CPE: Codeword Prediction Encoder

Satish Grandhi[1], Elsa Dupraz[2], Christian Spagnol[1], Valentin Savin[3], Emanuel Popovici[1]

[1] Department of Electrical and Electronic Engineering, University College Cork, Cork, Ireland

[2] Telecom Bretagne; UMR CNRS 6285 Lab-STICC, Brest, France

[3] CEA-LETI, Minatec Campus, Grenoble, France

sagrand@ue.ucc.ie, elsa.dupraz@telecom-bretagne.eu, christian.spagnol@ue.ucc.ie, valentin.savin@cea.fr, e.popovici@ucc.ie

I. INTRODUCTION

Low density parity check (LDPC) codes are known to provide excellent error correction performance that closely approaches the Shannon capacity of noisy transmission channels. However, in the future systems of communication and storage, errors may not only come from the transmission channels, but also from the faulty hardware [1]. Thus, the study of novel techniques for reliable data transmission using unreliable hardware is an increasing priority. LDPC decoders on unreliable hardware have been widely investigated recently [2] and it was shown that LDPC decoders are naturaly robust to hardware errors, with no need for additional circuitry. Unfortunately, it was also shown that LDPC encoders completely fail when they are built from unreliable gates [3]. The focus of the current work is thus on constructing reliable LDPC encoders built from unreliable gates.

This paper introduces a novel reliability driven fault tolerant methodology known as Codeword Prediction Encoder (CPE) for reliable LDPC encoding by augmenting extra logic to correct the errors introduced during the encoding process. The approach presented here can be seen as an expansion of the Check Symbols Generation [4], where circuitry is added to a combinatorial network to generate extra bit to ensure parity. We formalize these approaches and extend them to take full advantage of the power of error correction codes to enable correction of the faults, and not just detection. The CPE simulator provides a unified platform which comprises of novel encoder and fault tolerant LDPC decoders. Simulations results prove that it is possible to retrieve the original information by employing particular configurations of these encoders and decoders. In general, output BER is reduced by upto 10K times by adopting CPE mechanism as compared to transmitting data directly.

II. LDPC CODES AND ERROR MODELS

A binary information sequence u of length k has to be transmitted through a noisy channel. The data is protected against noise with an LDPC code that encodes the information sequence u into a codeword x of length $n > k$. An LDPC code is defined by its binary parity check matrix H of size $m \times n$. A binary vector x of length n is a codeword of the LDPC code if it satisfies $Hx^T = 0$, where T is the transpose operator. For LDPC codes, the parity check matrix H is sparse, *i.e.*, it contains only a few non-zero components. In the following, we will denote by d_v and d_c the number of 1's in each row and in each column of H, respectively.

Once the LDPC parity check matrix H is fixed, one can construct a generator matrix G of size $k \times n$, where $k =$ $n - m$, that verifies $HG^T = 0$. The encoding operation can then be realized from the generator matrix as $x = uG$. Several solutions have been proposed to construct a generator matrix G from the parity check matrix H, see [5] for a review. Most of the usual solutions consider systematic encoding, for which the codeword $x = [u, p]$ contains both the information sequence u and m parity bits given by p. In this case, the left hand side of the generator matrix G is the identity matrix of size $k \times k$.

LDPC codes were initially introduced under the assumption of reliable hardware. Here we present the Von Neumann error model we consider for the unreliable gates that are used in the encoder and in the decoder. A Von Neumann erroneous gate is modeled as an ideal logic gate cascaded with an independent error injecting XOR gate. The noise variable E at the XOR gate input takes value 1 with a given probability p_g that represents the gate error probability. Denote by $\tilde{x}$ the codeword at the output of an encoder realized from unreliable gates. Under this model, it was shown that most of the standard LDPC encoders completely fail when they are built from unreliable gates [3].

III. CODEWORD PREDICTION ENCODER (CPE)

Here, we consider two cases that are non-systematic encoding and systematic encoding. In case of non-systematic encoding as described in Fig 1(a), in addition to the parity bits p contained in $\tilde{x}$, we now compute m_a extra parity bits $\tilde{p}_a$ from u. The vector $\tilde{x}_a = [\tilde{x}, \tilde{p}_a]$ is called the augmented codeword. Before channel transmission, $\tilde{x}_a$ is passed through an LDPC decoder, denoted by D_{CPE}, in order to eliminate the encoding errors. After decoding, only $\tilde{x}$ is transmitted through the channel.

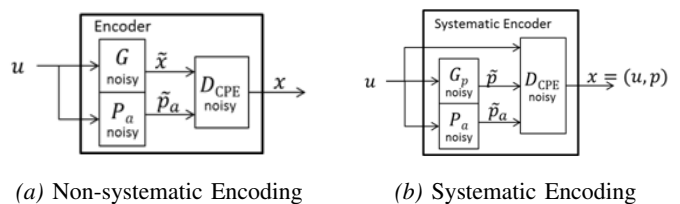*(a)* Non-systematic Encoding   *(b)* Systematic Encoding

Fig. 1. The CPE approach

The CPE approach for systematic encoding is illustrated in Fig 1(b). We denote by G_p the G sub-matrix of size $n - k \times k$, corresponding to parity bits. Thus, $x = [u, p]$, and the parity bits p can be computed by $p = u \cdot G_p$. In this case only the parity bits p can be affected by gate errors, while the data bits u are assumed to be error free. We also denote the circuit composed either by G and P_a (non-systematic case) or by G_p

and P_a (systematic case) as G_{CPE}. Both for systematic and non-systematic encoding, the additional parity bits $\tilde{p}_a$ are computed from a split-extended construction introduced in [6].

