FPGA Implementation and Verification of LDPC Encoder with Weight (3, 6) Approximate Lower Triangular Matrix

Yi Hua Chen, Meilin Su, Jheng Shyuan He
Oriental Institute of Technology
Institute of Information and Communication Engineering
New Taipei city, Taiwan
ff011@mail.oit.edu.tw, meilin@mail.oit.edu.tw, hbljustin@hotmail.com

Abstract—Compared with general linear block code encoding, LDPC encoding with lower triangular check matrix and approximate lower triangular check matrix carry out encoding directly by parity check matrix \(H\). This study used the weight (3, 6) approximate lower triangular regular parity check matrix to implement the LDPC encoding on the 5641R FPGA of the Software Define Radio system developed by National Instruments (NI) [1]. This study provided a detailed introduction to the encoding mechanism of the approximate lower triangular LDPC, and completed the implementation and hardware verification of FPGA SDR system.

Keywords—LDPC; Approximate Lower Triangular; SDR.

I. LDPC INTRODUCTION

Shannon proposed the mathematical theories of communications in 1948 [2], arguing that the system capacity \(C\) of a channel perturbed by additive white Gaussian noise (AWGN) is a function of the received signal power \(S\), the noise power \(N\), and the receiver bandwidth \(W\). It is possible to send information at the rate \(R\), where \(R \leq C\), through the channel with an arbitrarily small error probability by using a sufficiently complicated coding scheme. In 1962, Gallager proposed Low Density Check Codes (LDPC) [3], which is a linear block codes [4], and proved that the data transmission rate of LDPC code can approach the Shannon capacity \(C\).

II. APPROXIMATE LOWER TRIANGULAR LDPC ENCODING ARCHITECTURE

A. Characteristics of low density parity check matrix \(H\)

LDPC code is a linear block code based on sparse check matrix. General block codes can be encoded and decoded by generating matrix and parity check matrix. The approximate lower triangular encoding method used in this study directly uses parity check matrix for encoding.

B. Encoding steps of LDPC check matrix based on approximate lower triangular structure

Compared with general linear block code encoding, LDPC encoding with lower triangular check matrix [5] and approximate lower triangular check matrix [6] carry out encoding directly by parity check matrix \(H\). The encoding method by lower triangular check matrix structure is to use the Gaussian elimination to convert the check matrix \(H\) into lower triangular matrix structure (as shown in Fig. 1) before encoding. The encoding complexity is \(O(n^3)\), where \(n\) is the column of check matrix. However, lower triangular check matrix produced in this method is not consistent with the sparse characteristics. The encoding method by approximate lower triangular check matrix structure is to directly use a given lower triangular sparse check matrix, which may result in loss of encoding performance [7]. This study used the approximate lower triangular regular parity check matrix to implement the LDPC encoder on the 5641R FPGA of the Software Define Radio system.

Figure 1. Lower Triangular Parity Check Matrix Structural Diagram

Approximate lower triangular LDPC encoding was proposed by Richardson and Urbanke in 2001. The encoding is to disintegrate the check matrix \(H\) of (1) into six (A, B, C, D, T, E) sparse sub-matrix before working out the redundant bit \(p_1\), \(p_2\) according to the characteristics of the six sparse sub-matrix to complete the encoding. The encoding complexity is \(O(n^2)\), \(g\) is the row of matrix \(E\). Compared with the lower triangular encoding matrix, the complexity is lower and the encoding is consistent with the sparse characteristics; hence, the encoding performance is relatively higher. The approximate lower triangular LDPC encoding is illustrated as below:

First, a standard original matrix will be given as:
The matrix finally can be used should be obtained by processing of four steps as follows:

**Step 1:** Conduct column switching of the Matrix A and Matrix B of the original matrix with Matrix T. The switching results are: the diagonal lines of the Matrix T are all 1; the right upper corners are all 0 (as shown in Fig. 2).

**Step 2:** For the convenience of solving the redundant vector $p_1$, Gaussian elimination computation of check matrix $H$. Matrix $I$ in (2) represent the identity matrix.

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

**Step 3:** After the Gaussian elimination computation, we define $\Psi = -ET^{-1}B+D$, and check whether $\Psi$ is reversible. If it is irreversible, column switching with Matrix A should be redone.

**Step 4:** After confirming $\Psi = -ET^{-1}B+D$ as reversible, the weights of parity check matrix can be converted into the weights of the original parity check matrix (3, 6) to complete the parity check matrix that can be encoded.

