Performance of Taylor-Kuznetsov Memories Under Timing Errors

Elsa DUPRAZ[1], Bane VASIC[2], and David DECLERCQ[3]

[1] IMT Atlantique, Lab-STICC, Univ. Bretagne Loire, Brest, France
[2] Dept. of Electrical and Computer Eng., University of Arizona, Tucson, USA
[3] ETIS - ENSEA / Univ. Cergy-Pontoise / CNRS UMR-8051, Cergy-Pontoise, France

***Abstract*—Lowering the power supply of a circuit can induce transient errors in the memory cells and timing errors in the computation units. In this paper, we consider the Taylor-Kuznetsov (TK) memory architecture with transient errors in the memory cells and with timing errors in the correction circuit. We provide a theoretical analysis of the performance of TK memories under transient errors and timing errors. Our study is based on the analysis of the computation trees of the equivalent Gallager B decoders with and without timing errors. As a main result, we show that as the number of iterations goes to infinity, the error probability of the decoder with timing errors converges to the error probability of the decoder without timing errors. Monte Carlo simulations confirm this result even for moderate code lengths.**

I. INTRODUCTION

Over the past decade, the size of electronic chips has considerably reduced, while the integration factors have dramatically increased. Due to this huge scale change, energy consumption has become a major issue in the design of the next generations of electronic devices. Typical solutions involve decreasing the power supply of electronic chips by several orders of magnitude and/or reducing the clock period [1]. However, both lower power supply and clock period reduction make electronic components much more sensitive to noise. Due to this increased noise sensitivity, errors of different nature may appear in the storage and computation units built on such hardware. Lower power supply can induce transient errors that appear from time to time in the memory cells. Reducing the clock period makes computational units much more sensitive to timing errors that appear when the logic gate output does not switch before the clock rising edge [2].

In this context, Taylor [3] and Kuznetsov [4] were the first to propose a reliable memory architecture built from unreliable components. In the Taylor-Kuznetsov (TK) memory architecture, the information is stored as a codeword obtained from a Low Density Parity Check (LDPC) code. As the hardware introduces some errors in the stored information, the codeword is regularly extracted from the memory and passed through a correction circuit in order to correct the hardware errors. The memory architecture proposed in [3], [4] was further studied in [5], [6], and references therein. In particular, [5] showed the equivalence between the TK memory architecture and the Gallager B decoder subject to hardware errors. In this paper, we would like to characterize the performance of the TK architecture by taking into account both the transient errors that occur in the memory cells and the timing errors that appear in the correction circuit.

In this context, several recent works were devoted to the performance analysis of equivalent Gallager B decoders under simple transient and permanent data-independent models to represent hardware errors [7]–[9]. However, timing errors cannot be represented by the transient error models considered in [7]–[9] since they induce memory in the decoder. For example, in the timing errors model considered in [10], the gate output can be either the correct value, or the value from the previous iteration if the gate output did not switch before the clock rising edge. This model was initially proposed in [11] and its accuracy was verified by Monte-Carlo simulations on SPICE. In [10], [12], it was shown that Gallager B decoders with timing errors can actually achieve the same performance as error-free decoders. This result was demonstrated in [12] from a theoretical analysis of the decoders performance, and it was confirmed experimentally from Monte Carlo simulations in [10], [12].

The Gallager B decoder analyzed in [10], [12] is equivalent to a memory architecture with timing errors in the correction circuit, but does not take into account transient errors that occur in the memory cells. Consequently, in this paper, we propose a theoretical analysis of the performance of the TK memory architecture with transient errors in the memory cells and with timing errors in the correction circuit according to the model of [10].

