International Journal of Advanced Information Science and Technology (IJAIST) *ISSN: 2319:268*
Vol.2, No.4, April 2013 *DOI:10.15693/ijaist/2013.v2i4.156-160*

An Efficient Non-binary LDPC decoder using simplified min-sum algorithm

R.Hariprasad[1], V.M.Ananth[2]
[1]Department of ECE, Karpagam University, Coimbatore
[2]Asst. Professor/ ECE, Karpagam University, Coimbatore

***Abstract* - A novel technology is used to design and implement the Non-binary iterative codes are robust to different channel losses. However, based on the existing EMS decoding algorithms, the decoder design and implementations are very expensive because of their excessive computational density and memory usage. Based on the Simplified Min-Sum algorithm (SMSA), we present a proposed method for the check node processing. The simulation results demonstrate and analysis the bit error rate (BER), block error rate (BLER) over the signal to noise ratio (SNR) in the channel. The proposed system design and implementation using cadence encounter simulator. Furthermore, the proposed reduced design complexity issues and its provide significant savings on hardware, so it produces a good performance to reduce the bit error rate.**

Keywords* - *Low-density parity-check (LDPC) codes, non-binary codes, iterative decoding, extended min-sum algorithm.

I.INTRODUCTION

Binary low-density parity-check (LDPC) codes, discovered by Gallager in 1962 [1], were rediscovered and shown to approach Shannon capacity in the late 1990s [2]. Since their rediscovery, a great deal of research has been conducted in the study of code construction methods decoding techniques, and performance analysis. With hardware-efficient decoding algorithms such as the min-sum algorithm [3], practical decoders can be implemented for effective error-control. Therefore, binary LDPC codes have been considered for a wide range of applications such as satellite broadcasting, wireless communications, optical communications, and high-density storage systems As the extension of the binary LDPC codes over the Galois field of order q, non-binary LDPC (NB-LDPC) codes, also known as q-ary LDPC codes, were first investigated by Davey and MacKay in 1998 [4].

They extended the sum-product algorithm (SPA) for binary LDPC codes to decode q-ary LDPC codes and referred to this extension as the q-ary SPA (QSPA). Based on the fast Fourier transform (FFT), they devised an equivalent realization called FFT-QSPA to reduce the computational complexity of QSPA for codes with q as a power of 2 [4]. With good construction methods [5]–[9], NB-LDPC codes decoded with the FFT-QSPA outperform Reed-Solomon codes decoded with the algebraic soft-decision Koetter-Vardy algorithm [10]. As a class of capacity approaching codes, NB-LDPC codes are capable of correcting symbol-wise errors and have recently been actively studied by numerous researchers. However, despite the excellent error performance of NB-LDPC codes, very little research contribution has been made for VLSI decoder implementations due to the lack of hardware-efficient decoding algorithms.

Even though the FFT-QSPA significantly reduces the number of computations for the QSPA, its complexity is still too high for practical applications, since it incorporates a great number of multiplications in probability domain for both check node (CN) and variable node (VN) processing. Thus logarithmic domain approaches were developed to approximate the QSPA, such as the extended min-sum algorithm (EMSA), which applies message truncation and sorting to further reduce complexity and memory requirements [11], [12]. The second widely used algorithm is the min-max algorithm (MMA) [13], which replaces the sum operations in the CN processing by max operations. With an optimal scaling or offset factor, the EMSA and MMA can cause less than 0.2 dB performance loss in terms of signal-to-noise ratio (SNR) compared to the QSPA.

However, implementing the EMSA and MMA still requires excessive silicon area, making the decoder considerably expensive for practical designs [14]–[17]. Besides the QSPA and its approximations, two reliability-based algorithms were proposed towards much lower complexity based on the concept of simple orthogonal check-sums used in the one-step majority-logic decoding [18]. Nevertheless, both algorithms incur at least 0.8 dB of SNR loss compared to the FFT-QSPA. Moreover, they are effective for decoding only when the parity-check matrix has a relatively large column weight. Consequently, the existing decoding algorithms are either too costly to implement or only applicable to limited code classes at cost of huge performance degradation. Therefore, we propose a reduced-complexity decoding algorithm, called the simplified min-sum algorithm (SMSA), which is derived from our analysis of the EMSA based on the combinatorial optimization. Compared to the QSPA, the SMSA shows small SNR loss, which is similar to that of the EMSA and MMA. Regarding the complexity of the

CN processing, the SMSA saves around 60% to 70% of computations compared to the EMSA. Also, the SMSA provides an exceptional saving of memory usage in the decoder design. According to our simulation results and complexity estimation, this decoding algorithm achieves a favorable tradeoff between error performance and implementation cost.

