A SHUFFLED-BASED ITERATIVE DEMODULATION AND DECODING SCHEME FOR LDPC CODED FLASH MEMORY

Li-Chung Lee^{*}, Wei-Min Lai^{*}, Mao-Ruei Li^{*}, and Yeong-Luh Ueng[†]

*Dept. of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, R.O.C. [†]Dept. of Electrical Engineering and Inst. of Commun. Engineering, National Tsing Hua University, Hsinchu, Taiwan, R.O.C.

ABSTRACT

In previous studies, a very sparse low-density parity-check (LDPC) code was designed for triple-level cell (TLC) NAND flash using non-Gray mapping, which is able to achieve comparable error-rate performance to the conventional Gray mapping-based scheme. Although a sparse LDPC code can be of benefit to hardware implementations of an iterative demodulation and decoding (IDD) scheme, difficulties emerge, such as latency issue between the decoder and demodulator, when compared to non-IDD schemes. In this paper, a hardware-friendly structure interleaver is used such that a shuffle-based IDD scheme can be realized efficiently. Compared to the conventional Gray-based non-IDD scheme and layered-based IDD scheme, the proposed shuffled-based IDD scheme can provide a better hardware efficiency and better error-rate performance.

Index Terms—LDPC codes, coded modulation, iterative demodulation and decoding (IDD), Flash memory.

I. INTRODUCTION

With the progress of manufacturing processes, NAND flash memory that was originally constructed using a single-level cell (SLC) is now constructed using a multi-level cell (MLC) or a triple-level cell (TLC). Although more bits can be stored in a single cell, the reliability of the stored data decreases as the density of the flash memory increases, resulting in the frequent occurrence of errors. Conventional error correcting coding (ECC) schemes, such as BCH code, can no longer provide a satisfactory error-correcting capability; thus, low-density parity-check (LDPC) code is now considered as a prominent candidate for such schemes [1], [2], [3].

Generally speaking, SLC, MLC and TLC components can be modeled as 2-PAM, 4-PAM and 8-PAM modulation systems with Gray mapping, respectively, as shown in Fig. 1. If an iterative detection and decoding (IDD) scheme can be used, the mapping for 8-PAM (TLC) can be optimized using



Fig. 1. Flash memory cells are modeled as PAM modulation schemes, where the Gray mapping rules are used.

the criterion presented in [4]. The resultant mapping is non-Gray. In [5] and [6], the outer LDPC code was optimized for this non-Gray 8-PAM, and the density of the resultant LDPC code was reduced by one third while maintaining error-rate performances comparable to the conventional LDPC coded 8-PAM using Gray mapping.

Although a sparse parity-check matrix (PCM) brings advantages to hardware implementations of an LDPC decoder, some difficulties still exist for the IDD scheme in order to achieve throughput and area efficiency comparable to the conventional non-IDD scheme. To overcome this difficulty, the authors of [7] considered a layered-based IDD scheme. For this layered architecture, it is a challenge for the demodulator and the decoder to seamlessly process the same codeword. Consequently, both throughput performance and hardware efficiency are limited. As a result, a two-codeword processing scheme was used in [7] so as to overcome these limitations. However, the size of the memory required for storing both the channel messages (L_c) and the extrinsic messages (L_e) was doubled.

In this paper, a shuffle-based IDD scheme is proposed. Based on the vertical scheduling feature of the shuffle-based decoder, a hardware-friendly structure interleaver is used such that the one-codeword processing can be realized using the shuffle-based IDD scheme efficiently. In addition, the interface between the decoder and demodulator is optimized, thus, the memory usage for L_e and L_c is saved 53.5% comparing to that of [7]. Moreover, there is about a 0.5 dB performance gain when using the structure interleaver com-

This research was supported in part by Ministry of Science and Technology of the R.O.C. under grants MOST 103-2622-E-007-029 -CC2 and MOST 106-2221-E-007-018-MY3.

pared to the random interleaver. The proposed IDD scheme is realized using a 90nm CMOS process, and occupies a silicon area of 5.32 mm², which is equivalent to a gate count of 1888K, and provides a throughput of 1.55 Gbps at a clock frequency of 190 MHz. Compared to the conventional Graybased non-IDD scheme and the layered-based IDD scheme, the proposed shuffled-based IDD scheme can provide a better hardware efficiency and better error-rate performance.

