

Multiple-Rate Low-Density Parity-Check Codes with Constant Blocklength

Andres I. Vila Casado, Wen-Yen Weng, Stefano Valle, and Richard D. Wesel, *Senior Member, IEEE*

Abstract—This paper describes and analyzes low-density parity-check code families that support variety of different rates while maintaining the same fundamental decoder architecture. Such families facilitate the decoding hardware design and implementation for applications that require communication at different rates, for example to adapt to changing channel conditions. Combining rows of the lowest-rate parity-check matrix produces the parity-check matrices for higher rates. An important advantage of this approach is that all effective code rates have the same blocklength. This approach is compatible with well known techniques that allow low-complexity encoding and parallel decoding of these LDPC codes. This technique also allows the design of programmable analog LDPC decoders. The proposed design method maintains good graphical properties and hence low error floors for all rates.

Index Terms—Channel coding, low-density parity-check (LDPC) codes, multiple-rate codes.

I. INTRODUCTION

PRACTICAL communication systems often need to operate at several different rates. To keep the implementation as simple as possible, the same basic hardware architecture should be able to decode the encoded data at all possible rates. One way to achieve this with Low-Density Parity-Check (LDPC) codes is to generate higher-rate codes by puncturing lower-rate codes, as proposed in [1], [2] and [3]. However, puncturing reduces the code blocklength, which degrades performance. For the highest-rate codes, where the puncturing is most severe, the performance degradation is significant when compared to an LDPC code with the original blocklength.

Paper approved by K. Narayanan, the Editor for Coding and Communication Theory of the IEEE Communications Society. Manuscript received April 24, 2006; revised February 26, 2007 and August 30, 2007.

This work was supported by the state of California and three corporations; ST Microelectronics, Conexant, and Texas Instruments, through UC discovery grant COM 03- 2, UC MICRO grant 03-92, and UC MICRO grant 03-93 respectively. This paper was presented in part at the Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, 2004.

A. I. Vila Casado is with Mojix Inc, Los Angeles, CA 90025 USA (e-mail: andres@mojix.com).

W.-Y. Weng is with the Department of Computer Science and Information Engineering, Chung Hua University, Taiwan (e-mail: wenyen.weng@chu.edu.tw).

S. Valle is with STMicroelectronics Srl, Cornaredo (MI), Italy (e-mail: stefano.valle@st.com).

R. D. Wesel is with the Electrical Engineering Department, University of California, Los Angeles, CA 90095 USA (e-mail: wesel@ee.ucla.edu).

Digital Object Identifier 10.1109/TCOMM.2009.0901.060256

Another way to achieve this is to generate lower-rate codes by shortening higher-rate codes, as described in [2]. As with puncturing, shortening reduces the code blocklength, which degrades performance. For the lowest-rate codes where the shortening is most severe, the performance degradation is significant when compared to an LDPC code with the original blocklength.

This paper presents a code structure that supports a wide range of rates while maintaining a constant code blocklength. The basic idea is to generate higher rate codes (called effective codes in this paper) from a low-rate code (called the mother code in this paper) by reducing the number of rows in its parity check matrix. From an implementation point of view, rows in the parity-check matrix correspond to check nodes. We propose to reduce the number of rows by linearly combining the mother code rows, which is equivalent to replacing a group of check nodes with a single check node that sums all the edges coming into each of the original check nodes. This is also equivalent to the code that results by connecting the check nodes of the mother code through a new check node, as seen in Fig. 1. These equivalences hold as long as the combined check nodes do not have any variable-node neighbors in common.

Multiple-rate codes can be designed by generating effective codes solely by combining rows as discussed above, and in this paper these codes are called Strict Row-Combining (SRC) codes. A performance improvement can be obtained by adding a few edges and deleting a few other edges in the graph as the rows are combined. This allows the code to have good variable-node degree distributions at each rate. In this paper these codes will be called Row Combining with Edge Variation (RCEV) codes. Both approaches will be presented in the following subsections.

Section II describes the row-combining approach in detail. Section III explains how row combining helps to simplify decoder architectures and consequently to reduce chip area. Section IV describes how to lower the complexity of the encoder and decoder of row-combining codes. A design method for SRC codes is proposed in Section V. Section VI describes the RCEV code design approach that results in an improvement in performance by relaxing some constraints imposed in the SRC design. Section VII compares the performance of SRC, RCEV, single-rate stand-alone codes, and punctured codes. Section VIII delivers the conclusions.

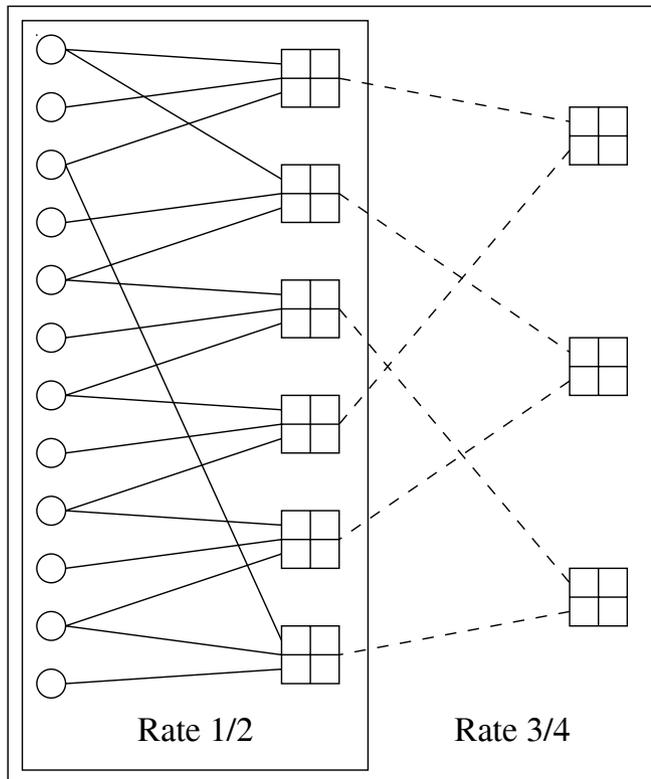


Fig. 1. Graph of a rate-3/4 LDPC code obtained from a rate-1/2 LDPC code via row combining.

