

# Reduced Complexity of Decoding Algorithm for Irregular LDPC Codes Using a Split Row Method

Rachid El Alami\*, Mostafa Mrabti, Cheikh Bamba Gueye

LESSI Laboratory, Department of Physics, Faculty of Sciences Dhar El Mehraz, Fez, 35000, Morocco

**Abstract** In this paper, we have proposed a novel method of decoding algorithm of irregular LDPC codes. A reduced complexity LDPC decoding method for regular LDPC code is extended to irregular LDPC codes. We present in this paper a full description of this method and its benefits for various row weight and length code word. The Split-Row method makes column processing parallelism easier to exploit and significantly simplifies row processors. Recently, irregular LDPC codes have received a lot of attention by many advanced standard, such as WiFi, WiMAX Mobile and digital video broadcasting (DVB-S2). Hence the idea to develop the “Split-Row Method” for irregular LDPC codes. In this context, we have performed an implementation on MATLAB of an irregular LDPC codes with different code word and code rate; simulation results over an additive white Gaussian channel show that the error performance of high row-weight codes with Split-Row decoding is within 0.3–0.5 dB of the Min-Sum algorithm. The study result shows that the “Split Row Method” is better for irregular code than regular LDPC codes.

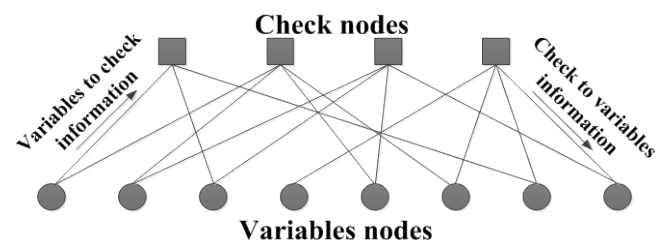
**Keywords** Low-Density-Parity-Check Codes, Split Row, Min Sum, Bit Error Rate

## 1. Introduction

Low density parity check (LDPC) codes are a class of linear block codes which were first introduced by Gallager in 1963[1]. Recently, LDPC codes have received a lot of attention because their error performance is very close to the Shannon limit when decoded using iterative methods[2]. They have emerged as a viable option for forward error correction (FEC) systems and have been adopted by many advanced standards, such as 10 Gigabit Ethernet (10GBASE-T)[3][4] and digital video broadcasting (DVB-S2)[5][6]. Also the next generations of WiFi and WiMAX are considering LDPC codes as part of their error correction systems[7][8].

In the present paper, the previously Split Row method for decoding regular Low-Density Parity-Check (LDPC) codes[9], is extended to irregular LDPC codes; a reduced complexity decoding method which splits each row module into two nearly-independent simplified portions. This method reduces the wire interconnect complexity between row and column processors and increases parallelism in the row processing stage. The Split-Row method also simplifies row processors which results in an overall smaller decoder; Defined as the null space of a very sparse  $M \times N$  parity check matrix  $H$ , an LDPC code is typically represented by a

bipartite graph, called Tanner graph, in which one set of  $N$  variable nodes corresponds to the set of code word, another set of  $M$  check nodes corresponds to the set of parity check constraints and each edge corresponds to a non-zero entry in the parity check matrix  $H$ . An LDPC code is known as  $(j,k)$  regular LDPC code if each column and each row in its parity check matrix have  $j$  and  $k$  non-zero entries, respectively. The construction of LDPC code is typically random. As illustrated in Figure 1, LDPC code is decoded by the iterative belief propagation(BP) algorithm[10] that directly matches its Tanner graph[11].



**Figure 1.** Tanner graph representation of a LDPC code and the decoding message flow

The paper is organized as follows: we begin in Section 2 by reviewing the standard iterative decoding of LDPC codes[12]. In section 3, we present the derived Split Row algorithm for irregular codes and discuss possible simplifications. Finally, in Section 4, we show simulation results and discuss the benefit of the approximation method.

