A New Early Termination Strategy for QC-LDPC Codes Based on the Layered Message Passing Decoding

Zhejun Zhang, Guanghui He, Jun Ma
School of Microelectronics, Shanghai Jiao Tong University, Shanghai, China
{albertadulf, guanghui.he, majun}@sjtu.edu.cn

***Abstract* - Low-density parity-check (LDPC) codes are one of the most popular linear block codes have ever been used in the communication system. The standard two-phase message passing decoding (TPMPD) algorithm and the layered message-passing decoding (LMPD) algorithm are widely used in LDPC decoder. However, the convergence speed of LMPD algorithm is much faster than that of TPMPD algorithm, particularly in the condition of high signal noise ratio (SNR). In this paper, we propose an early termination criterion for quasi-cyclic low-density parity-check (QC-LDPC) codes based on LMPD algorithm with fast convergence speed. The strategy is to dynamically reprocesses the uncorrected related layers while skipping the corrected ones. Simulations show that 3% reduction in terms of iteration numbers can be achieved with no performance loss compared to the traditional termination strategies, such as hard decision aided (HDA) strategy and parity check equations (PCE) strategy. Furthermore, the optimized early termination strategy achieves low complexity in hardware implementation.**

Keywords - Convergence speed, Hard decision aided (HDA); Quasi-Cyclic Low-Density parity-check (QC-LDPC) codes; Termination strategy.

I. Introduction

Low-density parity-check (LDPC) code was first introduced by Gallager [1] in 1962. In 1995, Mackay and Neal found that the decoding performance of LDPC code could nearly approach the Shannon Limit [2]. For the near-capacity performance and the potential of parallel implementation of decoders, LDPC codes have been adopted for several industrial standards, such as IEEE 802.16e, IEEE 802.11n, and China Mobile Multimedia Broadcasting (CMMB).

There are various algorithms such as two phase massage passing decoding (TPMPD) algorithm [3] and layered message passing decoding (LMPD) algorithm [10] proposed for decoding LDPC codes. In TPMPD algorithm, each iteration is processed by two phases, the check node processing (CNP) and the variable node processing (VNP). The CNP operates all the check nodes and passes the messages to the variable nodes while the VNP operates all the variable nodes and passes the messages to the check nodes. However, in the LMPD algorithm, one iteration consists of several sub-iterations. The CNP operates partial check nodes and passes the messages to the variable nodes while the VNP operates the partial results of the check nodes and updates all the a posteriori probability (APP) values for the CNP in next sub-iteration. Compared with TPMPD algorithm, LMPD algorithm achieves better decoding performance and faster convergence speed, especially in the high Signal Noise Ratio (SNR) condition.

The early termination strategy for the LMPD algorithm is an important technique since it can reduce the hardware complexity, the number of sub-iterations, and the power consumption significantly in hardware implementation. There are various algorithms such as hard decision aided (HDA) [11] strategy, parity check equations (PCE) strategy and partial parity check equations (PPCE) strategy proposed for LDPC termination strategies. However the HDA strategy consumes large memory while the PPCE and PCE strategies have high computation complexity. In this paper, an optimized early termination criterion is proposed with lower hardware complexity, faster convergence speed, and almost the same decoding performance compared with the traditional strategies.

The rest of this paper is organized as follows. Section II describes LDPC decoding algorithms and the traditional termination strategies. Section III proposes a new early termination strategy for QC-LDPC codes. Section IV optimizes the early termination strategy for hardware efficient implementation. Section V concludes the paper.

II. Preliminaries

There are several decoding algorithms for LDPC codes, such as belief propagation (BP) algorithm, sum product (SP) algorithm, min-sum (MS) algorithm, normalized min-sum (NMS) algorithm[5], offset min-sum (OMS) algorithm [9] and BCJR[4] decoding algorithm. Among these algorithms, BP, SP and BCJR decoding algorithm have the better performance, while MS, NMS and OMS algorithms have low complexity in terms of hardware implementation. On the other hand, fully parallel architecture [6], partially parallel architecture [7] and serially architecture [8] are adopted for different purposes in LDPC decoders. The fully parallel architecture provides high throughput at the cost of high complexity. The partially parallel architecture takes both complexity and throughput into consideration, and is applied to many LDPC hardware implementations.

