LDPC Encoder Design and FPGA Implementation in Deep Space Communication

Lei Cheng
Institute of Communication Engineering
PLA University of Science and Technology
Nanjing, China

Guangxia Li
Institute of Communication Engineering
PLA University of Science and Technology
Nanjing, China

Hongpeng Zhu
Institute of Communication Engineering
PLA University of Science and Technology
Nanjing, China

Lei Yang
Institute of Communication Engineering
PLA University of Science and Technology
Nanjing, China

Abstract—According to the requirements of deep space communication channel coding and referring to the standard parameters of Consultative Committee for Space Data Systems (CCSDS), a high rate LDPC code scheme set for deep space communications is adopted.[1][2] Due to the Low density parity check (LDPC) codes' excellent error performance, it's necessary for Field Programmable Gate Array(FPGA) to achieve CCSDS-LDPC coding. In this context, the serial encoding of CCSDS deep space communication standard (1536,1024) LDPC code is realized on the Xilinx company's XC5VSX100 chip. More specifically, the details of the encoding process is shown in this context. This design can be applied to other rate of LDPC codec based upon deep space communication standard which is set by CCSDS flexibly.

Keywords-Deep Space Communicatio;CCSDS;LDPC;FPGA

I. INTRODUCTION

In deep space communications, due to the dramatic increase in communication distance, the loss of communication signals' propagation in free space is large, the signal to noise ratio of the received signal is very low and the signal strength handled by communication systems is extremely weak. Therefore, improving the power efficiency of the system is one of the most important issues needed to be considered when designing deep space communications systems. Channel coding is an effective way to improve the power efficiency. At present, it can be said that without the assistance of the efficient channel coding, it is difficult to achieve deep space communications in low SNR condition.[3]

The high rate LDPC codes is recommended in deep space communication system by Consultative Committee for Space Data Systems(CCSDS). In 2010, the LDPC codes was used in China's lunar exploration system, 'Chang-e II'. It is first time that the LDPC codes is applied to the field of deep space exploration.

This paper discuss in detail the design and implementation of LDPC codes by FPGA (Field Programmable Gate Array). The discussion is based on CCSDS 131.1-O-2 standard on deep space communication 2/3 rate. This paper is focused on solving the key in achieving time throughput and resource consumption problems. At the same time, the intrinsic link among CCSDS deep space communications standards for various rate LDPC codes is taken into account. The scalability of coding is considered in the design as well.[4]

II. THE CODEWORD CONSTRUCTION OF LDPC

LDPC codeword include Gallager[7] construction method, Mackay construction method and PEG construction method based on random construction method, this construction method is to search check matrix by the development of a certain girth or distribution rules. Finite geometry and algebra constructor portfolio construction method based on Method.

Quasi-cyclic LDPC code actually belongs to Gallager LDPC codes, but it is not a random column permutation. Quasi-cyclic LDPC code is a class of constructor code with low complexity encoding. Its complexity relate to the generation matrix. A codeword of quasi-cyclic code which cyclic shift $p(p>1)$ times is still a codeword of the code. QC-LDPC can be structured by quasi-cyclic matrix, the encoded code can be achieved by using a register, and the hardware implementation is easier than other LDPC. CCSDS standard provide a QC-LDPC codeword which used for deep space communications. The matrix H of LDPC codeword with each rate is as follows:

$$H_{1/2}=\begin{bmatrix} 0_M & 0_M & I_M & 0_M & I_M \oplus \Pi_1 \\ I_M & I_M & 0_M & I_M & \Pi_2 \oplus \Pi_3 \oplus \Pi_4 \\ I_M & \Pi_5 \oplus \Pi_6 & 0_M & \Pi_7 \oplus \Pi_8 & I_M \end{bmatrix}$$

$$H_{2/3}=\begin{bmatrix} 0_M & 0_M & | & \\ \Pi_9 \oplus \Pi_{10} \oplus \Pi_{11} & I_M & | & H_{1/2} \\ I_M & \Pi_{12} \oplus \Pi_{13} \oplus \Pi_{14} & | & \end{bmatrix}$$

$$H_{3/4}=\begin{bmatrix} 0_M & 0_M & | & \\ \Pi_{15}\oplus\Pi_{16}\oplus\Pi_{17} & I_M & | & H_{2/3} \\ I_M & \Pi_{18}\oplus\Pi_{19}\oplus\Pi_{20} & | & \end{bmatrix}$$

