International Journal of Control and Automation
Vol. 9, No. 7 (2016), pp.273-280
http//dx.doi.org/10.14257/ijca.2016.9.7.25

QC-LDPC Encoder Structure for IEEE 802.16e/802.11n Standard

Xiumin Wang[1], Tingting Ge[1], Yi Wang[1], Zhengquan Li[1,2] and Nurul I. Sarkar[3]

1. College of Information Engineering, China Jiliang University, Hangzhou, China, 310018
2. National Mobile Communications Research Laboratory, Southeast University Nanjing, China, 210018
3. School of Computing & Mathematical Sciences, Auckland University of Technology, New Zealand
E-mail: wxm6341@163.com

Abstract

Due to the popularity of LDPC (Low Density Parity Codes) in modern communication systems, there is a growing need of LDPC encoder with a faster encoding speed and more application compatibility. In this paper, an efficient QC-LDPC encoder architecture is proposed, which is suitable for both IEEE 802.16e standard and IEEE 802.11n standard. Owing to the special structure of multi-diagonal parity matrices, the inverse matrix is replaced with back substitution. Simplified RU algorithm is applied to the encoder structure, and the parallelism is determined by the row number of parity matrices in the two standards. The proposed architecture has full configurability with concern to code rate and code length. Simulation results confirm that the encoders work successfully from 1/2 rate to 5/6 rate under both two standards. To validate its feasibility, we have tested all code lengths of IEEE 802.11n standard and IEEE 806.12e standard. They all work accurately with the proposed encoder architecture, with the output throughputs ranging between 538 Mbps and 2879 Mbps.

Keywords*: QC-LDPC, encoder, WIMAX, IEEE 802.11n*

1. Introduction

LDPC codes have excellent error correction ability close to Shannon capability [1], so they have been adopted by a variety of communication standards, such as 10GBASE-T, WIMAX (Worldwide Interoperability for Microwave Access) [2], DTMB (Digital Television Terrestrial Multimedia Broadcasting) and IEEE 802.11n [3], *etc.*

In recent years, many encoders have been presented for different standards with the aim of achieving lower complexity and higher throughput. In [4], encoder architecture for QC-LDPC codes in IEEE 802.11n/ac standards was proposed. To achieve high throughput, they have designed a low-complexity and rate-compatible cyclic shift. In [5], multi-stage cyclic shift structure was proposed, which can be applied to our design to replace the conventional barrel shift. In [6], an encoder for IEEE 802.11n was designed with low cost, and their results have shown that a fully parallel encoder would require 12 times more hardware resources than our proposed one.

Efficient encoder structures were also designed for IEEE 802.16e and IEEE 802.11n in [7-10]. In paper [7], semi-parallel architecture was proposed and an encoder for QC-LDPC codes of the two standards was implemented. A joint design of encoder and decoder of LDPC codes was proposed based on FPGA platform [8]. A throughput of up to 32Gbps was achieved for the encoder on FPGA in [9] and this architecture was compatible with different lengths using a semi-random **H**-matrix. A parameter-

ISSN: 2005-4297 IJCA

configurable QC-LDPC encoder structure was also developed in [10]. However, it achieved a lower throughput than our proposed architecture.

In this paper, we focus on the common structure of parity matrices in the two standards and identify the differences between them. Then we design the configurable encoder architecture for arbitrary numbers of code rate and code length. It has superior compatibility compared to other works which are designed for only one standard.

The remaining of the paper is organized as follows. Section 2 briefly introduces the parity matrices of QC-LDPC codes under IEEE 802.16e/IEEE 802.11n standard and simplifies the RU algorithm. Then the encoder structure suitable for any code length and code rate under the two standards is presented in Section 3. Section 4 shows the simulation results and comparisons with other works. Finally, conclusions are given in Section 5.

2. Simplified Algorithm for Two Standards

QC-LDPC codes are defined by the parity check matrix **H**. In both of the two standards, the parity matrix of size $m \times n$ is composed of identity sub-matrices and circular shifted sub-matrices and null sub-matrices (n=24). There are 19 code lengths in IEEE 806.12e ranging from 576 to 2304, while in IEEE 802.11n, there are three code lengths of 648, 1296 and 1944. For different lengths, they have different expansion factors z (For WIMAX standard, z factor varies from 24 to 96; For IEEE 802.11n, z factor varies from 27 to 81). Both of the two standards have various code rates of 1/2, 2/3, 3/4 and 5/6. The common ALT (Approximately Lower Triangular) structure of the multi-diagonal parity matrices is given in Figure.1 (r denotes code rate).