The rest of the paper is organized as follows. The NB-LDPC code and EMSA decoding are reviewed in Section II. The SMSA is derived and developed in Section III. The error performance simulation results are summarized in Section IV. In Section V, the SMSA is compared with the EMSA in terms of complexity and memory usage. At last, Section VI concludes this paper.

II. NB-LDPC CODES AND ITERATIVE DECODING

Let GF(q) denote a finite field of q elements with addition and multiplication. We will focus on the field with characteristic 2, i.e., $q = 2^p$. In such a field, each element has a binary representation, which is a vector of p bits and can be translated to a decimal number. Thus we label the elements in $GF(2^p)$ as $\{0,1,2,\ldots.2^p\text{-}1\}$. An (n,r) q-ary LDPC code C is given by the null space of an m x n sparse parity-check matrix $H = [h_{i,j}]$ over GF(q), with the dimension *r*.

The parity-check matrix H can be represented graphical by a Tanner graph, which is a bipartite graph with two disjoint variable node (VN) and check node (CN) classes. The j^{th} VN represents the j^{th} column of H, which is associated with the j^{th} symbol of the q-ary codeword The i^{th} CN represents its i^{th} row, i.e., the i^{th} q-ary parity check of H. The j^{th} VN and i^{th} CN are connected by an edge if $h_{i,j}$ 6 = 0. This implies that the j^{th} code symbol is checked by the i^{th} parity check. Thus for 0 <i < m and 0< j < n, we define $N_i = \{j : 0 \le j<n, h=i; j\ h_{ij} \neq 0\}$, and $M_j = \{i : 0 \le i<n, h_{ij} \le 0\}$. The size of N_i is referred to as the CN degree of the i^{th} CN, denoted as $|N_i.|$The size of M_j is referred to as the VN degree of the j^{th} VN, denoted as$|M_j|$. If both VN and CN degrees are invariable, letting dv = $|M_j|$. and dc=$|N_i.|$, such a codeis called a (dv, dc)-regular code. Otherwise it is an irregular code.

Similarly as binary LDPC codes, q-ary LDPC codes can be decoded iteratively by the message passing algorithm, in which messages are passed through the edges between the CNs and VNs. In the QSPA, EMSA, and MMA, a message is a vector composed of q sub-messages, or simply say, entries**.**Let $\Lambda_{j=}$ [Λ_j (0),Λ_j (1)…..$\Lambda_{j(q-1)}$]be the a priori information of the j -th code symbol from the channel. Assuming that Xj is the j^{-th} code symbol, the d^{-th} sub-message of Λ_j is a log-likelihood reliability (LLR) defined as Λ_j (d) = log(Prob(X_j=z_j)/Prob(X_j=d)). Z_j is the most likely (ML) symbol for , X_j i.e., z_j = argmax$_{d\in GF(q)}$ Prob(X_j = d), and z = $[z_j]_{j=1\ldots n}$ The smaller Λ_j (d) is, the more likely x_j = d is. Let $\alpha_{i,j}$ (d) and β_{ij} be the VN-to-CN (V2C) and CN-to-VN (C2V) soft messages between the i^{-th} CN and j^{-th} VN respectively. For all d€GF(q), the d-th entry of $\alpha_{i,j}$ (d), denoted as $\alpha_{i,j}$ (d), is the logarithmic reliability of d from the VN perspective is the symbol with the smallest reliability, i.e., the ML symbol of the V2C message. With $x_{i,j} = x_{i,j}* h_{i,j}$ we let $\alpha_{i,j}$ (d)= log(Prob($x_{i,j}$= $\alpha_{i,j}$)/Prob($x_{i,j}$ =d)) and $\alpha_{i,j}$ ($\alpha_{i,j}$) = 0.$b_{i,j}$ and $\beta_{i,j}$ (d) are defined from the CN perspective similarly.

III. EXTENDED MIN-SUM ALGORITHM

Initialization: Set z_j = arg min$_{d\in GF(q)}$ Λ_j (d). For all ij with $h_{i,j} \neq 0$, set $\alpha_{i,j}$ ($h_{i,j}$*d) = Λ_j (d). Set k = 0.

Step 1) Parity check: Compute the syndrome $z*H^T$ If $z*H^T$ = 0, stop decoding and output z as the decoded codeword; otherwise go to Step 2.

Step 2) If K =k_{max} , stop decoding and declare a decoding failure; otherwise, go to Step 3.

