Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **3**, May **2018**

Piecewise Linear Approximation Based on Taylor Series of LDPC Codes Decoding Algorithm and Implemented in FPGA

Zhong-Xun Wang, Yan Wang

Institute of Science and Technology for Opto-Electronics Information
Yantai University, Shandong, China
ytwzx3@126.com, wangy0535@163.com

Received July, 2016; revised January, 2018

Abstract. *In order to make the LDPC (Low Density Parity Check) codes more suitable for FPGA design. During the renewal process of check nodes, uses piecewise linear approximation based on Taylor series to replace non-linear logarithmic function in core-add operation. At the same time, select the reasonable segments to reduce the decoding complexity. The simulation results show that, use the simplified algorithm to design the decoder of FPGA process a certain practicalities, not only can it reduce the decoding complexity but also it can improve the decoding accuracy.*
Keywords: LDPC, Log-SPA, Piecewise Linear Approximation, Taylor Series, FPGA

1. **Introduction.** At present, widespread use of decoding algorithms of LDPC codes are: Log-SPA (Sum-Product Algorithm in Log-Likelihood-domain) algorithm, NMS (Normalized Min-Sum) algorithm, OMS (Offset Min-Sum) algorithm and so on [1, 2, 3, 4, 5, 6, 7]. All decoding algorithms listed can provide higher decoding accuracy. However, one part of Log-SPA algorithm is the non-linear logarithmic functions in core add operation. The correction factors in NMS decoding algorithms and the offset parameters in OMS decoding algorithms need to be set according to the practical situation. All these problems not only increase the complexity of decoding but also increase the difficulty of hardware implementation. A simplified decoding algorithm has been proposed based on the first-term McLaughlin series in [8]. This method could reduce the decoding complexity effectively at the expense of the decoding accuracy and add the correction factors which cannot adaptive to make up this part of accuracy.

In this paper a simplified algorithm is proposed, the proposed algorithm uses Taylor series to replace non-linear logarithmic function in core-add operation of Log-SPA algorithm. During the process of check nodes, we conduct a detailed analysis on the number of segments in the linear approximation. Thus, the complexity of decoding algorithm can be reduced by the reasonable selection of segments. At last, design the FPGA decoder by the simplified algorithm to verify the correction of the algorithm. The designed decoder processes certain practicalities, it makes a balance on decoding complexity and accuracy and resource consumption.

2. **The Decoding Algorithm of LDPC Codes.** In the decoding algorithm of Log-SPA, the processing procedure of check nodes and variable nodes adopt the algorithm

of forward-backward [9, 10]. The recursive algorithm of forward-backward is shown in Figure 1.

