

# A Fast-Convergence Decoding Method and Memory-Efficient VLSI Decoder Architecture for Irregular LDPC Codes in the IEEE 802.16e Standards

Yeong-Luh Ueng and Chung-Chao Cheng

Dept. of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, R.O.C.

**Abstract**—In this paper, we propose a modified iterative decoding algorithm to decode a special class of quasi-cyclic low-density parity-check (QC-LDPC) codes such as QC-LDPC codes used in the IEEE 802.16e standards. The proposed decoding is implemented by serially decoding block codes with identical parity-check matrix  $\mathbf{H}_1$  derived from the parity-check matrix  $\mathbf{H}$  of the QC-LDPC codes. The dimensions of  $\mathbf{H}_1$  are much smaller than those of  $\mathbf{H}$ . Extrinsic values can be passed among these block codes since the code bits of these block codes are overlapped. Hence, the proposed decoding can reduce the number of iterations required by up to forty percent without error performance loss as compared to the conventional message-passing decoding algorithm. A partially-parallel very large-scale integration (VLSI) architecture is proposed to implement such a decoding algorithm. The proposed VLSI decoder can fully take advantage of the proposed decoding to increase its throughput. In addition, the proposed decoder only needs to store check-to-variable messages and hence is memory efficient.

**Keywords**— Iterative decoding, low-density parity-check (LDPC) codes, decoder, very large-scale integration (VLSI) architecture, modulation and coding.

## I. INTRODUCTION

Low-density parity-check (LDPC) codes [1] have attracted tremendous research interest recently because of their excellent error-correcting performance and their potential of highly parallel implementation of decoder. Although the Shannon limit can be achieved by irregular LDPC codes [2], the very large-scale integration (VLSI) implementation of an irregular LDPC decoder remains a big challenge. A practical design approach of LDPC coding system called Block-LDPC [3][4] has been used to construct LDPC codes with effective VLSI implementation of decoder and good error-correcting performance. An LDPC code constructed by using Block-LDPC is indeed a quasi-cyclic (QC) LDPC code [5]. The irregular LDPC codes selected by the standard of IEEE 802.16e (WiMax) [9] are Block-LDPC codes.

Iterative message-passing decoding (MPD) based on sum-product algorithm (SPA) [6] is a well-known decoding method for LDPC codes. However, for such a decoding method, a large number of iterations which cause low throughput are demanded to recover the reliable information. In this paper, we propose a modified iterative decoding method to improve the convergence speed for Block-LDPC codes  $C$ . Let the  $M \times N$  matrix  $\mathbf{H}$  be the parity-check matrix of  $C$ , where  $M = zM_b$ ,  $N = zN_b$ . Each  $z \times z$  sub-matrix in  $\mathbf{H}$  is either a zero matrix or a circulant obtained by right cyclic shifting a  $z \times z$  identity matrix by certain columns. We partition the indexes of code bits of  $C$  into  $z$  sets  $S_i(i)$  with  $|S_i(i)| = N_i$ ,  $i = 0, 1, \dots, z-1$ . Since this partition is not disjoint, we have  $N_b < N_i < N$ . The

code bits of  $C$  indexed by  $S_i(i)$  form a linear block code  $C_i(i)$ ,  $i = 0, 1, \dots, z-1$ . From the quasi-cyclic structure of  $\mathbf{H}$ , we can construct  $C_i(i)$ ,  $i = 0, 1, \dots, z-1$ , which have the same  $M_b \times N_i$  parity-check matrix  $\mathbf{H}_1$ . The proposed decoding of  $C$  is implemented by serially decoding  $C_i(0)$ ,  $C_i(1)$ ,  $\dots$ , and  $C_i(z-1)$ . Extrinsic values are passed among codes  $C_i(i)$ ,  $i = 0, 1, \dots, z-1$ , since the code bits of  $C_i(i)$ ,  $i = 0, 1, \dots, z-1$ , are overlapped. As compared to the conventional MPD, the proposed decoding can reduce the number of iterations required by up to forty percent and hence significantly improve the convergence speed without error performance loss.

