beta)^{\alpha} \approx \alpha\beta$ for $\beta \ll 1$, where β is the block error rate for one scrubbing interval. That is, the block error rates after the same scrubbing intervals are approximately proportional to β for small β. This behavior is what we observe in Fig. 4. For example, at $T = 1200T_s$, the ratio between the block error rates of min-sum decoding and sum-product decoding is approximately 10^{-4}.

Fig. 5 shows the block error rates of equivalent RS, BCH, and LDPC codes that can contain 2048 bits of message at code rate 8/9. The simulation setup is the same as in Fig. 4 except for the length of the error correcting codes. The RS code is a shortened $(231, 205)$ code over GF(2^{10}) that can correct up to 13 soft symbol errors ($t = 13$ symbols, $d_{\min} = 27$ symbols). One BCH code (left curve) is a $(2300, 2048)$ code shortened from the $(4095, 3843)$ BCH code that can correct up to 21 soft errors in a codeword ($t = 21$ bits, $d_{\min} = 43$ bits). The other BCH code (right curve) is a $(2312, 2048)$ code shortened from the $(4095, 3831)$ BCH code that can correct up to 22 soft errors in a codeword ($t = 22$ bits, $d_{\min} = 45$ bits). The LDPC code is a $(2304, 2048)$ PEG code with the column weight 3 and girth 6. For each of min-sum decoding and sum-product decoding, the number of maximum iterations for the three lines is 10, 20, and 40 from top to bottom.

We observe similar behaviors of RS code, BCH code, and LDPC code with min-sum decoding as in Fig. 4. Interestingly, the advantage of the LDPC with sum-product decoding is diminished significantly so that even the sum-product decoding can be outperformed by BCH codes at block error rates below 10^{-5} if we assume that the slope of the curves are maintained. Therefore, for these 2kbit codes, using LDPC codes can provide gains only if target block error rates are above about 10^{-5}.

Fig. 6 shows the block error rates of equivalent RS, BCH, and LDPC codes that can contain 1024 bits of message at code rate 8/9. The simulation setup is also the same as in Fig. 4 except for the length of the error correcting codes. The RS code is a shortened $(144, 128)$ code over GF(2^8) that can correct up to 8 soft symbol errors ($t = 8$ symbols, $d_{\min} = 17$ symbols). One BCH code (left curve) is a $(1156, 1024)$ code shortened from the $(2047, 1915)$ BCH code that can correct up to 12 soft errors in a codeword ($t = 12$ bits, $d_{\min} = 25$ bits). The other BCH code (right curve) is a $(1145, 1024)$ code shortened from the $(2047, 1926)$ BCH code that can correct up to 11 soft errors in a codeword ($t = 11$ bits, $d_{\min} = 23$ bits). The LDPC code is a $(1152, 1024)$ PEG code with column weight 3 and girth 6. For each of min-sum decoding and sum-product decoding, the number of maximum iterations for the three lines is 10, 20, and 40 from top to bottom.

We can observe similar behaviors of RS code, BCH code, and LDPC code as in Fig. 5. The gain of sum-product decoding over min-sum decoding becomes even smaller than in Fig. 5. Comparing Figs. 4, 5, and 6, the gain of sum-product decoding

978-1-4244-5637-6/10/$26.00 ©2010 IEEE

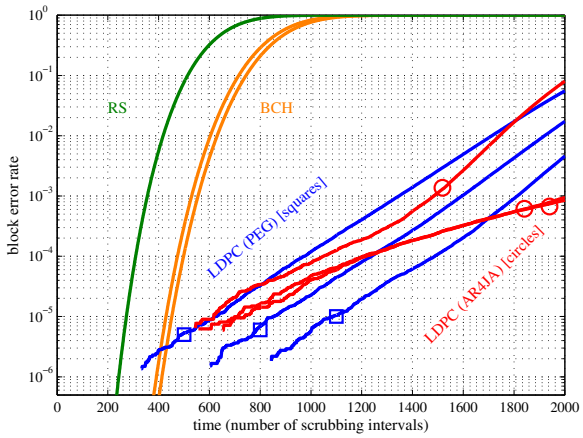


Fig. 7. Performance of two (1280, 1024) LDPC codes (min-sum decoding): an accumulate-repeat-by-4-jagged-accumulate (AR4JA) LDPC code and a PEG LDPC code. All codes are rate-$4/5$. The number of maximum iterations of LDPC decoder per scrubbing were 10, 20, and 40.

over min-sum decoding increases as the code length increases at a fixed code rate. In particular, the gain from the 2kbit code to the 4kbit code is very large whereas the gain from the 1kbit code to the 2kbit code is small.