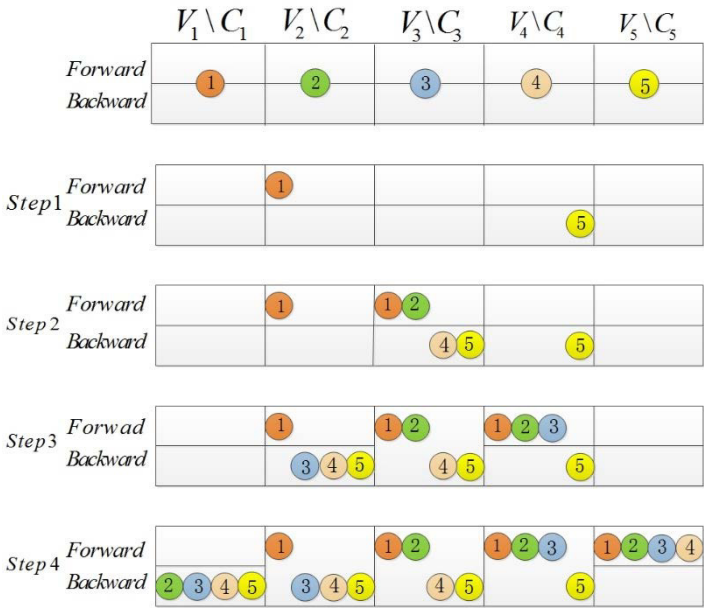

FIGURE 1. The process of forward-backward algorithm

For example, one check node connected to five variable nodes in Figure 1. In the recursive computation of forward-backward algorithm, forward operating and backward operating calculate simultaneously. Then, the operation result of each calculate will be used with the next variable nodes to continue the recursive operations (such as in Figure 1, at first, node 1 and node 2 conduct recursive computing, and then use their operation result calculate with node 3 by recursive operations). In step 4, after forward and backward operations are calculated respectively, each variable node will continue recursive operations with their corresponding calculated results.

Definition $L(U \oplus V) = 2\tanh^{-1}(\tanh(L(U))\tanh(L(V)))$, then the updating information of check nodes can convert into:

$$\begin{aligned} L^{(l)}(r_{ji}) &= 2\tanh^{-1}\left(\prod_{i' \in R_{j\backslash i}} \tanh\frac{L^{(l-1)}(q_{i'j})}{2}\right) \\ &= \sum_{i' \in R_{j\backslash i}} \oplus L(q_{i'j}) \end{aligned} \tag{1}$$

Where $\oplus$ is defined as core-add operation. The calculation formula is shown in the following equation:

$$\begin{aligned} L(U \oplus V) &= \log(1 + e^{(L(U)+L(V))}) - \log(e^{L(U)} + e^{L(V)}) \\ &= sign(L(U))sign(L(V))\min(|L(U)|, |L(V)|) \\ &\quad + \log(1 + e^{-|L(U)+L(V)|}) - \log(1 + e^{-|L(U)-L(V)|}) \end{aligned} \tag{2}$$

Where U and V are two statistically independent binary random variables with log likelihood ratio values of $L(U)$ and $L(V)$ respectively.

2.1. **The Decoding Algorithm of Log-SPA Based On Taylor Series.** These two non-linear logarithmic functions in equation (2) are also be called Jacobi correction term. The introduction of Jacobi correction item increase calculated amount in the process

of decoding, so we need further simplification to the equation of (2). Supposing that using Taylor series expansion and omitting greater orders than one to approximate the non-linear logarithmic function. x_0 represents tangency point of $g(x)$ and the optimal segmentation step length ($\Delta d = 0.75$) can be obtained by experiment. The function of non-linear logarithmic can be expressed by $g(x)$, as following:

$$\begin{aligned} g(x) &= \log(1 + e^{-|x|}) \\ &\approx g(x_0) + (x - x_0)g'(x) \\ &\approx \max\{0, g(x_0) + (x - x_0)g'(x)\} \end{aligned} \tag{3}$$

In order to obtain the best comprehensive decoding performance, it is necessary to select the reasonable value of segments λ. Thus, the complexity of piecewise linear approximation decoding algorithm based on Taylor series could be further reduced.

2.2. **The Analysis of Decoding Complexity.** The realization of core-add operation of $L(U \oplus V)$ is based on forward-backward algorithm. For regular LDPC codes, each check node is connected to d_c variable nodes. Row degree d_c is a fixed value. Each updates of the check node needs $3(d_c - 2)$ operations to calculate $L(U \oplus V)$. The piecewise linear approximation algorithm based on Taylor series does not require look-up table operations and non-linear logarithmic operations. The decoding computational complexity of various check nodes of the Log-SPA algorithm, the proposed algorithm based on Taylor series, the first order of McLaughlin series in [8],the Normalized MS algorithm and the Min-Sum algorithm are shown in Table 1.

