

Implementation of Rate-Compatible QC-LDPC Encoder Based on FPGA

Ruijia Yuan^a, Tianjiao Xie^{*}, Jianhua Zhang^b

The China Academic of Space Technology (Xi'an), China

^{*}Corresponding author e-mail: xiexietianjiao@163.com, ^aRuijiayuan2022@163.com,
^bjianhuazhang2022@163.com

Abstract. A rate-compatible LDPC encoder based on quasi-cyclic generation matrix is proposed. The encoder partitions and controls access to the ROM address, so it can be compatible with a variety of LDPC codes to generate matrix cyclic shift vectors. By adding routing options in the register cyclic shift circuit, it is compatible with matrix blocks of different sizes. Due to the adjustment of the initial shift count and the truncation of the check bit output, the virtual filling and shortening of the LDPC code is realized, which further expands the code length and code rate of the encoder. An LDPC encoder compatible with 4 code rates is implemented on a Xilinx Virtex5 xc5vfx130t FPGA. Compared with the existing design, this encoder requires only slightly more hardware resources than a single encoder to achieve the same data throughput. Compared with implementing all four different encoders, more than 40% of the hardware resources can be saved.

Keywords: rate-compatible, QC-LDPC, encoder, FPGA.

1. Introduction

Classical channel coding, such as convolutional code, RS code, etc., although the design is exquisite and the processing is simple, the coding gain can be provided is limited, and there is still a significant gap between the performance and the transmission capacity limit proposed by Shannon. As the most widely used modern channel coding, LDPC code has the error correction performance approaching the capacity limit [1], which can maximize the coding gain of the link. In addition, in order to make full use of the link resources to achieve efficient transmission in the case of large signal-to-noise ratio fluctuations and reliable transmission in the case of low signal-to-noise ratio, it is necessary to adjust the coding efficiency according to the characteristics of the satellite link to realize adaptive transmission with different code rates.

There are two kinds of encoding methods for LDPC codes. One is encoding directly according to the parity-check matrix by using the special structure of parity-check matrix, such as double diagonal structure [2]. This method is easy to implement encoding [3], widely used in the second generation satellite broadcasting standards (DVB-S2), wireless LAN 802.11 series, wireless metropolitan area network 802.16e, DVB-T2, 802.11 ac standards; second, the check matrix is transformed into a generating matrix with quasi-cyclic structure, and then the generating matrix is used to calculate the check bit. In general, the second case has fewer constraints on the check matrix, so in the case of the same code length, the error correction performance of LDPC codes in the second case is better. However, the coding complexity of this case is higher, and this kind of code has been widely used in CCSDS standards [4]. This paper mainly studies the high speed implementation technology of this encoder. The earlier study of this situation is the use of SRAA proposed in Reference [5] to encode by using the quasi-cyclic characteristics of the generating matrix, and the coding of the linear complexity of the code length is realized. In Reference [6], SRAA was improved to RLA, which greatly reduced the required hardware resources, and the DTMB standard QC-LDPC encoder based on RLA architecture was implemented. In Reference [7], a rate-compatible QC-LDPC encoder is implemented using TSMC 90 nm, the gate count is 221.3 K. When the clock frequency is 222 MHz, the throughput of the encoder is 273 Mbps.

In Reference [8], a high-speed parallel encoder architecture was proposed, and the encoder of LDPC (2048, 1024) was implemented on Xilinx Virtex5 xc5v1x110t FPGA, the 16-channel parallel encoding required only a small number of Slices, of which 3258 LUTs and 2176 FFs were used.

In this paper, a rate-compatible LDPC coder based on quasi-cyclic generation matrix is proposed. Under the condition of less hardware resources, a variety of rate-compatible LDPC coders are implemented. The problem of implementing a variety of LDPC coders in satellite communication under the condition of limited hardware resources is solved. A low complexity and rate-compatible efficient LDPC coder implementation method is given.