Our analysis is based on the comparison of the computation trees of the decoders with and without timing errors, as for the analysis of the performance of LDPC decoders under serial scheduling [13]. We first provide bounds on the computation tree of the equivalent LDPC decoder under transient errors and timing errors similar to those in [12]. From these bounds, we show that as the number of iterations goes to infinity, the error probability of the decoder with transient and timing errors converges to the error probability of the decoder with transient errors only. This result shows that timing errors do not affect the performance of the TK memory architecture. We confirm through Monte Carlo simulations the accuracy of the proposed analysis and show that the asymptotic performance of the TK memory architecture under timing errors is indeed the same as the performance of the architecture without timing errors even for moderate code lengths.

The outline of the paper is as follows. Section II introduces the TK memory architecture and the considered transient and timing errors models. Section III provides the theoretical analysis of the performance of the architecture under timing errors. Section IV presents the Monte Carlo simulation results.

II. System Model

In this section, we present the memory architecture and the error model describing the memory degradation induced by the faulty hardware. The memory architecture and the error model were originally introduced in [3], [4] and latter considered in [5], [6]. We explain how, as initially proposed by [3], LDPC codes can be used to overcome the memory degradation induced by the faulty hardware. We then describe the correction circuit that is used in the memory architecture and we introduce the timing errors model we consider for the correction circuit.

A. Memory Degradation

Consider a memory with a storage capability of k bits, and consider the discrete time instants $t = 0, \dots, T$. Denote by $\mathbf{x}^{(0)}$ the binary information vector of length k initially stored in memory at time instant $t = 0$, and denote by $\mathbf{x}^{(t)}$ the binary information vector of length k that is in the memory at time instant t. Let $x_v^{(t)}$ be the v-th component of the vector $\mathbf{x}^{(t)}$. The memory degradation between two successive time instants t and $t+1$ is modeled by a Binary Symmetric Channel (BSC) of parameter α, which is denoted BSC(α). In other words, $P(x_v^{(t)} = 1 | x_v^{(t-1)} = 0) = P(x_v^{(t)} = 0 | x_v^{(t-1)} = 1) = \alpha$. The BSC gives a symmetric and memoryless error model. Although such a model may not take into account all the errors induced by the faulty hardware in a realistic memory, we consider it here as a first step for the analysis.

Unfortunately, because of the memory degradation, the number of errors in $\mathbf{x}^{(t)}$ with respect to $\mathbf{x}^{(0)}$ increases with t. For large enough t, $\mathbf{x}^{(t)}$ will contain too many errors, and it will not be possible to recover the initial $\mathbf{x}^{(0)}$ from $\mathbf{x}^{(t)}$ anymore. In order to overcome this effect, the information vector is encoded by an LDPC code, as described in the following.

B. LDPC Codes

In order to obtain a memory capability of k bits, consider an LDPC code of dimension k defined by a parity check matrix H of size $m \times n$, with $k \leq m - n$. The Tanner graph of the code is composed of n Variable Nodes (VN) $v \in \{1, \dots, n\}$ and m Check Nodes (CN) $c \in \{1, \dots, m\}$. The degree of the VN v is denoted as d_v and the degree of the CN c is denoted as d_c. Here, the code is assumed to be regular, *i.e*, d_v does not depend on v, and d_c does not depend on c. Denote by $\mathcal{N}_v$ the set of CNs connected to the VN v, and denote by $\mathcal{N}_c$ the set of VNs connected to the CN c.

At time instant $t = 0$, d_v copies of the codeword $\mathbf{x}$ are written in the memory. Denote by $\mathbf{x}_j^{(t)}$, $j = 1, \cdots, d_v$, the j-th binary vector of length n that is contained in the memory at time instant t. Let $x_{j,v}^{(t)}$ be the v-th component of the vector $\mathbf{x}^{(t)}$. Between two successive time instants t and $t+1$, the vectors $\mathbf{x}_j^{(t)}$ stored in the memory undergo two operations. First, the vectors $\mathbf{x}_j^{(t)}$ are passed through BSC(α), which gives degraded vectors $\mathbf{y}_j^{(t)}$. Second, a correction circuit is applied to the $\mathbf{y}_j^{(t)}$ and outputs the vectors $\mathbf{x}_j^{(t+1)}$ that are written back in the memory at instant $t+1$.