The MPD for LDPC codes can be implemented by fully-parallel architecture which results in a high throughput decoder but with complex interconnections caused by a quite large number of irregular edges [10]. On the other hand, the serial architecture reduces the interconnection complexity by using a memory-based approach [11]. However, the throughput of decoder based on serial architecture is low [11]. In addition, two memory units are needed to store check-to-variable and variable-to-check messages. To balance the complexity of interconnections and throughput, a partially-parallel architecture, where certain logic devices have to be utilized in a time-multiplexed manner, was used in [12][13][14][15]. The LDPC decoders reported in [14] and [15] are based on the partially-parallel architectures proposed in [12] and [13], respectively.

In this paper, we also propose a partially-parallel VLSI architecture which is totally different from those given in [12][13][14][15] to implement the proposed decoding algorithm. We use a fully-parallel architecture to implement a high-speed decoder of  $\mathbf{H}_1$ . We then use the high-speed decoder of  $\mathbf{H}_1$  to serially decode  $C_i(0)$ ,  $C_i(1)$ ,  $\dots$ , and  $C_i(z-1)$ , since the parity-check matrices of  $C_i(i)$ ,  $i = 0, 1, \dots, z-1$ , are identical to  $\mathbf{H}_1$ . The fully-parallel implementation of the decoder for  $C_i(i)$  ( $\mathbf{H}_1$ ) does not result in complex interconnections since the code length,  $N_i$ , of  $C_i(i)$  is much shorter than the code length,  $N$ , of  $C$ . For the (2304,1152) LDPC code used in the IEEE 802.16e standards, we can have  $N = 2304$  and  $N_i = 63$ . The proposed VLSI decoder can fully take advantage of the proposed decoding method to increase the decoding throughput. In addition, the VLSI decoder only needs to store the check-to-variable messages and hence is memory efficient.

The remainder of this paper is organized as follows. Section II briefly reviews the LDPC codes used in the IEEE 802.16e standards, MPD based on SPA, and the conventional VLSI architectures for LDPC decoders. Section III proposes a decoding method for improved convergence speed. We also compare the proposed decoding method with the conventional

MPD in Section III. In Section IV, we propose a VLSI architecture for the proposed decoding method. We also show the implementation results and compare these results with those given in [14][15]. Finally, Section V addresses some concluding remarks.

## II. PRELIMINARIES

### A. Irregular LDPC codes in the IEEE 802.16e standards

In this section, we briefly review the parity-check matrix  $\mathbf{H}$  of Block-LDPC (QC-LDPC) code  $C$  used in the IEEE 802.16e standards. We construct the  $M \times N$  matrix  $\mathbf{H}$  based on an  $M_b \times N_b$  base parity check matrix  $\mathbf{H}_b$ , where  $M = zM_b$ ,  $N = zN_b$ , and  $z$  is a positive integer. In  $\mathbf{H}_b$ , each 0 is replaced by a  $z \times z$  zero sub-matrix and each 1 at the position  $(i, j)$  is replaced by a  $z \times z$  sub-matrix that is obtained by right cyclic shifting a  $z \times z$  identity matrix by  $p(i, j) \geq 0$  columns,  $0 \leq i \leq (M_b - 1), 0 \leq j \leq (N_b - 1)$ .  $\mathbf{H}_b$  and  $p(i, j)$  can be found in the IEEE 802.16e standards [9]. For the (2304,1152) LDPC code,  $M = 1152, N = 2304, z = 96, M_b = 12, N_b = 24$ . Fig. 1 shows the block-type parity-check matrix  $\mathbf{H}$  of this LDPC code, where the  $(i, j)$  element is  $p(i, j)$ .