TABLE 1. Computational complexity of various check nodes (row degree is d_c)

operations	Log-SPA	Taylor	McLaughlin	Normalized MS	Min-Sum
Additions	0	$6(d_c - 2) + (\lambda - 2) \cdot d_c$	$6(d_c - 2)$	0	0
Comparisons	0	$3(d_c - 2) + (\lambda - 2) \cdot d_c$	$3(d_c - 2)$	0	0
Multiplications	0	$6(d_c - 2)$	$6(d_c - 2)$	$d_c - 2 + \log d_c$	$d_c - 2 + \log d_c$
Absolute values	0	$3(d_c - 2)$	$3(d_c - 2)$	d_c	d_c
Table look-ups	$3(d_c - 2)$	0	0	0	0

The linear approximation function can be described by the function of $g(x) = ax + b$. a in the function is a fixed value, it represents the amplitude-scaling factor of x. So we can consider that the amounts of multiplications does not increase, it only increases additions and comparisons. As table1 shows, when d_c is a fixed value, the corresponding computational complexity will have a linear increase with λ. From table 1 the Normalized MS algorithm and the Min-Sum algorithm are shown the lowest decoding complexity effectively. But it's based on the expense of the decoding accuracy and add the correction factors. In table 1 when $\lambda = 4$, compared with McLaughlin approximation algorithm, the proposed algorithm will only increase $2d_c$ in additions and comparisons.The purpose of this article is to make a balance on decoding complexity and accuracy. Thus the proposed algorithm is the wise choice.

2.3. **The Analysis of Decoding Performance.** In Table 1, the complexity of the decoding will increase as the number of segments λ increases. The optimization of decoding algorithm can reduce the complexity of the algorithm and improve decoding accuracy and it's good for hardware implementation. Therefore, it's essential to make a balance on decoding complexity and accuracy and resource consumption in order to obtain the optimal decoding performance.

This experiment is conducted under the simulation environment of MATLAB. The encoded bits are modulated by binary phase shift keying (BPSK) and transmitted over an AWGN (Additive White Gaussian Noise) channel. The block sizes of randomly constructed regular LDPC codes been used is (504, 3, 6) .The maximum iteration is 25. Because of the Normalized MS algorithm (NMS) and the Offset MS algorithm (OMS) have the similar decoding performance, we just chose the Normalized MS algorithm for analysis in this article. The correction factor offor the Normalized MS algorithm is equal to 0.8 in this experiment. The BER performance of the Log-SPA algorithm, the Min-Sum algorithm, the Normalized MS algorithm, the McLaughlin approximation algorithm in [8] and the proposed algorithm (Taylor approximation $\lambda = 4$ and $\lambda = 8$) are shown in Figure 2.

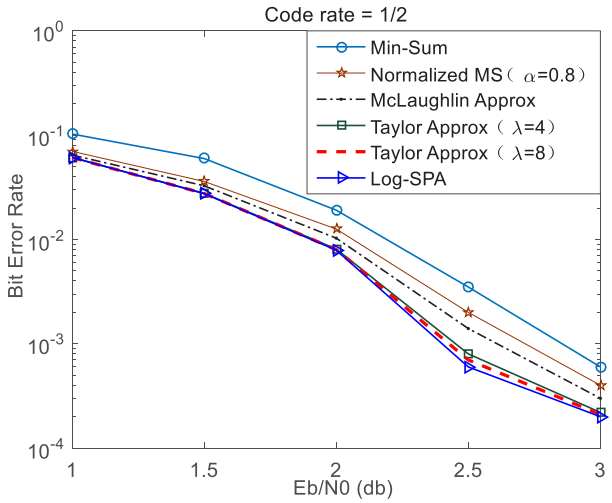

FIGURE 2. BER performance of (504, 3, 6) LDPC code and maximum of 25 iterations

Figure 2 depicts that the proposed algorithm (when $\lambda = 4$ and $\lambda = 8$) is approximately less than 0.1dB inferior to Log-SPA decoding algorithm, achieving essentially optimal Log-SPA decoding performance. From Figure 2, we can see that the proposed algorithm (when $\lambda = 4$ and $\lambda = 8$) outperform 0.1 to 0.2 dB both Normalized MS and McLaughlin approximation which cannot add the correction factors and obtain higher decoding accuracy.