2. Rate Compatible QC-LDPC Coding Algorithm

For QC-LDPC codes, the generator matrix with system form $\mathbf{G}_{qc}$ is expressed as:

$$\mathbf{G}_{qc} = [\mathbf{I}_k | \mathbf{P}_{k \times (n-k)}]_{k \times n} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{B}_{1,1} & \cdots & \mathbf{B}_{1,c} \\ \mathbf{0} & \mathbf{I} & \cdots & \mathbf{0} & \mathbf{B}_{2,1} & \cdots & \mathbf{B}_{2,c} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} & \mathbf{B}_{r,1} & \cdots & \mathbf{B}_{r,c} \end{bmatrix} \quad (1)$$

Here $\mathbf{I}_k$ denote the unit matrix of size $k \times k$, $\mathbf{P}_{k \times (n-k)}$ is the matrix of size $k \times (n-k)$, the generator matrix $\mathbf{G}_{qc}$ of size $k \times n$, $k=r \times L$, $n-k=L \times c$, the frame length of QC-LDPC code is $n=(r+c)L$. Where $\mathbf{B}_{i,j}$ is a circulant submatrix of $L \times L$ in size, as follows :

$$\mathbf{B}_{i,j} = \begin{bmatrix} (\mathbf{b}_{i,j})_1 & (\mathbf{b}_{i,j})_2 & \cdots & (\mathbf{b}_{i,j})_L \\ (\mathbf{b}_{i,j})_L & (\mathbf{b}_{i,j})_1 & \cdots & (\mathbf{b}_{i,j})_{L-1} \\ \vdots & \vdots & \ddots & \vdots \\ (\mathbf{b}_{i,j})_2 & (\mathbf{b}_{i,j})_3 & \cdots & (\mathbf{b}_{i,j})_1 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{i,j}^{(0)} \\ \mathbf{b}_{i,j}^{(1)} \\ \vdots \\ \mathbf{b}_{i,j}^{(L-1)} \end{bmatrix} \quad (2)$$

Obviously, each row vector $\mathbf{b}_{i,j}^{(l)}$ of $\mathbf{B}_{i,j}$ is the cyclic right-shift vector of the upper row vector $\mathbf{b}_{i,j}^{(l-1)}$, so if the first row vector $\mathbf{b}_{i,j}^{(0)}$ of $\mathbf{B}_{i,j}$ is known, $\mathbf{B}_{i,j}$ can be obtained, and $\mathbf{b}_{i,j}^{(0)}$ is called the generator of the cyclic submatrix.

For different QC-LDPC codes, the code rate corresponds to different frame length $(r+c)L$, where L is the length of the generator vector $\mathbf{b}_{i,j}^{(0)}$ of the cyclic submatrix $\mathbf{B}_{i,j}$, rL and cL are the number of rows and columns corresponding to the cyclic check bits of the generation matrix $\mathbf{G}_{qc}$ of QC-LDPC codes, respectively.

For QC-LDPC codes with different code rates, corresponding to different cyclic submatrix $\mathbf{B}_{i,j}$ generator vector $\mathbf{b}_{i,j}^{(0)}$, different r, c, l parameters. The check bits are divided into c sub-blocks with length L , namely $\mathbf{R}_{p \times cL} = (R_1, R_2, \dots, R_c)$, where $R_1 = (C_{j,1}, C_{j,2}, \dots, C_{j,L})$ $1 \leq j \leq c$, code word $\mathbf{v} = \mathbf{m} \mathbf{G}_{qc}(\mathbf{m}, c_1, c_2, \dots, c_c)$, So :

$$\mathbf{C}_j = \mathbf{M}_1 \mathbf{B}_{1,j} + \mathbf{M}_2 \mathbf{B}_{2,j} + \cdots + \mathbf{M}_r \mathbf{B}_{r,j} \quad 1 \leq j \leq c \quad (3)$$

Let $\mathbf{b}_{i,j}^{(l)}$ be the l th row of the cyclic matrix $\mathbf{B}_{i,j}$, then $\mathbf{b}_{i,j}^{(0)} = \mathbf{b}_{i,j}^{(l)} = \mathbf{b}_{i,j}$, when $1 \leq l \leq r$.

$$\mathbf{M}_i \mathbf{B}_{i,j} = (A_{(i-1)L+1}, \dots, A_{iL}) \begin{bmatrix} \mathbf{b}_{i,j}^{(0)} \\ \vdots \\ \mathbf{b}_{i,j}^{(L-1)} \end{bmatrix} = \sum_{l=1}^L A_{(i-1)L+k} \mathbf{b}_{i,j}^{(l)} \quad (4)$$

From formula (3) and formula (4), we can get the j -block check bit $\mathbf{C}_j$ as follows:

$$\mathbf{C}_j = \sum_{i=1}^r \sum_{l=1}^L A_{(i-1)L+k} \mathbf{b}_{i,j}^{(l)} \quad (5)$$

3. Rate-compatible QC-LDPC encoder architecture with high resource utilization

3.1 Rate-compatible QC-LDPC encoder architecture

This paper improves on the structure of the existing RLA encoder, and proposes an LDPC encoder compatible with multiple code rates. The encoder partitions and controls access to the ROM address, so it can be compatible with a variety of LDPC codes to generate matrix cyclic shift vectors. By adding routing options in the register cyclic shift circuit, it is compatible with matrix blocks of different sizes. Due to the adjustment of the initial shift count and the truncation of the check bit output, the virtual filling and shortening of the LDPC code is realized, which further expands the code length and code rate of the encoder. The encoder architecture is shown in Figure 1.

