Low-complexity Check Node Processing for Trellis Min-max Nonbinary LDPC Decoding

1st Huyen Pham Thi
National Laboratory of Information Security
Ha Noi, Viet Nam
phamhuyenmta87@gmail.com

2nd Hung Dao Tuan
National Laboratory of Information Security
Ha Noi, Viet Nam
daotuanhung@gmail.com

3rd Le Dinh Trang Dang
Le Quy Don Technical University
Ha Noi, Viet Nam
dldtrang@gmail.com

4th Hanho Lee
Dept. Information and Communication Engineering
Inha University
Incheon, Korea
hhlee@inha.ac.kr

5rd Tho Nguyen Huu
Le Quy Don Technical University
Ha Noi, Viet Nam
tho.nh@mta.edu.vn

***Abstract*—Nonbinary low-density-parity-check (NB-LDPC) code outperforms their binary counterpart in terms of error-correcting performance when the code length is moderate. Check node processing is a bottleneck of the NB-LDPC decoding. In this paper, a novel half-row modified two-extra-column trellis min-max (HR-mTEC-TMM) algorithm is proposed for the check node processing to reduce not only the complexity but also the storage memory. The check node unit (CNU) architecture corresponding to the proposed algorithm is designed for the (837, 726) NB-LDPC code over GF(32). The implementation results using 90-nm CMOS technology show that the proposed CNU architecture obtains a reduction of 28.3% for the area and 43.87% for the storage memory with an acceptable error-correcting performance loss, compared to existing work.**

***Index Terms*—NB-LDPC, Trellis min-max, check node processing, layered decoding, VLSI design**

I. Introduction

Nonbinary low-density parity-check (NB-LDPC) codes, which are defined over Galois Fields GF(q) ($q > 2$), outperform their binary counterpart in terms of error-correcting performance, burst error correction capability, and performance improvement in the error-floor region when the code length is moderate [1]. Nonetheless, the NB-LDPC decoding algorithms require complex computations, and their architectures have very high complexity and large memory requirements. In practical implementations, the main bottleneck of the NB-LDPC decoder architecture is a check node unit (CNU) [2].

For practical purposes, suboptimal decoding algorithms such as the extended min-sum (EMS) algorithm [3] and the min-max algorithm [4] have been introduced to reduce the decoding complexity. The min-max algorithm is interesting in the hardware design because of its less complexity, compared to the min-sum algorithm. Nevertheless, the sequential computations of forward-backward scheme in [4] cause a throughput problem and the high memory requirement to store the intermediate messages. Recently, a forward-backward four-way merger min-max decoding algorithm [5] has been proposed for NB-LDPC codes to greatly reduce the decoding time and obtain the highest throughput for the forward-backward scheme.

In [6], the relaxed min-max algorithm has been introduced using trellis representation to eliminate computing the forward-backward messages. Nonetheless, a lot of clock cycles are required for the check node processing that limit the maximum throughput of the decoder. To solve the throughput problem, the trellis EMS algorithm [7] has been introduced to generate the check node messages in a parallel way by inserting an extra column to the original trellis. To take advantage of the idea in [7], a simplified trellis min-max (STMM) algorithm [8] is proposed for NB-LDPC decoding with less complexity, compared to [7]. However, the practical implementation for NB-LDPC decoder has some drawbacks such as high CNU complexity, large storage requirement, and routing congestion that reduce the overall efficiency of the decoder. Recently, some modified trellis min-max decoding algorithms [9]–[11] have been proposed to further reduce the CNU complexity and the storage elements for the TMM algorithm.

In this paper, a novel half-row modified two-extra-column TMM (HR-mTEC-TMM) algorithm is proposed for the check node processing in the NB-LDPC decoding where a half of row is processed in each time. Thus, the CNU complexity is greatly reduced in not only the complexity but also the storage memory at the cost of some error-correcting performance loss, compared to the decoding algorithm processing the whole row. The CNU architecture of a (837, 726) NB-LDPC code over GF(32) was implemented using the HR-mTEC-TMM algorithm to demonstrate the efficiency of the proposal.

