A Study on VHDL Implementation of a Class of Irregular Structured LDPC Codes applied to 100 Gbps Optical Networks

Antônio Unias de Lucena
Renato Baldini Filho
DECOM-FEEC-UNICAMP
Av. Albert Einstein, 400
13083-852 Campinas – SP -Brazil
antoniounias@gmail.com

Abstract - **This paper presents a study on the VHDL implementation of a class of binary irregular structured LDPC codes (IS-LDPC) applied to 100 Gbps optical networks. A comparison between two iterative decoding algorithms for irregular structured LDPC codes, sum-product based on log-likelihood ratio and min-sum, is used to define the best choice for implementation. The performances of IS-LDPC codes are evaluated on an AWGN channel. The aim of this paper is to select out of a class IS-LDPC codes those ones with the best performance using the MS algorithm for VHDL implementation.**

Index Terms - LDPC codes; optical networks; VHDL.

I. INTRODUCTION

The increasing traffic in optical telecommunication networks has demanded links with even higher transmission rates. However, higher rates mean more noise and interference introduced by optical and electronic devices [1]. Efficient channel coding schemes, such as, low-density parity-check (LDPC) and turbo codes, have been devised to overcome those sources of errors [1],[2],[3],[4],[5],[6].

LDPC codes can achieve near optimum Shannon limit performance over the additive white Gaussian noise (AWGN) channel [8]. The sparseness of 1's in the binary parity check matrix **H** makes the iterative decoding particularly attractive. The iterative decoding of LDPC codes allows a high degree of parallelism, which makes it suitable for high data rate communications.

Iterative decoding algorithms for LDPC codes are bounded by a trade-off between decoding performance, in terms of bit error rate (*BER*), and implementation complexity. Moreover, the performance varies with the length and the structure of the parity check matrix of the LDPC codes.

Among of the iterative decoding algorithms, the sum-product (SP) algorithm achieves the best performance however it demands a high hardware complexity. An alternative is the min-sum (MS) algorithm that significantly reduces the implementation complexity at a cost of acceptable performance degradation. The complex computations at the check nodes are approximated to simple comparison and summation operations in the MS algorithm [7].

In general, LDPC codes can be categorized into regular and irregular codes. An LDPC code is regular if the weights of rows and columns in its parity check matrix are equal, otherwise it is irregular. Irregular LDPC codes have better performance than regular ones.

The aim of this paper is to select out of a class of irregular structured (IS) LDPC codes those ones with the best performance using the MS algorithm. The log-SP decoding algorithm is used as reference for the sake of performance comparison. The MS algorithm presents lower complexity for VHDL implementation than the log-SP decoding algorithm [7]. The IS-LDPC codes were designed to match with the specifications of the 100 Gbps optical networks.

II. IRREGULAR STRUCTURED LDPC CODES

The binary irregular structured (n, k) LDPC codes are built using a parity check matrix **H** generated by grouping circulant sub-matrices [2],[3]. Both code and the information lengths are denoted by n and k, respectively. A circulant matrix is generated by successive shifts of the first row (column) of a parent identity matrix $\mathbf{I}_m$ to obtain the following rows (columns). Fig. 1 shows two examples of circulant matrices $\mathbf{C}_{8,j}$ obtained from the parent identity matrix $\mathbf{I}_8$. The index j indicates the initial shift to the right of the first row of the identity matrix.

$$\mathbf{I_8} = \begin{pmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1 \end{pmatrix} \mathbf{C_{8,3}} = \begin{pmatrix} 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1\\ 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0 \end{pmatrix} \mathbf{C_{8,7}} = \begin{pmatrix} 0&0&0&0&0&0&0&1\\ 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0 \end{pmatrix}$$

Fig.1 Circulant matrices obtained from $\mathbf{I_8}$.