II. ROW-COMBINING CODES

Consider the example mother LDPC matrix in (1),

$$H_{\frac{1}{2}} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}. \quad (1)$$

This is a rate-1/2 mother LDPC matrix with blocklength 12 whose graph representation can be seen in Fig. 1. This is by no means a good LDPC code but the reader should see it as an example to explain row combining. Fig. 1 also shows that replacing each pair of nodes with a new single node transforms this rate-1/2 code into a rate-3/4 code. This is equivalent to summing the rows of the mother LDPC matrix that correspond to the check nodes that are combined, since the check nodes in the example do not have any common neighbors. In general, the mother matrix should be designed so that the rows that will be combined don't have ones in the same column.

Eq. (2) gives the effective rate-3/4 LDPC matrix that results from the row combining described in Fig. 1, where combining rows 1 and 4 of the mother matrix produces row 1 of the effective matrix, combining rows 2 and 5 of the mother matrix produces row 2 of the effective matrix, and combining rows 3 and 6 of the mother matrix produces row 3 of the effective

matrix,

$$H_{\frac{3}{4}} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}. \quad (2)$$

It is easy to see that many different rates can be obtained from the same mother code by changing the way rows are combined. For example using the mother code described in (1), combining three rows at a time generates the rate-5/6 LDPC matrix shown in (3). In this rate-5/6 matrix the first row results from combining the odd rows of the mother matrix and the second row results from combining the even rows of the mother matrix,

$$H_{\frac{5}{6}} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}. \quad (3)$$

In general, row combining changes the rate without changing the blocklength or the basic architecture of the decoder.

III. IMPACT OF ROW COMBINING ON THE HARDWARE ARCHITECTURE

A. Digital Decoders

The main goal of row-combining is to simplify the support of multiple rates. In general, row-combining won't lead to a faster decoder for any particular rate but provides a simple overall architecture with a smaller chip area required to support all the needed rates. Here are a couple of examples of how decoder simplification might be accomplished.

If the message passing is done through a memory, there is a list of memory addresses that tell the processor which variable-to-check messages to compute in order to compute a check-to-variable message. With row-combining, changing the rate of the code can be achieved by replacing the lists for the combined check nodes with a single list that is the union of those lists, thus changing only the quantity of messages read. This is a simple change that can be done on the fly with a careful implementation.

Another possible hardware implementation can be developed using the fact that the code produced by row combining is equivalent to the code that results from connecting the check nodes through another check node as shown in Fig. 1. An efficient hardware implementation would decode the higher rate code by performing the extra-message generation on top of the message computing done for the low rate codes, thus maintaining the basic hardware architecture. This idea was used in [4], where an architecture that exploits our row-combining structure was implemented. These examples, while not exhaustive, capture the essence of the simplification facilitated by row combining.

On top of these simplified multiple-rate decoders, the fact that the codes are designed such that the check nodes that will be combined don't have any neighbors in common guarantees some degree of parallelism. Check nodes that will be combined can be processed in parallel because they will never try to access the information of the same variable node. There are, of course, many ways to achieve parallelism. Our point here is that row combining is completely compatible with LDPC codes that support highly parallel decoding architectures.

B. Analog Decoders

Analog decoders of turbo-like codes were introduced simultaneously in 1998 by Hagenauer [5] and Loeliger *et al.* [6]. A good in-depth discussion of this alternative decoding hardware can be found in [7]. Analog decoders are analog circuits that oscillate until an equilibrium state is reached. Analog decoding shows promise because the decoders require low power and the convergence is typically faster than with digital decoders. Digital decoders can use the same hardware to decode different LDPC codes by re-programming the chip. Their analog counterparts are not programmable, and they need different circuits to decode different LDPC codes. Since many applications need various codes with different rates, the lack of programmability of analog decoders makes them less attractive than digital decoders when it comes to LDPC codes.

However, row-combining codes allow programmable analog decoders. An analog decoder that works for all the rates would consist of the circuit that decodes the mother code, with switches that turn on and off the connections to the new check nodes that increase the rate of the code. These connections are shown in Fig. 1 with dashed lines.

IV. LOW COMPLEXITY ENCODING AND DECODING

A. Block Structure for High-Speed Decoder Implementation

Despite the intrinsic parallelism of the decoding process, the first attempts to design high-throughput LDPC decoders encountered significant interconnection problems. In fact, the exploitation of such parallelism requires the use of several check-and-variable-node processors, each connected to different memories in order to access several messages simultaneously. The random character of the node connection in the Tanner graph results in difficult memory/processor placement and intractable-routing problems.

In order to solve these problems, LDPC codes can be designed to have an inherent structure as suggested by Mansour and Shanbag in [8]. This approach also enables the implementation of high-speed decoders without memory fragmentation such as those presented in [9] and [10]. In [8] the LDPC matrices have a block structure that consists of square sub-matrices each of size p . Each square sub-matrix is either a zero sub-matrix or a structured sub-matrix.

An example that illustrates the structured sub-matrices proposed in [8] for $p = 4$, is shown in (4). This sub-matrix, labelled as S_2 , results from performing a right cyclic shift of 2 columns on the identity matrix of size p . Each sub-matrix S_i is produced by cyclically shifting the columns of an identity matrix to the right i places,

$$S_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}. \quad (4)$$

This structured LDPC matrix allows the decoder to use at least p processors in parallel and doesn't preclude the implementation of faster decoders that use a multiple of p processors as suggested in [4].

B. Low-complexity Encoding Using Back Substitution

The parity-check matrix should also allow a simple encoder implementation. In [11] and [12] a low-complexity encoder for LDPC codes is found if its parity-check matrix H_0 is composed of two matrices $H_0 = [H_1|H_2]$ where H_2 is a square sub-matrix that has a particular structure. For those codes encoding complexity grows linearly with the code length. In [11] H_2 is a bi-diagonal matrix so the encoding process is simply multiplying the input vector by H_1^T and then processing by an accumulator.

