

Analysis of One-Step Majority Logic Decoding under Correlated Data-Dependent Gate Failures

Srdan Brkic
School of Electrical Engineering,
University of Belgrade
Email: brka05@gmail.com

Predrag Ivanis
School of Electrical Engineering,
University of Belgrade
Email: predrag.ivanis@etf.rs

Bane Vasić
Dep. of ECE and Dept. of Mathematics,
University of Arizona
Email: vasic@ece.arizona.edu

Abstract—In this paper we present analysis of one-step majority logic decoders made of unreliable components in the presence of data-dependent gate failures. Gate failures are modeled by a Markov chain, and based on the combinatorial representation of the fault configurations, a closed-form expression for the average bit error rate is derived for a regular LDPC code ensemble. Presented analysis framework is then used for obtaining upper bounds on decoding performance under timing errors.

I. INTRODUCTION

Increased integration factor of integrated circuits together with stringent energy-efficiency constraints result in an increased unreliability of today's semiconductor devices, and necessities for a new design paradigm for Very Large Scale Integration (VLSI) technologies in which fully reliable operation of hardware components is not guaranteed [1]. Error control coding as a method for ensuring fault-tolerance of systems build of unreliable hardware was introduced in the late sixties and early seventies by Taylor [2] and Kuznetsov [3] in the context of reliable storage. The equivalence between Gallager B decoder built from unreliable logic gates and Taylor-Kuznetsov fault-tolerant memory architectures was first observed by Vasic *et al.* in [4] and [5], and developed by Vasic and Chilappagari [6] into a theoretical framework for analysis and design of faulty decoders of LDPC codes.

Performance of LDPC codes under faulty iterative decoding in the limit of code length approaching infinity was studied by Varshney in [7], who has shown that under von Neumann failure model density evolution is applicable to faulty decoders which he used to examine performance of faulty Gallager A and belief-propagation algorithms. Density evolution analysis of a noisy Gallager B decoder was presented by Yazdi *et al.* in [8]. Similar analysis of faulty Gallager B algorithm using EXIT charts was done by Leduc-Primeau and Gross in [9]. More general finite-alphabet decoders were investigated by Huang and Dolecek in [10], while noisy min-sum decoder realization was considered by Ngassa *et al.* in [11].

One-step majority logic decoding introduced in the sixties by Rudolf [12] is an important class of algorithms in the context of faulty decoding and has been studied recently in a number of papers. In contrast to iterative decoders, the bit error rate performance of these decoders can be evaluated analytically for finite-length codes as shown by Radhakrishnan *et al.* [13]. The decoding process is terminated after only one

iteration, and the bit estimates are obtained by a majority vote on multiple parity check decisions. The faulty version of one-step majority logic decoders were investigated by Chilappagari *et al.* in [14] where the combinatorial characterization of a decoder correction capability in the presence of von Neumann gate failures was presented. The analysis was limited only to codes constructed based on projective geometries.

In the above references, a special type of so called *transient failures* is assumed. Transient failures manifest themselves at particular time instants but do not necessarily persist for later times. These failures have probabilistic behavior and we assume the knowledge of their statistics. The simplest such statistics is due to von Neumann failure model [15], which assumes that each component of a (clocked) Boolean network fails at every clock cycle with some known probability. Additionally, the failures are not temporally nor spatially correlated. In other words, failures of a given component are independent of those in previous clock cycles and independent of failures of other components.

However, the von Neumann failure model is only a rough approximation of the physical processes leading to logic gate failures. Actual probability of failure of logic gates is highly dependent on a digital circuit manufacturing technology and for high integration factors the failures are data-dependent and/or temporally correlated. For example, timing errors are heavily dependent on data values processed by the gate in previous bit intervals and can not be represented accurately by von Neumann model.

In this paper we propose more general approach to faulty gate modeling. In order to capture the effects of data-dependent and correlated nature of gate failures Markov chains are used. We derive a closed form expression of the bit error rate (BER) for an ensemble of regular LDPC codes free of four-cycles decoded by a faulty one-step majority logic decoding, in the presence of correlated gate failures. This result is a generalization of the results presented in [14] where only von Neumann errors are considered.

