LOW DENSITY PARITY CHECK CODES FOR TELEMETRY APPLICATIONS

Bob Hayes
L-3 Communications Cincinnati Electronics

ABSTRACT

Next generation satellite communication systems require efficient coding schemes that enable high data rates, require low overhead, and have excellent bit error rate performance. A newly rediscovered class of block codes called Low Density Parity Check (LDPC) codes has the potential to revolutionize forward error correction (FEC) because of the very high coding rates. This paper presents a brief overview of LDPC coding and decoding. An LDPC algorithm developed by Goddard Space Flight Center is discussed, and an overview of an accompanying VHDL development by L-3 Communications Cincinnati Electronics is presented.

KEY WORDS

Low Density Parity Check Codes, LDPC, Block Codes, Error Correction Codes, Forward Error Correction

I. INTRODUCTION

Forward error correction coding (FEC) is an integral part of nearly every communication system. Satellite communication systems have utilized a variety of error correction schemes for years using a variety of algorithms, many of which have been endorsed by the Consultative Committee for Space Data Systems (CCSDS). Among the popular FEC algorithms are Reed-Solomon coding, convolutional coding, and concatenated coding schemes.

Recent research has focused on innovative new ways of encoding data. The objective of any coding algorithm is to minimize the bit error rate (BER) of the encoded data stream, maximize the coding gain, and maintain the maximum possible data rate. A promising code, first developed in 1963 by Gallager and recently rediscovered by coding theorists, is known as the Low-Density Parity-Check Code (LDPC) [1]. LDPCs are block codes that have been shown to achieve very high data rates and excellent coding gain, and they can be constructed according to a variety of efficient encoding algorithms.

This paper provides a brief tutorial on the theory of LDPCs. Further, the mathematical basis for an experimental CCSDS realization of LDPCs based on quasi-cyclic LDPCs is presented. This

quasi-cyclic LDPC encoder is being implemented by L-3 Communications Cincinnati Electronics in both a low-cost Field Programmable Gate Array (FPGA) for missiles and aircraft, as well as in a radiation hardened FPGA for spacecraft.

This paper is organized as follows. The basic theory of LDPCs is presented in Section II. Quasi-cyclic LDPCs are explained in Section III. An experimental CCSDS LDPC encoder is described in Section IV. A discussion of an LDPC VHDL development at L-3 Cincinnati Electronics is presented in Section V.

II. LDPC THEORY

General Block Codes

Every block code can be represented as a mathematical matrix product. Consider k bits of raw information that are to be encoded and transmitted over a communication channel. A block code of size (n, k) operates on the k information bits and generates $n > k$ bits for transmission. The extra ($n - k$) bits added to the data stream by the block coder are parity bits which add redundancy to the code. Redundancy provides the ability to reduce errors which can occur during transmission. The rate of the code is k/n. The objective of every block code is to prevent as many errors as possible while introducing as little redundancy as possible, making the ratio k/n approach one.

Any block code can be described either by a polynomial representation or a matrix multiplication. Although some of the more advanced coding properties of block codes can be derived more cleanly with a polynomial notation, the matrix notation is especially useful for the purposes of this discussion because for certain block codes it gives rise to a special graphical representation called a Tanner graph. Mathematically, denote the 1 x k vector of raw (i.e., uncoded) information bits $\mathbf{m} = [m_0, m_1, ..., m_{k-1}]$. Let the output of the coding algorithm be a vector $\mathbf{c}$ of size 1 x n, $\mathbf{c} = [c_0, c_1, ..., c_{n-1}]$. Define the input/output relationship between $\mathbf{c}$ and $\mathbf{m}$ by a k x n generator matrix $\mathbf{G}$, described by the following relationship [2]:

