

Dr.T.Krishnamoorthy, J. Nonlinear Anal. Optim. Vol. 13(2) (2022), February 2022

Journal of Nonlinear Analysis and Optimization
Vol. 13(2) (2022), February 2022
<https://ph03.tci-thaijjo.org/>
ISSN : 1906-9685



J.Nonlinear Anal. Optim

Reliability-Based Enhancement IHRB Decoding Algorithm for Non-Binary LDPC With Low-Complexity

Dr.T.Krishnamoorthy
Associate Professor, Department of ECE
Sri Sai Institute of Technology and Science, Rayachoti
Email: krishnamoorthyphd@gmail.com

Abstract—Non-binary low-density parity-check (NB-LDPC) codes can achieve better error-correcting performance than their binary counterparts at the cost of higher decoding complexity when the codeword length is moderate. The recently developed iterative reliability-based majority-logic NB-LDPC decoding has better performance-complexity tradeoffs than previous algorithms. This paper first proposes enhancement schemes to the iterative hard reliability-based majority-logic decoding (IHRB-MLGD). Compared to the IHRB algorithm, our enhanced (E-)IHRB algorithm can achieve significant coding gain with small hardware overhead. Then low-complexity partial-parallel NB-LDPC decoder architectures are developed based on these two algorithms. Many existing NB-LDPC code construction methods lead to quasi-cyclic or cyclic codes. Both types of codes are considered in our design. Moreover, novel schemes are developed to keep a small proportion of messages in order to reduce the memory requirement without causing noticeable performance loss. In addition, a shift-message structure is proposed by using memories concatenated with variable node units to enable efficient partial-parallel decoding for cyclic NB-LDPC codes. Compared to previous designs based on the Min-max decoding algorithm, our proposed decoders have at least tens of times lower complexity with moderate coding gain loss.

Index Terms—Iterative majority-logic decoding, low-density parity-check (LDPC) codes, non-binary, partial-parallel, VLSI.

for the WiMax standard needs less than 2500 slices on a device of the same family to achieve 28 Mbps.

Recently, two algorithms were developed for decoding NB-LDPC codes: iterative hard reliability-based majority-logic decoding (IHRB-MLGD) and iterative soft reliability-based majority-logic decoding (ISRB-MLGD). In these algorithms, reliability messages are incorporated into majority-logic decoding and improved through an iterative process. Unlike previous BP-based algorithms, these two algorithms require only simple check sum computations over $\mathbb{F}_2$ in the check node processing. Moreover, only one set of reliability messages need to be stored for the received symbols and the messages passed from a variable node to all connected check nodes are the same. Hence, the memory required for storing messages can be greatly reduced. As a result, these iterative reliability-based majority-logic decoding algorithms can achieve effective complexity-performance tradeoff. Compared to the ISRB algorithm, the IHRB algorithm updates reliability messages based on the hard decisions instead of probabilities of the received symbols. Hence, at the cost of moderate coding gain loss, the IHRB algorithm has much lower computation complexity and memory requirement than the ISRB algorithm. Nevertheless, mapping the IHRB algorithm directly to hardware implementation still leads to high complexity.

This paper first proposes an enhanced (E-)IHRB algorithm. Through incorporating the probability information from the channel into the message initialization of the IHRB algorithm and excluding the contribution of the same check node from the variable-to-check (v-to-c) message, the E-IHRB algorithm can bridge the performance gap between the IHRB and ISRB algorithms with small complexity overhead. Novel partial-parallel architectures are also developed in this paper to efficiently implement the IHRB and E-IHRB algorithms through algorithmic and architectural optimizations. Many construction methods of NB-LDPC codes lead to quasi-cyclic (QC) or cyclic codes. For QC codes, our IHRB decoder processes one row of sub-matrices at a time, and the variable node units (VNUs) are implemented by simple logic when all messages in a vector are kept. Cyclic NB-LDPC codes have the advantage that encoders can be implemented easily by linear feedback shift registers. However, these cyclic codes usually involve finite fields of high order, in which case keeping all messages leads to large memory requirement.

