An Energy-efficient High-throughput Multi-rate QC-LDPC Decoder Supporting G.hn Applications

Hexi Hou, Jun Ma, Jiangpeng Li, and Guanghui He
School of Microelectronics
Shanghai Jiao Tong University, Shanghai
E-mail: {houhexi,majun,lijiangpeng,guanghui.he}@sjtu.edu.cn

***Abstract*—In this paper, an LDPC decoder fully compliant to the G.hn standard is presented. The decoder implements one kind of Layered Decoding Algorithm – turbo decoding massage passing (TDMP) algorithm for Quasi-Cyclic LDPC codes. A simple early termination strategy is presented to significantly increase the throughput and reduce the power consumption. Partially parallel architecture with an overlapped mechanism is proposed to improve the hardware efficiency. For the newly published standard G.hn, the uppermost expansion factor supported is 360 which is much larger than that in the 802.16e standard which brings higher throughput. In the architecture, look-up tables replace the permutation network to further reduce hardware complexity. Based on these methods, an LDPC decoder supporting the G.hn applications is implemented using a SMIC 130nm CMOS process. The decoder consumes 150 Kbits memory while achieving 1.54 Gbps throughput.**

I. INTRODUCTION

LOW-DENSITY Parity-Check (LDPC) codes, as one of the forward error correction (FEC) codes, first introduced by Gallager [1] in 1962 and rediscovered by MacKay and Neal [2] in 1996, were proved to approach the Shannon limit. LDPC codes have the potential for highly parallel decoder implementation that can achieve throughput ranging from Mbps to Gbps. These properties make LDPC codes one of the most attractive topics of interest in both academia and industry. Recently, LDPC codes have been employed in many transmission standards for wireless communication such as IEEE 802.11n, IEEE 802.16e, DVB-S2, and G.hn.

Structured codes such as quasi-cyclic (QC) LDPC codes [3], being a special class of LDPC codes, are well-suited for hardware implementation for the regularity of their parity check matrices. Recently, many VLSI implementations are presented for QC-LDPC codes in communication systems such as IEEE 802.16e systems [4] and China Multimedia Mobile Broadcasting (CMMB) systems [5].

Gallager's two-phase message passing (TPMP) algorithm [1] decodes a codeword by updating the messages between check nodes and bit nodes iteratively. It achieves the optimal performance at the cost of more complexity. The Min-Sum algorithm [6] achieves lower complexity with performance degradation. Some variations include normalized min-sum algorithm [7] and offset min-sum algorithm [8]. However, the convergence speed of these algorithms is slow, which limits high throughput applications. Another iterative decoding algorithm is layered decoding or turbo-decoding message passing (TDMP) [9], which carries out check node and bit node messages updating concurrently. The TDMP decoding algorithm convergences faster and needs less iterations to achieve higher throughput.

This work is supported in part by Shanghai PuJiang Program under Grant NO. 09PJ1405400 Shanghai Natural Science Foundation under Grant No. 10ZR1416500, and Hi-Tech Research and Development Program (863) under Grant No. 2009AA011705

TABLE I
ALL MODES FOR LDPC DECODER DESIGN

Code-rate	Codeword n(bits)	Info bits k(bits)	Code-rate	Codeword n(bits)	Info bits k(bits)
1/2	1920	960	16/18	1080	960
	8640	4320		4860	4320
2/3	1440	960	20/21	1008	960
	6480	4320		4536	4320
5/6	1152	960			
	5184	4320			

In this paper, a high-throughput and low complexity QC-LDPC decoder is proposed which can support up to360 parallelism and all the code rates and lengths defined in the G.hn standard [10]. In addition, a simple early termination strategy is adopted in the design to achieve high throughput and low power consumption. For multi-mode implementation, permutation network is replaced by look-up tables to further reduce hardware complexity. Based on the proposed methods, the designed LDPC decoder can achieve considerable high throughput with comparable low hardware complexity and can support the code-words defined in G.hn, as shown in Table I. In the following parts of the paper, code word (M, N, R, L) means the QC-LDPC code base matrix size of $M \times N$, the code rate is R and the length is L.

The rest of this paper is organized as follows. Section II introduces the overlapped layered min-sum decoding algorithm. Section III describes the low complexity structure of the partial layered decoder. The implementation results and conclusions are presented in section IV and V, respectively.