C. Correction Circuit

At every time instant t, in order to initialize the correction circuit for VN v, we need to uniquely associate each of the bit copies $y_{j,v}^{(t)}$ $(j = 1 \cdots, d_v)$ to one of the CNs $c \in \mathcal{N}_c$. We hence define a one-to-one mapping $\sigma : \mathcal{N}_c \to \{1, \cdots, d_v\}$ that orders all the nodes $c \in \mathcal{N}_c$ as $j = \sigma(c)$. The ordering is fixed once for all prior to the storage, and does not depend on the considered time instant.

At time instant t, the correction circuit is initialized with messages $\nu_{c,v}^{(t)}$ from VN v to CN c as $\nu_{c,v}^{(t)} = y_{\sigma(c),v}^{(t)}$. Let $\boldsymbol{\nu}^{(t)} = \nu_{c,\mathcal{N}_c \setminus v}^{(t)}$ denote all incoming messages to the CN c except from the VN v. For each CN $c \in C$, and for all $v \in \mathcal{N}_c$, parity check equations are computed as

$$\mu_{c,v}^{(t)} = \Psi(\boldsymbol{\nu}^{(t-1)}) = \bigoplus \boldsymbol{\nu}^{(t-1)} \tag{1}$$

where $\bigoplus$ is taken componentwise and denotes the XOR sum of the incoming messages. Then, let $\mu_{c,v}^{(t)}$ be the extrinsic message from a CN c to a VN v and let $\boldsymbol{\mu}^{(t)} = \mu_{\mathcal{N}_v \setminus c,v}^{(t)}$ denote all incoming messages to the VN v except the message from the CN c. For each VN $v \in V$ and for all $c \in \mathcal{N}_v$, the value of the bit copy $x_{\sigma(c),v}^{(t+1)}$ output by the correction circuit is decided with a majority voting operation defined as

$$x_{\sigma(c),v}^{(t+1)} = \Phi(\boldsymbol{\mu}^{(t)}) = \begin{cases} 1 & \texttt{MAJ}(\boldsymbol{\mu}^{(t)}) \geq b \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The function MAJ is defined as $\texttt{MAJ}(\boldsymbol{\mu}) = \sum \boldsymbol{\mu}$ wherein $\sum$ denotes the sum of its argument's components. The value b is a parameter of the correction circuit.

The correction circuit at time instant t corresponds to one iteration of a majority logic decoder. Further, it was shown in [5] that with this definition of the correction circuit, the memory architecture is equivalent to a modified faulty Gallager B decoder. The values $\nu_{c,v}^{(t)}$ and $\mu_{c,v}^{(t)}$ correspond to extrinsic messages that are exchanged in a decoder and the memory degradation model is equivalent to introducing errors in $\nu_{c,v}^{(t)}$ according to BSC(α). Hence, in the following, we often refer to the memory architecture as the decoder and we define the timing error model of the correction circuit over the messages $\nu_{c,v}^{(t)}$ and $\mu_{c,v}^{(t)}$.

D. Timing Errors Model for the Correction Circuit

In this section, we describe the error model that was proposed in [11] and considered in [10] in order to represent timing errors in the decoder.

Denote by Φ a deterministic Boolean function with d inputs $(u_1^{(t)}, \dots, u_d^{(t)})$ and one single output $w^{(t)} = \Phi(u_1^{(t)}, \dots, u_d^{(t)})$. The timing error model we consider is depicted in Figure 1. In this model, the output of the gate subject

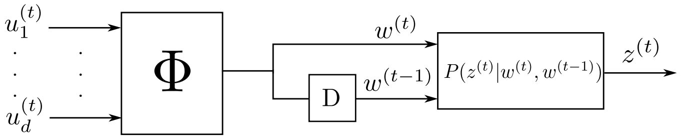

Fig. 1. Decoder timing error model [12].

