Hard-Decision Decoding of LDPC Codes Under Timing Errors: Overview and New Results

Srdan Brkic, Predrag Ivanis and Bane Vasić

Abstract—This paper contains a survey on iterative decoders of low-density parity-check (LDPC) codes built from unreliable logic gates. We assume that hardware unreliability comes from supply voltage reduction, which cause probabilistic gate failures, called *timing errors*. We are able to demonstrate robustness of simple Gallager B decoder to timing errors, when applied on codes free of small trapping sets, as well as positive effects that timing errors have on the decoding of codes with contains small trapping sets. Furthermore, we show that concept of guaranteed error correction can be applied to the decoders made partially from unreliable components. In contrast to the decoding under uncorrelated gate failures, we prove that bit-flipping decoding under timing errors can achieve arbitrary low error probability. Consequently, we formulate condition sufficient that memory architecture, which employs bit-flipping decoder, preserved all stored information.

Keywords—Bit-flipping, Gallager B, data-dependence, fault-tolerance, low-density parity-check codes, timing errors

I. INTRODUCTION

Replication of electronic devices and majority voting, proposed by von Neumann [1] and Winograd [2] is a classical way of increasing reliability of Boolean function computation and/or storage of information. Although simple to implement, it requires large quantity of redundant hardware in order to produce high level of reliability. Unfortunately, later attempts to reliably compute Boolean functions, built from unreliable logic gates, by using combinatorial schemes with lower redundancy were not successful. Dobrushin and Ortyukov [3] found Boolean functions for which construction of such schemes is even impossible.

On the other hand, in 1968. Taylor [4] and later Kuznetsov [5] provided much more promising results on reliable storage of information. Instead of multiplying memory cells, Taylor proposed that information is packed as a codeword of a low-density parity-check (LDPC) code and stored in registers, which are periodically refreshed by an error-correction circuit. Taylor proved that memory built only from unreliable components, with complexity increasing linearly with the quantity of stored information, is capable of preserving all information for arbitrary long time period.

The significance of Taylor's work, for long time not fully recognized in the coding theory community. However, four

decades after Taylor's original work ensuring fault-tolerance was recognized as one of the most significant challenges in the semiconductor industry [6], [7]. Shortly after, Taylor's work was rediscovered [8]–[14], which lead to a surge of interest in decoders of LDPC codes built from unreliable hardware, i.e., noise decoders. A number of relevant papers were dedicated to density evolution under uncorrelated gate failures [15]–[19], with rather pessimistic conclusions that arbitrary low error probability cannot be achieved, even when code length tends to infinity.

In contrary to a popular beliefs that hardware failures have negative impact on decoding, in our previous work we showed that deliberate failure insertion can be beneficial to the decoding process [20]–[22]. We attributed such behaviour to stochastic resonance, a phenomenon well exploited in neural systems, semiconductor and quantum devices.

In another research direction we investigated iterative decoders under realistic gate failure models such as correlated gate failures. The paper provides an overview of the current state of our research regarding behaviour of hard-decision decoders under special type of correlated gate failures, inevitable in synchronous CMOS circuits, called timing errors. Following this line of research we revisited an error-correction potential that noisy decoders have, which is not visible by the density evolution analysis under uncorrelated failures. In a series of articles we showed robustness of different error-correction techniques to timing errors – one-step majority logic decoders [23], Gallager B decoders [24], bit-flipping decoders [25], [26], as well as coded memory architectures [27]. In this paper we especially emphasize two major contributions: 1) discovered positive impact of timing errors to decoder's correction capability and 2) ability of noisy decoders to achieve arbitrary low error probability. The first phenomenon is explained on the example of Gallager B decoder, while the second contribution incorporates expander codes and bit-flipping decoders. Furthermore, we provided a lower bound on the number of channel errors that can be corrected by noisy bit-flipping decoders. In addition, we show that expander graph arguments can also be used to provide guarantees of reliable information storage, when noisy bit-flipping decoder is used as a correcting circuit.

The rest of the paper is organized as follows. Section II contains the preliminaries on codes on graphs, followed by the gate failure modeling approaches and state of the art density evolution analysis. In Section III we discuss the Gallager B decoding under timing errors, while guaranteed error correction of bit-flipping decoder is given in Section

P. Ivaniš and S. Brkic are with the University of Belgrade, School of Electrical Engineering, 11000 Belgrade, Serbia (e-mails: predrag.ivanis@etf.rs, srdjan.brkic@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).

IV. Section V discusses the application of LDPC codes in memories. Concluding discussion is given in Section VI.

II. PRELIMINARIES

A. Codes and Decoders on Graphs

Consider a graph-based representation of LDPC codes through a bipartite (Tanner) graph $G = (V \cup C, E)$, where V represents the set of n variable nodes, C is the set of $n\gamma/\rho$ check nodes, and E is the set of $n\gamma$ edges. An edge $e = (v, c)$ connects variable node $v \in V$ and check node $c \in C$ iff $H_{c,v} = 1$, where $\mathbf{H}$ denotes the parity check matrix of the code. All nodes connected to a particular node u are called neighbors and form a set $\mathcal{N}_u$. The number of neighbors of a variable node v (check node c) is denoted by $|\mathcal{N}_v| = \gamma_v$ ($|\mathcal{N}_c| = \rho_c$), where $|\cdot|$ denotes cardinality. If $\forall c \in C, \rho_c = \rho$, we say that the graph is ρ -right-regular and similarly if $\forall v \in V, \gamma_v = \gamma$ the graph is γ -left-regular. We say that the graph (and corresponding code) is (γ, ρ) -regular if it is in the same time ρ -right-regular and γ -left-regular. The length of the shortest cycle in Tanner graph is called girth, and denoted by g .