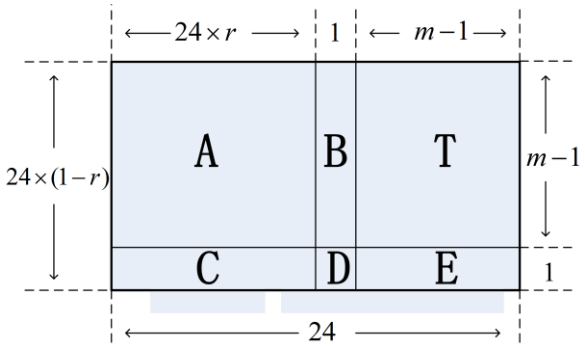

Figure 1. Common Structure of Parity Matrices

where, m=24×(1-r); **A** is (m-1)×(24×r) matrix; **B** is (m-1)×1 matrix; **T** is (m-1)×(m-1) matrix; **C** 1×(24×r) matrix; **D** is 1 × 1 matrix; and finally, **E** is 1×(m-1) matrix.

In RU encoding algorithm [11], we calculate the parity bits as follows

$$p_0 = (D + ET^{-1}B)^{-1}(C + ET^{-1}A)s, \tag{1}$$

$$p_1 = T^{-1}(As + Bp_0). \tag{2}$$

where, **s** denotes the information bits with 24×r z-bit vectors; $\mathbf{p_0}$ is the first part of parity bits with one z-bit vector; $\mathbf{p_1}$ is the second part of parity bits with m-1 z-bit vectors.

After analyzing the property of the standards, we find that the inverse matrix $(\mathbf{D}+\mathbf{ET}^{-1}\mathbf{B})^{-1}$ equals identity matrix in every rate except 3/4B rate in WIMAX standard. Take 5/6 rate in WIMAX standard for example, **B**= ($\mathbf{I_{80}}$, **I**, **O**), **D**= $\mathbf{I_{80}}$, **E**= (**O**, **O**, **I**). **I** and **O** respectively denote identity matrix and null matrix, and $\mathbf{I_{80}}$ denotes the 80^{th} right cyclic shift of identity matrix. Set $\mathbf{T}^{-1}\mathbf{B}=(x, y, w)^T$, then we can get $\mathbf{ET}^{-1}\mathbf{B}=w$. The value w can be obtained as follows

$$\begin{bmatrix} I & O & O \\ O & I & I \\ O & O & I \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} I_{80} \\ I \\ O \end{bmatrix}, \tag{3}$$

$$w = I - I_{80}. \tag{4}$$

Finally we can get

$$D + ET^{-1}B = D + w = I_{80} + I - I_{80} = I. \tag{5}$$

The simplification is suitable for other code rates. Then the equation (1) is changed into $\mathbf{p_0}=(\mathbf{C}+\mathbf{ET}^{-1}\mathbf{A})\mathbf{s}$. For 3/4B rate in WIMAX standard, $\mathbf{D}+\mathbf{ET}^{-1}\mathbf{B}$ equals to a circular right shifted matrix rather than an identity matrix.

The inverse matrix requires a large number of calculations which we'd better replace it with other calculations. Thus, we set $x=T^{-1}As$ then get $\mathbf{Tx}=\mathbf{As}$. We simplify it based on the dual-diagonal structure of matrix **T**, so we can get equations (6) and (7)

$$\begin{bmatrix} I & O & \cdots & O \\ I & I & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & \cdots & I & I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m-1} \end{bmatrix} = \begin{bmatrix} A_1 s \\ A_2 s \\ \vdots \\ A_{m-1} s \end{bmatrix}, \tag{6}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m-1} \end{bmatrix} = \begin{bmatrix} A_1 s \\ A_2 s \\ \vdots \\ A_{m-1} s \end{bmatrix}. \tag{7}$$

