FPGA Implementation of LDPC Encoder Architecture for Wireless Communication Standards

Ruslan Goriushkin, Pavel Nikishkin, Aleksei Ovinnikov,
Evgeny Likhobabin and Vladimir Vityazev, *Member*, *IEEE*
Department of Telecommunications and foundations of radio engineering
Ryazan State Radio Engineering University, RSREU
Ryazan, Russia
gorushkin.r.s@tor.rsreu.ru, nikishkin.p.b@tor.rsreu.ru, ovinnikov.a.a@tor.rsreu.ru,
likhobabin.e.a@tor.rsreu.ru, vityazev.v.v@rsreu.ru

***Abstract*—In this paper a pipeline architecture is proposed for FPGA implementation of a quasi-cyclic LDPC (QC-LDPC) encoder.**

The results are provided for implementation on a Xilinx ZYNQ-7 ZC706 Evaluation Board for code with rate 5/6 and block lengths from 576 to 2304. The design is parameterized and can be easily rebuilt to support various code rates and code lengths. The base matrices H and code parameters are taken from the IEEE 802.16e standard.

The number of logic elements (LUT), clock speed, and throughput of the encoder are presented for different code lengths. The throughput of up to 16 Gbps for IEEE 802.16e codes has been achieved.

***Index Terms*—LDPC, QC-LDPC, parallel and configurable architectures, FPGA implementation, encoding.**

I. Introduction

Low-density parity-check (LDPC) codes were invented by Gallager in 1963 [1] and then rediscovered by MacKay and Neal [2] in 1990's. Nowadays LDPC codes have received much attention due to their efficient decoding algorithm, excellent error correcting capability and their performance able to achieve up to 0.0045 dB of the Shannon's limit at a BER of 10^{-6} [3]. These codes are used in many communication standards such as IEEE 802.11n/ad/ay/ax (WiFi) [4], IEEE 802.16e (WiMAX) [5], ETSI 5G [6] and video broadcasting standards such as DVB-T2, DVB-S2, DVB-C2 and also find place in other fields, including error correction techniques for magnetic storage and flash memory. Several FPGA architectures have been proposed for LDPC encoders in the past [7] - [14] and known to be a hot topic nowadays [15] - [17]. Most of the previously proposed architectures did not possess universality properties and supported only one code rate or block size.

Architectures for quasi-cyclic LDPC (QC-LDPC) encoders were also proposed. QC-LDPC codes are used in 802.11 and 802.16e standards and allow to reduce size of memory for parity check matrices storage. The encoder specified in [16] is based on replacing the inverse matrix with back substitution, that allows to increase the encoding speed and reduce the quantity of memory bits required; the maximum throughput of this implementation is 1.2 Gbps. The pipeline encoder architecture proposed in [11] is flexible in terms of both the code rate and code length; this implementation features a maximum throughput of 6.28 Gbps.

But, increasing the data transfer rates requires the development of more efficient encoders and decoders. As an example of the transfer rates required by modern technologies, the IEEE 802.11ax specifies data rate up to 11 Gbps, and IEEE 802.11ay up to 40 Gbps [18].

Therefore, encoders proposed in [11] and [16] can't satisfy the growing needs of modern standards. In this paper we provide the parallel encoder realization for QC-LDPC codes, that meets the requirements of modern specifications for throughput, and has the flexibility to work with various code rates and sizes. In proposed design the parallel encoding is combined with a pipeline structure, that allows us to significantly reduce the latency and increase the throughput. The FPGA implementation details on FPGA ZYNQ-7 ZC706 Evaluation Board (xc7z045ffg900-2) and the implementation results of this architecture are also provided.

The paper is organized as follows: the next section contains several common definitions and notations used herein and the overview of the encoding process. The FPGA implementation of the LDPC encoder is described in section III. Section III contains several definitions, the common description of the simulation model and all LDPC codes used herein. The hardware implementation results are presented in section IV. Finally, Section V concludes this paper.

