

Efficient LDPC coding using a hybrid H-matrix

Tae-Jin Kim*, Chanho Lee*, Soon-Il Yeo**, and Tae Moon Roh**

* Department of Electronic Engineering, Soongsil University

1 Sangdo-5-dong, Dongjak-ku, Seoul 156-743, Korea

Phone : +82-2-820-0710, E-mail : chlee@ssu.ac.kr

** Basic Research Laboratory, Electronics and Telecommunications Research Institute,

Yuseong P.O. Box 106, Daejeon, 305-600, Korea

Phone: +82-42-860-6272, E-mail: tmroh@etri.re.kr

Abstract: Low-Density Parity-Check (LDPC) codes are recently emerged due to its excellent performance to use. However, the parity check matrices (H) of the previous works are not adequate for hardware implementation of encoders or decoders. This paper proposes a hybrid parity check matrix for partially parallel decoder structures, which is efficient in hardware implementation of both decoders and encoders. Using proposed methods, the encoding design can become practical while keeping the hardware complexity of partially parallel decoder structures.

Keywords : LDPC, Hybrid, H-matrix, Partially parallel, Implementation

1. INTRODUCTION

LDPC (Low Density Parity Check) code [1] proposed by R. G. Gallager in 1962 was too complex to implement and had been almost forgotten in spite of its powerful error-correcting capability. However, it was rediscovered by MacKay and Neal in 1990's and they made significant improvements on BER performance [2]. Chung, et al[3], showed that the threshold for a LDPC code of code rate 1/2 on the additive white Gaussian noise(AWGN) channel was within 0.0045dB of the Shannon limit and simulation results were within 0.04dB of the Shannon limit at a bit error rate of 10^{-6} using a block length of 10^7 .

Compared with turbo codes, LDPC codes exhibit better performance due to good distance properties and less complex and highly parallelizable decoding approaches [4]. Therefore, LDPC codes have been widely considered as a next-generation error-correcting code for telecommunication. However, encoding complexity of LDPC codes is still too high, and it is the major problem that needs to be solved for the implementation of LDPC codes. There have been some studies to reduce the encoding complexity by the methods of using a specially formed matrix such as a lower triangular matrix [5] and semi-random matrix [6].

The encoding process of standard LDPC codes requires to transfer a parity check matrix (H) into an equivalent systematic form, which can be accomplished by the Gaussian elimination [6]. The Gaussian elimination requires large memory and heavy calculation [6]. The encoding process with semi-random technique is much simpler than that using other matrices because it doesn't require Gaussian elimination [6]. Consequently linear time encoding is possible with very little memory.

The hardware implementation of LDPC decoders is another problem to be considered when we use the fully parallel decoding algorithm of LDPC codes [4]. Although the fully parallel decoders can achieve very high decoding speed, it is too complex to implement practically [7]. One of the best solution for the decoder architecture design is to directly instantiate the belief propagation (BP) algorithm [8] using hardware [7]. In fully parallel

decoding structures, all check nodes and variable nodes have their own processor and exchange messages between each check node and variable node in fully parallel. In order to lower the hardware complexity, the number of check node and variable node processors need to be reduced. In partially parallel decoding structures, part of variable nodes and check nodes perform the message passing process in time-division multiplexing mode [9]. Therefore, there is trade-off between decoding throughput and hardware complexity in partially parallel structures.

Although the hardware complexity of LDPC decoders is reduced using partially parallel structures, the partially parallel structures have a potential problem of encoding complexity because their parity check matrices may not be suitable for efficient encoding process.

In this paper, we propose a hybrid model that combines partially parallel decoder structures and semi-random technique to have efficient encoding process and decoding process. We design an LDPC encoder using the proposed model and implement it using an FPGA.

2. SEMI-RANDOM TECHNIQUE

An H-matrix generated with semi-random technique, consists of two parts: H^d and H^p . H^d is a randomly generated form and H^p is a deterministic form [6]. The matrix structure makes encoding processes simple because the deterministic form of H^p shown in Fig. 1 is a square matrix.

$$H^p = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdot & \cdot & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & \cdot & \cdot & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & \cdot & \cdot & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & \cdot & \cdot & 0 & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \cdot & \cdot & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdot & \cdot & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdot & \cdot & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdot & \cdot & 0 & 0 & 1 & 1 \end{pmatrix}$$

Fig.1. Deterministic matrix in semi-random technique

Although an H^d - matrix is randomly generated, some efficient forms of matrices are preferred for better coding performance if possible. Once an H^d - matrix is generated, an H matrix is constructed as $H = [H^d, H^p]$. When codeword $C = [d, p]$, where p and d are parity bits and information bits, respectively, parity bits can be easily calculated according to the following equation [6].

$$p_i = \sum_j h_{1j}^d d_j, p_i = p_{i-1} + \sum_j h_{ij}^d d_j, \text{ when } P = \{p_i\}, d = \{d_i\} \quad (1)$$

Equation (1) can be easily implemented as shown in Figure 2 [10].

