Evaluation of the Robustness of LDPC Encoders to Hardware Noise

Elsa DUPRAZ and David DECLERCQ
ETIS - ENSEA / Univ. Cergy-Pontoise / CNRS UMR-8051, Cergy-Pontoise, France

***Abstract*—This paper analyzes the robustness of Low Density Parity Check (LDPC) encoders on faulty hardware. The faulty hardware effect on the encoder is represented by an error model at the XOR gate level. We review the existing LDPC encoding solutions [1], [2] and the code constructions [3]–[5] that guarantee low encoding complexity. For each of the existing solutions [1]–[5], we provide the analytic expression of the encoding error probability, and we use it to evaluate the robustness of the encoders to hardware noise. We then identify the two best encoding solutions in terms of robustness and we compare their performance with Monte-Carlo simulations.**

I. INTRODUCTION

Over the past few years, reliability has become a major issue in the design of electronic devices. A huge increase in the integration factors coupled with important chip size reduction will make the next generations of electronic devices much more sensitive to noise [6]. As a consequence, in the future systems of communication and storage, errors may not only come from the transmission channels, but also from the faulty hardware. In this context, there is a need to evaluate the robustness of Low Density Parity Check (LDPC) encoders and decoders running on faulty hardware.

The robustness of LDPC decoders was widely investigated for a large range of decoders. The performance of hard decoders under faulty hardware was analyzed in [7] (Gallager A) and [8] (Gallager B), while soft decoders were considered in [7] (Belief Propagation), and [9] (quantized Min-Sum). The results of [10] also show how to design Finite Alphabet Iterative Decoders (FAIDs) strongly robust to hardware noise.

On the other hand, faulty LDPC encoders have not been studied much so far, and to the best of our knowledge, [11] is the only work considering the faulty encoding problem. In [11], Hachem et al. evaluated the level of hardware noise that can be tolerated in the encoder. However, the results of [11] do not indicate how to construct a practical encoder and do not propose any particular robust encoding solution.

When the hardware is assumed perfect, many efforts have been made for the design of low complexity LDPC encoders. Richardson and Urbanke [1] proposed three methods for the construction of encoders from a given parity check matrix. Li et al [2] also proposed low complexity encoding architectures for Quasi-Cyclic (QC) codes. Particular code constructions such as Zig-Zag codes [3], Irregular Repeat Accumulate (IRA) codes [4], and Low Density Generator Matrix (LDGM) codes [5], are known to guarantee low encoding complexity.

In this paper, we evaluate the robustness of the above existing encoding solutions under faulty hardware. We represent the faulty hardware effect on the encoder by an error model for the XOR gates used in the encoder. We then review the existing encoding solutions [1], [2], and code constructions [3]–[5]. For each of the considered encoding solutions and code constructions, we provide an analytic expression of the error probability of the encoder. From the error probability expressions, we then evaluate the robustness of all the encoders. To finish, we identify the two best encoding solutions in terms of robustness to faulty hardware, and we compare their performance with Monte-Carlo simulations.

The outline of the paper is as follows. Section II introduces the notations for LDPC codes and describes the XOR gate error model we consider. Section III evaluates the robustness of the general encoding solutions [1] while Section IV considers the particular code constructions [3]–[5]. Section V provides the Monte-Carlo simulation results.

II. LDPC CODES AND ERROR MODELS

In this section, we first introduce our notations for LDPC codes. We then describe the XOR gate error model that represents the faulty hardware effect on the encoder .

A. LDPC Codes

Denote by H a binary parity check matrix of size $n \times m$. An LDPC code is defined as the null-space of the parity check matrix H. A binary vector $\mathbf{x}$ of length n is a codeword if and only if it verifies

$$H^T \mathbf{x} = \mathbf{0}. \tag{1}$$

With LDPC codes, the parity check matrix H is sparse and has to be designed in order to obtain good decoding performance, see [12] for instance. Once H is fixed, the corresponding encoder has to be constructed.