II. OVERLAPPED LAYERED MIN-SUM DECODING WITH EARLY TERMINATION STRATEGY

A. *Layered normalized min-sum algorithm*

Decoding algorithms based on the TPMP and their variants such as min-sum or normalized min-sum algorithms convergence speed is low due to the fact that the check node updating and the bit node updating are carried out separately

TABLE I
TYPE SIZE FOR PAPERS

(1)		94	73						55	83			7	0										
(9)	12				83	24		43				51									0	0		
(5)			39				84			41	72						0	0						
(12)	43					66		41				26	7											0
(7)			95	53						14	18								0	0				
(10)						94		59			70	72										0	0	
(4)	61		47						65	25						0	0							
(2)		27				22	79	9				12		0	0									
(11)			7	65					39	49													0	0
(6)					46	40		82				79	0					0	0					
(8)		11	73				2			47										0	0			
(3)				24	22	81		33				0			0	0								

which are not ideal for high throughput implementation. In this design, TDMP algorithm is adopted to achieve high throughput. To reduce the computational complexity and enhance error correction performance, layered normalized min-sum algorithm is chosen in this paper.

In the layered method, the parity check matrix H is partitioned into M layers with each layer corresponding to one submatrix, $H^T=[H_1^T, H_2^T, \ldots H_M^T]$. The column weight of each submatrix can not exceed 1. Each iteration is partitioned into M sub-iterations with each sub-iteration decodes a sub-code and updates the bit node information. Let $Lr_{ij}^{(k,m)}$ and $Lq_{ji}^{(k,m)}$ respectively denote the check-to-bit soft message for check node *i* to bit node *j* and the bit-to-check soft message for bit node *j* to check node *i* in the *k*th iteration at the *m*th ($0 \le m \le M-1$) layer. $LQ_j^{(k,m)}$ is the log-likelihood (LLR) soft message of variable node *j* in the *k*th iteration at the *m*th layer. At the beginning of each sub-iteration, Lq_{ji} is recovered by LQ_j and Lr_{ij} . Then check node updating executes, which formulated below:

$$Lq_{ji}^{(k,m)} = LQ_j^{(k,m-1)} - Lr_{ij}^{(k,m-1)} \quad (1)$$

$$Lr_{ij}^{(k,m)} = \left(\prod_{\substack{j'\in N(i)\\ j'\neq j}} sign(Lq_{j'i}^{(k,m-1)})\right) \times \min_{\substack{j'\in N(i)\\ j'\neq j}} (|Lq_{j'i}^{(k,m-1)}|) \times \alpha \quad (2)$$

$$LQ_j^{(k,m)} = Lq_{ji}^{(k,m)} + Lr_{ij}^{(k,m)} \quad (3)$$

Where *N(i)* denotes the set of variable nodes connected to the check node *i*. α is the normalized factor, which in this design, according to the G.hn standard code-words, is assigned with 0.8 by simulation. At the end of the decoding procedure, the decoded information bits are decided by the sign of LQ_j .

B. Overlapped decoding scheme

As previously discussed, each sub-iteration can be viewed as 3 steps: read from the memory, compute for updates, and write back to the memory. If these steps work separately and serially, it will result in hardware inefficiency. In [11], for one check node, all the corresponding bit-to-check messages are read simultaneously and computed altogether, but this is not suitable for high parallelism design such as the G.hn standard. The decoder in [4] searches for the dependency of adjacent sub-codes and reduces writing back updated messages. Unfortunately, that needs extra register files and more complex control logic. In the proposed design, an overlapped decoding scheme is used to reduce the stalls and improve the hardware efficiency. Fig. 1 gives the overlapped scheme.

