Taylor-Kuznetsov fault-tolerant memories: a survey and results under correlated gate failures

Srdjan Brkic[1], Predrag Ivanis[2], Goran Djordjevic[3] and Bane Vasic[4]

Abstract – **This paper gives a brief survey of information theoretic results on fault-tolerant memory systems. The main focus is on Taylor-Kuznetsov memory architecture which has been shown to achieve nonzero computational capacity. A new approach for analyzing fault-tolerant memories that takes into account gate failure correlation is also presented. The analysis was done by modelling gate failures by Markov chain.**

Keywords – **Fault-tolerance, Fault-tolerant memory, Low-density parity-check codes, Taylor-Kuznetsov memory.**

I. INTRODUCTION

It is widely recognized that decreasing transistor size can produce many advantages in terms of computation efficiency. More the transistors placed on the chip, the larger the computational power. Also, new nano-scale technologies reduce the interconnect delays, enabling faster computation.

However, using small transistors poses numerous technological challenges. First, the energy dissipation increases which can be critical issue in mobile devices. For reducing the energy leakage the dynamic voltage and frequency scaling (DVFS) is widely used [1]. The effectiveness of DVFS is studied in [2] where the aggressive voltage scaling is proposed. Increased noise sensitivity is a second drawback as the reliability of a digital circuit decreases with transistors size. Thus, fault tolerance is recognized as one of the top challenges in semiconductor technology [3].

Physical reasons for semiconductor devices failures vary with used technology but can be broadly divided into three main categories: permanent, intermittent and transient [4]. Permanent faults are irreversible faults caused by manufacturing defects, device wear-out, or heavy ion radiation [5]. Intermittent faults occur because of unstable hardware. Often intermittent faults precede the occurrence of permanent faults. Transient faults (TFs), also referred to as soft errors, are mainly due to single or multiple event upsets or timing errors [6]. These errors have probabilistic behaviour and can be described statistically. TFs were of great concern in memories rather than logic circuits due to so called masking factors, but using sub-threshold voltages it is possible that digital circuit will be affected by multiple errors that may exhibit correlation [7]. Timing errors occur when a system operates at a very high data rate, and are caused by timing jitter. Because of the sampling clock fluctuations or signal propagation delays, the output signal of a gate may be sampled or used in the next stage before it reaches a steady value, leading to an incorrect output. Such errors are dependent on gate history, i.e. data values processed by the gate in previous bit intervals.

The fault-tolerant systems, from the information theory point of view, have been investigated over past decades. First contribution to this research field was made by Von Neumann who examined performance of circuits with faulty gates [8]. One variant of the Von Neumann's fault-tolerant architecture called triple modular redundancy was widely studied [9]-[12]. Also, large number of fault-tolerant nano-structures, proposed in the literature, are based on Von Neumann's work [13]-[16].

The fault-tolerant memories were also examined by Taylor [17], [18] who proposed use of low-density parity-check (LDPC) codes as restoration organs in faulty memories. His work was continued and refined by Kuznetsov [19]. Recently, as understanding of LDPC codes increases, research in this direction continues [20]-[24].

In this paper, we focus on the design of memory circuits built from unreliable components. We advocate usage of error correcting codes, especially LDPC codes for enabling fault-tolerance and some of our initial results are presented in this paper. We adapted the approach in which the memory elements and correcting logic circuits are considered unreliable, which is different from the state-of-the-art system in which only memory elements are faulty, while the operation of error correction circuitry is assumed to be reliable.

Fault-tolerant system analysis, presented in this paper, is done assuming Markovian types of errors. This error model corresponds to timing errors, which are dominant in extremely high-speed digital circuitry. Interestingly, this type of errors has not been studied much in the fault-tolerant literature.

The rest of the paper is organized as follows. In Section II the previous work related to this topic is presented which includes the theoretical foundation of the fault-tolerance problem and pioneering work done by Taylor and Kuznetsov. In Section III we give a brief description of novel approach we are taking for better understanding and improving the fault-tolerant systems. Section IV is dedicated to error modelling, and numerical results are presented in Section V. Finally, some concluding remarks are given in Section VI.

[1]Srdjan Brkic is with Innovation Centre, School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11000 Belgrade, Serbia, E-mail: brka05@gmail.com,

[2]Predrag Ivanis is with School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11000 Belgrade, Serbia, E-mail: predrag.ivanis@etf.rs

[3]Goran Djordjevic is with Faculty of Electronic Engineering, University of Nis, Aleksandra Medvedeva 14, 18000 Nis, Serbia, E-mail: goran@elfak.ni.ac.rs

[4]Bane Vasic is with the University of Arizona, AZ 85721 USA, E-mail: vasic@ece.arizona.edu.

II. PRIOR WORK

A. Theoretical foundation – a coding theory perspective

The conventional approach to deal with faulty gates is multiplexing. The multiplexing technique originates from von Neumann's theory of computation by circuits with faulty gates [8]. Von Neumann showed that, under certain conditions, the error at the output of multiplexing scheme can be arbitrary small, for a fixed probability of a component gate failure. The system he proposed was organized in such a manner, that failure of whole system cannot be caused by the failure of a single component, or a small number of components. His fault-tolerant circuit was comprised of the so-called computational organ consisting of many replicas of faulty gates and a restoration organ that performs a majority vote, as it is shown in Fig 1. Although multiplexing has the advantage of simplicity, it leads to extremely large hardware redundancy when gate reliability is small.