II. Definitions and notations

For QC-LDPC codes encoder implementations only base parity-check matrixes H_b are stored, that allows to significantly reduce memory consumption for matrix set storage. But the structure of the base matrix limits parallelism in checks computation. Parity check matrices H_b that are used in this paper are specified in [5], and encoding procedure carry out as follows.

This work was supported by Russian Science Foundation under Grant 17-79-20302.

The standard IEEE 802.16e specifies 12 different codes supporting coding rates of 1/2, 2/3, 3/4 and 5/6. The LDPC encoder is systematic, i.e. it encodes an information block, $s = (s_0, s_1, ..., s_{(k-1)})$, of size k into a codeword, x, of size n, $x = (s_0, s_1, ..., s_{(k-1)}, p_0, p_1, ..., p_{(n-k-1)})$. A parity check matrix H has size $m \times n$ and can be obtained from the base matrix H_b. The matrix H is defined as $H = P^{H_b}$, where P is cyclic-permutation matrix size of $z \times z$. The matrix H_b has size $m_b \times n_b$, where $m_b = m/z$, $n_b = n/z$.

Encoding of LDPC codes uses the following property of the base parity-check matrix:

$$H_b \times x^T = 0^T. \tag{1}$$

The base matrix H_b can be divided into two parts: $H_b = [H_{b1} \;\; H_{b2}]$. H_{b1} is of size $m_b \times k_b$. It corresponds to the information bits with $k_b = n_b - m_b$. The matrix H_{b2}, in turn, corresponds to the parity-check bits and is of size $m_b \times m_b$. Expression (1) can be rewritten as:

$$\begin{bmatrix} H_{b1} & H_{b2} \end{bmatrix} \begin{bmatrix} s \\ p \end{bmatrix} = 0^T. \tag{2}$$

After solving (2), we get:

$$p = H_{b2}^{-1} H_{b1} s. \tag{3}$$

However, direct realization of (3) has a high encoding complexity. H_{b2} can be partitioned into two parts:

$$H_{b2} = \begin{bmatrix} h_{b2} & | & H'_{b2} \end{bmatrix} =$$

$$= \left[\begin{array}{c|ccccccc} h_b(0) & 0 & -1 & -1 & -1 & \dots & \dots & -1 \\ -1 & 0 & 0 & -1 & -1 & \dots & \dots & -1 \\ -1 & -1 & 0 & 0 & -1 & \dots & \dots & -1 \\ \vdots & -1 & \dots & \ddots & \ddots & -1 & \dots & -1 \\ h_b(i) & -1 & \dots & \dots & \ddots & \ddots & \dots & -1 \\ -1 & -1 & \dots & \dots & \dots & \ddots & \ddots & -1 \\ \vdots & \dots & \dots & \dots & \dots & -1 & 0 & 0 \\ h_b(m_b-1) & -1 & \dots & \dots & \dots & -1 & -1 & 0 \end{array}\right] \tag{4}$$

Column vector h_{b2} has 3 elements with values which are equal to or greater than 0. All other values of the vector are -1.

Matrix H'_{b2} has a dual diagonal structure where each element has a value in accordance with (5).

$$h'_{b2} = \begin{cases} 0, & \text{if } i = j \text{ or } i = j+1 \\ -1, & \text{elsewhere} \end{cases}, \tag{5}$$

where i and j are row and column indexes of matrix H_{b2} respectively. Therefore, expression (3) can be rewritten as (6)

$$H_{b2} \times p = H_{b1} \times s. \tag{6}$$

III. Encoder implementation

A pipeline implementation structure is proposed to increase the throughput. Input data is updated every cycle and previous data is moved through the pipeline structure of the encoder.