to timing errors is a random variable $z^{(t)}$. The random variable $z^{(t)}$ is described by the following conditional probability distribution

$$\begin{cases} P(z^{(t)} = \ w^{(t)} \ |w^{(t)}, w^{(t-1)}) & = 1 - \varepsilon, \\ P(z^{(t)} = w^{(t-1)}|w^{(t)}, w^{(t-1)}) & = \varepsilon. \end{cases} \tag{3}$$

It defines an error model with memory since the value of $z^{(t)}$ can depend on the value of $w^{(t-1)}$ at the previous iteration. According to (3), the random variable $z^{(t)}$ can take only two values, namely $w^{(t)}$ and $w^{(t-1)}$. This model captures the probability that the gate output did not switch before the clock time.

In the correction circuit with timing errors, we assume that the initialization of the messages $\nu_{v,c}^{(t)}$ is error-free. We also assume that at the first time instant (labeled as 1), the correction circuit is error-free. This assumption was already considered in [10] so that the decoding of the current codeword does not depend on the codeword previously stored in the memory. In practice, this may be done by waiting for several clock cycles for the signal to stabilize before storing back the bit copies at the output of the correction circuit.

In the following, we denote by $\nu_{c,v}^{(t)}$ and $\mu_{c,v}^{(t)}$ the messages that are exchanged in the error-free decoder, and by $\widetilde{\nu}_{c,v}^{(t)}$ and $\widetilde{\mu}_{c,v}^{(t)}$ the messages that are exchanged in the decoder with timing errors at iteration t. In the decoder with timing errors, for $t > 1$, the deterministic CN and VN mappings (1) and (2) are followed by the timing errors model represented by (3). For simplicity, here, we assume that the parameter ε is the same for VN computation and for CN computation, which is reasonable since the two mappings are implemented on the same hardware with the same clock period. However, if needed, the theoretical analysis we present may be easily generalized to values of ε that are different for CNs and VNs.

III. Performance Analysis

This section provides a theoretical analysis of the performance of the memory architecture constructed from a correction circuit with timing errors. Denote by $P_e^{(t)} = P(x_{j,v}^{(t)} \neq x_v)$ the error probability of the memory architecture. The performance of the memory can be characterized in terms of its stability. The memory is said to be stable if the error probability $P_e^{(t)}$ converges to a fixed point that permits the successful decoding with a powerful BP or ML decoder of the codeword stored in the memory at any time instant t. In other words, the limit as t goes to infinity of the error probability $P_e^{(t)}$ must be lower than the threshold of the considered powerful decoder.

Since the memory architecture defined in Section II is equivalent to a modified Gallager B decoder, we can characterize its error probability by using the same tools as for the analysis of LDPC decoders. As for the analysis of the performance of decoders under serial scheduling [13], our analysis is based on the comparison of the computation graphs of the decoder under timing errors and of the decoder without timing errors. In the following, we will assume, as in [13], [14], that these computation graphs are cycle-free and we will refer to them as computation trees. See [15, Chapter 4] for the complete definition of a computation tree.

A. Computation Tree Analysis

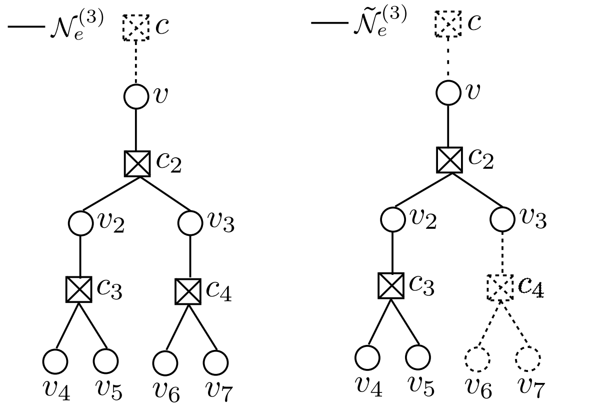

Fig. 2. For both figures, the solid lines represent the computation trees $\mathcal{N}_e^{(3)}$ and $\widetilde{\mathcal{N}}_e^{(3)}$ of the perfect decoder (left) and of the decoder with timing errors (right), respectively. For the decoder with timing errors, we assume that only one error is introduced in the computation of the message from v_3 to c_2 at time instant $t = 2$.