Let $\mathbf{N_p}$ be the set of the natural prime numbers. This set can be used to generate circulant sub-matrices that compose the parity check matrix **H**. For instance, fig. 2 shows a matrix **H** for a (32, 16) irregular structured code. Notice that **H** is in the systematic form $\mathbf{H} = [\mathbf{I}_{n-k} | \mathbf{P}]$ where **P** is a parity sub-matrix built by grouping four circulant sub-matrixes defined by the

first four elements of $\mathbf{N_p}$. This matrix **H** provides a fast encoding process [2]. The square null sub-matrix $\mathbf{O_8}$ has dimension 8.

$$\mathbf{H} = \begin{pmatrix} \mathbf{I_8} & \mathbf{O_8} & \mathbf{C_{8,2}} & \mathbf{C_{8,3}} \\ \mathbf{O_8} & \mathbf{I_8} & \mathbf{C_{8,5}} & \mathbf{C_{8,7}} \end{pmatrix}$$

Fig. 2 Parity check matrix **H** for the (32,16) irregular structured code.

Notice that $\mathbf{I}_m$ is used to generate the circulant sub-matrixes $\mathbf{C}_{m,j}$ of **P**, whereas $\mathbf{I}_{n-k}$ is related to the systematic part of **H**.

III. Iterative Decoding Process

Tanner graph is a graphic representation of the parity check matrix **H** of a LDPC code. This graph is composite of two sets of nodes: variable nodes v_i and check nodes c_j. There is a connection between v_i and c_j when the entry h_{ij} of the matrix **H** is equal to 1.

The log-SP algorithm is a well-known decoding algorithm for LDPC codes. It operates on the Tanner graph representation of the parity check matrix of a code. The most straightforward variant of the log-SP algorithm is the a posteriori probability (APP) decoding. A simplified version of the log-SP algorithm is the MS or maximum-likelihood sequence detection (MLSD). The iterative decoding process is illustrated in fig. 3. However, before describing the steps of the decoding algorithm, it is convenient to set some definitions:

$L(c_i)$: intrinsic information received by the decoder.

$L(r_{ji})$: message evaluated by the check node c_j and sent to the variable node v_i.

$L(q_{ij})$: message evaluated by the variable node v_i and sent to the check node c_j.

V_j: variable nodes connected to the check node c_j.

$V_{j\backslash i}$: variable nodes connected to the check node c_j with exception of the variable node v_i.

C_i : check nodes connected to the variable node v_i

$C_{i\backslash j}$: check nodes connected to the variable node v_i with exception of the check node c_j.

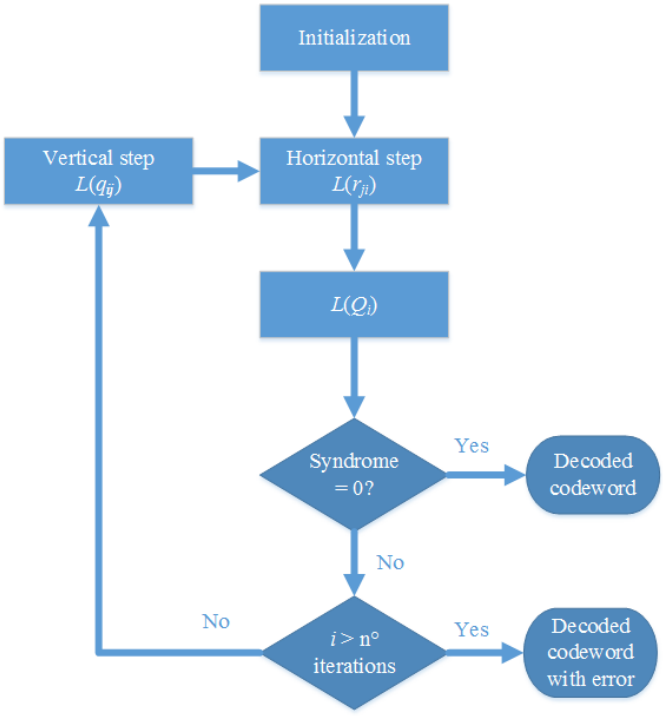

Fig. 3 Decoding fluxogram for a LDPC code.