It should be pointed out that the complexity of the decoding will increase as the number of segments λ increases from Table 1. The complexity of decoding algorithm can be reduced by the reasonable selection of segments. Thus, in order to ensure higher decoding accuracy and lower decoding complexity, $\lambda = 4$ is the optimal choice. The proposed algorithm is evidenced with small performance degradation against the optimal Log-SPA decoding algorithm and lower computational complexity (Taylor approximation $\lambda = 4$). So, selecting the piece-wise linear approximation algorithm (Taylor approximation $\lambda = 4$) as the decoding algorithm to design the FPGA decoder in the next section which can make a balance on decoding complexity and accuracy and resource consumption.

3. **Design FPGA Decoder Based On Simplified Decoding Algorithm.** In this paper, the partial-parallel method will be adopted to design the decoder on the platform of FPGA and the chip of 5CSEMA5F31C6 [11] (the series chip of Cyclone V is produced by Altera company) will be used to design the decoder [12, 13]. The block sizes of randomly constructed regular LDPC codes been used is (1024, 3, 6).The maximum iteration is 10.

In the form of (Q, F) to express Q bits fixed-point quantization, using F bits to represent the decimal digits [14, 15].

3.1. **The design of Check Nodes Unit (CNU) module.** The proposed decoding algorithm of LDPC codes in this paper mainly reflects on the processing procedure of check nodes. Using piecewise linear approximation to replace Jacobi correction term in the decoding algorithm of Log-SPA can avoid the non-linear logarithmic function. The schematic diagram of data processing of Jacobi correction term and the description of I/O ports are shown in Figure 3.

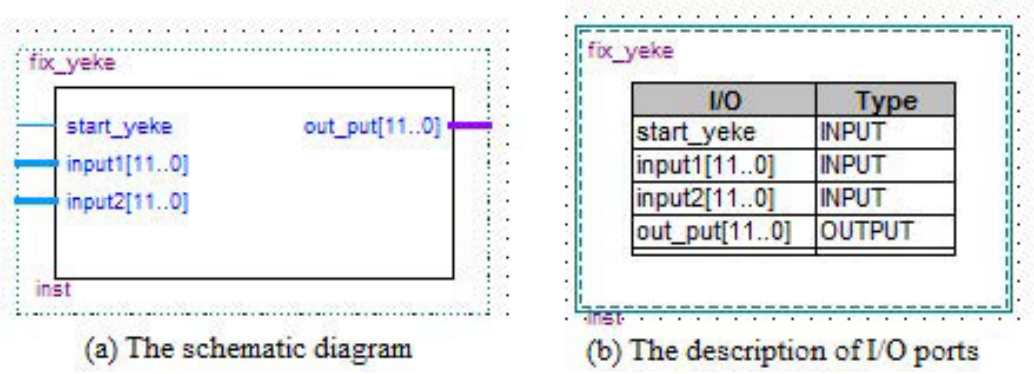

FIGURE 3. The schematic diagram and the description of I/O ports of Jacobi correction term

The fixed-point decimal multiplication will be used to realize piecewise linear approximation in the decoding algorithm and the manner of Q will be adopted to quantify the data, it is the same as each data multiplied by 2^{F}. The input port of start_yeke is an level-enable signal, the input length of input1 and input2 are all $m+\mathrm{F}$. The form of linear approximation function can be described by $g(x) = ax + b$. b is a fixed value obtained by the way of original $b$multiply by 2^{F}, when $\mathrm{F} = 4$, the value of the decimal form is 11. For the value of a, we will use the form of $2^{-\alpha}$ (α is a positive integer) to approximate, the value of a can be obtained by right shifted times. Since the design adopt the regular (1024,3,6) LDPC code, each check node will calculate with five variable nodes by Jacobi operations.

The data in the processing procedure of Check module updates and the timing simulation are shown in Figure 4 and Figure 5 respectively.