The computing systems proposed by von Neumann achieve arbitrarily small error probability only in the asymptotic limit of infinite redundancy (the redundancy is defined as the ratio between the number of components of the redundant noisy systems and the number of components of the original fault-free system). The analysis of fault-tolerant hardware can be done from perspective of error correcting coding. It is obvious that Von Neumann architecture corresponds to a repetition coding technique, for which is known to achieve small probability of error at the price of decreasing the coding rate down to zero. From the results obtained by Shannon concerning the reliability of noisy communication system it is also known that error probability can be reduced to an arbitrary low level using finite redundancy [25]. In view of this fact, we can conclude that use of better error correcting codes (ECCs), than repetition coding scheme, can improve performance of fault-tolerant systems.

The first attempt to achieve reliable communication with non-zero coding rates techniques was done by Elias [26]. Although he failed to design systems with finite redundancy, he showed that finite redundancy can be achieved only if the coder and decoder complexities grow at most linearly with the code length. The described characteristic was observed at convolutional encoders and sequential decoders. Also, this property was discovered in a LDPC family of ECCs, introduced by Gallager [27], which are largely used in nowadays communication systems.

Usage of LDPC codes in fault-tolerant systems reduces the restoration organ complexity. Taylor, in [18] proved that a Boolean function computed by a system of k reliable components can also be computed by a system of $O(k\log(k))$ unreliable components. If a LDPC decoder is used for the restoration organ, the $\log(k)$ factor comes as a consequence of asymptotical number of decoding iterations [18]. In practical LDPC decoder realizations, however, only the computing components corresponding to a single iteration need to be actually implemented in hardware, which reduces the space complexity of the fault-tolerant system.

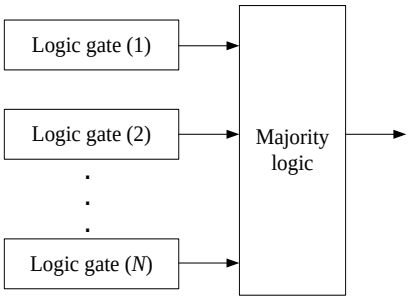

Fig. 1. Von Neumann fault-tolerant system architecture.

Moreover, a practical system may use only a finite number of decoding iterations, while preserving very high reliability.

B. Taylor-Kuznetsov approach to fault-tolerant memories

Taylor [18] was the first to investigate the capacity and fault-tolerant architectures of storage systems built entirely from unreliable components. His results were refined by Kuznetsov. Taylor and Kuznetsov tried to derive results analogous to the ones obtained by Shannon on the capacity of communication systems. Taylor proved that a memory built from unreliable components can achieve storage capacity C for all memory redundancies greater that $1/C$. In the Taylor-Kuznetsov (TK) model, the information is stored in a coded form, i.e., the stored vector is a codeword of some (n, k) block ECC. The memory in which the bits are stored is also assumed to be unreliable. It is connected to a correcting circuit (a restoration organ in von Neumann's terminology), which periodically updates the values in the memory according to some decoding scheme.

Taylor and Kuznetsov showed that the redundancy necessary to ensure the reliability of a memory grows asymptotically linearly with the memory size. Taylor also described the importance of LDPC codes arguing that no decoding scheme other than iterative LDPC decoding can achieve non-zero storage capacity. He also derived bounds on memory failure probability under assumption of uncorrelated memory failures. In the TK scheme, a memory failure is said to occur if the error pattern in the memory is uncorrectable by a noiseless decoder in predefined number of iterations.

We next describe the TK scheme in a similar fashion as it was previously done in [20], [21] and [24].

Initial assumption is that the information to be stored is first encoded by a regular binary LDPC code of length n and dimension k. Parity check matrix $\boldsymbol{H}$ of a (J,K)-regular LDPC code is of size $m\times n$ and has J 1's in every column and K 1's in every row. The stored codeword, denoted as $\boldsymbol{x}=(x_1,x_2,...,x_n)$, consist of n variable bits x_i, $(i=1,...,n)$ which are involved in exactly J parity-check equations. What parity checks are satisfied can be determined by multiplying codeword $\boldsymbol{x}$ and the transpose of a parity check matrix ($\boldsymbol{H}^{\mathrm{T}}$), i. e. $\boldsymbol{x}\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{c}$. The vector $\boldsymbol{c}=(c_1,c_2,...,c_m)$ is called a syndrome and its size is equal to m, the number of rows in parity check matrix $\boldsymbol{H}$. Each value c_j $(j=1,...,m)$ corresponds to the value of j-th parity-check sum. Parity check c_j is said to be satisfied if $c_j=0$ and unsatisfied if $c_j=1$.

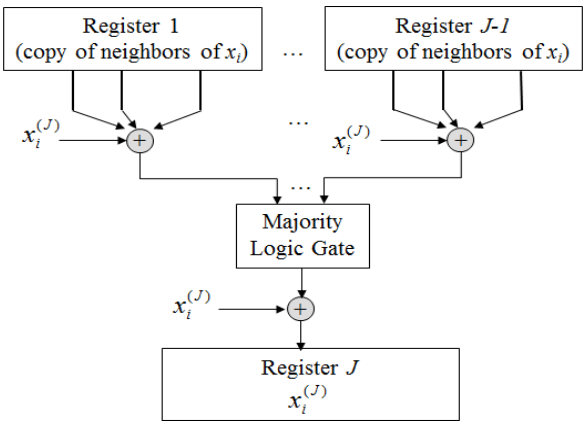

Fig. 2. A block diagram of the Taylor-Kuznetsov memory architecture.