Denote by $\mathbf{u}$ the information sequence of length $k = n-m$. The encoding operation has to transform the information sequence $\mathbf{u}$ into a codeword $\mathbf{x}$ that satisfies (1). For this, [1] proposes three general solutions to construct an encoder from a given parity check matrix H. Although very general as they apply to any H, these solutions exhibit high encoding complexity in $O(n^2)$. On the other hand, particular code constructions such as Zig-Zag codes [3], IRA codes [4], and LDGM codes [5], are known to guarantee low encoding complexity.

In the following, before evaluating the robustness of the existing encoding solutions [1] and of the particular code constructions [3]–[5] under faulty hardware, we first introduce

the error model we consider for the faulty hardware effect on the encoder.

B. XOR Gate Error Model

All the encoding solutions that will be considered in this paper can be realized from XOR gates only. Consequently, we assume that hardware errors are introduced at the XOR gate level. Denote by p_{xor} the error probability of a 2-inputs XOR gate. The faulty 2-inputs XOR operator $\tilde{\oplus}$ is defined as

$$a \,\tilde{\oplus}\, b = \begin{cases} a \oplus b & \text{with prob. } 1 - p_{\text{xor}}, \\ 1 \oplus (a \oplus b) & \text{with prob. } p_{\text{xor}}, \end{cases} \tag{2}$$

where a and b are binary digits and $a \oplus b$ is the (perfect) XOR sum of a and b. The error model described in (2) is memoryless and data-independent. It is considered as a first step of the analysis.

The expression $(a_1 \,\tilde{\oplus}\, \ldots \,\tilde{\oplus}\, a_K)$ gives the faulty XOR sum of K binary digits $(a_1, \ldots, a_K)$. The error probability

$$P_e^{(K)}(p_{\text{xor}}) = \Pr\left((a_1 \,\tilde{\oplus}\, \ldots \,\tilde{\oplus}\, a_K) \neq (a_1 \oplus \cdots \oplus a_K)\right) \tag{3}$$

can be expressed from [13, Section 3.8] as

$$P_e^{(K)}(p_{\text{xor}}) = \frac{1}{2} - \frac{1}{2}(1 - 2p_{\text{xor}})^{(K-1)}. \tag{4}$$

The error probability $P_e^{(K)}(p_{\text{xor}})$ depends on the number $(K-1)$ of involved 2-inputs XOR gates, and on the XOR gate error probability p_{xor}. However, $P_e^{(K)}(p_{\text{xor}})$ does not depend on the order the elementary XOR operations are performed.

In the following, we rely on (4) to provide analytic expressions of the error probabilities of the existing encoding solutions [1]–[5] under faulty hardware. From the error probability expressions, we then evaluate the robustness of the encoding solutions.

III. General Encoding Solutions

In this section, we describe the three general encoding solutions of [1], that are encoding from the generator matrix, Lower Triangular encoding, and Approximate Lower Triangular encoding. We evaluate the robustness of these encoding solutions under the XOR gates error model.

In the remaining of the paper, codewords will be in systematic form $\mathbf{x} = [\mathbf{u}, \mathbf{p}]^T$, where $\mathbf{u}$ is the information sequence of length k, and $\mathbf{p}$ is the parity vector of length m.

A. Encoding from the Generator Matrix

From Gaussian elimination, the parity check matrix H can be put in systematic form $H = [P,\ I_m]^T$, where I_m is the identity matrix of size $m \times m$ and P is a matrix of size $m \times (n-m)$. A generator matrix G of size $n \times k$ can be constructed as $G = \left[I_{(n-m)},\ P^T\right]^T$. The encoding operation consists of computing the codeword $\mathbf{x} = [\mathbf{u}, \mathbf{p}]^T$ as

$$\mathbf{x} = G\mathbf{u}. \tag{5}$$

As the matrix P obtained from Gaussian elimination is not sparse in general, the encoding operation (5) has a high encoding complexity in $O(n^2)$.

For a given parity check matrix H in systematic form, we express the error probability of the encoding operation (5) as

$$P_e = \frac{1}{n}\sum_{i=1}^{m}\left(\frac{1}{2} - \frac{1}{2}(1 - 2p_{\text{xor}})^{(N_i-1)}\right), \tag{6}$$

where N_i is the number of non-zero components in the i-th line of P.