The rest of the paper is organized as follows. In Section II the preliminaries about majority logic decoding are discussed. In Section III we give a description of novel approach to error modeling. Section IV is dedicated to theoretical analysis of the decoder, while numerical results are presented in Section V. Finally, some concluding remarks are given in Section VI.

II. PRELIMINARIES

A. Decoding of LDPC Codes

Let C be a (γ, ρ) -regular binary LDPC code of length n , with parity check matrix H . The parity check matrix can be represented with a bipartite graph called the Tanner graph. Each column in the parity check matrix corresponds to a variable node and each row corresponds to a check node in the Tanner graph, and a variable node v and a check node c are adjacent if and only if $H_{c,v} = 1$. A vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is a codeword if and only if $H\mathbf{x}^T = \mathbf{0} \pmod{2}$. A codeword \mathbf{x} is stored in a memory, and when read from the memory each bit x_v is flipped by probability α and observed as r_v . We refer to r_v as value of the variable node v . The number of flipped bits is called the Hamming distance between the stored codeword \mathbf{x} and the read-back word \mathbf{r} , and is denoted as $d_H(\mathbf{x}, \mathbf{r})$.

Let E_x represent a set of edges incident on a node x in a Tanner graph (x can be either variable or check node), then the one-step majority decoding may be summarized as follows.

- 1) Each variable node v sends r_v along the edge in E_v .
- 2) Each check node c sends m_c along each edge in E_c where

$$m_c = \sum_{e' \in E_c \setminus \{e\}} m_{e'} \pmod{2}. \quad (1)$$

- 3) At each variable node v , an estimate of the corresponding bit value of \hat{r}_v is made

$$\hat{r}_v = \begin{cases} 1 & \text{if } |\{e' \in E_v : m_{e'} = 1\}| \geq \lfloor \gamma/2 \rfloor + 1, \\ 0 & \text{if } |\{e' \in E_v : m_{e'} = 0\}| \geq \lfloor \gamma/2 \rfloor + 1, \\ r_v & \text{otherwise.} \end{cases} \quad (2)$$

B. Transient Gate Failures

Let $f : \{0, 1\}^m \rightarrow \{0, 1\}$, $m > 1$, be an m -argument Boolean function. The relation between the input arguments y_1, y_2, \dots, y_m and the output z of a *perfect* gate realizing this function is $z = f(y_1, y_2, \dots, y_m)$. For a *faulty* gate, this input-output relation is $z = f(y_1, y_2, \dots, y_m) \oplus e$, where \oplus is the Boolean XOR and the error $e \in \{0, 1\}$ is a Bernoulli random variable. Denote by $\mathbf{y} = (y_1, y_2, \dots, y_m)$ the gate input vector, i.e., a vector of arguments. Denote by $\{\mathbf{y}^{(k)}\}_{k \geq 0}$ a time-sequence of input vectors, and by $\{e^{(k)}\}_{k \geq 0}$ the corresponding failure sequence. In the manuscript we will interchangeably use the terms “failure” and “error” meaning that the failures are “additive” errors. In a classical transient failure model the error values $\{e^{(k)}\}_{k \geq 0}$ are either independent of input sequence $\{\mathbf{y}^{(k)}\}_{k \geq 0}$ (von Neumann failure model) or dependent only on gate inputs from the current bit interval.

C. Combinatorics

A *composition* of a positive integer i , is an integer vector $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_\gamma)$ such that $\sum_{j=1}^{\gamma} \lambda_j = i$. The integers $\lambda_j \geq 0$ are called *parts* or *summands*, and γ is called the *composition length*. Note that we slightly modified the standard definition of the composition by allowing the parts to be equal to zero. Consequently, all compositions have the

same length, which simplifies representation of error patterns, given in Section IV.

Let $[l]$ denote the set of first l positive integers, i.e., $[l] = \{1, 2, \dots, l\}$. A restricted composition in which no part size exceeds l is an $[l]$ -composition. The number of the compositions of an integer i whose part sizes do not exceed a fixed integer l is denoted by $T_{i,l}$ and can be calculated based on the methods given in [16]. Alternatively, an $[l]$ -composition can be rewritten using its frequency representation $(1^{f_1}, 2^{f_2}, \dots, l^{f_l})$, where f_j denotes the number of parts equal to j , $1 \leq j \leq l$. For example, $(1, 2, 1, 1)$ is an $[2]$ -composition of 5 of length 4, with frequency representation $f_1 = 3, f_2 = 1$.