Because matrix **E** has only one identity sub-matrix ($\mathbf{E}=(\mathbf{O},\mathbf{O},\cdots,\mathbf{I})$), so $\mathbf{Ex}=\mathbf{ET}^{-1}\mathbf{As}=\boldsymbol{x}_{m-1}$ and from equation (7) we can get $\boldsymbol{x}_{m-1}=\mathbf{A_{m-1}s}+\mathbf{A_{m-2}s}+\cdots\mathbf{A_2s}+\mathbf{A_1s}$. Finally the first part of parity bits can be calculated as $\mathbf{p_0}=\mathbf{Cs}+\mathbf{A_{m-1}s}+\mathbf{A_{m-2}s}+\cdots\mathbf{A_2s}+\mathbf{A_1s}$. For the second part of parity bits, the simplification is similar as equations (7), and the inverse matrix is replaced with back substitution. For every *z*-bit vector in $\mathbf{p_1}$ we have $\mathbf{p_1}=\mathbf{p_1}'+(\mathbf{As}+\mathbf{Bp_0})$ and $\mathbf{p_1}'$ denotes the value of $\mathbf{p_1}$ at previous clock cycle. Finally, the common encoding steps for the two standards are given in Table 1.

Table 1. Simplified Encoding Steps for Two Standards

Steps	Calculations
1.	Parallel calculate A_1s, A_2s, $\cdots$, $A_{m-1}s$, Cs
2.	Get $p_0=Cs+A_{m-1}s+A_{m-2}s+\cdots A_2s+A_1s$
3.	Serially calculate $A_1s+B_1p_0$, $\cdots$, $A_{m-1}s+B_{m-1}p_0$
4.	Serially get $p_1=p_1'+(As+Bp_0)$ in *z*-bit vectors

3. Common Encoder Architecture for the Two Standards

3.1. Overall Encoding Structure

As shown in Figure.2, the encoding structure is suitable for any code length and rate. The encoding process can be divided into four steps, which is similar with the steps in Table 1. The information bits $\boldsymbol{s}$ is input serially in *z*-bit vectors. The parallel ROMs and left shifts in the first dashed box is set to get the accumulated matrix-to-vector multiplications ($\mathbf{A_1s}$, $\mathbf{A_2s}$, $\cdots$, $\mathbf{A_{m-1}s}$, $\mathbf{Cs}$). The successive step is to send the output results to a group of *m-1* XOR gates to get $\mathbf{p_0}$. In the third dashed box, $\mathbf{A_1s}+\mathbf{B_1p_0}$, $\cdots$, $\mathbf{A_{m-1}s}+\mathbf{B_{m-1}p_0}$ are serially calculated and stored in the RAM. Then they

are read address by address to calculate $\mathbf{p_1}$. After $\mathbf{p_1}$'s calculation is finished, $\mathbf{p_1}$ will be output together with $\mathbf{s}$ and $\mathbf{p_0}$.

