

High Speed Rate Compatible Ldpc Codes For Long Distance Transmission with Fec

Mr. SK.MASTAN¹, Miss.P.RAJINI DEVI²

1. Associate Professor, Department of ECE, Jayamukhi Institute of Technological Sciences, Warangal, India

2. Department of ECE, Jayamukhi Institute of Technological Sciences, Warangal, India.

Abstract: In this paper, we propose a rate-compatible FEC scheme based on LDPC codes together with its soft-ware reconfigurable unified FPGA architecture. By FPGA emulation, we demonstrate that the proposed class of rate-compatible LDPC codes based on puncturing and general-ized LDPC coding with an overhead from 25% to 46% provides a coding gain ranging from 12.67 dB to 13.8 dB at a post-FEC BER of 10-15. As a result, the proposed rate-compatible codes represent one of the strong FEC candidates of soft-decision FEC for both short-haul and long-haul optical transmission systems.

Keywords: Rate-compatible (RC) codes, irregular low-density parity-check (LDPC) codes, cycle counting algorithm, Extrinsic Message Degree (EMD), puncturing technique, extension scheme.

1. Introduction

Low-density parity-check (LDPC) codes belonging to a class of linear block codes are among the superior error-correcting codes with simple decoding procedure. In general, LDPC codes can be represented by sparse parity-check matrix (PCM) H which contains only fewer entries of non-zero elements. The sparseness of non-zero elements in rows and columns of H determines the degree distribution of the code. If there is a uniform degree distribution in a row (check node) weight and column (variable node) weight of PCM H , then the LDPC code is called the regular LDPC code otherwise, it is an irregular LDPC code. Based on the degree of parallelism, LDPC decoders are classified into two categories: (1) LDPC decoders with highly parallel design and (2) LDPC decoders with partially parallel design. LDPC decoders with a high degree of parallelism are most often preferred in designing of power-efficient and high throughput LDPC decoders. These types of decoders usually require higher processing units and have high interconnect complexity. In order to minimize the excessive routing and hardware complexity of the highly parallel decoding architectures, reduced complexity min-sum and split-row min-sum schemes were proposed. On the other hand, the LDPC decoders with partially parallel designs are preferred for designing decoders with smaller area and high core utilization. The major drawbacks of LDPC decoders with partially parallel design is that they achieve low decoding throughput with the increased memory overhead.

Modern high-speed optical communication systems require high-performance FEC engines that can support throughputs of 100Gbit/s or multiple thereof, which have low-power consumption, while providing high net coding gains at a target bit-error rate of 10-15, and which are preferably adaptable to the time-varying optical channel conditions. Thanks to their remarkable error-correcting capabilities and recent development of corresponding integrated circuits, low-density parity-check (LDPC) codes and turbo-product codes have led to their recent inclusion in several standards in optical community, such as ITU-T G.975, G.709; and hence represent the key technologies enabling the development of 400Gb/s, 1Tb/s Ethernet, and beyond. Recently, soft-decision binary and non-binary LDPC codes with an outer hard-decision code, usually high-rate Reed-Solomon (RS) or BCH code, pushing the system BER to levels below target BER. However, the implementation of such a concatenated LDPC codes requires a thorough understanding of the LDPC code performance and a properly designed interleaver between the LDPC code and the outer code should be employed to avoid the errors at the output of LDPC decoder.

II. LDPC CODES

When the channel state information (CSI) is known at the transmit end and the data transmission takes place over time-varying channels, an error control coding scheme with a fixed code rate is not the best solution. In such situations, an error-correction scheme with flexibility in code rates is desirable since such scheme can encode data at different rates depending on the reliability of the channel. High-rate codes are applied to achieve higher data throughput if the channel condition is good, otherwise low-rate codes are used to ensure reliable transmission. Thus, both capacity and reliability can be realized in such scenarios. However, deploying many pairs of encoders and decoders is not feasible in practical applications due to their high memory and storage costs. Rate-compatible (RC) codes refer to a family of codes where higher rate codes are embedded in lower rate codes; in other words, the factor graphs of higher rate codes are subgraphs of lower rate codes. For example, Lin and Yu designed an RC coding scheme for a hybrid automatic repeat-request with forward error correction (ARQ/FEC) system, where the transmitter sends additional redundant bits upon request until the decoder claims a successful decoding. Having been applied to convolutional codes and turbo codes, RC techniques are proven not only to enhance system performance but also to only require low hardware complexity thanks to the structure of a single pair of encoder and decoder.