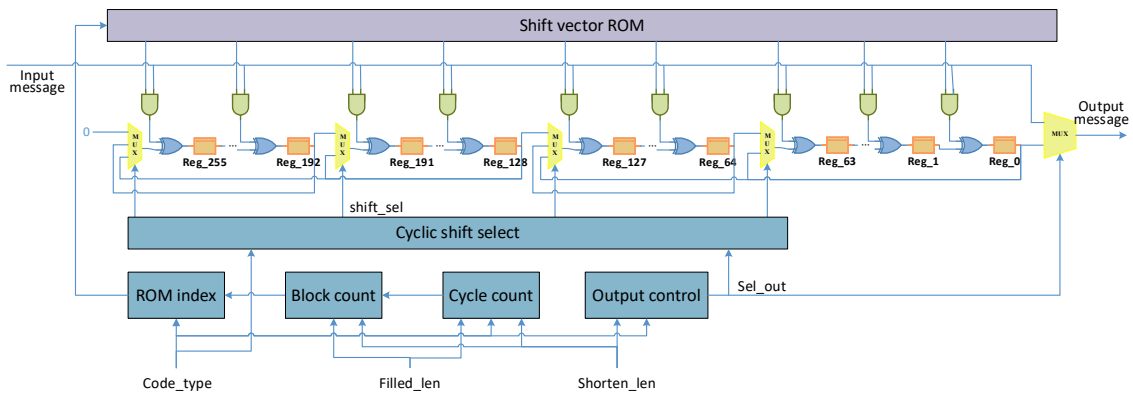


Figure 1. Rate-compatible QC-LDPC encoder architecture.

According to the input parameter *Code_type*, the encoder obtains three coding parameter values by looking up the table, namely, the block size of the generator matrix L , the number of rows of the generator matrix R and the number of parity rows C ; When the information is input, the cycle counter and block counter count the current cycle of sub-matrix and row blocks. The cycle counter starts from $(N_f \bmod L)$, where N_f is the number of virtual filled bits, and adds 1 bit count per input. When the count value is $L-1$, the cycle counter continues counting from 0. The initial count value of the block counter is N_f / L . When the cyclic count value is equal to $L-1$, the block count value is added to 1, and End count at $R-1$. The encoder completes the input and verification calculation of N_f bits by changing the initial count value, which can effectively realize the virtual filling of the code.

The check calculation circuit multiplies the input information bit and the shift vector output by the shift vector ROM, and adds them to the output value of the previous shift register and then stores them into the current shift register. The shift vector ROM calculates the shift vector index of the current information block according to the block count value, and the correct shift vector is obtained through the index. In the check calculation circuit, the position of the previous shift register may change with the code rate and the current block count. The cyclic shift selection circuit controls the selection relationship of the shift register according to the code type and the output control signal *Sel_out*, and produces the selection signal *shift_sel* of the cyclic shift register. When *Sel_out* is equal to 0, that is to say, the check bit calculation is not completed. If the current index x of the register satisfies $x=L \times i-1$, i is any positive integer less than $C-R$, the register with the index $L \times (i-1)$ is selected as the previous register, otherwise the register with the index $x-1$ is selected. The check calculation and signal selection circuits compatible with 64 and 128 cyclic submatrices are given in Figure 1.

The check calculation circuit includes P_{max} binary multipliers, P_{max} binary adders, P_{max} registers and several multiplexers, where P_{max} is the maximum check bit length in all coding types. The verification module multiplies each P_{max} bit shift vector with the information bits entered by the current encoder, and the result of the multiplication is added to the output value of the previous shift register and sent to the current shift register. The *shift_sel* signal controls the switch of the multiplexer,

and the position of the multiplexer is determined by the different code type parameters of the implementation. A multiplexer is added to the front of the x th shift register, where x satisfies $x=L \times i-1$, i is any positive integer less than $C-R$.