The rest of the paper is constructed as follows. In Section II, a review of the NB-LDPC decoding algorithm is given. The proposed low-complexity check node processing is introduced in Section III. Section IV presents the proposed CNU architecture. The implementation results and comparison with previous works are discussed in Section V. Finally, conclusions are drawn in Section VI.

978-1-5386-6542-8/18/$31.00 ©2018 IEEE

Algorithm 1 Layered Min-Max Decoding Algorithm [11]

```
Input: $L_n(a)$= ln(Pr($c_n = z_n|channel$)/Pr($c_n = a|channel$))
       $Q_n^{1,0}(a) = L_n(a)$; $R_{mn}^0(a) = 0; k = 1$
1: while $k \leq I_{max}$ do
2:     for $l = 1$ to $M$ do
3:         $\tilde{Q}_{mn}^{k,l}(a) = Q_n^{k,l-1}(h_{mn}a) - R_{mn}^{k-1,l}(a)$
4:         $\tilde{Q}_{mn}^{k,l} = \min_{a \in GF(q)}(\tilde{Q}_{mn}^{k,l}(a))$
5:         $Q_{mn}^{k,l}(a) = \tilde{Q}_{mn}^{k,l}(a) - \tilde{Q}_{mn}^{k,l}$
6:         $R_{mn}^{k,l}(a) = \Theta\{Q_{mn}^{k,l}(a) \in N(m)\}$
7:         $Q_n^{k,l}(h_{mn}^{-1}a) = Q_{mn}^{k,l}(a) + R_{mn}^{k,l}(a)$
8:     end for
9: end while
Output: $\tilde{c}_n = \arg\min(Q_n^{k,l}(a))$
```

Algorithm 2 Half-row Modified TEC-TMM Algorithm

```
Input: $\mathbf{Q_{mn}}, z_n = \arg\min_{a \in GF(q)} Q_{mn}(a); \forall n \in N(m)/2$
1: $\beta = \sum_{j=0}^{\lceil d_c/2 \rceil - 1} z_j \in GF(q)$
2: $\Delta Q_{mj}(\eta = a \oplus z_j) = Q_{mj}(a)$; $(0 \leq j < \lceil d_c/2 \rceil)$
3: $\{m1_{1fin}(a), I_{1fin}(a)\} = \varphi\{\Delta Q_{mk}(a)|_{k=0}^{\lceil d_c/2 \rceil - 1}\}$
4: $\{\Delta Q'(a), d1(a), d2(a)\}$=
   $\min_{\eta'_k(a) \in conf(1,2)}\{\max_{k=1,2}(m1_{1fin}(\eta'_k(a)))\}$;
   $\Delta Q''(a) = \min 2_{\eta'_k(a) \in conf(1,2)}\{\max_{k=1,2}(m1_{1fin}(\eta'_k(a)))\}$
5: $\{\Delta Q'_{m1}, \Delta Q'_{m2}, a_{m1}, a_{m2}\} = \Psi\{\Delta Q'(a)\}$
   $\{\Delta Q''_{m1}, \Delta Q''_{m2}\} = \Phi\{\Delta Q''(a), a_{m1}, a_{m2}\}$
Output: $\begin{cases} P = \{\Delta Q'_{m1}, \Delta Q'_{m2}, \Delta Q''_{m1}, \Delta Q''_{m2}, a_{m1}, a_{m2}\} \\ d1(a), d2(a) \\ z_n^* = z_n \oplus \beta \end{cases}$
```

II. Review of NB-LDPC Decoding Algorithm

A. NB-LDPC codes

NB-LDPC codes, which are a kind of linear block code, are defined by a sparse parity-check matrix **H** having *M* rows and *N* columns. Let h_{mn} be a nonzero element of the matrix **H** that belongs to the GF($q = 2^p$). Furthermore, NB-LDPC codes are presented by a Tanner graph corresponding to the matrix **H**, where N columns correspond to N variable nodes, and M rows correspond to M check nodes. Let d_v and d_c be the variable node degree (column weight) and the check node degree (row weight) of matrix **H**, respectively. A regular NB-LDPC code is considered in this paper with the fixed values of d_c and d_v.