Let \mathbf{q}_l be a vector corresponding to one lexicographically ordered u -subset of the set $[l]$ and let the vector \mathbf{q}_r contain the remaining elements of $[l]$. We create a vector \mathbf{q} by juxtapositioning \mathbf{q}_l and \mathbf{q}_r . We can arrange all possible vectors \mathbf{q} into rows of an $\binom{l}{u}$ by l array $Q^{u,l}$. For example, if $l = 4$ and $u = 2$, the rows of $Q^{2,4}$ are $(1, 2, 3, 4)$, $(1, 3, 2, 4)$, $(1, 4, 2, 3)$, $(2, 3, 1, 4)$, $(2, 4, 1, 3)$ and $(3, 4, 1, 2)$. The array $Q^{u,l}$ referred to as the *error configuration matrix* will be instrumental in book-keeping of data-dependent patterns in Section IV.

III. THE CORRELATED ERROR MODEL

Note that, similarly as in [14], we only analyze the message passing version of one-step majority logic decoder. This decoder needs $\gamma(\rho - 1)$ -input XOR gates at each check node and a γ -input majority logic gate at each variable node. Each check node makes an estimate of the value of a variable node based on the values of other variable nodes. The value of the variable node itself is not used in its estimation. The final decision is made on the basis of majority of the estimates, resulting in probability of error of an estimated bit greater than or equal to the probability of failure of the majority logic gate. Since the error probability of the majority logic gate lower bounds the bit error rate (BER) performance, majority logic gates must be made highly reliable. Otherwise, the probability of error is determined by this final gate and not the error control scheme. Thus, it is reasonable to make an assumption that majority logic gates are perfect and that only XOR gates are faulty.

We analyze a system in which different codewords are stored in an unreliable memory. While stored in the register, each bit may be flipped, independently of other bits, with probability α . At every cycle a different codeword is read from the memory and decoded by an unreliable one-step majority decoder. Equivalently, we may assume that the sequence of codewords $\{\mathbf{x}^{(k)}\}_{k \geq 0}$ is transmitted through the Binary Symmetric Channel (BSC) with crossover probability α and then successively decoded by a single one-step majority decoder built from perfect majority logic gates and faulty XOR gates.

In contrast to the state-of-the-art modeling of faulty gates that considers only the failure dependence on the current input values, our model captures more accurately the data and time dependence of the failures. Namely, we assume that $e^{(k)}$, the error at time k , is affected by the current and $M - 1$ prior consecutive gate input vectors, i.e., its probability depends on the vector sequence $\{\mathbf{y}^{(j)}\}_{j \in [k-(M-1), k]}$, where M is a positive

integer. Denote this probability by $\Pr\{e|\mathbf{s}^{(k)}\}$, where the *gate state* $\mathbf{s}^{(k)}$ at time k is defined as $\mathbf{s}^{(k)} = (\mathbf{y}^{(j)})_{j \in [k-(M-1), k]}$. Obviously, for $M = 1$, we have the classical failure model. As previously stated, in our one-step majority logic decoder only XOR gates are unreliable. The number of states grows exponentially with M and ρ , i.e., for an $(\rho - 1)$ -input XOR gate used in our decoder there are $2^{M(\rho-1)}$ states.

The inputs of a (perfect) majority logic gate are the outputs of γ XOR gates in the neighboring check nodes. Thus, at time k these gates can be associated with a *state array* $\sigma^{(k)} = (\mathbf{s}_1^{(k)}, \mathbf{s}_2^{(k)}, \dots, \mathbf{s}_\gamma^{(k)})$, whose elements represent states of particular XOR gates. Based on $\sigma^{(k)}$, an *error probability vector* can be formed as $\epsilon^{(k)} = (\epsilon_1^{(k)}, \epsilon_2^{(k)}, \dots, \epsilon_\gamma^{(k)})$, $\epsilon_m^{(k)} = \Pr\{e = 1|\mathbf{s}_m^{(k)}\}$, $1 \leq m \leq \gamma$. The values of error probability vector can be obtained by measurements or by simulation of selected semiconductor technology. Thus, in our analysis we assumed that these values are known.