### B. Message-passing decoding based on sum-product algorithm (SPA)

Let  $\lambda_j = \ln(Pr(v_j = 0|y_i)/Pr(v_j = 1|y_i))$  be the channel reliability value of bit (variable node)  $v_j$ , where  $y_i$  is the noise-corrupted form of  $v_j$ . Let  $R_{ij}[k]$  ( $Q_{ij}[k]$ ) be the check-to-variable (variable-to-check) message from check (variable) node  $i$  to variable (check) node  $j$  at the  $k$ th iteration, and  $R[i]$  ( $C[j]$ ) be the index set of variable (check) nodes involving check (variable) node  $i$  ( $j$ ).

Initialization: For  $k = 0$ , the check-to-variable messages  $R_{ij}[0]$  from the  $i$ th check node to the  $j$ th variable node are initialized to zero for all  $i$ , with  $j \in R[i]$ .

At iteration  $k$ : Operations at variable nodes: For each variable node  $j$ , compute  $Q_{ji}[k]$  corresponding to each of its check node neighbors  $i$  according to

$$Q_{ji}[k] = \lambda_j + \sum_{i' \in C[j] \setminus \{i\}} R_{i'j}[k-1]. \quad (1)$$

Operations at check nodes: For each check node  $i$ , compute  $R_{ij}[k]$  corresponding to each of its variable node neighbors  $j$  according to

$$R_{ij}[k] = -S_{ij}[k](\Psi(|\sum_{j' \in R[i] \setminus \{j\}} \Psi(|Q_{j'i}[k]|)|)) \quad (2)$$

where  $\Psi(|x|) = \ln(|\frac{\exp(x)-1}{\exp(x)+1}|)$ , and  $S_{ij}[k] = \prod_{j' \in R[i] \setminus \{j\}} \text{Sign}(Q_{j'i}[k])$ .

Hard decision: At iteration  $N_i$ , for each variable node  $j$ , compute the *a posteriori* reliability value  $\Lambda_j$  according to

$$\Lambda_j = \lambda_j + \sum_{i \in C[j]} R_{ij}[N_i]. \quad (3)$$

Hard decisions are then made based on the sign of  $\Lambda_j$ ,  $j = 0, 1, \dots, N-1$ .

### C. Conventional VLSI architectures for LDPC decoders

The message passing decoding for LDPC codes can be implemented by fully-parallel architecture [10] which is shown in Fig. 2. The fully-parallel implementation of the message-passing algorithm results in a high-throughput decoder but with complex interconnections caused by a quite large number of irregular edges [10]. The interconnection complexity can be reduced by employing the serial architecture [11] in which a shared processing unit (PU) computes all the rows or columns one after another. The interconnection complexity can also be reduced by employing the partially-parallel architecture [12][13][14][15] in which each PU takes in charge of several number of rows or columns. As a PU is shared for a number of rows or columns, the number of PUs becomes much smaller than that of the fully-parallel architecture. Fig. 3 and Fig. 4 shows the conventional LDPC decoder based on the serial architecture and partially-parallel architecture, respectively. The quantized log-likelihood ratios (channel values) of the received code bits are fed into the decoder. The processing units  $PU_{cn}$  and  $PU_{vn}$  perform the operations at check nodes and variable nodes, respectively. The memory unit  $MU_{cn}$  is used to store the extrinsic value (check-to-variable message)  $R_{ij}[k]$ . The memory unit  $MU_{vn}$  is used to store variable-to-check messages  $Q_{ji}[k]$ . At the final  $N_i$  iteration, hard decisions of the code bits are produced by the processing unit  $PU_{hd}$ . Whereas the fully-parallel architecture computes all the messages simultaneously, the serial (partially-parallel) architecture computes messages row-by-row or column-by-column because there is only one PU (a small number of PUs) for each step. Therefore, variable messages calculated by a variable PU are stored into a memory and accessed later by a check PU, and vice versa. Notably, two memory units ( $MU_{vn}$  and  $MU_{cn}$ ) are required for either the serial or the partially-parallel architecture.