JPL has designed structured LDPC codes based on protographs and circulants [14], [15]. This construction enables high-speed decoder implementations because the component protographs that are the building blocks to the bigger code graph can be decoded in parallel. The structure of the protograph then determines the threshold and error floor of the overall code. Divsalar et al. [16] recognized that a protograph described by simple accumulate and repeat operators can yield codes with sharp waterfalls and low error floors. We plot the performance of the rate 4/5 information block size 1024-bit accumulate repeat-by-4 jagged accumulate (AR4JA) LDPC code in Fig. 7 and compare the performance to an equivalent rate and length RS and BCH codes: (160, 128) RS code over GF(2^8), $(1277, 1024)$ BCH code shortened from $(2047, 1794)$ BCH code that can correct up to 23 soft errors in a codeword ($t = 23$ bits, $d_{\min} = 47$ bits), and $(1288, 1024)$ BCH code shortened from the $(2047, 1783)$ BCH code that can correct up to 24 soft errors in a codeword ($t = 24$ bits, $d_{\min} = 49$ bits). As with the PEG LDPC code, the AR4JA code outperforms the RS and BCH codes for high block error rates.

V. Conclusions

To protect against radiation induced errors in space, memory systems either store the same information repeatedly in different memory locations or use a conventional ECC scheme such as BCH or RS codes. In this work, we developed a simple channel that models single bit errors due to radiation assuming bit errors occur independently from bit-to-bit. We showed that modern LDPC codes can be used in place of conventional RS codes to improve the radiation tolerance of memory modules. Instead of decoding hard (i.e., 0 or 1) bits, LDPC decoding uses soft information provided by the channel. We showed how to compute soft symbol reliabilities on our channel for input to soft-decision LDPC decoders. We considered two LDPC code constructions: one generated by progressive edge growth and the other is based on protographs and compared their performances to equivalent rate and length BCH and RS codes. The simulation results suggest that LDPC codes can extend the lifetime of memory systems over equivalent BCH or RS codes at relevant target block error rates in radiation environment.

Acknowledgment

This research is supported in part by a California Institute of Technology/Jet Propulsion Laboratory Strategic University Research Partnership grant.

References

[1] A. M. Saleh, J. J. Serrano, and J. H. Patel, "Reliability of scrubbing recovery-techniques for memory systems," *IEEE Trans. Rel.*, vol. 39, no. 1, pp. 114–122, Apr. 1990.

[2] P. P. Shirvani, N. R. Saxena, and E. J. McCluskey, "Software-implemented EDAC protection against SEUs," *IEEE Trans. Rel.*, vol. 49, no. 3, pp. 273–284, Sep. 2000.

[3] G. C. Cardarilli, M. Ottavi, S. Pontarelli, M. Re, and A. Salsano, "Data integrity evaluations of Reed Solomon codes for storage systems," in *Proc. the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT04)*, Cannes, France, Oct. 2004.

[4] H. Kaneko, "Error control coding for semiconductor memory systems in the space radiation environment," in *Proc. the 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05)*, Monterey, CA, Oct. 2005.

[5] T. C. MacLeod, R. Sayyah, W. H. Sims, K. A. Varnavas, and F. D. Ho, "Satellite test of radiation impact on Ramtron 512K FRAM," in *10th Annual Non-Volatile Memory Technology Symposium (NVMTS'09)*, Portland, OR, Oct. 2009.

[6] D. N. Nguyen, "Radiation effects on NAND flash memories," in *10th Annual Non-Volatile Memory Technology Symposium (NVMTS'09)*, Portland, OR, Oct. 2009.

[7] S. Jeon, E. Hwang, B. V. K. Vijaya Kumar, and M. K. Cheng, "Investigation of memory protection using low-density parity-check (LDPC) codes," in *10th Annual Non-Volatile Memory Technology Symposium (NVMTS'09)*, Portland, OR, Oct. 2009.

[8] S. Jeon, B. V. K. Vijaya Kumar, E. Hwang, and M. K. Cheng, "Evaluation of error correcting codes for radiation-tolerant memory," *JPL Interplanetary Network Progress Report*, to appear.

[9] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[10] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Progressive edge-growth tanner graphs," in *Proc. IEEE Global Commun. Conf. 2001 (IEEE GLOBECOM'01)*, vol. 2, San Antonio, TX, Nov. 2001, pp. 995–1001.

[11] Z. Li and B. V. K. Vijaya Kumar, "A class of good quasi-cyclic low-density parity check codes based on progressive edge growth graph," in *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, vol. 2, Pacific Grove, CA, Nov. 2004, pp. 1990–1994.

[12] A. Vardy, "The intractability of computing the minimum distance of a code," *IEEE Trans. Inf. Theory*, vol. 43, no. 6, pp. 1757–1766, Nov. 1997.

[13] W. E. Ryan and S. Lin, *Channel Codes: Classical and Modern*. Cambridge University Press, 2009.

[14] J. Thorpe, "Low-density parity-check codes constructed from protographs," *JPL Interplanetary Network Progress Report*, vol. 42-154, Aug. 2003.

[15] K. S. Andrews, D. Divsalar, S. Dolinar, J. Hamkins, C. R. Jones, and F. Pollara, "The development of turbo and LDPC codes for deep-space applications," *Proc. IEEE*, vol. 95, no. 11, pp. 2142–2156, Nov. 2007, special issue on technical advances in deep space communications and tracking: part 2.

[16] D. Divsalar, S. Dolinar, C. Jones, and J. Thorpe, "Construction of protograph LDPC codes with minimum distance linearly growing with block size," in *Proc. IEEE Global Commun. Conf. 2005 (IEEE GLOBECOM'05)*, St. Louis, MO, Nov. 2005.