After encoding, the J copies of every coded bit x_i $\left\{x_i^{(1)}, x_i^{(2)}, ..., x_i^{(J)}\right\}$ and stored in J registers. Fig. 2, shows the steps for updating $x_i^{(J)}$, the J-th copy of the bit x_i. All bit-copies initially have the same values. Registers are made from memory cells that are considered to be unreliable. New estimates of each of these copies are obtained by using one combination of J-1 checks. The estimates are obtained as follows.

1. Evaluate parity checks for each bit-copy (exclude one distinct parity check from the original set of check for each bit-copy).
2. Flip the value of a particular bit-copy if half or more of the parity checks are unsatisfied.
3. Iterate (1) and (2).

There are a total of J-1 parity checks for each bit copy. The decision element in this case is a majority logic gate whose output is 1 if half or more of the parity checks are nonzero. The correction is accomplished by XOR gate and the previous value of bit-copy. In this iterative decoding scheme, each parity check requires K-1 input bits (other than the bit-copy we are trying to estimate). Since each of these input bits has J different copies, there are J copies for each particular bit that can be used for estimating $x_i^{(j)}$, j-th copy of bit x_i. When estimating a copy $x_i^{(j)}$, of the bit x_i using an estimate of bit x_l, we use the bit-copy $x_l^{(s)}$ in whose evaluation the parity check involving x_i is omitted.

The equivalence between TK scheme and Gallager-B decoding algorithm (described in [27] and [28]), is now well established [20]. We next briefly present the modification of TK scheme that allows us to examine TK scheme in the context of iterative decoding of LDPC codes. The modification was originally proposed in [24].

Parity checks computation in TK scheme gives assessment regarding satisfaction of observed parity check equations. If we exclude a bit-copy that has been estimated from parity check equations, we can evaluate the bit-copy rather than the parity. All evaluated bit-copies then can be compared using majority vote. Thus, each bit-copy $x_i^{(j)}$, (j=1,...,J) of variable bit x_i can be represented by edges of the Tanner graph incident on x_i.

This modified TK scheme uses less computation operation compared to original TK scheme and enable us to design the memories that can be examined using graphs and large knowledge about iterative decoding of LDPC codes [20]. In [24] another modification was proposed which implied that after every iteration of a Gallager-B decoding algorithm syndrome was computed. If all parity checks were satisfied the correct codeword was stored in memory registers. It was shown that this modification significantly increase memory reliability.

Also, fault-tolerant memory architecture based on expander graphs of LDPC codes was presented in [21]. The correcting circuit were organized according to parallel bit flipping algorithm, described in [29]. Theoretical boundary concerning the fraction of errors that can be tolerated by this memory architecture was also derived.

III. Fault-tolerant Memory Analysis

In this section we present a novel approach for analysis and construction of reliable memories built from unreliable components.

Traditional models of memory error correcting coding systems assume perfect and deterministic operation of an error correcting decoder, and localize the errors only in the memory elements (as illustrated in Fig. 3. (a)). If the reliability of logic gates used in the coder and correcting circuit (decoder) is many orders of magnitude higher than the reliability of memory cells, the errors in the memory have dominant effect on system performance. However, if digital logic in the coder and decoder is built of faulty components, then the errors and noise do affect operations performed in coder/decoder, and reduce the reliability of the whole system (Fig. 3. (b)). We assume that devices have finite reliability, and analyze an error correction system with low-hardware overhead based on LDPC codes.

The LDPC codes were intensively investigated during the past decade, but until know, only memory architectures based on Gallager-B and bit flipping algorithms were examined.

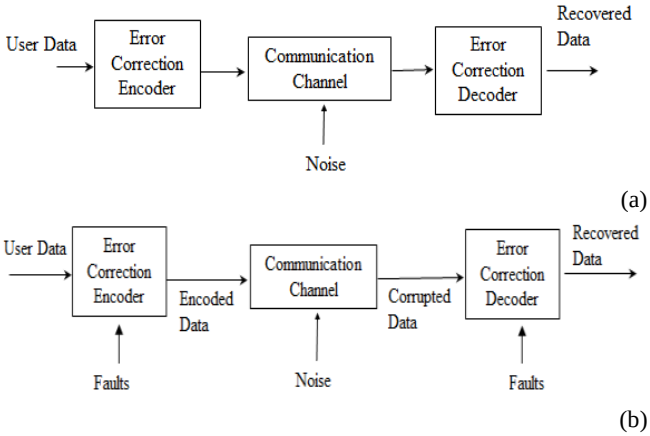

Fig. 3. Block diagram of a memory system (a) classical - the noise affects only memory, but the operation of the coder/decoder is fault-free, (b) under consideration - all components including logic gates in the coder/decoder are unreliable.

Performance achievable by e.g., Min-Sum or FAID algorithms in fault-tolerant memories, as described in [30] and [31], are not yet known.

Also, a choice of LDPC code which code words are used for storing information in the memory registers can significantly influence the memory fault-tolerance. Our investigation is aimed at analysis of quasi-cyclic LDPC codes based on circulants. This type of LDPC codes are designed to have low error floors on the binary symmetric channel, by ensuring that their Tanner graphs are free of certain small trapping sets [32].