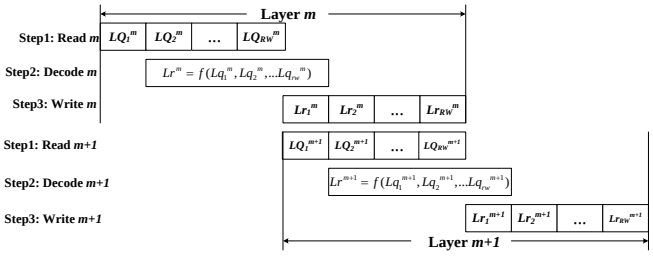

Fig. 1 Overlapped scheme of decoder

Let RW represents the row weight of the H matrix. In each sub-iteration, layer *m*, the LLR messages are read from the memory serially which will take RW cycles. During the reading procedure, the decoding unit repeatedly updates check-to-bit message. At the writing step, the updated LQ_j is written back to memory serially. Meanwhile, the reading procedure for layer *m+1* executes so that dual-port ram is needed. This overlapped scheme can significantly improve the throughput and hardware efficiency by about 50%. There is still another problem that the writing and reading for next layer may conflict when the reading address has not been written. To avoid this, some row transformation has to be done to the base matrix of H. Table 1 gives the transformed table for code rate 1/2 which ensures the successive layers do not have the same column index. For some codes which conflict can not be avoided, the overlapped scheme changes the reading order of LQ_j , as presented in Table 1, the darked 70 in row-6 is read at the last of the layer.

C. Early termination strategy

In earlier implementations of LDPC decoders, the decoding procedure ends and outputs the decoded bits when a predefined max iteration num is reached which is not economical when signal-to-noise ratio (SNR) is high since many cycles and power are wasted, or check the valid code word *v* with equations $Hv^T = 0$ which takes too much hardware complexity for checked equations when the code word is long. The early termination strategy adopted in [12] detects the non-converged code word, but can not terminate converged case in advance. A flexible early termination strategy is proposed in [4] which compares decoded results in two successive iterations. In this design, for (8640, 4320)

LDPC codes, the comparison operation still takes much time and so a simple but efficient termination strategy is employed.

The early termination stores the sign bits and compares of LLR corresponding to the successive sub-iterations. If all the decoded results in the two successive sub-iterations are the same which means unchanged, the *et_flag* is added 1, otherwise, the *et_flag* is set to 0. The decoder is informed to terminate operations when *et_flag* equals the num of layers which in the proposed design is M. Compared with [11], in this way, the check mechanism takes less extra executed clock cycles and less iterations is needed on average.

D. Simulation results

To verify the proposed algorithm and proposed early termination strategy, a (24, 8, 2/3, 8640) LDPC codes with code rate 2/3 defined in the G.hn standard is chosen for simulation. The quantity of simulation data is 10^8 bits. The TDMP normalized min-sum algorithm terminated by a predefined maximum iterations, TDMP normalized min-sum algorithm with early termination strategy $Hv^T = 0$, and the proposed algorithm with early termination are simulated. As shown in Fig. 1, the proposed algorithm maintains error correction performance as good as the optimal TDMP min-sum algorithm but need less cycles which brings higher throughput.

III. MULTI-MODE PARTIAL PARALLEL DECODER ARCHITECTURE

A. Quantization scheme

In hardware implementation of LDPC decoders, memory takes most area. And the word length of the LLR messages and extrinsic messages directly determines the total area of the decoder. But simply reduce the word length may greatly decrease the performance. So it is important to determine proper word length with acceptable performance degradation.

Fig. 2 shows the BER curves on the additive white Gaussian noise (AWGN) channel for different quantization schemes. The scheme (n, k) means n bits quantization with 1 sign bit and k bits fraction part. As the curves show, integer part greatly influences the decoding performance. For trading off, a (6, 3) uniform quantization scheme is employed.

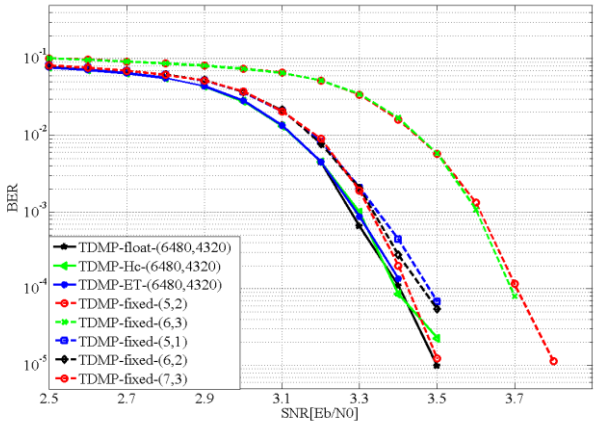

Fig. 2 Simulation for (24, 8, 2/3, 6840) code in G.hn

B. Overall Proposed Decoder Architecture

The proposed TDMP iterative decoder architecture, which is shown in Fig. 2, is composed of several parts: 1) central controller; 2) input module; 3) LLR modules and extrinsic messages modules which stores the check-to-bit messages; 4) routers and de-routers; 5) parallel process units(PUs); 6) ROMs stores the column index of the base matrix and the offset value of the parity check matrix of H; 7) Early termination module; 8) Output module.