$$\mathbf{c} = \mathbf{mG} \quad (1)$$

The matrix $\mathbf{G}$ is called the generator because it provides a mapping from any input sequence to a distinct output sequence. Therefore, defining the matrix $\mathbf{G}$ is sufficient to define the encoding procedure in its entirety. Once the generator matrix $\mathbf{G}$ is defined, decoding of the encoded message at the receiver is accomplished by a parity check matrix $\mathbf{H}$ of dimension n x k. For simple block codes such as Hamming Codes, decoding is accomplished via the use of syndrome computation, which relies on the fact that the rows of $\mathbf{G}$ lie in the null space of the parity check matrix $\mathbf{H}$. The syndrome, $\mathbf{s}$, is a k x 1 vector calculated as

$$\mathbf{s} = \mathbf{rH}, \quad (2)$$

where **r** is the noise-corrupted received sequence. If the syndrome is the all-zeros vector, it is assumed that no errors occurred. If **s** is not all-zeros, then the syndrome patterns correspond to specific error patterns that can be detected by the block code.

Low Density Parity Check Codes

With so many different block coding schemes in existence, new coding algorithms must demonstrate superior performance in order to gain acceptance in industry. One such "new" coding scheme is known as a Low Density Parity Check Code (LDPC). LDPCs have existed for many years but they have only achieved widespread popularity in recent years. These codes exhibit very desirable properties for highly congested frequency bands. Their performance can be very close to the theoretical Shannon capacity, requiring potentially little coding overhead to achieve very low BERs at very high data rates. For example, coding rates of $\frac{7}{8}$ and $\frac{15}{16}$ have been demonstrated with high coding gain. Further, LDPCs can be constructed in such a way as to utilize a great deal of parallelism in hardware implementations, reducing hardware complexity and hence reducing the cost of hardware designs. Because of these advantages, LDPCs are sometimes used in digital data storage systems, since they introduce minimal redundancy while operating at data rates in the region of gigabytes per second [5].

In addition to having a generator **G** and parity check matrix **H**, LDPCs are often represented by a Tanner graph. The Tanner graph can be derived from the generator matrix, **G**, or the parity check matrix, **H**. Tanner graphs are comprised of two separate types of nodes: check nodes and variable nodes. Each connection between a check node and a variable node denotes a '1' in the parity check matrix. An example of such a graph is given in Figure 1.

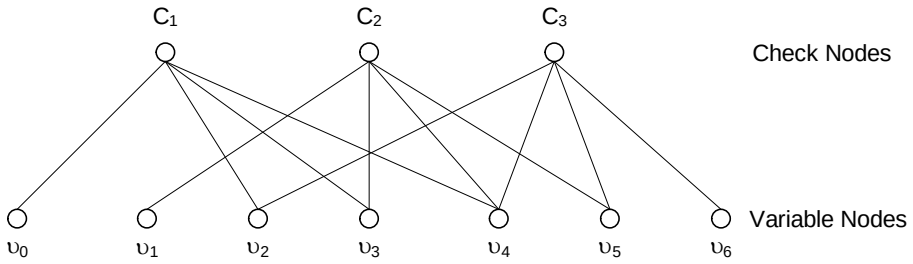

Figure 1. Example Tanner Graph.

This Tanner graph corresponds to the parity check matrix in (3):

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (3)$$

The properties of an LDPC code are derived from its Tanner graph. For example, LDPCs typically have more coding power if the minimum length cycle through the graph, called the girth, is maximized. This enables the decoding system to be optimized.

A defining feature of LDPCs is that the parity check matrix, **H**, is a sparse matrix. By definition, this means that there are very few 1s in **H**. This implies that the parity check matrix of (3) is not an LDPC, since the ratio of 0s to 1s is less than 1. The locations of the 1s in the parity check matrix are determined by desired properties of the resultant code, and are usually selected to ensure a given structural property for decoding. LDPCs can be classified into two general categories [3]:

1) Random LDPC codes are generated by computers according to desired structural guidelines. Computer search routines can be written to generate LDPCs such that their Tanner Graphs satisfy certain constraints. The resulting codes are called random because the resulting locations of the 1s are usually not sufficiently structured to enable simple encoding.
2) Structured LDPC codes are generated based on coding theory and combinatorics. These codes can have suboptimal coding properties relative to random codes, but structured codes have simpler encoding algorithms. For this reason, structured LDPC codes are simpler to implement in hardware for complexity-constrained applications, such as telemetry missions.