In this paper, LDPC codeword is the QC-LDPC code CCSDS 131.1-O-2 protocol, 1536 bits of the code length, 1024 bits of information length, rate 2/3, sub-matrix size is 256 × 256, M is 256. A parity check matrix of the LDPC code is configured in $H_{2/3}$ above.[1]

In $H_{2/3}$, 0_{256} is the zero matrix of 256 × 256, I_{256} is the unit matrix of 256 × 256. Π_k is a permutation matrix of 256 × 256, its i-th row is a column, and the others are 0, the value of $\pi_k(i)$ can be get in the follow.

$$\pi_k(i)=64\times\left(\left(\theta_k+\lfloor i/64\rfloor\right)\bmod 4\right)+\left(\phi_k\left(\lfloor i/64\rfloor\right)+i\right)\bmod 64$$

The values of θ_k and ϕ_k is in the following table。

TABLE I. THE DESCRIPTION OF θ_k AND ϕ_k

k	θ_k	$\phi_k(0)$	$\phi_k(1)$	$\phi_k(2)$	$\phi_k(3)$
1	3	59	0	0	0
2	0	18	32	46	44
3	1	52	21	45	51
4	2	23	36	27	12
5	2	11	30	48	15
6	3	7	29	37	12
7	0	22	44	41	4
8	1	25	29	13	7
9	0	27	39	9	2
10	1	30	14	49	30
11	2	43	22	36	53
12	0	14	15	10	23
13	2	46	48	11	29
14	3	62	55	18	37

In this paper, H matrix can be obtained by calculate:

$$H=\begin{bmatrix}
-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&0&-1&-1&-1&-1&-1&-1&-1&0&-1&-1&59\\
-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&0&-1&-1&-1&-1&-1&-1&0&0&-1&-1\\
-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&0&-1&-1&-1&-1&-1&-1&0&0&-1\\
-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&-1&0&-1&-1&-1&-1&-1&-1&0&0\\
27&30&43&-1&0&-1&-1&-1&0&-1&-1&-1&0&-1&-1&-1&-1&-1&-1&-1&0&-1&-1&-1&18&52&23&-1\\
-1&39&14&22&-1&0&-1&-1&-1&0&-1&-1&-1&0&-1&-1&-1&-1&-1&-1&-1&0&-1&-1&-1&32&21&36\\
36&-1&9&49&-1&-1&0&-1&-1&-1&0&-1&-1&-1&0&-1&-1&-1&-1&-1&-1&-1&0&-1&27&-1&46&45\\
30&53&-1&2&-1&-1&-1&0&-1&-1&-1&0&-1&-1&-1&0&-1&-1&-1&-1&-1&-1&-1&0&51&12&-1&44\\
0&-1&-1&-1&14&-1&46&62&0&-1&-1&-1&-1&-1&11&7&-1&-1&-1&-1&22&25&-1&-1&0&-1&-1&-1\\
-1&0&-1&-1&55&15&-1&48&-1&0&-1&-1&29&-1&-1&30&-1&-1&-1&-1&-1&44&29&-1&-1&0&-1&-1\\
-1&-1&0&-1&11&18&10&-1&-1&-1&0&-1&48&37&-1&-1&-1&-1&-1&-1&-1&-1&41&13&-1&-1&0&-1\\
-1&-1&-1&0&-1&29&37&23&-1&-1&-1&0&-1&15&12&-1&-1&-1&-1&-1&7&-1&-1&4&-1&-1&-1&0
\end{bmatrix}$$

In the encoder, frame is the unit of information. The frame length of the input information is 1024 bits. When the coding is completed, the frame length is 1536 bits.