encoders can be implemented easily by linear feedback shift registers. However, these cyclic codes usually involve finite fields of high order, in which case keeping all messages leads to large memory requirement. Novel schemes are developed in this paper to store only a small proportion of the messages without incurring noticeable performance loss. In addition, a shift-message decoder architecture is proposed for cyclic NB-LDPC codes to enable efficient partial-parallel processing. The message shifting is accomplished through concatenating memories with VNUs to reduce the area requirement. It is non-trivial to extend the IHRB decoder architecture to implement the E-IHRB algorithm since recording the messages from check nodes may lead to large memory overhead, especially when the column weight of the code is not small. By making use of the properties of the IHRB algorithm, an innovative approach is proposed in this paper to reverse the contributions from check nodes through remembering only a few extra values for each variable node. Compared to the Min-max decoder architectures, which are the most efficient existing QCNB-LDPC decoder designs, the proposed E-IHRB decoder architecture has at least tens of times lower complexity with moderate coding gain loss.

The structure of this paper is as follows. Section II introduces the IHRB algorithm for NB-LDPC decoding. The proposed E-IHRB algorithm is detailed in Section III. Then the IHRB and E-IHRB decoder architectures are presented in Sections IV and V, respectively. After complexity analyses and comparisons are done in Section VI, conclusions are drawn in Section VII.

II. IHRB-MLGD ALGORITHM

An LDPC code is a linear block code that can be defined by the corresponding parity check matrix H or the associated Tanner graph. In the Tanner graph, a check (variable) node represents a row (column) of H , and the i th check node is connected to the j th variable node if the corresponding entry $h_{i,j}$ in H is nonzero. To simplify notations, this paper considers regular NB-LDPC codes, whose H matrix has constant row weight d_c and constant column weight d_v . To reduce the decoder hardware complexity, QC, and cyclic NB-LDPC codes can be constructed using the methods in [13]. The H matrix of a QCNB-LDPC code can be divided into sub-matrices that are zero or shifted identity matrices with nonzero entries replaced by elements of $GF(q)$. For a cyclic NB-LDPC code, the H matrix consists of a single circulant matrix whose entries are elements of $GF(q)$ or a column of circulant matrices.

Algorithm A: IHRB-MLGD algorithm

Initialization: $R_{j,l}^{(0)} = \gamma$ if $z_j^{(0)} = \alpha_l$; $R_{j,l}^{(0)} = 0$ if $z_j^{(0)} \neq \alpha_l$
for $k = 0 : I_{\max}$
A1: Stop if $z^{(k)} H^T = 0$
 for $i = 0$ to $m - 1$
 for $j \in N_i$
A2: $\sigma_{i,j} = h_{i,j}^{-1} \sum_{u \in N_i \setminus j} z_u^{(k)} h_{i,u}$
A3: if $(\sigma_{i,j} = \alpha_l)$
 $R_{j,l}^{(k)} = R_{j,l}^{(k)} + 1$
 $R_{j,l}^{(k+1)} = R_{j,l}^{(k)}$
 for $j = 0$ to $n - 1$
A4: $z_j^{(k+1)} = \text{field element of } \max_l R_{j,l}^{(k+1)}$