In order to evaluate the robustness of the encoding operation (6), we have constructed a collection of parity check matrices from the Progressive Edge Growth (PEG) algorithm [14]. All the constructed parity check matrices have the same variable node degree $d_v = 3$, but different check node degrees d_c and information sequence lengths m. For each of the constructed parity check matrices H, we have calculated the encoding error probability P_e from (6).

Figure 1 (a) represents the obtained encoding error probabilities with respect to m for $p_{\text{xor}} = 10^{-3}$. The encoding error probabilities are high because the matrices P are not sparse. Furthermore, the error probabilities increase with the information sequence length m. When m is large enough, the error probabilities become even higher than the Belief Propagation (BP) thresholds of the decoder [12], which makes it impossible for the decoder to recover the correct codeword $\mathbf{x}$, even when the channel is noiseless. We also note that the error probabilities increase when the code rate decrease. This is expected because the error probability (6) increases with the N_i which themselves increase with m and n.

As a conclusion, encoding from the generator matrix not only induces high encoding complexity but also exhibits poor robustness to hardware errors. In order to reduce the encoding complexity, an encoding solution called Approximate Lower Triangular encoding has been proposed in [1]. Approximate Lower Triangular encoding is based on Lower Triangular encoding which we now describe.

B. Lower Triangular Encoding

From Gaussian elimination, the parity check matrix H can also be put in lower triangular form $H = [Q\ \ T]^T$. T is a lower triangular matrix of size $m \times m$ with ones in the diagonal and non-zero components in the lower part of the matrix only. Q is a matrix of size $m \times (n-m)$. The parity part $\mathbf{p}$ of the codeword $\mathbf{x}$ can be computed from (1) by back-substitution as

$$p_1 = \sum_{k=1}^{n-m} H_{1,k}u_k, \tag{7}$$

$$\forall i = 2, \ldots, m, \quad p_i = \sum_{k=1}^{n-m} H_{i,k}u_k + \sum_{k=1}^{i-1} H_{i,(n-m)+k}p_k.$$

As the matrices Q and T are not sparse, the encoding complexity is still in $O(n^2)$.

We now express the error probability of Lower Triangular encoding. Denote by N_i the number of non-zero components in the i-th line of Q, and denote $\mathcal{T}_i$ the positions of the non-zero components in the i-th line of T, excluding the diagonal

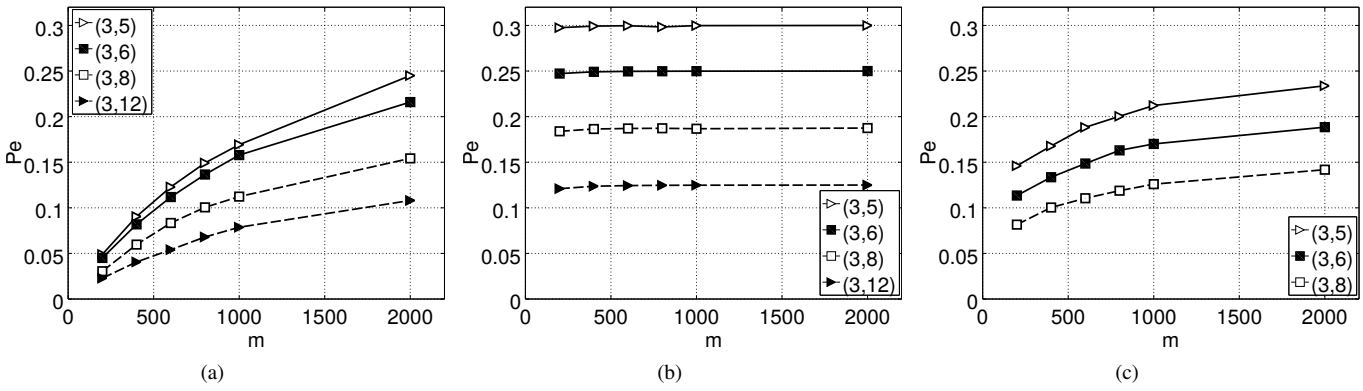

Fig. 1. Error probabilities with respect to m, with $p_{\text{xor}} = 10^{-3}$ (a) Encoding from the generator matrix, (b) Lower Triangular encoding, (c) Approximate Lower Triangular encoding