The output of the codeword is controlled by the signal `shift_sel`. When the block count value is less than or equal to $C-1$, the `Sel_out` output 0 value makes the encoder output the information bit. Otherwise, the `Sel_out` output 1 value, the shift register is shifted right, and the check bit is shifted outward from `Reg_0` one by one. Until the block count is equal to $(R-1-\text{Shorten_len})/L$ and the block count is equal to $(L-1-\text{Shorten_len}) \bmod L$, the output of the check bit ends, where `Shorten_len` denotes the number of shortened bits.

3.2 Storage of cyclic shift vector

The proposed encoder stores the first row vector of each block of different QC-LDPC code generator matrices in shift vector ROM, so as to complete the compatibility of the calibration calculation circuit with multiple codes. The control circuit obtains the ROM index by `Code_type` and the block count value numerical calculation, which is the read address of shift vector ROM and generates the shift vector required for the current encoding.

The proposed encoder stores the first row vector of each block of different QC-LDPC code generation matrices in shift vector ROM, so as to complete the compatibility of the calibration calculation circuit with multiple codes. The control circuit calculates the ROM index according to `Code_type` and block calculation, which is the read address of shift vector ROM and generates the shift vector required for current encoding.

The bit width of shift vector ROM is equal to P_{max} , which is the maximum parity length of all coding types. We use L_i to represent the block size of the generator matrix of the i th LDPC code, R_i to represent the block number of the parity matrix row of the i th encoding type, and C_i to represent the block number of the information row of the i th encoding type. Then the check length of the i th LDPC code is $L_i \times (R_i - C_i)$. The shift vector ROM is stored in C_i shift vectors corresponding to each coding type from 0 address. The starting bit of the shift vector is right aligned, and 0 bit is filled on the left side when $L_i \times (R_i - C_i)$ is less than the ROM bit width. When storing C_i shift vectors corresponding to a coding type, the shift vectors of each information block of the generated matrix are from left to right, and the storage address increases in turn. When storing C_i shift vectors of a coding type, each information block of the generator matrix are from left to right, and the storage address increases in turn.

The ROM start reading address `Addr_start` of the current LDPC type is the sum of the blocks of all the information rows of the previous encoding type, namely $\text{Addr_start} = \sum C_i, i < \text{Code_type}$. The ROM index of the shift vector of the current block is equal to the sum of `Addr_start` and the block count B_c , namely $\text{ROM_index} = \sum C_i + B_c, i < \text{Code_type}$.

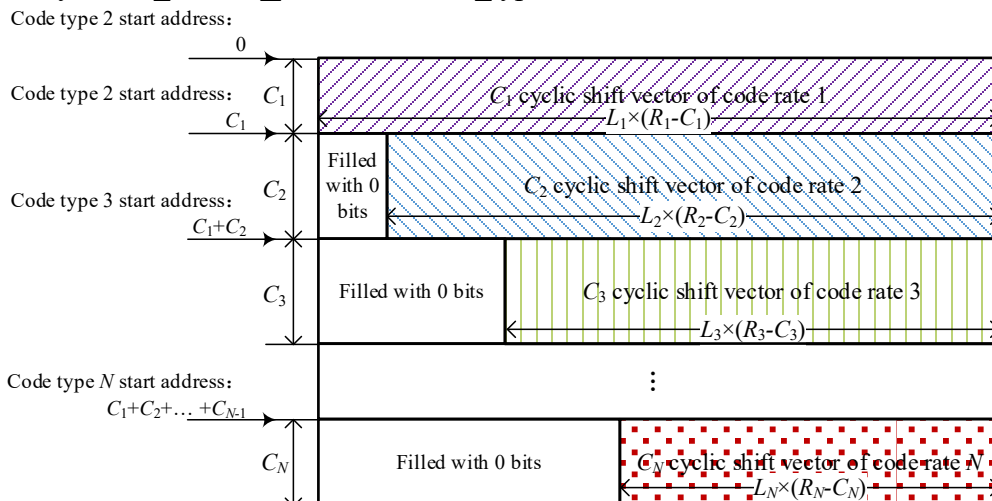


Figure 2. Storage of cyclic shift vector

4. FPGA implementation results

A high-speed rate-compatible LDPC encoder is implemented on FPGA using the method proposed in this paper. The encoding adopts four QC-LDPC codes in CCSDS, namely LDPC (1536, 1024), LDPC (2048, 1024), LDPC (6144, 4096) and LDPC (8192, 4096). Specific parameters are shown in table 1:

Table 1. Parameters of LDPC codes implemented.

LDPC code	Parameters		
	L	R	C
LDPC (1536,1024)	64	24	16
LDPC (2048,1024)	128	16	8
LDPC (6144,4096)	256	24	16
LDPC (8192,4096)	512	16	8

In Table 1, L is defined as the block size of the generator matrix, R is the number of rows of the generator matrix, and C is the number of parity rows.