IV. The Correlated Failure Model

Both Von Neumann and Taylor-Kuznetsov models assume that faults of memory elements and logic gates are transient, i.e. faults occur at a particular time instants but do not necessary persist for later times. It is also assumed that gates fail independently of each other and that defects are not permanent, i.e. faulty gate may produce a correct output at some point in time. Such failure mechanism is referred to as von Neumann type of errors. In this paper we consider a more general failure model in which failures of a given logic gate are data-dependent and correlated in time.

Our model can be used for analyzing physical phenomena such as logic hazard [33], or noise influence to chips with sub-threshold voltages. We formed a model of faulty gates through its first order statistics (probability of erroneous circuit output) as a function of logic gates inputs or outputs. For that purpose we used Markov chains.

As it can be seen in Fig. 2, the TK scheme is composed of (J-1)-input majority logic gates, K-input XOR gates and 2-input XOR gates. Because all multi-input gates in hardware are implemented as circuits composed of 2-input Boolean logic gates, our analysis is done at the 2-input Boolean function level. The correct binary output value of a given Boolean function, $O_c(k)$ at discrete time k, (k>0), depends on $I_1(k)$ and $I_2(k)$, the logic gate binary input values. This is illustrated in Fig. 4. The errors are inserted into logic gate by performing XOR operation between correct gate output sequence $\{O_c(k)\}_{k>0}$, and the error sequence $\{e(k)\}_{k>0}$, producing the actual output sequence $\{O_e(k)\}_{k>0}$, (Fig. 4). The error sequence $\{e(k)\}_{k>0}$, represents the binary time series which describes the statistics of errors. If the k-th value of error pattern is '1', i.e. $e(k)$=1 (k>0), the output of Boolean logic gate at time k will be faulty, i.e., the k-th actual output value will not correspond to correct one ($O_e(k) \neq O_c(k)$). The error sequence is modelled as a finite Markov chain. The error pattern statistics depends on gate inputs and outputs, and in principle there are two ways to define such dependence. In the first approach the $e(k)$ at discrete time k taking some value depends on the current gate output $O_c(k)$ as well as M, M>0, previous gate outputs $O_c(k\text{-}1)$, … $O_c(k\text{-}M)$. In the second approach, the probability of error pattern depends on gate inputs $I_1(k)$, $I_2(k)$, $I_1(k\text{-}1)$, $I_2(k\text{-}1)$... $I_1(k\text{-}M)$, $I_2(k\text{-}M)$). In this paper, our focus was on output dependence model because its complexity is half of that of the second model, while still capturing the essential characteristics of data dependence.

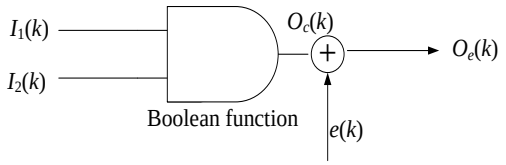

Fig. 4. Error insertion into a 2-input Boolean function.

Let S be a Markov source generating the error sequence composed of 2^{M+1} states s_i, $1 \leq i \leq 2^{M+1}$, i.e. $S = \{s_1, s_2, ..., s_{2^{M+1}}\}$. Every state corresponds to one possible logic gate output sequence of length M+1 and captures the different data-dependent failures, expressed through different error probabilities $p(s_i)=\Pr\{e(k)=1|s_i\}$, $1 \leq i \leq 2^{M+1}$, k>0, where Pr{.} denotes probability. According to current binary output value $O_c(k)$, k>0, from each state s_i only transition to two next states s_j and s_m are possible, which happens with probabilities p_{ij} and p_{im}, respectively. The transition probabilities p_{ij}, depend on statistic of sequence $\{O_c(k)\}_{k>0}$, and must satisfy $p_{ij}+p_{im}=1$, $1 \leq a \leq 2^{M+1}$, $a \epsilon \{i,j,m\}$. Although large value of M can give us possibility to describe data-dependence in more details, complexity of model increases exponentially. Also, long correlation is not expected in faulty logic gates. Thus, we use the simplest model with M=1, presented in Fig. 5, which can adequately illustrate data-dependent failures. We next analyze AND, OR and XOR faulty logic gates.

At the output of AND logic gate correct value '1' appears only if both inputs are equal to '1'. Thus, if due to increased noise level one input changes its value, the gate output will be faulty. We can conclude that gate output '1' is more prone to errors. Consequently, the error probability in the state s_4='11' is the highest and in the s_1='00' state is the lowest. The last conclusion can be formulated as follows

$$p_{AND}(s_i) = \Pr\{e=1 \mid s_i\} = A_i \Pr\{e=1 \mid s_4\}, \quad 1 \leq i \leq 4, \quad (1)$$

where $p_{AND}(s_i)$ denotes the error probability at the output of AND logic gate in state s_i, $1 \leq i \leq 4$, and A_i represent the scaling coefficients, dependent of state s_i. From the discussion presented above, it is clear that must hold

$$A_1 \leq A_0 \leq A_4 = 1, \quad A_0 \in \{A_2, A_3\}. \quad (2)$$

Similar analysis can be performed for the OR logic gate. The correct output value '0' can be changed to incorrect value '1' even if only one input is faulty. Thus, in this case the state '00' is the most sensitive and we can write the following expression

$$p_{OR}(s_i) = \Pr\{e=1 \mid s_i\} = B_i \Pr\{e=1 \mid s_1\}, \quad 1 \leq i \leq 4, \quad (3)$$