The encoder structure can be constructed using an input buffer, an interleaver, and a parity bit generator as shown in Fig 3. The H^d - matrix determines the interleaving operations. It is the only part of encoder blocks to be modified when a different H^d matrix is applied. Therefore, we have to find a good H^d - matrix for efficient encoding. In a semi-random matrix structure, the choice of a good H^d -matrix is the major point of good performance and efficient hardware implementation.

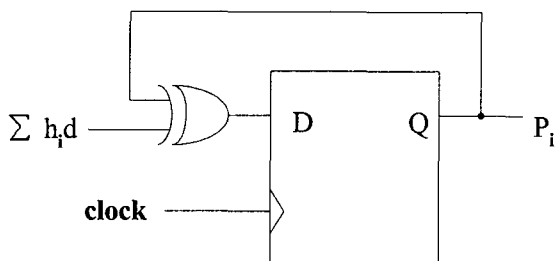


Fig.2. Parity bit generation circuit

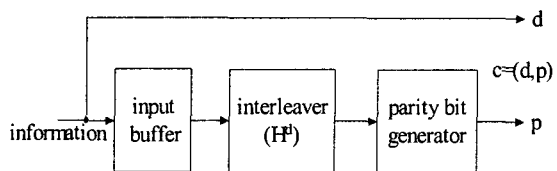


Fig.3. structure of an encoder using semi-random technique

3. PARTIALLY PARALLEL DECODER STRUCTURE

Partially parallel decoder structures are originally proposed by Zhang et al [11]. First, they designed a good partially parallel decoder structure, and then a new parity check matrix form, which is constructed with shifted identity matrices, is generated based on the structure. However, their model does not support a flexible code rate and a degree distribution, which is required to achieve very good error-correcting performance [9]. To solve the problem of the original partly parallel decoder structure, a modified model, a matrix expansion method, is proposed [9]. An expanded matrix is constructed using an $(M_s \times N_s)$ base matrix and $(p \times p)$ shifted identity matrices.

To construct a base matrix, we need to apply bit-filling algorithm [12]. Since the matrix by bit-filling algorithm has a large girth, which can be a standard of good error correcting ability, we can avoid small cycles that make

performance worse and can expect to have a good error correcting matrix. In Fig 4, we randomly expand it by a factor p to obtain a $(pM_s \times pN_s)$ matrix after a base matrix with bit-filling algorithm is generated [9]. Each 0 in the base matrix is expanded to a $(p \times p)$ zero sub-matrix O , and 1's are expanded to a $(p \times p)$ sub-matrix $T_{u,v}$. $T_{u,v}$ is a $(p \times p)$ identity matrix with cycle shift right by the number of a randomly generated integer. The method of replacing 1's of a base matrix and expanding with an identity matrix make the design flexible.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} T_{1,1}O & O & O & O & T_{1,5}O \\ O & T_{2,2}O & T_{2,4}O & O & O \\ T_{3,1}O & O & O & O & O & T_{3,6} \end{bmatrix}$$

Base Matrix($M_s \times N_s$) Expanded matrix($pM_s \times pN_s$)

Fig.4. Matrix expansion

When we implement decoder structures with the expanded matrices, all check nodes and variable nodes don't have to have their own computational units, and time-division multiplexing mode can be applied as shown in [11].

As shown in Fig 5[9], the numbers of check node processor unit (CNU) and variable node processor unit (VNU) are M_s and N_s , respectively, which are the numbers of rows and columns of the base matrix. In fully parallel structures, $p \cdot M_s$ CNU's and $p \cdot N_s$ VNU's are required and it is p times the processor units in partially parallel structures. However, the partially parallel decoder completes each decoding iteration in $2p$ cycles, while the fully parallel decoder needs only 2 cycles for decoding iteration.

