C30-20070315-003

Title:	LDPC Coding Proposal for LBC
Abstract:	This contribution provides an LDPC coding proposal for LBC
Source:	Alcatel-Lucent, Huawei, LG Electronics, QUALCOMM Incorporated, RITT, Samsung, ZTE
Contact:	Noah Jacobsen, Anthony Soong, Ji Wook Chung, Tom Richardson, Aamod Khandekar, Sung-Eun Park, Jun Xu, Xin Yu jacobsen@alcatel-lucent.com, asoong@huawei.com, jwchung@lge.com, [tomr,aamodk]@qualcomm.com, se.park@samsung.com, [xin.yu,xu.jun2]@zte.com.cn
Date:	March 15, 2007
Recommendation:	Review and adopt

Notice

Contributors grant a free, irrevocable license to 3GPP2 and its Organizational Partners to incorporate text or other copyrightable material contained in the contribution and any modifications thereof in the creation of 3GPP2 publications; to copyright and sell in Organizational Partner's name any Organizational Partner's standards publication even though it may include all or portions of this contribution; and at the Organizational Partner's sole discretion to permit others to reproduce in whole or in part such contribution or the resulting Organizational Partner's standards publication. Contributors are also willing to grant licenses under such contributor copyrights to third parties on reasonable, non-discriminatory terms and conditions for purpose of practicing an Organizational Partner's standard which incorporates this contribution.

This document has been prepared by contributors to assist the development of specifications by 3GPP2. It is proposed to the Committee as a basis for discussion and is not to be construed as a binding proposal on contributors. Contributors specifically reserve the right to amend or modify the material contained herein and nothing herein shall be construed as conferring or offering licenses or rights with respect to any intellectual property of contributors other than provided in the copyright statement above.

Introduction to LDPC Codes

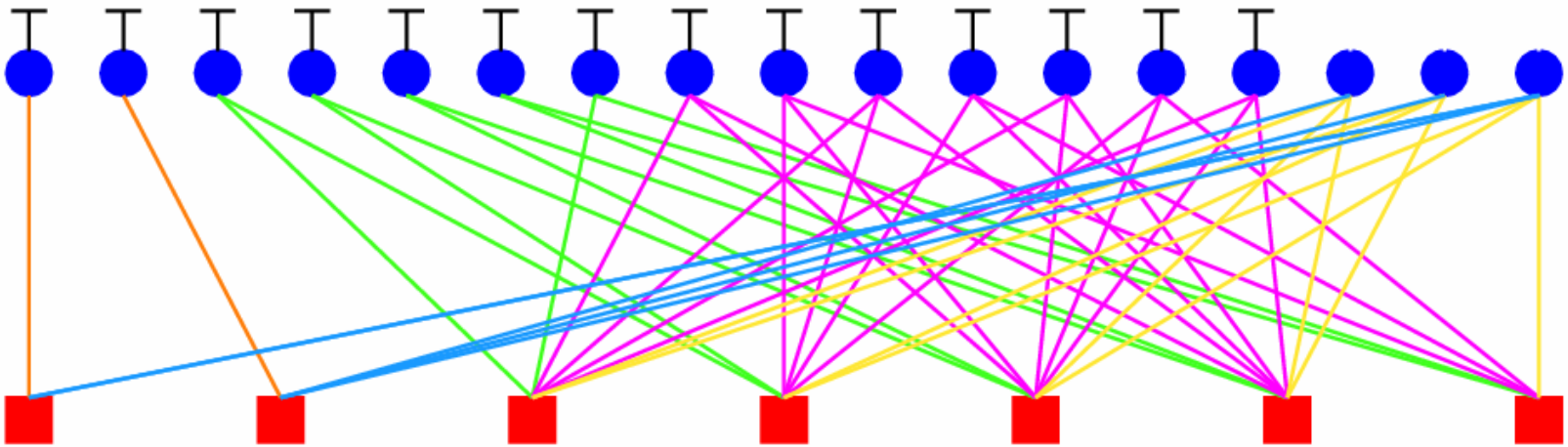
- LDPC (Low Density Parity Check) codes are defined by a sparse parity check matrix
  - Parity check matrix consists of a set of linear constraints on the transmitted bits
- Can also be represented in the form of a Tanner graph
  - Bipartite graph consisting of variable nodes and parity check nodes
  - Variable nodes represent bits in the codeword
    - In addition, can also contain some “punctured” bits that are not transmitted on the channel
  - Check nodes represent the constraints that define the code
    - The bits corresponding to the variable nodes that are connected to any given check node are constrained to sum to 0
    - A sparse parity check matrix corresponds to a sparse tanner graph
- Belief-propagation decoding
  - Consists of passing beliefs, in the form of log-likelihood ratios, along the edges of the Tanner graph
  - In each iteration, pass LLRs once from variable nodes to check nodes, and once from check nodes to variable nodes
  - Complexity of belief propagation is proportional to the number of edges in the graph

Multi-Edge-Type LDPC codes

- Generalization of irregular LDPC – richer language for describing structure.
- Structures optimized using density evolution.
- Each edge type carries distinct density.
- Sophistication of structure does not induce complexity in implementation – in fact the opposite is true.
- Multi-edge-type framework used for designing base graph

Tanner graphs for multi-edge type LDPC

- Tanner graph for multi-edge type.
- Color indicates edge type.
- Dongles indicate transmitted bits.
- Variable nodes and check nodes.