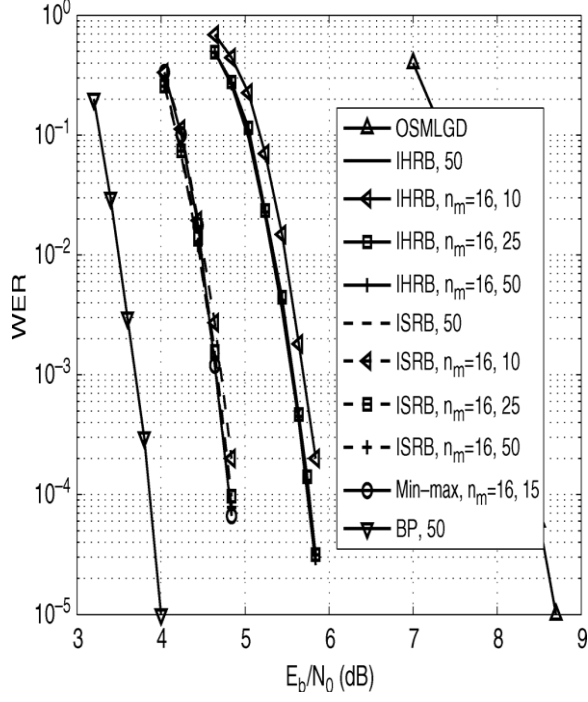


Fig. 1. WERs of NB-LDPC decoding for a (255, 175) cyclic EG code over

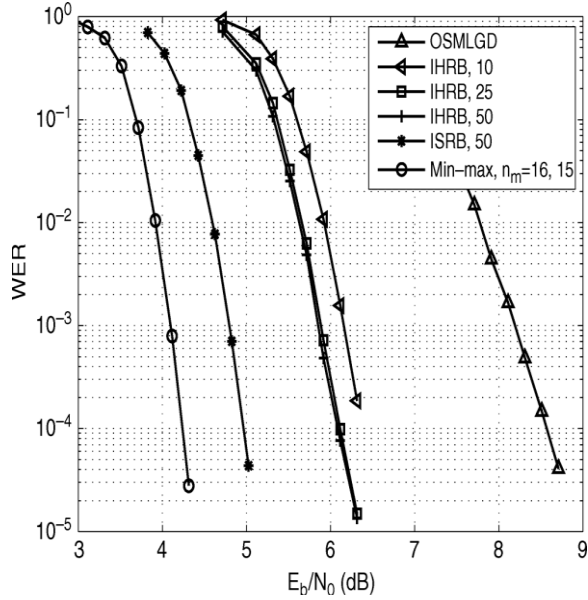


TABLE I
WORD LENGTHS OF RELIABILITY MEASURES USED IN SIMULATIONS

Algorithm	IHRB	ISRB	Min-max
Word length w (bits)	3	9	5

III. ENHANCED IHRB-MLGD ALGORITHM

One reason that the IHRB algorithm has performance loss compared to the ISRB algorithm is that the soft information is lost in the initialization of the reliability measures: the reliability measure for the hard-decision symbol is set to γ , while all other measures in the same vector are set to zero. Recording different initialization values does not cost extra memory, if the word length is not changed. Hence, the IHRB algorithm can be enhanced by initializing the reliability measures according to the probability information from the channel in a way similar to that in the ISRB algorithm. Nevertheless, to reduce the decoder hardware complexity, the maximum reliability measure in each vector should be set to a constant γ in the format of $2^m - 1$, as before, so that it does not need to be stored, and the clipping can be simplified as will be detailed in later sections. Such initialization can be done first as $\gamma \cdot p_i$, where p_i is a constant that has similar effect as the γ in the ISRB decoding. The optimum value of p_i can be derived from simulations, and is affected by the data format of reliability measures and dynamic range of the symbol reliabilities from the channel. Then, the reliability measures in each vector are offset by the same value so that the maximum measure becomes γ . In case some measures become negative after the offset, zero is used instead. Accordingly, the reliability measures will also remain as integers in the range of $[0, \gamma]$ during the decoding if this enhancement is adopted.

Another approach for initializing the reliability measures using soft information has been proposed in [10]. In this approach, each reliability measure is offset by the smallest measure in the vector, and then divided by the difference of the largest and smallest measures in the vector. Compared to this approach, our initialization scheme is more hardware friendly, since it needs constant multiplications instead of integer divisions. Moreover, in the case that the probabilities of different finite field elements for a received symbol are very close, the approach in [10] can increase the gaps of the reliability measures for different finite field elements. As a result, it may become harder for the decoding to converge to the second most reliable or other field element, even if it is the correct one. On the other hand, our initialization does not change the relative gaps between the reliability measures in a vector, and does not have this problem.