Step 3) CN processing: Let the configurations £$_i$($x_{i,j}$=d) be the sequence $[\alpha_{i,j}]_{j\in Ni}$ such that $\sum_{j\in Ni} x_{i,j}$=d and x i;j = d. With a preset scaling factor 0 < c ≤ 1, compute the C2V messages by

$$\beta_{i,j}(d) = c.\min_{L_i(x_{i,j}=d)} \sum_{j^1\in N_i\setminus j} \alpha_{i,j^1}(x_{i,j^1}) \quad (1)$$

Step 4) VN processing: k←k + 1. Compute V2C messages in two steps. First compute the primitive messages by

$$\hat{\alpha}_{i,j}\left(h_{i,j}\otimes d\right) = \lambda_j(d) + \sum_{i^1\in M_j\setminus i} \beta_{i^1,j}(h_{i^1,j}\otimes d) \quad (2)$$

Step 5) Message normalization: Obtain V2C messages by normalizing with respect to the ML symbol

$$\alpha_{i,j} = \arg\min_{d\in GF(q)} \hat{\alpha}_{i,j}(d) \quad (3)$$

$$\alpha_{i,j}(d) = \hat{\alpha}_{i,j}(d) - \hat{\alpha}_{i,j}(\alpha_{i,j}(d)) \quad (4)$$

Step 6) Tentative Decisions:

$$\hat{\lambda}_j(d) = \lambda_j(d) + \sum_{\iota\in M_i} \beta_{i,j}(h_{i,j}\otimes d) \quad (5)$$

$$Z_j = \min_{d\in GF(q)} \hat{\lambda}_{i,j}(d) \quad (6)$$

IV.THE SIMPLIFIED MIN-SUM ALGORITHM

Initialization: Set z_j = arg min$_{d\varepsilon GF(q)}\lambda_j$(d). For all *i, j* with $h_{i,j} \neq 0$, set $a_{i,j} = h_{i,j}$ x z_j and $\bar{\alpha}_{ij}(h_{ij}\otimes\delta) = \lambda_j(\delta\oplus z_j)$ set k=0.

Step 1) and 2) (The same as Step 1 and 2 in the EMSA)
Step 3.1) Compute the C2V hard messages:

$$bi,j = \sum_{j' \varepsilon Ni \setminus j} \oplus a_{i,j'} \quad (7)$$

Step 3.2) Compute the step-1 soft messages:

$$\bar{\beta}_{i,j}^{(1)}(\delta) = \min_{j' \varepsilon Ni \setminus J} \bar{\alpha}_{i,j'}(\delta) \quad (8)$$

Step 3.3) Compute the step-2 soft messages by selecting the combination of k symbols

$$\tilde{\beta}_{i,j}^{(k)}(\delta) = \min_{\sum_{l=1}^{\oplus k} \delta_l = \delta} \sum_{l=1}^{k} \tilde{\beta}_{i,j}^{(1)}(\delta_l) \quad (9)$$

Step 3.4) Scaling and reordering:

with $0 < c \leq 1, \tilde{\beta}_{i,j}(\delta) \approx c.\beta^{n}{}_{i,j}(\delta)$ (10)

For $d \neq b_{i,j}, \beta_{i,j}(d) = \beta'_{i,j}(b_{i,j} \oplus d)$; otherwise $\beta_{i,j}(b_{i,j}) = 0$.
Step 4) (The same as Step 4 in the EMSA)
Step 5) Message normalization and recording:

$$a_{i,j} = \arg \min_{d \varepsilon GF(q)} \bar{\alpha}_{i,j}(d) \quad (11)$$

$$\alpha_{i,j}(d) = \bar{\alpha}_{i,j}(d) - \bar{\alpha}_{i,j}(a_{i,j}) \quad (12)$$

$$\bar{\alpha}_{i,j}(\delta) = \alpha_{i,j}(\delta \oplus a_{i,j}) \quad (13)$$

Step 6) (The same as the Step 6 in the EMSA)
Go to Step 1.
As a result, the soft message generation is conducted in two steps (Step 3.2 and 3.3). To compute C2V message $\bar{\beta}_{i,j}$, first in Step 3.2 we compute the minimal entry values $\min_j$, $\bar{\alpha}_{i,j}(\delta)$ over all j'ε Ni\j for each δ ε GF(q) \ 0. Then in Step 3.3, the minimal values are used to generate the approximation of $\bar{\beta}_{i,j}(\delta)$. Instead of the configurations of all d_c VNs in Ni, (20) optimizes over the combinations of k symbols chosen from the field. Comparing Theorem 3 to (19) and (20), we can find that by our approximation method, in the SMSA, the optimization is performed over the VN set and symbol combination set separately and thus has the advantage of a much smaller search space.