term. The successive error probabilities $P_{e,i}$ in the parity bits p_i can be calculated recursively as

$$P_{e,1} = \frac{1}{2} - \frac{1}{2}(1-2p_{\text{xor}})^{N_1-1},$$
$$\forall i = 2,\ldots,m, \quad P_{e,i} = \frac{1}{2} - \frac{1}{2}(1-2p_{\text{xor}})^{N_i-1}\prod_{k\in\mathcal{T}_i}(1-2P_{e,k}).$$

The overall encoding error probability is given by

$$P_e = \frac{1}{n}\sum_{i=1}^{m} P_{e,i}. \tag{8}$$

Figure 1 (b) represents the encoding error probabilities with respect to m for $p_{\text{xor}} = 10^{-3}$, for the same parity check matrices H considered in Section III-A. We see that whatever the considered code, the error probability is very high even for small values of m. This is due to the non-sparsity of G and T, and also to error propagation induced by the recursive computation of the parity bits (7). In addition, in all cases, the error probabilities reach saturation levels that correspond to error probabilities of $1/2$ over the parity bits.

Lower Triangular encoding shows both high complexity and no robustness to hardware noise. However, it is an building block of a lower complexity encoding solution called Approximate Lower Triangular encoding [1], which we present in the next paragraph.

C. Approximate Lower Triangular Encoding

In [1], it is shown that from line and column permutations, the matrix H can be put in the following form

$$H = \begin{bmatrix} A & B & T \\ C & D & E \end{bmatrix}^T \tag{9}$$

where A (of size $(m-g)\times(n-m)$), B ($(m-g)\times g$), C ($g\times(n-m)$), D ($g\times g$), and $E((m-g)\times g)$ are sparse matrices and T ($(m-g)\times(m-g)$) is a lower triangular sparse matrix. The parameter g is called the gap of the code. The block matrices $A,\ldots,E,T$, are sparse because (9) is obtained from line and column permutations only. Here, the codeword will be decomposed as $\mathbf{x} = [\mathbf{u}, \mathbf{p}_1, \mathbf{p}_2]^T$, where $\mathbf{p}_1$ and $\mathbf{p}_2$ are binary vectors of length g and $(m-g)$, respectively.

In [1], it is shown that the encoding operation is equivalent to solving the system

$$A\mathbf{u} + B\mathbf{p}_1 + T\mathbf{p}_2 = 0 \tag{10}$$
$$(-ET^{-1}A + C)\mathbf{u} + (-ET^{-1}B + D)\mathbf{p}_1 = 0 \tag{11}$$

with respect to $\mathbf{p}_1$ and $\mathbf{p}_2$. The encoding can thus be realized in two steps

1) Solve (11) by computing $\mathbf{p}_1 = -\Phi^{-1}(-ET^{-1}A + C)\mathbf{u}$ where $\Phi = -ET^{-1}B + D$. The matrix Φ^{-1} is not sparse.
2) Solve (10) by computing $\mathbf{p}_2$ recursively as in (7). The matrices A, B, and T are sparse.

Operations 1 and 2 induce a total encoding complexity in $O(n + g^2)$. The authors of [1] show that it is possible to obtain (9) with a gap value g such that the encoding complexity is in $0.017^2n^2 + O(n)$. The encoding complexity is still in $O(n^2)$, but the constant is very small.

The error probabilities of the encoding operations (10) and (11) can be obtained from (6) and (8). Figure 1 (c) gives the error probabilities with respect to m for $p_{\text{xor}} = 10^{-3}$ for the parity check matrices constructed in Section III-A. Here again, the error probabilities are high because of non-sparse computation in (11) and of iterative, although sparse, computation in (10). As a consequence, Approximate Lower Triangular encoding is not robust neither to hardware errors.