where $p_{OR}(s_i)$ denotes the error probability at the output of OR logic gate in state s_i, $1 \leq i \leq 4$, and B_i represent the scaling coefficients, dependent of state s_i. It can be noted that following condition must be satisfied

$$B_4 \leq B_0 \leq B_1 = 1, \quad B_0 \in \{B_2, B_3\}. \quad (4)$$

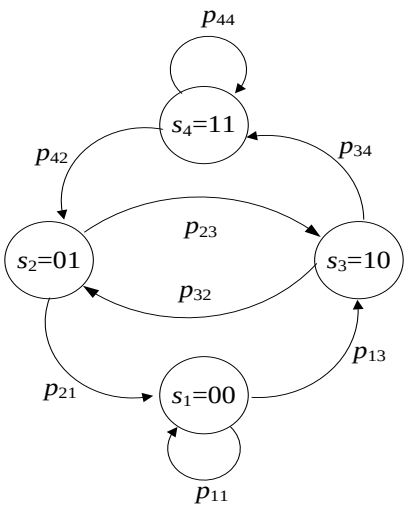

Fig. 5. Output dependence finite state error model.

Every change of input values of XOR logic gate will produce an error. Thus, probabilities of output error will be the same regardless of current state, $p_{XOR}(s_i)=p_{XOR}$, $1\le i\le 4$, i.e. XOR error pattern can be modelled as Binary Symmetric Channel (BSC).

As previously described, using AND, OR and XOR logic gates every multiple-input logic gate in TK scheme can be implemented. For example, faulty 3-input majority logic gate can be presented as a digital circuit composed of three faulty AND gates and two faulty OR gates (Fig. 6).

The results of majority logic gates analysis are graphically presented in Fig. 7 and Fig. 8, for several values of A_i and B_i coefficients and inputs statistics. The input statistics are described by probability that input values I_i (i=1,2,3) are equal to '1', denoted as P_1. The failure model coefficients are given in normalized form as follows

$$\begin{aligned} &B_1 = A_4 = 1, \\ &B_2 = B_3 = A_2 = A_3 = 1/p, \\ &B_4 = A_1 = 1/p^2. \end{aligned} \tag{5}$$

In this way all coefficients are described by a single parameter (p). It should be noted that this coefficient relations are purely theoretical and they have not been validated by real measurements.

The output error probability of 3-input majority logic gate dependence of average component failures is presented in Fig 7, for several values of parameter p(=1,2,3) and two input probabilities P_1=0.5 and P_1=0.9. It can be noticed that input statistics have the most influence on logic gate performance. A majority logic gate output is equal to '1', if half or more inputs are equal to '1'. Thus, when ones and zeros appear at the gate inputs with equal probabilities (P_1=0.5) more gate output values will be faulty, compared to case when almost all inputs are '1' (P_1=0.9). When P_1=0.5, parameter p, which describes presented Markov model, does not have any impact on logic gate performance. So, for that case the presented model is excessive and can be replaced by uncorrelated error model.

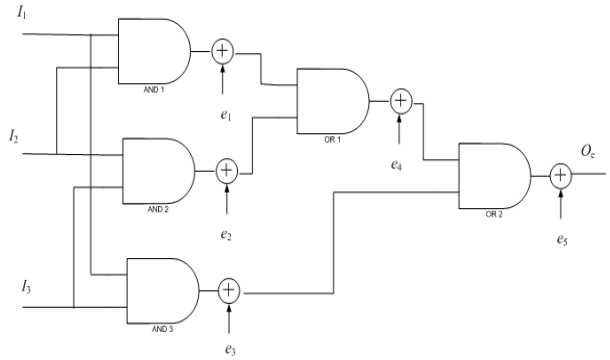

Fig. 6. Digital circuit scheme of faulty 3-input majority logic gate.

When P_1=0.9 differences caused by error correlation exist, but logic gate failures can be well approximated by uncorrelated errors.

Furthermore, as the output error probability is a linear function of average component error, a faulty majority logic gate can be modelled as correct one at which output the error pattern is inserted. Thus, if probabilities of input values are known, a simpler model can be used, which is characterized only by output error probability, as it is presented in Fig. 9.

It should be noted that simplification, presented in Fig. 9, is accurate only if autocorrelation function of error pattern is unchanged. It is well known that, using autoregressive (AR) models, random process with arbitrary autocorrelation function can be generated [34]. But, in this case, as simulation analysis has shown, presented Markov chain produced the output error pattern which greatly resembles the uncorrelated one. Thus, it is sufficient to determine first order statistics of the error pattern.

It should be also emphasized that we did not take into account asymmetry of digital circuit scheme. As can be seen in Fig. 6, the input combination (I_1,I_3) will be affected with only two error patterns (e_3 and e_5), while on paths of other two input combinations, three error sequences exist. What effect the circuit asymmetry has on faulty logic gate performance will be determined in our future work.

The performance comparison of majority logic gates with different number of inputs is presented in Fig. 8, when p=2. The number of inputs of majority logic gate used in TK scheme depends of column weight of used LDPC code parity check matrix. Thus, the comparison of different majority logic gates can give us an insight what LDPC code will produce less errors in the decoding phase. Also, code correcting capabilities have a significant influence in choosing the right code for TK scheme. It can be noted that 2-input majority logic gate (which is actually a simple OR logic gate) has the lowest output error probability when P_1=0.5. But, when P_1=0.9, the simulation has shown that the gate with largest number of inputs (4-input gate) outperforms other logic gates. The gates with more inputs are less sensitive to errors when input value '1' is more frequent then value '0'.