## III. PROPOSED DECODING METHOD FOR IMPROVED CONVERGENCE SPEED

### A. Proposed decoding method

For  $i = 0, 1, \dots, z-1$ , let  $\mathbf{H}'_1(i)$  be an  $M_b \times N$  matrix which contains the  $i$ -th,  $(i+z)$ -th,  $(i+2z)$ -th,  $\dots$ , and  $(i+(M_b-1)z)$ -th rows of  $\mathbf{H}$ . For  $i = 0, 1, \dots, z-1$ , let  $S_l(i)$  be an index set which indicates the non-zero columns of  $\mathbf{H}'_1(i)$ . For example, if only the first, the second, and the third columns of  $\mathbf{H}'_1(0)$  are non-zero, then  $S_l(0) = \{1, 2, 3\}$ . For  $i = 0, 1, \dots, z-1$ , let  $\mathbf{H}_1(i)$  be an  $M_b \times N_l$  matrix which is obtained by deleting the all-zero columns of  $\mathbf{H}'_1(i)$ . Hence,  $\mathbf{H}_1(i)$  contains the columns of  $\mathbf{H}'_1(i)$  indexed by  $S_l(i)$  and  $|S_l(i)| = N_l$ . From the quasi-cyclic structure of  $\mathbf{H}$ , we can find that matrices  $\mathbf{H}_1(i), i = 0, 1, \dots, z-1$ , are identical to  $\mathbf{H}_1 = \mathbf{H}_1(0)$  and  $S_l(i+1) = \bigcup_{j=0}^{N_b-1} \{q|q-jz = (k+1-jz) \bmod z; jz \leq k < (j+1)z, k \in S_l(i)\}, i = 0, 1, \dots, z-2$ . In addition, sets  $S_l(i), i = 0, 1, \dots, z-1$ , are not the same and  $S_l(i) \cap (\bigcup_{j=0, j \neq i}^{z-1} S_l(j)) \neq \phi$  for  $i = 0, 1, \dots, z-1$ , where  $\phi$  is the null set. Notably,  $N_l$  is not equal to  $N_b$  and  $N_l$  is much smaller than  $N$ .

The code bits of  $C$  indexed by  $S_l(i)$  form a linear block code  $C_l(i)$ ,  $i = 0, 1, \dots, z-1$ . We can find that the  $M_b \times N_l$  matrix  $\mathbf{H}_1(i) = \mathbf{H}_1$  is the parity-check matrix of  $C_l(i)$ ,  $i = 0, 1, \dots, z-1$ . The block diagram of the proposed decoding is given in Fig. 5. The proposed decoding of code  $C$  is implemented by serially decoding  $C_l(0)$ ,  $C_l(1)$ ,  $\dots$ , and  $C_l(z-1)$ . We first decode  $C_l(0)$  using the channel values of code bits of  $C$  indexed by  $S_l(0)$  and then decode  $C_l(1)$  using the channel values of code bits of  $C$  indexed by  $S_l(1)$  and the extrinsic information provided by the decoding of  $C_l(0)$  and so on. After decoding  $C_l(z-2)$ , we then decode  $C_l(z-1)$ . Such one-round decoding of  $C_l(i)$ ,  $i = 0, 1, \dots, z-1$ , is called a global iteration for the decoding of  $C$ . After decoding  $C_l(z-1)$ , we then re-decode  $C_l(0)$  and so on. The number of global iterations is denoted by  $N_g$ . We can use the MPD based on SPA with  $N_l$  iterations to decode  $C_l(i)$ ,  $i = 0, 1, \dots, z-1$ . Since  $S_l(i) \cap (\bigcup_{j=0, j \neq i}^{z-1} S_l(j)) \neq \emptyset$  for  $i = 0, 1, \dots, z-1$ , the decoding of  $C_l(i)$  can use the extrinsic values provided by the decoding of  $C_l(j)$ ,  $j \neq i$ . Since in the decoding of  $C_l(i)$ , we can use extrinsic information provided by the decoding of other block codes  $C_l(j)$ ,  $j \neq i$ , within the same global iteration, the speed of convergence is faster than that of the conventional iterative MPD.