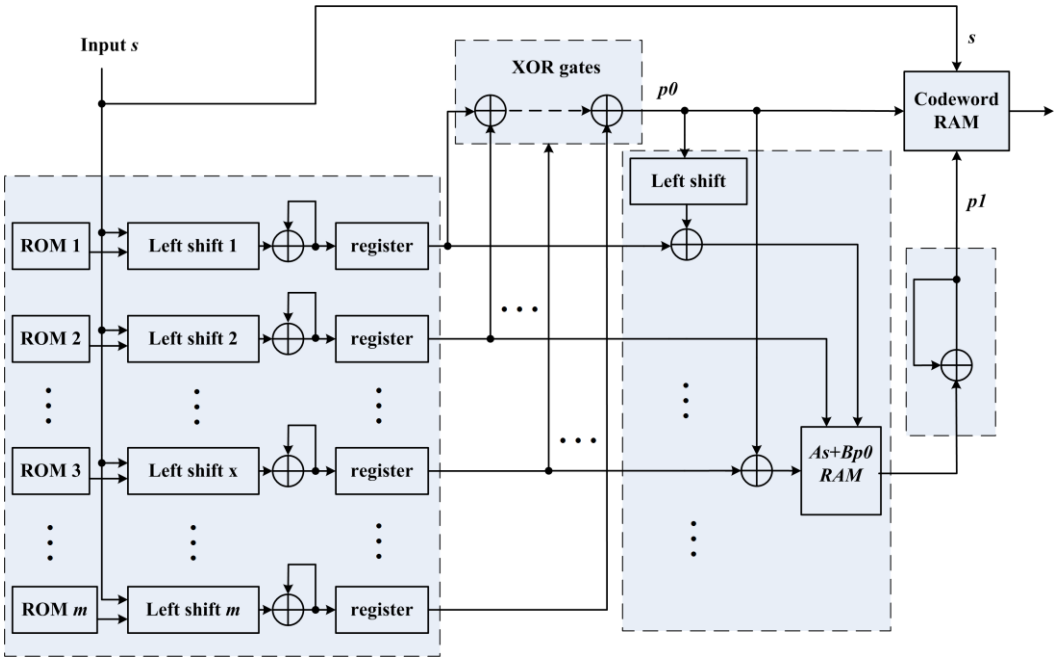

Figure 2. Common Encoding Structure for the two standards

3.2. Matrix-to-Vector Multiplications and $\mathbf{p_0}$ Calculation

The multiplication of a $z \times z$ right shifted cyclic matrix and a z-bit vector is equal to the left shifted vector with the same shift value. So the input information vectors should be shifted according to the values in the base parity matrix. In this design, we use a group of m ROMs to store the shift values in different row. In other words, the parallelism is m and shift value from i-th ROM is sent to i-th left shift to calculate $\mathbf{A_i s}$ and $\mathbf{CS}$. After all shift values are read and the multiplications are finished, m-1 groups of $\mathbf{A_i s}$ and one group of $\mathbf{CS}$ are accumulated in XOR gates to get $\mathbf{p_0}$.

A barrel shift is a traditional choice to do matrix-to-vector multiplications. It needs $\log_2 z$ clock cycles to finish the shift operation because it judges '0' or '1' every bit and gives proper command of shift values. In [5], adjacent two bits are used to tell four different shift conditions: '00', '01', '10' and '11'. So it needs $\log_4 z$ clock cycles to finish the shift of a z-bit vector. Obviously, this kind of left shift is conclusive to the encoding speed with negligible increase of logic elements. So we apply this kind of left shift to our design. Take factor z=48 for instance, we divide the shift value 'cnt[5:0]' into three parts: 'cnt[5:4]' is used to tell the shift values 48, 32, 16 and 0; while 'cnt[3:2]' and 'cnt[3:2]' are used to tell the shift values of their corresponding positions ('cnt[3:2]' denotes shift values of 12, 8, 4 and 0, and 'cnt[1:0]' denotes shift values of 3, 2, 1 and 0). It takes 3 clock cycles while a barrel shift requires 6 clock cycles. If the width of 'cnt' is singular, it needs one more clock cycle to judge the last bit.

3.3. Calculations for $\mathbf{p_1}$

$\mathbf{A_i s}$ are serially input into the third dashed box in m-1 groups to calculate corresponding $\mathbf{A_i s + B_i p_0}$. Due to the uniform form of matrix $\mathbf{B}$'s density, there is one left shift and two XOR gates in this dashed box. For any rate and code length in the two standards, $\mathbf{p_0}$ should be shifted to calculate the first z-bit $\mathbf{A_1 s + B_1 p_0}$. For most cases, $\mathbf{B_i p_0}$ is equal to null

matrix because of the null sub-matrices in matrix **B**. That's to say, $\mathbf{A_i s}$ is directly stored into RAM in these situations. Apart from a shift cyclic sub-matrix, there is also an identity sub-matrix in matrix **B**. So we need one more XOR gate to calculate $\mathbf{A_x s+p_0}$ (*x* is determined by code rate). After all these calculations are finished, $\mathbf{A_i s+B_i p_0}$ should be read address by address to get $\mathbf{p_1}$. The value of $\mathbf{p_1}$ is equal to the exclusive-or result of $\mathbf{As+Bp_0}$ and $\mathbf{p_1}$'s previous value. It is calculated and output in *z* bits every clock cycle. Ultimately, information bits **s**, parity bits $\mathbf{p_0}$ and $\mathbf{p_1}$ are output through the codeword RAM orderly.

Although the encoder architecture can be applied to the two standards without any change, it is worth mentioning that the left shift in the third dashed box has certain shift value '*1*' for IEEE 802.11n standard. The encoder structure is suitable for any code length and code rate except 3/4B rate in WIMAX. For 3/4B rate, it differs from other code rates because $\mathbf{D+ET^{-1}B}$ is equal to a cyclic shift matrix rather than an identity matrix. So after outputting from the XOR gates in the second dashed box, a shift register should be added to get $\mathbf{p_0}$ for the implementation of 3/4B rate encoder. The other logic elements stay the same.

