On Fault Tolerance of the Gallager B Decoder under Data-Dependent Gate Failures

Srdan Brkic, *Student Member, IEEE,* Omran Al Rasheed, *Student Member, IEEE,*
Predrag Ivaniš, *Member, IEEE,* and Bane Vasić, *Fellow, IEEE*

***Abstract*—In this letter we characterize the effect of data-dependent gate failures on the performance of the Gallager B decoder of low density parity check codes. We show that this type of failures makes the decoder dependent on a transmitted codeword, thus rendering inapplicable the traditional analysis tools such as density evolution and trapping sets. By using Monte Carlo simulations, we identify two operating regions - one in which hardware unreliability leads to significant performance degradation, and one in which the performance loss is negligible. Based on these results, we propose a simple modification of the decoder that ensures its fault-tolerance.**

***Index Terms*—data-dependence, fault-tolerance, Gallager B decoder, low-density parity-check codes, timing errors.**

I. Introduction

Increasingly stringent requirements for semiconductor device energy-efficiency has lead to a point where computation performed by using these devices is no longer reliable [1]. The need to ensure fault-tolerance on inherently unreliable hardware has resulted in an increased interest in the analysis and design of novel and powerful error control schemes. The main direction of recent research in this area includes investigation of low-density parity-check (LDPC) codes and their decoders implemented on unreliable hardware, which relies on a theoretical framework developed by Vasic and Chilappagari [2]. The large body of knowledge on the analysis of codes on graphs and iterative decoding, has enabled further progress in the analysis of fault-tolerant schemes based on LDPC codes (see [3]–[7] and references therein).

The majority of the results on decoding by circuits made of unreliable hardware relies on modelling the logic gate unreliability as transient, independent failures - a model proposed by von Neumann [8] in the fifties. In the von Neumann failure model, in each clock cycle, components of a (clocked) Boolean network fail with some known probability. Additionally, failures of a given component are independent of those in previous clock cycles and independent of failures of other components. The simplicity of this model makes it amenable to theoretical analysis, but at same time limits its applicability. In new energy-efficient CMOS technologies, the logic gate failures are highly data-dependent and correlated in time [9]. The main source of incorrect gate outputs in these technologies is timing violations, or *timing errors*. They depend on logic gate inputs, but they are most damaging when the gate output changes its value [9], [10].

In order to characterize the hardware unreliability phenomenon accurately, Brkic *et al.* in [7] used a Markov chain timing error model, which has enabled them to analyze the behavior of one-step majority logic decoders. Timing errors in the context of the stochastic decoders have been considered recently by Perez-Andrade *et al.* [11] who have shown an inherent tolerance to timing errors of these type of decoders. While the error injection at transistor level was thoroughly analyzed, the decoder data-dependence was not considered.

In this letter we investigate the effects of timing errors to the performance of the Gallager B decoder, a simple hard decision decoder with a good trade off between the complexity and the error-correcting capability. Based on the Markov model proposed in [7], we evaluate the frame error rate (FER) for several classes of LDPC codes. We first demonstrate the inadequacy of von Neuman error model, and then show that the decoder performance is highly dependent on the sequence of codewords that is being decoded. Finally, we propose a simple modification of the decoder that increases the robustness to timing errors. Due to the lack of analytic techniques suitable to this failure model, we use Monte Carlo simulations.