## 2. Standard Iterative Decoding of LDPC Codes

\* Corresponding author:

rachid.elalami@usmba.ac.ma (Rachid El Alami)

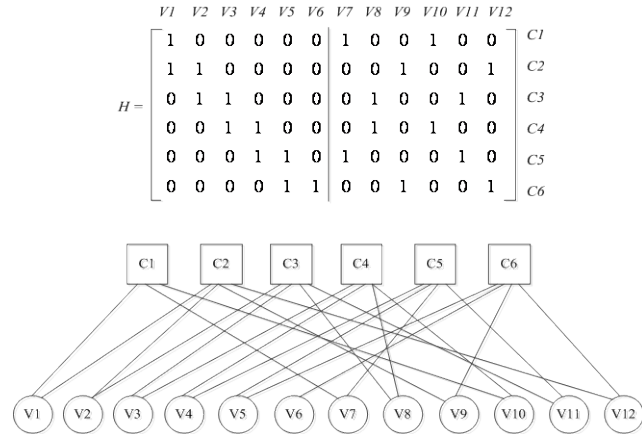
Published online at <http://journal.sapub.org/jwnc>

Copyright © 2012 Scientific & Academic Publishing. All Rights Reserved

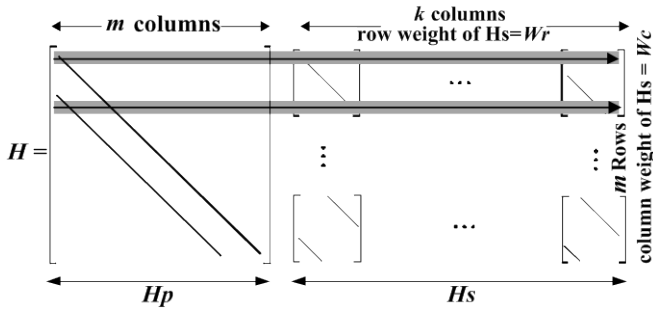
LDPC codes are defined by an  $M \times N$  binary matrix called the parity check matrix  $H$ . The number of columns represented by  $N$  defines the code length. The number of rows in  $H$ , represented by  $M$ , defines the number of parity check equations for the code. Column weight  $W_c$  is the number of ones per column and row weight  $W_r$  is the number of ones per row.

LDPC codes can also be described by a bipartite graph or Tanner graph[11]. The parity check matrix and corresponding Tanner graph of an LDPC code with code length  $N=12$  bits are shown in Figure 2. Each check node  $C_i$  corresponding to row  $i$  in  $H$  is connected to variable node  $V_j$  corresponding to column  $j$  in  $H$ . LDPC codes can be iteratively decoded in different ways depending on the complexity and error performance requirements.

Sum-Product(SP)[10] is near-optimum decoding algorithms which are widely used in LDPC decoders and is known as standard decoders. This algorithm perform row and column operations iteratively using two types of messages: check node message  $\alpha$  and variable node message  $\beta$ . The parity check matrix and the block diagram of the standard decoding algorithm are shown in Figure 3 and Figure 4 respectively.



**Figure 2.** Parity check matrix and Tanner graph representation of an irregular LDPC code with code length  $N = 12$  bits. Check node  $C_i$  represents a parity check constraint in row  $i$  and variable node  $V_j$  represents bit  $j$  in the code

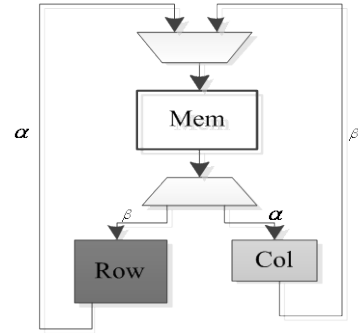


**Figure 3.** Parity check matrix of an irregular LDPC code with code length  $N$ , highlighting row processing operations using standard decoding

The entire  $H$  matrix, for an irregular LDPC code is composed of two sub matrix; the systematic sub matrix  $H_s$  and the parity one  $H_p$ , Figure 3 and Figure 4. The sub matrix  $H_p$

is that adopted by WiMax Mobile[13].