IV. ANALYSIS OF THE FAULTY DECODER

In this section we present an analytical method for performance evaluation of an ensemble of regular LDPC codes with girth no smaller than six decoded by a faulty one-step majority decoder described in previous section.

In a Tanner graph of a code with girth at least six, the variable nodes connected to the the neighboring γ checks, E_v , of a variable node v , are all distinct. Without loss of generality we can label the check nodes as $E_v = [\gamma]$, and the set of the $\gamma(\rho-1)$ variable nodes by the elements of $[\gamma(\rho-1)]$. Consider now a weight i ($i > 0$) error pattern on these variable nodes, and consider the computation tree for a variable node v . It is a tree with the root v and the leafs in $[\gamma(\rho-1)]$, i out of which are erroneous. The distribution of the weight i error pattern on the leaf nodes of the computation tree can be represented by a restricted $[\rho-1]$ -composition of i with γ parts. Each part λ_m , $1 \leq m \leq \gamma$ represents the number of erroneous nodes connected to the m -th check node (XOR gate). Since each check node in E_v is connected to $\rho-1$ variables other than v , the part size cannot exceed $\rho-1$. For example, if $\gamma = 3$, $E_v = \{1, 2, 3\}$ and $i = 3$, $\lambda = (2, 0, 1)$ corresponds to an error pattern in which the errors are at $\lambda_1 = 2$ variable nodes connected to the check $c = 1$ and $\lambda_3 = 1$ variable nodes connected to the check $c = 3$, while no variable connected to $c = 2$ is in error.

In a decoder built entirely from reliable components, every odd number of erroneous neighboring nodes connected to the same XOR gate will result in sending an incorrect bit estimate to node v . The number of incorrect estimates sent to node v in the error pattern λ can be obtained as follows

$$\delta = \sum_{t=1}^{\lceil \frac{\rho-1}{2} \rceil} f_{2t-1}, \quad (3)$$

where f_{2t-1} , $1 \leq t \leq \lceil \frac{\rho-1}{2} \rceil$, are defined in Section II-C.

In a perfect decoder determining the value δ is sufficient to know if bit x_v is correctly decoded, assuming the error pattern λ . But, in the faulty decoder, due to XOR gate failures, some

of the correct estimates can become incorrect, and vice versa, some incorrect estimates can be received as correct ones.

Let ϵ be the error probability vector of unreliable XOR gates, used for decoding bit x_v . Denote as $\epsilon^o = (\epsilon_1^o, \epsilon_2^o, \dots, \epsilon_\delta^o)$ a vector of error probabilities that correspond to all XOR gate which, if operate perfectly, would send incorrect bit estimates to node v . The remaining $\gamma-\delta$ elements of the error probability vector ϵ form a vector ϵ^e . Then we have the following lemma establishing the harmfulness of the error pattern corresponding to the composition λ .

Lemma 1. *The probability that node v receives p incorrect bit estimates in an error configuration corresponding to λ is*

$$P(p, \lambda, \epsilon) = \sum_{u=u_l}^{u_h} P_w\{u, \delta\} P_c\{p-u, \gamma-\delta\}, \quad (4)$$

where $u_l = \max(0, p-\gamma+\delta)$, $u_h = \min(\delta, p)$,

$$P_w\{u, \delta\} = \sum_{t=1}^{\binom{\delta}{u}} \prod_{\substack{m=1 \\ u>0}}^u \left(1 - \epsilon_{q_t, m}^o\right) \prod_{\substack{m=u+1 \\ u<\delta}}^{\delta} \epsilon_{q_t, m}^o, \quad (5)$$

$$P_c\{p-u, \gamma-\delta\} = \sum_{t=1}^{\binom{\gamma-\delta}{p-u}} \prod_{\substack{m=1 \\ p>u}}^{p-u} \epsilon_{r_t, m}^e \prod_{\substack{m=p-u+1 \\ p-u<\gamma-\delta}}^{\gamma-\delta} \left(1 - \epsilon_{r_t, m}^e\right), \quad (6)$$

where $q_{t,m}$ and $r_{t,m}$ denote the elements in t -th row and m -th column of the error configuration matrices $Q^{u, \delta}$ and $Q^{p-u, \gamma-\delta}$, respectively.