Unfortunately it is difficult or impossible to maintain a bi-diagonal structure for the H_2 portion of the mother LDPC matrix and all of the effective LDPC matrices in the context of row combining. One solution found to this problem is to use a model presented in [12] where H_2 is a lower triangular matrix. This will allow a low-complexity encoder based on back-substitution as explained in [12]. Back substitution obtains the parity bits by solving a sequence of linear equations that have only one unknown bit. This structure can be maintained for all the rates, as will be shown in the following subsection.

C. Structured Row-Combining Codes

The challenge presented when trying to design a structured row-combining code is that the mother code (denoted by M) and all the effective codes (denoted by E) must have both the sub-matrix structure (for parallel decoding) and the lower triangular structure (for low-complexity encoding).

There is an easy way to maintain the sub-matrix structure for all the effective codes. Instead of combining individual rows, rows of sub-matrices will be combined so that if sub-matrix A and sub-matrix B are combined it implies that row i of A and row i of B will be combined for $i = \{1, \dots, p\}$. The resulting sub-matrix will be equivalent to the superposition of sub-matrix A and sub-matrix B .

If the exact sub-matrix structure of [8] is to be maintained for all the effective codes, the mother code and all effective codes have to be designed so that among the combined sub-matrices at most only one is non-zero (and has the S_i structure). However, if the sub-matrix that results from the superposition of two or more non-zero S_i sub-matrices also has good parallel properties (which depends on the hardware architecture of the decoder), this design criteria can be relaxed to include such superpositions.

Furthermore, it's desirable that the the square H_2 sub-matrices of both the mother code and all effective codes have a lower triangular structure that would allow back-substitution encoding. This is achieved by imposing a constraint in selection of the rows to be combined, assuming that the H_2 sub-matrix of the mother code is lower triangular. Let us assume that row-combining generates an effective code E_i that has r_i check-nodes. As long as the bottom r_i rows of the mother code remain in their pre-row-combining relative positions and are not combined among themselves, the square H_2 sub-matrix (with size r_i) of the effective code E_i is lower triangular. The examples presented in Section II satisfy this condition. The size-6 square H_2 sub-matrix of the rate-1/2 mother LDPC matrix shown in Eq. (1) is lower triangular. The effective rate-3/4 code has $r_i = 3$ check-nodes and the

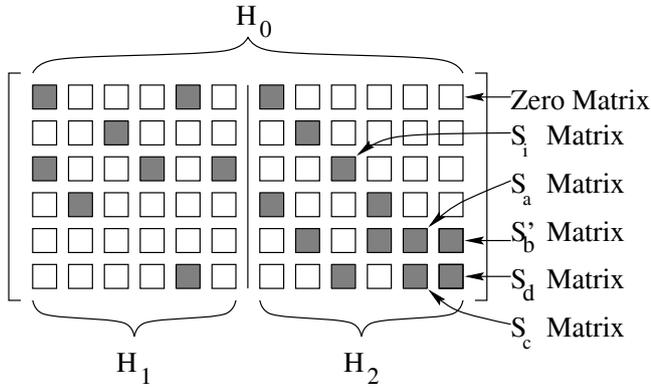


Fig. 2. Structured LDPC matrix.

combining is performed so that the bottom 3 rows are not combined among themselves and remain in their pre-row-combining relative positions. It can be seen in Eq. (2) that the size-3 square H_2 sub-matrix of the effective rate-3/4 code is also lower triangular.

One way to combine the sub-matrix structure with the lower triangular structure of H_2 is to make H_2 block lower triangular. This implies that the sub-matrices along the diagonal of H_2 will have the S_i structure and all the sub-matrices that are above this diagonal will be zero sub-matrices. The problem with this structure is that the rightmost p columns (where p is the size of the sub-matrices) will be degree-one columns which will negatively affect the performance of the codes. One possible solution of the problem is to make the bottom right matrix have the bi-diagonal structure described in (5). This structure allows back-substitution and only 1 column will have degree 1,

$$S_s = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}. \quad (5)$$

Another solution can be seen in Fig. 2 which shows a LDPC structure that allows both parallel decoding and low-complexity encoding. In this structure the four bottom right matrices are carefully designed such that they allow back-substitution encoding. For the S_i sub-matrices with subscripts $i = a, b, c, d$, the subscripts (number of columns shifted) must satisfy $(a - c + d - b) \bmod p = 1$. The sub-matrix labelled S'_b is an S_i sub-matrix with the first row set to all zeros. This structure guarantees that the last $2p$ parity bits can be generated by solving one parity check equation with one unknown bit, thus allowing back substitution. In this structure only 1 column has degree 1.

V. DESIGN OF STRUCTURED SRC CODES

This section proposes a design method for structured row-combining codes given the blocklength of the code, the mother and effective rates and the sub-block size p . Since only row combining is allowed to generate the high-rate matrices, these codes are called Strict Row-Combining (SRC) codes.

The first step is the selection of both the variable-node degree distribution and the check-node degree distribution.

As seen in Fig. 1, the number of neighbors of the variable nodes remains the same as the rate changes, thus the variable-node degree distribution will also remain unchanged. This implies that this degree distribution cannot be optimized for the different rates of the code, so a degree distribution that's optimized for the highest-rate code is chosen. This is due to the fact that row-combining imposes some specific constraints on the variable-node degree distribution of the lower-rate codes, which will be explained in detail in Section VI.

A concentrated degree distribution is a degree distribution in which every node has the same degree or all the degrees are within one of each other. Concentrated check-node degree distributions tend to approximate theoretical optimality [13]. Therefore, the check node degree distribution of the mother and effective codes should be concentrated, if possible.