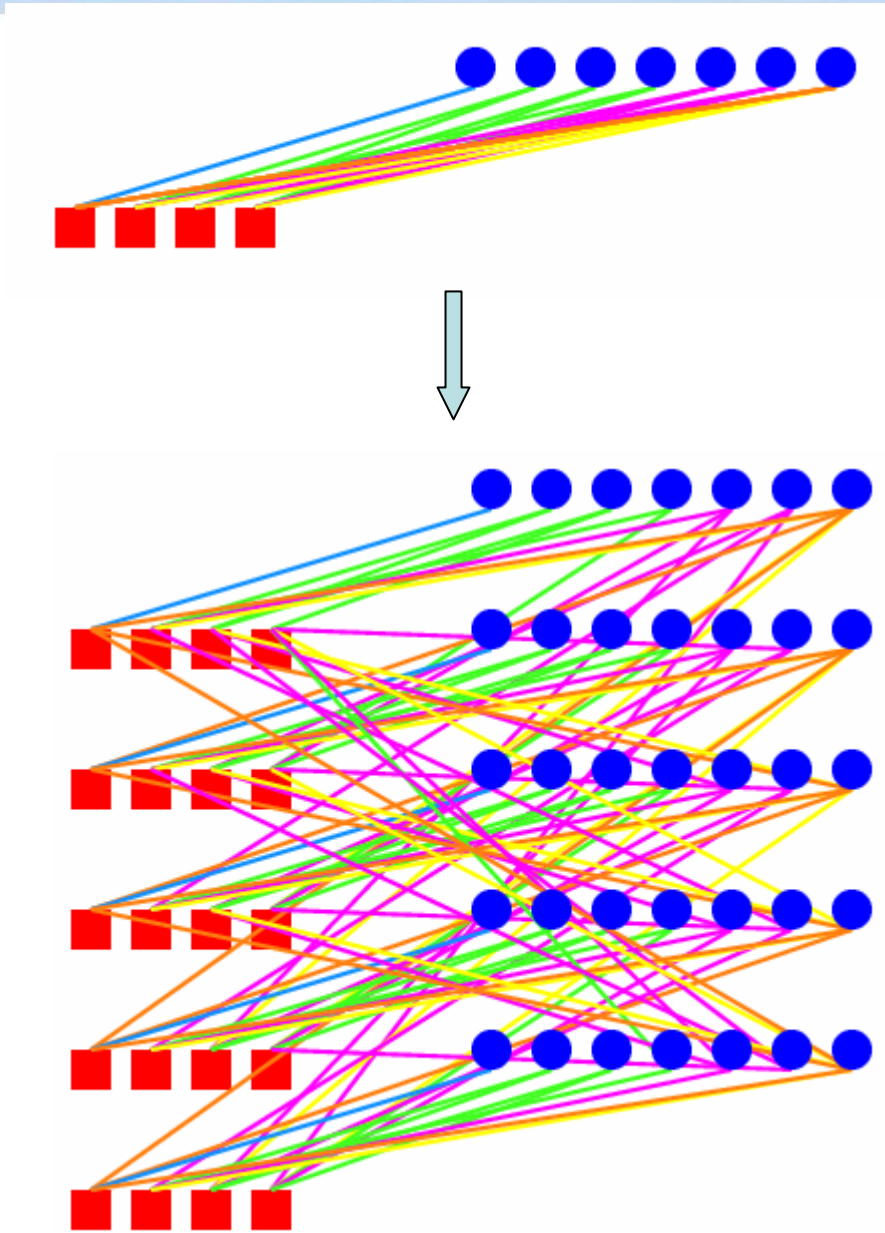


Matched-Lifted LDPC Codes

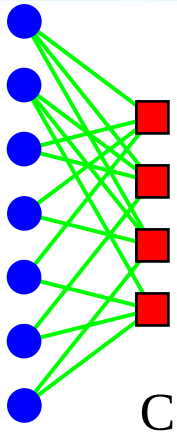
- Lifting is a technique to generate a large LDPC code from several copies of a small "base code"
  - Enables implementation of parallel encoding and decoding algorithms
  - Reduces description complexity for the large code
- Start with a small parity check matrix, say of size (n_0,k_0)
  - This code is called the base code
- To get a lifting of size L:
  - Replace each non-zero entry in the parity check matrix by a LxL permutation matrix: each edge in the Tanner graph with L edges
  - Yields a code of size (Ln_0,Lk_0) with L times the number of edges in the base code
- Matched Lifting
  - The different LxL permutation matrices are chosen from a group of order L
    - A simple and common choice is the cyclic group (i.e., cyclic shift matrices).
  - Alternative description
    - Generate L copies of the base code graph
    - Index the copies of the base graph using a group G of order L
    - Associate an element g_e to each lifted bundle of L edges. Connect variable node in copy g to check node in copy g_e g
    - Choosing a cyclic group would be equivalent to using cyclic shift matrices in the earlier description

Lifting

Example uses cyclic lifting.



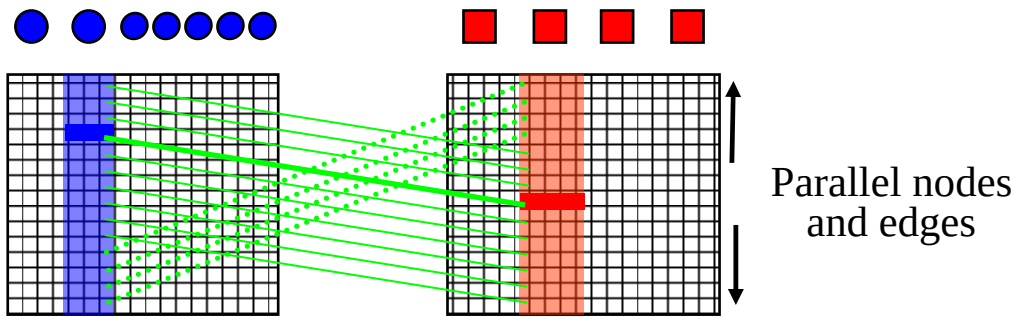
Liftings of LDPC Codes



$$\equiv \quad H = \begin{bmatrix} 1&0&1&1&1&0&0 \\ 1&1&1&0&0&1&0 \\ 1&1&0&1&0&0&1 \\ 0&1&0&0&1&1&1 \end{bmatrix} \quad \Rightarrow \quad \mathbf{H} = \begin{bmatrix} \sigma_1 & 0 & \sigma_7 & \sigma_9 & \sigma_{11} & 0 & 0 \\ \sigma_2 & \sigma_4 & \sigma_8 & 0 & 0 & \sigma_{13} & 0 \\ \sigma_3 & \sigma_5 & 0 & \sigma_{10} & 0 & 0 & \sigma_{16} \\ 0 & \sigma_6 & 0 & 0 & \sigma_{12} & \sigma_{14} & \sigma_{15} \end{bmatrix}$$

Construct a small graph and replace each element of the parity check matrix with a k x k matrix.

Edges (1s) are replaced with (e.g. cyclic) permutation matrices.