### B. Simulation results

In this section, the proposed decoding is applied to rate-1/2 LDPC codes in the IEEE 802.16e standards and is examined in terms of the convergence speed of error performance. We consider binary phase shift keying (BPSK) and additive white Gaussian noise (AWGN) channel. Fig. 6 shows the results of bit error rate (BER) of the length-2304 LDPC code using the proposed decoding with various combinations of  $N_l$  and  $N_g$ . We can find that under the condition of  $N_l \times N_g = k$ , where  $k$  is a positive constant,  $N_l = 1$  and  $N_g = k$  are the best choices. Also included in Fig. 6 are the BER results of the LDPC code using the conventional iterative decoding. We find that the proposed decoding with  $N_l = 1$  and  $N_g = 30$  achieve similar BER as compared to the conventional decoding with  $N_i = 50$ . Hence the proposed decoding can reduce the number of iterations required by up to forty percent and increase the speed of convergence without BER loss as compared to the conventional decoding. For the length-1152 LDPC code, similar conclusion can be made.

## IV. A MEMORY-EFFICIENT VLSI ARCHITECTURE FOR THE PROPOSED ITERATIVE DECODING

### A. Proposed decoder architecture

Fig. 7 shows the VLSI architecture of the proposed decoder. The quantized log-likelihood ratios (channel values) of the received code bits are fed into the decoder. The processing units  $PU_{cn}$  and  $PU_{vn}$  perform the operations at check nodes and variable nodes, respectively, for  $\mathbf{H}_1$ . Please refer to [7] for the detail architectures and the associated quantization parameters of processing units  $PU_{cn}$  and  $PU_{vn}$ . The memory unit  $MU_{cn}$  is used to store the extrinsic value (check-to-variable message)  $R_{ij}[k]$ . At the final  $N_g$  global iteration, the hard decisions

of the code bits are produced by the processing unit  $PU_{hd}$ . Note that the hardware complexities of  $PU_{cn}$  and  $PU_{vn}$  are proportional to  $N_l$  instead of  $N$ . For the (2304, 1152) LDPC code,  $N_l = 63$  and  $N = 2304$ . If we use a fully-parallel architecture to implement the decoder of  $C_l(i)$ , i.e.,  $PU_{cn}$  and  $PU_{vn}$ , we can achieve higher throughput as compared to the pure serial architecture. In addition, the improved convergence speed can further increase the throughput. As compared to the pure serial architecture, we do not need memory unit  $MU_{vn}$ . As compared to the fully-parallel architecture, we do not have complex interconnections since the code length of  $C_l(i)$  is much less than that of  $C$ . Since there are many partially-parallel architectures in the literature, we will compare representative partially-parallel architectures proposed in [12][13][14][15] in Section IV.B. Since the modified MSA (min-sum algorithm) with 5-bit quantization [8] has similar performance loss as compared to the SPA with 4-bit non-uniform quantization proposed in [7], we use the 4-bit non-uniform quantized SPA to implement the decoder of  $C_l(i)$  for saving memory. The simulation results of  $C$  using 4-bit non-uniform quantized SPA are shown in Fig. 8.