Proof: See Appendix A.

The following lemma gives the bit miscorrection probability under the fixed error probability vector, when the effect of different memory failure configurations is taken into account.

Lemma 2. *If memory registers fail independently with probability α , then the probability that the codeword bit x_v of a (γ, ρ) -regular LDPC code is incorrectly decoded by a faulty one-step majority logic decoder, whose gate fail according the error probability vector ϵ , is given by*

$$P_v(\alpha, \epsilon) = \sum_{i=0}^{\gamma(\rho-1)} \alpha^i (1-\alpha)^{\gamma(\rho-1)-i} \times \left[\frac{(-1)^\gamma + 1}{2} \alpha b_{i, \lceil \gamma/2 \rceil} + \sum_{p=\lceil \gamma/2 \rceil + 1}^{\gamma} b_{i, p} \right], \quad (7)$$

where

$$b_{i, p} = \sum_{j=1}^{T_{i, \rho-1}} P(p, \lambda^{i, j}, \epsilon) \prod_{m=1}^{\gamma} \binom{\rho-1}{\lambda_m^{i, j}} \quad (8)$$

and $\lambda^{i, j} = (\lambda_1^{i, j}, \lambda_2^{i, j}, \dots, \lambda_\gamma^{i, j})$, is the j -th composition of integer i , where $1 \leq j \leq T_{i, \rho-1}$.

Proof: See Appendix B.

Let $\{\mathbf{x}^{(k)}\}_{k \geq 0}$ be a codeword sequence stored in the memory registers. Clearly, decoding error of $\mathbf{x}^{(k)}$ depends

on $M - 1$ codewords previously read from a memory. Let $\mathbf{y}_{m,v} = \{\mathbf{y}_{m,v}^{(j)}\}_{j \in [k-(M-1), k]}$, $1 \leq m \leq \gamma$, $1 \leq v \leq n$, be sequence of code bits that, if stored with no errors, will appear at inputs of m -th XOR gate connected to node v , in the time interval $[k - (M - 1), k]$. Then, using the Lemmas 1 and 2 we formulate our main theorem which captures decoder performance under correlated data-dependent gate failures.

Theorem 1. *The average bit error rate (BER) of (γ, ρ) -regular LDPC code, when codeword sequence $\{\mathbf{x}^{(j)}\}_{j \in [k-(M-1), k]}$ is read from a memory and decoded by an unreliable one-step majority logic decoder is*

$$\begin{aligned} \bar{P}_e(\mathbf{x}^{(k)} | \mathbf{x}^{(k-1)}, \dots, \mathbf{x}^{(k-M+1)}) &= \frac{1}{n} \sum_{v=1}^n \sum_{t=1}^{2^{(\rho-1)\gamma M}} P_v(\alpha, \epsilon^{(t)}) \\ &\times \prod_{m=1}^{\gamma} \alpha^{d_H(\mathbf{s}_m^t, \mathbf{y}_{m,v})} (1 - \alpha)^{M(\rho-1) - d_H(\mathbf{s}_m^t, \mathbf{y}_{m,v})}. \end{aligned} \quad (9)$$

Proof: See Appendix C.

For a special case of von Neumann errors, the probability $P_v(\alpha, \epsilon^{(t)})$ is independent of state arrays and above expression reduces to equation (7). In addition, for the perfectly reliable hardware, the equation (4) simplifies to $P(p, \lambda, \epsilon) = 1$ only when $p = \delta$, and it is equal to zero otherwise.