Let $\mathbf{x} = (x_1, x_2, \dots, x_n)$ denote a codeword of an LDPC code, where $x_v \in \{\pm 1\}$ represents the value associated with the variable node v . The vector received from the channel is represented by $\mathbf{y} = (y_1, y_2, \dots, y_n)$, where $\Pr\{x_v = y_v\} = 1 - p, \forall v \in V$.

An iterative decoder follows the factor-graph approach – nodes of Tanner graph iteratively exchange messages, where a message passed from a check c to a variable node v during the ℓ -th iteration we denote by $\mu_{c \rightarrow v}^{(\ell)}$, while a message passed in the opposite direction is denoted by $\nu_{v \rightarrow c}^{(\ell)}$. The messages that leave a particular node are calculated using functions implemented locally within the node (*node update functions*), based on the messages sent by its neighbours during the previous iterations.

During the past decades variety of different decoders have been proposed, with different performance-complexity trade-offs. In the most complex decoders, like Sum-Product algorithm (SPA), variable node update functions produce floating-point probability estimates, while the practically most significant ones are decoders which exchange quantized messages, called FAIDs (Finite-Alphabets Iterative Decoders), which in some cases even outperform SPA [28]. On the opposite side of the spectrum are hard-decision decoders in which, during the decoding iteration, a node sends/receives single binary value. Usually, check to variable message reveals if parity check is satisfied, while variable to check message contains sign of the transmitted bit likelihood denoted by $\Omega_{v,c}$. Thus,

$$\nu_{v \rightarrow c}^{(\ell)} = \begin{cases} \text{sgn}(\Omega_{v,c}) & \text{if } \Omega_{v,c} \neq 0, \\ y_v & \text{otherwise.} \end{cases}$$

where $\text{sgn}(\cdot)$ denotes signum function. On the other hand, check to variable messages are $\mu_{c \rightarrow v}^{(\ell)} = \prod_{v' \in \mathcal{N}_c \setminus v} \nu_{v' \rightarrow c}^{(\ell-1)}$.

The likelihood $\Omega_{v,c}$ is obtained by summing messages received from a set of neighbours $\mathcal{S}_{v,c} \subseteq \mathcal{N}_v$ as $\Omega_{v,c} = \sum_{c' \in \mathcal{S}_{v,c}} \mu_{c' \rightarrow v}^{(\ell)}$. We consider two classes of hard-decision

decoders: 1) message passing Gallager B decoders and 2) bit-flipping decoders, for which $\mathcal{S}_{v,c}$ takes following elements

$$\mathcal{S}_{v,c} = \begin{cases} \{\mathcal{N}_v \setminus c\} & \text{if Gallager B,} \\ \mathcal{N}_v & \text{if Bit-Flipping.} \end{cases}$$

Note that in the case of bit-flipping decoder variable to noise messages sent from a particular node are all the same. The decoding process terminates when all parity check equations are satisfied or maximal number of iterations is reached.

It should be observed that operations in the check node c require implementation of ρ_c XOR gates with $\rho_c - 1$ inputs. Similarly, variable node processor v is composed of γ_v majority logic (MAJ) gates with $\gamma_v - 1$ inputs, when Gallager B decoder is considered, or a single γ_v MAJ gate for the case of bit-flipping decoder.

B. Modeling Logic Gates Failures

Let $f : \{\pm 1\}^m \rightarrow \{\pm 1\}$, $m > 1$, be an m -argument node update function, which at decoding iteration ℓ produces the message $z^{(\ell)} = f(y_1^{(\ell)}, y_2^{(\ell)}, \dots, y_m^{(\ell)})$, where $y_1^{(\ell)}, y_2^{(\ell)}, \dots, y_m^{(\ell)}$ are input arguments at time ℓ . Due to unreliability of the logic gates used to calculate f , the result is not $z^{(\ell)}$ but $\hat{z}^{(\ell)} = z^{(\ell)}e^{(\ell)}$, where $e^{(\ell)} \in \{\pm 1\}$ is the error at time ℓ .

In the vast majority of reference literature related to coding-enhanced electronic devices, it is assumed that $e^{(\ell)}$ is a Bernoulli random variable, which statistics does not change with decoding iterations nor does it depend on function input arguments. In other words, gate failures are spatially and temporally uncorrelated, which correspond to the von Neumann failure model [1]. However, this modeling approach is too general and does not always produce practically relevant conclusions. For example, if hardware unreliability comes from supply voltage reduction failures are highly influenced by gate input arguments, i.e., we say that failures are *data-dependent*.