LDPC codes are commonly decoded by an iterative message passing algorithm which consists of two sequential operations: row processing or check node update and column processing or variable node update. In row processing, all check nodes receive messages from neighboring variable nodes, perform parity check operations and send the results back to neighboring variable nodes. The variable nodes update. In row processing, all check nodes receive messages from neighboring variable nodes, perform parity check operations and send the results back to neighboring variable nodes. The variable nodes update soft information associated with the decoded bits using information from check nodes, and then send the updates back to the check nodes, and this process continues iteratively.



**Figure 4.** Block diagram of a typical standard decoder

## 2.1. Sum- Product Decoding

We assume a binary code word  $(x_1, x_2, \dots, x_N)$  is transmitted using a binary phase-shift keying (BPSK) modulation. Then the sequence is transmitted over an additive white Gaussian noise (AWGN) channel and the received symbol is  $(y_1, y_2, \dots, y_N)$ .

We define  $V(i) = \{j : H_{ij} = 1\}$  as the set of variable nodes which participate in check equation  $i$ .  $C(j) = \{i : H_{ij} = 1\}$  denotes the set of check nodes which participate in the variable node  $j$  update. Also  $V(i) \setminus j$  denotes all variable nodes in  $V(i)$  except node  $j$ .  $C(j) \setminus i$  denotes all check nodes in  $C(j)$  except node  $i$ . Moreover, we define the following variables which are used throughout this paper.

$\lambda_j$ : is defined as the information derived from the log-likelihood ratio of received symbol  $y_i$ ,

$$\lambda_i = \ln \left( \frac{P(x_i = 0 | y_i)}{P(x_i = 1 | y_i)} \right) \quad (1)$$

$\alpha_{ij}$ : is the message from check node  $i$  to variable node  $j$ . This is the row processing output.

$\beta_{ij}$ : is the message from variable node  $j$  to check node  $i$ . This is the column processing output.

SPA decoding can be summarized in these four steps:

1. *Initialization*: For each  $i$  and  $j$ , initialize  $\beta_{ij}$  to the value of the log-likelihood ratio of the received symbol  $y_j$ , which is  $\lambda_j$ . During each iteration,  $\alpha$  and  $\beta$  messages are computed and exchanged between variable nodes and check nodes through

the graph edges according to the following steps numbered 2–4.

2. *Row processing or check node update:* Compute  $\alpha_{ij}$  messages using  $\beta$  messages from all other variable nodes connected to check node  $C_i$ , excluding the  $\beta$  information from  $V_j$ :

$$\alpha_{ij} = \prod_{j' \in V(i) \setminus j} \text{sign}(\beta_{ij'}) \times \phi \left( \sum_{j' \in V(i) \setminus j} \phi(|\beta_{ij'}|) \right) \quad (2)$$

Where the non-linear function

$$\phi(x) = -\log \left( \tanh \frac{|x|}{2} \right) \quad (3)$$

The first product term in Equation (2) is called the parity (sign) update and the second product term is the reliability (magnitude) update.

3. *Column processing or variable node update:* Compute  $\beta_{ij}$  messages using channel information ( $\lambda_j$ ) and incoming  $\alpha$  messages from all other check nodes connected to variable node  $V_j$ , excluding check node  $C_i$ .

$$\beta_{ij} = \lambda_j + \sum_{i' \in C(j) \setminus i} \alpha_{i'j} \quad (4)$$

4. *Syndrome check and early termination:* When column processing is finished, every bit in column  $j$  is updated by adding the channel information ( $\lambda_j$ ) and  $\alpha$  messages from neighboring check nodes.

$$z_j = \lambda_j + \sum_{i' \in C(j)} \alpha_{i'j} \quad (5)$$

From the updated vector, an estimated code vector  $X = \{x_1, x_2, \dots, x_N\}$  is calculated by:

$$x_i = \begin{cases} 1, & \text{if } z_i \leq 0 \\ 0, & \text{if } z_i > 0 \end{cases} \quad (6)$$

If  $H \cdot X^T = 0$ , then  $X$  is a valid code word and therefore the iterative process has converged and decoding stops. Otherwise the decoding repeats from step 2 until a valid code word is obtained or the number of iterations reaches a maximum number,  $I_{max}$ , which terminates the decoding process.