The log-SP decoding algorithm is divided into the following steps [8]:

a) *Initialization*: the intrinsic message $L(c_i) = 2y_i/\sigma^2$ is evaluated where y_i is the received signal and σ^2 is the variance of the AWGN (additive white Gaussian noise) channel.

b) *Horizontal step*: each check node then sends to the variable nodes its new probabilities of 0 and 1, which are evaluated from the probabilities received from the variable nodes, excluding the check node that is going to receive that probabilities. The probabilities are evaluated by

$$L(r_{ji}) = \prod_{i' \in V_{j\backslash i}} \alpha_{i'j} . \Phi\left(\sum_{i' \in V_{j\backslash i}} \Phi(\beta_{i'j})\right), \quad (1)$$

where

$$\Phi(x) = -\log\left[tanh\left(x/2\right)\right] = \log\left(\frac{e^x+1}{e^x-1}\right). \quad (2)$$

c) *Vertical step*: each variable node sends to its connected check nodes the probabilities of 0 and 1. New probabilities are evaluated by summing the received probabilities from the check nodes, excluding the probabilities of the check node that is going to receive them. The new probabilities are evaluated by $L(q_{ij}) = L(c_i) + \sum_{j' \in C_{i\backslash j}} L(r_{j'i})$.

d) *Syndrome*: after horizontal and vertical steps, each entry of the codeword **c** is updated using $L(Q_i) = L(c_i) + \sum_{j \in C_i} L(r_{ji})$ and

$$c_i = \begin{cases} 1 & \text{if } L(Q_i) < 0 \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The syndrome is then evaluated and if it is a null vector the decoding process stops and the information is recovered.

Otherwise the process continues until the number of iterations set by the algorithm is reached.

When the log-SP decoding algorithm is implemented in VHDL, the vertical step consumes the greatest amount of logical elements. This happens because it is necessary to perform logarithmic operations to evaluate Φ(x). The function Φ(x) can be implemented in VHDL by lookup tables. However, if the **H** matrix has a greater number of 1's the number of lookup tables can be prohibitive. Notice that it is necessary a lookup table for each h_{ij} = 1 of the parity check matrix **H** of a LDPC code.

An alternative to the log-SP algorithm to reduce the number of logical elements is the MS decoding process. The vertical step of the MS algorithm is simplified by evaluating the minimum values of probabilities from the check nodes. Let $\alpha_{ij} = \text{sgn}(L(q_{ij}))$ and $\beta_{ij} = \text{abs}(L(q_{ij}))$, then $\Phi(x)$ can be evaluated by the following approximation [4]

$$\Phi\left(\sum_{i'} \Phi(\beta_{i'j})\right) \approx \Phi\left(\Phi\left(\min_{i'} \beta_{i'j}\right)\right) = \min_{i' \epsilon V_{j\setminus i}} \beta_{i'j}. \quad (4)$$

This simplifies the evaluation of $L(r_{ji})$ to

$$L(r_{ji}) = \prod_{i' \epsilon V_{j\setminus i}} \alpha_{i'j} . \min_{i' \epsilon V_{j\setminus i}} \beta_{i'j}. \quad (5)$$

Therefore there is no need of lookup tables for the MS decoding algorithm. Notice also that there is no need of knowing the channel characteristics [7], i.e., the initialization of the algorithm can be made by $L(q_{ij}) = y_i$.

IV. Results

The IS-LDPC codes are designed to operate at 100 Gbs optical networks. Therefore, the encoding and decoding algorithms of the (2000, 1000) and (4000, 2000) IS-LDPC codes should operate at the frequencies of 50 MHz and 25 MHz, respectively, for the VHDL implementation. The performances of the codes are analyzed on an additive white Gaussian noise (AWGN) channel, which is considered a good statistical model for the impairments found in an optical network. Then the goal of this work is to adjust the dimension of the parent identity matrix $\mathbf{I}_m$ that generates the IS-LDPC code to narrow the performance gap between log-SP and MS decoding algorithms.

For instance, four (2000, 1000) IS-LDPC codes are built using circulant sub-matrixes generated by different-size parent identity matrixes ($\mathbf{I}_{50}$, $\mathbf{I}_{100}$, $\mathbf{I}_{200}$ and $\mathbf{I}_{500}$) to find that with the best MS decoding performance.

A. (2000, 1000) IS-LDPC codes

Figures 4 to 7 show the performance, in terms of bit error rate (*BER*) versus energy per bit/unilateral noise power spectral density (E_b/N_0), of the (2000, 1000) IS-LDPC using log-SP and MS decoding schemes. Each code is decoded using five iterations. Four codes were implemented using parent identity matrixes with dimension values: 50, 100, 200 and 500.