The check-node degree distributions depend on the selection of the rows to be combined. This selection is the next step in the proposed design of SRC codes. Since the check-node degree distribution of the mother code is concentrated, then the check node degree distribution for the higher-rate effective code will also be concentrated if all the rows in the effective LDPC matrix result from combining the same number of rows of the mother LDPC matrix. It is also necessary that the row combining maintains the structure of the codes presented in Section IV.

There is a simple way to achieve this if the desired rates have the $(a - 1)/a$ form. The mother code is set to have a square matrix with a concentrated check node degree distribution. Such a mother code has rate 0 and is used only to generate effective codes; it is not a useful stand alone code. Combining a rows (or rows of sub-matrices in the case of structured codes) at a time, generates a code with rate $(a - 1)/a$ as long as the total number of rows of the mother matrix is a multiple of a . This effective code will have a concentrated check node degree distribution. This shows that SRC codes can maintain a concentrated check node degree distribution among all its rates if they all are in the $(a - 1)/a$ form. Rates of this form comprise a useful set of rates. In standards such as IEEE 802.11n, all LDPC code rates have this form. Furthermore, puncturing and/or shortening can be used with row combining in order to support more rates. As long as the number of punctured and/or shortened bits is small, the effective blocklength of the code is not significantly diminished.

Given that two check nodes that will be combined cannot have any common neighbors, row combining introduces some constraints in the construction of the LDPC matrices. These constraints may limit the degrees of freedom of the LDPC design, especially in the case of structured LDPC codes where two sub-matrices that will be combined can't be both non-zero. In order to minimize this problem, the overall number of row combinations across all rates must be as low as possible. This can be achieved by first choosing the row combining that generates the highest-rate code. Once this is done, the row-combining that generates the second highest rate-code is chosen using as many row combinations from the highest rate-code as possible. If all the row-combining strategies are chosen this way the overall number of combinations will be minimized. This minimization of the number of combinations

is also beneficial from an implementation point of view.

The remaining issue is to assign the positions and right cyclic shifts of the non-zero sub-matrices in the mother code. It is well known that the performance of the LDPC codes is limited by the fact that their graphs contain cycles which compromise the optimality of the belief propagation decoding. These cycles generate error floors in the performance of LDPC codes in the high SNR regions. However, the negative effect of the cycles can be reduced using graph-conditioning techniques such as those described in [14] and [15]. So the last step in the design will be to use the ideas described in these works, adapting them to work on structured SRC codes.

As explained in [14] not all cycles degrade the performance of the code in the same way. Among the most dangerous structures that can be found in LDPC bi-partite graphs are stopping sets. A stopping set is a variable node set where all its neighbors are connected to the set at least twice. This implies that if all the variable nodes that belong to a stopping set are unreliable, the decoding will fail.

Small stopping sets must be avoided. This is a complex problem to attack directly. To indirectly increase the size of stopping sets, [14] proposes the ACE algorithm, which is based on maximizing the ACE metric of small cycles. The ACE metric of a cycle is the sum of the number of neighbors of each of the variable nodes in the cycle minus two times the number of variable nodes in the cycle.

According to this algorithm, the LDPC matrix is constructed by generating a single column randomly until one is found where all the cycles of length less than or equal to a previously specified threshold (denoted as $2d_{ACE}$) that contain its corresponding variable node have an ACE metric higher than or equal to another previously specified threshold (denoted as η_{ACE}). This process sequentially produces all the columns starting from the one with the lowest degree.

The constraints specified in [15] also help to avoid small stopping sets in the graph, particularly if applied to the high-degree columns. In [15] two new metrics are defined called β_c and β_p and they are the number check nodes connected only once to a cycle or a path respectively. In the same manner as the ACE algorithm, random columns are generated and they must satisfy some constraints based on previously specified thresholds which are denoted here as d_{SS} , γ_c and γ_p . Specifically, a randomly generated column is valid if all the cycles of length less than or equal to $2d_{SS}$ that contain its corresponding variable node have a β_c metric of value higher than or equal to γ_c and if all the paths of length less than or equal to d_{SS} that contain its corresponding variable node have a β_p metric of value higher than or equal to γ_p .

The generation of the mother and effective matrices is done using simultaneous graph conditioning. This means that the previously described column by column generation is still used and every column generated must satisfy the graph constraints specified for the mother code and all the effective codes. Different graph constraints can be used for different matrices, which is necessary because the achievable values of these constraints change with the rate as shown in [16]. In the case of structured SRC codes, instead of a column by column generation, all the columns that correspond to the same sub-matrices will be generated at the same time.

The design procedure of structured SRC codes is shown in Algorithm 1 where V_j denotes a set of variable nodes that belong to a sub-matrix, V_j^M denotes a column of sub-matrices in the mother code M and $V_j^{E_i}$ denotes a column of sub-matrices in the effective code E_i . Algorithm 1 may not converge because the conditions are too restrictive. In that case, we can relax the graph-constraints by lowering the values of d_{ACE} , η_{ACE} , d_{SS} , γ_c , and/or γ_p . However, even if there aren't any graph-constraints, Algorithm 1 may not converge because there may be too many effective rates that impose too many constraints thus the number of effective-rates must be lowered.

Algorithm 1 SRC code design

- 1: Choose rates, blocklength (n) and sub-matrix size (p)
 - 2: Choose variable node degree distributions
 - 3: Choose rows of sub-matrices to be combined in order to generate the higher-rate codes
 - 4: **for** all columns of sub-matrices V_j **do**
 - 5: Randomly generate V_j^M according to the degree distribution selected in 2
 - 6: **if** V_j^M doesn't satisfy the graph constraints set for M **then**
 - 7: Discard V_j and go to line 5
 - 8: **end if**
 - 9: **for** all effective codes E_i **do**
 - 10: Compute $V_j^{E_i}$ according to the row combinations selected in 3
 - 11: **if** two non-zero sub-matrices are combined **then**
 - 12: Discard V_j and go to line 5
 - 13: **end if**
 - 14: **if** $V_j^{E_i}$ doesn't satisfy the graph constraints set for E_i **then**
 - 15: Discard V_j and go to line 5
 - 16: **end if**
 - 17: **end for**
 - 18: **end for**
-

VI. ROW COMBINING WITH EDGE VARIATION (RCEV) CODES

The main disadvantage with SRC codes is that the mother code and all of the effective codes of an SRC code have the same variable-node degree distribution. With strict row combining, edges are neither created nor deleted. This is problematic since in principle different rates require different variable-node degree distributions for theoretical optimality, as stated in [13]. Row combining with edge variation (RCEV) codes allow the addition and deletion of edges as rows are combined so the degree-distributions can be different for different rates. The key to maintaining a simplified decoder architecture is to make the number of additions and deletions small compared to the total number of edges in the graph.