Reducing the power voltage supply prolongs the digital signal delay and makes it unfit for timing constraints established for the case of nominal (higher) voltage supply. If we do not slow down the system clock the logic gate output will not always switch correctly, leading to so called *timing errors* [7], [29], [30]. In other words we can write $\Pr\{e^{(\ell)} = -1 | z^{(\ell)} \neq z^{(\ell-1)}\} = \varepsilon$, where $\varepsilon > 0$. On the other hand, when the gate output does not change during a decoding iteration, we assume that signal propagation is zero, i.e., $\Pr\{e^{(\ell)} = -1 | z^{(\ell)} = z^{(\ell-1)}\} = 0$. It should be noted that, in practice, different gate input arguments (which force gate output to switch) cause failures with different probabilities [30]. However, guided by Einstein's observation that *everything should be made as simple as possible, but no simpler*, we sacrifice precision in order to produce more fundamental results. In that spirit, we can assume that ε denotes the highest possible failure probability obtained experimentally, or by the circuit-level simulations, for chosen technological parameters (clock frequency, supply voltage, temperature). Nevertheless, it typically differs from a gate type to a gate type and we denote by $\varepsilon_{\oplus}$ and ε_{MAJ} the failure rates of XOR and MAJ gates, respectively. Note that failure injection can be described

by mapping $\Upsilon : \{\pm 1\}^3 \rightarrow \{\pm 1\}$ where the actual gate output $\hat{z}^{(\ell)}$ can be obtained by

$$\hat{z}^{(\ell)} = \Upsilon(z^{(\ell)}, z^{(\ell-1)}, e^{(\ell)}) = z^{(\ell)}(e^{(\ell)})^{(z^{(\ell)} - z^{(\ell-1)})/2}.$$

In the subsequent sections we show how timing errors affects the performance of hard-decision decoders.

C. Density Evolution and Transient Gate Failures

Density evolution (DE) technique, is recognized as one of the most important mathematical tools used to predict performance of LDPC codes and message passing decoders, on various communication channels. Let $P_n^{(\ell)}(p)$ denote the bit error probability of an LDPC code of length n after ℓ decoding iterations, when transmission channel crossover probability is equal to p . The goal of DE analysis is to find *channel error probability threshold* p^* such that for all $p < p^*$, $\lim_{\ell \rightarrow \infty} P_\infty^{(\ell)}(p) = 0$ [31]. In other words, DE proves the existence of codes, for which in the asymptotic block length the Gallager B decoder made of fully reliable components can correct all received errors.

Recently, DE approach was extended to the case of noisy decoders and transient von Neumann gate failures. [15], [16], [19], [32]. For example, the bit error achievable for the noisy Gallager B decoder applied on (γ, ρ) -regular code ensemble can be expressed by the following recursive form [16]

$$P_\infty^{(\ell+1)} = (1 - 2\varepsilon_{MAJ}) \left[\sum_{i=0}^{\gamma-1-b} \Lambda_i(\delta(P_\infty^{(\ell)})) + p \times \frac{1 + (-1)^{\gamma+1}}{2} \Lambda_{b-1}(\sigma(P_\infty^{(\ell)})) \right] + \varepsilon_{MAJ}, \quad (1)$$

where $\Lambda_i(t) := \binom{\gamma-1}{i} t^i (1-t)^{\gamma-1-i}$, $\alpha(t) := 0.5 \times (1 + (1-2t)^{\rho-1})$, $\sigma(t) := (1-\varepsilon_\oplus)\alpha(t) + \varepsilon_\oplus(1-\alpha(t))$, $b = \lfloor (\gamma+1)/2 \rfloor$.

On the contrary to its reliable counterparts noisy Gallager B decoders, as it was shown in [16], do not achieve arbitrary low error probability, even in the asymptotic case. Instead, we usually speak of η -reliability. Even when $\varepsilon_{MAJ} = 0$, from Eq. (1) follows that, for $\gamma > 3$, $\lim_{\ell \rightarrow \infty} P_\infty^{(\ell)} = O(\varepsilon_\oplus^2)$. This is formalized in the following proposition.

Proposition 1. *The bit error probability after ℓ decoding iterations, $P_n^{(\ell)}(p, \varepsilon_\oplus, \varepsilon_{MAJ})$, in the asymptotic case*

$$\eta = \lim_{\ell \rightarrow \infty} \lim_{n \rightarrow \infty} P_n^{(\ell)}(p, p_\oplus, p_{MAJ}) > 0$$

for any LDPC ensemble decoded using noisy Gallager B decoder, for every channel noise level p and every decoder noise level $\varepsilon_\oplus > 0$ and/or $\varepsilon_{MAJ} > 0$.

The above claim is numerically expressed in Fig. 1, for several different levels of logic gate unreliability. In Section IV we show that the above proposition does not necessarily hold for the case of timing errors. Indeed, we prove that decoders partially built from unreliable gates can achieve zero error probability.

Fig. 1: The η -reliability levels for noisy Gallager B decoder applied on (3,5)-regular code ensemble.

III. NOISY GALLAGER B DECODER

We start our discussion with the simple, but important observation that timing error occurrence during the first decoding iteration depend on the update functions values prior to the decoding. In general, we do not have any control over these inherited values, which can dramatically degrade decoder performance. Let $\nu_{v \rightarrow c}^{(-1)}$ and $\mu_{c \rightarrow v}^{(-1)}$ 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. Then, we can formalize their influence in the following theorem.