Fig. 4 shows the performance of an IS-LDPC code built from an identity matrix with dimension 50. The log-SP algorithm presents 1.3 dB gain over the MS algorithm at *BER* = 10^{-4}.

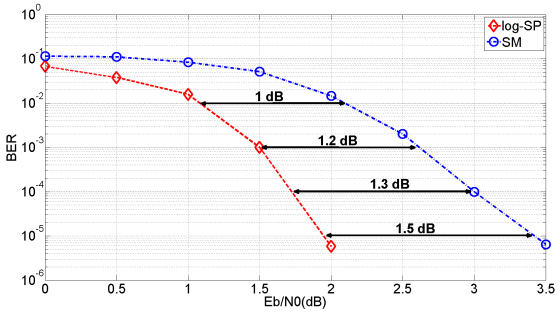

Fig. 4 (2000, 1000) IS-LDPC codes with parent identity matrix $\mathbf{I}_{50}$.

Fig. 5 presents the performance curves for (2000, 1000) IS-LDPC code using an identity matrix of dimension 100. The algorithm log-SP performs 0.4 dB, in terms of E_b/N_0, better than the MS algorithm for a *BER* = 10^{-4}.

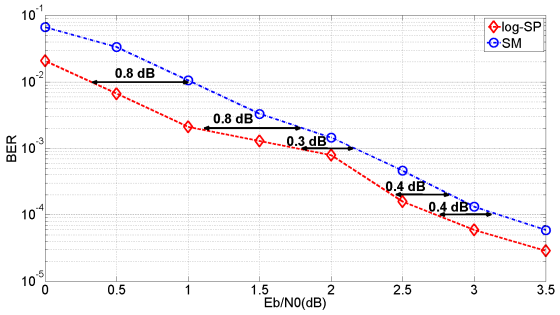

Fig. 5 (2000, 1000) IS-LDPC codes with $\mathbf{I}_{100}$.

Fig. 6 shows the performance of an IS-LDPC code built using an identity matrix with dimension 200. The log-SP algorithm presents 0.15 dB gain over the MS algorithm, for a *BER* = 10^{-4}.

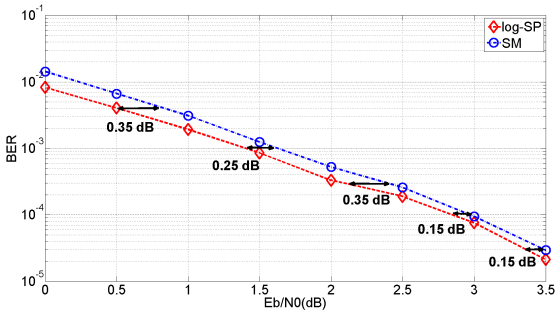

Fig. 6 (2000, 1000) IS-LDPC codes with $\mathbf{I}_{200}$.

Finally, fig. 7 presents an IS-LDPC code built with an identity matrix of dimension equal to 500. The decoding performances for both algorithms are almost coincident. The difference in performance is smaller than 0.1 dB between log-SP and MS algorithms, for a *BER* = 10^{-4}.

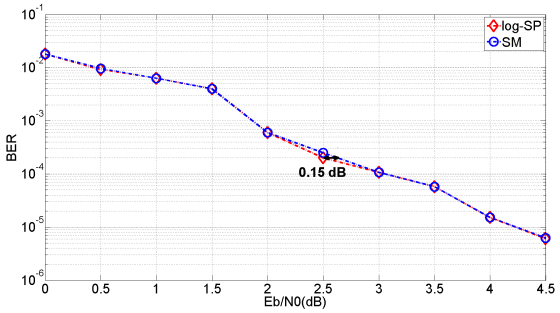

Fig. 7 (2000, 1000) IS-LDPC codes with $\mathbf{I}_{500}$.