We can conclude that input values statistics have a key role in majority logic gate analysis. If we can determine the probabilities of input values we can easily generate a simpler model of faulty majority logic gate and choose the LDPC code, which will be more resistant to hardware failures.

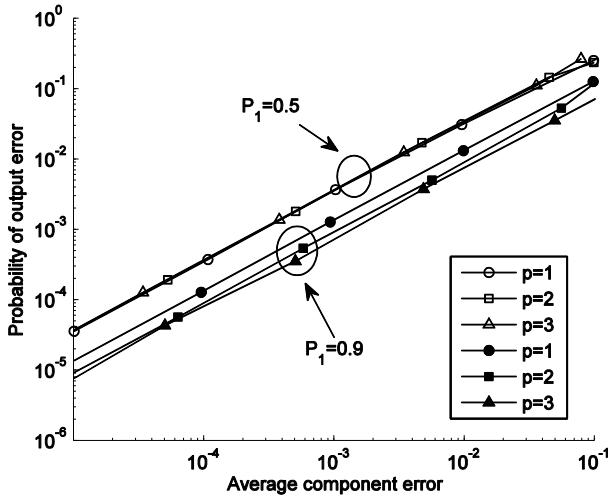

Fig. 7. Probability of error in 3-input majority logic gate.

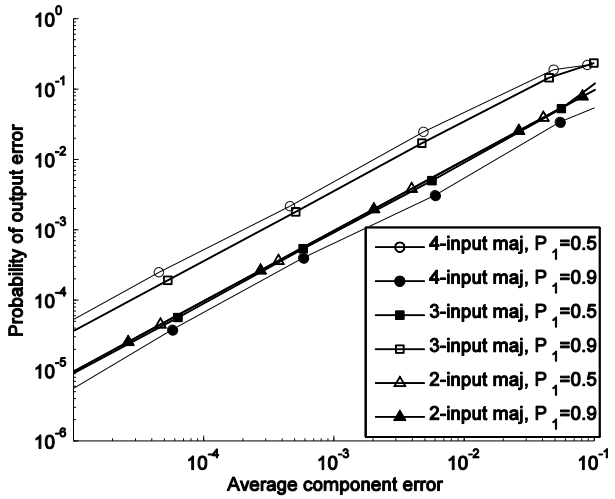

Fig. 8. Comparison of majority logic gates with different number of inputs for p=2.

However, determining the logic gate input statistics in the TK scheme is not a simple task due to strong dependence among logic gates, and it will be focus of our future work.

The faulty multi-input XOR gates are also an integral part of the TK scheme and, as already mentioned, can be represented as 2-input faulty XOR gates. In Fig. 10 the 3-input XOR gate model is presented.

The number of inputs of XOR logic gate is defined by the row weight of chosen LDPC code parity check matrix. Thus, it is interesting to compare XOR gates with different number of inputs.

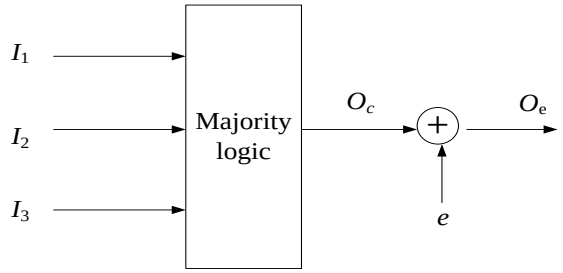

Fig. 9. Equivalent scheme of faulty 3-input XOR logic gate.

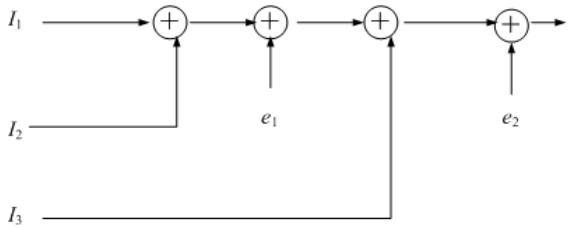

Fig. 10. A scheme of a faulty 3-input XOR logic gate.

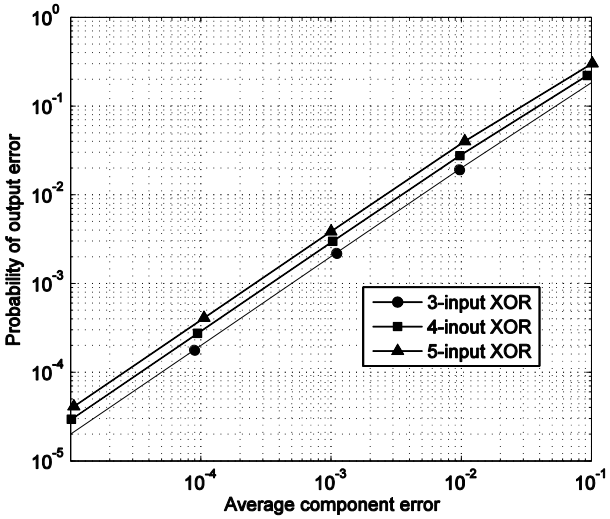

Fig. 11. Comparison of XOR logic gates with different number of inputs (output dependence model).