II. OVERVIEW OF PREVIOUS CONTRIBUTIONS

II-A. LDPC coded 8-PAM scheme using the layeredbased IDD

Fig. 2(a) shows the mapping for 8-PAM (TLC) obtained using the criterion presented in [4]. In [5] and [6], the degree distribution of the outer LDPC code was optimized for this non-Gray 8-PAM using density evolution. The resultant degree distribution for the variable nodes, i.e., d_v , is shown in Fig. 2(b). The designed degree distribution for the non-Gray mapping has more lower degrees, e.g., 90% of degree 2, compared to the Gray mapping, which implies that the complexity of the resultant decoder is reduced. However, an IDD approach should be used for this non-Gray scheme.

Fig. 3 shows an IDD-based LDPC coded 8-PAM scheme, where the 8-PAM demodulator and an LDPC decoder iteratively exchange soft information using the log-likelihood ratio (LLR) value. Then, after several exchanges of LLR values between the decoder and the demodulator, the decoder generates a decoded codeword. Each exchange of LLR values in the LDPC decoder is denoted as the inner iteration, and the corresponding iteration number is superscripted using the format k_i . In contrast, the exchange between the decoder and the demodulator is denoted as the outer iteration, superscripted using the format k_o .

The following is a brief description of the procedure used for the outer iteration. For the k_o th outer iteration, the demodulator computes the channel LLR values, $L_c^{k_o}$, using both the noisy received symbol and the extrinsic LLR values, L_e , computed by the LDPC decoder. In [7] a layer-based IDD scheme was proposed. In addition, a two-codeword (CW) IDD schedule was used, as shown in Fig. 3(b), in order to increase the hardware efficiency and improve the throughput performance of the IDD receiver, where CW1 and CW2 are processed together in the k_o th iteration. However, to store all the required L_c (L_e) values, the size of the memory for L_c and L_e needs to be doubled.

II-B. Shuffled-based LDPC decoding using the normalized min-sum (NMS) algorithm

An LDPC code can be defined by an $M \times N$ PCM, where the PCM contains M check nodes (CNs) and Nvariable nodes (VNs). In the decoder, LLRs are iteratively exchanged between CNs and VNs, and the *a posteriori* probability (APP) LLR values, $\Lambda_0, \Lambda_1, ..., \Lambda_{N-1}$, are then produced once the exchange procedure ends. A hard decision



Fig. 2. (a) The mapping rule for 8-PAM obtained using the criterion presented in [4]; and (b) the variable node degree distributions of LDPC codes for both Gray mapping and non-Gray mapping.



Fig. 3. The approach proposed in [7]: (a) an IDD-based LDPC coded 8-PAM scheme, where Π and Π^{-1} denote the interleaver and deinterleaver, respectively; (b) the two-codeword schedule used for the IDD scheme.

is then performed on $\Lambda_n^{k_i}$, n = 0, 1, ..., N - 1, to yield a decoded codeword, where $\Lambda_n^{k_i}$ represents the APP value after k_i+1 inner iterations. In addition, $R_{m,n}$ and $Q_{m,n}$ denote the check-to-variable (C2V) message from the *m*th check node to the *n*th variable node and the variable-to-check (V2C) message from the *n*th variable node to the *m*th check node, respectively. Further, M(n) denotes the index set for all CNs neighboring the *n*th VN. Similarly, N(m) denotes the index set for all VNs neighboring the *m*th CN.

Suppose that the PCM is divided into G block columns. In the shuffled-based decoding, V2C and C2V messages are calculated and exchanged in sequential block columns. Let $N_L(m)$ be a subset of N(m) that contains the elements corresponding to block columns 0, 1, 2, ..., g - 1 and let $N_R(m) = N(m) \setminus N_L(m)$. The decoding operations for block column j, can be described using steps 1 to 4 below:

1) APP updating: Initially,
$$\Lambda_n^0 = L_{c,n}^{\kappa_o}$$
, then for $k_i > 0$:

$$\Lambda_n^{k_i} = L_{c,n}^{k_o} + \sum_{m' \in M(n)} R_{m',n}^{k_i - 1}.$$
 (1)

2) Hard decision and syndrome calculation: A hard deci-



Fig. 4. Architecture for the proposed shuffled-based IDD scheme.

sion for each VN is made based on the sign of the APP values and then the syndrome values are calculated.