II. Preliminaries

A. LDPC codes and Decoding Algorithm

Let (N, K) be an LDPC code, with code length N and code rate K/N, represented by a bipartite graph $G = (V \cup C, E)$, where V is the a of N variable nodes, C is a set of check nodes, and E is a set of edges. An edge is an unordered pair (v, c) which connects two nodes $v \in V$ and $c \in C$. The parity check matrix $\mathbf{H}$ is a bi-adjacency matrix of G. Nodes v and c are called *neighbors* iff $\mathbf{H}_{c,v} = 1$, i.e. there is an edge between them. Let $\mathcal{N}_v$ ($\mathcal{N}_c$) be a set of neighbors of a variable node v (check node c). In this letter we consider (γ, ρ)-regular binary LDPC codes which means that $|\mathcal{N}_v| = \gamma$, $\forall v \in V$ and $|\mathcal{N}_c| = \rho$, $\forall c \in C$, where $|\cdot|$ denotes cardinality.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ denote a codeword of an LDPC code that is transmitted over a Binary Symmetric Channel (BSC) with crossover probability α, where $x_v \in \{\pm 1\}$ is the polar representation of a bit value associated with a variable node v and let a vector received by the Gallager

This work was supported by the Seventh Framework Programme of the European Union, under Grant Agreement number 309129 (i-RISC project) and by the Serbian Ministry of Science under project TR32028. The work of B. Vasić was supported in part by the NSF under grants CCF-0963726 and CCF-1314147 and the United States Department of State Bureau of Educational and Cultural Affairs through the Fulbright Scholar Program.

S. Brkic, O. Al Rasheed and P. Ivaniš are with the University of Belgrade, Serbia, School of Electrical Engineering, (e-mails: srdjan.brkic@ic.etf.rs, omrano84@hotmail.com, predrag.ivanis@etf.rs), B. Vasić is with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, 85721 USA (e-mail: vasic@ece.arizona.edu).

B decoder from the BSC be $\mathbf{y} = \{y_1, y_2, \ldots, y_N\}$. Let $\nu_{v,c}^{(\ell)}$ ($\nu_{c,v}^{(\ell)}$) be messages passed on an edge (v, c) from(to) variable node to(from) check node during the ℓ-th decoding iteration, respectively. We next summarize the Gallager B decoder.

- *Variable to check node update*: For each variable node $v \in V$. At iteration $\ell = 0$: $\nu_{v,c}^{(0)} = y_v$, $\forall c \in \mathcal{N}_v$. At iteration $\ell > 0$:

$$\nu_{v,c}^{(\ell)} = \begin{cases} -y_v & \text{if } |\{c' \in \mathcal{N}_v \setminus c : \nu_{c',v}^{(\ell-1)} = -y_v\}| > \lfloor \gamma/2 \rfloor, \\ y_v & \text{otherwise.} \end{cases} \tag{1}$$

- *Check to variable node update*. For each check node $c \in C$ and $\forall v \in \mathcal{N}_c$, at iteration $\ell \geq 0$:

$$\nu_{c,v}^{(\ell)} = \prod_{v' \in \mathcal{N}_c \setminus \{v\}} \nu_{v',c}^{(\ell)}. \tag{2}$$

The decoding is terminated when all parity-check equations are satisfied or the maximum number of iterations is reached.

The decoder comprises of the *processing units* that correspond to the nodes in the bipartite graph representation of the decoder. Each check node (CN) unit is composed of ρ XOR gates, each with $\rho - 1$ inputs, needed for calculation of the check to variable node messages. A variable node (VN) processing unit calculates the variable to check node messages, and employs γ majority logic (MAJ) gates, each with $\gamma - 1$ inputs [2]. The decoder also requires additional logic gates for the final bit estimations and parity-checks calculation. If we allow these gates to be unreliable, the performance of the decoder would be determined by the failure probabilities of these gates, not by the error control scheme. Thus, it is reasonable to assume that these gates are perfect, and that only gates used to calculate messages that are passed on edges of the bipartite graph are faulty. Such reliable gates in the final decision circuitry can be realized by using larger transistors, slowing down the clock or using higher voltage supply. Similar assumption was also used in other relevant literature [2], [7].

Note that we considered that the threshold in the variable-to-check-node update operations is fixed, opposed to the original solution proposed by Gallager. Although the threshold adaptation increases the performance, it requires the additional logic which makes it more complex than described algorithm. The use of a fixed threshold value represents a good trade-off between performance and efficiency, especially if the targeted error rates are low.