Consider a QC-LDPC code, the parity check matrix H is M×N. Then this H matrix can be represented by a base matrix $H_b = M_b \times N_b$, each element in this H_b is a quasi-cyclic identical matrix with the size of Z×Z, $M = M_b \times Z$ and $N = N_b \times Z$, Z is the extension factor. Thus LMPD algorithm can

This work is supported in part by Shanghai Natural Science Foundation under Grant No. 10ZR1416500, Shanghai PuJiang Program under Grant No. 09PJ1405400 and 863 Program of China under Grant No. 2009AA011705.

be divided into M_b sub-iterations and each sub-iteration paralleling operates Z check nodes. Therefore the LMPD algorithm processes M_b layers for one iteration. The detailed description of HDA, PCE, and PPCE strategy is given below.

A. *Hard-decision aided (HDA)*

After each iteration, the HDA criterion tries to find out the difference in hard-decision output between two consequent iterations. If there's no difference in hard-decision output between these two consequent iterations, the decoder comes to an end. Otherwise, the decoding process continues until it reaches the maximum iteration number.

Obviously, this kind of termination criterion should process at least two iterations. The throughput will be limited in an ultra high SNR condition which only needs one decoding iteration.

B. *Parity Check Equations (PCE)*

The PCE criterion uses the parity check matrix H and the hard-decision codes C. According to the equation $HC^T=0$, if C satisfies the equation, the decoder is terminated and outputs the hard-decision. Otherwise the decoding continues until it reaches the maximum iteration number.

Generally, the length of many LDPC codes is over one thousand. Thus, the calculation of the equation will be large and as a result it may limit the throughput. In addition, the additional processing units are required in hardware implementation.

C. *Partial Parity Check Equations (PPCE)*

The PPCE criterion is similar to the PCE criterion. The difference is that the PPCE uses H_p and C_p to replace H and C. In LMPD algorithm, the parity check matrix H is partitioned into $P = M_b$ layers denoted as $H_0, H_1, \ldots, H_{Mb-1}$ respectively, where H_p is an Z×N sub-matrix and C_p is the decoding code of H_p, $0 \le p \le M_b-1$. Consider the equation $H_pC_p^T=0$, in one iteration, if all the partial results C_p ($0 \le p \le M_b-1$) satisfy the equation, the decoder is terminated and outputs the hard-decision. Otherwise it will continue to process the next iteration until it reaches the maximum iteration number.

PPCE strategy consumes about $1/ M_b$ processing units of that in PCE strategy to calculate the parity check equations. The decoding performance of PPCE may slightly decrease.

III. New Early Termination Strategy

Consider a 6×12 parity check matrix H in Fig. 1. H is a regular matrix, and its column weight is 2 and the row weight is 4. The LMPD algorithm is adopted to decode this LDPC code. Each iteration is divided into 6 sub-iterations with the parallelism factor to be 1. As is described in PPCE, the termination criterion will use 6 sub-equations to judge whether the decoding process should be stopped. We denote the hard-decision codes to be $C_0, C_1, \ldots, C_{11}$. Suppose that in PPCE, the second sub-iteration doesn't satisfy the equation $H_pC_p^T=0$ while all the other sub-iterations satisfy it. Observing that the C_p of the second sub-iteration contains C_0, C_2, C_5, C_{10}, and it represents that errors most probably occur in C_0, C_2, C_5 and C_{10}. Thus, it only needs to reprocess the sub-iterations whose C_p are related with C_0, C_2, C_5 and C_{10}. Then the first, the fourth and the fifth sub-iterations are found to be the error related sub-iterations. In the following iterations, the error related sub-iterations are reprocessed. It takes only four sub-iterations in the following iterations, and this method reduces two additional sub-iterations compared with the PPCE criterion.

$$\begin{array}{c|cccccccccccc}
 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\
\hline
1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
2 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
3 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
4 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
5 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1
\end{array}$$

Fig. 1 6×12 H matrix

A. Algorithm

As described above, the new early termination strategy skips the correct rows while reprocessing those error related rows to decrease the number of sub-iterations with no performance loss. In a QC-LDPC code, the M×N parity check matrix H is divided into M_b layers according to the rows of the H_{base} matrix. Each layer contains Z rows of H. Let v_p be a row vector which contains the decoded bits associated with code C_p for $p = 0, 1, \ldots, M_b-1$. We denote the 1×N row vector J from J_0 to J_{N-1} as the judging bits for N columns in H matrix to present whether the error occurs in a specific column. Let Ω be a set which contains the indices of the layers that should be processed in the following iterations. The content of Ω will be dynamically updated based on the vector J. $L_M = I_M M_b$ is the maximum number of layers that is allowed to decode one codeword, and I_M is the maximum iteration number. In addition, t and L denote the number of sub-iterations contained in Ω and the total number of processed layers respectively. The detailed algorithm is described as follows.