A QC-LDPC encoder compatible with four code rates was implemented on Xilinx Virtex5 xc5vfx130t FPGA hardware platform, and the results were compared with those of the encoders with four fixed bit rates. Vivado 2018.2 was used for synthesis and layout. FPGA implementation results and resource occupancy of rate compatible encoder and four fixed rate LDPC encoders are shown in Table 2.

Table 2. FPGA implementation results of rate compatible encoder.

LDPC Encoder	Registers	LUTs	clock frequency
LDPC (1536,1024)	1058 (1%)	1043 (1%)	285Mbps
LDPC (6144,4096)	4132 (5%)	4090 (4%)	270Mbps
LDPC (2048,1024)	2081 (2%)	1292 (1%)	285Mbps
LDPC (8192,4096)	8214 (10%)	4378 (4%)	250Mbps
Total resources for four encoders	15485(18%)	10803(10%)	/
Rate-compatible encoder proposed	8633(10%)	5944(6%)	250Mbps

It can be seen from Table II that the resource consumption of the CCSDS fixed-rate encoder is proportional to the number of check bits of the code word. The maximum clock frequencies of different encoders are between 250M and 285M. If four encoders are used to implement four LDPC codes with different code rates, the total required Registers are 15485, and the required LUTs are 10803, accounting for 18 % and 10 % of the chip resources. Using the proposed rate-compatible encoder implementation method, since the cyclic shift check calculation circuit is effectively reused, the number of Registers and LUTs used by the rate-compatible encoder are 8633 and 5944, respectively, accounting for 10 % and 6 % of the chip resources. It should be noted that the implementation resources of rate-compatible encoder do not contain BLOCK RAM resources. The maximum ROM depth of the generated matrix cyclic shift vector is only 48. Due to the small depth, LUTs are used instead of BLOCK RAM resources in the implementation.

According to the implementation results in table II, compared with four different fixed rate encoders, the proposed rate-compatible encoder only uses about 56% Registers and 55% LUTs. At the same time, the rate-compatible encoder does not significantly reduce the maximum clock frequency, and the maximum clock frequency can reach 250 M.

5. Conclusion

The rate-compatible encoder proposed in this paper can effectively reuse the cyclic shift vector generation circuit and cyclic shift check calculation circuit, reduce the consumption of hardware resources, achieve high resource utilization, and achieve high clock frequency, which is suitable for the design of LDPC encoder system compatible with multiple quasi-cyclic production matrix structures.

Acknowledgments

This work was financially supported by National Natural Science Foundation of China under Grant 61801377.

References

- [1] Ali Dehghan; Amir H. Banihashemi. On the Tanner Graph Cycle Distribution of Random LDPC, Random Protograph-Based LDPC, and Random Quasi-Cyclic LDPC Code Ensembles. *IEEE Transactions on Information Theory*, 2018, 64(6), pp. 4438-4451.
- [2] Ahmed Mahdi, Nikos Kanistras, Vassilis Paliouras. A Multirate Fully Parallel LDPC Encoder for the IEEE 802.11n/ac/ax QC-LDPC Codes Based on Reduced Complexity XOR Trees. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. 2021, 29(1), pp. 51-64.
- [3] R. J. Yuan, B. M. Bai, and S. Tong. "FPGA-based design of LDPC encoder with throughput over 10 Gbps." *Journal of Electronics & Information Technology* 33.12 (2011): 2942-2947.
- [4] TM Synchronization and Channel Coding, CCSDS Std. 131.0-B-3, Sept. 2017.
- [5] Z. Li, L. Chen, L. Zeng, S. Lin, and W. H. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes," *IEEE Trans. On Communications*, vol. 54, no. 1, pp. 71-81, Jan. 2006.
- [6] F. Wang, P. Zhang, X. Wan, J. Liu, "Design of a Multi-rate Quasi-Cyclic Low-Density Parity-Check Encoder based on Pipelined Rotate-Left-Accumulator Circuits," *The 2014 7th International Congress on Image and Signal Processing*, pp. 1105-1109.
- [7] Y. M. Lin, H. T. Li, M. H. Chung, and A. Y. Wu, "Byte-reconfigurable LDPC codec design with application to high-performance ECC of NAND flash memory systems," *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 62, no. 7, pp. 1794-1804, July 2015.
- [8] Dimitris Theodoropoulos, Nektarios Kranitis, Antonios 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.