As a result of encoding, parity bits p are obtained from information bits s. During the encoding process information blocks are divided into $k_b = n_b - m_b$ groups u of z bits each.

$$u = [u(0)\, u(1) \,...\, u(k_b - 1)], \tag{7}$$

where each element of u is a column vector as follows:

$$u(i) = [S_{iz}\, S_{(iz+1)} \,...\, S_{(i+1)z-1}]^T \tag{8}$$

Using the model matrix H_{bm}, the parity sequence p is determined in groups ν of z bits,

$$\nu = [\nu(0)\, \nu(1) \,...\, \nu(m_b - 1)] \tag{9}$$

where each element of ν is a column vector as follows:

$$\nu(i) = [p_{iz}\, p_{(iz+1)} \,...\, p_{(i+1)z-1}]^T. \tag{10}$$

The LDPC encoder realization is performed in two stages: initialization and parallel computation. On the initialization stage the parity check bit vector $\nu(0)$ is computed by:

$$P_{p(x,k_b)}\nu(0) = \sum_{j=0}^{k_b-1} \left(\sum_{q=0}^{m_b-1} P_{p(q,j)} \right) u(j). \tag{11}$$

On the recursion stage parallel computation is performed. The parity check bit vectors $\nu(1) \sim \nu(m_b - 1)$ are concurrently computed by:

$$\nu(i) = \sum_{j=0}^{k_b-1} \left(\sum_{q=0}^{m_b-1} P_{p(q,j)} \right) u(j) + \sum_{q=i}^{m_b-1} P_p(q, k_b)\nu(0), \tag{12}$$

where $i = 1,\, ...,\, m_b - 1$

The parallel encoding method could significantly reduce the latency at the expense of extra storage for the sum [5]:

$$c(i) = \sum_{q=i}^{m_b-1} P_{p(q,j)}. \tag{13}$$

A. Initialization

Firstly, the parity check bit vector $\nu(0)$ is computed by (11), where P_i is an identity matrix of size $z \times z$, which is circularly right shifted by i and x is the row index of element $p_{(q,j)}$ from the matrix H_{b1}.

The expression (11) is computed by summing the product of matrix H_{b1} elements with the input data vectors $u(j)$ row by row and then column by column. Each product can be obtained by circular right shifting of the information block $u(j)$ as shown in (14).

$$P_{p_{(q,j)}} u(j) = \begin{cases} 0, & \text{if } p_{(q,j)} = -1 \\ u(j), & \text{if } p_{(q,j)} = 0 \\ u_{shifted}(j), & \text{if } p_{(q,j)} > 0 \end{cases}, \tag{14}$$

The non-zero elements $p(q, j)$ of H_{b1} specify circular right shift of the information block $u(j)$. The shifted information block $u_{shifted}(j)$ is obtained as a result of shifting. If $p(q, j)$ is equal to 0, the information block $u(j)$ is not changed. If $p(q, j)$ is equal to -1, the information block $u(j)$ is 0. Therefore equation (14) can be implemented by a barrel shifter. Every clock cycle a new $u(j)$ comes to the input of the m_b barrel shifters from the Input block as shown in Figure 1. The shifts are simultaneously performed for each data block $u(j)$ in m_b shifters within one clock cycle. Each data block is accumulated in the Column Accumulator after shifting.

After performing $k_b - 1$ shift-sum operations, the Controller block sends a reset signal to the Column Accumulator and a read signal to the Column Accumulator Buffer. The Column Accumulator Buffer receives new data from the Column Accumulator every $k_b - 1$ clock cycle. After accumulating m_b data blocks in the Column Accumulator Buffer, a vector of size $m_b \times z$, necessary to calculate $\nu(0)$, is obtained. Vector