To conclude, the general encoding solutions not only induce important encoding complexity, but they also show poor robustness to hardware errors. To overcome the complexity issue, several particular code constructions [2]–[5] have been shown to guarantee low encoding complexity. The next section describes these particular code constructions and evaluate their robustness to faulty hardware.

IV. Particular Code Constructions

In this section, we evaluate the robustness of the encoding for two particular code constructions that are Zig-Zag codes [3], and LDGM codes [5]. We also discuss IRA codes [4], QC-codes [2], and concatenated LDGM codes [5].

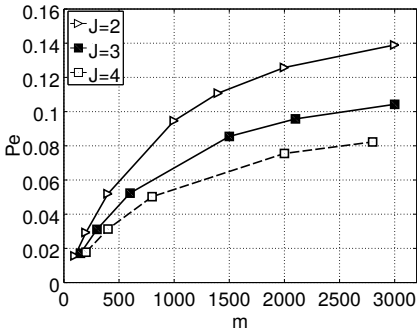

Fig. 2. Zig-Zag codes, error probabilities with respect to m, with $p_{\text{xor}} = 10^{-3}$

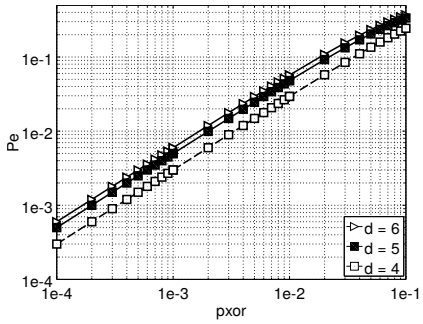

Fig. 3. Error probability with respect to p_{xor} for LDGM codes

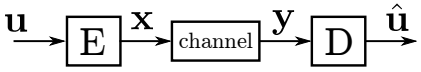

Fig. 4. Encoder alone

A. Zig-Zag codes

For Zig-Zag codes, denote by $u_{i,j}$ the information bits, with $i \in \{1, \ldots, I\}$, $j \in \{1, \ldots, J\}$, $m = I$, and $n = I \times J$. The parity bits p_i, $i \in \{1, \ldots, I\}$, are calculated iteratively as

$$p_1 = \sum_{j=1}^{J} u_{1,j}, \quad \forall i = 2, \ldots, I, \quad p_i = p_{i-1} + \sum_{j=1}^{J} u_{i,j}. \quad (12)$$

The codeword $\mathbf{x}$ is composed by all the information bits $u_{i,j}$, and by the parity bits p_i. A Zig-Zag code has a low encoding complexity in $O(n)$.

We express the successive error probabilities $P_{e,i}$ of the parity bits p_i recursively as

$$P_{e,1} = \frac{1}{2} - \frac{1}{2}(1 - 2p_{\text{xor}})^{(J-1)}$$
$$\forall i = 2, \cdots, I, \; P_{e,i} = \frac{1}{2} - \frac{1}{2}(1 - 2p_{\text{xor}})^{(J-1)}(1 - 2P_{e,i}).$$

The overall encoding error probability is given by

$$P_e = \frac{1}{I(J+1)} \sum_{i=1}^{I} P_{e,i}. \quad (13)$$

Figure 2 represents the encoding error probabilities P_e with respect to m for various values of J and for $p_{\text{xor}} = 10^{-3}$. The error probabilities are high because of error propagation in (12), and as a consequence Zig-Zag encoding is not robust to hardware noise. The error probabilities are lower than for *e.g.*, the encoding with the generator matrix (see Figure 1 (a)). However, the Zig-Zag codes we consider are codes with high rate $r = J/(J-1)$ and low correction capabilities.

An IRA code [4] is the concatenation of an irregular repetition code and of a Zig-Zag code. The hardware noise does not affect the repetition encoding, but it affects the Zig-Zag part. We have shown that Zig-Zag encoding is not robust to hardware errors, and as a consequence, IRA encoding is not robust to hardware errors.

