**3GPP TSG-RAN WG1 NR R1-1706175**

**Spokane, USA, 3th– 7thApr 2017**

**Agenda item:** **8.1.4.1.2**

**Source: Mediatek Inc.**

**Title: Multi-codebook embedded compact QC-LDPC designs**

**Document for:** **Discussion/Decision**

# Introduction

In R1-88 meeting, there was some agreements relating to LDPC code.

**Agreement:**

* Number of base graphs for eMBB is FFS between 1 and 2
* Evaluate the potential gains from 2 base-graphs compared to a single base-graph until RAN1#88bis

**Agreement:**

* The largest info block size supported by LDPC encoder Kmax and the largest shift size Zmax defined for a H matrix are selected from the following set of {Kmax, Zmax} pairs:
  + {8192, 256}, {8192, 512}, {FFS near 8192, 320}

**Agreement:**

* For at least one base graph,
  + the parity check matrix consists of five sub-matrices (A, B, C, D, E)

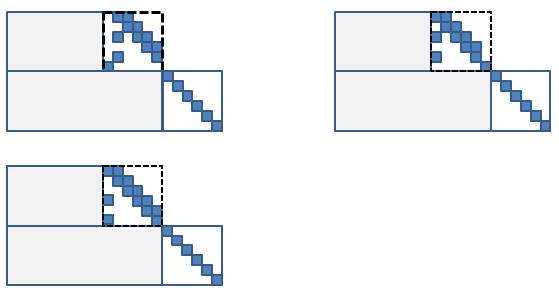
A

C

D

E

B

* A corresponds to systematic bits
* B is square and corresponds to parity bits
  + The first or last column may be weight 1
    - The non-zero value is in the last row and this row is weight 1 in B
    - If there is a weight 1 column, then the remaining columns contain a square matrix such that:
      * First column has weight three
      * The columns after the weight three column have a dual diagonal structure (i.e., main diagonal and off diagonal)
  + If there is no weight 1 column
    - B consists of only a square matrix such that:
      * First column has weight three
      * The columns after the weight three column have a dual diagonal structure (i.e., main diagonal and off diagonal)
  + E.g.:  
    
* Other structures can be considered for other base graph(s), if any
* Can be revisited if another structure is shown to be superior in performance and complexity

In this contribution, we

1. Further optimize the compact QC-LDPC code
   1. Enhance the required SNR at BLER=1e-4.
2. Integrate non-row orthogonal (non-RO), quasi-RO and pure-RO into a single base matrix.
   1. Non-RO for higher code rates (CRs), quasi-RO for medium CRs and pure-RO for lower CRs.
   2. The design concept is to keep the performance as the first priority and try to accommodate the structure to adapt to the higher throughput.
3. Embed two LDPC codes into a single base matrix
   1. The first one is to support code rates from 0.33 to 0.89 with the maximal code block size of 8192.
   2. The second one is to support code rates from 0.2 to 0.66 with code block sizes smaller than or equal to 6144.
4. Propose the valid set design of lifting factor to further smooth the CBS granularity performance.

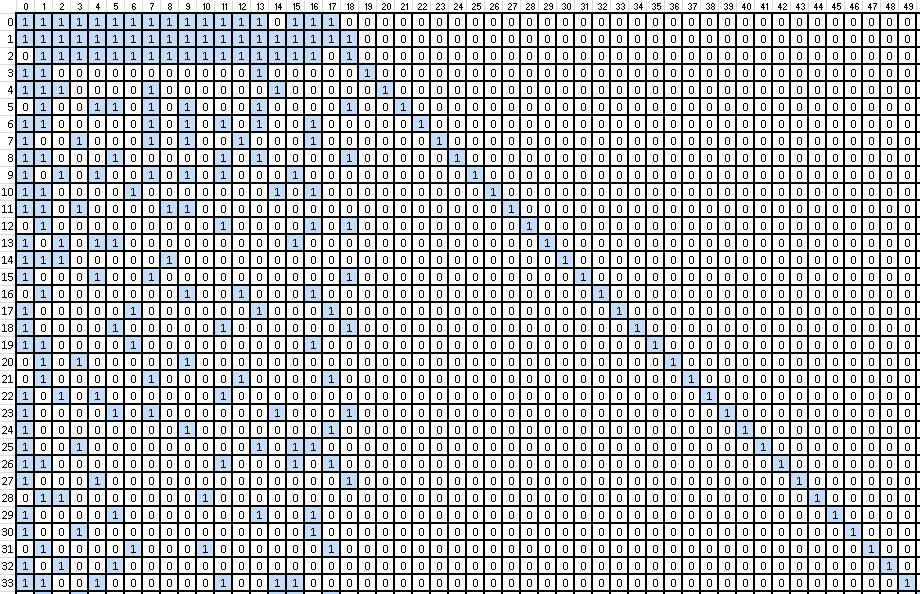
We proposed two different compact QC-LDPC codes, QC-LDPC Code A and QC-LDPC Code B. They have similar structure but with different design criterion around orthogonality.. The corresponding performance and throughput analysis can be found in [1] and [2].