Since structured LDPCs have a more concise representation than random LDPCs or computer-generated LDPCs, structured LDPCs are far better suited to implementation for missiles, aircraft, or spacecraft. Structured LDPCs also can have simpler decoders than random codes, since the decoder can take advantage of symmetry to create a more modular decoder.

A specific type of LDPC of interest for missile and spaceflight applications is the quasi-cyclic (QC) LDPC, which has desirable structural properties suitable for efficient VHDL implementation.

LDPC Advantages

Bit Error Rate. If designed properly, LDPCs can achieve very low BER while requiring low coding overhead when compared with other coding schemes. Of particular interest when comparing LDPCs with other coding methods is the "waterfall curve," which is a graph of how rapidly the error rate decreases as a function of SNR. A measure of the effectiveness of a coding scheme such as LDPC is the steepness of this curve – a very steep curve enables the system designer to declare a critical SNR threshold beyond which transmission is quasi-error-free. As shown in Figure 2, the waterfall curves of LDPCs can be very steep, indicating excellent BER performance once a critical SNR threshold (called the "knee" of the curve) is crossed. Figure 2 illustrates several coding schemes of interest to the telemetry and spacecraft community, including three different LDPCs (one by Goddard Space Flight Center, two by JPL), a Reed-Solomon code, a convolutional code, and a concatenated code.

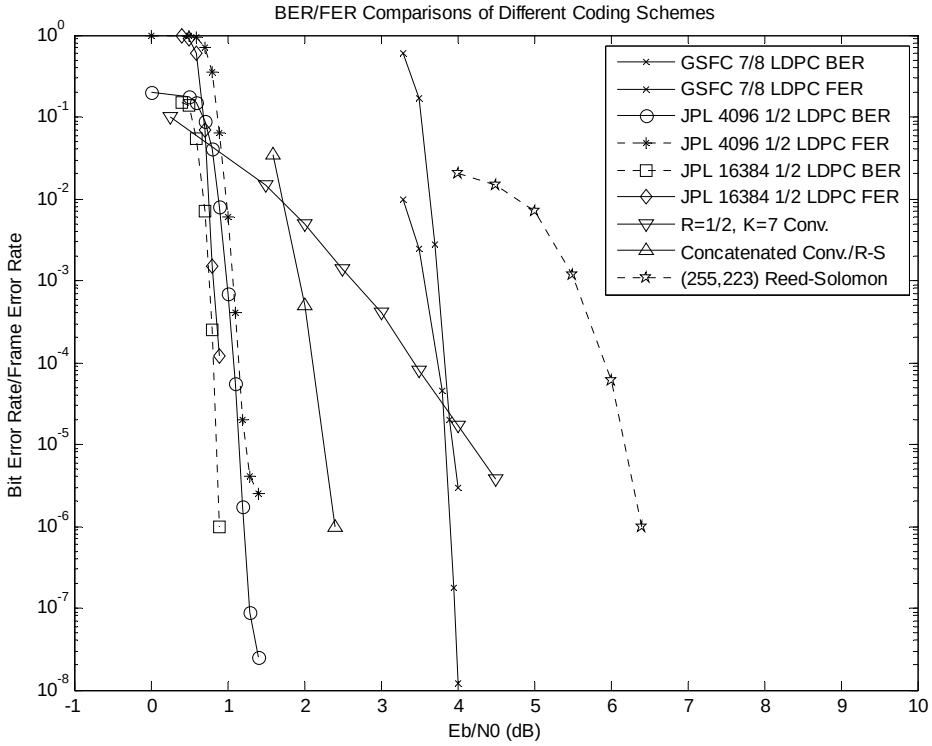

Figure 2. Waterfall Curves For Various Coding Schemes.