The LLR memory, which stores the initial LLR messages from the channel and updated LLR messages, are two-port memories. Then messages can be read out for next sub-iteration while the updated bit-to-check messages being written back without conflict. To support all the code words defined in the G.hn standard, the upmost parallel degree which is 360 corresponding to (24, 12, 1/2, 8640). In each single clock cycle, totally 360×6 bits information should be read out, so 18 two-port RAMs with 24×120 bits are needed. The EX_memory which stores the extrinsic messages, are single-port RAMs. Extrinsic messages are stored in the compressed form, that is for each single check node, all check-to-variable messages are compressed to minimum magnitude, second minimum magnitude, minimum index and the sign bits. The maximum bits for one cycle is $216\times(5+5+20+5)$ corresponding to (24, 4, 5/6, 5184), so 64 single-port 12×120 RAMs are needed.

Messages read from the LLR RAMs and EX memories are distributed to the process units by the LLR_router and the EX_router and the updated LLR messages and extrinsic messages are written back by LLR_derouter and EX_derouter. The reference address information directly comes from Index-Rom and Offset-Rom in which respectively stores column index and offset value for all code words including modes and rates. The size of Index-Rom is 24×100 bits and Offset-Rom is 48×140 bits for all the codes and modes.

In each sub-iteration, the Early Termination function stores the signs of LLR messages read from the memory, and compares with updated LLR messages produced by the PU. Then it carries out the previously presented early termination strategy.

C. Architecture of Processing Unit

The processing unit is composed of calculate unit, recover

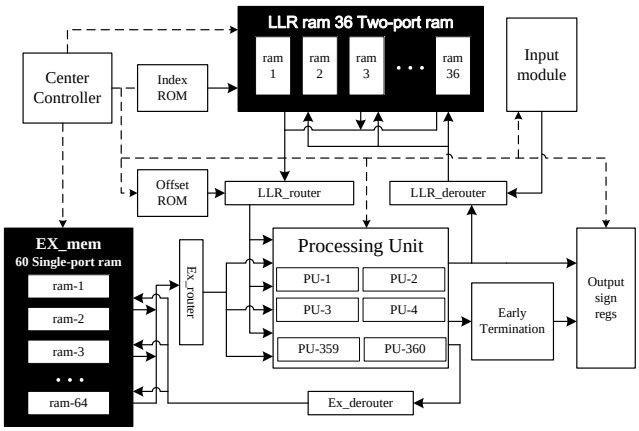

Fig. 3 Overall architecture of LDPC decoder

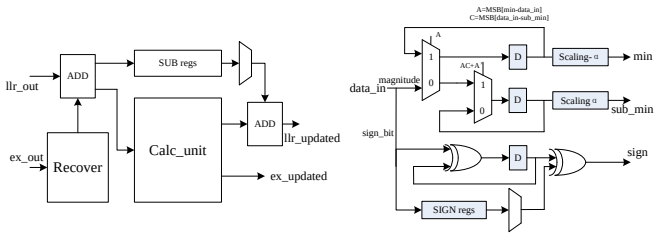

Fig. 4(a,b) Architecture of processing and calculate unit

unit, add module and the sign-bit registers. The recover unit transforms the input compressed extrinsic messages to serially check-to-variable messages based on the minimum flag. The sub-regs temporarily stores the subtraction results and then output the results to update the LLR messages. To get the updated LLR messages, according to the equation (2), the magnitude and sign-bit are respectively computed by the calculation unit which can be divided into two parts: one is for the magnitude computation, the other is for sign-bit generation. To solve the irregular issue in the code word of the G.hn standard, for implementation simplicity, set the input of the irregular bit as the most supported positive number to avoid affecting final results.

To support all the modes and rates presented in the G.hn standard, for partially parallel decoding, up to 360 processing unit should be instantiated. For energy saving, in decoding other words, unused processing units will be powered off.