3) V2C updating: For all CNs that link to VN $n, m \in N(n)$,

$$Q_{m,n}^{k_i} = \Lambda_n^{k_i} - R_{m,n}^{k_i-1}.$$
 (2)

4) C2V updating: For all $n \in N(m)$,

$$\begin{split} S_{m,n}^{k_i} &= \prod_{n' \in N_L(m)} sgn(Q_{m,n'}^{k_i}) \cdot \prod_{n' \in N_R(m) \setminus n} sgn(Q_{m,n'}^{k_i-1}) \\ R_{m,n}^{k_i} &= \alpha \cdot S_{m,n}^{k_i} \cdot min\{\min_{n' \in N_L(m)} |Q_{m,n'}^{k_i}| \\ &, \min_{n' \in N_R(m) \setminus n} |Q_{m,n'}^{k_i-1}|\}, \ 0 < \alpha < 1. \end{split}$$

The computations for block column 0 to block column G-1 are sequentially performed once in a single inner iteration.

III. THE PROPOSED SHUFFLED-BASED IDD SCHEME

III-A. Challenges of the shuffle-based IDD scheme

Fig. 4 shows the proposed IDD architecture. Due to the usage of the random interleaver, bits that are corresponding to a single 8-PAM symbol belong to different block columns of a PCM. Although the shuffle-based LDPC decoder is able to send the extrinsic value $L_{e,j}^{k_i}$ to the demodulator in real time, the demodulator still has to wait until all the L_e values of a codeword returned by the decoder to perform the demodulation procedure or calculate $L_c^{k_o}$. In other words, there will be a hardware idle issue.

The other potential problem is the existence of a hardware conflict as shown in Fig. 5. This occurs for some PCMs, where the weight of the *j*th block column is larger than that of the (j + 1)th block column. In these circumstances, when the C2V recover and the APP adder are ready to output data from the (j + 1)th block column to the V2C calculator, the V2C calculator is still busy on processing the data from the *j*th block column. Therefore, it results in idle for both the C2V recover and the APP adder.

These problems can be solved by using the following techniques. First, we arrange the block columns based on a descending degree. Then, a random interleaver is replaced by a hardware-friendly structure interleaver. The following subsection details the design of the structure interleaver.



Fig. 5. Timing diagram of the shuffled-based LDPC decoder when the hardware conflict occurs.



Fig. 6. The hardware-friendly structure interleaver.

III-B. Proposed hardware-friendly structure interleaver

The structure interleaver is depicted in Fig. 6. In this figure, bits corresponding to a single 8-PAM symbol are distributed in three consecutive block columns, where $b_{j,z}$, z = 0, 2, ..., Z - 1, denotes the *z*th bit in block column *j*, and *Z* is the circulant size. For the proposed interleaver, bits $b_{j,z}$, $b_{j+1,z}$ and $b_{j+2,z}$ are mapped to symbol $S_{Z|\frac{j}{2}|+z}$.

Comparing to the conventional random interleaver, the proposed structure interleaver enables the decoder only to compute and send the L_e values of three consecutive blocks (e.g., Block 0, Block 1 and Block 2) to the demodulator, instead of the L_e values of all block columns. Then, the demodulator utilizes these L_e values to generate $L_c^{k_o}$ and return them back to the decoder. As shown in Fig. 7, after the decoder calculates the extrinsic values of block columns 0 to 2 ($L_{e,0-2}^{k_o}$), then it sends $L_{e,0-2}^{k_o}$ back to the demodulator. Next, while the demodulator busies on computing $L_{e,0-2}^{k_o}$ using $L_{e,0-2}^{k_o}$, the decoder calculates the L_e values of next three block columns ($L_{e,3-5}^{k_o}$). As a result, the shuffle-based IDD scheme is able to realize the one-codeword processing scheme with minimized hardware idle.

Next, for the shuffle-based IDD scheme, the interface between the decoder and the demodulator can further be optimized. When computing the L_c values of block columns



Fig. 7. The proposed scheduling of the decoder and the demodulator in the shuffle-based IDD scheme.



Fig. 8. The proposed demodulator and decoder interface.

Table I. Implementation results

	Conventional	Layer-based	Shuffle-based
	non-IDD	IDD [7]	IDD
Code	(18432, 16704)		
Technology	90nm		
Alogrithm	NMS		
Max. Iteration number	15		
Quantization (bits)	5	6	
Clock frequency (MHz)	166	166	190
Throughput (Mbps)	679.9	1100	1555
Gate count	1297K	1891K	1888K
Area (mm ²)	3.66	5.33	5.32
Hardware efficiency (Mbps/mm ²)	185.76	206.37	292.19

j to j + 2, the demodulator only requires the L_e values corresponding to these three columns. Thus, the L_e values of block columns 0 to j - 1 and block columns j + 3 to G - 1 do not need to be reserved, and hence, the memory for L_e of all the block columns can be shrunk and replaced with the L_e buffer for 3 block columns.