Performance of XOR gates with 3, 4 and 5 inputs are presented in Fig 11. It can be noted that increasing the number of inputs causes higher output error probability. In multi-input XOR gate every odd number of 2-input XOR gate failures will produce output errors. In XOR logic gates with more inputs more 2-input XOR gate failure combination can generate an output error.

Because errors in 2-input XOR gates are uncorrelated, the number of inputs is only parameter that affects multi-input XOR gate performance. The input values statistics and circuit asymmetry do not have any impact on output error probability in our XOR gate failure model.

V. Numerical Results

In this section we present the simulation results for TK memory architecture with codeword length equal to 15 bits. We used LDPC code constructed by Euclidean geometry principle, EG(15,7). The parity check matrix of this code has four 1's in every row and column. The length of the shortest cycle of the Tanner graph representation is equal to 6.

We assumed that memory failures are independent and they can happen with probability P_m. The failures of correcting circuit are modelled by using Markov model described in the Section IV. Thus, the probability that output of Boolean function is incorrect in the worst state P_b, $P_b=p_{AND}(s_4)=p_{OR}(s_1)=p_{XOR}$, and value of parameter p from Eq. 5 completely defines the failure model.

We assumed that the memory contents pass through a BSC after a time period T and then are updated by the message passing Gallager-B decoder. Initially, all-zero code word is stored in memory registers. The message passing can be run for any number of iterations. It is also assumed that the time for update is smaller compared to T and that memory contents do not change while the update is in progress.

The bit error rate (BER) curves for described memory architecture when $P_m=10^{-3}$ and $p=2$, are presented in Fig 12, for several values of P_b. The update process is terminated after four iteration of Gallager-B algorithm. It can be noted that when correcting circuits faults are of the same order of magnitude as memory elements faults, the reliable memory cannot be achieved. If logic gate error probability has lower values ($P_b=10^{-4}$, $P_b=10^{-5}$) BER does not increase rapidly and stays below memory error probability for several time steps T.

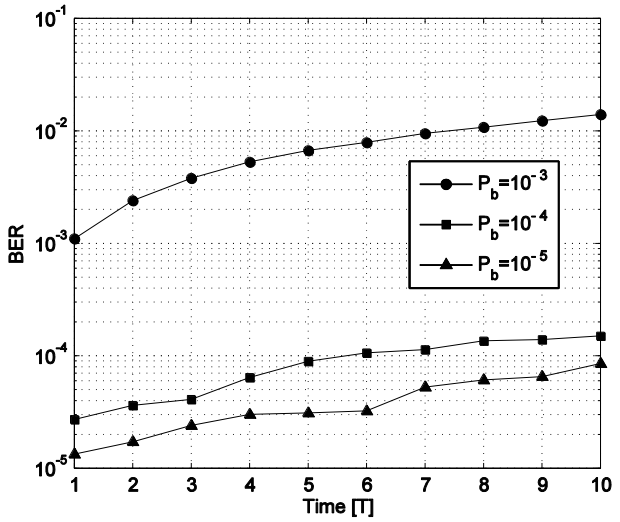

Fig. 12. Performance of TK scheme with EG(15,7) LDPC code in a correlated error model (p=2).

VI. SUMMARY

Due to nano-scale technologies development, assurance of the fault-tolerance became a critical issue. Using LDPC codes, as it is done in TK scheme, can significantly increase the memory reliability.

In this paper we have examined the transient faults influence to performance of multi-input digital gates used in TK memory architecture. The error correlation model, based on Markov chain has been introduced and performance of majority logic gates and multi-input XOR gates have been obtained.

It has been observed that input statistics have dominant affect on faulty majority logic gate performance, compared to error correlation modelled by Markov chain. Thus, simplification of faulty majority gate model has been presented. The error probability at the output of multi-input XOR logic gate has been determined by number of gate inputs.

ACKNOWLEDGEMENT

This work was supported by the Seventh Framework Programme of the European Union, under Grant Agreement number 309129 (i-RISC project). It is also funded in part by the NSF under grant CCF-0963726. Parts of this material were published in [20], [21] and [24] and in the i-RISC proposal.

REFERENCES

[1] V. Gutnik, A.P. Chandrakasan, "Embedded Power Supply for Low-Power DSP", *IEEE Transaction on Very Lagre Scale Integration (VLSI) Systems*, vol. 5, no. 4, pp. 425-435, Dec. 1997.

[2] A. Khajeh, C. Shih-Yang, A.M. Eltawil, F. J. Kurdahi, "Power Management for Cognitive Radio Platforms", In *Proceedings of IEEE Global Telecommunications Conference 2007, GLOBECOM '07*, pp. 4066-4070, Nov. 2007.

[3] Internationl Technology Roadmap for Semiconductors, http://www.itrs.net/

[4] P. Larsson, C. Svensson, "Noise in Digital CMOS Circuits", *IEEE Journal of Solid-State Circuits*, vol. 29, no. 6, pp. 655–661, June 1994.

[5] H. Naeimi, A. DeHon, "A Greedy Algorithm for Tolerating Defective Crosspoints in NanoPLA Design", In *Proceedings of IEEE International Field-Programmable Technology*, pp. 49-56, Sept. 2004.

[6] N. Miskov-Zivanov and D. Marculescu, "A Systematic Approach to Modeling and Analysis of Transient Faults in Logic Circuits", In *Proceedings of 10th International Conference on Quality Electronic Design*, pp. 408-413, March 2009.