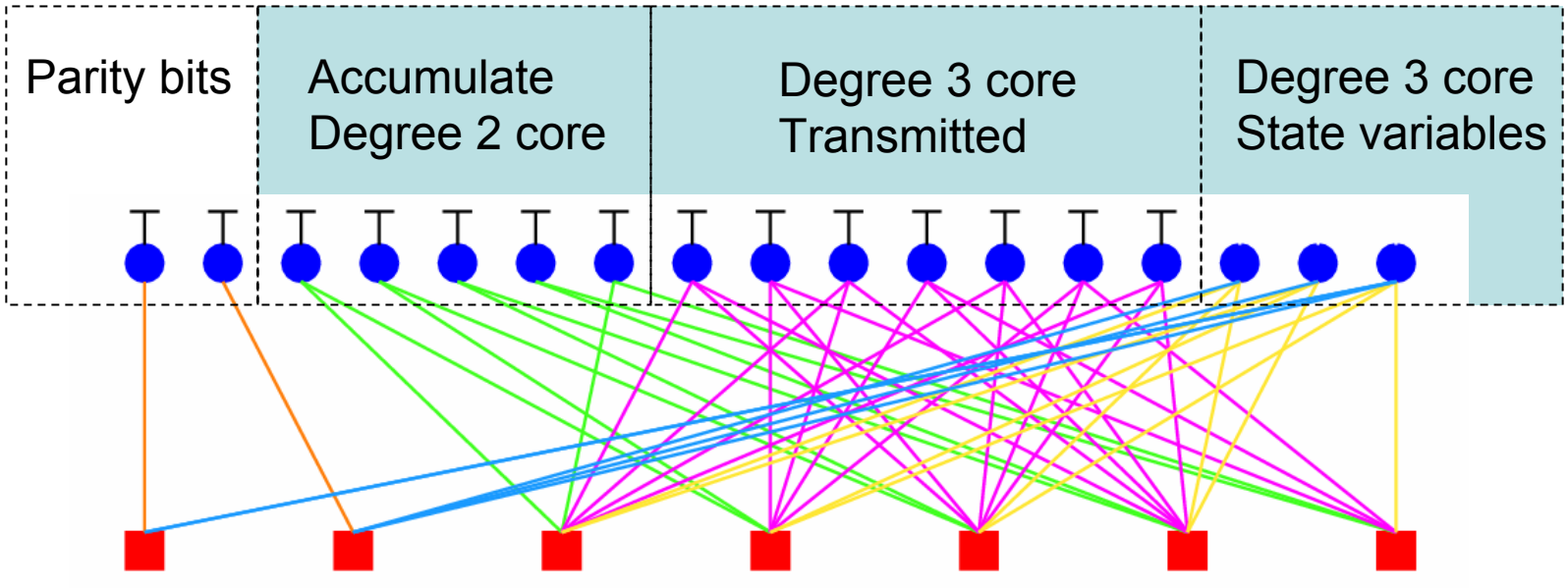
- Compact representation of graph.
- Simple, parallel encoding.
- Efficient parallel implementation of decoder.

Parallelizing Matched Lifted LDPC Codes

- Lifted LDPC codes enable parallel encoder and decoder implementations in various forms.
- Graph descriptions are compact – fundamental complexity reduction.
- Edge parallel decoder implementation:
  - Edges in the base graph are processed in a serial fashion
  - Parallelism achieved by simultaneously processing L copies of the same edge
  - Requires implementation of group permutations in a switching network
- Node parallel decoder implementation:
  - Different copies of the base graph are processed in a serial fashion
  - Parallelism achieved by simultaneously processing different nodes in the base graph
- We propose to use cyclic liftings (i.e., use of cyclic permutation matrices) with the lifting size being restricted to powers of 2
  - Cyclic liftings can be easily implemented using a counting operation, especially for the node-parallel approach
  - Restriction on the lifting size ensures that all the different lifting sizes have a large common factor
    - Important property for an edge-parallel implementation
- Proposed structure thus provides efficient support for both node-parallel and edge-parallel implementations.

Typical base structure

High rate example: low rates have many more parity bits.



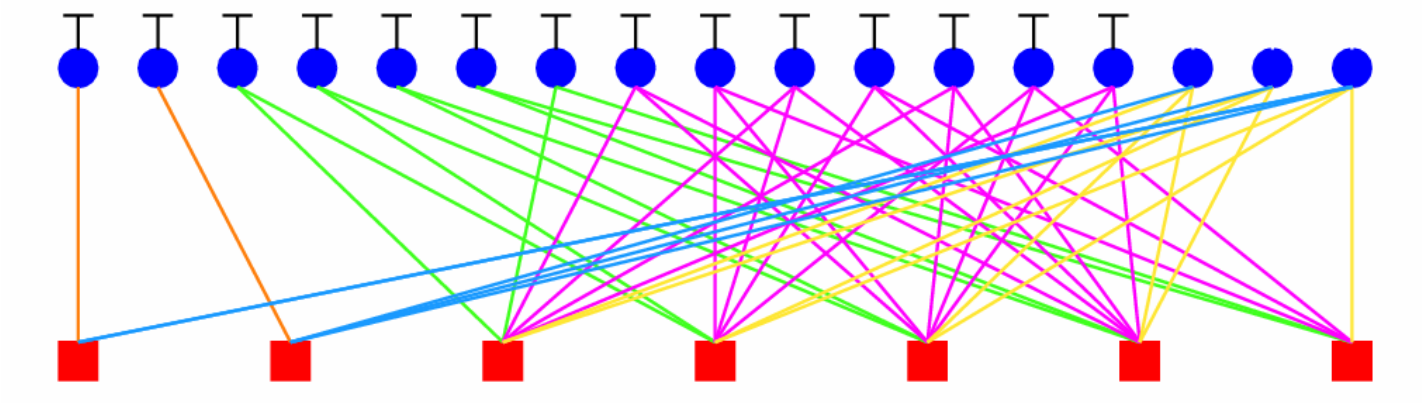
Features of the multi-edge LDPC codes used

- Some state variable (punctured) nodes.
  - Gives higher thresholds with lower degrees: performance/complexity improvement.
  - Provably superior on the BEC.
- Degree 2 accumulate chain. Structure in common with IRA codes. Also called dual diagonal structure. Facilitates simple encoding.
- Degree 1 variable nodes. Parity bits optimized for threshold considerations, very convenient for HARQ. Very simple for encoding.
- Core graph consists of degree 3, punctured degree 3 and degree 2 accumulate chain.