V. NUMERICAL RESULTS

The analysis presented in previous section is general and can be used for analyzing decoders built in different nano-scale technologies. In this section we present numerical results for a special case of transient errors that are result of timing constrains – *timing errors*. Due to the sampling clock fluctuations or signal propagation delays, the output signal of a gate may be sampled or used in the next stage before it reaches a steady value, leading to an incorrect output. Such errors are dependent on gate history, i.e. data values processed by the gate in previous bit intervals. As the erroneous output of a logic gate will appear only if output changes its values it is usually sufficient to consider failure dependence on current and only one previous bit interval, i.e. $M = 2$. Thus, two complementary subsets of faulty XOR gate states, denoted as \mathcal{S}_1 and \mathcal{S}_2 , can be identified. The first subset \mathcal{S}_1 is constituted by states in which gate output remains unchanged and we have $\Pr\{e | \mathbf{s}^{(k)}\} = 0, \forall \mathbf{s}^{(k)} \in \mathcal{S}_1$. If a gate output changes its value, failure is possible and states from the subset \mathcal{S}_2 have non-zero error probabilities. For simplicity, we assume that all error probabilities are the same, i.e. $\Pr\{e | \mathbf{s}^{(k)}\} = \epsilon, \forall \mathbf{s}^{(k)} \in \mathcal{S}_2$.

The performance of the faulty decoder for two limiting cases are presented in Fig. 1. The best case, denoted as X_{00} corresponds to storage and consecutive decoding of two same codewords. The decoder will operate worst if two complementary codewords are stored, denoted as X_{01} . Performance of faulty decoders are upper bounded under decoding of X_{01} and lower bounded under X_{00} . When decoding X_{00} XOR gate failures are rare and have limited influence on decoder performance, which results in practically the same BER values

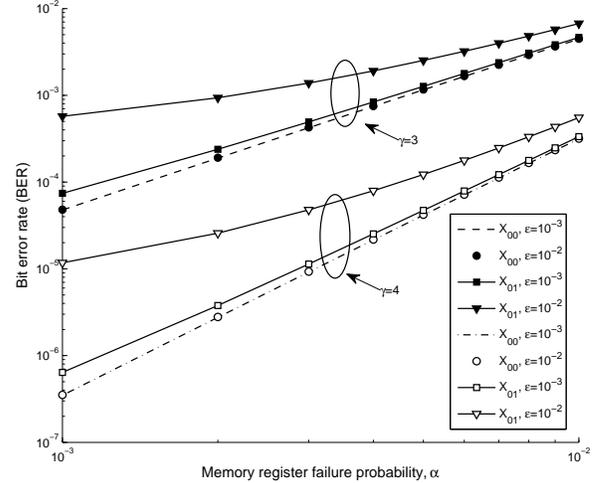


Fig. 1. Performance of faulty decoder under correlated gate failure ($\rho = 5$).

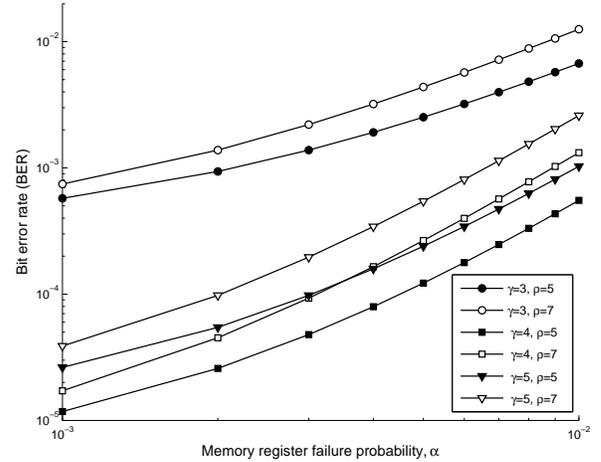


Fig. 2. Decoders upper bounds for different (γ, ρ) classes of LDPC codes ($\epsilon = 10^{-2}$).

for both ϵ values ($\epsilon = 10^{-3}, 10^{-2}$). Gap between bounds depends on column weight of matrix H . The upper bounds for different (γ, ρ) classes of LDPC codes are presented in Fig. 2, where negative influence of matrix H row weight can be noticed.

VI. CONCLUSION

In this paper we developed an analytical method used for performance evaluation of memory architectures based on one-step majority gate decoders under more realistic failure condition, then previously considered in the literature. Additionally, different ensembles of regular LDPC codes were examined and pronounced BER dependence of codewords order of decoding was noticed.