Theorem 1. [33] *The frame error rate of the Gallager B decoder in the presence of timing errors is independent of the transmitted codeword $\mathbf{x}$ iff $\nu_{v \rightarrow c}^{(-1)} = x_v A_v$ and $\mu_{c \rightarrow v}^{(-1)} = x_v B_v$, $\forall v \in V$ and $\forall c \in C$, where $A_v, B_v \in \{\pm 1\}$.*

The above theorem reveals that decoder performance depend on the transmitted codeword order, since it is not possible to adjust coefficients A_v and B_v based on the transmitted codeword (which is unknown prior to the decoding). Incorrect switches from the first iteration accumulate on channel induce errors and in a sense increase the initial error rate. In [33] we have shown that the most harmful codeword ordering can degrade frame error rate (FER) for several orders of magnitude. Logic failures during subsequent iterations are rare and can be compensated during iterative decoding process. In addition, ensuring error-free computation during the first operation brings another benefit, depicted in the following corollary.

Corollary 1. *The frame error rate of the Gallager B decoder with failure-free first decoding iteration is independent of the transmitted codeword.*

In [33] we have shown that the decoder satisfying the above corollary can perform as good as its counterpart built from reliable components. A failure-free decoding iteration can be accomplished by increasing the system clock period and letting signal level stabilize. Furthermore, Dupraz *et al.* [34] recently provided rigorous proof that if the number iterations and code length go to infinity the error rate of the noisy decoder converges to the error rate of fully reliable decoder. However, asymptotic analysis masks an important property which dominantly influence the performance of finite-length codes – *trapping sets* [35]. We next formally define trapping sets.

Definition 1. *Erroneous variable node v is eventually correct if for some $L > 0$, for every $\ell \geq L$, holds $\hat{x}_v^{(\ell)} = x_v$, where $\hat{x}_v^{(\ell)}$ represents decoded value of x_v after the ℓ -th decoding iteration.*

Definition 2. *A set variable nodes $\mathbf{T}$ which are not eventually correct is called a trapping set, denoted by (a, b) , where $|\mathbf{T}| = a$ and b is the number of parity nodes connected to odd number of nodes in $\mathbf{T}$.*

As it was shown by Vasic *et al.* [36], revealing the the most dominant trapping sets (trapping sets with lowest cardinalities) for a chosen decoder can lead to construction of codes with lower error-floors. Dzung *et al.* [37] designed codes free of small trapping sets, especially adjusted to Gallager B decoding, called Latin-Square (LS) codes. On the other hand, other classes like well known Tanner’s quasi-cyclic (QC) codes [38] exhibit poor error-floor performance, but are easier to construct since do not require elaborate information related to decoder’s behaviour.

Here we show that, instead of designing a specific code, error-floor performance can be improved by allowing decoder to work on unreliable hardware. This surprising effect we first observed in the case of von Neumann failures [20], [39] and showed that message perturbations caused by gate failures can help decoder to escape from trapping sets. Furthermore, in the infinite number of iterations all trapping sets break, and the decoding process always converge to a valid codeword. In the limiting case simple noisy hard-decision decoder becomes maximum likelihood (ML) decoder.

However, practicality of the above approach mainly depends on the ability of circuit designer to construct logic gates which produce uncorrelated failures with specific narrow range error probabilities. On the other hand, timing errors inherently exist in synchronous VLSI circuits that operate in a subthreshold regime. Instead of introducing perturbations in every logic gate in the decoder, the timing error injection assures that only messages that frequently switch their values become candidates for hardware induced perturbation. Frequent switches of Tanner graph messages are present in large number of trapping sets. We start with the example of trapping set presented in Fig. 2.

Consider a subgraph given in Fig. 2.(a) induced by a set of variable nodes $V' = \{v_1, v_2, \dots, v_5\}$. It is well known (5,3) trapping set connected to check nodes from a set $C' = \{c_1, c_2, \dots, c_8\}$. Note that white circles correspond to correct variables, while black circles correspond to erro-

Fig. 2: Subgraph corresponding to (5,3) trapping set: (a) initially erroneous bits, (b) decoding with perfectly reliable decoder, (c) timing error appearance, (d) decoding with noisy decoder.

neous variables. Similarly, white squares denote satisfied parity checks, while unsatisfied parity checks are illustrated by black squares. Suppose that variables v_2 , v_4 and v_5 are erroneously received from the channel. After the first iteration decoder built from reliable hardware will correct all erroneous variables, meanwhile corrupting variables v_1 and v_3 , as illustrated in Fig. 2.(b). It can be shown that the second iteration annuls the effect of the previous one, and the decoding process returns to the beginning. In the subsequent iterations the number of erroneous variables will alternately change from three to two and via versa, never reducing to zero.

Suppose now that decoding is performed by noisy decoder and that during the third decoding iteration message sent by variable node v_4 to c_5 is perturb due to its inability to meet timing constrains (Fig. 2.(c)). The perturbation prevents corruption on the variable node v_3 , as illustrated in Fig. 2.(d), and in only one iteration all errors will be corrected.