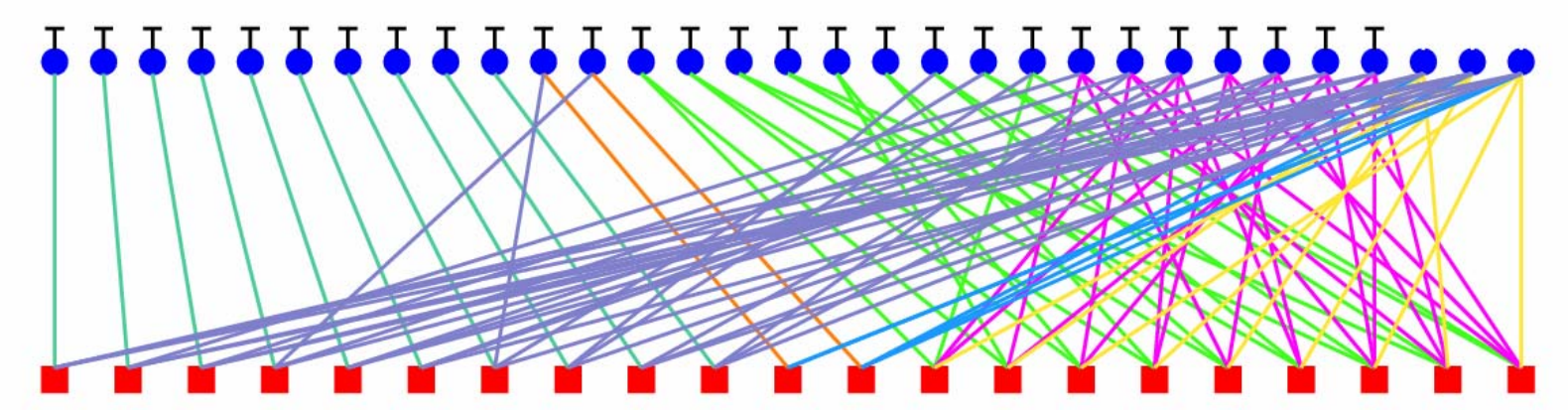
HARQ Design

- Higher rate codes achieved by puncturing parity bits and some degree 2 bits from lower rate codes.
- Punctured variable nodes are of two types
  - Parity bits: variable node of degree 1 connected to a new check node of arbitrary degree
  - Degree 2 variable nodes: Puncturing these can be viewed as shortening the (dual diagonal) accumulate chain by merging of check nodes.
- Resort to repetitions when E_b/N_0 gains achievable by adding more redundancy are negligible and to limit memory requirements
- This design ensures that extra complexity is not incurred at the initial transmission (in terms of the number of edges) because of incremental redundancy
  - Hardware need not process edges associated to punctured nodes, saving computation.

Example of HARQ extension



Tanner graphs for first (14,10) transmission and HARQ extension to (28,10). Note lengthening of degree 2 accumulate chain and additional parity bits.

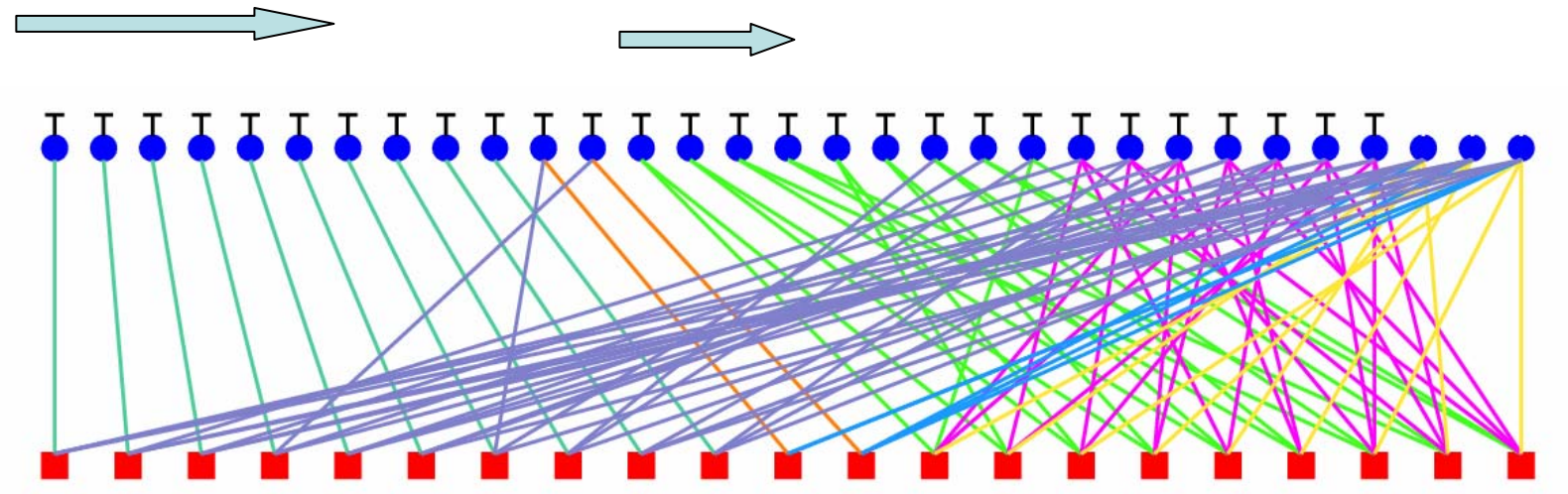


Example of HARQ extension

- This figure shows how a higher rate code is generated from the lower rate code
- Example shown is the same as on the previous slide, except that it shows how to go from the lower-rate code to the higher rate code

Puncture here to get higher rates.

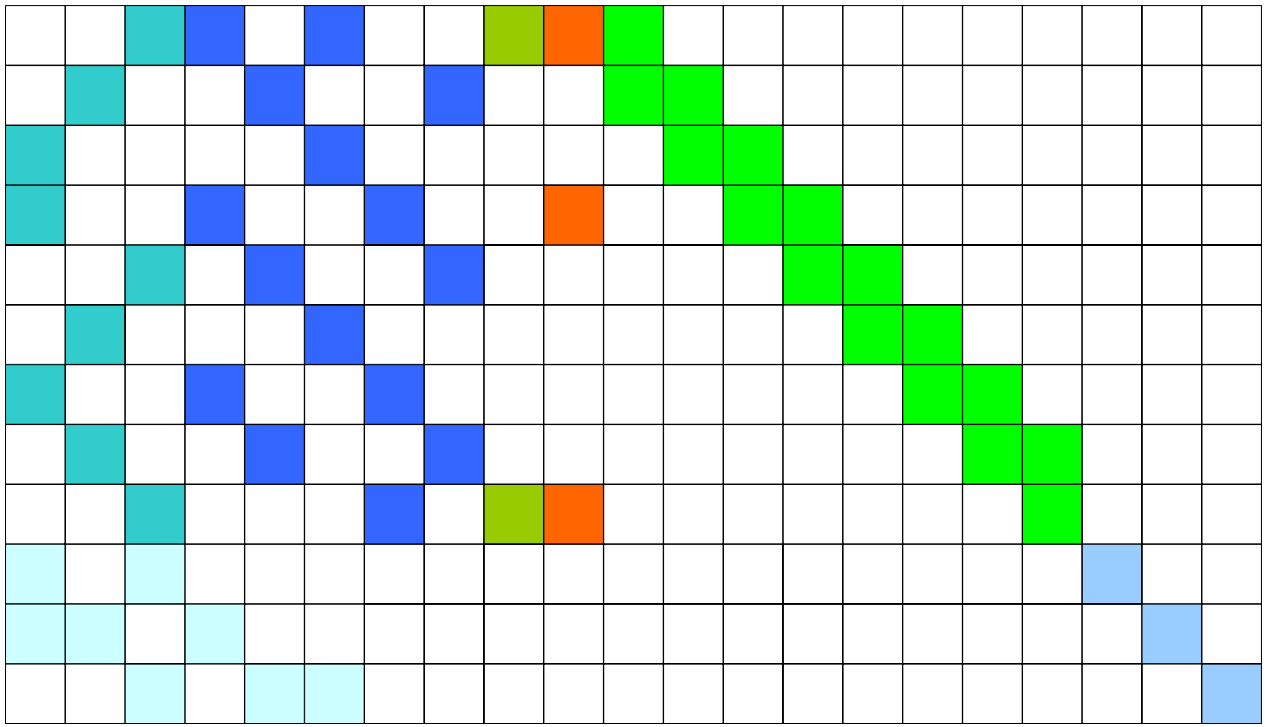
At certain particular lengths, puncture here.