## 2.2. Min – Sum Decoding

The check node or row processing stage of Sum – Product decoding can be simplified by approximating the magnitude computation in Equation (2) with a minimum function. The algorithm using this approximation is called Min-Sum (MS)[14,15]:

$$\alpha_{ij} = \prod_{j' \in V(i) \setminus j} \text{sign}(\beta_{ij'}) \times \min_{j' \in V(i) \setminus j} (|\beta_{ij'}|) \quad (7)$$

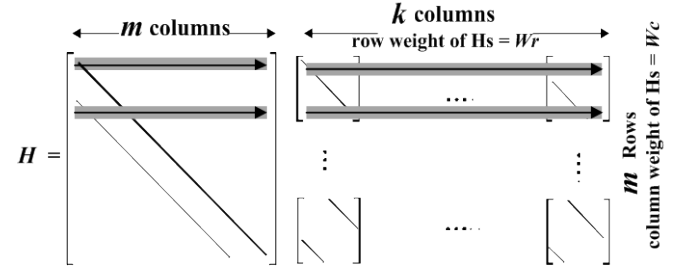
In MS decoding, the column operation is the same as in Sum – Product decoding. The error performance loss of MS decoding can be improved by scaling the check ( $\alpha$ ) values in Equation (7) with a scale factor which normalizes the approximations[16,17].

This scale factor is different for different code rates and can be obtained experimentally.

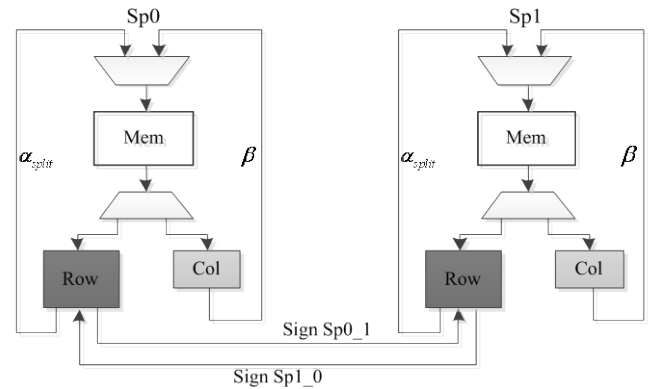
$$\alpha_{ij} = s \times \prod_{j' \in V(i) \setminus j} \text{sign}(\beta_{ij'}) \times \min_{j' \in V(i) \setminus j} (|\beta_{ij'}|) \quad (8)$$

## 3. Split – Row Method Decoding Algorithm for Irregular Codes

The parity check matrix for the row processing stage of the proposed algorithm is shown in Figure 5. As shown in the figure, the row processing stage is divided into two independent halves. This architecture has three major benefits: 1) it doubles parallelism in the row processing stage; 2) it decreases the number of memory accesses per row processor; 3) it makes each row processor simpler. These three factors combine to make row processors (and therefore the entire LDPC decoder) smaller, faster, and more energy efficient. In addition, the Split-Row method makes parallelism in the column processing stage easier to exploit. To reduce performance loss due to errors from this simplification, the sign computed from each row processor is passed to its corresponding “half processor” with a single wire in each direction these are the only wires between the two halves. A block diagram of the Split-Row decoder with two memory blocks is shown in Figure 6.



**Figure 5.** Parity check matrix highlighting row processing operation with the Split-Row algorithm. For simplicity, a Quasi-Cyclic structure is shown for the systematic sub matrix  $H_s$ .