The full benefits produced by timing errors can be noticed by examining average FER level as a function of logic gate unreliability, which is shown in Fig. 3. As illustrative example we choose QC code with $|V| = 155$, $|C| = 64$, code rate $R = 0.4$ denoted by QC(155,64) and measure FER by Monte Carlo simulation, for the channel rate $p = 0.1$. Performance of QC code are compared with a LS code with the same construction parameters (number of variable/check nodes, code rate, minimum distance), but which is free of small trapping sets – denoted by LS(155,64). Although LS code outperforms QC when perfectly reliable hardware is used, timing errors compensate all structural deficiencies of QC code, improving its performance for an order of magnitude. Thus, QC code achieves slightly lower error rate then its LS counterpart. On the other hand, as it was shown in [33], LS code is

Fig. 3: Influence of timing errors to QC(155,64) code ($p = 0.01$).

highly resistant to timing errors, but they do not improve its performance. Note also that QC code retains low error rate even for high failure rates. This can be attributed to the low number of signal switches that can cause timing errors. Furthermore, switches appear with high probability in messages that are involved in trapping sets, while majority of (correct) messages outside of a trapping set are stable and according to our model will not perturb their values. Recall that timing errors are consequence of reduction of power supply voltage, i.e., voltage scaling, which is not easy to force to produce certain fixed error probability. However, due to nature of decoding process, timing error is a rare event, which simplifies the voltage scaling mechanism.

Another important decoding parameter is convergence time, i.e., the average number of iterations after which decoder produces a valid codeword or get stuck in a trapping set. In both cases prolonging the decoding additionally does not change the decoder's performance. The Gallager B decoder, made of reliable gates, is known to be a fast decoder and it is able to make the most of its error-correction potential after only 10 to 20 decoding iterations. As shown in Fig. 4, timing errors do not dramatically increase the convergence time, and noisy Gallager B decoder remains desirable solution for high-speed applications.

IV. NOISY BIT-FLIPPING DECODER

We here examine important characteristic of noisy decoders – their ability to guarantee correction of a certain number of “worst case” channel errors. As discussed in Section II-C, under the von Neumann failure model decoder cannot provide any guarantees. Even when computation is performed by fully reliable hardware, provable error-correction is restricted to only small number of decoders. Sipser and Spielman, in a now classical paper [40], linked the error correction capability of

Fig. 4: Influence of timing errors to decoding convergence time ($p = 0.01$).

the bit-flipping decoder with expansion property of Tanner graph. In the subsequent work Burshtein and Miller [41] extended their work to the Gallager B decoder, while Feldman *et al.* [42] showed that linear programming can also correct a fixed fraction of worst channel errors, if an underlying graph is a good expander. Here we, by the example of noisy bit-flipping decoder, show that decoders built partially from unreliable gates can achieve arbitrary small error probability. We start with formal definition of expansion property.

Definition 3. [40] A Tanner graph G of a (γ, ρ) -regular LDPC code is a $(\gamma, \rho, \alpha, \delta)$ expander if for every subset S of at most αn variable nodes, at least $\delta|S|$ check nodes are incident to S .

Expander property allows code rate and minimum code distances to stay constant as code length increases. In the prominent paper [43], Barg and Zemor proved that expander codes achieve Shannon's capacity on binary symmetric channels. Here we show that the number of errors that can be corrected by noisy bit-flipping decoder increases linearly with code length, if MAJ gate operate reliably and if during the first iteration number of XOR gate failures does not exceed C_{XOR} .

Theorem 2. [26] Consider a $(\gamma, \rho, \alpha, (7/8 + \epsilon)\gamma)$ expander, $1/8 \geq \epsilon > 0$. The bit-flipping decoder built from unreliable check nodes can correct any pattern of

$$|V_1| < 3(3 + 8\epsilon)\alpha n/32 - \sqrt{2}C_{XOR} \quad (2)$$

errors.

Note that in addition to the above constraints, we also require that the expansion of Tanner graph is greater than $7/8\gamma$. In order to overcome worst case error patterns combined with hardware failures, code structure needs to be expanded more –

in [40] it was shown that perfect bit-flipping decoder requires expansion of only $3/4\gamma$. Now it is not hard to extrapolate the above result into probabilistic channels and prove that noisy decoders can achieve arbitrary small error probability. The following theorem depicts fundamentally different decoder's behaviour compared to the result given by Proposition 1.

Theorem 3. Consider binary symmetric channel with crossover probability $p < 3(3 + 8\epsilon)\alpha/32$. Then, if exists expander code $(\gamma, \rho, \alpha, (7/8 + \epsilon)\gamma)$ of length n , $1/8 \geq \epsilon > 0$, the probability of error after decoding, $P_n^{(\infty)}(p)$, satisfies

$$\lim_{n \rightarrow \infty} P_n^{(\infty)}(p) = 0. \quad (3)$$

As shown above expander codes behave good asymptotically, however, constructing practical codes which are good expanders is not an easy task. It is known that random bipartite graphs are good expanders with high probability, but are infeasible for practical applications. Although Capalbo *et al.* [44] provided explicit construction of codes with arbitrary expansion (called lossless expanders), high expansion comes with expense in decoder's complexity. Nevertheless, their method proves the existence of codes which are able to achieve zero error probability when decoded by noisy decoders.