Construction 1A. An M by N matrix (M rows, N columns) is created at random with weight per column t (e.g., $t = 3$), and weight per row as uniform as possible, and overlap between any two columns no greater than 1. (The weight of a column is the number of non-zero elements; the overlap between two columns is their inner product.)

Construction 2A. Up to $M/2$ of the columns are designated weight 2 columns, and these are constructed such that there is zero overlap between any pair of columns. The remaining columns are made at random with weight 3, with the weight per row as uniform as possible, and overlap between any two columns of the entire matrix no greater than 1.

Constructions 1B and 2B. A small number of columns are deleted from a matrix produced by constructions 1A and 2A, respectively, so that the bipartite graph corresponding to the matrix has no short cycles of length less than some length l . The above constructions do not ensure that all the rows of the matrix are linearly independent, so the $M \times N$ matrix created is the parity matrix of a linear code with rate at least $R \equiv K/N$, where $K = N - M$. We report results on the assumption that the rate is R . The generator matrix of the code can be created by Gaussian elimination. We simulated a Gaussian channel with binary input $\pm a$ and additive noise of variance $\sigma^2 = 1$. If one communicates using a code of rate R then it is conventional to describe the signal to noise ratio by $\frac{E_b}{N_0} = \frac{a^2}{2R\sigma^2}$ and to report this number in decibels as $10 \log_{10} E_b/N_0$.

Construction of RC-LDPC Codes Using Puncturing:

Given a mother or parent LDPC code containing K information bits, the code rate is given by $R = K/N$ where N is the block length. Suppose that m represents the message with K bits from the source, c denotes the encoded data with N bits, c' is the punctured data, and \hat{m} denotes the estimate of the original message using a belief propagation (BP) decoding algorithm given the unpunctured data r . The log-likelihood ratio (LLR) of a punctured bit is set to 0 at the beginning of the decoding process. Suppose that P bits are punctured before transmission, so the resulting code rate is given by $R' = K/(N - P)$ and the puncturing rate by $\rho = P/N$. We assume that the decoder has perfect knowledge regarding the puncturing pattern, i.e., the positions of punctured bits in a code word. Otherwise, some side information is needed to send the puncturing pattern to the receiver end. Puncturing is a common and simple method to construct RC codes, for which a higher rate is achievable by means of removing a subset of encoded bits c .

A randomly chosen puncturing pattern can be used to realize the rate compatibility at the expense of significant performance degradation. Intentional puncturing methods have been investigated for short block-length LDPC codes ranging from asymptotic analysis to grouping and sorting variable nodes. In contrast to those methods, the proposed puncturing schemes aim to reduce the performance loss caused by puncturing from a cycle distribution perspective, i. e., the puncturing pattern is selected in the sense that the removed bits will break a certain number of short cycles, which significantly improves the connectivity of the TG.

III. Proposed method

FPGA architecture of proposed RC LDPC Decoder

In this Section, we first provide the overall architecture of our FPGA emulation platform. After that we provide the details of the pro-posed FPGA architecture for the proposed RC LDPC decoder. The puncturing-based LDPC decoder is used as a reference.

Overview Architecture

The performance of the proposed rate-adaptive LDPC codes is demonstrated using a set of FPGAs, whose high-level diagram is presented in Fig. 1. This high level verification tool is similar to others and consists of three main parts: a set of Gaussian noise generators, a partially pipelined LDPC decoder, and an error counter block. The Gaussian noise generator generates noise samples using a combination of Box-Muller algorithm and central limit theorem following the guidelines. Such generated sequence of noise samples is multiplied with standard deviation of noise σ and fed to the LDPC decoder with quantized LLRs. There are three types of software reconfigurable registers to be initialized in the high-level diagram. The first set of registers are instantiated to define the noise variance and the codeword to be transmitted, the second set of registers to configure the rate adaptive LDPC decoder, including the number of layered iterations, the amount of bit to be punctured, and the number of generalized check nodes; and the third set of registers stores the number of uncoded errors, coded errors, and transmitted code words.