**Figure 6.** Block diagram of the proposed Split-Row decoder

From a mathematical point of view, all steps are similar to the SP algorithm except the row processing step. In each half of the Split-Row decoder's row operation, the parity (sign) bit update is the same as in the SP algorithm. The magnitude part is updated using half of the messages in each row of the parity check matrix. We denote the parity check matrix  $H$  divided into half column wise by  $H_{Split}$ .  $V_{Split}(i) = \{j: H_{ijSplit} = 1\}$

denotes the set of variable nodes in each half of the parity check matrix which participates in check equation  $i$ . Therefore, modifying Equation (2) using half of the messages yields:

$$\alpha_{ijSplit} = \prod_{j' \in V(i)j} \text{sign}(\beta_{ij'}) \times \phi \left( \sum_{j' \in V_{Split}(i)j} \phi(|\beta_{ij'}|) \right) \quad (9)$$

As  $\phi(x)$  is a positive function then the sum of a portion of the positive values less than or equal to all:

$$\sum_{j' \in V_{Split}(i)j} \phi(|\beta_{ij'}|) \leq \sum_{j' \in V(i)j} \phi(|\beta_{ij'}|) \quad (10)$$

Also  $\phi(x)$  is a decreasing function, therefore the following inequality holds:

$$\phi \left( \sum_{j' \in V_{Split}(i)j} \phi(|\beta_{ij'}|) \right) \geq \phi \left( \sum_{j' \in V(i)j} \phi(|\beta_{ij'}|) \right) \quad (11)$$

Returning to the computation of  $\alpha$  in Equation (2) and Equation (9), we obtain:

$$|\alpha_{ijSplit}| \geq |\alpha_{ij}| \quad (12)$$

By normalizing the  $\alpha_{ijSplit}$  values with a scale factor  $S$  less than one we can improve the error performance of the Split-Row algorithm.

$$\alpha_{ijSplit} = S \times \prod_{j' \in V(i)j} \text{sign}(\beta_{ij'}) \times \phi \left( \sum_{j' \in V_{Split}(i)j} \phi(|\beta_{ij'}|) \right) \quad (13)$$

Application of the Split-Row method to Min-Sum is equally viable. Similar to the Min-Sum algorithm, the optimal value for  $S$  varies over different code rates and SNR values and can be obtained experimentally.

## 4. Error Performance Simulation Results

To evaluate the extended method, Min Sum Split Row for irregular LDPC codes, we performed simulations assuming binary phase-shift keying (BPSK) modulation and transmission over the additive white Gaussian noise (AWGN) channel. These simulations have been carried out using MATLAB. For each simulation, a curve showing the bit-error rate (BER) versus  $E_b/N_0$  was computed. The  $E_b$  stands for energy per bit and the  $N_0$  stands for the noise power spectral density ratio.

The following labeling is used for the figures: “MS” for normalized Min-Sum, “MS Split-Row” for the extended method Min-Sum Split-Row algorithm and “S” for the scaling factor.

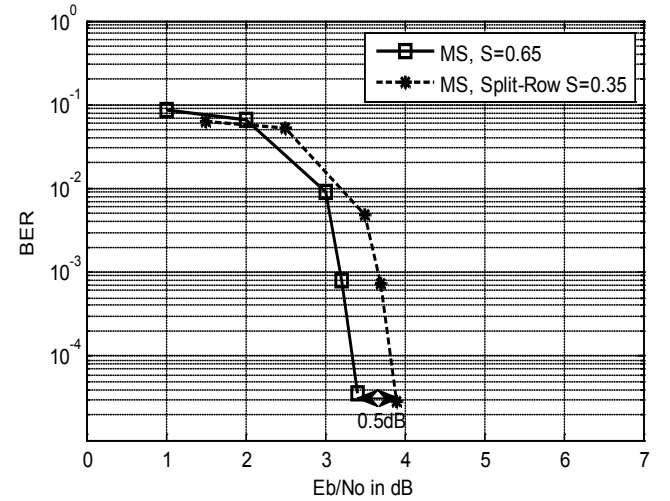
The performances of irregular LDPC codes are illustrated in the figures given below. Simulations were run to determine the performance of these LDPC in AWGN channel with BPSK modulation. In each case the proposed decoding algorithm was used. The decoder stops when a non-valid code word is found.

Figure 7 shows the performance of the LDPC code with input frame size of 1536 bits and (6,18) as weights for the systematic sub matrix  $H_s$ . The code rate of this LDPC code is  $R=0.75$ . The max number of iterations is set to  $I_{max}=15$ . We fix the threshold  $I_{max}$  to satisfy a tradeoff between the performance correction and the speed of the decoder.