B. NB-LDPC Decoding Algorihm

A horizontal layered decoding algorithm, which is well known for higher convergence speed compared to the flood decoding algorithm, is applied in this paper. Algorithm 1 presents the layered decoding algorithm for the NB-LDPC codes. Symbols c_n and z_n define the n-th reference symbol of a received codeword and the n-th hard-decision symbol with the highest reliability, respectively. Starting the decoding process is implemented by obtaining the log-likelihood ratio (LLR) vectors $L_n(a)$ with a size of q that are the channel information.

At the first layer of the first iteration, the *a posteriori* information as $Q_n(a)$ corresponding the variable node n is equal to $L_n(a)$. The check node to variable node (C2V) messages $R_{mn}(a)$ are equal to zero. k and l define the loop index for k-th iteration and the layer index for l-th layer, respectively. Then, the variable node to check node (V2C) messages $\tilde{Q}_{mn}(a)$ are calculated from the $Q_n(a)$ messages permuted using the nonzero element h_{mn} of matrix **H**, as shown in step 3. The normalization of V2C messages are performed as shown in step 4 and 5 to stabilize the numerical growth that the LLR value of the symbol with highest possibility in each vector is equal to zero. Step 6 presents the computation of the C2V messages $R_{mn}(a)$ using the Θ function that relates to the HR-mTEC-TMM algorithm applied for the check node processing. Step 7 involves in the computation of the updated messages $Q_n(a)$, which is undergone the reverse permutation before processing a new layer. The decoding process is repeatedly implemented until the maximum number of iteration I_{max} is reached. Finally, the output codeword $\tilde{c}_n(a)$ is the most reliable symbol corresponding to $Q_n(a)$ message.

III. Low-complexity Check Node Processing

In this section, the novel HR-mTEC-TMM algorithm is proposed to greatly reduce the decoder complexity by a half, compared to previous works processing full row. In addition, the memory requirement as well as the exchanged messages between variable and check node are significantly reduced. The decoder corresponding to the HR-mTEC-TMM algorithm achieves a good trade of between the area efficiency and the error-correcting performance. The HR-mTEC-TMM algorithm is especially efficient for designing the decoders with high coderate.

The HR-mTEC-TMM algorithm for the first half row is represented in Algorithm 2, where a half of V2C messages are processed each time. Steps 1 to 4 in the HR-mTEC-TMM algorithm are similar as the two-extra-column trellis min-max (TEC-TMM) algorithm in [11]. $m1_{1fin}(a)$ and $I_{1fin}(a)$ are the minimum values and their indexes of the first half row, which are calculated by the φ function using the V2C input messages. Then, the two extra columns as $\Delta Q'(a)$ and $\Delta Q''(a)$ are constructed as shown in step 4. A modification is implemented on the two extra columns in step 5, where only the first two smallest values and their field elements as $\{\Delta Q'_{m1}, \Delta Q'_{m2}, a_{m1}, a_{m2}\}$ in the first extra column $\Delta Q'(a)$ are found using the Ψ function, and are kept for updating the C2V messages. Then, the two values in the second extra column such as $\Delta Q''_{m1}$ and $\Delta Q''_{m2}$ corresponding to the field elements $\{a_{m1}, a_{m2}\}$ are derived from the second extra column $\Delta Q''(a)$ using the Φ function. It is remarked that only four LLR values as $\{\Delta Q'_{m1}, \Delta Q'_{m2}, \Delta Q''_{m1}, \Delta Q''_{m2}\}$ are used for updating the C2V messages $R_{mn}(a)$ instead of $2 \times (q-1)$ values in the two extra columns. Let the LLR values be quantized with w bits. Output bits of the HR-mTEC-TMM algorithm $4 \times w + 2 \times p + 2 \times (q-1) \times \lceil log_2(d_c/2) \rceil + p \times \lceil d_c/2 \rceil$