In the decoder, the message on the edge $e = (v, c)$ from VN v to CN c in time instant t can be expressed as a stochastic function $f_{v,c}^{(t)}$ of the initial codeword bits $x_{v'}$ (that give the all equal bit copies $x_{j,v'}^{(0)}$) corresponding to the set of all VNs v' in the computation tree. Note that $f_{v,c}^{(t)}$ is a stochastic function since its depends on the realizations of the transient errors that result from the memory degradation. Denote by $\mathcal{N}_e^{(t)}$ the computation tree of edge e at time instant t for the decoder without timing errors, and denote by $\widetilde{\mathcal{N}}_e^{(t)}$ the computation tree of the decoder with timing errors. For the decoder without timing errors, $\mathcal{N}_e^{(t)}$ by definition includes all the VNs and CNs at distance strictly less than $2t-1$ of v [14]. On the opposite, according to the model defined in (3), $\widetilde{\mathcal{N}}_e^{(t)}$ is not a complete computation tree but a random graph which depends on the timing errors that occur in the decoder.

Before we establish the relation between the size of the computation tree and probability of error, we first give examples of computation trees of the decoder without timing errors and of the decoder with timing errors.

Example. *The left part of Figure 2 represents the computation tree of the decoder without timing errors at time instant $t = 3$. The right part represents the computation tree of the decoder with timing errors at $t = 3$, assuming that one timing errors have been introduced in the computation of the messages from v_3 to c_2 at time instant $t = 2$. In the decoder without timing errors, a message exchanged from VN v to CN c at time instant t can be expressed as a function $f_{v,c}^{(\ell)}$ of a given set of codeword* *bits. In the decoder with timing errors, this function is denoted $\tilde{f}_{v,c}^{(t)}$ and applies on a possibly different set of codeword bits. The computation trees of Figure 2 are obtained from these functions as follows.*

- *At time instant $t = 1$, since no timing error is introduced, we get $\nu_{v_2,c_2}^{(1)} = \tilde{\nu}_{v_2,c_2}^{(1)} = f_{v_2,c_2}^{(1)}(y_2)$, $\nu_{v_3,c_2}^{(1)} = \tilde{\nu}_{v_3,c_2}^{(1)} = f_{v_3,c_2}^{(1)}(y_3)$. Also $\mu_{c2,v}^{(\ell)} = \tilde{\mu}_{c2,v}^{(\ell)} = f(y_2, y_3)$. Hence, $\mathcal{N}_e^{(2)} = \tilde{\mathcal{N}}_e^{(2)} = \{v, c_2, v_2, v_3\}$ and the two decoders have the same computation tree.*
- *At time instant $t = 2$, we get $\nu_{v_2,c_2}^{(2)} = \tilde{\nu}_{v_2,c_2}^{(2)} = f_{v_2,c_2}^{(2)}(y_2, y_4, y_5)$ and $\nu_{v_3,c_2}^{(2)} = f_{v_3,c_2}^{(2)}(y_3, y_6, y_7)$. Since one timing error is introduced when computing $\tilde{\nu}_{v_3,c_2}^{(2)}$, we get $\tilde{\nu}_{v_3,c_2}^{(2)} = \tilde{\nu}_{v_3,c_2}^{(1)} = \tilde{f}_{v_3,c_2}^{(2)}(y_3) = f_{v_3,c_2}^{(1)}(y_3)$ that is the message from time instant $t = 1$. As a result, $\mu_{c_2,v}^{(2)} = f_{c_2,v}^{(2)}(y_2, y_3, y_4, y_5, y_6, y_7)$, while $\tilde{\mu}_{c_2,v}^{(2)} = \tilde{f}_{c_2,v}^{(2)}(y_2, y_3, y_4, y_5)$. This gives the computation trees $\mathcal{N}_e^{(3)}$ and $\tilde{\mathcal{N}}_e^{(3)}$ in Figure 2.*