Examining the the number of variable nodes with certain expansion property in structural LDPC codes is know to be an NP-complete problem. However, it is possible to link some structural parameters of a code with minimal graph expansion, as shown by Chilappagari *et al.* [45]. Namely, they proved that error correction of a code constructed under $\gamma \geq 4$ -left-regular graph increases exponentially with graph's girth. We were able to extend their results for the case of noisy decoders, as shown in the following theorem.

Theorem 4. [26] Consider an LDPC code with γ -left-regular Tanner graph with $\gamma \geq 8$ and girth $g = 2g_0$. Then, the bit-flipping decoder built from unreliable check nodes can correct any error pattern $|V_1|$ such that $|V_1| < 9n_0(\gamma/4, g_0)/32 - \sqrt{2}C_{XOR}$, where

$$n_0(\gamma/4, g_0) = n_0(\gamma/4, 2j+1) = 1 + \frac{\gamma}{4} \sum_{i=0}^{j-1} \left(\frac{\gamma}{4}\right)^i, \quad g_0 \text{ odd},$$

$$n_0(\gamma/4, g_0) = n_0(\gamma/4, 2j) = 2 \sum_{i=0}^{j-1} \left(\frac{\gamma}{4}\right)^i, \quad g_0 \text{ even}. \quad (4)$$

Note that we were able to provide lower error correction bounds for only codes with $\gamma \geq 8$. The corresponding numerical values are given in the table below.

TABLE I: Lower bounds $N_e = \lfloor 9n_0(\gamma/4, g_0)/32 - \sqrt{2}C_{XOR} \rfloor$ on the guaranteed error correction [25].

γ	8	8	8	8	8	8	12	12	12	12	12	12
g	8	10	12	8	10	12	8	10	12	8	10	12
C_{XOR}	0	0	0	2	2	2	0	0	0	2	2	2
N_e	1	1	3	0	0	1	2	3	7	0	0	4

As can be observed, the error-correction capability of the noisy bit-flipping decoder is modest, but its low complexity

Fig. 5: The memory architecture with bit-flipping decoder.

makes it a desirable solution in coded memories, which is described in Section V. Again we stress the importance of ensuring that the first decoding iteration is error-free ($C_{XOR} = 0$), since failures from the first iteration linearly reduce the correction capability of the decoder.

V. APPLICATION TO MEMORIES

Memories usually occupy large portion of VLSI devices and are especially vulnerable to manufacturing process variations, voltage drops, or thermal radiation. Thus, different modern volatile/non-volatile memories store information in coded forms, and are equipped with error-correction circuits. Theoretical fundamentals of coded memories are given by Taylor [4] and Kuznevsov [5] in late sixties and early seventies, respectively. Taylor was the first who considered memories enhanced by LDPC codes, and provided framework for design and analysis of memories built entirely from unreliable components. The memory stability concept plays central role in the Taylor's framework.

Let us assume that at time $t = 0$ a codeword on an LDPC code is written in unreliable memory. The memory cell contents is updated at equidistant time intervals $\tau, 2\tau, 3\tau, \dots$ by one iteration of chosen hard-decision decoder, made also from unreliable hardware. Note that if the information is extracted even in the time instant just after an update cycle, it is still erroneous with the probability lower bounded by the unreliability of the last logic gate used in the decoder. For that reason another hard-decision decoder, built from reliable logic gates, is added into memory architecture and used only when information is read from the memory. Thus, we declare a *memory failure* only if the read information cannot be correctly decoded by the addition decoder in finite number of iterations. We next defined the memory stability.

Definition 4. Memory complexity is defined as the sum of all memory cells and 2-input Boolean functions used in the memory architecture.

Definition 5. The memory which stores K information bits is stable if the following is satisfied:

- i) Memory complexity is limited by θK , where θ is fixed.

Fig. 6: The performance of bit-flipping memory based on $PG(2, 2^3)$ code.

- ii) For every time instance $t > 0$, $i \delta > 0$, memory failure probability at instant t satisfies $P_f(t) < \delta$.

It should be noted that only memories coded by LDPC codes satisfy stability condition. Here we examine the memory architecture that employs noisy bit-flipping decoder, as shown in Fig. 5. Let us assume that the fraction of memory cell failures between two update cycles is bounded by α_m . Similarly, let the fraction of MAJ gates that fails during a update cycle be bounded by α_γ . Then, we can formulate the following theorem.

Theorem 5. [27] *The proposed memory architecture based on a $(\gamma, \rho, \alpha, (7/8 + \epsilon)\gamma)$ expander code can preserve all stored bits for an arbitrary long time period if*

$$\alpha_m + \alpha_\gamma < 2\epsilon(3 + 8\epsilon)\alpha. \quad (5)$$

It is not difficult to show that in the asymptotic case the proposed memory satisfies stability conditions. We now illustrate the behaviour of finite-length memories assuming that failures of memory cells are uncorrelated (von Neumann failures) and happen with probability p_m , while logic gate failures follow timing error mechanism (Fig. 6). For the illustrative example, we assume that memory information is coded by the projective-geometry code, denoted by $PG(2, 2^3)$.

Since code-length is finite, after sufficiently long time memory content will be permanently corrupted. The strong impact that LDPC coding has on the memory reliability is self-explanatory. We only draw the reader's attention to usefulness of timing errors to preserving memory content even longer.