Bit Transmission Order

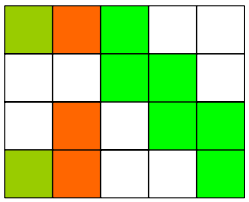
- Information bits are first converted to coded bits using the lowest rate code
  - Coded bits consists of information bits as well as all defined parity bits
  - Parity bits consist of degree two variable nodes (accumulator chain) as well as degree one variable nodes (parity checks)
- The coded bits generated by this operation are then permuted
  - The order of systematic bits is unchanged and they stay before the parity bits
  - Permutation changes only the order of the parity bits
  - Parity check matrices will be presented in bit transmission order.
- HARQ transmissions are made in order from the permuted coded bit stream
  - First transmission starts with the systematic bits
  - Subsequent HARQ retransmissions begin with the bits in the buffer immediately after the position where the previous HARQ retransmission ended
  - Permutation operation enables transmission of some degree one variable nodes before all the degree two variable nodes are transmitted, as described in the previous slide

Representative base matrix structure



Transmission proceeds left to right except for some reordering of non-information columns.

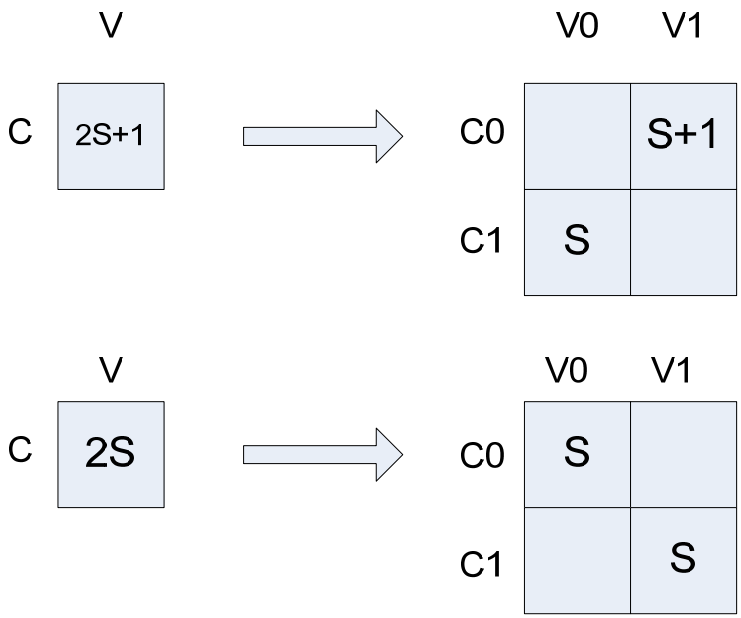
Encoding structure



n+1 x n matrix of dual diagonal structure of length X, loop-closing degree 2 variable node, and loop-closing degree 3 variable node provide basis for encoding. Encoding structure may appear reordered.

Given a base graph G and a cyclic lifting of size Z where Z is even there is an equivalent representation of the complete graph as a base graph G' of twice the size and a cyclic lifting of size Z/2. For each node in G there are two nodes in G': an even and an odd node. Consider a variable node v in G connected to constraint node c in G. In G' this gives rise to nodes v0,v1 and c0,c1. The Z/2 lifted copies of v0 and c0 correspond to the even copies of v and c respectively in the Z-lifting of G. If a permutation S in the lifting of G connecting v to c is even then v0 connects to c0 and v1 connects to c1 in G'. If S is odd then v0 connects to c1 and v1 connects to c0.

Expansion of Lifted Parity check matrix



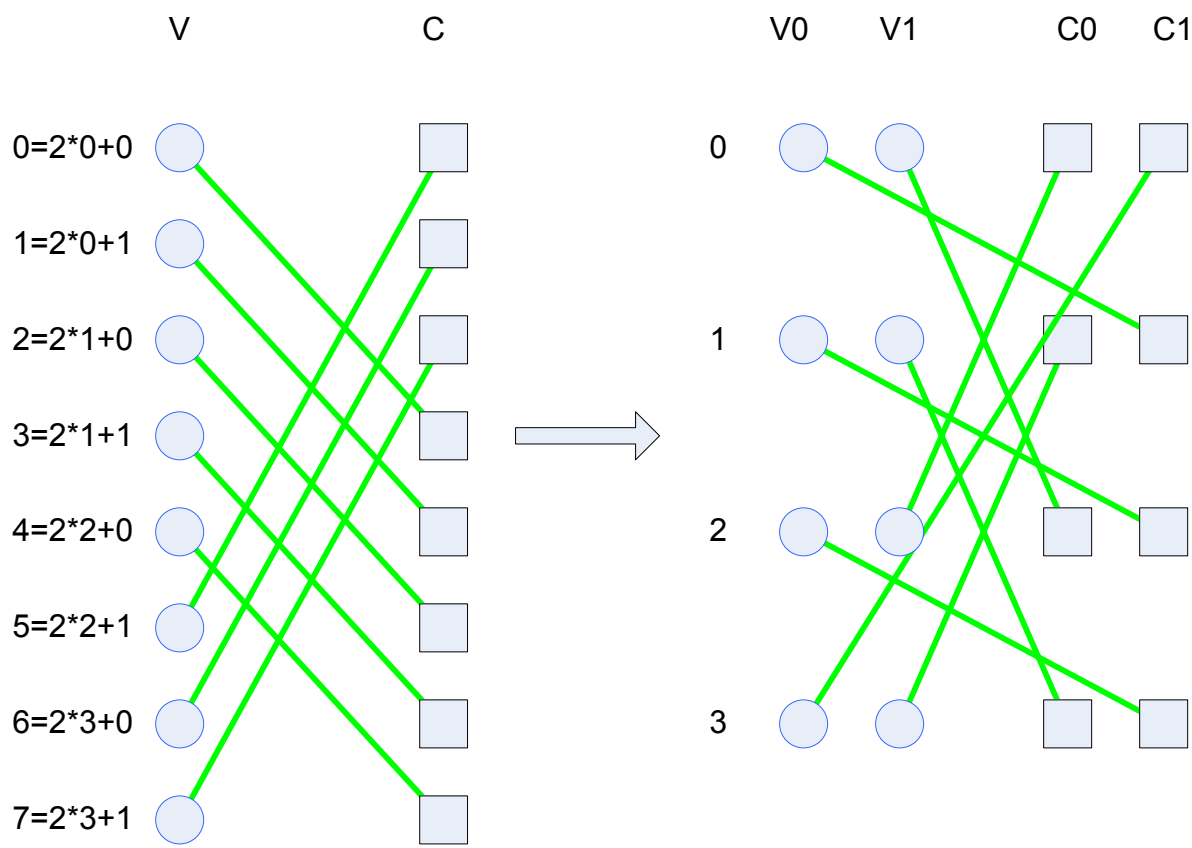
Each matrix entry becomes 2x2 block. Odd lifting of size 2S+1 becomes cross-diagonal with S and S+1 as shown, and even 2S becomes diagonal S and S.