The decoder and the demodulator interface and the associated memory bank are detailed in Fig. 8. Each time the decoder obtains a block size of $Z L_c$ values from the memory bank, it computes $Z L_e$ values and places these L_e values into the buffer. After the extrinsic values of three consecutive block columns being computed and placed into the buffer, the demodulator extracts $3Z L_e$ values from the buffer and performs the demodulation procedure. Later on, the demodulator generates and places the $3Z L_c$ values into the memory bank for the decoder.

III-C. Simulation results

A (18396, 16644) rate-0.9 LDPC obtained from [5] is adopted in this paper. Fig. 9 shows the error rate performances for the shuffle-based IDD scheme with and without the structure interleaver, the layer-based IDD scheme presented in [7], and the conventional non-IDD scheme using the Gray mapping. It can be observed that the shuffle-based IDD scheme without the structure interleaver has a similar performance comparing to the layer-based IDD scheme. In addition, the performance of the shuffle-based IDD scheme. In addition, the performance of the shuffle-based IDD scheme with the structure interleaver is improved by 0.5 dB. The implementation results are shown in Table I, where a 90 nm 1P9M TSMC CMOS process, a clock frequency of 190 MHz are used, and the FO4 delay of the adopted 90 nm technology is 27.8 ps.

Fig. 10 shows the improvements in the L_c and L_e memory usage. For the two-codeword layer-based scheme adopted in



Fig. 9. Bit error rate (BER) performance for both the proposed non-Gray and the conventional Gray based schemes in the AWGN channel.



Fig. 10. Comparison of the L_c and L_e memory size.

[7], the gate count (memory bank area) is about 1028 K. While for the one-codeword shuffle-based scheme with the random interleaver, the gate count is about 825 K, which is about 19.7% reductions. Finally, for the one-codeword shuffle-based scheme with the structure interleaver and the optimized L_e memory (replaced with buffer), the result is 478 K gate counts and 53.5% reductions from the two-codeword scheme.

IV. CONCLUSION

In this paper, we have proposed a high-efficiency shufflebased IDD scheme for the TLC flash memory, where the structure interleaver is proposed such that the memory requirements for both the L_e and L_c memory can be reduced. In addition, we also have optimized the interface between the LDPC decoder and the 8-PAM demodulator. Therefore, the memory bank area is cut down by 53.5% and 40% increasing in throughput performance. The BER performances for the proposed scheme is about 0.5 dB better than that of the layerbased IDD scheme proposed in [7]. Finally, implementation results show that the hardware efficiency can achieve 292.19 Mbps/mm², which is 41.6% larger than that of [7].

V. REFERENCES

R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. IT-8, pp. 21-28, Jan. 1962.

- [2] J. Kim, J. Cho, and W. Sung, "A high-speed layered min-sum LDPC decoder for error correction of NAND Flash memories," *Midwest Symposium on Circuits and Systems*, vol. 1361, pp. 0–3, 2011.
- [3] M.-R. Li, H.-C. Chou, Y.-L. Ueng, and Y. Chen, "A low-complexity LDPC decoder for NAND flash applications," *IEEE International Symposium on Circuits* and Systems, pp. 213-216, 2014.
- [4] F. Schreckenbach, et al., "Optimization of symbol mappings for bit-interleaved coded modulation with iterative decoding," *IEEE Communications Letters*, pp. 593-595, 2013.
- [5] J.-H. Shy, "LDPC coded modulation and its applications to MLC flash memory," *NTHU Master Thesis*, 2013.
- [6] H.-C. Lee, J.-H. Shy, Y.-M. Chen, and Y.-L. Ueng, "LDPC coded modulation for TLC flash memory," 2017 IEEE Information Theory Workshop (ITW), Nov. 2017.
- [7] M.-R. Li, T.-Y. Kuan, H.-C. Lee, and Y.-L. Ueng, "An IDD receiver of LDPC coded modulation scheme for flash memory applications," *IEEE Circuits and Systems* (APPCCAS), pp.289-292, Oct.2016.
- [8] J. Xu, L. Chen, I. Djurdjevic, S. Lin and K. Abdel-Ghaffar, "Construction of regular and irregular LDPC codes: geometry decomposition and masking," *IEEE Transactions on Information Theory*, Vol. 53, No. 1, Jan. 2007.
- [9] H.-C. Lee, M.-R. Li, J.-K. Hu, P.-C. Chou, and Y.-L. Ueng, "Optimization techniques for the efficient implementation of high-rate layered QC-LDPC decoders," *IEEE Transactions on Circuits and Systems I-Regular Papers*, Vol. 64, No. 2, pp. 457-470, Feb. 2017.