In addition to CPE, we introduce a second level of protection based on the critical gates. The criticality degree of a node X, denoted by cdeg(X), is defined as the number of output nodes to which X is connected by at least one path. In our simulations, we fix a criticality threshold (CT): Nodes X with cdeg(X) > CT are considered to be "protected" (e.g. by increasing area), so as to make then reliable (error-free). Hence, errors are injected only in nodes X with cdeg(X) < CT.

IV. Experimental Results

In our simulations, both encoder and decoder are assumed to be error prone. We consider regular LDPC codes with different column weights for the parity check matrix, namely $d_v = 3$ and $d_v = 4$, and different coding rates, namely $r = 1/2$ and $r = 3/4$. We have employed three state-of-the-art reliability enhanced LDPC decoders within the CPE CAD flow: Min-Sum (MS), Self-Corrected Min-Sum (SCMS), and Gallager B with Extended Alphabet(Gal-B).

TABLE I. CRITICAL GATE COUNT FOR DIFFERENT ENCODING SCHEMES

Encoder	G_{CPE} Node Count	CT=10	CT=20	CT=50
dv3-r12	44399	3373	1844	833
dv3-r34	28182	2288	1240	537
dv4-r12	45175	3424	1851	824
dv4-r34	27167	2112	1183	488

A. Critical Nodes

Table I lists the count of total & critical nodes within the four encoding schemes, for critical thresholds CT set to 10, 20, and 50. As expected, lower the value of critical threshold, higher the number of critical nodes within the encoder. As a tradeoff, a lower critical threshold is also expected to lead to lower encoding error probability. To illustrate this, we employed an encoder with $r = 3/4$ and $d_v = 4$ and D_{CPE} was set to Min-Sum model. As depicted in Fig. 2, the output BER value reduces with the critical threshold values. It infers that more the number of nodes safe guarded from possible soft errors, higher the possibility of retrieving the original information.
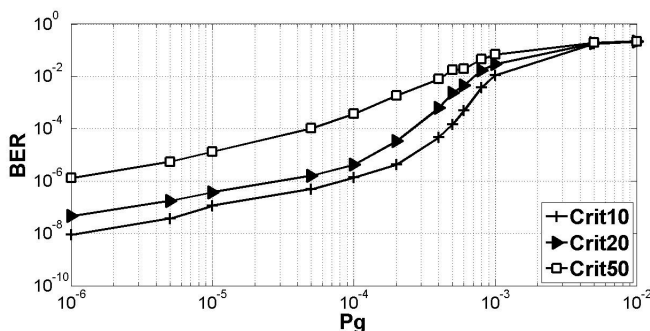

Fig. 2. Critical Threshold impact on Output BER

B. Impact of Decoder Configuration

For faulty decoders, we assume that the output of every variable and check node function computation is flipped with a probability $p = 10^{-3}$. For non-binary message alphabets, flipping the output value means that a value different from the correct one is selected uniformly at random from the alphabet. Fig. 3 highlights the output BER values for different faulty decoders and the default non-CPE approach when the critical threshold is set to 20. The encoder employed in this particular case has the following parameters $r = 3/4$ and $d_v = 3$. Clearly, SCMS and MS decoders provide the best performance by reducing the error rates to upto 10K times better than the default encoder. Gal-B provides upto 100 times improvement in terms of error correction. It also illustrate the performance of CPE compared to the default encoding mechanism. This represents a significant improvement, by more than 5 orders of magnitude.
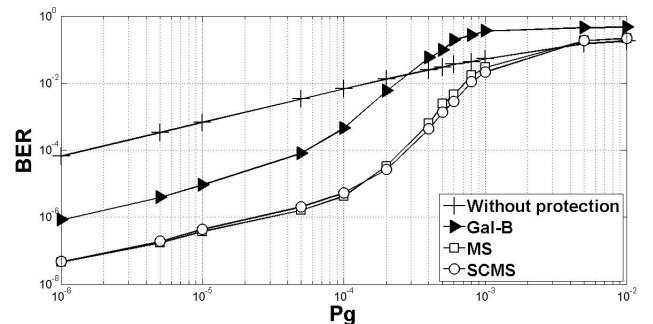

Fig. 3. Decoder Configuration impact on Output BER

V. Conclusion

A novel fault tolerant methodology known as Codeword Prediction Encoder (CPE) for reliable data transmission using unreliable hardware is proposed. Simulation results show that performance of CPE is much better as compared to transmitting data by employing traditional encoding methodology. It is shown that by employing Min-sum decoding mechanisms and a strong encoder $r = 1/2$ and $d_v = 4$, it is possible to correct all errors given that gate errors smaller than $P_g = 6e^{-4}$. In general, CPE performance improvement of upto 10K is observed when compared to the normal encoding mechanism.

REFERENCES

[1] S. Borkar, "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation," *Micro, IEEE*, vol. 25, no. 6, pp. 10–16, 2005.

[2] C. K. Ngassa, V. Savin, E. Dupraz, and D. Declercq, "Density evolution and functional threshold for the noisy Min-Sum decoder," *IEEE Transactions on Communications*, vol. 63, no. 5, pp. 1497–1509, May 2015.

[3] E. Dupraz and D. Declercq, "Evaluation of the robustness of LDPC encoders to hardware noise," in *IEEE BlackSeaCom conference*, 2015, pp. 1–5.

[4] M. R. Choudhury and K. Mohanram, "Low cost concurrent error masking using approximate logic circuits," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 32, no. 8, pp. 1163–1176, 2013.

[5] T. Richardson and R. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 638–656, 2001.

[6] V. Savin, "Split-extended LDPC codes for coded cooperation," in *International Symposium on Information Theory and its Applications (ISITA)*, 2010, pp. 151–156.