4. Simulations and Comparisons

To validate the proposed encoder architecture, the encoder was synthesized on Cyclone II P2C70F896C6 device using Verilog HDL language. We applied the QC-LDPC encoder structure to IEEE 802.16e and IEEE 802.11n standard. Encoders of different rates in the maximum codeword length are implemented and their comparisons are presented in Figure.3 and Figure.4 (for IEEE 802.16e standard, the maximum codeword length is 2304; for IEEE 802.11n standard, the maximum codeword length is 1994). What's more, expansion factor *z* in IEEE 802.16e is 96, and expansion factor *z* in IEEE 802.11n standard is 81. In Figure 3, we can see that throughput rises with the rise of code rate. Encoders under IEEE 802.16e have throughputs ranging between 1.9Gbps to 2.8Gbps; while encoders under IEEE 802.11n have slightly lower throughputs ranging between 1.6Gbps to 2.6Gbps. Figure 4 shows logic elements consumption of encoders under the two standards. The two curves are very close and they approximately vary from 3000 to 11000.

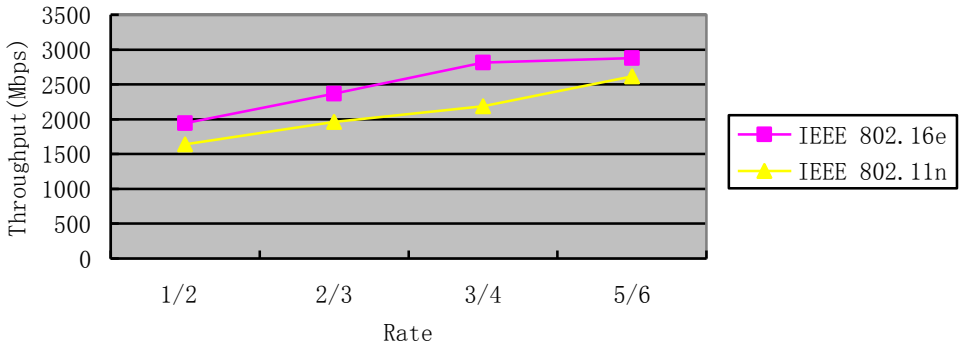

Figure 3. Throughput Comparisons in Different Rates

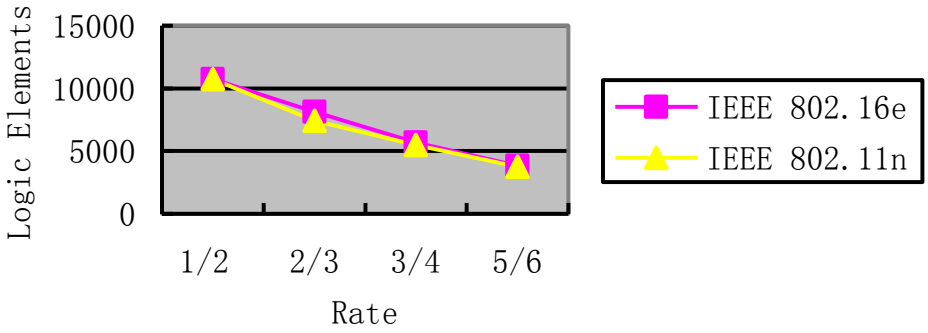

Figure 4. Logic Elements Comparisons in Different Rates

The encoder structure also runs successfully for other code lengths, so a comparison is provided in Table 2. between the two standards in terms of resources utilization and throughput. It is obvious that encoders of the two standards have approximately close logic elements consumption and memory bits consumption. The throughputs of encoders in IEEE 802.11n are slightly higher.

Table 2. Resource Summaries of Different Code Lengths

Standard	IEEE 802.16e (1/2 rate)			IEEE 802.11n (1/2 rate)		
Code length	576	1152	2304	648	1296	1944
Frequency	250	238	200	248	238	192
Logic Elements	2479	4891	10748	2754	4859	10742
Memory Bits	984	1392	2064	1017	1458	2208
Throughput (Mbps)	538	1026	1943	611	1176	1638

To improve the superiority of the proposed encoder structure, we compare it with the recent literature in Figure 3. The blank in the table means that it is not given in the corresponding literature. In [7], the throughput of the encoder with code length of 648 under IEEE 802.11n is 360Mbps, while our encoder under the same condition has the throughput of 611Mbps. For WIMAX standard, although we consume more logic elements than [7] with code length of 2304, the throughput of our design is higher. The work [8] achieves lower throughput and frequency than our work. Obviously, our work has higher throughput and lower cost when compared to [12]. Based on the above analysis with the counterparts, our design has higher throughputs and consumes less logic elements relatively.