Algorithm3. Generate the look-up table for GF(q).
1: **for** δ' = 1….q-1 **do**
2: **for** δ'' = (δ' ⊕ 1)…q-1 **do**
3: δ = δ' ⊕ δ'';
4: D(δ).Add(δ', δ'');
5: **end**
6: end

V. SIMULATION RESULTS

In this section, we demonstrate the performance of the above proposed SMSA for decoding NB-LDPC codes. The existing algorithm EMSA is used for performance comparison. The SMSA includes the fixed point (SMSA) and floating point (SMSA) versions. The three codes over GF(2^4), GF(2^6), and GF(2^8) are considered. We show that the SMSA fixed point has very good performance for different finite fields and modulations. And the SMSA fixed point has small performance loss compared to the SMSA floating point over GF(2^4) and GF(2^5). We study the fixed-point realizations of SMSA and find that it is exceptionally suitable for hardware implementation. To investigate the effectiveness of the SMSA, we evaluate the block error performance of the (620,310) code over GF(2^5) taken from the parity-check matrix of the code is a 10 x 20 array of 31 x 31 circulant permutation matrices and zero matrices. The SMSA floating point shows its reliability with higher channel randomness. In this work deeply analysis the convergence speed of SMSA and show that it converges almost as fast as EMSA.

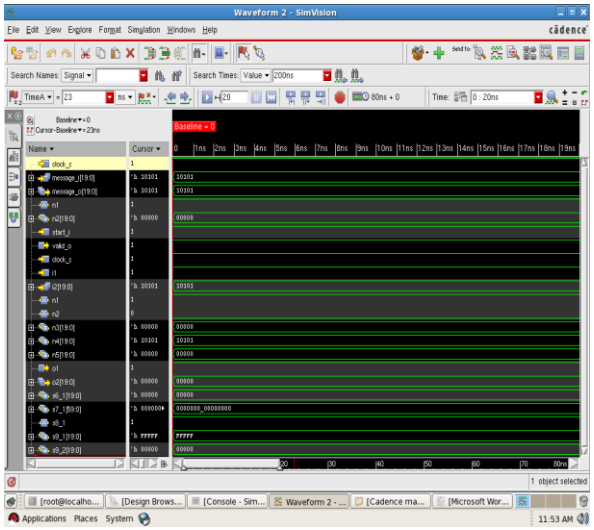

Fig 1. LDPC Decoding output

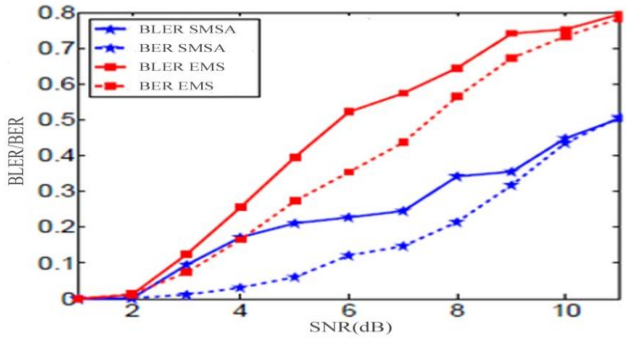

Fig 2. Comparison of EMS and SMSA with block error rate

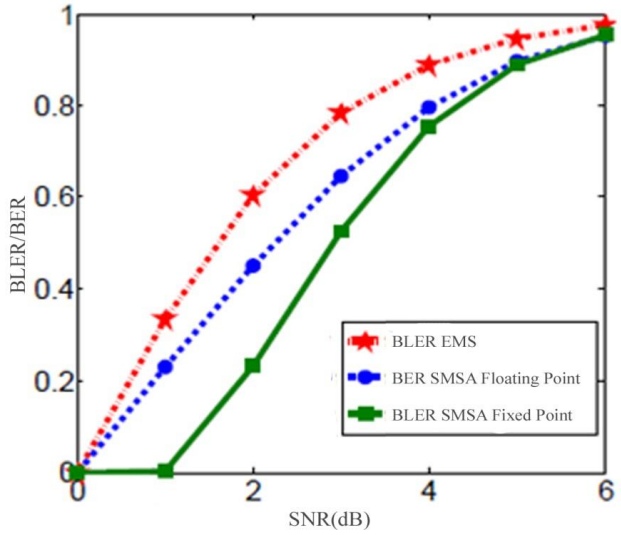

Fig 3. Comparison of floating point and fixed point with BLER

VI. CONCLUSIONS

In this paper, we have presented a hardware-efficient decoding algorithm, called the SMSA, This algorithm is devised based on significantly reducing the search space of combinatorial optimization. Two practical realizations, the Fixed point and Floating point SMSAs, are proposed for effective complexity-performance tradeoffs. Simulation results show that with field size up to 256, the fixed point SMSA has negligible error performance loss compared to the SMSA over the EMSA. The important feature of SMSA is simplicity. Based on our analysis, the SMSA has much lower computational complexity and memory usage compared to other decoding algorithms for NB-LDPC codes.