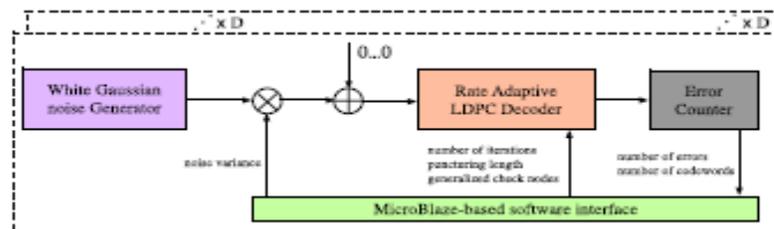


Fig. 1. High-level schematic of the FPGA-based evaluation platform.

Rate-Compatible LDPC Decoder Architecture

Fig. 2(a) presents the overview of the re-configurable binary LDPC decoder, which consists of two parts; namely, memories and processors. The processors can be classified into following categories:

- (i) variable-node processor (VNP) corresponding
 - (ii) Scaled min-sum check-node processor (SMS-CNP)
 - (iii) MAP based CNP (MAP-CNP, also referred as generalized-CNP) and
 - (iv) early termination unit (ETU).
- Four sets of block RAMs are used in the implementation of RC LDPC decoder: (i) block RAM with size of stores the channel LLRs, (ii) block RAM stores the messages from check-nodes to variable-nodes,
- (iii) Block RAM with size of n stores the decoded bits, and

- (iv) Additional block RAM inside each CNP stores the intermediate values. In discussion above, γ de-notes the column weight, n is the codeword length. There are two types of reconfigurable routers used in the implementation as well: (i) puncturing-mux is used to select the input either be the outputs of VNP or the largest positive LLR, and (ii) generalized-mux selects either the outputs of SMS-CNP or MAP-CNP.

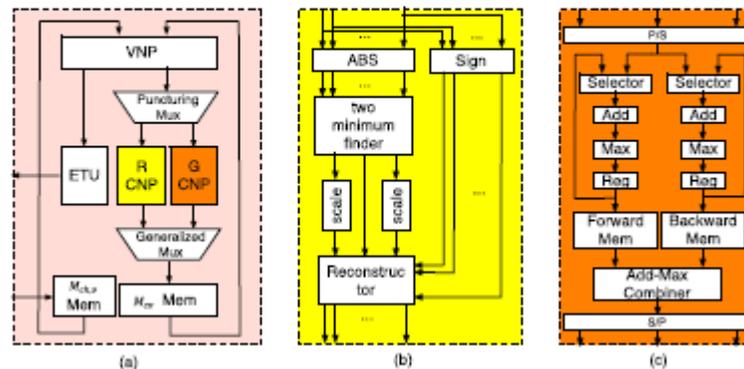


Fig.2. Rate-adaptive LDPC decoder architecture: (a) overall architecture, (b) architecture of regular CNP, and (c) architecture of BCJR-based generalized CNP.

As with other codes, the maximum likelihood decoding of an LDPC code on the binary symmetric channel is an NP-complete problem. Performing optimal decoding for a NP-complete code of any useful size is not practical.

However, sub-optimal techniques based on iterative belief propagation decoding give excellent results and can be practically implemented. The sub-optimal decoding techniques view each parity check that makes up the LDPC as an independent single parity check (SPC) code. Each SPC code is decoded separately using soft-in-soft-out (SISO) techniques such as SOVA, BCJR, MAP, and other derivate thereof. The soft decision information from each SISO decoding is cross-checked and updated with other redundant SPC decoding of the same information bit. Each SPC code is then decoded again using the updated soft decision information. This process is iterated until a valid code word is achieved or decoding is exhausted. This type of decoding is often referred to as sum-product decoding.