VI. CONCLUSION

The paper contains a survey on decoding of LDPC codes when logic gates used in the decoder are prone to timing errors. We mainly focused on hard-decision decoders, like Gallager B and bit-flipping decoders, which simplicity enables deeper

insight into timing errors influence on the decoding process. We established the conditions necessary for decoding performance to be independent on the transmitted codeword, and showed that adding timing errors deliberately can help decoder to correct some otherwise uncorrectable errors patterns. Timing errors appear as a consequence of supply voltage reduction, and they do not require any additional hardware. Furthermore, as power supply reduces the logic gates become more energy-efficient. The same effect is noticed in the memories which employ noisy bit-flipping decoder.

On the other hand, our analysis of the noisy bit-flipping decoder was mostly aimed to prove existence of noisy decoders that can achieve arbitrary low error probability. We show that simple bit-flipping decoder can correct fixed fraction of errors even if it is partially built from unreliable hardware.

ACKNOWLEDGEMENT

This work was supported by the Serbian Ministry of Science under project TR32028, and in part by the NSF under Grant ECCS-1500170, and the Indo-US Science and Technology Forum (IUSSTF) through the Joint Networked Center for Data Storage Research (JC-16-2014-US). The results presented in this paper are in parts published in IEEE Transactions on Information Theory, IEEE Communication Letters and presented at IEEE International Symposium on Information Theory 2014 and 2016 and Information Theory Workshop 2015.

REFERENCES

- [1] 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.
- [2] S. Winograd and J. D. Cowan, *Reliable Computation in the Presence of Noise*. Cambridge, MA, USA: MIT Press, 1963.
- [3] R. L. Dobrushin and S. I. Ortyukov, "Lower bound for the redundancy of self-correcting arrangements of unreliable functional elements," *Problems of Information Transmission*, vol. 13, no. 2, pp. 59–65, 1977.
- [4] M. Taylor, "Reliable information storage in memories designed from unreliable components," *Bell System Technical Journal*, vol. 47, pp. 2299–2337, 1968.
- [5] A. Kuznetsov, "Information storage in a memory assembled from unreliable components," *Problems of Information Transmission*, vol. 9, pp. 254–264, 1973.
- [6] 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. 1718–1751, Oct. 2010.
- [7] S. Zaynoun, M. S. Khairy, A. M. Eltawil, F. J. Kurdahi, and A. Khajeh, "Fast error aware model for arithmetic and logic circuits," in *Proceedings of 30th IEEE International Conference on Computer Design (ICCD)*, Montreal, QC, Sept.–Oct. 2012, pp. 322–328.
- [8] 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.
- [9] M. Ivkovic, S. K. Chilappagari, and B. Vasic, "Construction of memory circuits using unreliable components based on low-density parity-check codes," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM 06)*, San Francisco, CA, USA, Nov. 2006, pp. 1–5.
- [10] S. K. Chilappagari and B. Vasic, "Reliable memories built from unreliable components based on expander graphs," *arXiv:0705.0044v1 [cs.IT]*, May 2007.
- [11] B. Vasic, S. 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, Feb. 2006, paper 160, [Online Available: <http://ita.ucsd.edu/workshop/06/papers/160.pdf>].