B. The Failure Model

The recent work in the area of low-powered combinatorial circuits has identified the increased signal propagation delay as the main cause of the circuits unreliability [10], [12]. Timing violations happen when the propagation delay is longer than what can be tolerated by the circuit design, which results in propagating an outdated output value to the rest of the circuit.

Let $f : \{\pm 1\}^m \to \{\pm 1\}$, $m > 1$, be an m-argument Boolean function, which at time instant ℓ produces the result $z^{(\ell)} = f(y_1^{(\ell)}, y_2^{(\ell)}, \ldots, y_m^{(\ell)})$, where $y_1^{(\ell)}, y_2^{(\ell)}, \ldots, y_m^{(\ell)}$ are input arguments at time ℓ. Due to unreliability of the logic gates used to calculate f, the result is not $z^{(\ell)}$ but $\mu^{(\ell)} = z^{(\ell)} e^{(\ell)}$, where $e^{(\ell)} \in \{\pm 1\}$ is the error at time ℓ. In the von Neumann failure model $e^{(\ell)}$ is a Bernoulli random variable, and does not depend on the gate input arguments [8].

In the timing failure model $e^{(\ell)}$ is data-dependent, and the probability that the logic gate fails to switch is $\Pr\{e^{(\ell)} = -1 | z^{(\ell)} \neq z^{(\ell-1)}\} = \varepsilon$, where $\varepsilon > 0$. On the other hand, when the gate output is unchanged during two consecutive time instants, the function f is always correctly computed as assumed in [9] and [10], i.e. $\Pr\{e^{(\ell)} = -1 | z^{(\ell)} = z^{(\ell-1)}\} = 0$. This corresponds to the gate switching probabilistic model [10], that was shown to have reduced complexity with minor degradation of accuracy when compared to more complex models that take into account that different input patterns can cause gate failures with different probabilities [9], [10].

The value ε depends on the technology parameters and can be obtained experimentally or by the circuit-level simulations. It typically differs from a gate type to a gate type. We denote by $\varepsilon_{\oplus}$ and ε_{MAJ} the failure rates of XOR and MAJ gates, respectively. The gate output at time instant ℓ, $\mu^{(\ell)}$, is obtained by the mapping $\Upsilon : \{\pm 1\}^3 \to \{\pm 1\}$ as follows

$$\mu^{(\ell)} = \Upsilon(z^{(\ell)}, z^{(\ell-1)}, e^{(\ell)}) = z^{(\ell)} (e^{(\ell)})^{(z^{(\ell)} - z^{(\ell-1)})/2}, \tag{3}$$

where $\Pr\{e^{(\ell)} = -1\} = \varepsilon$. Note that the failure model is being redefined to take advantage of the fact that $e^{(\ell)} = 1$ with probability 1, when $z^{(\ell)} = z^{(\ell-1)}$.

III. Failures of the Gallager B Decoding Algorithm

The gate failure model introduced in Section II-B acts as a binary channel with memory. We next explain how that memory makes the performance of the faulty Gallager B decoder dependent on the transmitted codewords. Consider the received vector $\mathbf{y} = \mathbf{x} \cdot \mathbf{n}$, where "$\cdot$" denotes pointwise multiplication of the codeword vector $\mathbf{x}$ and the noise vector $\mathbf{n}$. Let $\nu_{v,c}^{(-1)}$ and $\nu_{c,v}^{(-1)}$ be output values of the update functions sent between variable node v and check node c in the time instant prior to the initial decoding iteration of a current codeword. These are the messages sent in the last iteration of decoding the previous received word. The following theorem defines the conditions under which the probability of successful decoding is independent of the transmitted codeword.

Theorem 1. *The frame error rate of the Gallager B decoder in the presence of timing errors is independent of the transmitted codeword* $\mathbf{x}$ *iff* $\nu_{v,c}^{(-1)} = x_v A_v$ *and* $\nu_{c,v}^{(-1)} = x_v B_v$, $\forall v \in V$ *and* $\forall c \in C$, *where* $A_v, B_v \in \{\pm 1\}$.