One of the most critical differences in the optimal variable-node degree distribution for different rates, is the number of degree-two variable nodes. In order to have good error floor properties the number of degree-two variable nodes cannot exceed the number of check nodes as shown in [16]. Having

more degree-two nodes than check nodes implies that there will be cycles composed by only degree-two nodes and check-nodes. These cycles are stopping sets and have been shown to degrade the performance of the codes [14]. These cycles will grow smaller and more numerous as the number of degree-two nodes increases, further worsening the performance of the codes.

As a result, the maximum number of degree-two variable nodes for a family of SRC codes is given by the number of check nodes of the highest-rate effective code. This limits the performance of the lower rate codes since their optimal degree-distribution generally requires a significantly larger number of degree-two variable nodes [13]. The difference in the distributions depends on the rates of both the mother code and all the effective codes. The loss in performance due to this limitation increases as the range of possible rates of the SRC codes grows larger.

In order to avoid this problem in RCEV codes, the number of degree-two variable nodes in the mother code is set to be the optimal for the lowest-rate code. For high-rate effective codes, edges are added to some degree-two variable nodes so that the maximum number of degree-two nodes is less than or equal to the number of check nodes in the graph. Therefore, when generating the effective code E_i , there is a set of degree-two variable nodes (S^{E_i}) that have degree 3 in the code E_i .

This, unfortunately, is not the only problem generated by the common degree-distribution. Layered Belief Propagation (LBP), a decoding method that improves the convergence speed and allows a low complexity hardware architecture, was introduced in [17] and [18]. LBP decoding requires the parity-check matrix to be divided into sub-matrices that can have at most one "1" per column and one "1" per row. Thus, in order to generate an LDPC code that can be decoded with LBP, the maximum variable-node degree is the number of sub-matrices in a column, which is the number check nodes divided by the size of the sub-matrices. For SRC codes that support LBP, the maximum variable-node degree must be the same for all rates, and it is given by the number of check nodes of the highest rate code divided by the size of the sub-matrices. As stated in [13], increasing the maximum variable-node degree results in a better code. The SRC common degree-distribution imposes a strict limit on the maximum variable-node degree, and thus a performance limit when LBP decoding is used.

The technique used in RCEV codes to avoid this problem is the following. If during the row combining, a non-zero sub-matrix is added to another non-zero sub-matrix, one of them is discarded in order maintain the structure mentioned before. This allows RCEV codes to have different maximum variable-node degrees for different rates which will improve the performance of the lower rate codes.

RCEV codes are designed using the steps for SRC codes given in Section V, along with the edge variation techniques described in this section as shown in Algorithm 2.

VII. PERFORMANCE COMPARISON

The following figures show the performance of row-combining codes, designed using all the techniques presented above, on the AWGN channel with a flooding LDPC decoder.

Algorithm 2 RCEV code design

```

1: Choose rates, blocklength ( $n$ ) and sub-matrix size ( $p$ )
2: Choose variable node degree distributions
3: Choose the variable-node sets  $S^{C_i}$ 
4: Choose rows of sub-matrices to be combined in order to
   generate the higher-rate codes
5: for all columns of sub-matrices  $V_j$  do
6:   Randomly generate  $V_j^M$  according to the degree distri-
   bution of the lowest-rate code
7:   if  $V_j^M$  doesn't satisfy the graph constraints set for  $M$ 
   then
8:     Discard  $V_j$  and go to line 6
9:   end if
10:  for all effective codes  $E_i$  do
11:    Compute  $V_j^{E_i}$  according to the row combinations
    selected in 4
12:    if two non-zero sub-matrices are combined then
13:      if The degree of  $V_j^M$  is less than maximum
      variable-node degree then
14:        Discard  $V_j$  and go to line 6
15:      else
16:        Discard one of the non-zero sub-matrices
17:      end if
18:    end if
19:    if  $V_j^{E_i}$  doesn't satisfy the graph constraints set for
     $E_i$  then
20:      Discard  $V_j$  and go to line 6
21:    end if
22:    if  $V_j \in S^{E_i}$  then
23:      randomly add a non-zero sub-matrix to  $V_j^{E_i}$ 
24:    end if
25:  end for
26: end for

```

Fig. 3 shows the performance of a structured SRC code family and the corresponding four stand-alone codes selected for the IEEE 802.11n standard [19]. The specifications of the structured SRC code family are the following. The mother code has rate 0, while the four effective codes have rates 1/2, 2/3, 3/4, and 5/6. The blocklength of the codes is 1944 bits, the size of the sub-matrices is 27x27, a maximum of 15 iterations was used in the simulation. It has at most one "1" per column and one "1" per row given and therefore allows the simplified decoder proposed in [17].

Fig. 3 shows that the SRC codes perform very well under these conditions (blocklength, sub-matrix size, and number of iterations). The sub-matrix size is small enough to allow a sufficiently large maximum variable-node degree. The effect of the non-optimality of the variable-node degree distribution for the rate-1/2 code is small for 15 iterations. This is because for such a small number of iterations performance is dominated by the high-degree variable nodes. Thus, the lower-than-optimal number of degree-two nodes of the rate-1/2 code does not dramatically affect performance. For fifty iterations, the rate-1/2 code performance is affected, as we'll see shortly.