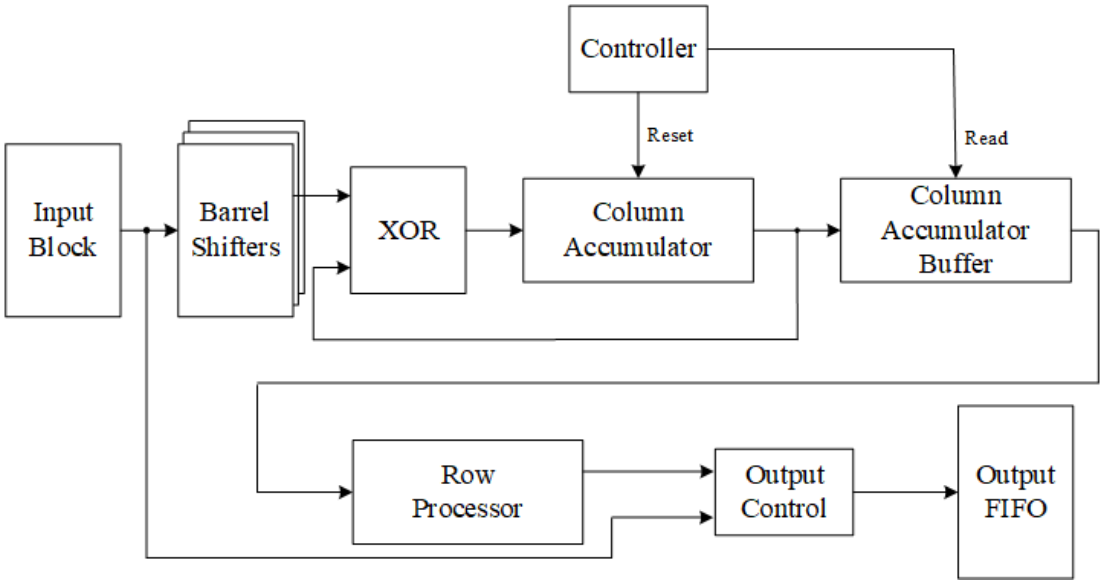

Figure 1. Structure of the FPGA LDPC encoder

$\nu(0)$ is calculated by performing the XOR operation with all k blocks. The XOR operation is executed line by line with saving the temporary results $c(i)$ inside the Row Processor. These results are required to calculate parity bit vectors $\nu(1)$ to $\nu(m_b - 1)$.

B. Parallel computation of parity bit vectors $\nu(1)$ to $\nu(m_b-1)$

The second step is computation of remaining parity bit vectors from $\nu(1)$ to $\nu(m_b-1)$. Calculations can be made in accordance with (6) and (12). Equation (6) can be rewritten as:

$$\nu(i) = c(i) + \sum_{q=i}^{m_b-1} P_p(q, k_b)\nu(0), \tag{15}$$

where vectors $c(i)$ were computed in the previous step according to (13).

Expression $P_p(q, k_b)\nu(0)$ in (15) represents a circular right shift version of the vector $\nu(0)$. Shift values are defined in h_{b2}. The standard IEEE 802.16e specifies values for h_{b2} as follows:

$$h_b(i) = \begin{cases} 0, & \text{if } i = m_b/2 \\ 1, & \text{if } i = 0 \text{ or } i = m_b - 1 \\ -1, & \text{elsewhere} \end{cases}, \tag{16}$$

The matrix H_{b2} in (4) can be rewritten as follows:

$$H_{b2} = \begin{bmatrix} 1 & 0 & -1 & -1 & -1 & \dots & \dots & -1 \\ -1 & 0 & 0 & -1 & -1 & \dots & \dots & -1 \\ -1 & -1 & 0 & 0 & -1 & \dots & \dots & -1 \\ \vdots & -1 & \dots & \ddots & \ddots & -1 & \dots & -1 \\ 0 & -1 & \dots & \dots & \ddots & \ddots & \dots & -1 \\ -1 & -1 & \dots & \dots & \dots & \ddots & \ddots & -1 \\ \vdots & \dots & \dots & \dots & \dots & -1 & 0 & 0 \\ 1 & -1 & \dots & \dots & \dots & -1 & -1 & 0 \end{bmatrix} \tag{17}$$

The $\nu(j)$ parity bit vector can be computed every clock cycle. The addition in Equation (15) is performed by XORing $c(i)$ with the sum of corresponding values of the shifted versions of the vector $\nu(0)$.

The Output Control block is designed to form the output sequence $x = [s\ p]$ consisting of the data sequence s and the parity-check bit vector p.