III. LDPC CODING ALGORITHM

Richardson proposed an efficient coding method for randomly constructed codes, which can effectively solve computational problem, the core idea of the algorithm is to assume $M\times N$ dimensional random constructed parity check matrix transformation through the ranks, with the following changes to the structure of the approximate lower triangular matrix form. As shown in FIG.

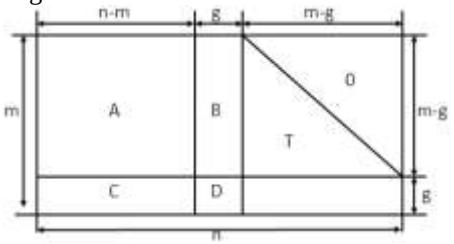

Figure 1. The structure of the approximate lower triangular matrix

The size of each sub-matrix of A respectively is $(m-g)\times(n-m)$, B is $(m-g)\times g$,C is $g\times(n-m)$, D is $g\times g$. Codeword set $C=(s,p_1,p_2)$, where S is the information bit, parity bit sequence represented p_1 and p_2 together. p_1 contains g bits, p_2 m-g bits.

Switch the rows and columns of the matrix, the elements do not change, therefore, the converted approximate lower triangular matrix remains the sparsity. Guarantee transformation T is full rank, the diagonal elements of T are all "1". Multiply left the matrix of the transformed with matrix

$$\begin{bmatrix} I & 0 \\ -ET^{-1} & I \end{bmatrix}$$

You can get:

$$\begin{bmatrix} I & 0 \\ -ET^{-1} & I \end{bmatrix}\begin{bmatrix} A & B & T \\ C & D & E \end{bmatrix}=\begin{bmatrix} A & B & T \\ -ET^{-1}A+C & -ET^{-1}B+D & 0 \end{bmatrix}$$

According to $HC^T=0$, you can get the following two equations:

$$As^T+Bp_1^T+Tp_2^T=0$$

$$(-ET^{-1}A+C)s^T+(-ET^{-1}B+D)p_1^T=0$$

If the matrix $-ET^{-1}B+D$ is invertible, we can obtain:

$$p_1^T = -(-ET^{-1}B + D)^{-1}(-ET^{-1}A + C)s^T$$

First p_1^T calculate, then according to the following formula, p_2^T can be calculated.

As for matrix computation takes too many resources in the FPGA, especially matrix inversion, so it is difficult to realize the algorithm in hardware. In this paper, the design of IP core will solve the problem. The inversion results will be stored in RAM, which can save a lot of resources.

In this paper we use the encoding method as follows:

a) H is a matrix of size $3M \times 7M$. The matrix H is divided into two portions P and Q, where P's size is $3M \times 3M$, Q's size is $3M \times 4M$ .M is 256.
b) W is a matrix of size $4M \times 3M$.
c) G matrix composed by the matrix W and the unit matrix. $G=[I_{4M} \quad W]$

IV. LDPC CODE ENCODER DESIGN

The following brief analysis the structure of the encoder. From the figure, we may find the structure do parity calculations by using cyclic shift register[5].

a) Initially, first row of G matrix were loaded in 2M/m shift registers by size of m = M/4, then phase the value of the register and the input bit, put the results into a 2M wide parity bit register.
b) Put in the twice bit, the 2M/m shift registers cycle one shift, then phase with the current input bit, XOR the results and data in the parity bit register, and then put the results into the 2M wide parity bit register.
c) When shift M/4 times, reload the first line value of the second matrices, repeat the above operations.

When coded bits are fully put in, parity bits are generated, so the coding structure works with high throughput.

The encoder structure may be achieved by using a simple shift register, [6] the structure shown in the following FIG.