Low Overhead. It is noteworthy that the LDPC codes of Figure 2 achieve excellent BER performance by using relatively little coding overhead. For example, the "GSFC 7/8 LDPC" code has a waterfall curve with a threshold between 3.5-4.0 dB SNR. Although the GSFC code requires a higher SNR to reach the knee of the curve than many of the other coding schemes, the GSFC code is a rate 7/8 code and uses substantially less bandwidth for coding overhead. This is one of the tradeoffs in the code design process.

Low Error Floor. Error correction codes can have an "error floor," which is a change in slope of the waterfall curve. Error floors are undesirable, and often codes are designed with the requirement that the code either eliminate an error floor or suppress that floor below a very low threshold. For example, in Figure 2, the JPL rate ½ LDPC code with an input block length of 4096 bits shows an error floor around 10^{-8} FER. In this region, the BER curve (and frame error rate curve, FER) begins to level off around 1.4 dB SNR. The sources of this error floor have been discussed in the literature, and the reader is referred to [8] for a discussion on current techniques of error floor reduction.

LDPCs can have extremely low error floors, which is another one of their chief advantages over competing coding schemes. The JPL LDPCs have error floors below 10^{-6}, which is suitable for many far-earth applications. Near-earth/telemetry missions can have even more strict requirements for error floor, and the GSFC LDPC has been shown to have an error floor below 10^{-10}.

Decoding

The properties of any code are highly dependent upon the decoding algorithm used at the receiver. LDPCs are designed to work with a very specific type of decoding algorithm. The reason that a low density **H** is desirable is a function of the decoding algorithm, as first specified by Gallager [1]. LDPC decoding can be accomplished by computing *a posteriori* probabilities on a decoding tree for each bit. The decoding algorithm operates iteratively, meaning that it loops through a decoding process several times until an exit criterion is reached. It has been shown in the literature that use of iterative decoding for LDPCs yields BER curves that decrease exponentially as the length of the block of input data increases [5]. The specific type of decoding algorithm assumed for all results in this paper is called the sum product algorithm.

The decoding process is known as "belief propagation," and it ties very closely with the parity check matrix **H** and the associated Tanner graph. The decoding process can be viewed as having two distinct steps – a horizontal step (horizontal from the viewpoint of the parity check matrix **H**) and a vertical step. Each pair of steps constitutes a cycle. After each cycle, a best guess for the frame of data is generated and a parity check syndrome is computed.

- If the syndrome is the all-zeros vector, this indicates that either 1) no error occurred, or 2) an undetected error occurred. The system assumes that the decoded message is correct, and the decoder operates on the next block of data.
- If the syndrome is not the all-zeros vector, this indicates the guess was not correct, in which case the decoder iterates and executes another cycle of the algorithm. Generally, there is a maximum number of iterations that are permitted, and if the syndrome never goes to zero, that block of data is known to contain a detected error.

III. QUASI-CYCLIC LDPCs

Although FEC decoding algorithms are often considered to be more complex than encoders, in the case of LDPC codes the encoder is of special interest. It has been shown that the complexity of the LDPC decoder designed with the sum product algorithm grows linearly with the block length of the code [7]. However, the encoder complexity growth is quadratic because it involves a very computationally intensive matrix multiplication procedure. Because of this quadratic complexity, methods of reducing computations at the encoder are of great practical interest. This section addresses a method of decreasing encoder complexity by introducing a repeatable structure into submatrices of the generator matrix, **G**.

Within the class of structured LDPCs, a popular subclass of codes is called quasi-cyclic LDPCs. As described below, quasi-cyclic codes possess a structure that enables the encoder to operate on large blocks of data without having to store a large generator matrix. They also offer the advantage of simple shift-register based arithmetic during the encoding procedure. A brief description is given below.