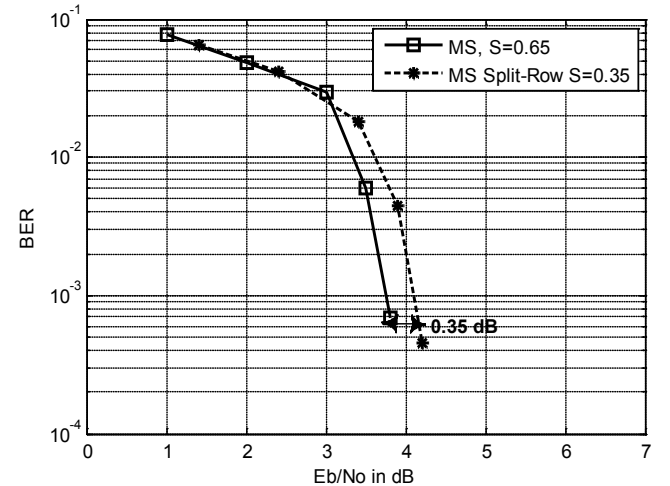
In the Fig. 8, the bit error performance of LDPC code for an input frame size of 1520 bits and a (6,32) for the weights of  $H_s$ . The code rate is  $R=0.84$  and the maximum number of iteration is set to  $I_{max}=15$ .

Figure 9 depicts the bit error performance of LDPC code for an input frame size of 2014 bits and a (6,32) for the weights of  $H_s$ . The code rate is  $R=0.84$  and the maximum number of iteration is set to  $I_{max}=7$ .

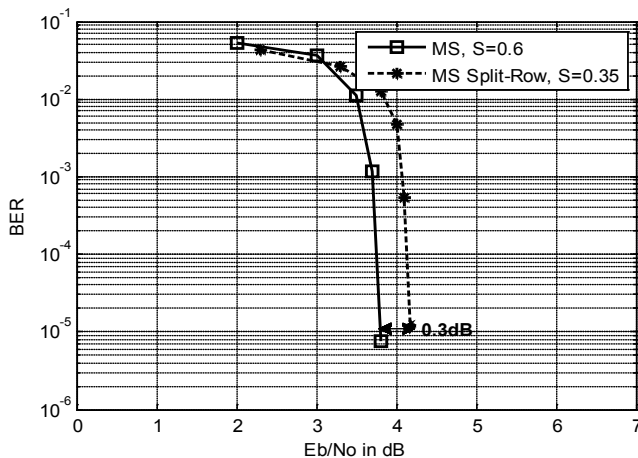
From the conducted simulation study shown on Figure 7 and Figure 8 we can conclude that for a high row weight the performance gap between the standard Min Sum and the Min Sum Split Row is better than a low row weight; for example, in Figure 7, the gap is about 0.5dB while it is about 0.35dB in Figure 8.



**Figure 7.** Error performance of the Split-Row decoder for irregular LDPC code,  $N=1536$  bits



**Figure 8.** Error performance of the Split-Row decoder for irregular LDPC code with code length  $N=1520$  bits



**Figure 9.** Error performance of the Split-Row decoder for irregular LDPC code with code length  $N=2014$  bits

And from the simulations results shown on the three figures, we can conclude that for a high code length and a high row weight the performance of the Split Row decoder is better.

## 5. Conclusions

In this contribution, we have extended the previous version the Split Row method developed for regular LDPC codes[9] to Irregular Codes. Simulation results show that the Split Row method is better for irregular code than regular LDPC codes while maintaining the same level of complexity.

Future work will focus on implementation of the architecture of the present method of decoding algorithm for irregular LDPC codes. It consists also, to improve the proposed architecture of the decoder in order to achieve the best performance in terms of speed, throughput and flexibility.

From the architecture of the present method mentioned in this paper; we have proposed a "Multi-Split-Row"[18] LDPC decoding method which allows further reductions in routing complexity, it also allows to have a faster decoder, compared to the actually proposed Split-Row decoding method presented in this paper.