Table 3. Comparisons between the Proposed and Other Encoders

Works	Standard	Code length	Code Rate	Frequency	Logic Elements	Throughput (Mbps)
[7]	IEEE802.11n	648	1/2	180	--	360
	IEEE802.16e	2304	1/2	104.5	4305	1250
[8]	IEEE802.16e	2304	5/6	200	2366	1660
[12]	IEEE802.16e	2304	1/2	--	65964	600
the Proposed	IEEE802.11n	648	1/2	248	2754	611
	IEEE802.16e	2304	1/2	200	10748	1943
			5/6	238	3838	2879

5. Conclusions

In the paper, an efficient QC-LDPC encoder architecture has been presented for both IEEE 802.16e and IEEE 802.11n standards. Due to the similarity of the parity matrices under the two standards, we have simplified RU encoding algorithm and divided the encoding process into four steps. The encoding architecture is a serial-in/serial-out system and they are executed in *z* bits every clock cycle (expansion factor *z* is determined by the code length). To improve its accuracy, the proposed architecture is implemented with different code lengths and code rates under the two standards. Compared with its counterparts, it makes a better trade-off between resource consumption and encoding throughput. The proposed work is suitable for not only the two standards, but also other QC-LDPC codes with similar properties.

Acknowledgement

This work was supported by National Natural Science Foundation of China under Grant No. 61379027 & 61571108, the Open Research Fund of National Mobile Communications Research Laboratory of Southeast University under Grant No. 2011D18. What's more, the author would like to appreciate for the referees' careful reading and constructive comments.

References

[1] R. G. Gallager, "Low-density parity-check codes", IRE Trans. Inf. Theory, IT-8, vol. 1, **(1962)**.
[2] IEEE Std. 802.16e-2005 and IEEE Std. 802.16-2004/Cor1-2005, Local and Metropolitan area networks, IEEE,**(2006)**.
[3]I EEE P802.11n/D10: 'Draft IEEE Standard for Local Metropolitan networks–Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC), and Physical Layer (PHY) specifications: Enhancements for Higher Throughput', **(2006)**.
[4] Y. Jung, C. Chung, J. Kim and Y. Jung, "7.7Gbps encoder design for IEEE 802.11n/ac QC-LDPC codes", in 2012 International SoC Design Conference (ISOCC'2012), **(2012)**, pp. 215-218.
[5] D. Liu, H. Liu and B. Zhang, "Study on encoding algorithms for QC-LDPC codes with dual-diagonal parity check matrix", Journal of National University of Defense Technology, vol. 36, no. 2, **(2014)**.
[6] J.M. Pe´rez and V. Ferna´ndez, "Low-cost encoding of IEEE 802.11n", Electron. Lett., vol. 44, no. 4, **(2008)**.
[7] J. K. Kim, H. Yoo and M. H. Lee, "Efficient encoding architecture for IEEE 802.16e LDPC codes", IEICE Trans. Fundamentals., E91-A, vol. 12, **(2008)**.
[8] R. Yuan and B. Bai, "FPGA-based Joint Design of LDPC Encoder and Decoder", J. Electron. Inf. Tech., vol. 34, no. 1, **(2012)**.
[9] A. A. A. Hariri, F. Monteiro, L. Sieler, and A. Dandache, "Configurable and high-throughput architectures for Quasi-Cyclic Low-Density Parity-Check codes", in 21st IEEE International Conference on Electronics, Circuits and Systems. (ICECS'2014), **(2014)**, pp. 790-793.
[10] Zhao and L. Li, "Online high-speed programmable quasi-cyclic LDPC code encoder structure", J Tsinghua Univ. (Sci & Tech), vol. 49, no. 7, **(2009)**.
[11] T. Richardson and R. Urbanke, "Efficient encoding of low-density parity-check codes", IEEE Trans. Inf. Theory, vol. 47, no. 3, **(2001)**.
[12] Z. Ma, Y. Li, and X. Wang, "A Quasi-Parallel Encoder of Quasi-Cyclic LDPC Codes in IEEE 802.16e", in 1st International Conference on Information Science and Engineering (ICISE), **(2009)**, pp. 2492-2495.

Authors

Xiumin Wang, she is Professor and Associate Dean of College of Information Engineering in China Jiliang University. She hosted several projects: a project of National Natural Science Foundation of China in 2013, a science and technology project of the State General Administration of Quality Supervision in 2009 and a major science and technology project of Zhejiang Province in 2010. She also presided over four projects of Education Department of Liaoning Province Science Foundation and two key research projects of Hangzhou Science and Technology Program Soft.