REFERENCES

[1] Weiguo Tang and Jie Huang "A Nonbinary LDPC Decoder Architecture with Adaptive Message Control" IEEE Transactions on vol. 20, no. 4, Nov 2012.

[2] R. Gallager., "Low-density parity-check codes," IEEE Transactions on Information Theory, vol. 8, no. 1, pp. 21–28, 1962.

[3] D. MacKay, "Good Error-Correcting Codes Based on Very Sparse Matrices," IEEE Transactions on Information Theory, vol. 45, no. 2, pp. 399–431, Mar 1999.

[4] M. Fossorier, M. Mihaljevic, and H. Imai, "Reduced Complexity Iterative Decoding of Low-Density Parity Check Codes Based on Belief Propagation," IEEE Transactions on Communications, vol. 47, no. 5, pp. 673–680, May 1999.

[5] M. Davey and D. MacKay, "Low-density parity check codes over GF(q)," IEEE Communications Letters, vol. 2, no. 6, pp. 165–167, 2002.

[6] A. Bennatan and D. Burshtein, "Design and analysis of nonbinary ldpc codes for arbitrary discrete-memoryless channels," IEEE Transactions on Information Theory, vol. 52, no. 2, pp. 549–583, 2006.

[7] S. Song, L. Zeng, S. Lin, and K. Abdel-Ghaffar, "Algebraic constructions of nonbinary quasi-cyclic ldpc codes," in IEEE International Symposium on Information Theory. IEEE, 2006, pp. 83–87.

[8] L. Zeng, L. Lan, Y. Tai, B. Zhou, S. Lin, and K. Abdel-Ghaffar, "Construction of nonbinary cyclic, quasi-cyclic and regular ldpc codes: a finite geometry approach," IEEE Transactions on Communications, vol. 56, no. 3, pp. 378–387, 2008.

[9] B. Zhou, J. Kang, Y. Tai, S. Lin, and Z. Ding, "High performance non-binary quasi-cyclic LDPC codes on Euclidean geometries LDPC codes on euclidean geometries," IEEE Transactions on Communications, vol. 57, no. 5, pp. 1298–1311, 2009.

[10] B. Zhou, J. Kang, S. Song, S. Lin, K. Abdel-Ghaffar, and M. Xu, "Construction of non-binary quasi-cyclic LDPC codes by arrays and array dispersions," IEEE Transactions on Communications, vol. 57, no. 6, pp. 1652–1662, 2009.

[11] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of reed-solomon codes," IEEE Transactions on Information Theory, vol. 49, no. 11, pp. 2809–2825, 2003.

[12] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF(q)," IEEE Transactions on Communications, vol. 55, no. 4, p. 633, 2007.

[13] A. Voicila, F. Verdier, D. Declercq, M. Fossorier, and P. Urard, "Architecture of a low-complexity non-binary LDPC decoder for high order fields," in IEEE International Symposium on Communications and Information Technologies. IEEE, 2007, pp. 1201–1206.

[14] V. Savin, "Min-Max decoding for non binary LDPC codes," in IEEE International Symposium on Information Theory, 2008, pp. 960–964.

[15] J. Lin, J. Sha, Z. Wang, and L. Li, "Efficient decoder design for nonbinary quasicyclic LDPC codes," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57, no. 5, pp. 1071–1082, 2010.

[16] X. Zhang and F. Cai, "Efficient partial-parallel decoder architecture for quasi-cyclic non-binary LDPC codes," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 58, no. 2, pp. 402–414, Feb. 2011.

[17] X. Chen, S. Lin, and V. Akella, "Efficient configurable decoder architecture for non-binary quasi-cyclic LDPC codes," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 59, no. 1, pp. 188–197, Jan. 2012.

[18] C. Chen, Q. Huang, C. Chao, and S. Lin, "Two low-complexity reliability-based message-passing algorithms for decoding non-binary ldpc codes," IEEE Transactions on Communications, vol. 58, no. 11, pp. 3140–3147, 2010.

[19] X. Chen and C.-L. Wang, "High-throughput efficient non-binary ldpc decoder based on the simplified min-sum algorithm," IEEE Transactions on Circuits and Systems I: Regular Papers, in publish.