- [12] S. Chilappagari and B. Vasic, "Fault tolerant memories based on expander graphs," in *Proceedings of IEEE Information Theory Workshop*, Tahoe City, CA, USA, 2-7 Sep. 2007, pp. 126-131.
- [13] S. Chilappagari, M. Ivkovic, and 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.
- [14] B. Vasic, S. Brkic, and P. Ivanis, "Low complexity memory architectures based on LDPC codes: Benefits and disadvantages," in *Proceedings of 12th Int. Conf. on Telecommun. in Modern Satellite, Cable and Broadcasting Services (TELSIKS 2015)*, Nis, Serbia, 14-17 Oct. 2015, pp. 1-8.
- [15] L. Varshney, "Performance of LDPC codes under faulty iterative decoding," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4427-4444, July 2011.
- [16] S. M. S. Tabatabaei Yazdi, H. Cho, and L. Dolecek, "Gallager B decoder on noisy hardware," *IEEE Transactions on Communications*, vol. 61, no. 5, pp. 1660-1673, May 2013.
- [17] C. Ngassa, V. Savin, E. Dupraz, and D. Declercq, "Density evolution and functional threshold for the noisy min-sum decoder," *IEEE Trans. Commun.*, vol. 63, no. 5, pp. 1497-1509, May 2015.
- [18] S. M. S. Tabatabaei Yazdi, C. H. Huang, and L. Dolecek, "Optimal design of a Gallager B noisy decoder for irregular LDPC codes," *IEEE Communications Letters*, vol. 16, no. 12, pp. 2052-2055, Dec. 2012.
- [19] E. Dupraz, D. Declercq, B. Vasic, and V. Savin, "Finite alphabet iterative decoders robust to faulty hardware: Analysis and selection," in *8th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, Bremen, Germany, Aug. 2014, pp. 1-10.
- [20] P. Ivanis and B. Vasic, "Error error eicitur: A stochastic resonance paradigm for reliable storage of information on unreliable media," *IEEE Trans. Commun.*, vol. 64, pp. 3596-3608, Sep. 2016.
- [21] B. Vasic, P. Ivanis, D. Declercq, and K. LeTrung, "Approaching maximum likelihood performance of LDPC codes by stochastic resonance in noisy iterative decoders," in *Proceedings of Information Theory and Applications Workshop*, San Diego, CA, February 2016.
- [22] B. Vasic, P. Ivanis, and S. Brkic, "Multi-bit flipping algorithms with probabilistic gradient descent," in *Proceedings of Information Theory and Applications Workshop*, San Diego, CA, February 2017.
- [23] S. Brkic, P. Ivanis, and B. Vasic, "Analysis of one-step majority logic decoding under correlated data-dependent gate failures," in *Proceedings of IEEE International Symposium on Information Theory (ISIT 2014)*, Honolulu, USA, June-July 2014, pp. 2599-2603.
- [24] S. Brkic, O. Al-Rasheed, P. Ivanis, and B. Vasic, "On fault tolerance of the Gallager B decoder under data-dependent gate failures," *IEEE Commun. Lett.*, vol. 19, no. 8, pp. 1299-1302, Aug. 2015.
- [25] S. Brkic, P. Ivanis, and B. Vasic, "Guaranteed error correction of faulty bit-flipping decoders under data-dependent gate failures," in *Proc. of IEEE International Symposium on Information Theory (ISIT 2016)*, Barcelona, Spain, 10-15 July 2016.
- [26] —, "Majority logic decoding under data-dependent logic gate failures," *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6295-6306, Oct. 2017.
- [27] S. Brkic, P. Ivanis, and V. B., "Reliability of memories built from unreliable components under data-dependent gate failures," *IEEE Communications Letters*, vol. 19, no. 12, pp. 1089-1092, Dec. 2015.
- [28] S. K. Planjery, D. Declercq, L. Danjean, and B. Vasić, "Finite alphabet iterative decoders, Part I: Decoding beyond belief propagation on the binary symmetric channel," *IEEE Trans. Commun.*, vol. 61, no. 10, pp. 4033-4045, Nov. 2013.
- [29] 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.
- [30] J. Chen, C. Spagnol, S. Grandhi, E. Popovici, S. Cotofana, and A. Amaricai, "Linear compositional delay model for the timing analysis of sub-powered combinational circuits," in *Proc. IEEE Comp. Soc. Annu. Symp. VLSI*, July 2014.
- [31] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599-618, Feb. 2001.
- [32] A. Balatsoukas-Stimming and A. Burg, "Density evolution for min-sum decoding of LDPC codes under unreliable message storage," *IEEE Commun. Lett.*, vol. 18, no. 5, pp. 849-852, May 2014.
- [33] S. Brkic, O.-A. Rasheed, P. Ivanis, and V. B., "On fault-tolerance of the gallager b decoder under data-dependent gate failures," *IEEE Communications Letters*, vol. 19, no. 8, p. 12991302, Avg. 2015.
- [34] E. Dupraz, D. Declercq, and B. Vasic, "Asymptotic error probability of the Gallager B decoder under timing errors," *IEEE Communications Letters*, vol. 21, no. 4, pp. 1089-1092, Apr. 2017.
- [35] T. Richardson, "Error floors of LDPC codes," in *Proc. 41st Allerton Conf. on Commun., Control, and Computing*, Monticello, USA, Oct. 2003, pp. 1426-1435.
- [36] B. Vasić, S. Chilappagari, D. Nguyen, and S. Planjery, "Trapping set ontology," in *Proc. 47th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, USA, Sep. 30-Oct. 2 2009, pp. 1-7.
- [37] D. V. Nguyen, S. K. Chilappagari, B. Vasić, and M. W. Marcellin, "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.
- [38] R. Tanner, D. Sridhara, A. Sridharan, T. Fuja, and D. Costello, "LDPC block and convolutional codes based on circulant matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 2966-2984, Dec. 2004.
- [39] B. Vasic, P. Ivanis, S. Brkic, and R. V., "Fault-resilient decoders and memories made of unreliable components," in *Proceedings of 10th Information Theory and Applications Workshop (ITA 2015)*, San Diego, CA, Feb. 2015, paper 273, [Online Available:] http://ita.ucsd.edu/workshop/15/files/paper/paper_273.pdf.
- [40] M. Sipser and D. Spielman, "Expander codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1710-1722, Nov. 1996.
- [41] D. Burshtein and G. Miller, "Expander graph arguments for message-passing algorithms," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 782-790, Feb. 2001.
- [42] J. Feldman, T. Malkin, R. A. Servedio, C. Stein, and M. J. Wainwright, "LP decoding corrects a constant fraction of errors," *IEEE Trans. Inf. Theory*, vol. 53, no. 1, pp. 82-89, Jan. 2007.
- [43] A. Barg and G. Zemor, "Error exponents of expander codes," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1725-1729, Jun. 2002.
- [44] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson, "Randomness conductors and constant-degree lossless expanders," in *STOC '02: Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*. New York, NY, USA: ACM Press, 2002, pp. 659-668.
- [45] S. K. Chilappagari, D. V. Nguyen, B. Vasic, and M. W. Marcellin, "On trapping sets and guaranteed error correction capability of LDPC codes and GLDPC codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1600-1611, Apr. 2010.