Step 1 [Initialization] $L = 0$, $t = M_b$, $\Omega=\{0, 1, \ldots, M_b-1\}$.

Step 2 Set $J_0 = 0, J_1 = 0, \ldots, J_{N-1} = 0$.

Step 3 [Layered process] Decode C_k, for all $k\in\Omega$sequentially. Increase L by one after decoding every layer. If $L = L_M$, go to Step 7.

Step 4 [Set judging flags Process] If $H_kv_k^T \neq 0$, set $J_z = 1$ ($H_k(i, z) = 1$, $0 \le i \le Z-1$), $0 \le k \le M_b-1$.

Step 5 [Update layers] Decode C_k, for all $k\in\Omega$sequentially. If all $J_z = 0$ ($H_k(i, z) = 1$, $0 \le i \le Z-1$), Delete k from Ω. Increase L by one after decoding every layer. If $L = L_M$, go to Step 7.

Step 6 [Early termination] If $\Omega=\Phi$, go to step 7. Otherwise go to Step 2.

Step 7 [Decision] Make hard decision based on the signs of the a posteriori probability (APP) values. Then terminate the decoder.

B. Simulation Results

In this paper, we adopt the (9216, 4608) Block-LDPC code specified in China Mobile Multimedia Broadcasting (CMMB) standard to simulate the proposed strategy. The base matrix of this code H_b is 72×144, and the parallelism factor is 64. The partially parallel LMPD based on NMS algorithm withα = 0.8 is used to be the decoding algorithm in simulation. The average number of processed sub-iterations (ANPS) is defined to indicate the convergence speed. Fig. 2 and TABLE I respectively show the BER results and ANPS results of using HDA, PCE, PPCE, and the new early termination strategy which is presented as NET. Fig. 2 shows that these four termination strategies are almost the same in decoding performance. Table I indicates that the ANPS with the proposed strategy is about 3% less than other strategies, especially in high SNR condition.

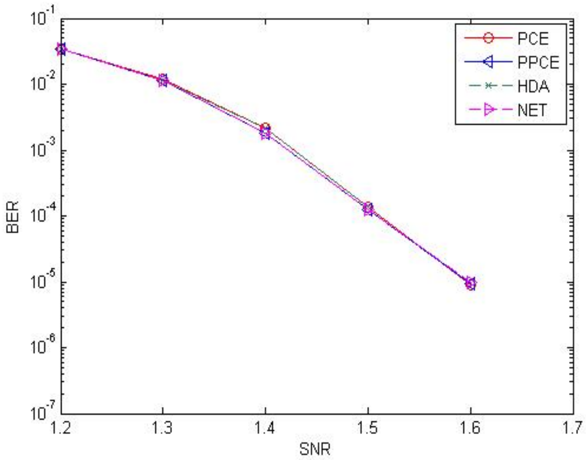

Fig. 2 BER performance of C using various termination strategies (L_M = 2160) with LMPD NMS (α = 0.8)

Table I Anps Performance of C Using Various Termination Strategies (l_m = 2160) with LMPD NMS (α = 0.8)

SNR\ANPS	PCE	PPCE	HDA	NET
1.2 dB	1896.2	1901.7	1893.3	1893.6
1.3 dB	1495.5	1493.4	1497.3	1479.6
1.4 dB	1120.2	1110.9	1121.2	1094.1
1.5 dB	875.3	873.2	874.8	855.4
1.6 dB	738.6	738.1	736.3	720.8

IV. Optimized Early Termination Strategy For Hardware Implementation

In section III, we proposed a new early termination strategy for decreasing the ANPS with no performance loss. However, the process operated in Step 4 for calculating the results of $H_p v_p^T=0$ consumes lots of hardware resources. In addition, in a long codeword LDPC code, the memory for the judging flags from J_0 to J_{N-1} is large. Consider these two points mentioned above, we optimize the early termination strategy for hardware implementation in this section.