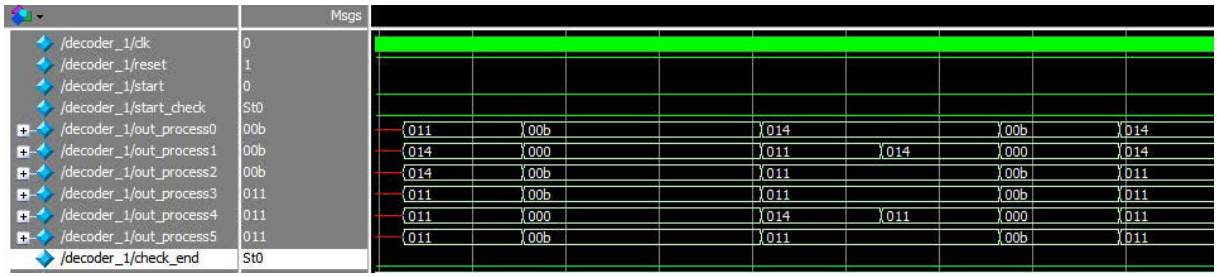

FIGURE 4. Data in the process of check module updating

3.2. **The design of main control module.** During normal running, the decoder is controlled by main control module. The iterative decoding algorithm of LDPC codes needs to update large amount of information, so it is required to determine when to do what. The main control module controls the initialization of the information, the process

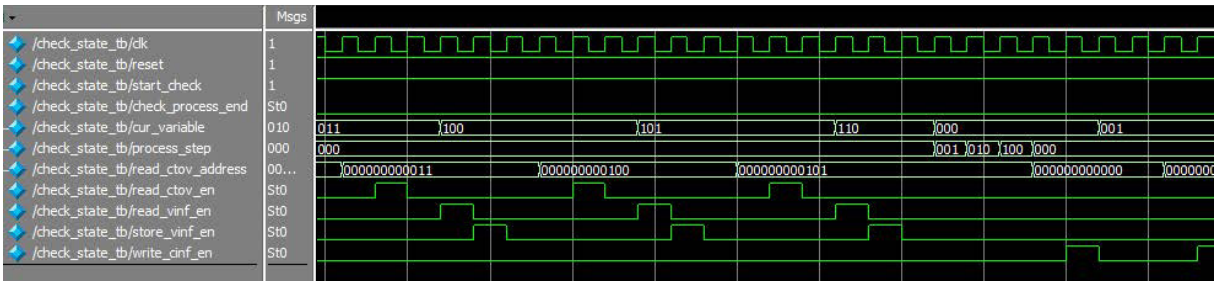

FIGURE 5. Timing simulation of check module

of check nodes and variable nodes and the number of iterations in the decoder. The state transition diagram of main control module is shown in Figure 6.

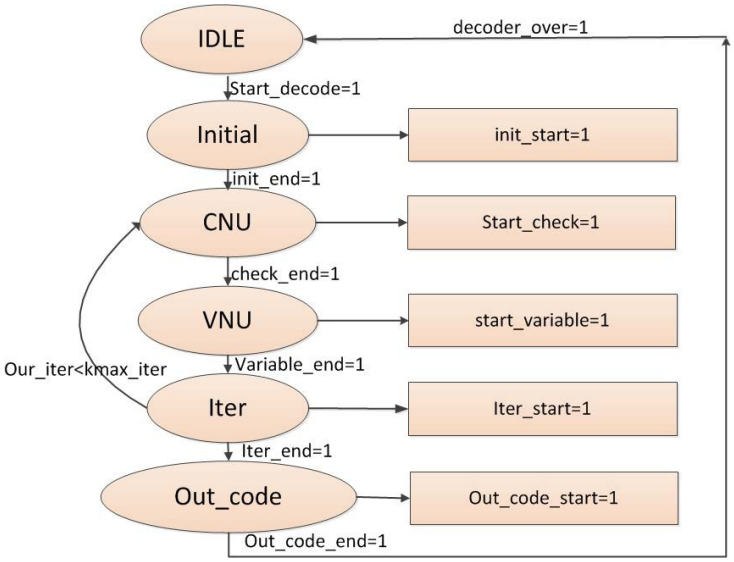

FIGURE 6. State transition diagram of main control module

In Figure 6, the beginning state of the decoder is IDLE, when it receives enable signal of Start_decode = 1 the decoder will enter the state of Initial, and then it starts the initialization process. The state transition diagram of main control module as shown above. At the ending of main control module,it receives enable signal of decoder_over = 1, then, the processing ends.