# QC-LDPC Code A

## Base Matrix Design

The proposed single-edge base matrix A of Check Nodes (CNs) 0~33 is shown in Figure 1. This proposal follows the working agreement of R1-88. The blocks of Variable Nodes (VNs) 0~15 correspond to information bits and the other VN blocks correspond to parity bits. Blocks of VN0 and VN1 are punctured in the beginning of the initial transmission. For flexible message sizes, the zero-padded bits are allocated from right to left starting from VN15. For rate-matching, the parity bits are punctured from right to left. Therefore, for CR=8/9, the sub-matrix on the upper left corner of the base matrix is used, and for CR=1/3, the sub-matrix on the upper left corner of the base matrix is used. For the CRs down to 1/5, the entire base matrix is used with 4 zero-padding blocks. The base matrix is attached in the excel file.

This base matrix has extremely good performance and is designed to be compact with quasi-RO or pure-RO which enables efficient implementations for both block-parallel LDPC decoders and row-parallel LDPC decoders. The implementation complexity and performance comparison are discussed in [1] and [2].



Puncturing columns

Figure 1: Proposed LDPC base matrix A of CN0~CN33

**Proposal 1:** The most compact as possible base matrix should be considered provided its performance meets or exceeds that of other competing proposals.

## Multi-codebook Embedded Base Matrix

QC-LDPC codes have the characteristic that lower CRs would result in larger decoding latency because of the larger number of edge blocks. Therefore, it is preferential to have a QC-LDPC code with a smaller Kb corresponding to a smaller total edge block count such that the decoding latency is reduced for lower CRs. Some proposals directly extend the base matrix without considering the degraded latency. In our proposal, we consider to use zero padding in order to maintain large lifting sizes and hence reduce the decoding latency. Using this method, the extended number of rows to cover CR down to 0.2 can also be kept small too. A two codebook embedded base matrix is illustrated in Figure 2. The first QC-LDPC codebook uses with limited zero-padding blocks to support CRs from 0.33 to 0.89 and the second QC-LDPC codebook use (with 4 or more zero-padding blocks) to support CRs from 0.2 to 0.67.