A. Algorithm Of Optimized Early Termination Strategy

Consider a QC-LDPC code, the parity check matrix H can be represented by the base matrix H_b and the extension factor Z. The parallelism factor chooses the extension factor Z for partial parallel decoding in LMPD algorithm. In the early termination criterion proposed above, we use the 1×N vector J to store the judging flags of the error columns. Consider the base matrix H_b, each column of H_b represents a bunch of columns in H. Thus the Z columns in H can be gathered to the corresponding column in H_b. As a result, the number of judging flags J is reduced to N_b. A 1×N_b vector J' is used to store the judging flags in the optimized strategy. And the memory for the judging flags becomes one Z^{th} of the early termination criterion proposed in section III. The checking units (CUs) are designed to replace the operation of calculating equation $H_p v_p^T = 0$ in the strategy described above. The CUs are implemented to calculate the sum of C_p in GF(2). This modification can largely reduce the hardware complexity from the matrix multiplication to the sum in GF(2).

t is denoted as the number of layers stored in Ω, and it is initialized to be M_b. The length of vector J' is N_b and J' is represented as J'_0, J'_1, …, J'_{Nb-1}. The variable T represents the early termination flag, it keeps zero when the results of checking units (CUs) in all layers are zero. The zero value of T is a sign to terminate the decoder in this strategy. The variable s counts for the processed layers in an iteration. The parallelism of the checking units (CUs) is Z, and CUs are implemented to calculate the sum in GF(2) of the hard-decision of C_p in the specific rows. The judging unit (JU) is designed to check the judging flags related to the specific layer k according to the base matrix H_b. The algorithm of the optimized early termination strategy is described as follows.

Step 1 [Initialization] L = 0, t = M_b, $\Omega=\{0, 1, \ldots, M_{b-1}-1\}$.

Step 2 Set $J'_0 = 0$, $J'_1 = 0$, … , $J'_{Nb-1} = 0$, Set T = 0, s = 0.

Step 3 [Layered process] Decode C_k, for all k∈Ωsequentially. If any of the results of CUs is not zero, set $J'_m = 1$ (m| $H_b(k,m) = 1$, $0 \le m \le N_b$) and T = 1. Increase L and s by one after decoding every layer. If L = L_M, go to Step 7. Otherwise, if s = t, go to Step 4.

Step 4 [Early Termination] If T = 0, go to step 7. Otherwise set s = t and go to Step 5.

Step 5 [Re-process] Decode C_k, for all k∈Ωsequentially. If the result of JU is zero, delete k from Ω, decrease t by one. Increase L and decrease s by one after decoding every layer. If L = L_M, go to Step 7. Otherwise, if s = 0, go to Step 6.

Step 6 [Early Termination] If $\Omega=\Phi$, go to Step 7. Otherwise go to Step 2.

Step 7 [Hard Decision] Make hard decision based on the signs of the APP values.

Step 8 [Free Memory] FreeΩ and terminate the decoder.

B. *Simulation of Optimized Early Termination*

We use the (9216, 4608) Block-LDPC code specified in CMMB standard in the simulation. Fig. 3 shows the decoding performance by using three different termination strategies (OET denotes the optimized early termination strategy). The results in Fig. 3 show that the optimized criterion performs nearly the same performance as PPCE or NET strategy. Table II illustrates the average number of processed sub-iterations (ANPS). As listed in Table II, the optimized strategy gets almost the same ANPS as PPCE strategy in low SNR condition (1.2 dB, 1.3 dB, 1.4 dB) while it consumes about 2% ANPS less than PPCE strategy in the SNR condition (1.5 dB, 1.6 dB). However the optimized strategy processes about 1% ANPS more than criterion proposed in section III in the SNR condition (1.5 dB, 1.6 dB). Table III describes the hardware complexity of these strategies. The optimized strategy reduces about 98% memory on comparing with the HDA strategy while it achieves lower complexity in calculation for hardware implementation contrast to PCE and PPCE strategy.