Proof: Let $\mu_{c,v}^{(\ell)}(\mathbf{y})$ and $\mu_{v,c}^{(\ell)}(\mathbf{y})$ be, respectively, the messages passed from a check node c to a variable node v and from a variable node v to a check node c, at iteration ℓ, given the received $\mathbf{y}$. They are obtained from Eq. (3) based on the correctly computed messages $\nu_{c,v}^{(\ell)}(\mathbf{y})$ and $\nu_{v,c}^{(\ell)}(\mathbf{y})$ and the corresponding error values $e_{c,v}^{(\ell)}$ and $e_{v,c}^{(\ell)}$.

Let us assume $\nu_{v,c}^{(-1)} = x_v A_v$ and $\nu_{c,v}^{(-1)} = x_v B_v$, for some $A_v, B_v \in \{\pm 1\}$. We use mathematical induction to prove the theorem statement. From the variable node symmetry condition at iteration $\ell = 0$, we have $\nu_{v,c}^{(0)}(\mathbf{y}) = \nu_{v,c}^{(0)}(\mathbf{x} \cdot \mathbf{n}) = x_v \nu_{v,c}^{(0)}(\mathbf{n})$ [3]. As these values are received from the channel

(not calculated using the combinatorial circuit), they are passed without errors to the neighboring check nodes. From the check node symmetry follows $\nu_{c,v}^{(0)}(\mathbf{y}) = x_v \nu_{c,v}^{(0)}(\mathbf{n})$ [3], and we have

$$\begin{aligned}\mu_{c,v}^{(0)}(\mathbf{y}) &= \Upsilon(\nu_{c,v}^{(0)}(\mathbf{y}), x_v B_v, e_{c,v}^{(0)}) \\ &= x_v \nu_{c,v}^{(0)}(\mathbf{n}) (e_{c,v}^{(0)})^{x_v(\nu_{c,v}^{(0)}(\mathbf{n}) - B_v)/2} \\ &= x_v \Upsilon(\nu_{c,v}^{(0)}(\mathbf{n}), B_v, e_{c,v}^{(0)}) = x_v \mu_{c,v}^{(0)}(\mathbf{n}). \end{aligned} \quad (4)$$

Similarly, we conclude $\mu_{v,c}^{(1)}(\mathbf{y}) = x_v \Upsilon(\nu_{v,c}^{(1)}(\mathbf{n}), A_v, e_{v,c}^{(1)}) = x_v \mu_{v,c}^{(1)}(\mathbf{n})$.

Let us assume $\mu_{v,c}^{(\ell)}(\mathbf{y}) = x_v \mu_{v,c}^{(\ell)}(\mathbf{n})$, $\forall v \in V$, $\forall c \in C$ and $\ell > 1$. From the fact that $\prod_{v:\mathcal{N}(c)} x_v = 1$ and from the check node symmetry defined in [3], it follows that $\nu_{c,v}^{(\ell)}(\mathbf{y}) = x_v \nu_{c,v}^{(\ell)}(\mathbf{n})$. Then, similarly as in Eq. (4), we have $\mu_{c,v}^{(\ell)}(\mathbf{y}) = \Upsilon(x_v \nu_{c,v}^{(\ell)}(\mathbf{n}), x_v \nu_{c,v}^{(\ell-1)}(\mathbf{n}), e_{c,v}^{(\ell)}) = x_v \mu_{c,v}^{(\ell)}(\mathbf{n})$. Furthermore, by invoking again the variable node symmetry condition and applying it to Eq. (3), we obtain $\mu_{v,c}^{(\ell+1)}(\mathbf{y}) = x_v \mu_{v,c}^{(\ell+1)}(\mathbf{n})$. As all messages passed between a node v and its neighbors are equal to the product of x_v and the corresponding message when $\mathbf{n}$ is received, the decoder performance is independent of the transmitted codeword.