Non-edge entries expand to 2x2 non-edge entries.

Expansion : Tanner graph

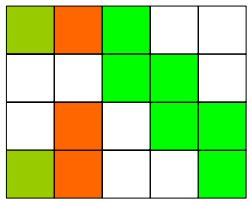
Even elements are those of the form 2x+0

Odd elements are those of the form 2x+1

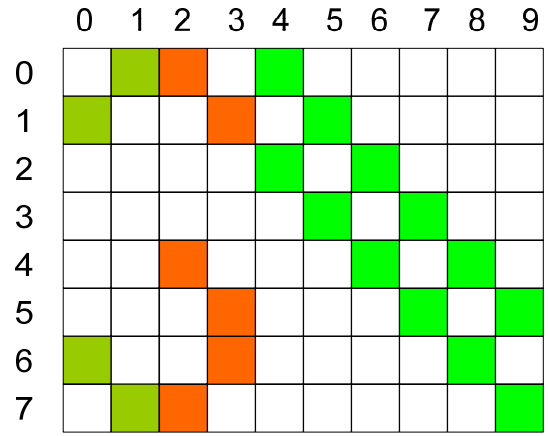


Original base pair v,c becomes v0,v1, c0,c1. Shift 3 in lifting of size 8 in original becomes lifting of 1 from v0 to c1 and 2 from v1 to c0.

Representative Expansion of Encoding structure



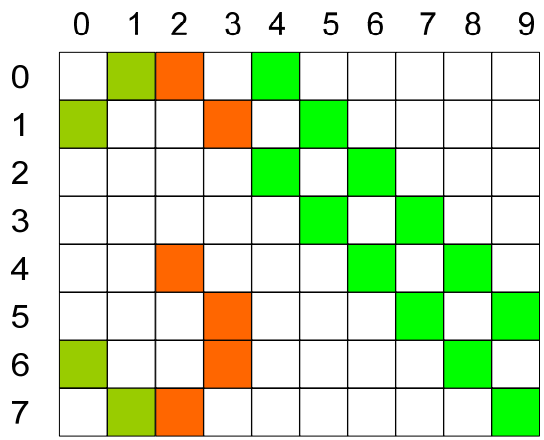
G



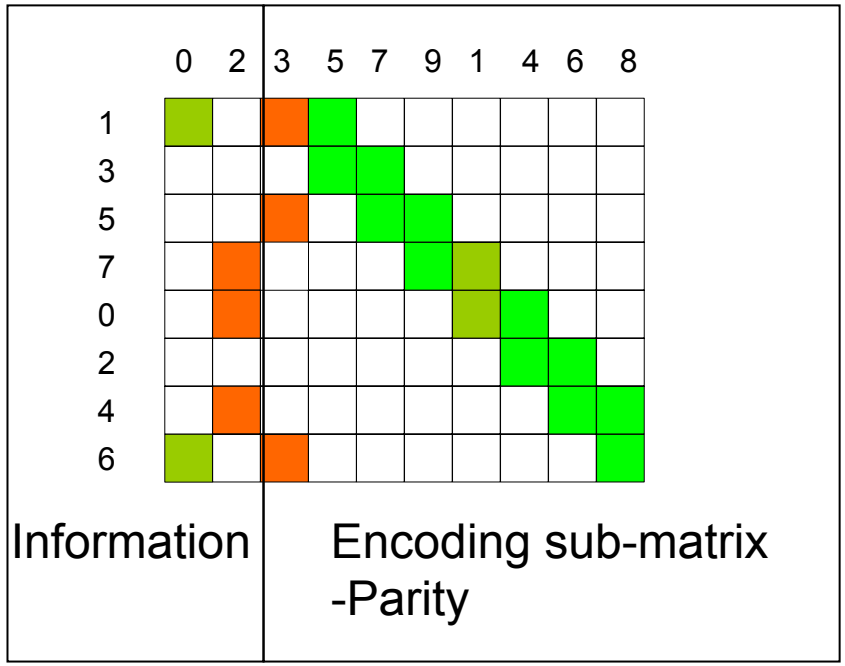
G’

Each base node is split into an even and an odd copy. The loop-closing degree 2 variable node and the loop-closing degree 3 variable node each have one edge in the loop with an even lifting and one with an odd lifting. Furthermore, in G the odd/even liftings in the loop closing are mixed at the constraints (one odd and one even lifting in each case). All other liftings are assumed even.

Representative Expansion of Encoding structure



G’



Reordering shows the presence of a 2X+1 long dual diagonal structure with a loop closing degree 3 variable node. This then forms the new base encoding structure. Liftings in G are chosen so that in G’:

1. Each degree 2 variable node in the dual diagonal structure has the same lifting on each edge.
2. The loop closing edges on the degree 3 node have the same lifting value s.
3. The non-loop edge of the degree 3 has lifting permutation 0 (identity) so the lifting structure of this node is s-0-s.