In order to perform our analysis, we need the notion of tree inclusion defined as follows. Given two subtrees $\mathcal{N}_1$ and $\mathcal{N}_2$ of the Tanner graph, $\mathcal{N}_1$ is said to be included in $\mathcal{N}_2$, and denoted $\mathcal{N}_1 \subseteq \mathcal{N}_2$, if all the (check and variable) nodes in $\mathcal{N}_1$ are also in $\mathcal{N}_2$. From the message exchange in Example, it can be noticed that $\mathcal{N}_e^{(2)} \subseteq \widetilde{\mathcal{N}}_e^{(3)} \subseteq \mathcal{N}_e^{(3)}$. This relation may be generalized to any error pattern and at any time instant, as stated in the following theorem.

Theorem 1 ([12])**.** *For any edge e and at any time instant $t > 0$,*

$$\mathcal{N}_e^{(t+1)} \subseteq \widetilde{\mathcal{N}}_e^{(3t)} \subseteq \mathcal{N}_e^{(3t)}. \tag{4}$$

Proof: The proof is the same as the proof of [12, Theorem 1]. Indeed, transient errors in the memory cells do not change the computation trees of the equivalent decoders, but only their error probabilities (see Theorem 2). ■

The above theorem shows that, at time instant $3t$, the computation tree of the decoder with timing errors is bounded by the computation trees of the decoder without timing errors at time instants $t + 1$ and $3t$. Here, the functions $f_{v,c}^{(t)}$ that represent the message exchange in the decoder are stochastic in the sense that they depend on the errors that are introduced in the memory. These errors impact the output values of the considered functions but do not change the computation trees of the decoders.

Note that the bounds (4) on computation trees cannot, in general, be rewritten into bounds on error probabilities, unless the Belief Propagation (BP) decoder is considered (see [14, Theorem 7]). However, these bounds permit to analyze the error probability of the memory under timing errors when t grows to infinity.

B. Asymptotic Error Probability of the Memory Architecture

Denote by $P_e^{(t)}$ the bit error probability of the memory architecture without timing errors at time instant t, and denote by $\widetilde{P}_e^{(t)}$ the bit error probability of the memory architecture with timing errors. The memory degradation model is output symmetric, and the deterministic VN and CN mappings followed by the timing error model (3) are also symmetric, see [16], [17]. As a result, $P_e^{(t)}$ and $\widetilde{P}_e^{(t)}$ do not depend on the stored codeword. In the analysis, we thus assume without loss of generality that the all-zero codeword was initially stored in the memory.

The expression of $P_e^{(t)}$ can be obtained with density evolution for faulty decoders, as described in [7]. The error probability $\widetilde{P}_e^{(t)}$ could be expressed with the density evolution technique proposed for decoders with memory [17], but its expression would be very difficult to derive and to evaluate. Hence, in the following, instead of deriving the expression of $\widetilde{P}_e^{(t)}$ for any t, we only give the asymptotic error probability $\widetilde{P}_e^{(+\infty)}$.

Theorem 2. *If the error probability $P_e^{(t)}$ has a limit $P_e^{(+\infty)}$ when t goes to infinity, then $\widetilde{P}_e^{(+\infty)} = P_e^{(+\infty)}$.*

Proof: The concentration theorem of [7] shows that, at a given iteration, the fraction of incorrect messages exchanged in the LDPC decoder under transient errors converges to its expected value as n goes to infinity. As a result, the error probability of a given decoder under transient errors only depends on the considered computation tree. The concentration theorem hence permits to conclude that the decoder without timing errors under $\widetilde{\mathcal{N}}_e^{(t')}$ gives the same error probability as the decoder with timing errors under $\mathcal{N}_e^{(t)}$ when $\widetilde{\mathcal{N}}_e^{(t')} = \mathcal{N}_e^{(t)}$. In addition, by letting t go to infinity in (4), we get $\lim_{t\to\infty} \widetilde{\mathcal{N}}_e^{(t)} = \lim_{t\to\infty} \mathcal{N}_e^{(t)}$, which gives $\widetilde{P}_e^{(+\infty)} = P_e^{(+\infty)}$, since the same computation tree gives the same error probability. ■