This comparison at 15 iterations is practically important since there are many applications that only allow a small number of iterations of the decoder. For example, in most

TABLE I
VARIABLE-NODE DEGREE DISTRIBUTIONS OF STRUCTURED SRC AND RCEV CODES

SRC $p=27$		SRC $p=54$		RCEV $R=1/2$		RCEV $R=2/3$		RCEV $R=3/4$		RCEV $R=5/6$	
x	λ_x	x	λ_x	x	λ_x	x	λ_x	x	λ_x	x	λ_x
1	.00014	1	.00014	1	.00014	1	.00015	1	.00014	1	.00015
2	.09496	2	.09496	2	.2554	2	.19021	2	.13308	2	.09971
3	.57151	3	.52389	3	.29327	3	.42863	3	.46673	3	.60009
7	.33338	6	.38101	10	.45119	8	.38101	9	.40005	6	.30005

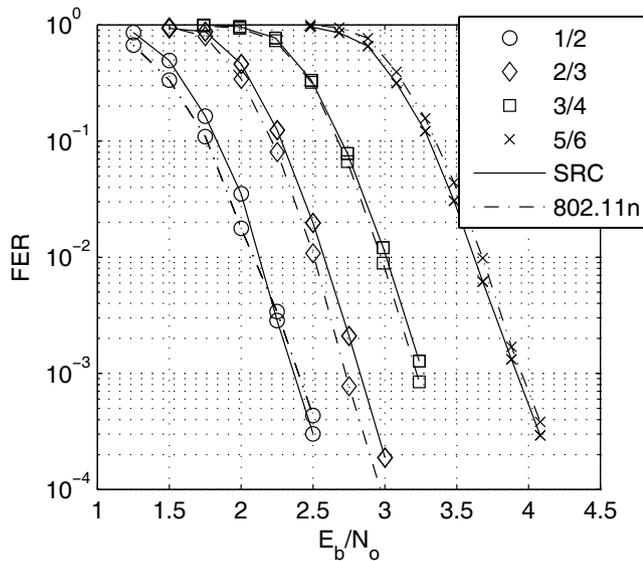


Fig. 3. Performance of structured SRC with $p=27$ and IEEE 802.11n codes with blocklength 1944 on an AWGN channel. Maximum number of iterations equal to 15.

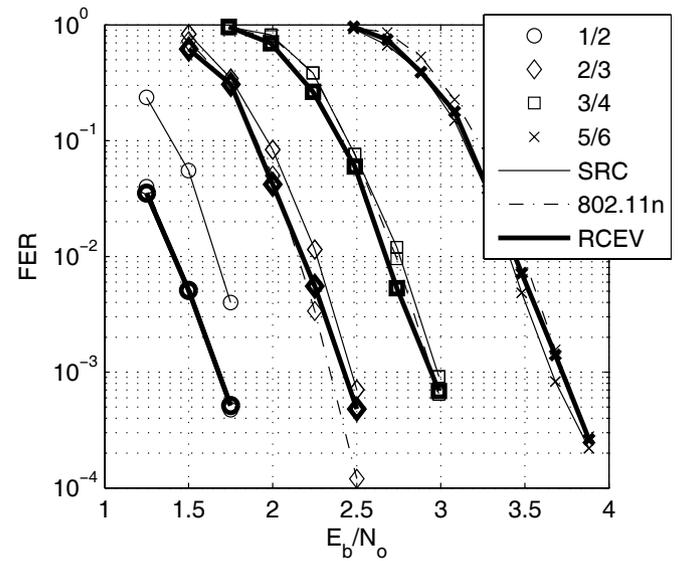


Fig. 4. Performance of structured SRC with $p=54$, structured RCEV, and IEEE 802.11n codes with blocklength 1944 on an AWGN channel. Maximum number of iterations equal to 50.

wireless applications the channel decoding must be done in a very small amount of time, which limits the maximum number of iterations allowed.

Fig. 4 presents another comparison of the performance of a structured SRC code, the four stand-alone codes selected for the IEEE 802.11n standard, and a structured RCEV code. The row-combining codes have a mother code with rate 0, while the four effective codes have rates 1/2, 2/3, 3/4, and 5/6. The variable-node degree distribution of both codes can be found in Table I. The blocklength of the codes is 1944 bits, the size of the sub-matrices of the row-combining codes is 54x54 and a maximum of 50 iterations was used in these simulations. SRC and RCEV codes of the same rate have similar total number of edges. The maximum difference, found in the rate-3/4 code, is a 7%.

As expected, Fig. 4 shows the loss in performance of the SRC low rate codes. The SRC rate-1/2 code show a performance loss of more than 0.2 dB with respect to the IEEE 802.11n codes with the same blocklength and this is due to their inadequate variable-node degree distribution.

As observed in Fig. 4 the FER of the lower rates of the RCEV code are significantly better than those of the lower rates of the SRC code. This gain follows from the improved degree distributions of the RCEV codes over those of the SRC codes. The magnitude of the difference between the degree distributions of the rate-1/2 codes can be seen in Table I.

For the high rate codes there is very little difference between the performances of the RCEV and SRC codes, since their degree distributions are very similar. RCEV techniques allow the optimization of the degree distributions for each rate.

Fig. 4 also shows that the RCEV codes perform close to the IEEE 802.11n codes. As expected there is a small loss due to the fact that RCEV codes must satisfy the row-combining constraints. This slightly reduces the degrees of freedom during the design process, thus making the set of all the possible RCEV codes a subset of all the possible stand-alone codes. Table II shows the graph-conditioning constraints used to generate these row-combining codes and a detailed description of them can be found in [20].

As mentioned before, the row-combining codes presented in this section have a rate-0 mother code. We previously designed codes that had a rate-1/2 mother code with the same variable-node degree distribution [21]. This mother code forces the effective rate-2/3 code to have a non-concentrated check-node degree distribution. This translates into a 0.3 dB loss at a FER of 10^{-3} in the performance of the rate-2/3 code when compared to the 802.11n rate 2/3 code. Having a rate-0 mother code reduces the loss to 0.1 dB as shown in Fig. 3.