On the other hand, if the theorem conditions were not satisfied, i.e. if $\nu_{v,c}^{(-1)} \neq x_v A_v$ or $\nu_{c,v}^{(-1)} \neq x_v B_v$, then it would follow that $\mu_{v,c}^{(1)}(\mathbf{y}) \neq x_v \mu_{v,c}^{(1)}(\mathbf{n})$. As messages $\mu_{v,c}^{(1)}$ continue to further propagate in the subsequent iterations, the error correction, and thus FER, depends on $\mathbf{x}$. ■

The conditions required for FER independence of $\mathbf{x}$ are highly unrealistic. They can be satisfied only by adjusting the initial logic gate states to a transmitted codeword, which is unknown to the decoder. Thus, in the case of timing errors the decoder error-correction capability is conditional on $\mathbf{x}$.

On the other hand, we have shown that the main reason for such decoder behavior is related to failing to satisfy the check and the variable node symmetry conditions in the first decoding iteration. If the first iteration were free of logic gate failures, the symmetry conditions would be satisfied. The fault-free iteration can be achieved by forcing all transistors in the decoding circuitry to reach a stationary state. Practically, this can be done by slowing down the clock in the first iteration and letting the signal level stabilize. Since the clock is slower, there are no timing errors and the computations are reliable. The decoder in which the first iteration is reliable in the rest of the letter is called *the modified Gallager B decoder*.

IV. Numerical Results

The past work related to the faulty Gallager B decoder was mostly dedicated to the infinite code length analysis under independent identically distributed (i.i.d.) failures (von Neumann failure model) [4]–[6]. Due to the decoder asymmetry, the density evolution technique cannot be applied to the timing error model, when failures are present in the first iteration. Although, the modified decoder satisfy the symmetry conditions, the presence of memory in the VN and CN operations cause the complexity of density evolution computation to grow at least exponentially with the number of iterations [13]. For that reason in this letter we focus on the finite length simulation analysis, where the maximal number of decoding iterations is limited to 100. We analyze LDPC codes free of small trapping sets. Such codes can be constructed, for example, from Latin Squares (LS) [14], or based on Progressive Edge Growth (PEG) technique [15]. In order to capture the effects of data-dependence three simulation modes are used:

- mode M_0: only the all-zero codewords are transmitted;
- mode M_H: the two codewords with large Hamming distance are transmitted alternately in a consecutive transmissions;
- mode M_R: randomly chosen codewords are transmitted.

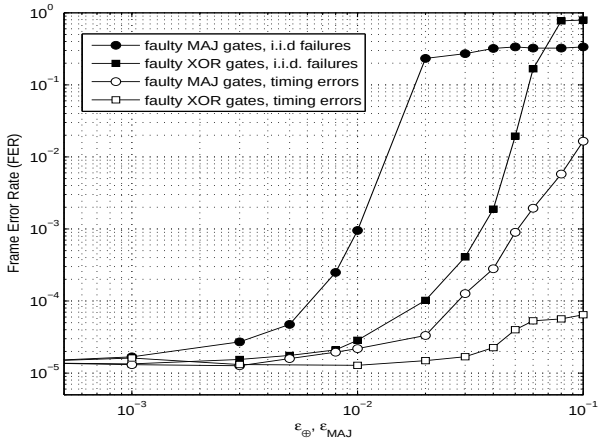

Fig. 1: The comparison of i.i.d. and timing error models for the LS(155,64) code (α=0.01).

In Fig. 1 we illustrate the decoder performance for the case of LS(155,64) code [14] evaluated under two failure models, when the most realistic setup M_R is used. It is a (3,5)-regular code with $|V| = 155$ and $|C| = 93$. We consider two different scenarios: (i) when CN processing is reliable and only MAJ gates are faulty, and (ii) when VN processing is reliable and only XOR gates are faulty. Note that no performance degradation is observed for logic gate failure rates below 10^{-3} regardless of the failure model. When logic gates failures become more frequent, a *threshold* is reached, and the FER rapidly increases. The threshold depends on the logic gate type and has lower values when the MAJ gates are faulty.