### B. Implementation results

The performances of the decoders of (2304, 1152) and (1152, 576) irregular LDPC codes based on the proposed partially-parallel architecture are summarized in Table I. Also included in Table I are the results of LDPC code given in [14][15]. The LDPC decoders reported in [14] and [15] are based on the overlapped architecture proposed in [12] and the TDMP (Turbo-decoding message-passing) architecture proposed in [13], respectively. Since the throughput and area are technology dependent, in Table I, we use total clock cycles for convergence, memory size, and gate counts as measures of performance. As compared to the LDPC decoder presented in [14], our decoder needs a lower number of iterations to reach the performance of convergence and has a slightly larger number of clock cycles per iteration. As compared to the LDPC decoder presented in [15], our decoder needs a slightly larger number of iterations to reach the performance of convergence and has a lower number of clock cycles per iteration. In conclusion, our decoder needs a lower number of clock cycles to reach the performance of convergence as compared to the LDPC decoders presented in [14] and [15]. In addition, our LDPC decoder requires less memory as compared to the LDPC decoder presented in [15] and lower number of gate counts as compared to the LDPC decoder presented in [14].

## V. CONCLUDING REMARKS

We propose a decoding method with improved convergence speed for the LDPC codes used in the IEEE 802.16e standards. Based on this decoding method, we propose a decoder architecture for VLSI implementation. Our decoder does not need to store the variable-to-check messages and hence our decoder is memory efficient. As compared to other LDPC decoders in the literature, the proposed decoder needs a lower number

TABLE I  
COMPARISON OF RATE-1/2 LDPC DECODERS.

	Overlapped Decoder [14]	Proposed Decoder	TDMP Decoder [15]	Proposed Decoder
Code Length	1024	1152	2304	2304
Code Structure	Regular	Irregular	Irregular	Irregular
Memory (bits)	N.A.	15150	60228	30300
Gate Counts	457K	157K	N.A.	157K
Clock Cycles per Iteration	38	48	228	96
Total Iterations for Convergence	50	30	25	30
Total Clock Cycles for Convergence	1900	1440	5700	2880

	94	73								55	83		7	0											
	27												12	0	0										
			24	22	81		33																		
61		47								65	25						0	0							
			39									41	72							0	0				
					46	40					82										0	0			
				95	53							14	18									0	0		
	11	73					2				47											0	0		
12						83	24			43				51								0	0		
							94			59				70	72								0	0	
			7	65						39	49													0	0
43								66	41				26	7										0	

of clock cycles to reach the performance of convergence and requires either less memory or less gate counts. The proposed decoding method and VLSI architecture are not restricted to the LDPC codes used in the IEEE 802.16e standards and can be applied to any Block-LDPC code which is a special class of QC-LDPC codes.

REFERENCES

[1] R. Gallager, "Low-Density Parity-Check Codes," *IRE Trans. Inf. Theory*, vol. 7, pp. 21-28, Jan. 1962.

[2] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular codes," *IEEE Trans. Inform. Theory*, 47(2): 619-37, Feb. 2001.

[3] Hao Zhong; Tong Zhang; "Design of VLSI implementation-oriented LDPC codes" in *Proc. IEEE Semiann. Vehicular Technology Conf.*, Oct. 2003, pp. 670-673.

[4] Hao Zhong; Tong Zhang; "Block-LDPC: A Practical LDPC Coding System Design Approach," *IEEE Transactions on Circuit and Systems-I: regular papers*, Vol. 52, No. 4, pp.766-775, April 2005

[5] S. Lin and D. J. Costello, Jr., *Error Control Coding*. Pearson Prentice-Hall, 2nd ed. 2004

[6] D. J. C. MacKay, "Good Error-Correcting Codes based on Very Sparse Matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 3, pp. 399-431, Jan. 1999.

[7] Lee, J.K.-S.; Thorpe, J.; "Memory-efficient decoding of LDPC codes" , in *Proc. ISIT*, Sept. 2005, pp. 459 - 463

[8] J. Chen and M. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check codes," *IEEE Trans. Commun.*, vol. COM-50, pp. 406-414, March 2002.

[9] (Online) <http://www.ieee802.org/16/tge/>

[10] A. Blanksby and C. Howland, "A 690-mW 1-Gb/s, rate-1/2 low-density parity-check code decoder," *IEEE J. Solid-State Circuits*, vol. 37, no. 3, pp. 404-412, Mar. 2002.