Notice that by increasing the dimension of the parent identity matrix the performances of the log-SP and MS decoding algorithm become closer and closer. Further the increase in dimension reduces the number of 1's in **H,** which reduces the VHDL implementation complexity. Notice also that the (2000, 1000) IS-LDPC code generated by $\mathbf{I}_{50}$ presents the best performance for MS decoding algorithm. However, this code has more branches between variable and check nodes than the others.

B. (4000, 2000) IS-LDPC codes

Figures 8 to 12 show the performance, in terms of *BER* versus E_b/N_0, for the (4000, 2000) IS-LDPC codes using log-SP and MS decoding algorithms. Again each code is decoded using five iterations. Five codes were implemented using parent identity matrixes with dimension values: 50, 100, 200, 500 and 1000.

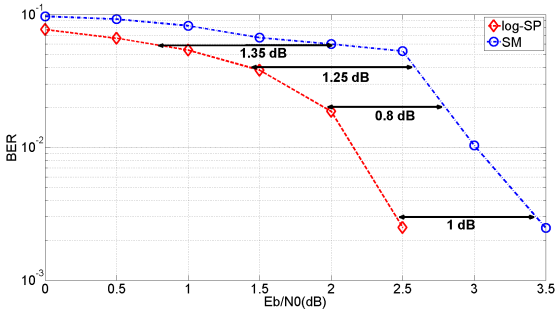

Fig. 8 (4000, 2000) IS-LDPC codes with $\mathbf{I}_{50}$.

Fig. 8 shows the performance of a (4000, 2000) IS-LDPC code with identity matrix with dimension 50. The log-SP algorithm presents around 1 dB gain over the MS algorithm for $BER = 10^{-3}$.

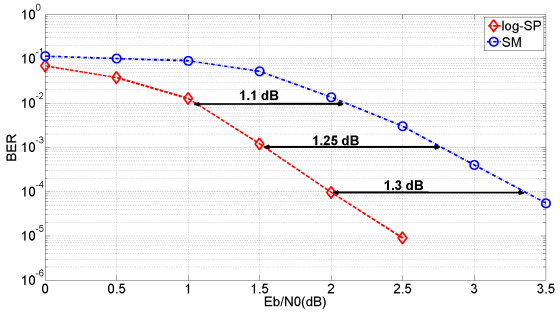

Fig. 9 (4000, 2000) IS-LDPC codes with $\mathbf{I}_{100}$.

Fig. 9 presents the performance curves for the IS-LDPC code using an identity matrix of dimension 100. The algorithm log-SP performs 1.3 dB, in terms of E_b/N_0, better than the MS algorithm, for $BER = 10^{-4}$.

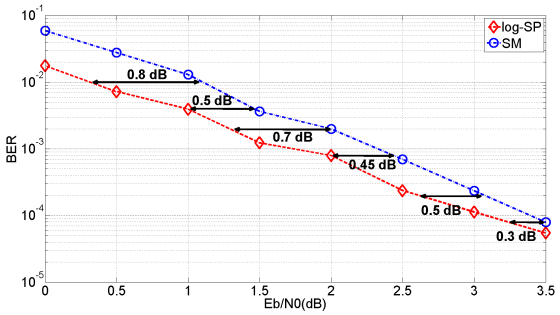

Fig. 10 (4000, 2000) IS-LDPC codes with $\mathbf{I}_{200}$.

Fig. 10 shows the performance of a IS-LDPC code built using an identity matrix with dimension 200. The log-SP algorithm presents 0.3 dB gain over the MS algorithm, for $BER = 10^{-4}$.

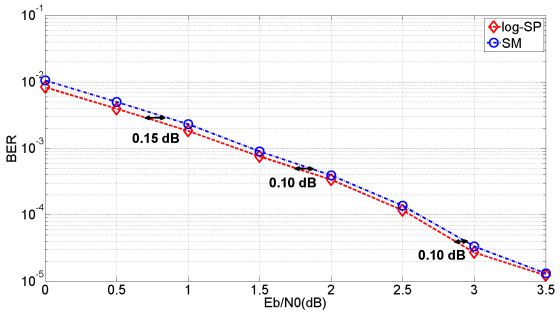

Fig. 11 (4000, 2000) IS-LDPC codes with $\mathbf{I}_{500}$.