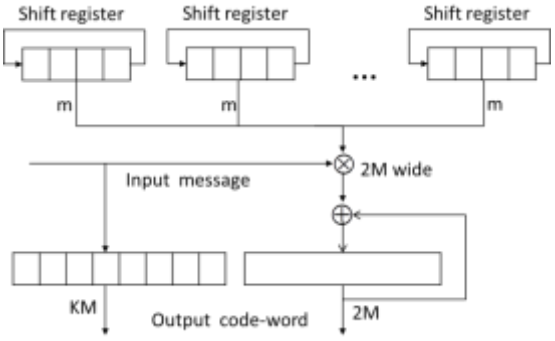

Figure 2. The encoder structure

The encoding algorithm based on a generator matrix:

$$c = S \cdot G = S \cdot (I \mid B) = (S \mid S \cdot B)$$

Where S is the input information bits, matrix I is the unit matrix, matrix B corresponding to the cyclic shift section of generator matrix G. when doing multiplication between vector and non-sparse matrix B, B can be split into cyclic shift matrix for computing, storage array rotate only need to store the first row, the use of circular shift register can get to the next line.

The design of IP core

In this paper the program FPGA use slice resource to save RAM resources. Therefore, only one RAM is used in this paper, for writing the cyclic shift matrix.

From the data sheet of IP core, the type of data file formats is .coe, contains two parameters:

(1), memory_initialization_radix, namely vector base, which can be 2,10,16, behalf the data of binary, decimal, hexadecimal , we use a hex ;

(2), memory_initialization_vector, initialization data based on vector. Among them, the number of vectors should be consistent with the generator ROM size of IP core.

Based on the generated matrix W, the data depth is 128.

Interface block diagram of the encoder is shown in the following FIG. Encoder 1024 information bits to generate encoded 1536 bits. The encoder uses a systematic code, the first 1024 bits of information encoded are the same of the input bits, the next 512 bits are the parity bit.

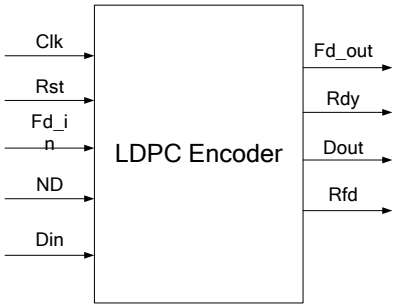

Figure 3. Interface block diagram of the encoder

Below is the encoder timing diagram:

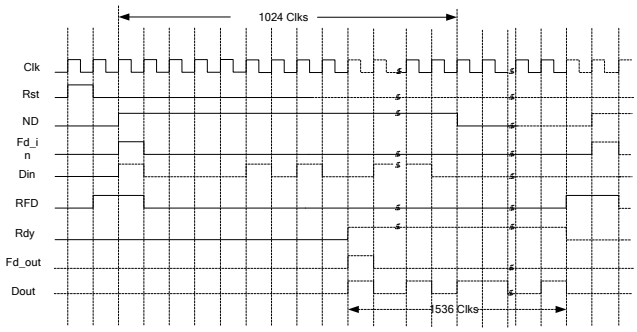

Figure 4. the encoder timing diagram

V. FPGA IMPLEMENTATION RESULTS

We achieve the CCSDS standard LDPC encoder On the XC6SLX100 chip of Xilinx company. The resource consumption of encoder shown in the table:

TABLE II. **DEVICE UTILIZATION SUMMARY**

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	2660	126576	2%
Number of Slice LUTs	1257	63288	1%
Number of fully used LUT-FF pairs	1081	2836	38%
Number of bonded IOBs	9	326	2%
Number of Block RAM/FIFO	1	268	0%
Number of BUFG/BUFGCTRLs	1	16	6%

Partial results of the encoder operation shown in the FIG .

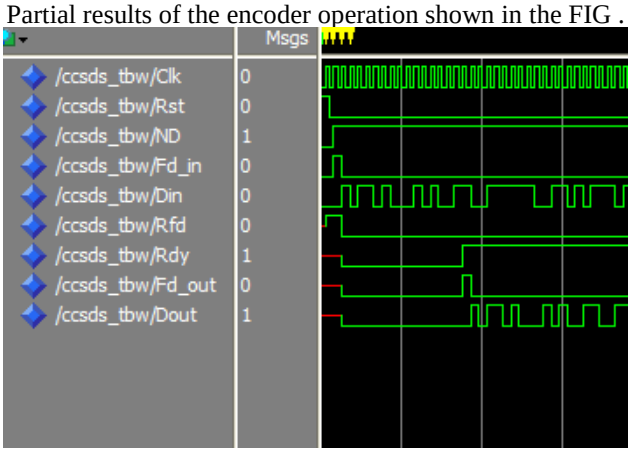

Figure 5. Partial results of the encoder operation

Wherein the input information bits is Din, Dout is the output from the information bits. Encoder implemented coding delay is : the first one encoder output coded bits delayed 16 clock cycles than the input first one information bit.