Our future research is directed to finding optimal mapping strategies that minimize negative effects of date-dependence.

Another direction is to study the performance of the Taylor-Kuznetsov architecture where the information is stored in a coded form and the values in the memory are periodically updated according to majority-logic decoding.

APPENDIX A (PROOF OF LEMMA 1)

$P_w\{u, \delta\}$ represents the probability that u , $0 \leq u \leq \delta$, of the δ incorrect estimates remain incorrect after operations in faulty XOR gates. Similarly, $P_c\{p-u, \gamma-\delta\}$ is the probability that $p-u$ initially correct estimates, become incorrect due to gate unreliability, $0 \leq p-u \leq \gamma-\delta$. Summation of all its products, under previously defined constraints for u , $u \in [u_l, u_h]$, gives the total probability of receiving p incorrect estimates.

APPENDIX B (PROOF OF LEMMA 2)

A composition $\lambda^{i,j}$ defines only the number of erroneous variable nodes connected to a particular check node, but errors may occur anywhere among the $\rho-1$ nodes. Thus, probability of the composition $\lambda^{i,j}$ leading to p incorrect estimates is equal to $P(p, \lambda^{i,j}, \epsilon) \prod_{m=1}^{\gamma} \binom{\rho-1}{\lambda_m^{i,j}}$. The overall probability that i errors in memory registers cause p incorrect bit estimates, denoted as $b_{i,p}$, can be calculated, by finding all of $T_{i,\rho-1}$ compositions of integer i .

The bit x_v will be incorrectly decoded if majority of its estimates are incorrect. Thus, for odd values of γ , only probabilities of p being greater or equal to $\lfloor \gamma/2 \rfloor + 1$ leads to miscorrection. If γ is even, then there is a possibility of a tie (equal number of correct and incorrect estimates). For such cases $\gamma/2$ incorrect estimates can result in miscorrection, which happens with probability $\alpha b_{i, \lfloor \gamma/2 \rfloor}$. We can now express bit miscorrection probability if i of its neighboring nodes are erroneous, as follows

$$\Pr\{\hat{r}_v \neq x_v | i \text{ errors}\} = \frac{(-1)^\gamma + 1}{2} \alpha b_{i, \lfloor \gamma/2 \rfloor} + \sum_{p=\lfloor \gamma/2 \rfloor + 1}^{\gamma} b_{i,p},$$

where a binary factor $((-1)^\gamma + 1)/2$ is added as a parity indicator of γ . An i error configuration occurs with probability $\Pr\{i \text{ errors}\} = \alpha^i (1-\alpha)^{\gamma(\rho-1)-i}$. The product $\Pr\{i \text{ errors}\} \Pr\{\hat{r}_v \neq x_v | i \text{ errors}\}$ can be calculated for all possible values of i , $0 \leq i \leq \gamma(\rho-1)$ and their sum gives the total probability of the bit miscorrection.

APPENDIX C (PROOF OF THEOREM 1)

The equation (7) gives miscorrection probability for an arbitrary chosen bit under one hardware failure scenario, i.e. one state array $\sigma^{(t)}$.

A particular XOR state $s_m^{(t)}$, $1 \leq m \leq \gamma$, will appear if memory failures change only certain bits of code sequence $\mathbf{y}_{m,v}$. The number of such bits is equal to the Hamming distance between perfectly stored code sequence and the XOR state $s_m^{(t)}$. As the inputs of XOR gates are not mutually dependent, the probability of the state array $\sigma^{(t)}$ occurrence can be derived by multiplexing individual XOR state probabilities and we have

$$P(\sigma^{(t)}) = \prod_{m=1}^{\gamma} \alpha^{d_H(s_m^{(t)}, \mathbf{y}_{m,v})} (1-\alpha)^{M(\rho-1)-d_H(s_m^{(t)}, \mathbf{y}_{m,v})}.$$

The error probability of a bit $x_v^{(k)}$ under assumption that fixed sequence of $M-1$ codewords was previously read from the memory can be derived by summing the products $P(\sigma^{(t)}) P_v(\alpha, \epsilon^{(t)})$ obtained for all possible error vectors $\epsilon^{(t)}$, $1 \leq t \leq 2^{(\rho-1)\gamma M}$ and the BER can be derived by performing one additional averaging over all code bits.