Define the output of Equation (1) to be a codeword space given by $\mathbf{c} = [c_0, c_1, ..., c_{n-1}]$. A code is *cyclic* if and only if for every code **c**, the left (or right) cyclic shift of **c** is also a code. For

example, the vector $\mathbf{c}' = [c_{n-1}, c_0, c_1, ...c_{n-2}]$ would be a code in the above example. In this case it is commonly said that **c'** is a right cyclic shift of **c**. Whenever a cyclic code **c** is generated from a generator matrix **G**, the shifted nature of **c** implies a cyclic property in the generator matrix. In turn, this implies a cyclic property in the parity check matrix **H**. A matrix is said to be cyclic if every row is a right or left shift of its adjacent rows. A quasi-cyclic code has a generator matrix **G** that is not truly cyclic, but it has blocks that are cyclic. The following discussion on QC-LDPCs is based on the presentations of [3] and [5] in relation to the Goddard LDPC of rate 7/8.

Consider a quasi-cyclic parity check matrix $\mathbf{H_{qc}}$ with the following structure:

$$\mathbf{H_{qc}} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & \cdots & \mathbf{A}_{1,t} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} & \cdots & \mathbf{A}_{2,t} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{c,1} & \mathbf{A}_{c,2} & \cdots & \mathbf{A}_{c,t} \end{bmatrix} \quad (4)$$

The matrix $\mathbf{H_{qc}}$ is composed of a series of square submatrices, $\mathbf{A}_{i,j}$, called circulants, each of which is cyclic. The matrix $\mathbf{H_{qc}}$ can be constructed such that the following two properties are always true:

- The weight (# of 1s in each row) of each circulant is small compared to its size
- No two rows (or columns) of $\mathbf{H_{qc}}$ have more than one 1-component in common (called the Row-Column, or RC, constraint)

These two constraints ensure that the parity check matrix has a low density of ones (hence the name, *low density* parity check codes) and that the Tanner graph has minimum girth properties. In a rough sense, this property of the Tanner graph is related to the minimum Hamming distance of the code – the greater the Hamming distance, the lower the BER of the code.

The advantage to encoding data with a quasi-cyclic LDPC (as opposed to an LDPC without a quasi-cyclic structure) is twofold. First, the encoding complexity can be made linear with the code length using shift registers. Second, the encoder and decoder routing complexity for integrated circuits is reduced [6].

IV. CCSDS QC-CODES BASED ON GODDARD ALGORITHM

Recently, Goddard Space Flight Center has proposed a QC-LDPC encoder. This is a general use code that has many desirable properties for applications including missiles, aircraft, and spaceflight. The details of this code are provided in [3] as well as in a newly released CCSDS Orange Paper [6]. In this section, we provide a brief overview of this code, its properties, and implementation details.

The code specified herein is based upon a (8176, 7154) LDPC code. This code is then shortened to (8160, 7136). The parity check base matrix is composed of submatrices $\mathbf{A}_{i,j}$. As in equation

(3), these submatrices are cyclic, and the resulting parity check matrix **H** is quasi-cyclic. In the Goddard code, each $\mathbf{A}_{i,j}$ is a 511 x 511 square circulant, and **H** is written as

$$\mathbf{H} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & \cdots & \mathbf{A}_{1,15} & \mathbf{A}_{1,16} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} & \cdots & \mathbf{A}_{2,15} & \mathbf{A}_{2,16} \end{bmatrix} \quad (5)$$

Each of the circulants $\mathbf{A}_{i,j}$ is sparse, and each row of each circulant has a Hamming weight of 2. The total weight for each row of **H** is 32 since there are 16 circulants per block row of **H**. The locations of the 1s for each circulant matrix is critical to the design of the code, and these locations are given in [6].