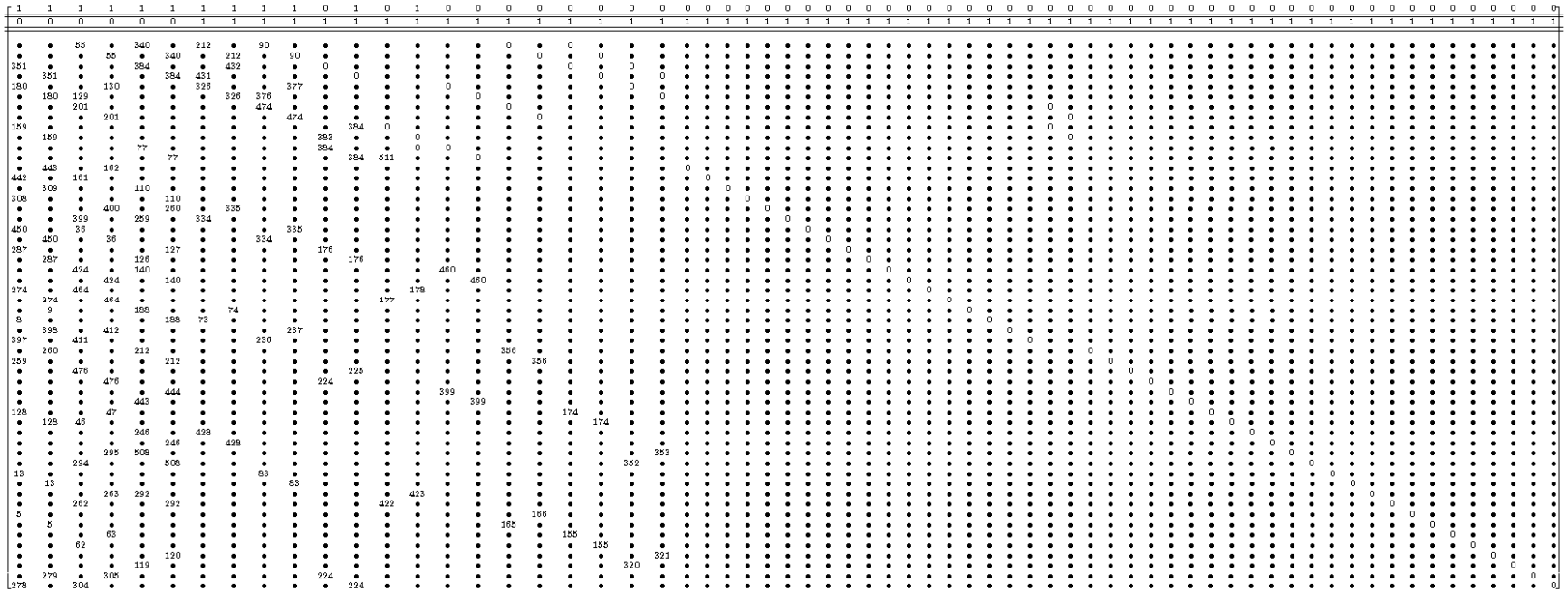
Example lifted parity check matrix (L=1024)

- Top row: 1 = information bits, 2 = odd elements are information bits, 0 = parity bits
- Second row: 0 = punctured bits, 1 = transmitted bits.
- Matrix:
  - bullets – no lifted edge present
  - Integer – cyclic shift (mod 1024) of lifted edge

Rightmost column of full information bits (top row=1) (5th column in example) used for zero padding. The last bits in the column are zero-padded.

1	1	1	1	1	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
•	110	680	424	180	•	•	•	0	0	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
702	•	768	863	•	0	•	•	•	0	0	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
360	259	•	652	753	•	•	0	•	•	0	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	402	•	•	948	•	•	•	0	•	•	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•	•	•	•	•
318	•	•	•	•	767	0	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•	•	•	•	•
•	•	154	•	•	768	1023	0	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
885	323	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
617	•	220	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	799	519	669	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
900	72	•	•	669	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
574	•	253	•	•	352	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	848	280	•	•	•	•	920	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
548	928	•	•	•	•	355	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
17	•	376	147	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
795	823	•	•	473	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•	•	•	•	•	•	•
519	•	424	•	•	•	•	•	712	•	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•	•	•	•	•	•
•	952	•	•	•	449	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•	•	•	•	•
•	•	887	•	•	•	•	798	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•	•	•	•
256	93	•	•	•	•	•	•	•	348	•	•	•	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•	•	•
•	•	492	856	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•	•
•	589	1016	•	•	•	•	•	•	•	705	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•
26	•	•	•	166	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•
•	525	584	•	•	•	845	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•
10	•	•	•	•	•	•	•	331	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•
•	125	•	•	•	•	•	•	•	310	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•
•	•	239	•	•	•	•	•	•	•	641	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0	•	•	•
557	609	•	•	•	448	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0	•	•

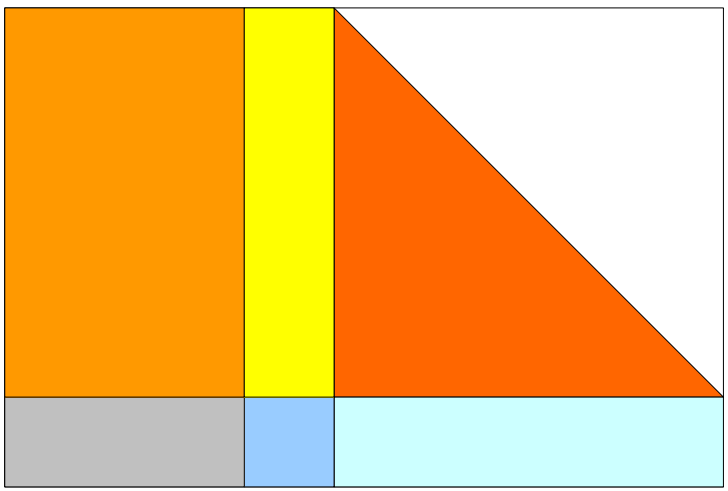
Expanded example graph



Each matrix entry in original example is expanded to a 2x2 block.
Column 11 is the even portion of encoding degree 3 (information bits) from the original graph..
Column 12 is the odd portion of encoding degree 3. Note s-0-s structure in first three shifts (reordered here).
Column 13 is the even portion of loop-closing degree 2 from the original graph (information bits).
Column 14 is the odd portion of loop closing degree 2 (parity bits).
(Note: degrees are actually larger than stated above because of additional parity bits.)

Encoding: explanation of derivation

$$\begin{bmatrix} B & A & T \\ E & D & C \end{bmatrix}$$



Up to reordering of columns and rows, lifted parity matrix takes the form shown here where T is lower triangular and

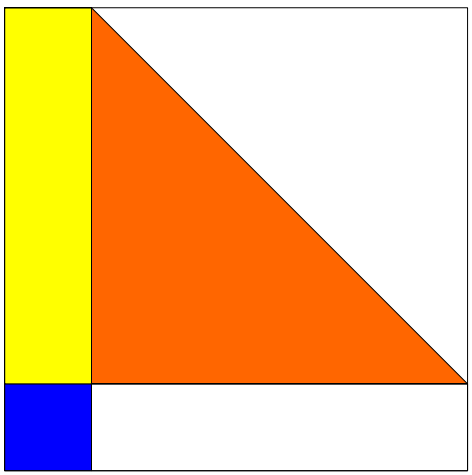
$$\begin{bmatrix} A & T \\ D & C \end{bmatrix}$$

is invertible.