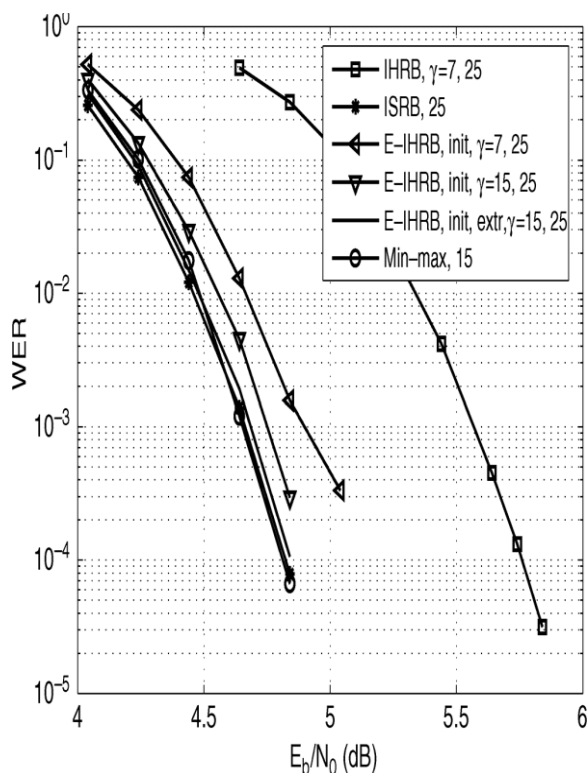
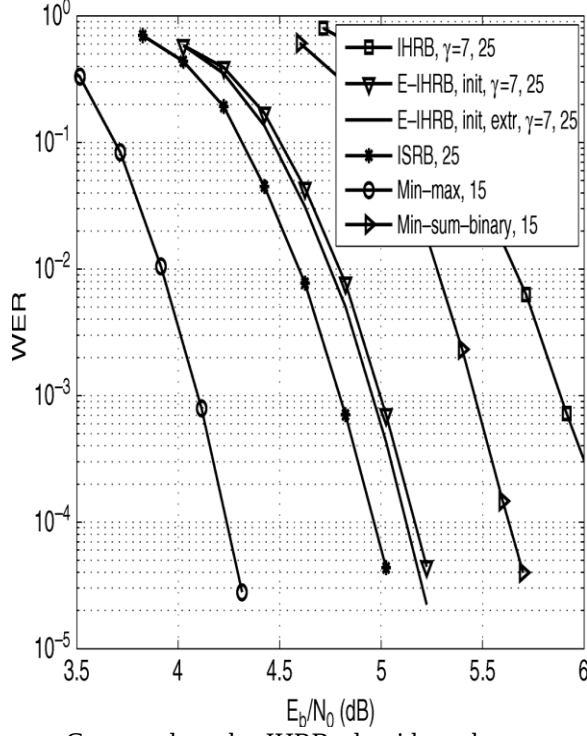


Fig. 2. WERs of NB-LDPC decoding for a (255,175) QC code over $GF(2^5)$



Compared to the IHRB algorithm, there are two major differences in the ISRB algorithm: soft reliability measure initialization and soft reliability measure updating. It should be noted that if the ISRB algorithm adopts soft initialization, but not soft updating, it reduces to an algorithm very similar to the E-IHRB algorithm with only soft initialization. Nevertheless, our proposed initialization leads to lower decoder hardware complexity since the maximum reliability measure in a vector is always a constant. Extra performance improvement is achieved in the ISRB algorithm by updating the reliability measures using soft information. However, as explained previously, this leads to significantly longer word length and accordingly larger memory and more complicated clipping logic. Alternatively, extra coding gain can be achieved by using extrinsic information as proposed in our second enhancement scheme to the IHRB decoder, which still updates the reliability measures using hard decisions. In summary, there are analogies between the soft initializations in the ISRB and proposed E-IHRB algorithms. However, extra coding gain over soft initialization can be achieved with much lower hardware complexity using the second enhancement scheme in our proposed E-IHRB algorithm.

Although the computations in the IHRB and E-IHRB algorithms are simpler than those in the ISRB and other soft NB-LDPC decoding algorithms, mapping them directly to hardware still leads to very high complexity, especially when the involved finite field has high order. Efficient partial-parallel architectures for these two decoding algorithms are developed next through algorithmic and architectural modifications for both QC and cyclic NB-LDPC codes.