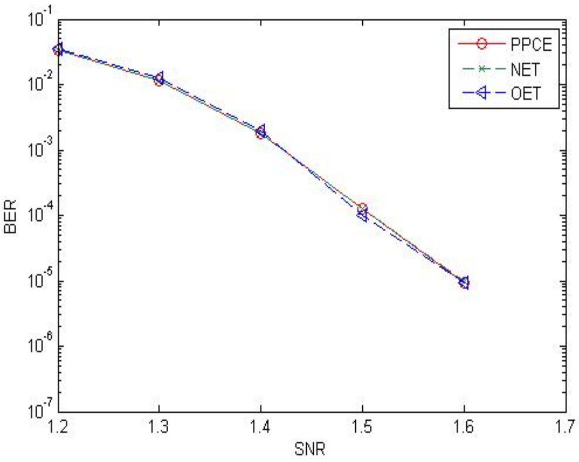

Fig. 3 BER performance of C using various termination strategies (LM = 2160) with LMPD NMS (α = 0.8)

Table II Anps Performance of C Using Various Termination Strategies (LM = 2160) with LMPD NMS (α = 0.8)

SNR\ANPS	PPCE	NET	MET
1.2 dB	1901.7	1893.6	1913
1.3 dB	1493.4	1479.6	1491.1
1.4 dB	1110.9	1094.1	1110.4
1.5 dB	873.2	855.4	861.7
1.6 dB	738.1	720.8	727.2

Table III Comparasion of Hardware Complexity

Strategies/ Complexity	Adders	Comparators	Multiplication	Memory
PCE	4608×9216-1	/	4608×9216	/
PPCE	64×9216-1	/	64×9216	/
HDA	/	9216	/	9216 bits
NET	64×9216-1	/	64×9216	9216 bits
OET	6×64-1	/	/	144 bits

V. Conclusion

The proposed early termination strategy achieves faster convergence speed over the existing LMPD termination strategy. About 3% iteration number is reduced with no decoding performance loss. However, this early termination strategy is not hardware efficient due to the large storage memory for judging flags and the high calculation complexity for the matrix multiplication. Furthermore, we propose an optimized strategy for a hardware efficient implementation. With the memory reduction of the judging flags and the simplification of the checking calculation, it reduces 98% in storage memory as compared to HDA strategy and achieves lower calculation complexity as compared to PCE and PPCE strategy.

References

[1] G. Gallager, "Low density parity check codes," IRE Trans. Info. Theory, vol. IT-8, pp. 533-547, Jan. 1962.

[2] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," Electron. Lett., vol. 33, pp. 457–458,Mar. 1997.

[3] 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.

[4] M. M. Mansour and N. R. Shanbhag, "A 640-Mb/s 2048-bit programmable LDPC decoder chip," IEEE JSSCC., vol. 41, no. 3, pp. 634-698, Mar. 2006.

[5] J.-H. Chen, "Reduced Complexity Decoding Algorithms for Low-Density Parity-Check Codes and Turbo Codes," Ph. D. dissertation, University of Hawaii, USA, 2003.

[6] C. J. Howland and A. J. Blanksby, "Parallel decoding architectures for low density parity check codes," in Proc. IEEE ISCAS, May 2001, vol. 4, pp. 742-745.

[7] Z. Cui and Z. Wang, "Area-efficient parallel decoder architecture for high rate QC-LDPC codes," in proc. IEEE ISCAS, May 2006, pp. 5107-5110.

[8] M. Cocco, J. Dielissen, M. Heijligers, A. Hekstra, and J. Huisken, "A scalable architecture for LDPC decoding," in Proc. IEEE DATE, Feb. 2004, vol. 3, pp. 88-93.

[9] Y. Sun, M. Karkooti, J. R. Cavallaro. "VLSI Decoder Architecture for High Throughput Variable Block-size and Multi-rate LDPC Codes," in Proc. IEEE ISCAS, New Orleans, USA, May, 2007:2104-2107.

[10]S.-C. Chou, M. K. Ku, and C. Y. Lin, "Switching Activity Reducing Layered Decoding Algorithm for LDPC Codes," in Proc. IEEE ISCAS, pp.528-531, Seattle, USA, May 2008.

[11]R.-Y. Shao, S. Lin, and M.P.C. Fossorier, "Two simple stopping criteria for turbo decoding," in IEEE Trans. on Commu., Vol. 47, pp. 1117-1120, Aug. 1999.