[11] E. Yeo, P. Pakzad, B. Nikolić, and Anantharam, "VLSI architectures for iterative decoders in magnetic recording channels," *IEEE Trans. Magn.*, vol. 37, pp. 748-755, Mar. 2001.

[12] Y. Chen and K. K. Parhi, "Overlapped message passing for quasi-cyclic low-density parity check codes," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 6, pp. 1106-1113, Jun. 2004.

[13] M. M. Mansour and N. R. Shanbhag, "High-throughput LDPC decoders," *IEEE Trans. VLSI System*, vol. 11, no. 6, pp. 976-996, Dec. 2003.

[14] S. H. Kang and I. C. Park, "Loosely coupled memory-based decoding architecture for low density parity check codes," *IEEE Trans. Circuit Syst. I, Reg. Papers*, vol. 51, no. 6, pp. 1106-1113, Jun. 2004.

[15] K. K. Gunnam, G. S. Choi, M. B. Yeary, and M. Atiqzaman, "VLSI architectures for layered decoding for irregular LDPC codes of WiMAX," (Online) <http://www.ieee802.org/16/tge/>

Fig. 1. Block-type parity-check matrix  $H$  of LDPC codes with rate 1/2 in the IEEE 802.16e standards, where the  $(i, j)$  element is  $p(i, j)$ .

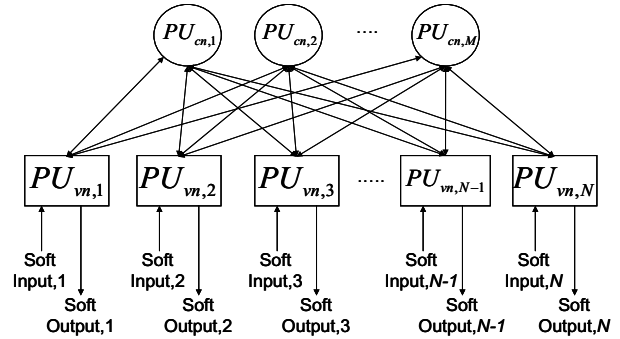


Fig. 2. LDPC decoder using the conventional fully-parallel architecture.

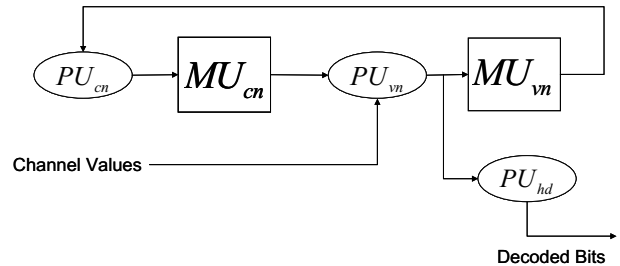


Fig. 3. LDPC decoder using the conventional serial architecture.

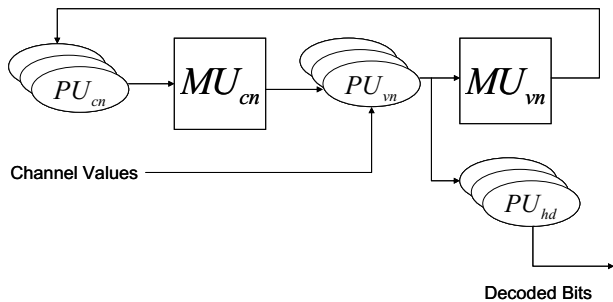


Fig. 4. LDPC decoder using the conventional partially-parallel architecture.

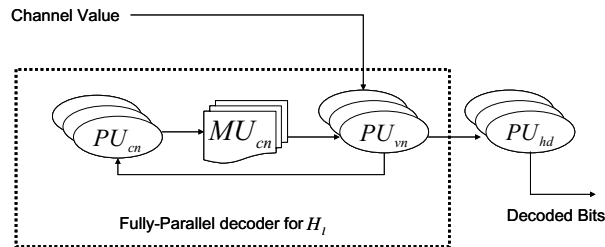


Fig. 7. Proposed VLSI decoder architecture for the proposed decoding algorithm.