TABLE 1. Synthesis results for IEEE 802.16e

n	q	Cells	Clock (MHz)	Throughput (Gbps)
576	24	1484	260	5.2
960	40	2756	225	7.5
1440	60	4008	215	10.75
1920	80	5492	205	13.67
2304	96	6837	200	16

IV. RESULTS

The proposed encoder was implemented using a FPGA ZYNQ-7 ZC706 Evaluation Board (xc7z045ffg900-2). The design is parameterized and can be resynthesized to support various code rates, lengths and circulant sizes. Thus the LDPC encoder was implemented for IEEE 802.16e codes with codeword lengths of 576, 960, 1440, 1920, 2304 and for one random generated QC-LDPC codes with codeword length 59880.

The encoder is tested at a 5/6 coding rate for all code block sizes provided in IEEE 802.16e standard. The results are shown in the Table 1. It should be noted that the highest throughput is obtained at a frequency of 200 MHz if the codeword length is 2304 (circulant is $q = 96$). The lowest throughput is obtained at a frequency of 260 MHz if the codeword length is 576 ($q = 24$).

It can be seen, that in the proposed architecture, throughput depends on the size of the circulant. The use of a larger circu-

TABLE 2. Comparison of FPGA for LDPC Encoders

	[11]	[16]	present work
Block length	576 - 2304	2304	576 - 2304
Code rate	5/6	5/6	5/6
Cells (Logic Element)	3391-12306	2580-11399	1484-6837
Frequency, MHz	196.23 - 150.69	117	260-200
FPGA Technology	Altera STRATIX EP1S25F672C6	Altera FPGA Cyclone II EP2C70F896C6	ZYNQ-7 ZC706 Evaluation Board (xc7z045ffg900-2)
Throughput, Gbps	3.32 - 6.28	1.2	5.2 - 16

lant will increase the throughput. For example, experimental synthes was conducted on the random generated QC-LDPC code. The matrix parameters of this code were as follows: $m_b = 30$, $k_b = 120$, circulant size $z = 499$, coding rate was 3/4, codeword length was 59880. When using this matrix on the xc7z045ffg900-2 board, 292 234 cells and 89% LUT were used. The encoder frequency was 60 MHz and the throughput was 22.45 Gbps.

Also a comparison with known FPGA implementations of WiMax encoders was done, the results are presented in Table 2. As can be seen encoder from [11] operates at a frequency of 196.23 MHz for codeword length 576 and has a throughput of 6.28 Gbps. If the codeword length is 2304, the encoders from [11] operate at frequencies of 150.69 MHz and have throughput of 5.67 Gbps. Encoder from [16] operates only with codeword size 2304 on frequency 117 MHz with throughput 1.2 Gbps. Consequently the proposed design of encoder achieves the maximum throughput for IEEE 802.16e codes among considered ones, and at the same time have flexibility working with different code parameters.

V. Conclusion

The process of designing an LDPC pipeline encoder for QC-LDPC codes was specified herein. It was shown that proposed design has from 2.54 to 13.3 time greater throughput than known implementation of IEEE 802.16e encoders with different architecture.

The maximum throughput obtained for the developed LDPC encoder architecture is 22.45 Gbps.

At the same time the proposed design may be used to synthesize encoders not only for IEEE 802.16e and IEEE 802.11n standards, but for any QC-LDPC code.

References

[1] R. G. Gallager, "Low-density parity-check codes", Cambridge, MA: M.I.T. Press, 1963.

[2] D.J.C. MacKay, R.M. Neal, "Near Shannon limit performance of low density parity check codes", Electron. Lett., vol. 32, no. 18, pp. 1645-1646, Aug. 1996.

[3] S. Y. Chung, G. D. Forney, T. J. Richardson, and R. L. Urbanke, "On the design of low-density parity check codes within 0.0045 dB of the Shannon limit," IEEE Commun. Lett., vol. 5, pp. 58–60, Feb. 2001.