The timing simulation of main control module is shown in Figure 7.

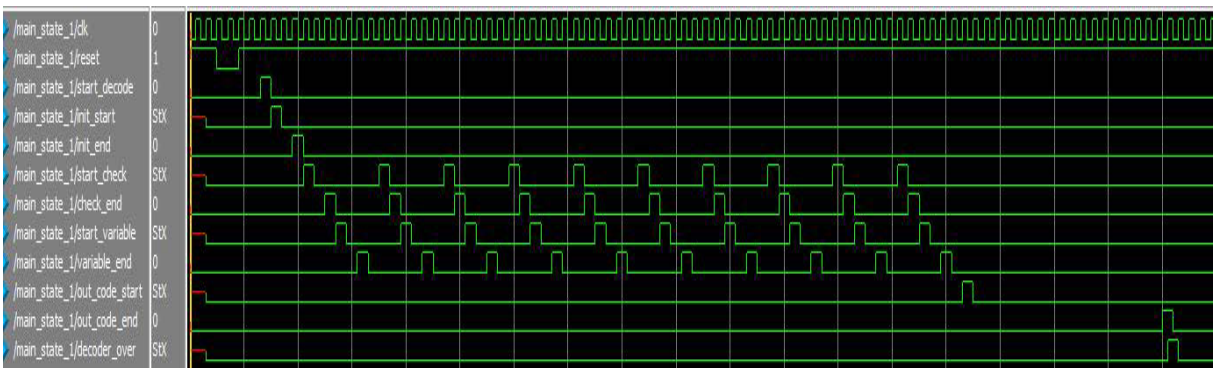

FIGURE 7. Timing simulation of main control module

3.3. **The output code words of the decoder.** When it achieves the maximum iteration, the hard decision information of variable module will transfer to the module of output buffer. Then the decoder receives enable signal of out_code_start = 1, the module of output buffer will read out the code words. When it receives all of code words, the decoder_over = 1 will be set. From Figure 8, the output code word information of the decoder can be observed then. For using (the code length is 1024) long code, it can't display all the code word. The forepart of code word sequences are shown in Figure 8. The output of the code words have a perfect match with encoded information in the MATLAB environment. At the same time it reflects the rationality of the simplified decoding algorithm and the correctness design of the decoder.