IV. PARTIAL-PARALLEL ARCHITECTURES FOR IHRB-MLGD

Since the E-IHRB algorithm is based on the IHRB algorithm, this section presents architecture design for the IHRB algorithm first. Then the modifications needed to implement the E-IHRB algorithm are introduced in the next section.

Fully-parallel decoders require overwhelming complexity for NB-LDPC codes that are not very short. On the other hand, serial design cannot achieve fast decoding speed. Hence, this paper considers partial-parallel architectures that can achieve speed-area tradeoff. Many construction schemes for NB-LDPC codes in lead to either cyclic or QC codes. Compared to random codes, these codes enable more efficient partial-parallel processing due to the regularity in the H matrix. In this section, partial-parallel IHRB decoder architectures are developed for these two types of codes.

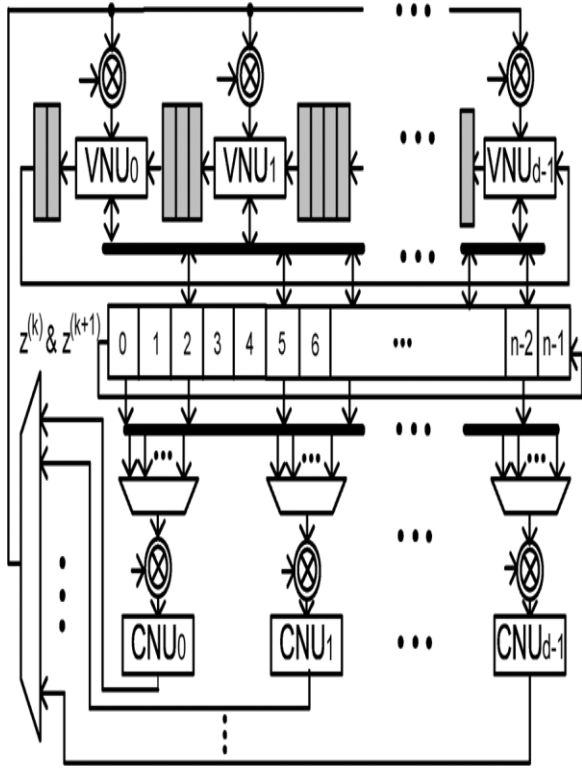


Fig. 6. IHRB-MLGD architecture for cyclic NB-LDPC codes

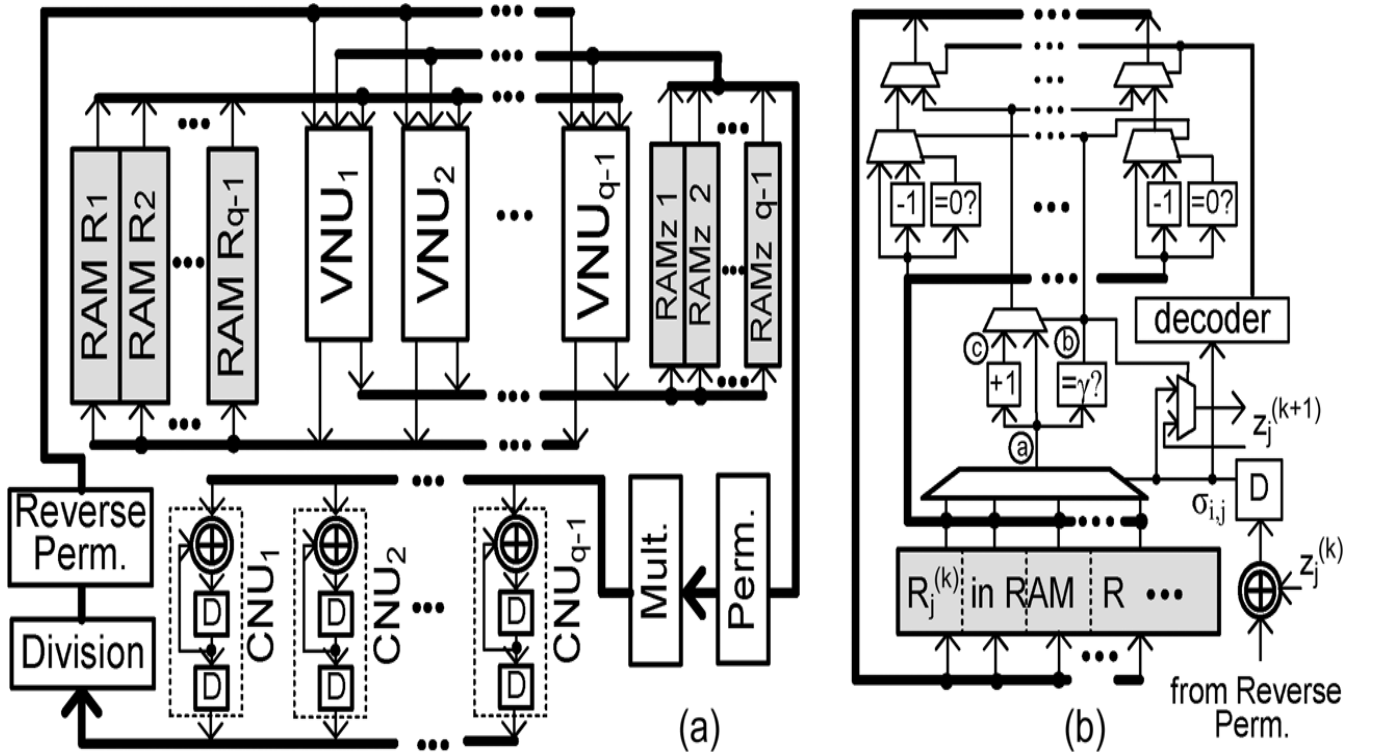


Fig. 5. (a) IHRB-MLGD architecture for QCNB-LDPC codes. (b) VNU architecture when all messages are kept.

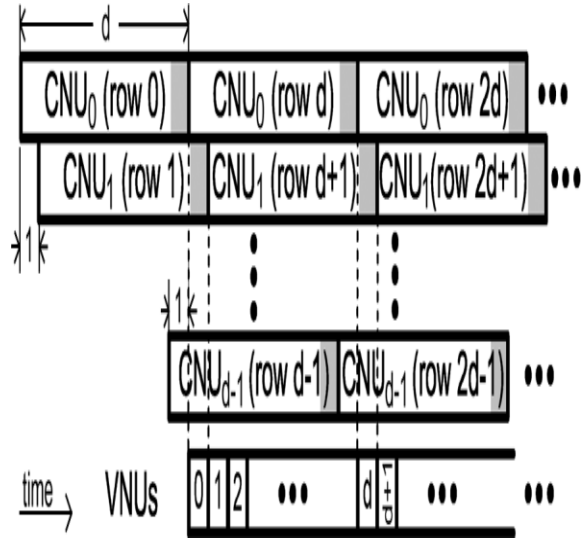


Fig. 7. Computation scheduling in IHRB-MLGD for cyclic codes.

VII. CONCLUSION

This paper proposed enhancement schemes to the IHRB decoding algorithm for NB-LDPC codes. The proposed schemes lead to significant coding gain with small complexity overhead. In addition, efficient architectures were developed for

both QC and cyclic NB-LDPC codes based on the IHRB and E-IHRB algorithms. With moderate Performance loss, the proposed decoders can achieve at least tens of times higher efficiency compared to previous designs based on the Min-max algorithm. Future work will be devoted to further improving the performance and reducing the hardware complexity of MLGD-based algorithms for NB-LDPC decoding.

References

- [1] V. Savin, "Min-Max decoding for non binary LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory*, 2008, pp. 960–964.
- [2] 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 *Proc. Int. Symp. Commun. Inf. Tech.*, 2007, pp. 1201–1206.
- [3] J. Lin, J. Sha, Z. Wang, and L. Li, "An efficient VLSI architecture for nonbinary LDPC decoders," *IEEE Trans. Circuits Syst. II, Exp. Briefs* vol. 57, no. 1, pp. 51–56, Jan. 2010.