A comparison in the performance of RCEV codes and punctured LDPC codes is shown in Fig. 5. The punctured codes correspond to the ones presented in [3]. The RCEV code was designed to have very similar degree distributions

TABLE II
GRAPH-CONDITIONING CONSTRAINTS OF STRUCTURED SRC AND RCEV CODES

Code	SRC $p=27$				SRC $p=54$				RCEV			
Rate	1/2	2/3	3/4	5/6	1/2	2/3	3/4	5/6	1/2	2/3	3/4	5/6
d_{ACE}	10	2	3	3	10	3	3	2	6	2	3	2
η_{ACE}	3	4	3	3	3	3	3	4	3	4	3	4
d_{SS}	4	4	4	4	4	4	4	4	4	4	4	4
γ_c	4	4	4	3	3	3	3	3	4	4	4	3
γ_p	4	4	4	3	3	3	3	3	4	4	4	3

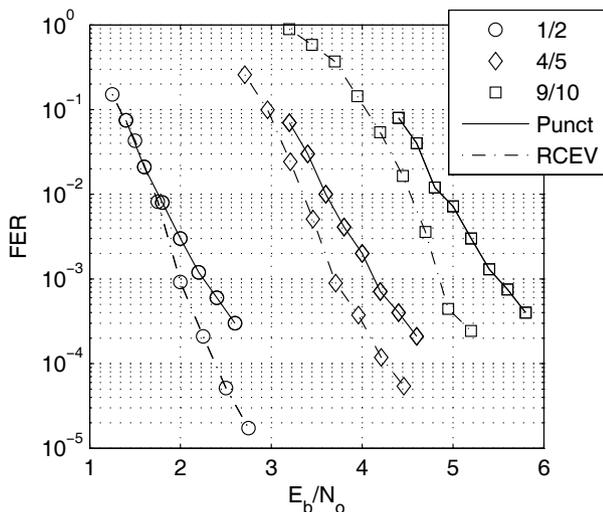


Fig. 5. Performance of RCEV codes and punctured LDPC codes.

to the punctured codes so that the comparison would be fair. The blocklength of the mother code of the punctured codes is 1024 while the blocklength of the RCEV codes is 1030. There is a clear performance gap between the higher rate codes. This is due to the fact that puncturing reduces the effective blocklength of the code.

Sphere-packing bounds [22] give a lower bound on the SNR that can support reliable communication as a function of blocklength. The gap between the sphere-packing bounds of several rate-9/10 row-combining code and rate-9/10 punctured codes with equal mother-code blocklength (based on the different blocklengths) can be seen in Fig. 6. The loss in performance of the punctured codes decreases as the mother-code blocklength increases. However, the theoretical gap at a blocklength of 1030 is half of the performance gap shown in Fig. 5. That punctured codes aren't always the best codes for their effective blocklength may help to explain why there is a bigger gap in the actual performance of the codes.

VIII. CONCLUSIONS

As we know from information theory, channel codes approach capacity-achieving performance as blocklength goes to infinity. Both theory and practice confirm that codes with longer blocklengths perform better. Recently, puncturing and shortening have been used to provide a variety of rates in the context of a single decoder architecture, but these techniques shorten the code blocklength as rates move away from the rate of the mother code.

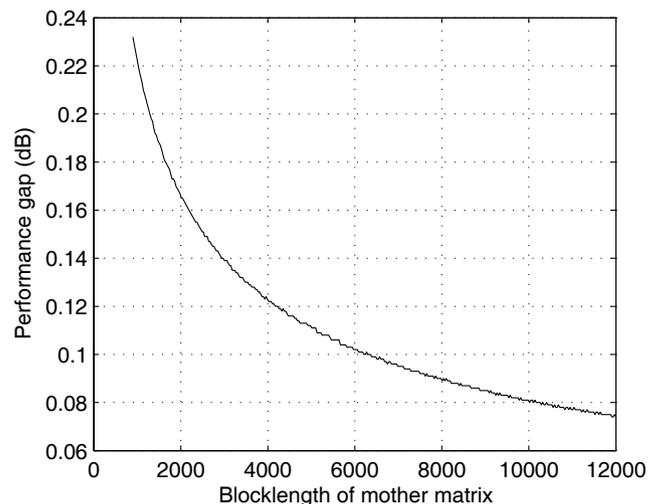


Fig. 6. Sphere-packing bound gap at a FER of 10^{-3} between row-combining codes and punctured codes. Both codes have the same rate (9/10) and have equal mother-code blocklength. The mother code of the punctured codes has rate 1/2.

Multiple-rate LDPC codes that avoid this blocklength reduction can be generated using a row-combining approach. Structures that reduce complexity of both encoder and decoder can be applied and maintained through all the rates, and graph conditioning algorithms can be applied to the design of the codes to produce good error floor performances at all the rates.

Structured SRC codes allow the use of simple hardware architectures (or programmable analog-decoding circuits) that can be used to decode the mother and all the effective rates. The design of these codes imposes some constraints that barely affect the performance when the codes are used at a small number of iterations. As the maximum number of iterations increases, the performance gap between SRC codes and stand-alone codes also increases.

RCEV codes relax the constraints required for SRC codes, and show a gain in performance with respect to SRC codes for a large number of iterations. This increase in performance comes at a cost of architecture complexity. The performance of RCEV codes is very close to the performance of stand-alone codes for a large number of iterations.

These codes become attractive for applications that require both performance close to capacity and a low decoder complexity due to the fact that RCEV codes have still a strong inherent structure that can be exploited in the hardware decoder.

ACKNOWLEDGMENT

The authors would like to thank Dr. Jun Shi at Intel for his help with the sphere-packing bounds.