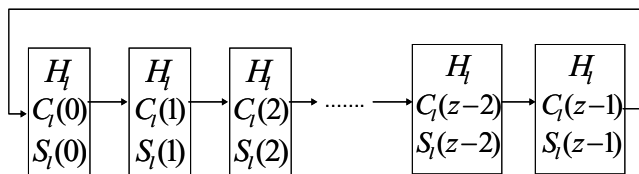


Fig. 5. Proposed decoding method of QC-LDPC code  $C$  based on serial decoding of block codes  $C_i(i)$ ,  $i = 0, 1, 2, \dots, z - 1$ .

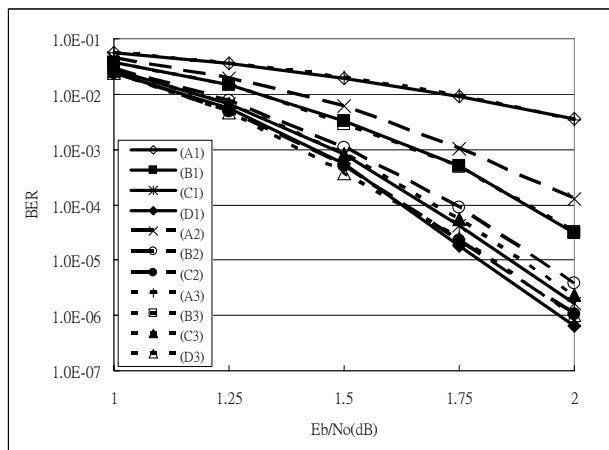


Fig. 6. BER of  $C$  using the proposed or conventional decoding algorithm. The iteration number of conventional decoding is denoted as  $N_i$ . (A1) Proposed,  $N_l = 1$ ,  $N_g = 5$ ; (B1) Proposed,  $N_l = 1$ ,  $N_g = 10$ ; (C1) Proposed,  $N_l = 1$ ,  $N_g = 20$ ; (D1) Proposed,  $N_l = 1$ ,  $N_g = 30$ ; (A2) Proposed,  $N_l = 2$ ,  $N_g = 5$ ; (B2) Proposed,  $N_l = 2$ ,  $N_g = 10$ ; (C2) Proposed,  $N_l = 2$ ,  $N_g = 20$ ; (A3) Conventional,  $N_i = 7$ ; (B3) Conventional,  $N_i = 15$ ; (C3) Conventional,  $N_i = 30$ . (D3) Conventional,  $N_i = 50$ .

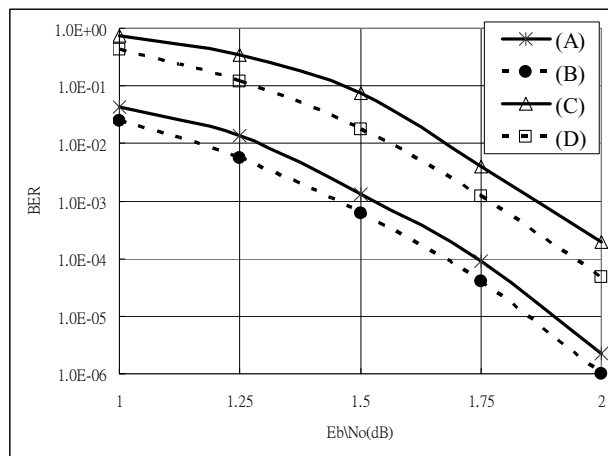


Fig. 8. BER and FER (frame error rate) of  $C$  using the proposed decoding algorithm with  $N_l = 1$  and  $N_g = 25$ . (A) BER, 4-bit quantization; (B) BER, floating point; (C) FER, 4-bit quantization; (D) FER, floating point.