The timing error model reduces to the von Neumann model when a gate is constantly in a "bad state", i.e. changes its output value in every clock cycle. In that sense, the von Neumann model can be seen as pessimistic and misleading, which is especially pronounced in a case of highly unreliable XOR logic gates. The use of the timing error model reveals robustness of the Gallager B decoder to XOR gate failures. For example, even for the large bad state failure probability $\varepsilon_{\oplus} = 0.1$ the decoder retains low FER value. On the other hand, according to the von Neumann model, the faulty decoder is in average outperformed by an uncoded system.

For low and moderate channel error rates, in most of the cases only a few of 155 code bits are received incorrectly and the most of $\nu_{v,c}^{(0)}$ messages represent the correct bit estimates. If during the first iteration the number of gate failures is also low, most of the variable to check node messages will remain

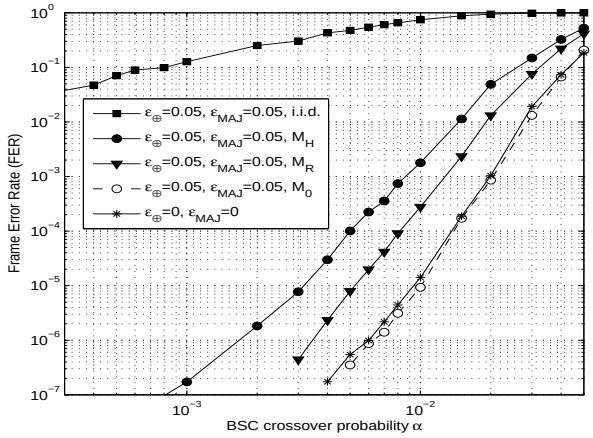

Fig. 2: The performance dependence on codeword decoding order for LS(155,64) LDPC code.

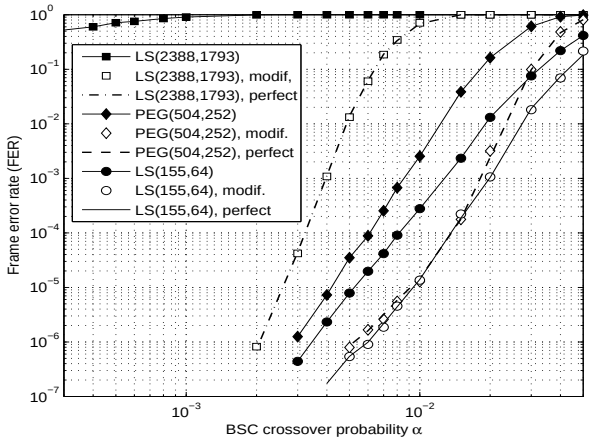

Fig. 3: The performance comparison of different codes ($\varepsilon_{\oplus} = \varepsilon_{MAJ} = 0.05$) under simulation mode M_R.

unchanged. However, if the number of gate failures is not negligible, after the first iteration, the number of incorrect bit estimates can be significantly increased. The effect of gate failures in the first iteration is illustrated in Fig. 2. It can be noticed that these failures have dominant influence on the decoder performance. When the same word is transmitted (mode M_0) there is no performance degradation even for high bad state failure rate ($\varepsilon_{\oplus} = \varepsilon_{MAJ} = 0.05$). On the other hand, mode M_H reveals that the worst case FER degradation can be of several orders of magnitude. Significant degradation is also notable in the more realistic scenario, which corresponds to the successive transmission of randomly chosen codewords.

In Fig. 3 we illustrate the improvement achieved by using the modified Gallager B decoder in which the first iteration is fault-free, for several $\gamma = 3$ codes. It can be seen that the modified Gallager B decoder, although built from unreliable components, performs approximately the same as perfectly reliable decoder. On the other hand, the FER of the Gallager B decoder without modification is several magnitude higher. For example, the code LS(2388,1793) is non-operational for wide range of channel error probabilities.