B. QC codes

For QC-codes, the encoding solutions proposed in [2] have a circuit complexity that is linear with the codeword length n. The low circuit complexity is due to parallel computation and electronic components reuse. However, the actual number of operations needed to realize the encoding is still in $O(n^2)$. As a consequence, the encoder error probabilities will be in the same order of magnitude of the encoding solutions presented in Section III, and encoding for QC-codes will not be robust to hardware errors.

C. LDGM codes

Consider the the parity check matrix $H = \begin{bmatrix} P & I_m \end{bmatrix}^T$ and the generator matrix $G = \begin{bmatrix} I_{(n-m)} & P^T \end{bmatrix}^T$ in systematic forms. With LDGM codes [5], the parity matrix P is directly constructed sparse. As a result, both the matrices H and G are sparse. The encoding can be realized from (5) and has a complexity in $O(n)$.

The encoding error probability is given by (6), where $N_i = d$, and d is the column degree for the matrix P. For LDGM codes, P_e does not vary with m because d is fixed and does not depend on m. Figure 3 represents the error probabilities with respect to p_{xor} for various column degrees d. As P is sparse, the encoding error probabilities are small.

LDGM codes are thus naturally robust to hardware noise. However, the decoder performance of LDGM codes is not as good as the performance of LDPC codes, as we now illustrate with Monte-Carlo simulations. Note that to improve the decoder performance, a usual solution is to concatenate two LDGM codes [5]. However, the second LDGM code has to be a very high rate code, with very high values of column degree d. From (6) large values of d imply high encoder error probability, and as a consequence, encoding with two concatenated LDGM codes will not be robust to hardware noise.

V. Monte-Carlo Simulations

In this section, we discard all the encoding solutions with error propagation (Lower Triangular, Approximate Lower Triangular, Zig-Zag codes) and evaluate the performance of the three following encoding solutions.

1) *LDPC codes, encoding from the generator matrix* (transmission scheme of Figure 4).

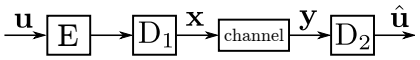

Fig. 5. Encoder and decoder

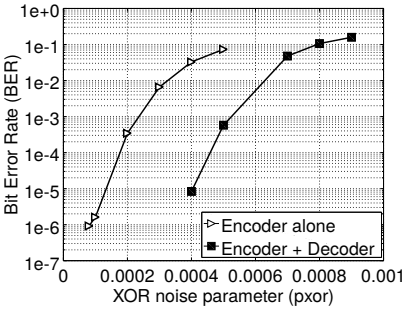

Fig. 6. BER with respect to p_{xor} for solutions 1 and 2

2) *LDPC codes, encoding from the generator matrix, decoder at the encoder* (transmission scheme of Figure 5). We add a decoder D_1 at the encoder part. D_1 has to recover the codeword before transmission on the channel.
3) *LDGM codes, encoding from the generator matrix* (transmission scheme of Figure 4).

In all cases, both the encoder and the decoder are faulty. All the decoders are faulty 7-levels offset min-sum decoders with the Full-Depth error model described in [9] and decoder noise parameter $p = 10^{-3}$.

We first compare Solutions 1 and 2. We choose a regular $(3, 6)$-code with $m = 500$ and we set 100 iterations for the decoders. The channel parameter is fixed to $\alpha = 0.03$. Figure 6 represents the Bit Error Rate (BER) with respect to p_{xor} for Solutions 1 and 2. As expected, we observe an important loss in performance when there is no decoder D_1 at the encoder part.

We now compare Solutions 2 and 3. We consider codes of rate $1/4$ with $m = 400$. For LDPC codes, we consider a $(3, 4)$ regular codes. For LDGM codes, the matrix P is constructed as a $(4, 6)$ regular codes. The BERs with respect to α of both solutions are represented in Figure 7 for various values of p_{xor}. For LDGM codes, we see that the BER does not vary much with p_{xor}. We also see that despite their robustness to hardware errors, LDGM codes give poor BER performance, and in particular high error floor. On the opposite, we see that for LDPC codes, a small variation of p_{xor} can induce an important loss in BER performance. For $p_{\text{xor}} = 10^{-3}$ and $p_{\text{xor}} = 8.10^{-4}$, D_1 is not able to fully completely the original codeword. For $p_{\text{xor}} = 5.10^{-4}$, D_1 can correct almost all the encoder noise and we get better BER performance.