IV. Implementation Results

Based on the proposed architecture, a multi-mode QC-LDPC decoder for G.hn system is implemented in the SMIC 130nm CMOS technology. The decoder supports all the code words defined in the G.hn standard. The parallel degree has a maximum value with 360 for (24, 12, 8640, 1/2) code. The decoder proposed in this design can run at a peak frequency of 300 MHz with a throughput of about 1.54 Gbps at the maximum 10 iterations and occupy an area of about 12.75 mm^2 with 150 Kbits memory. The working frequency and the maximum iteration number can be adjusted to satisfy the throughput requirements of different systems. Table III shows the main parameters of the proposed decoder with some other LDPC decoders similar to G.hn or 802.16e standards. The decoder gets much higher working frequency and throughput for higher parallel degree at the cost of increased area. Further work can be done to reduce the area complexity.

Table III
Type Size for Papers

	[4]	[12]	[9]	Proposed
Technology(nm)	180	180	180	130
Codeword Lengths	576~2304	7493	2048	1920~8640
Parallelism	Partial	Partial	Partial	Partial
Max Iterations	2~8	15	10	10
Frequency(MHz)	83.3	125	125	300
Throughput(Mbps)	60~222	104.5	80	1540
Memory(Kbits)	76.8	236	51.7	150
Area(mm^2)	8.29	9.76	14.3	14.75

V. Conclusion

In this paper, an energy efficient high throughput multi-mode QC-LDPC decoder supporting the G.hn standard is presented. The proposed decoder uses TDMP normalized min-sum algorithm with overlapped schedule which achieves high throughput. To further increase the throughput, the decoder employs a simple early termination strategy and use a large parallel degree. For multi-mode design, two look-up tables are adopted replacing the permutation network to reduce the hardware complexity. To verify the proposed architecture and design techniques, a QC-LDPC decoder used for the G.hn standard is implemented in the SMIC 130 nm which can achieve a maximum throughput of 1.54 Gbits/s at the maximum 10 iterations with memory size of 150 Kbits. The decoder can be easily modified for other applications with similar algorithms.

References

[1] R. G. Gallager, "Low density parity check codes," *IRE Trans. Inf. Theory*, vol. IT-8, no. 1, 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, no. 18, pp. 1645-1446, 1996.

[3] M. Fossorier, "Quasi-cyclic low-density parity-check codes from cirulant permutation matrices", *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1788-1793, Aug. 2004.

[4] X. Shih, C. Zhan, C. Lin and A. Wu, "An 8.29 mm^2 52 mW Multi-Mode LDPC Decoder Design for Mobile WiMAX System in 0.13 μ m CMOS Process," IEEE J. Solid-State Circuits, vol. 43, pp. 672-683, Mar. 2008.

[5] J. P. Li, G. H. He, H. X. Hou, Z. J. Zhang and J. Ma, "Memory Efficient Layered Decoder Design with Early Termination of LDPC Codes", in *Proc. IEEE ISCAS*, May 2011

[6] J. Chen and M. C. Fossorier, "Decoding low-density parity-check codes with normalized APP-based algorithm", in *Proc. IEEE GLOBECOM*, Nov. 2001, vol. 2, pp. 1026-1030.

[7] J. Chen, A. Dholakia, E. Eleftheriou and M. P. Fossorier, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. Commun*, vol. 53, no. 8, pp. 1288-1299, Aug. 2005.

[8] D. E. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," in *Proc. SIPS*, 2004, pp. 107-112.

[9] M. M. Mansour and N. R. Shanbhag, "A 640-Mb/s 2048-bit programmable LDPC decoder chip," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 684–698, Mar. 2006

[10] G.hn, [Online] G.9960 "Unified high-speed wire-line based home networking transceivers-System architecture and physical layer specification", Available: http://www.itu.int/itu-t/recommendations/index.aspx?ser=G.

[11] D. Bao, B. Xiang, R. Shen, A. Pan, Y. Chen and X. Y. Zeng, "Programmable Architecture for Flexi-Mode QC-LDPC Decoder Supporting Wireless LAN/MAN Applications and Beyond," *IEEE Trans. Circuits and Systems I: Regular Paper*, vol. 57, no. 1, pp. 125-138, 2010.

[12] B. Xiang *et al.*, "An area-efficient and low-power multirate decoder for quasi-cyclic low-density parity-check codes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 10, pp. 1447–1460, Oct. 2010.