[7] N. George, C. Elks, B. Johnson, and J. Lach, "Transient Fault Models and AVF Estimation Revisited", In *Proceedings of IEEE/IFIP International Conference on Dependable Systems & Networks* (DSN), pp. 477-486, June, 2010.

[8] J. Von Neumann, "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components", in C. E. Shannon and J. McCarthy, Eds. *Automata Studies*. Princeton: Princeton University Press, pp. 43-98, 1956.

[9] T. M. Austin, "Diva: A Reliable Substrate for Deep Submicron Microarchitecture Design", In *Proceedings of the 32nd annual ACM/ IEEE International Symposiumon on Microarchitecture*, pp. 196-207, 1999.

[10] S. Buchner, M. Baze, D. Brown, D. McMorrow, and J. Melinger, "Comparison of Error Rates in Combinatorial and Sequential Logic", *IEEE Transactions on Nuclear Science*, vol. 44, no. 6, pp. 2209-2216, Dec. 1997.

[11] P. Larsson and C. Svensson, "Noise in Digital CMOS Circuits", *IEEE Journal of Solid-State Circuits*, vol. 29, no. 6, pp. 655-661, June 1994.

[12] N. Pippenger, "Invariance of Complexity Measures for Networks with Unreliable Gates", *Journal of Association of Computing Machinery*, vol. 36, no. 3, pp. 531-539, July 1989.

[13] H. Jie; P. Jonker, P., "A System Architecture Solution for Unreliable Nanoelectronic Devices", *IEEE Transactions on Nanotechnology*, vol. 1, no. 4, pp. 201-208, Dec. 2002.

[14] S. Roy, V. Beiu, "Majority Multiplexing-Economical Redundant Fault-Tolerant Designs for Nanoarchitectures", *IEEE Transactions on Nanotechnology*, vol. 4, no. 4, pp. 441-451, July 2005.

[15] H. Goronkin and Y. Yang, "High-Performance Emerging Solid-State Memory Technologies", *MRS Bulletin, special issue*

on High-Performance Emerging Solid-State Memory Technologies, vol. 29, no. 11**,** pp. 805-813, Nov. 2004.

[16] K. Nikolic, A. Sadek, and M. Forshaw, "Architectures for Reliable Computing with Unreliable Nanodevices", In *Proceedings of IEEE-NANO*, pp. 254–259, Oct. 2001.

[17] M. Taylor, "Reliable Information Storage in Memories Designed from Unreliable Components", *Bell System Technical Journal*, 47, pp 2299-2337, 1968.

[18] M. G. Taylor, "Reliable Computation in Computing Systems Designed from Unreliable Components", *Bell System Technical Journal*, vol. 47, no. 10, pp. 2239-2366, Dec. 1968.

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

[20] 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.

[21] S. K. Chilappagari and B. Vasic, "Fault Tolerant Memories Based on Expander Graphs", In *Proceedings of IEEE Information Theory Workshop*, Tahoe City, CA, USA, pp. 126-131, Sep. 2007.

[22] S. K. Chilappagari, M. Ivkovic, and B. Vasic, "Analysis of One Step Majority Logic Decoders Constructed from Faulty Gates", In *Proceedings of IEEE Internationl Symposium on Information Theory*, Seattle, WA, USA, pp. 469-473, July 2006.

[23] S. K. Chilappagari, B. Vasic, and M. Marcellin, "Can the Storage Capacity of Memories Built from Unreliable Components Be Determined?", In *Proceedings of Information Theory and Applications Workshop*, San Diego, CA, USA, pp. 41-43, Jan. 2008.

[24] 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 2006, GLOBECOM '06*, San Fransisco, CA, USA, pp. 1-5, Nov. 2006.

[25] C. E. Shannon, "A Mathematical Theory of Communication", *Bell System Technical Journal*, vol. 27, no. 3-4, pp. 379-423, 623-656, 1948.

[26] P. Elias, "Computation in the Presence of Noise", *IBM Journal of R&D*, vol. 2, no. 4, pp. 346-353, Oct. 1958.

[27] R. G. Gallager, "Low Density Parity Check Codes", Cambridge, MA: M.I.T Press, 1963.

[28] A. Shokrollahi, "An Introduction to Low-Density Parity-Check Codes", *Theoretical Aspects of Computer Science: Advanced Lectures*, Springer-Verlag New York, Inc., New York, NY, 2002.

[29] M. Sipser, D. Spielman, "Expander Codes", *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1710-1722, Nov. 1996.

[30] J. Zhao, F. Zarkeshvari, A. H. Banihashemi, "On Implementation of Min-Sum Algorithm and Its Modifications for Decoding Low-Density Parity-Check (LDPC) codes", *IEEE Transactions on Communication*, vol. 53, no. 4, April 2005.

[31] S. Planjery, D. Declercq, L. Danjean, and B. Vasic, "Finite Alphabet Iterative Decoders for LDPC Codes Surpassing Floating-Point Iterative Decoders", *Electronics Letters*, vol. 47, no. 16, pp. 919-921, Aug. 2011.

[32] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, D. Costello, "LDPC Block and Convolutional Codes Based on Circulant Matrices", *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 2966-2984, Dec. 2004.

[33] J. F. Wakerly, *Digital Design: Principles and Practices*, 3rd edition, Prentice Hall, Englewood Cliffs, New Jersey, 1999.

[34] S. Haykin, *Adaptive Filter Theory*, 3rd edition, Prentice Hall, Englewood Cliffs, New Jersey, 1996.