The partially parallel decoding scheme has still encoding complexity problems although implementation of a decoder becomes efficient in its hardware size and flexible with expanded matrices, which are constructed for partly parallel decoder structures.

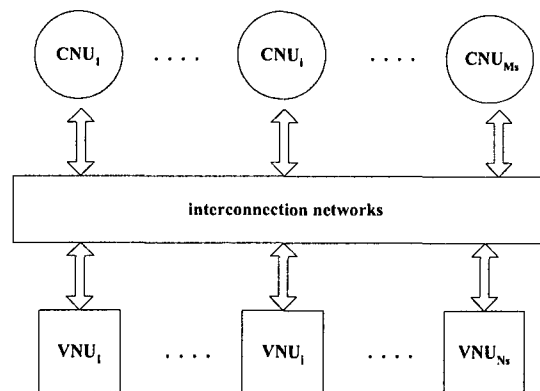


Fig.5. Partially parallel decoder structure

4. HYBRID PARTIALLY PARALLEL STRUCTURE

We need to modify the H-matrix in order to reduce the encoding complexity of the partially parallel structure as mentioned above. As we have already shown in section 2, H-matrix generated using semi-random technique consists

of two parts, H^d and H^p . H^d -matrix with a random form can be replaced by another one. The decoder structure according to the semi-random technique has a large hardware complexity that the implementation is not easy. On the other hand, the encoder for the partially parallel decoding scheme is not adequate for implementation. The structure of a decoder in semi-random technique is not adequate for the partially parallel structure because of the random form of the H^d matrix. We replace the H^d -matrix in semi-random technique with an expanded matrix which is used in the partially parallel decoding scheme. By applying semi-random technique to partially parallel structures, it is possible to practically implement encoders and decoders of LDPC codes.

We made a simple H-matrix to construct the proposed hybrid H-matrix. We first made a (8 x 8) base matrix using bit-filling algorithm and expanded it to (512 x 512) matrix with (64 x 64) identity matrices. Each identity matrix that replaces the position of 1's in the base matrix is shifted right by randomly generated integers. In Fig 6, the numbers in the matrix represent the number of right-shift in each identity matrix.

The encoder structure is almost same as the one in Fig 3, except for the interleaver block. The structure of the interleaver is simpler than that of a randomly generated H^d -matrix. The H-matrix will be the combination of the H^d -matrix in Fig 6 and the (512 x 512) H^p -matrix in Fig 1. The size of the H-matrix is (1024 x 512) and its code rate is 1/2. When we implemented an LDPC encoder using this matrix for an FPGA, Xilinx XCV600E-hq240-6, the result of implementation was 151 slices out of 6912 (utilization : 2%) and 2 block rams out of 28. In fact, real utilization of block rams is 0.2%, because only 512 bits out of 294912 bits. This result shows that to use the hybrid H-matrix has a low hardware complexity.

$$H^d = \begin{pmatrix} 0 & 41 & 35 & 62 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 33 & 0 & 0 & 44 & 0 \\ 22 & 0 & 46 & 0 & 0 & 18 & 0 & 0 \\ 16 & 0 & 0 & 0 & 9 & 0 & 0 & 49 \\ 0 & 49 & 0 & 0 & 0 & 59 & 0 & 41 \\ 0 & 0 & 0 & 43 & 51 & 38 & 0 & 0 \\ 0 & 0 & 27 & 0 & 0 & 0 & 60 & 7 \\ 0 & 12 & 0 & 0 & 62 & 0 & 25 & 0 \end{pmatrix}$$

Fig.6. Generated Base Matrix for implementation

The decoder structure with the hybrid H-matrix is similar to that of partially parallel decoder since the H^d -matrix is the same as left half of the H-matrix of the partially parallel coding scheme. The structure is slightly changed due to the H^p -matrix, which corresponds the right half of the H-matrix of the partially parallel decoding scheme. The VNU block and the interleaver in Fig.7 are mainly different from those of the partially parallel decoder to process the H^p -matrix part. The CNU block and the deinterleaver are also changed slightly. When we implemented an LDPC decoder using Hybrid partially parallel structure for an FPGA, Xilinx XCV600E-hp240-6, the result of implementation was 2434 slices out of 6912 (utilization : 35%) and 72 block rams out of 72. However, real utilization of block rams is

6%, because only 17920 bits out of 294912 bits.