Theorem 2 shows that the performance of the memory architecture with timing errors reaches the performance of the memory architecture without timing errors when the value of t is large enough. As a result, the stability of the memory does not depend on the timing errors in the correction circuit, but only on the transient errors introduced in the memory by the faulty hardware. This confirms what was observed experimentally in [10] for the Gallager B decoder for a high number of iterations. In the simulation results we now present, we verify the accuracy of our analysis by comparing the asymptotic error probability of the memory architecture with timing errors to the asymptotic error probability of the memory architecture without timing errors.

IV. Numerical Results

In this section, we evaluate through simulations the Bit Error Rate (BER) performance of the Gallager B decoder under transient errors and timing errors according to the models of Section II. We consider a BSC of parameter α and we evaluate the Gallager B performance for two regular quasi-cyclic codes with column weight $d_v = 4$ and length $n = 1296$. The first code has rate $1/2$ and row weight $d_c = 8$, while the second code has rate $3/4$ and $d_c = 16$.

Fig. 3 represents the BER of the Gallager B decoder for the two considered codes for $\ell = 100$ iterations and for $\varepsilon = 0$, $\varepsilon = 0.2$. For both codes, the performance of the decoder with timing errors is the same as the performance of the decoder without timing errors, despite the fairly large value $\varepsilon = 0.2$.

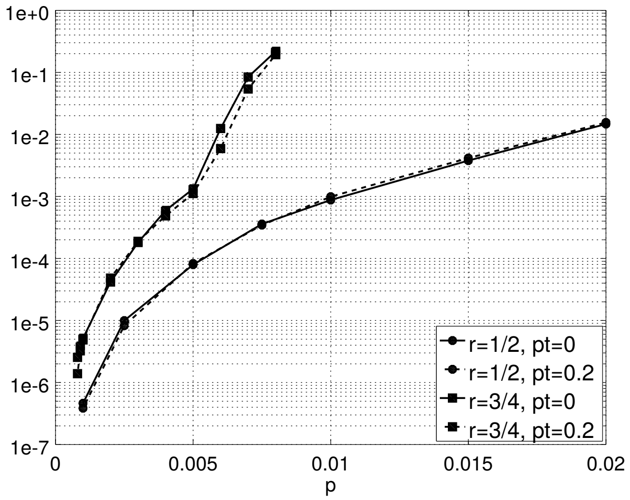

Fig. 3. BER of Gallager B decoder after 100 iterations.

Note that the value of ε is not known nor the knowledge of it used by the correction circuit.

The results of Fig. 3 are in accordance with Theorem 2 that shows that the asymptotic performance is the same with and without timing errors. The above results confirm what was experimentally observed in [10] for the Gallager B decoder under timing errors but without transient errors on the Latin Square (LS) codes $\mathrm{LS}(155, 64)$ and $\mathrm{LS}(2388, 1793)$ of [18]. The same conclusions were obtained in [12] on $(3, 5)$ codes and $(3, 6)$ codes of length $n = 1000$.

V. Conclusion

In this paper, we provided an analysis of the asymptotic performance of the TK memory architecture under transient errors in the memory cells and timing errors in the correction circuit. We showed that as t goes to infinity, the error probability of the architecture with transient and timing errors converges to the error probability of the architecture with transient errors only. As a result, the stability of the memory does not depend on the timing errors that are introduced in the correction circuit. Monte Carlo simulations confirmed this result even for moderate code lengths.