For encoding purposes, it is required to derive the generator matrix **G** from the parity check matrix described above. It has been shown that the generator matrix can be expressed in the following form:

$$\mathbf{G} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{B}_{1,1} & \mathbf{B}_{1,2} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{B}_{2,1} & \mathbf{B}_{2,2} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{B}_{3,1} & \mathbf{B}_{3,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{I} & \mathbf{0} & \mathbf{B}_{13,1} & \mathbf{B}_{13,2} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} & \mathbf{B}_{14,1} & \mathbf{B}_{14,2} \end{bmatrix} \quad (6)$$

Each of the $\mathbf{B}_{i,j}$ submatrices of (6) is a circulant matrix of size 511x511. Although the $\mathbf{B}_{i,j}$ submatrices are derived from the sparse parity check matrix **H**, the circulants of the generator matrix **G** do not share this sparse structure.

Some important features of (6) are detailed below:

- The generator matrix is in systematic form. The first 7154 bits to be generated by the LDPC encoder are therefore the original 7154 message bits. The final 1022 bits are the parity bits. Therefore the first 7154 bits can be passed through with no computations.
- The parity bit computation requires *all message bits*. Thus the parity bit computations cannot be complete until all message bits are present.
- To completely specify **G** in memory, all that is necessary is to specify the first row of each of the $\mathbf{B}_{i,j}$ submatrices. Since each submatrix is circulant, each subsequent row of each submatrix is a right circulant shift of the row above it. This reduces memory requirements.

Because of these features of the generator matrix **G**, a very efficient encoding scheme has been derived by GSFC. By loading the first row of each submatrix into a shift register and then performing the appropriate multiplications, parity bits can be generated. Initially there are 14 x 2 (28) shift registers, each of length 511 (as determined by the size of $\mathbf{B}_{i,j}$). This initial step produces two parity bits. After these two parity bits are generated, each shift register is right shifted and the same computations are performed, generating two more parity bits. This process

is repeated until 511 iterations have been performed, generating the 1022 parity bits. A block diagram of the process, provided by GSFC [9], is given below in Figure 3.