REFERENCES

- [1] J. Ha and S. W. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2824–2836, Nov. 2004.
- [2] T. Tian, C. Jones, and J. Villasenor, "Rate-compatible low-density parity-check codes," in *Proc. IEEE ISIT*, July 2004, p. 152.
- [3] J. Kim, A. Ramamoorthy, and S. W. McLaughlin, "Design of efficiently-encodable rate-compatible irregular LDPC codes," in *Proc. IEEE ICC*, June 2006, pp. 1131–1136.
- [4] M. Rovini, N. E. L'Insalata, F. Rossi, and L. Fanucci, "VLSI design of high-throughput multi-rate decoder for structured LDPC codes," in *Proc. 8th Euromicro Conf. Digital System Design*, Aug. 2005, pp. 202–209.
- [5] J. Hagenauer, "Decoding of binary codes with analog networks," in *Proc. Information Theory Workshop*, Feb. 1998, pp. 13–14.
- [6] H.-A. Loeliger, F. Lustenberger, M. Helfenstein, and F. Tarkóy, "Probability propagation and decoding in analog VLSI," in *Proc. IEEE Int. Symp. Inform. Theory*, page 146, Aug. 1998, p. 146.
- [7] F. Lustenberger, "On the Design of Analog VLSI Iterative Decoders," PhD thesis, diss., ETH No 13879, Nov. 2000.
- [8] M. M. Mansour and N. R. Shanbhag, "Low power VLSI decoder architectures for LDPCs," *Int. Symp. Low Power Electronics and Design*, pp. 284–289, June 2002.
- [9] M. Eroz, F.-W. Sun, and L.-N. Lee, "An innovative low-density parity-check code design with near-Shannon-limit performance and simple implementation," *IEEE Trans. Commun.*, vol. 54, no. 1, Jan. 2006.
- [10] V. Novichkov, H. Jin, and T. Richardson, "Programmable vector processor architecture for irregular LDPC codes," in *Proc. Conf. Inform. Systems Sciences*, Mar. 2004, pp. 1141–1146.
- [11] M. Yang, W. E. Ryan, and Y. Li, "Design of efficiently encodable moderate-length high-rate irregular LDPC codes," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 564–571, Apr. 2004.
- [12] T. Richardson and R. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 638–656, Feb. 2001.
- [13] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, Feb. 2001.
- [14] T. Tian, C. Jones, J. Villasenor, and R. Wesel, "Avoidance of cycles in irregular LDPC construction," *IEEE Trans. Commun.*, vol. 52, no. 8, pp. 1242–1247, Aug. 2004.
- [15] A. Ramamoorthy and R. D. Wesel, "Construction of short block length irregular LDPCs," in *Proc. IEEE ICC*, June 2004, pp. 410–414.
- [16] W. Weng, A. Ramamoorthy, and R. Wesel, "Lowering the error floors of irregular high-rate ldpc codes by graph conditioning," in *Proc. IEEE VTC-Fall*, Sep. 2004, pp. 2549–2553.
- [17] M. M. Mansour and N. R. Shanbhag, "High-throughput ldpc decoders," *IEEE Trans. VLSI Syst.*, vol. 11, pp. 976–996, Dec. 2003.
- [18] D. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," in *Proc. Signal Processing Systems (SIPS)*, Oct. 2004, pp. 107–112.
- [19] IEEE P802.11n/D1.05 October 2006, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Enhancements for Higher Throughput (Draft).
- [20] [Online]. Available: <http://www.ee.ucla.edu/~csl/files/rccodes.html>

- [21] A. I. Vila Casado, W. Weng, and R. D. Wesel, "Multiple rate low-density parity-check codes with constant block length," in *Proc. Asilomar Conf. Signals, Systems Computers*, Nov. 2004, pp. 2010–2014.
- [22] C. E. Shannon, "Probability of error for optimal codes in a gaussian channel," *Bell System Technical J.*, vol. 38, pp. 611–656, May 1959.



RFID applications.



802.11n standard where he contributed to channel model definition, the PHY emulation for MAC simulations and LDPC code definition. His research interests are digital transmission systems, mainly channel coding, modulation and estimation.



Okawa Foundation award for research in information and telecommunications, and the Excellence in Teaching Award from the Henry Samueli School of Engineering and Applied Science. He has authored or co-authored over a hundred conference and journal publications.

Andres I. Vila Casado received his B. S. in electrical engineering from the Politecnico di Torino, Turin, Italy in 2002. He received his M. S. and Ph. D. degrees in electrical engineering from the University of California, Los Angeles in 2004 and 2007 respectively. At UCLA and at Politecnico di Torino he conducted research on communication theory with a focus on channel coding and information theory. He is currently a Research Scientist at Mojix, Inc. where he conducts research on physical layer communications and Bayesian estimation for

Wen-Yen Weng received his B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan in 1997. He received his M.S. and Ph.D. in electrical engineering from the University of California, Los Angeles, in 2001 and 2007 respectively. He is currently an Assistant Professor in the Computer Science and Information Engineering Department in Chung-Hua University, Hsin-Chu, Taiwan. His research interests include channel coding and signal processing for communication systems.

Stefano Valle was born in Milan (Italy) on 1969. He received the Electrical Engineering degree (1995) and the Electrical and Communications Engineering PhD (1999) from Politecnico di Milano, Milan, Italy. Until 2000 he was with the Digital Signal Processing group of Dipartimento di Elettronica e Informazione, Politecnico di Milano as a Research Assistant. In 2000, he joined STMicroelectronics (Italy), where he worked on modulation, channel modeling and error correction codes for wireless systems. He was voting member during the definition of the IEEE

Richard D. Wesel is a Professor with the UCLA Electrical Engineering Department and is the Associate Dean for Academic and Student Affairs for the UCLA Henry Samueli School of Engineering and Applied Science. He joined UCLA in 1996 after receiving his Ph.D. in electrical engineering from Stanford. His B.S. and M.S. degrees in electrical engineering are from MIT. His research is in the area of communication theory with particular interest in channel coding. He has received the National Science Foundation (NSF) CAREER Award, an