Finally, it is worth noting that the results of the theoretical analysis could be straightforwardly extended to other LDPC decoders such as BP or Min-Sum. For BP decoders, the bounds on the computation trees of Theorem 1 could even be rewritten into bounds on error probabilities for a finite number of iterations, although the considered timing error model may be less realistic for this decoder.

Acknowlegement

This work is funded in part by NSF under grants CCF-1314147 and ECCS-1500170, and supported by Indo-US Science and Technology Forum (IUSSTF) through the Joint Networked Center for Data Storage Research (JC-16-2014-US).

References

[1] V. De, "Near-threshold voltage design in nanoscale CMOS," in *Design, Automation & Test in Europe Conference & Exhibition*, 2013, pp. 612–612.

[2] A. Amaricai, V. Savin, O. Boncalo, N. Cucu-Laurenciu, J. Chen, and S. Cotofana, "Timing error analysis of flooded LDPC decoders," in *IEEE International Conference on Microwaves, Communications, Antennas and Electronic Systems*, 2015, pp. 1–5.

[3] M. G. Taylor, "Reliable information storage in memories designed from unreliable components," *Bell System Technical Journal*, vol. 47, pp. 2299–2337, Dec. 1968.

[4] A. V. Kuznetsov, "Information Storage in a Memory Assembled from Unreliable Components," *Problems of Information Transmission*, vol. 9, no. 3, pp. 254–264, 1973.

[5] B. Vasić and S. Chilappagari, "An information theoretical framework for analysis and design of nanoscale fault-tolerant memories based on low-density parity-check codes," *IEEE Transactions Circuits Systems I, Regular Papers*, vol. 54, no. 11, pp. 2438–2446, Nov. 2007.

[6] S. Brkic, P. Ivanis, and B. Vasic, "Analysis of one-step majority logic decoding under correlated data-dependent gate failures," in *IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2014, pp. 2599–2603.

[7] L. Varshney, "Performance of LDPC codes under faulty iterative decoding," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4427–4444, 2011.

[8] F. Leduc-Primeau and W. Gross, "Faulty Gallager-B decoding with optimal message repetition," in *Proc. 50th Annual Allerton Conference on Communication, Control, and Computing*, Oct. 2012, pp. 549–556.

[9] C.-H. Huang, Y. Li, and L. Dolecek, "Gallager B LDPC decoder with transient and permanent errors," *IEEE Transactions on Communications*, vol. 62, no. 1, pp. 15–28, 2014.

[10] S. Brkić, O. Al Rasheed, P. Ivaniš, and B. Vasić, "On fault tolerance of the Gallager B decoder under data-dependent gate failures," *IEEE Communications Letters*, vol. 19, no. 8, pp. 1299–1302, 2015.

[11] 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.

[12] E. Dupraz, D. Declercq, and B. Vasić, "Asymptotic error probability of the Gallager B decoder under timing errors," *IEEE Commun. Letters*, vol. PP, no. 99, pp. 1–1, Jan. 2017.

[13] E. Sharon, S. Litsyn, and J. Goldberger, "Efficient serial message-passing schedules for LDPC decoding," *IEEE Transactions on Information Theory*, vol. 53, no. 11, pp. 4076–4091, 2007.

[14] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, 2001.

[15] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Univ. Linköping, Sweden, Dept. Elec. Eng., 1996.

[16] E. Dupraz, D. Declercq, B. Vasić, and V. Savin, "Analysis and design of finite alphabet iterative decoders robust to faulty hardware," *IEEE Transactions on Communications*, vol. 63, no. 8, pp. 2797–2809, 2015.

[17] E. Janulewicz and A. Banihashemi, "Performance analysis of iterative decoding algorithms with memory over memoryless channels," *IEEE Transactions on Communications*, vol. 60, no. 12, pp. 3556–3566, 2012.

[18] D. Nguyen, S. Chilappagari, M. Marcellin, and B. Vasić, "On the Construction of Structured LDPC Codes Free of Small Trapping Sets," *IEEE Trans. Inform. Theory*, vol. 58, no. 4, pp. 2280–2302, April 2012.