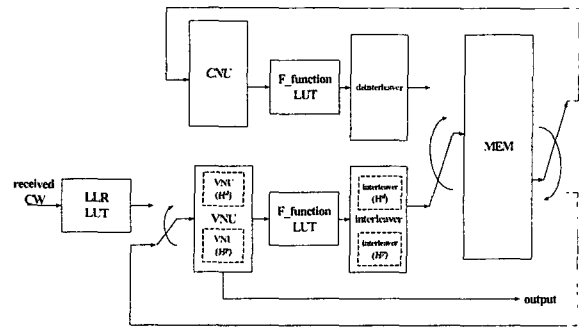


Fig.7. The decoder structure of the proposed decoding scheme

Table1. shows results of implementation of encoder and decoder.

Table1. Results of Implementation

<Encoder>		
Resource	Number	Utilization Rate
Slices	151	2%
Block RAMs	512bits /294,912bits	0.2%

Operating Frequency: 90MHz

<Decoder>		
Resource	Number	Utilization Rate
Slices	2434	35%
Block RAMs	17,920bits /294,912bits	6%

Operating Frequency: 30MHz

The hardware complexity, throughput and memory size of three methods of LDPC codes are compared in table2. The results of the decoder with the semi-random technique are from the fully parallel decoding. As we mentioned, there is a trade-off between throughput and hardware complexity. The proposed decoding scheme with hybrid H-matrix has mostly the same hardware complexity as the partially parallel methods. On the other hands, hardware complexity of the proposed encoder is simpler than the encoder with the partially parallel coding scheme, and the linear time encoding is possible.

Table2. Comparison of three methods
(Block length (N) = 1024, code rate = 1/2)

	Semi-random[6]	Partially parallel [9]	Hybrid Partially parallel
HW complexity in encoder	N	>>N	N
CNU	512	8 (p=64)	8 (p=64)
VNU	1024	16	16
Throughput in decoder (iteration 1 time)	2	2p (p=64)	2p (p=64)
Required memory in decoder	L + P x Ns	L + Ns	L + Ns

5. CONCLUSIONS

The encoding scheme using a semi-random H-matrix is efficient in hardware implementation while the decoder is not adequate for hardware implementation. The partially parallel decoding scheme is efficient in hardware implementation while the corresponding decoder is not practical for hardware implementation

We propose a hybrid H-matrix to combine the advantage of both coding scheme. The encoder using the hybrid H-matrix has the hardware complexity of the encoder with the semi-random technique. The decoder using the hybrid H-matrix has the similar operation characteristics and hardware complexity to the partially parallel decoder.

References

- [1] R. G. Gallager, "Low density parity check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21-28, Jan. 1962.
- [2] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 32, pp. 1645-1646, Aug. 1996.
- [3] S.-Y. Chung, G. D. Forney Jr., T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045dB of the Shannon limit," *IEEE Commun. Lett.*, 1999.
- [4] Tong Zhang, Z. Wang, and K. K. Parhi, "ON FINITE PRECISION IMPLEMENTATION OF LDPC DECODER" in *Proc. of 2001 IEEE Int. Symp. on Circuits and Systems, Sydney*, May 2001.
- [5] T. J. Richardson, and R. Urbanke, "Efficient Encoding of Low-Density Parity-Check Codes," *IEEE Trans. Inform. Theory*, vol. 47, No. 2, Feb. 2001.
- [6] Li Ping, W. K. Leung, and Nam Phamdo, "Low density parity check codes with semi-random parity check matrix," *IEE Electronics Lett.*, Nov. 1999.
- [7] Tong Zhang, and Keshab K. Parhi, "A 54 Mbps (3,6)-regular FPGA LDPC decoder," *Signal Processing Systems, 2002. IEEE Workshop*, pp. 16-18, Oct. 2002.
- [8] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399-431, Mar. 1999.
- [9] Hao Zhong and Tong Zhang, "Design of VLSI Implementation-Oriented LDPC codes," *IEEE Vehicular Technology Conference*, Oct. 2003.
- [10] R. Echard, and Shih-Chun Chang. "The π -rotation low-density parity check codes," *IEEE Global Telecom. Conf.*, vol. 2, pp. 25-29, Nov. 2001.
- [11] T. Zhang, and K. K. Parhi, "VLSI implementation-oriented (3,k)-regular low-density parity check codes," *IEEE Workshop, signal processing systems*, Sept. 2001.
- [12] J. Campello, and D. S. Modha, "Extended bit-filling and ldpc code design," *IEEE Global Telecom. Conf.*, 2001, pp. 985-989