ACKNOWLEDGEMENT

This work was supported by the Seventh Framework Programme of the European Union, under Grant Agreement number 309129 (i-RISC project). It is also funded in part by the NSF under grants CCF-0963726 and CCF-1314147.

REFERENCES

- [1] S. Ghosh and K. Roy, "Parameter variation tolerance and error resiliency: New design paradigm for the nanoscale era," *Proceedings of the IEEE*, vol. 98, no. 10, pp. 17181751, Oct. 2010.
- [2] M. Taylor, "Reliable information storage in memories designed from unreliable components," *Bell System Technical Journal*, 47, pp. 2299-2337, 1968.
- [3] A. Kuznetsov, "Information storage in a memory assembled from unreliable components," *Problems of Information Transmission*, 9, pp. 254-264, 1973.
- [4] B. Vasic, S. K. Chilappagari, S. Sankaranarayanan, and R. Radhakrishnan, "Failures of the Gallager B decoder: analysis and applications," in *Proceedings of 2nd Information Theory and Applications Workshop (ITA 2006)*, San Diego, CA, USA, Feb. 2006, paper 160, [Online Available:] <http://ita.ucsd.edu/workshop/06/papers/160.pdf>
- [5] S. K. Chilappagari, B. Vasic, "Fault tolerant memories based on expander graphs," in *Proceedings of IEEE Information Theory Workshop*, pp. 126-131, Tahoe City, CA, USA, Sep. 2-6 2007.
- [6] B. Vasic and S. K. Chilappagari, "An Information Theoretical Framework for Analysis and Design of Nanoscale Fault-Tolerant Memories Based on Low-Density Parity-Check Codes," *IEEE Transactions on Circuits and Systems I, Regular Papers*, vol. 54, no. 11, pp. 2438-2446, Nov. 2007.
- [7] L. Varshney, "Performance of LDPC codes under faulty iterative decoding," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4427-4444, July 2011.
- [8] S. M. Sadeh Tabatabaei Yazdi, H. Cho, L. Dolecek, "Gallger B decoder on noisy hardware," *IEEE Transactions on Communications*, vol. 61, no. 5, pp. 1660-1673, May 2013.
- [9] F. Leduc-Primeau and W. Gross, "Faulty Gallager-B decoding with optimal message repetition," in *Proceedings of 50th Allerton Conference on Communication, Control, and Computing*, Monticello, USA, Oct. 2012, pp. 549-556.
- [10] C. H. Huang and L. Dolecek, "Analysis of finite alphabet iterative decoders under processing errors," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, pp. 5085-5089.
- [11] C. Kameni Ngassa, V. Savin and D. Declercq, "Min-Sum-based decoders running on noisy hardware," in *Proceedings of IEEE GLOBECOM*, Atlanta, USA, Dec. 2013.
- [12] L. D. Rudolf, "A class of majority logic decodable codes," *IEEE Transactions on Information Theory*, vol. 13, no 2, pp. 305-307, Apr. 1967.
- [13] R. Radhakrishnan, S. Sankaranarayanan, and B. Vasic, "Analytical performance of one-step majority logic decoding of regular LDPC codes," in *Proceedings of IEEE International Symposium on Information Theory (ISIT 2007)*, Nice, France, June 2007, pp. 231-235.
- [14] S. Chilappagari, M. Ivkovic, B. Vasic, "Analysis of one step majority logic decoders constructed from faulty gates," in *Proceedings of IEEE International Symposium on Information Theory (ISIT 2006)*, Seattle, USA, July 2006, pp. 469-473.
- [15] J. Von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in *Automata Studies*, C.E. Shannon and J. McCarty, eds., Princeton Univ. Press, pp. 43-98, 1956.
- [16] M. E. Malandro, "Integer compositions with part sizes not exceeding k ," [Online Available] <http://arxiv.org/abs/1108.0337>