[4] 802.11n-2009 - IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput, 29 Oct. 2009, DOI: 10.1109/IEEESTD.2009.5307322

[5] 802.16e-2005 - IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems - Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, 28 Feb. 2006, DOI: 10.1109/IEEESTD.2006.99107

[6] 3GPP TS 38.212 version 15.2.0 Release 15 - 3GPP Standard 5G; NR; Multiplexing and channel coding, Jul. 2018

[7] A. Hariri, F. Monteiro, L. Sieler, A. Dandache, "A High Throughput Configurable Parallel Encoder Architecture for Quasi-Cyclic Low-Density Parity-Check Codes", IEEE 19th International On-Line Testing Symposium (IOLTS), 2013, pp. 163-166, DOI: 10.1109/IOLTS.2013.6604069

[8] C. Hui, Y. Wang, X. Lu, "Implementation of a High Throughput LDPC Codec in FPGA for QKD System", 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT), 2013, 25-28 Oct. 2016, DOI: 10.1109/ICSICT.2016.7998780

[9] Z. He, S. Roy, P. Fortier, "Powerful LDPC Codes for Broadband Wireless Networks: High-performance Code Construction and High-speed Encoder/Decoder Design", 2007 International Symposium on Signals, Systems and Electronics, 30 July-2 Aug. 2007, DOI: 10.1109/ISSSE.2007.4294441

[10] M. Gomes, G. Falcão, A. Sengo, V. Ferreira, V. Silva, "High Throughput Encoder Architecture for DVB-S2 LDPC-IRA Codes", 2007 Internatonal Conference on Microelectronics, 29-31 Dec. 2007, DOI: 10.1109/ICM.2007.4497709

[11] S. Kopparthi, D. M. Gruenbacher, "Implementation of a Flexible Encoder for Structured Low-Density Parity-Check Codes", 2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, 22-24 Aug. 2007, pp. 438-441, DOI: 10.1109/PACRIM.2007.4313268

[12] Z. He, S. Roy, "High-speed Design of Adaptive LDPC Codes for Wireless Networks", The 2008 IEEE Northeast Workshop on Circuits and Systems (NEWCAS'08), 22-25 June 2008, DOI: 10.1109/NEWCAS.2008.4606368

[13] A. Hariri, F. Monteiro, L. Sieler, A. Dandache, "Configurable and High-Throughput Architectures for Quasi-Cyclic Low-Density Parity-Check Codes",2014 21st IEEE International Conference on Electronics, Circuits and Systems (ICECS), Marseille, France, 7-10 Dec. 2014, DOI: 10.1109/ICECS.2014.7050104

[14] H. Yin, X. Yang, Z. Yang, "A Design and Implementation of LDPC Encoder Based on Mobile-Multimedia-Broadcasting System", 2009 International Conference on Management and Service Science, Wuhan, China, 20-22 Sept. 2009, DOI: 10.1109/ICMSS.2009.5301316

[15] C. Kun, S. Qi, L. Shengkai, P. Chengzhi, "Implementation of encoder and decoder for LDPC codes based on FPGA", Journal of Systems Engineering and Electronics Vol. 30, No. 4, August 2019, pp.642 – 650, DOI: 10.21629/JSEE.2019.04.02

[16] W. Xiumin, G. Tingting, L. Jun, S. Chen, H. Fangfei, "Efficient Multi-rate Encoder of QC-LDPC Codes Based on FPGA for WIMAX Standard", Chinese Journal of Electronics Vol.26, No.2, Mar. 2017, pp. 250-255, DOI: 10.1049/cje.2017.01.006

[17] D. Theodoropoulos, N. Kranitis, A. Paschalis, "An Efficient LDPC Encoder Architecture for Space Applications", 2016 IEEE 22nd International Symposium on On-Line Testing and Robust System Design (IOLTS), pp. 149-154, 4-6 July 2016, DOI: 10.1109/IOLTS.2016.7604689

[18] IEEE 802.11ay: Next-Generation 60 GHz Communication for 100 Gb/s Wi-Fi, 27 Oct. 2017, DOI: 10.1109/MCOM.2017.1700393