As a conclusion, LDGM encoders are robust to hardware errors at the price of a high error floor. Solution 2 is less robust to hardware errors but shows better decoding performance when the hardware noise is small enough.

VI. Conclusion

From the error probability analysis, we have shown that most of the existing encoding solutions are not robust to hardware noise. The Monte-Carlo simulations show that adding a decoder at the encoder part can help but does not provide a strongly robust encoding solution. At the end, only LDGM codes provide a robust encoding solution. However, the Monte-Carlo simulations show that LDGM codes exhibit poor decoding performance and in particular high error floor.

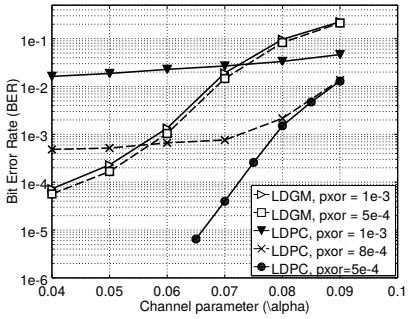

Fig. 7. BER with respect to α for solutions 2 and 3

As a result, the encoding operation should be better protected by adding some redundancy either at the hardware level or at the software level.

Acknowlegement

This work was funded by the Seventh Framework Programme of the European Union, under Grant Agreement number 309129 (i-Risc) and by the European Project NEWCOM#.

References

[1] T. Richardson and R. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. on Inf. Th.*, vol. 47, no. 2, pp. 638–656, 2001.
[2] Z. Li, L. Chen, L. Zeng, S. Lin, and W. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes," *IEEE Trans. on Comm.*, vol. 53, no. 11, pp. 1973–1973, 2005.
[3] L. Ping, X. Huang, and N. Phamdo, "Zigzag codes and concatenated zigzag codes," *IEEE Trans. Inf. Th.*, vol. 47, no. 2, pp. 800–807, 2001.
[4] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proc. 2nd Int. Symp. on Turbo codes*, 2000, pp. 1–8.
[5] J. Garcia-Frias and W. Zhong, "Approaching shannon performance by iterative decoding of linear codes with low-density generator matrix," *IEEE Comm. Letters*, vol. 7, no. 6, pp. 266–268, 2003.
[6] S. Zaynoun, M. Khairy, A. Eltawil, F. Kurdahi, and A. Khajeh, "Fast error aware model for arithmetic and logic circuits," in *IEEE 30th International Conference on Computer Design (ICCD)*. IEEE, 2012, pp. 322–328.
[7] L. Varshney, "Performance of LDPC Codes Under Faulty Iterative Decoding," *IEEE Trans. Inf. Th.*, vol. 57, no. 7, pp. 4427–4444, 2011.
[8] C.-H. Huang, Y. Li, and L. Dolecek, "Gallager B LDPC Decoder with Transient and Permanent Errors," *IEEE Trans. Comm.*, vol. 62, no. 1, pp. 15–28, 2014.
[9] C. K. Ngassa, V. Savin, E.Dupraz, and D. Declercq, "Density Evolution and Functional Threshold for the Noisy Min-Sum Decoder," *Submitted to IEEE Transactions on Communications*, May 2014.
[10] E. Dupraz, D. Declercq, B. Vasic, and V. Savin, "Finite alphabet iterative decoders robust to faulty hardware: analysis and selection," in *8th Int. Symp. on Turbo Codes*, 2014, pp. 1–10.
[11] J. Hachem, I. Wang, C. Fragouli, S. Diggavi *et al.*, "Coding with encoding uncertainty," in *Proc. Int. Symp. on Inf. Th.*, 2013, pp. 276–280.
[12] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Th.*, vol. 47, no. 2, pp. 619–637, 2001.
[13] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.
[14] X. Hu, E. Eleftheriou, and D. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inf. Th.*, vol. 51, no. 1, pp. 386–398, 2005.