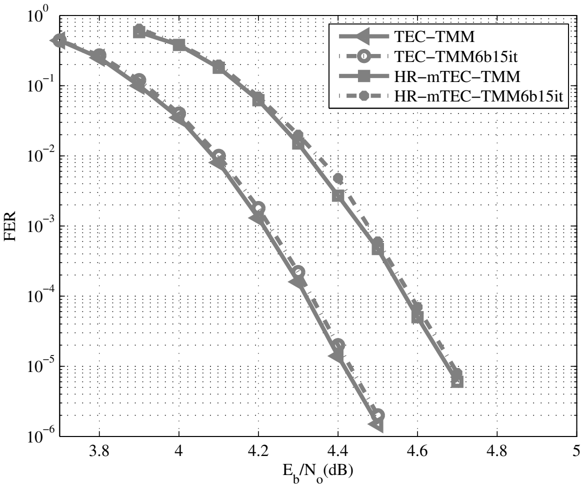

Fig. 1. FER performance of the (837, 726) NB-LDPC code over GF(32) under the AWGN channel at 15 iterations.

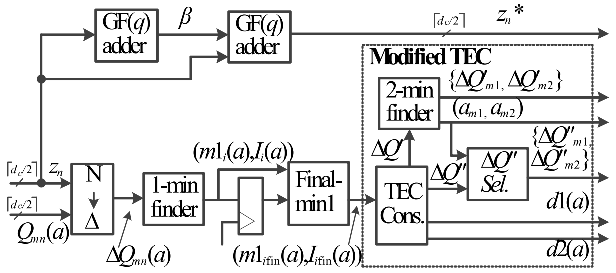

Fig. 2. Top-level CNU architecture for HR-mTEC-TMM algorithm.

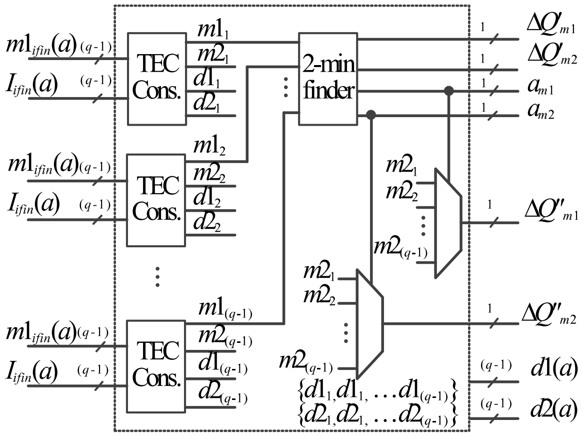

Fig. 3. Modified two-extra-column constructor.

are stored in the memory and exchanged between the check node and the variable node.

The process of the second half row is similar as the first half row. Only the difference is that the minimum values and their indexes, which are used to construct two extra columns, are those of full row. Let $m1_2(a)$ and $I_2(a)$ be the minimum values and their indexes in the second half row, which are calculated as shown in (1). Then, the final values such as $m1_{2fin}(a)$ and $I_{2fin}(a)$, which are used to construct the two extra columns in the second half row, are calculated as shown in (2).

$$\{m1_2(a), I_2(a)\} = \varphi\{\Delta Q_{mk}(a)|_{k=\lceil d_c/2\rceil}^{d_c-1}\} \tag{1}$$

$$\begin{aligned}\{m1_{2fin}(a), I_{2fin}(a)\} = \{&min(m1_{1fin}(a), m1_2(a)),\\ &I_{1fin}(a), I_2(a)\}\end{aligned} \tag{2}$$

Fig. 1 shows the frame error rate (FER) performance of the (837, 726) NB-LDPC code over GF(32) with $(d_c, d_v) = (27, 4)$ under the additive white Gaussian noise (AWGN) channel and binary phase shift keying modulation. It is obvious that the FER performance degradation of the HR-mTEC-TMM is almost 0.2 dB, compared to original TEC-TMM algorithm [11]. This performance loss is caused by the approximate calculation in the first half row using only a half of V2C messages. However, the complexity of the CNU architecture is greatly reduced since only a half of V2C messages is processed. In addition, the fixed-point simulation using $w_b = 6$-bit quantization depicts a negligible performance loss, compared to the floating-point simulation.