In all proposed codes D is 1×1.

Encoding: explanation of derivation

$$\begin{bmatrix} A & T \\ \phi & 0 \end{bmatrix}$$



Gaussian elimination on

$$\begin{bmatrix} A & T \\ D & C \end{bmatrix}$$

yields

$$\begin{bmatrix} A & T \\ \phi & 0 \end{bmatrix}$$

where $\phi = CT^{-1}A + D$

Encoding: operations: 3 steps

$$\begin{bmatrix} B & A & T \end{bmatrix} \begin{bmatrix} x^s \\ 0 \\ y \end{bmatrix} = 0$$

Set information bits x^s. Solve for y: back substitution, identical to erasure decoding

$$x^{p_2} = \phi^{-1}(Cy + Ex^s)$$

Multiplication by ϕ^{-1} to get parity bits x^{p_2}

$$\begin{bmatrix} B & A & T \end{bmatrix} \begin{bmatrix} x^s \\ x^{p_2} \\ x^{p_1} \end{bmatrix} = 0$$

Solve for x^{p_1}: back substitution - repeat step 1 with different data

Encoding Simplifications

- Encoding effectively done in expanded graph.
- Parameters chosen so that Φ^{-1} is identity matrix.
- The provided LDPC code parity check matrices are for the lowest rates, they are punctured to provide higher rates.
- The codes are structured as an inner “core” LDPC of high rate with outer parity bits.
- The encoding procedure described above can be applied to the core only, except for the last step where the parity bits are also determined.

Support for Multiple Packet Sizes - I

- LBC system supports a variety of packet sizes, determined by
  - Spectral efficiency (i.e., packet format), assignment bandwidth, pilot overhead, MIMO rank, etc
- To support all these packet sizes and code rates, we propose the following strategy
  - Generate a set of base codes for values of k_B ranging from 6 to 11
  - For each k_B, generate a sequence of HARQ extensions spanning all the code rates supported by the LBC system
    - Eg. for $k_B = 6$, the code provided is a (30,6) code, i.e., lowest code rate 0.2 (final lowest code rate TBD).
    - Higher code rates are generated by puncturing this code
  - For all the base codes thus defined, generate liftings of size 2^l for a range of values of l, eg. $2^l = 16, 32, 64, 128, 256, 512, 1024$
- Scaling of the lifting entries
  - Corresponding to each edge e in the base graph, the shift parameter s_e' for a smaller lifting size L_1 can be generated from the shift parameter s_e for the largest lifting size L_0 according to a scaling and flooring operation
  - Exact formula is given by $s_e' = \lfloor s_e L_1 / L_0 \rfloor$
  - This scaling operation preserves the relationship described in the preceding slides between encoding parameters for all lifting sizes.

Encoding Preserved under Scaling

-1	H	0		
		0	0	
	0		0	0
0	H-1			0

Let 2^L be maximum lifting size. (e.g L=10).
For each degree 2 variable node in the dual-diagonal structure both edges use the same permutation, e.g. 0.
Degree 3 variable node permutations are H,0,H-1 where
$H=2^{L-3}S$ for some S in 0,1,…,7.
Loop closing degree 2 shifts are -1 and 0 (mod 2^L).

If we scale the lifting size from 2^L to 2^l where l is in the set {4,5,…,L} then we observe that the structure is unchanged except that -1 should be interpreted modulo 2^l and $H=2^{l-3}S$.

After expansion of the scaled matrix the encoding structure is as shown below, where $H'=2^{l-4}S$ and -1 is modulo 2^{l-1}

	0	1	2	3	4	5	6	7	8	9
0		0	H'		0					
1	-1			H'		0				
2					0		0			
3						0		0		
4			0				0		0	
5				0				0		0
6	0			H'					0	
7		0	H'-1							0

	0	2	3	5	7	9	1	4	6	8
1	-1		H'	0						
3				0	0					
5			0		0	0				
7		H'-1				0	0			
0		H'					0	0		
2								0	0	
4		0							0	0
6	0		H'							0

Support for Multiple Packet Sizes - II

- For any desired packet size (n,k), choose k_B, l (at least 4) and n_B :

$$l = \lceil \log_2 (k/11) \rceil$$
$$k_B = \lceil 2^{-l} k \rceil$$
$$n_B = \lceil 2^{-l} (n - (k - 2^l k_B)) \rceil$$

- To get desired code rate and size

  zero-pad $z=2^l k_B - k$ information bits ($< 2^l$ bits) starting from information bit $2^l k_B - z - 2^l$ (enumeration starts with 0)
  (zero-pad end of last full information column)
  puncture the last $2^l n_B - (n - (k - 2^l k_B))$ bits ($< 2^l$ bits)

- This procedure allows to generate good rate-compatible codes for any values of k and n using a small number of mother codes. (HARQ extensions share the same k and zero-pad.)

Support for Multiple Packet Sizes - III

- Attachment XXXXXX shows the set of parity check matrices for all values of k_B between 6 and 11
  - Matrix columns are ordered from left to right in the order of transmission
- The matrices specify a lifted LDPC code in the following manner:
  - Size of the matrix corresponds to the size of the base graph
  - The two rows above the matrix indicate the following:
    - Whether a given variable node is an information node
    - Whether a given variable node is transmitted or is a state variable (punctured node)
  - A black circle in the matrix indicates that no edge is present
  - An integer in the matrix indicates that an edge is present and the value of the integer specifies the cyclic permutation to be used for lifting that edge
    - Integers take values between 0 and L-1, and specify elements of the group Z_L
      - Specify a lifting of order L
      - Here, L is the maximum lifting size
    - Liftings of smaller order (limited to smaller powers of two) are defined implicitly by using the division and scaling operation described in the preceding slides

LDPC Support for AT and AP

- Support for LDPC coding to be optional at both AT and AP
  - Separate capability parameter for FL and RL
- If both AT and AP support LDPC codes, LDPC codes are used for all packet sizes above a certain minimum size
  - This minimum size is sufficient in order to demodulate overhead information and complete initial capability negotiation
  - Turbo codes are used for transmitting all overhead information
- There is also a capability field specifying maximum packet size supported using turbo codes
  - Ensures that a terminal implementing LDPC coding does not also have to implement a high-rate turbo decoder

Conclusion

- Proposed an LDPC coding scheme that:
  - Provides performance comparable to or better than that of turbo codes
  - Built in support for parallelization, enabling implementation of high-speed decoders
  - Provides efficient support for HARQ