## ACKNOWLEDGEMENTS

The authors thank Mostafa Mrabti have help us achieve this works and very helpful discussions; and gratefully acknowledge support from Maroc Telecom and CNRST.

## REFERENCES

[1] Robert. G. Gallager, "Low-density parity check codes," IRE Transaction Info.Theory, vol. IT-8, pp. 21–28, Jan. 1962.  
[2] David John Cameron MacKay, "Good Error-Correcting

Codes Based on Very Sparse Matrices," IEEE Transactions on Information Theory, vol.45, pp. 399–431, Mar.1999.  
[3] "IEEE P802.3an, 10GBASE-T Task Force," <http://www.ieee802.org/3/an>.  
[4] Cohen, A.E., Parhi, K.K, "A Low-Complexity Hybrid LDPC Code Encoder for IEEE 802.3an (10GBase-T) Ethernet," in IEEE Transactions on Signal Processing Vol. 57, Issue 10, pp. 4085-4094, 2009.  
[5] "T.T.S.I. Digital Video Broadcasting (DVB) Second Generation Framing Structure for Broadband Satellite Applications," <http://www.dvb.Org>.  
[6] Kim, S.-M., Park, C.-S., Hwang, S.-Y, "A Novel Partially Parallel Architecture for High-Throughput LDPC Decoder for DVB-S2," IEEE Transactions on Consumer Electronics Vol. 56, Issue 2, pp. 820-825, 2010.  
[7] Wang, Y.-L., Ueng, Y.-L., Peng, C.-L., Yang, C.-J, "A Low-Complexity LDPC Decoder Architecture for WiMAX Applications," in *Proc. Int. Symp on VLSI Design, Automation and Test*, Hsinchu, April 2011.  
[8] Shin, K.-W. , Kim, H.-J, "A Multi-Mode LDPC Decoder for IEEE 802.16e Mobile WiMAX", *Journal of Semiconductor Technology and Science*, Vol. 12, Issue 1, pp 24-33, Mar. 2012.  
[9] Tinoosh. Mohsenin and Bevan Baas, "Split-Row: a Reduced Complexity, High Throughput LDPC Decoder Architecture," in *Proc. ICCD*, Oct. 2006.  
[10] David John Cameron MacKay and Radford. R Neal, "Near Shannon Limit Performance of Low Density Parity-Check Codes," *Electronic Letter*, Vol. 32, Issue 18, pp. 1945-1946, August 1996.  
[11] M. R. Tanner, "A Recursive Approach to Low Complexity Codes," IEEE Transactions on Information Theory, vol. 27, September 1981.  
[12] Rachid El Alami, Cheikh Bamba Gueye, Mohamed Boussetta, Mostafa Mrabti and Mohcine Zouak, "Bit Flipping - Sum Product Algorithm for Regular LDPC Codes" in *Proc. Int. Symp on I/V Communications and Mobile Networks*, Rabat, Morocco, Oct. 2010.  
[13] Teodor B. Iliev, Georgi V. Hristov, Plamen Z. Zahariev, Mihail P. Iliev, "Application and Evaluation of the LDPC Codes for the Next Generation Communication Systems", *Novel Algorithms and Techniques In Telecommunications, Automation and Industrial Electronics*, pp. 532-536, 2008.  
[14] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of block and convolutional codes," IEEE Transaction of Information Theory, vol. 42, pp. 429–445, Mar. 1996.  
[15] M. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," IEEE Transaction Communications, vol. 47, pp. 673–680, May 1999.  
[16] J. Chen and M. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check codes," IEEE Transaction Communications, vol. 50, pp. 406–414, Mar. 2002.  
[17] J. Chen, A. Dholakia, E. Eleftheriou, and M. Fossorier, "Reduced complexity decoding of LDPC codes," IEEE

Transaction Communications, vol. 53, pp. 1288–1299, Aug. 2005.

- [18] Rachid El Alami, Mostafa Mrabti, “Reduced Complexity of

Decoding Algorithm for Irregular LDPC Codes Using a Multiple Split Row Method,” International Journal of Research and Reviews in Mechatronic Design and Simulation, vol. 1, pp. 12-17, March 2011.