The decoding of the SPC codes is often referred to as the "check node" processing, and the cross-checking of the variables is often referred to as the "variable-node" processing.

In a practical LDPC decoder implementation, sets of SPC codes are decoded in parallel to increase throughput. In contrast, belief propagation on the binary erasure channel is particularly simple where it consists of iterative constraint satisfaction.

IV. Simulation results

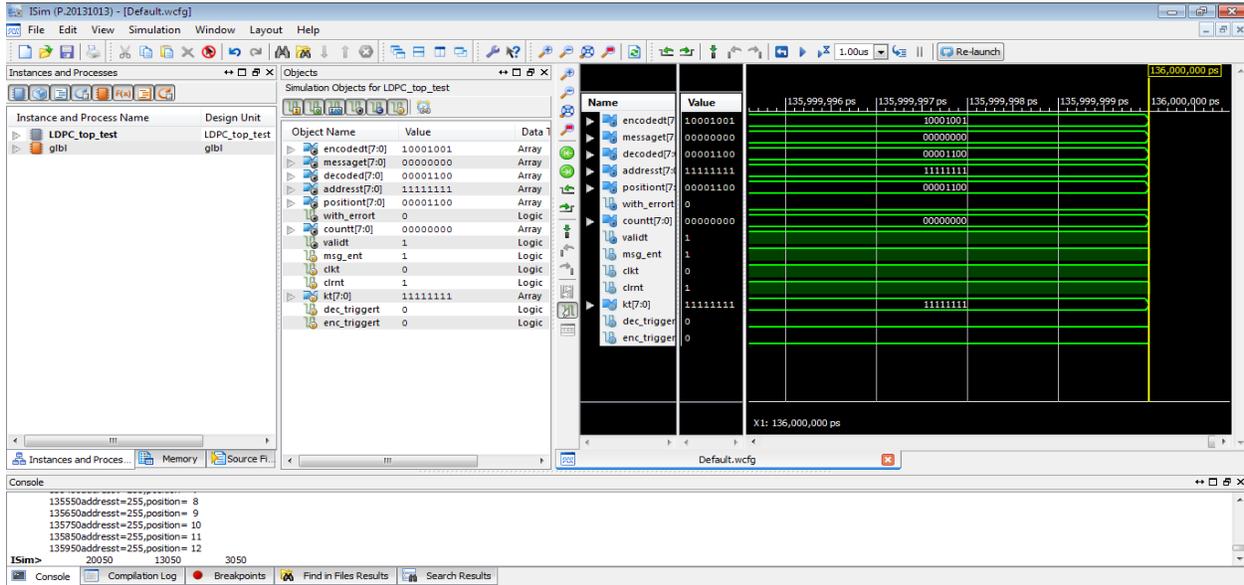


Fig.3.Simulated output

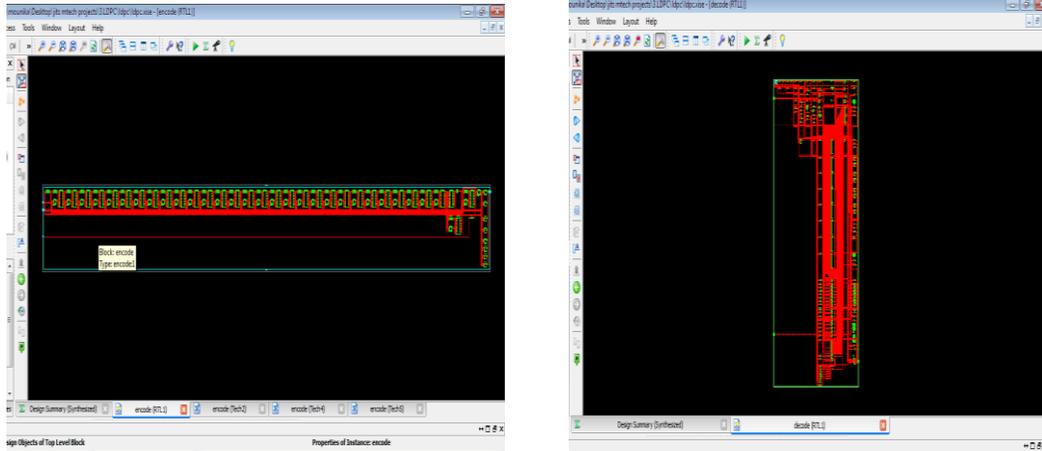


Fig.4.RTL schematic for encoder and decoder

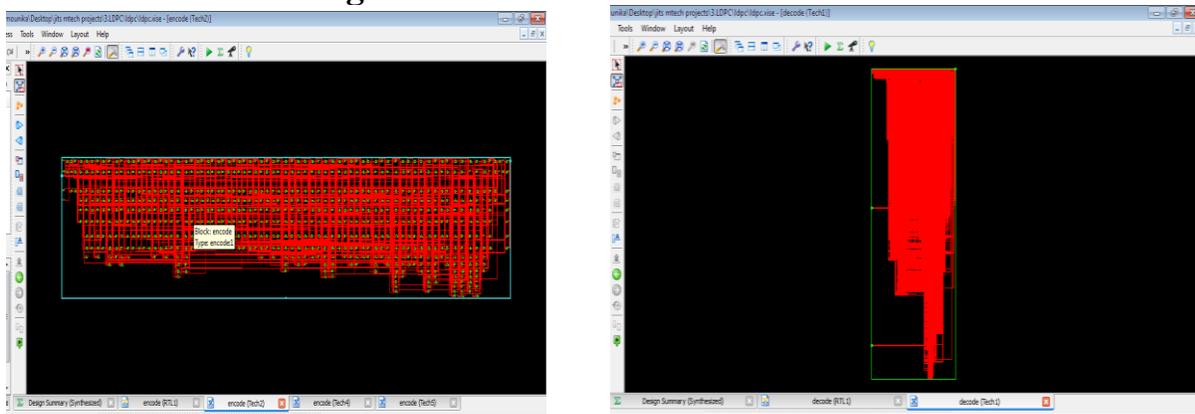


Fig.5.Technology Schematic of encoder and decoder

V.CONCLUSION

An important outcome of the work with algebraic codes was the demonstration that highly redundant parity-check matrices can lead to very good iterative decoding performances without the need for very long block lengths. While the probability of a random graph having a highly redundant parity-check matrix is vanishingly small, the field of combinatorial designs offers a rich source of algebraic constructions for matrices which are both sparse and redundant.

REFERENCES:

- [1] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.
- [2] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. IT-27, no. 5, pp. 533–547, September 1981.
- [3] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. 30th ACM Symp. on the Theory of Computing*, 1998, pp. 249–258.
- [4] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 585–598, February 2001.
- [5] S.-Y. Chung, G. D. Forney, Jr., T. J. Richardson, and R. L. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Letters*, vol. 5, no. 2, pp. 58–60, February 2001.
- [6] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 638–656, February 2001.
- [7] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 399–431, March 1999.
- [8] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Upper Saddle River, NJ 07458: Prentice Hall, 1995.
- [9] S. Lin and D. J. Costello Jr., *Error Control Coding*, 2nd ed. New Jersey: Prentice Hall, 2004.
- [10] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam: North-Holland, 1977.
- [11] W. C. Huffman and V. Pless, *Fundamentals of Error Correcting Codes*. Cambridge University Press, Cambridge UK, 2003.
- [12] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, October 1996.
- [13] A. Papoulis, *Probability, random variables and stochastic processes*. 2nd Ed. Singapore: McGraw-Hill, 1984. 80
- [14] T. J. Richardson and R. L. Urbanke, "The capacity of low-density paritycheck codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, February 2001.
- [15] M. G. Luby, M. Mitzenmacher, and Shokrollahi, "Analysis of random processes via and-or tree evaluation," *Proceedings of the 9th Annual ACM/SIAM Symposium on Discrete Algorithms*, pp. 364–373, 1998.