Throughput is calculated as follow:

Throughput=$(n_{LDPC}/C_p)\times f_{max}$

Where, f_{max} is the maximum operating frequency of the encoder which can be achieved, n_{LDPC} is the number of information bits of one frame of the LDPC codes, C_p is the number of the main encoder clock cycles to complete a frame of encoded data.

The encoder can reach a maximum operating frequency of 155MHz. For a coded frame, from the first input data to the last encoded data, 1553 cycles are needed, the value of C_p is 1553. The number of information bits is 1024, the throughput is 102.2 Mbps. Assuming the encoder working frequency of 100MHz, the throughput frequency of the encoder can achieve 65.9Mbps.

VI. CONCLUSION

The specific cyclic symmetry features makes it easier for QC-LDPC codes to implement on FPGA. This paper take (1536,1024) LDPC code for example, considering high-speed , high efficiency , performance, portability and other factors, to achieve a kind of realization structure of QC-LDPC encoder which is suitable for deep space communications CCSDS standards proposed[6][10]. It is shown in Section 1 that the structures of standard other rate's H matrix of LDPC and 2/3 rate's H matrix have high consistency. Therefore, each sub-module and interleaved iterative structure in this design can be easily extended to the implementations of other rate LDPC[8]. Comprehensive results show that: in the case of small foot print, the encoder can achieve higher data throughput. Thus it lays the foundation of the practical application of LDPC codes[1]. The implementation structure discussed in this article has a good engineering application in terms of both performance and resource consumption. As a result, it has a better prospect for the application. These aspects provide the basis of LDPC codes' application in deep space communications[10].

REFERENCES

[1] CCSDS. Low density parity check codes for use in near-earth and deep space applications [S]. CCSDS 131.1-O-2, Orange Book, Washington DC, USA, 2007.

[2] Klinc D, Ha J, McLaughlin S W, et al. LDPC codes for the Gaussian wiretap channel[J]. Information Forensics and Security, IEEE Transactions on, 2011, 6(3): 532-540.

[3] Pusane A E, Smarandache R, Vontobel P O, et al. Deriving good LDPC convolutional codes from LDPC block codes[J]. Information Theory, IEEE Transactions on, 2011, 57(2): 835-857.

[4] Chen X, Kang J, Lin S, et al. Accelerating FPGA-based emulation of quasi-cyclic LDPC codes with vector processing[C]//Proceedings of the Conference on Design, Automation and Test in Europe. European Design and Automation Association, 2009: 1530-1535.

[5] Smarandache R, Vontobel P O. Quasi-cyclic LDPC codes: Influence of proto-and Tanner-graph structure on minimum Hamming distance upper bounds[J]. Information Theory, IEEE Transactions on, 2012, 58(2): 585-607.

[6] Wang Z, Cui Z. Low-complexity high-speed decoder design for quasi-cyclic LDPC codes[J]. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2007, 15(1): 104-114.

[7] GALLAGER R G. Low Density Parity Check Codes[J]. IRE Transaction on Information Theory, 1962, 8(1) : 21−28.

[8] Darabiha A, carusone A c, Kschischang F R. Block-Interlaced LDPC Decoder with Reduced Interconnect Comple xity[J]. 1EEE ansactions on circuits and systems II, 2008

[9] MacKay DJ c. Good Error-Correcting Codes Based on Very sparse Mat rices[J]. IEEE Transactions on Information Theory, 1999,45(2):399−431

[10] Andrews K, Dolinar S, Divsalar D, et a1. Design of Low Density Parity Check(LDPC)Codes for Deep Space Applications. Pasadena IPN,2004