Fig. 11 presents an IS-LDPC code built with an identity matrix of dimension 500. The difference in performance was 0.1 dB between log-SP and MS algorithms, for $BER = 10^{-4}$.

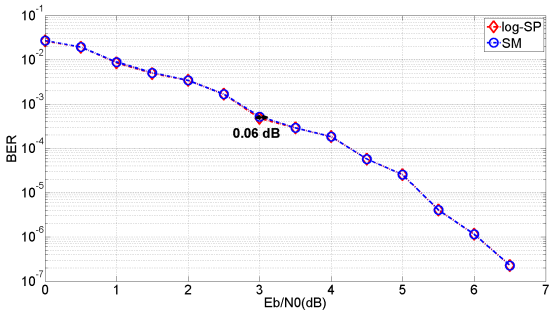

Fig. 12 (4000, 2000) IS-LDPC codes with $\mathbf{I}_{1000}$.

Finally, fig. 12 presents an IS-LDPC code built with an identity matrix of dimension equal to 1000. The decoding performances for both algorithms are coincident.

Again, the performance gap between the log-SP and MS decoding algorithm narrows when the dimension of the parent identity matrix increases. Therefore, the (4000, 2000) IS LDPC codes using parent identity matrixes with dimension 500 and 1000 present the lowest VHDL implementation complexity.

V. Conclusion

The class of IS-LDPC codes can be generated in an easy and direct way. Moreover, its parity check matrix structure reduces the iterative decoding complexity and as consequence it reduces also the VHDL implementation complexity.

As expected the sum-product decoding algorithm performs always better than the min-sum algorithm. However, a decrease in performance around 0.2 dB by using the MS decoding algorithm instead of log-SP algorithm is very reasonable, because the MS algorithm provides a significant reduction in the number of logical elements in FPGA and VHDL implementations.

For the (2000, 1000) IS-LDPC codes, the code built with the parent identity matrix of dimension 500 is under the 0.2 dB log-SP/MS performance threshold. On the other hand, for the (4000, 2000) IS-LDPC codes, the codes built with parent identity matrix of dimension 500 and 1000 have very close performance for both decoding algorithms and they are also under the 0.2 dB threshold. Therefore, those codes present lower complexity for VHDL implementation. Notice that the decoding performances between log-SP and MS algorithms are very close for codes with the parent identity matrix with high dimension.

Acknowledgements

This work was partially supported by FAPESP under contract nº. 2012/01789-4.

References

[1] Bo Yuan, Li Li and Zhongfeng Wang, Efficient Forward Error Correction Decoder Design for High-Speed Optical Networking, Instech, chapter 11, pp. 267 – 288.

[2] M. Jobes, A VLSI Architecture and the FPGA Implementation for multi-rate LDPC Decoding, MSc thesis, McMaster University, 2009.

[3] M. Karkooti, Semi-Parallel Architectures for Real-Time LDPC Coding, MSc thesis, Rice University, 2004.

[4] M. M. Mansour, N. Shanbhag, Low Power VLSI Decoder Architectures for LDPC Codes, Proceedings of the 2002 International Symposium on Low Power Electronics and Design, pp. 284 – 289, 2002.

[5] S. Myung, K. Yang, Quasi-Cyclic LDPC Codes for Fast Encoding, IEEE Transactions on Information Theory, Vol. 51, issue: 8, pp. 2894 - 2901, 2005.

[6] I. B. Djordjevic, M. Arabaci, and L. L. Minkov, Next Generation FEC for High-Capacity Communication in Optical Transport Networks, Journal of Lightwave Technology, Vol. 27, Issue 16, pp. 3518-3530, Aug. 2009.

[7] M. R. Islam, D. S. Shafiullah, M. M. A. Faisal, I. Rahman Optimized Min-Sum Decoding Algorithm for Low Density Parity Check Codes, International Journal of Advanced Computer Science and Applications, Vol. 2, No. 12, 2011.

[8] W. E. Ryan, *An Introduction to LDPC Codes*, http://tuk88.free.fr/LDPC/ldpcchap.pdf

[9] B. S. Nugroho, *LDPC code using MATLAB and C MEX*, http://sites.google.com/site/bsnugroho/ldpc.