IV. Low-complexity Check Node Unit Architecture

Fig. 2 shows the top-level CNU architecture for the HR-mTEC-TMM algorithm, where each module is related to a step in the Algorithm 2. It is obvious that the number of V2C messages derived the CNU module is $\lceil d_c/2\rceil$, which is equal to a half of the full row processing. Thus, the complexity of modules such as normal to delta transformation and 1-min finder is reduced by almost a half. First, the transformation module is used to convert the V2C messages $Q_{mn}(a)$ from normal to delta domain using the control signals z_j. The transformation module includes $\lceil d_c/2\rceil$ reordering networks, as shown in [2], where each reordering network requires $q \times log_2 q$ w_b-bit multiplexers. The check node syndrome β is generated by a tree adder structure. The delta-to-normal domain transformation is performed in the variable node processing using $\lceil d_c/2\rceil$ reordering networks with the control signals $z_j^* = z_j \oplus \beta$. The φ function, which is responsible for finding the minimum value and its index from $\lceil d_c/2\rceil$ inputs, is implemented by the 1-min finder. $(q-1)$ 1-min finder modules are required because $(q-1)$ rows of the delta trellis except the first row have to perform the φ function. The 1-min finder architecture is designed as shown in [11]. It is remarked that for the first half row processing, outputs of the 1-min finder not only derive the final minimum module for the next processing but also store in registers for the second half row processing. Outputs of the Final-min1 in the first half row processing are the minimum values and indexes of the first half V2C messages. However, the ones in the second half row processing are the minimum values and indexes of the whole V2C messages in a row.

Fig. 3 shows the modified two-extra-column constructor

TABLE I
THE NUMBER OF STORAGE BITS IN THE CHECK NODE PROCESSING

GF(32) $d_c = 27$	**Exchanged bits** ($w = w_b - 1$ bits)	**Memory**
[12] TMM	$3 \times (q-1) \times w + 2 \times (q-1) \times \lceil log(d_c) \rceil + d_c \times p$	910
[11] TEC-TMM	$2 \times (q-1) \times (w + \lceil log(d_c) \rceil) + d_c \times p$	755
[9] mT-MM	$2 \times (q-1) \times \lceil log(d_c) \rceil + (q+1) \times w + (d_c+2) \times p$	620
Proposed	$4 \times w + 2 \times p + 2 \times (q-1) \times \lceil log(d_c) \rceil + p \times \lceil d_c/2 \rceil$	348

TABLE II
COMPARISON OF COMPLEXITY BETWEEN PROPOSED CNU AND PREVIOUS WORKS

GF(32) $d_c = 27$	**Gate count** (NAND)	**Quantization** ($w_b = 6$ bits)	**Reduction**
[8] STMM	222,000	6	50%
[12] TMM	154,806	6	28.3%
[10] TMM	133,273	6	16.8%
[11] TEC-TMM	122,500	6	9.5%
Proposed	110,875	6	

in detail. $(q-1)$ two-extra-column constructors, which are similar as the one in [11], are used to calculate the LLR values for $(q-1)$ non-zero field elements in two extra columns $\Delta Q'(a)$ and $\Delta Q''(a)$ based on the final minimum values and their indexes. In the HR-mTEC-TMM algorithm, only the first two minimum values and their field elements are required to calculate. Thus, $(q-1)$ LLR values of the first extra column $\Delta Q'(a)$ are given to the 2-min finder to find the first two minimum values $\{\Delta Q'_{m1}, \Delta Q'_{m2}\}$ and their field elements $\{a_{m1}, a_{m2}\}$. After that, two LLR values in the second extra column $\Delta Q''(a)$ corresponding to these field elements are adopted. Finally, the indexes of the final minimum values $I_{ifin}(a)$ are used to obtain the deviations as $d_1(a)$ and $d_2(a)$.

V. COMPLEXITY AND COMPARISON

Table I shows the number of storage bits in the check node processing in the proposed algorithm and previous works for $d_c = 27$ and $w_b = 6$ bits over GF(32). It is obvious that the proposed algorithm greatly reduces the storage bits, compared to previous works since the proposed algorithm uses only four LLR values to generate the C2V messages. Compared to [12], the number of storage bits is 2.6 times lower. Compared to [9], [11], the proposed work reduced the number of storage bits by 43.87% and 53.9%, respectively.