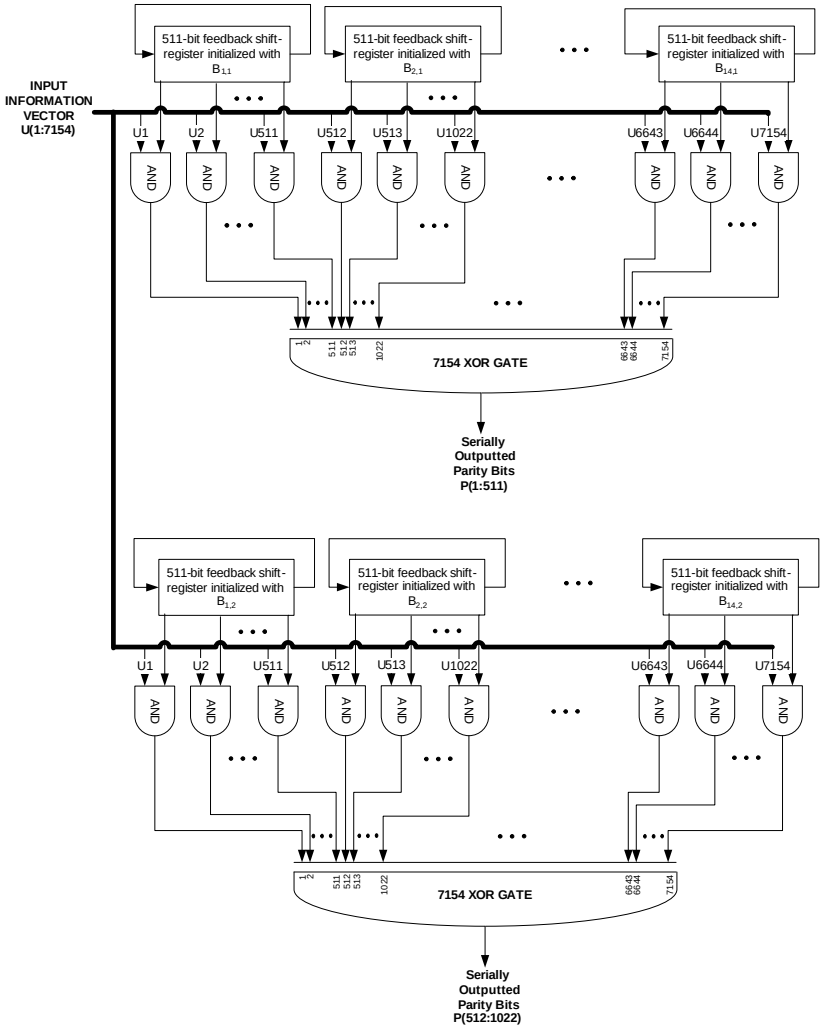

Figure 3. Encoding Block Diagram for GSFC LDPC Rate 7/8 Code.

The process shown in Figure 3 can be implemented efficiently in an FPGA because of the shift register multiplication.

V. L-3 CE HARDWARE DEVELOPMENT

L-3 CE is currently developing an FPGA-based implementation of the GSFC encoder based on the structure of Figure 3. This product will be available 3Q/4Q 2007. The L-3 CE FPGA implementation has advantages over an ASIC solution due to the tremendous cost of designing and qualifying an ASIC for spaceflight.

The FPGA implementation of L-3 CE is characterized by the following features:

- After shortening, input data block size is 7136 bits, output data block size is 8160 bits
- Rate 7/8
- Supports data rates up to 200 Mbps per encoder
- Targets both rad-hard and commercial FPGAs
- Uses external memory to buffer input data
- Uses flow control to throttle the data source when the buffer is full
- Implements CCSDS standard frame header
- Supports user-defined frame headers
- Supports encoder by-pass mode
- Error floor below 10^{-10}

In addition to this development, L-3 Cincinnati Electronics is also developing a JPL version of LDPC codes based on protograph theory. This development is concurrent with the GSFC development and will be available 4Q 2007.

CONCLUSIONS

This paper has presented some of the basic underlying mathematical concepts of LDPC codes. These codes exhibit excellent coding performance and, if properly designed, incur a minimal bandwidth penalty. A quasi-cyclic LDPC designed by Goddard Space Flight Center has been discussed, and the performance advantages of this code have been detailed in the context of both spaceflight and telemetry missions. This rate-7/8 code is being implemented in an FPGA by L-3 Cincinnati Electronics for telemetry and spaceflight missions.

REFERENCES

[1] R. G. Gallager, Low-Density Parity Check Codes. Cambridge, MA: MIT Press, 1963.

[2] S. Wicker, Error Control Systems for Digital Communication and Storage. Prentice Hall, 1995.

[3] Z. Li, L. Chen, L. Zeng, S. Lin, and W. Fong, "Efficient Encoding of Quasi-Cyclic Low-Density Parity-Check Codes," *IEEE Transactions on Communications*, Vol., 54, No. 1, pp. 71-81, January 2006.

[4] R. M. Tanner, "A Recursive Approach to Low Complexity Codes," *IEEE Transactions on Information Theory*, vol. IT-27, no. 9, pp. 533-547, September 1981.

[5] H. Song and B. Kumar, "Low-Density Parity-Check Codes For Partial Response Channels," *IEEE Signal Processing Magazine*, January 2004.

[6] "Low Density Parity Check Codes For Use In Near-Earth and Deep-Space Applications," CCSDS Experimental Specification CCSDS 131.1-O-1.

[7] Christian B. Schlegel and Lance Pérez, Trellis and Turbo Coding. IEEE Press, 2004.

[8] Stefan Laendner, Thorsten Hehn, Olgica Milenkovic, and Johannes B. Huber, "Two Methods for Reducing the Error-Floor of LDPC Codes," submitted for publication and available at http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0701006.

[9] NASA Goddard Technical Standard GSFC-STD-9100, Release Pending.