Codeword vector $X$ is composed of two parts including the information vector $m$ and redundant vectors $(p_1, p_2)$. $HX^T = 0$ can lead to (3) (4) by inference:

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

(3)

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

(4)

After obtaining by (3) (4) the redundant vectors $(p_1, p_2)$, codeword $X = [m \ p_1 \ p_2]$ can be obtained by adding the information vector $m$ to complete the entire approximate lower triangular LDPC encoding process.

$$p_i^T = -\Psi^{-1}(-ET^{-1}A + C)m^T$$

(5)

### III. REALIZATION OF APPROXIMATE LOWER TRIANGULAR LDPC ENCODING IN FPGA

#### A. Transforming processes of the approximate lower triangular parity check matrix

On the SDR platform of NI LabVIEW 5641R FPGA, we transform the original given weight (3, 6) parity check matrix into a weight (3, 6) approximate lower triangular regular parity check matrix to implement the LDPC encoding circuit. The number of row and column number $g$ of the approximate lower triangular matrix are $N = 12$, $M = 6$ and $g = 2$. The number of “1” in each column of the weight (3, 6) approximate lower triangular regular parity check matrix is fixed to 3, and 6 in each row. The weight (3, 6) approximate lower triangular regular matrix is divided into six sub-matrix of $A$, $B$, $C$, $D$, $T$, $E$ (as shown in Fig. 3).

**Step 1:** Rearranging the sequence of column into 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 8, 9, so as to transform the diagonal terms of the matrix $T$ into “1”.

**Step 2:** Using the Gaussian elimination to convert the upper check matrix $H$ into the lower one with some modifications of the matrix C, D, E.

**Step 3:** Define $\Psi = -ET^{-1}B+D$. Interchange column 5 with column 8 of matrix $D$. So that matrix $\Psi$ become reversible.

**Step 4:** After confirming $\Psi = -ET^{-1}B+D$ as reversible, the weights of parity check matrix can be converted into the weights of the original parity check matrix (3, 6) to complete the entire approximate lower triangular LDPC encoding process.
Step 4: Converting the upper check matrix $H$ into a weight $(3, 6)$ approximate lower triangular regular parity check matrix by rearranging the column into $1, 2, 3, 4, 10, 6, 7, 5, 11, 12, 8, 9$.

<p>| | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 7. Weight $(3, 6)$ approximate lower triangular parity check matrix

In the 5641R FPGA card, several Sub-VI programs are written to solve the redundant vectors $p_1, p_2$. First, the check matrix $H$ is cut into 6 block matrices. Then, according to (5), $p_1^T$ is disintegrated into three Sub-VI programs of $Am^T$, $ET^TAm^T$, $Cm^T$. Multiply the Sub-VI programs of $Cm^T$ and $ET^TAm^T$ with $-\psi^{-1}$ to complete the $p_1^T$ circuit (as shown in Fig. 8). Input $p_1^T$ into (6), by multiplying the two Sub-VI programs of $Am^T$, $Bp_1^T$ with $-T^T$, the $p_2^T$ circuit can be obtained (as shown in Fig. 9).

Since the $p_1, p_2$ calculation process is considerably complex; circuits of $p_1^T, p_2^T$ are set as Sub-VI programs for simplification. Finally, by adding the transparent information vector $m$, the encoding systematic codeword $X$ can be obtained (shown in Fig. 10).

B. Hardware verification of the LDPC encoding circuit

The hardware verification of the LDPC encoding circuit as show in Fig. 11 can be realized by the operation of the equation $HX^T = 0$. The random generated codeword $X$ is transposed and multiplied with the weight $(3, 6)$ approximate lower triangular regular parity check matrix. It indicates that there are some errors exist, if $HX^T$ is nonzero.
C. Circuit elements of LDPC encoder

After finishing the LabVIEW FPGA program of the approximate lower triangular LDPC encoder, we download it into the 5641R FPGA SDR modules for compilation as shown in Fig. 12.

![Compilation status](image)

Figure 12. Approximate lower triangular LDPC encoder compilation status

The LDPC encoder compilation status is shown in Fig. 12. We could summarize the hardware circuit elements in Fig. 13 from the encoder compilation status as shown in Fig. 12. There are eight circuit elements used in the weight (3, 6) approximate lower triangular regular parity check matrix, (as shown in Fig. 13): Add/Subtractors for adding and subtracting processes; Counters for counting process; Registers for saving bit information of Flip-Flop logical gates in processing; Latches for saving hard bit information of the digital circuit; Comparators including 2-bit equal, 4-bit greatequal, 4-bit less, 4-bit lessequal; 16-bit 6 to 1 multiplexers; Tristates for saving buffer information; and one-bit XOR.

![Circuit elements](image)

Figure 13. Circuit elements of LDPC encoder

IV. Conclusions

LabVIEW FPGA is a programming language that directly builds programming codes in hardware. In addition to saving complex detail to speed up design process, it can design all pulse periods by programming according to functional needs. On this SDR platform, we completed the approximate lower triangular LDPC encoding circuit and used the orthogonal characteristic of $H^T X = 0$ to verify the accuracy of encoding programs. LDPC is based on Message Passing Algorithm (MPA) [8] to decode. In the near future, we will use the programming decoding circuits of three commonly used LDPC decoding algorithms including Sum-of-Product Algorithm (SPA), Minimum-Sum Algorithm (MSA), Normalized Min-Sum Algorithm (NMSA) [9][10][11], with additional white Gaussian noise channel (AWGN Channel), BPSK Modulation and decoding hardware verification as well as LDPC code error performance on 5641R FPGA module card of NI SDR system will be conducted.

REFERENCES