Table II presents the synthesis results of the proposed CNU architecture for the (837, 726) NB-LDPC code over GF(32) using Synopsys design tools and a TSMC 90-nm CMOS standard cell library. Compared to recent works in [10], [12], the gate count of the proposed CNU is significantly reduced by 16.8% and 28.3%, respectively. This reduction is obtained because only a half of the V2C messages are required for the check node processing, and 1-min finders are used to find the minimum values instead of 2-min finders in [10], [12]. In [11], the original TEC-TMM algorithm was proposed to construct two extra columns based on only the minimum values. In the HR-mTEC-TMM algorithm, two extra columns are constructed similarly as the work in [11]. However, only a half of the V2C messages are required for the check node processing. As the result, the complexity of the proposed CNU is decreased by 9.5%, compared to [11].

VI. CONCLUSION

In this paper, a novel HR-mTEC-TMM algorithm is proposed for the check node processing to reduce both the complexity and the storage memory. The CNU architecture corresponding to the proposed algorithm is designed to demonstrate the efficiency of the proposal. The implementation results show a reduction of 28.3% for the area and 43.87% for the storage memory at the cost of the acceptable error-correcting performance loss.

ACKNOWLEDGMENT

This work was supported by National Laboratory of Information Security, Ha Noi, Viet Nam, and by WCSL (World Class Smart Lab) research grant directed by Inha University.

REFERENCES

[1] H. C. Davey and D. J. MacKay, "Low-density parity check codes over GF(q)," in *Information Theory Workshop*, Jun. 1998, pp. 165–167.

[2] J. Lin, J. Sha, Z. Wang, and L. Li, "Efficient decoder design for nonbinary quasicyclic LDPC codes," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 5, pp. 1071–1082, May. 2010.

[3] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF(q)," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633–643, Apr. 2007.

[4] V. Savin, "Min-max decoding for non binary LDPC codes," in *in Proc. IEEE Int. Symp Inf. Theory*, Toronto, ON, Canada, Jul. 2008, pp. 960–964.

[5] H. P. Thi, S. Ajaz, and H. Lee, "High-throughput partial-parallel block-layered decoding architecture for nonbinary LDPC codes," *Integration, the VLSI Journal*, vol. 59, pp. 52–63, Sep. 2017.

[6] F. Cai and X. Zhang, "Relaxed min-max decoder architectures for nonbinary low-density parity-check codes," *, IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 21, no. 11, pp. 2010–2023, Nov. 2013.

[7] E. Li, F. Garcia-Herrero, D. Declercq, K. Gunnam, J. O. Lacruz, and J. Valls, "Low latency T-EMS decoder for non-binary LDPC codes," in *in Proc. Conf. Rec. 47th Asiloma Conf. Signals, Syst. Comput. (ASILOMAR)*, Nov. 2013, pp. 831–835.

[8] J. O. Lacruz, F. García-Herrero, D. Declercq, and J. Valls, "Simplified trellis min-max decoder architecture for nonbinary low-density parity-check codes," *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 23, no. 9, pp. 1783–1792, Sep. 2015.

[9] J. O. Lacruz, F. García-Herrero, M. J. Canet, and J. Valls, "High-performance NB-LDPC decoder with reduction of message exchange," *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 24, no. 5, pp. 1950–1961, May. 2016.

[10] J. O. Lacruz, F. Garca-Herrero, M. J. Canet, and J. Valls, "Reduced-complexity nonbinary LDPC decoder for high-order galois fields based on trellis min–max algorithm," *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 24, no. 8, pp. 2643–2653, Aug. 2016.

[11] H. P. Thi and H. Lee, "Two-extra-column trellis min–max decoder architecture for nonbinary LDPC codes," *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 25, no. 5, pp. 1787–1791, May. 2017.

[12] J. O. Lacruz, F. García-Herrero, and J. Valls, "Reduction of complexity for nonbinary LDPC decoders with compressed messages," *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 23, no. 11, pp. 2676–2679, Nov. 2015.