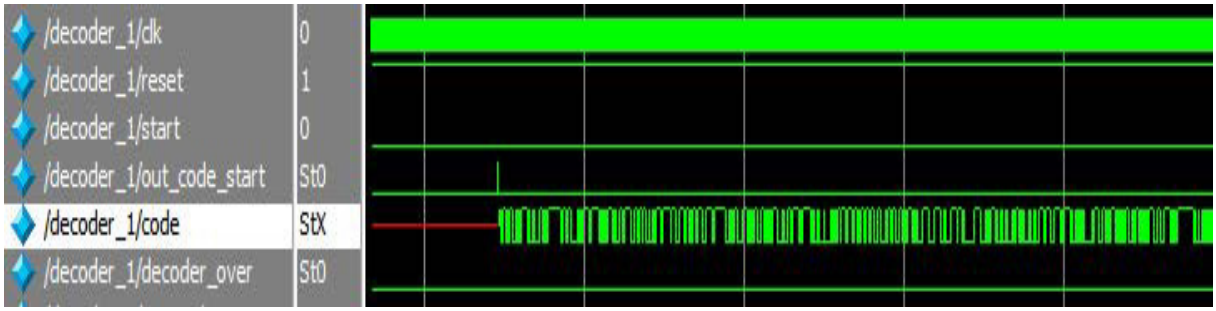

FIGURE 8. The output code words of the decoder

4. **Conclusion.** Based on the chip of 5CSEMA5F31C6 (the series chip of Cyclone V is produced by Altera company), we design and simulate the simplified LDPC decoding algorithm using the platform of quartusII 14.1 and modlesim in this paper. It can be seen from the compile comprehensive report, fewer than 50 MHZ clock frequencies, the resource consumption are as follows: the consumption of unit ALMs is 6%, at the same time, the consumption of unit RAM and ROM is 5%. Because the decoding algorithm in this paper uses lots of simplified Jacobi operations, more resource is needed than the MS decoding algorithm which is easily implemented with hardware, but our decoding algorithm will largely improve the decoding accuracy. Thus, the algorithms in this paper can make a good balance among the decoding complexity, the decoding speed and the decoding accuracy.

Acknowledgment. This work was supported by the scientific and technological projects of Shan dong province (2012J0030009).

REFERENCES

[1] Q. Wu, J. J.Mei. Research on LDPC Decoding Based on Modified Min-sum Algorithm, *Proceedings of the second Radio Communications Technology*, pp. 38-40, 2012.

[2] X.C.Chen D.P.Liu. Modified Decoding Algorithm of LDPC Codes, *Proceedings of the 39th Journal of University of Electronic Science and Technology of China*, pp. 39-42, 2010.

[3] X.Y.Hu, E. Evangelos, A. D. Michael. Efficient Implementations of the Sum-Product Algorithm for Decoding LDPC Codes, *Proceedings of IEEE Global Telecommunications*, pp. 036-1036E, 2001.

[4] J. H. Chen, D. Ajay, E. Evangelos. Reduced-complexity decoding of LDPC Codes, *Proceedings of the 8th IEEE Transactions on Communications* , pp. 1288-1299, 2005.

[5] Pandya. N, Honary B. Low-complexity belief propagation decoding for low-density parity-check codes, *Proceedings of the 43th Electronics Letters*, pp. 990-991, 2007.

[6] C. L. Wang, X. H. Chen, Z. W. Li, S. H. Yang. A Simplified Min-Sum Decoding Algorithm for Non-Binary LDPC Codes. *Proceedings of the first IEEE Transactions on Communications*, pp. 24-32, 2013.

[7] G. Lechner, J. Sayir. Improved Sum-Min Decoding of LDPC Codes, *Proceedings of the Conference on International Symposium on Information Theory and its Applications*, pp. 10-13, 2004.
[8] Papaharalabos S, Mathiopoulos P. T. Simplified Sum-product Algorithm for Decoding LDPC Codes with Optimal Performance. *Proceedings of the 15th Electronics Letters*, pp. 1049-1051, 2009
[9] Y. Xiao. The codec of Turbo and LDPC codes and their applications, *Proceedings of the Posts & Telecom Publishers, Beijing*, pp. 34-36, 2010.
[10] D. F. Yuan, H. G. Zhang, The Theory and Application of LDPC Codes, *Proceedings of the Posts & Telecom Publishers*, pp. 21-24, 2008.
[11] Altera Corporation. All rights reserved. *ALTERA, ARRIA,* December 2013 Available via: http://easydatasheet.cn/ALTERA/5CSEMA5F31C6.
[12] T. Zhang, Parhi K. K. A 54 Mbps (3,6) Regular FPGA LDPC Codes Decoder, *Proceedings of IEEE Workshop about Signal Proeessing Systems*, pp. 1058-1063, 2008.
[13] C. Spanol, W. Marna. Popovici Emanuel, FPGA Implementations of LDPC codes over GF(2) Decoders. *Proceedings of the IEEE Workshop on Signal Processing Systems*, pp. 273-278, 2007.
[14] W. Liang, L. Liu, X Shen, F. Ye, J. Y. Ren, LDPC Decoder Architecture with Dynamic Quantization. *Proceedings of the 37th Computer Engineering and Applications*, pp. 232-234, 2011.
[15] Z. Y. Zhang, L. Dolecek, V. Anantharam. Quantization Effects in Low-Density Parity-Check Decoders, *Proceedings of IEEE International Communications*, pp. 6231-6237, 2007.