V. Conclusion

We have shown that the performance degradation of the Gallager B decoder caused by the timing violations depends of the sequence of transmitted codewords. In addition, we proposed the modification that ensures the decoder robustness to hardware unreliability. This result raises a number of open questions in analysis of faulty decoders. It is known that the capacity of LDPC codes under the von Neumann error model can not be achieved [3], [4]. However, the finite code length analysis presented in this letter suggests more promising results for the timing error model. Additionally, in our future work we will apply the presented framework to investigation and design of fault-tolerant LDPC code-based memories.

References

[1] S. Ghosh and K. Roy, "Parameter variation tolerance and error resiliency: New design paradigm for the nanoscale era," *Proc. of the IEEE*, vol. 98, no. 10, pp. 1718–1751, Oct. 2010.

[2] 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 Trans. on Circuits and Syst. I, Reg. Papers*, vol. 54, no. 11, pp. 2438–2446, Nov. 2007.

[3] L. Varshney, "Performance of LDPC codes under faulty iterative decoding," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4427–4444, July 2011.

[4] S. M. Sadegh Tabatabaei Yazdi, H. Cho, and L. Dolecek, "Gallager B decoder on noisy hardware," *IEEE Trans. Commun.*, vol. 61, no. 5, pp. 1660–1673, May 2013.

[5] C. H. Huang, Y. Li, and L. Dolecek, "Gallager B LDPC decoder with transient and permanent errors," *IEEE Trans. Commun.*, vol. 62, no. 1, pp. 15–28, Jan. 2014.

[6] F. Leduc-Primeau and W. Gross, "Faulty Gallager-B decoding with optimal message repetition," in *Proc. of 50th Allerton Conf. Commun., Control, and Computing*, Monticello, USA, Oct. 2012, pp. 549–556.

[7] S. Brkic, P. Ivanis, and B. Vasic, "Analysis of one-step majority logic decoding under correlated data-dependent gate failures," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT 2014)*, Honolulu, USA, June–July 2014, pp. 2599–2603.

[8] 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, July 1956, pp. 43–98.

[9] J. Chen, C. Spagnol, S. Grandhi, E. Popovici, S. Cotofana, and A. Amaricai, "Linear compositional delay model for the timing analysis of subpowered combinational circuits," in *Proc. of IEEE Comp. Soc. Annual Symp. on VLSI*, July 2014.

[10] A. Amaricai, S. Nimara, O. Boncalo, J. Chen, and E. Popovici, "Probabilistic gate level fault modeling for near and sub-threshold cmos circuits," in *Proc. 17th Euromicro Conf. on Digital Syst. Design (DSD)*, Verona, Avg. 2014, pp. 473–479.

[11] I. Perez-Andrade, X. Zuo, R. Maunder, B. Al-Hashimi, and L. Hanzo, "Analysis of voltage- and clock-scaling-induced timing errors in stochastic LDPC decoders," in *Proc. IEEE Wireless Commun. and Networking Conf. (WCNC)*, Shanghai, Apr. 2013, pp. 4293–4298.

[12] S. Zaynoun, M. S. Khairy, A. M. Eltawil, F. J. Kurdahi, and A. Khajeh, "Fast error aware model for arithmetic and logic circuits," in *Proc. of 30th IEEE Int. Conf. on Computer Design (ICCD)*, Montreal, QC, Sept.–Oct. 2012, pp. 322–328.

[13] J. E. and A. Banihashemi, "Performance analysis of iterative decoding algorithms with memory over memoryless channels," *IEEE Trans. Commun.*, vol. 16, no. 12, pp. 3556–3566, Dec. 2014.

[14] D. V. Nguyen, S. K. Chilappagari, M. W. Marcellin, and B. Vasic, "On the construction of structured LDPC codes free of small trapping sets," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2280–2302, Apr. 2012.

[15] X. Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2012.