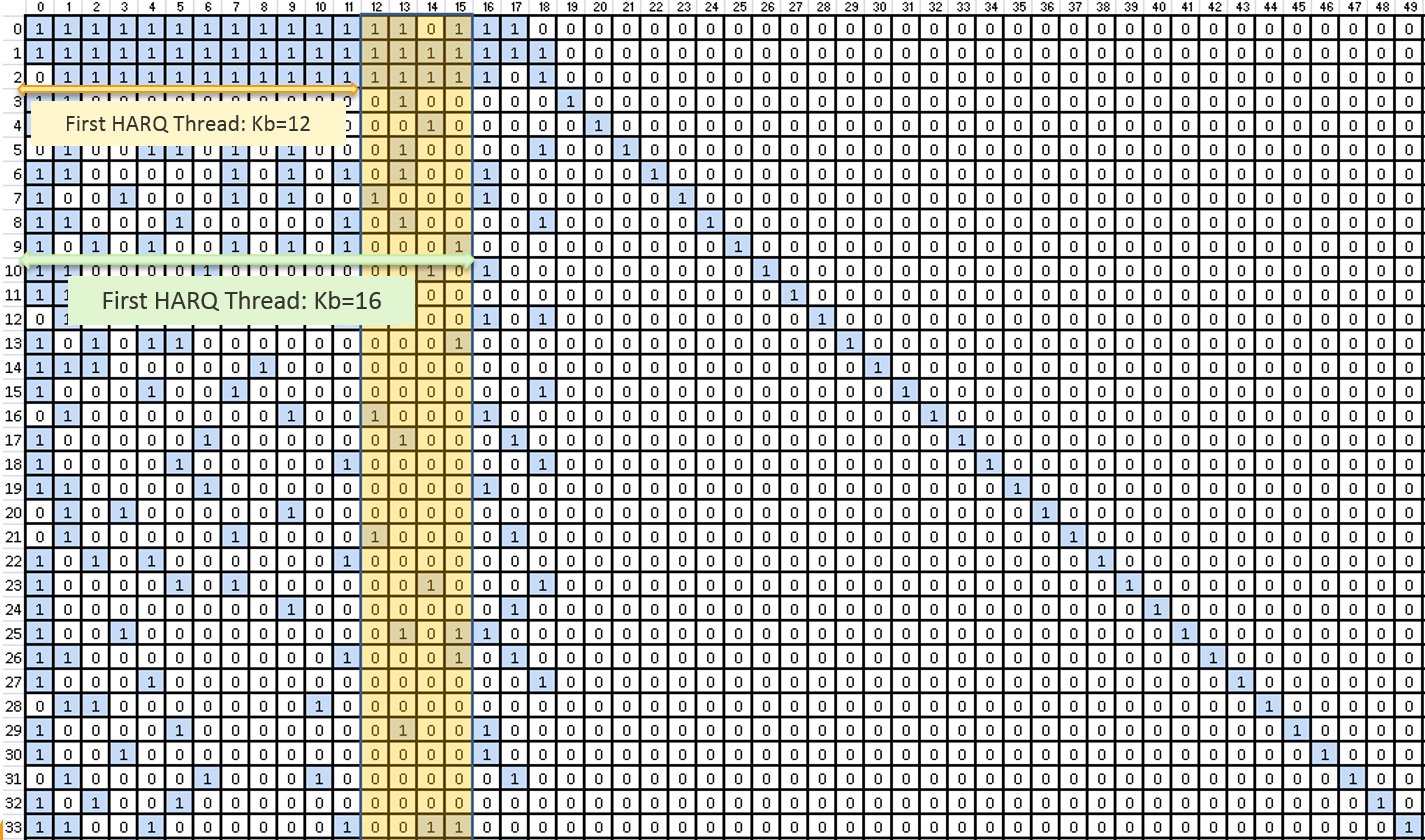


Figure 2: Multi codebook embedded base matrix A

The decision on which codebook is used depends on the initial CR.

The rationale behind this design is that it is hard for a gNB to decide on an initial transmission with CR larger than 0.67 based on CQI reporting. It may, however, still encounter a code block error even after retransmission(s) of all parity bits down to CR=0.33. At this point there is no other option other than to instigate chase combining through repetition, resulting in inferior performance when compared against the possibility to further lowering the CR.

Since both QC-LDPC codebooks share the same base matrix, we can expect that the control overhead, description overhead and the routing complexity of the decoder would be smaller compared with those proposals with two or more entirely different base matrices.

**Proposal 2:** NR eMBB data channel coding should prioritize the proposal with only one base matrix to reduce the control overhead, description overhead and routing complexity of the decoder when the performance is comparable.

## Row Orthogonality

A good QC-LDPC code usually has very different check node degrees, i.e., the row weights of the base matrix are quite different. In the row-parallel decoder, the hardware engine is required to support the CN block processing with the maximum CN degree. Therefore, some hardware modules are forced into idle mode when the decoder processes CN blocks with lower check node degrees. This results in poor hardware utilization. To increase the hardware utilization, it is preferential to decode several CN blocks that are orthogonal to each other in an orthogonal group. The number of orthogonal groups of rows in the base matrix is referred to as the layer number. The throughput of a row-parallel decoder highly depends on the number of layers in the base matrix.

A slight complication to this is that most good QC-LDPC codes have some puncturing VN blocks among the information VN blocks. It is common that the column weights of the punctured VN blocks are higher than average such that the decoder can speed up the convergence in decoding (LLR) messages for the punctured VN blocks. This characteristic is not good for a pRO design to reduce the layer number as a high weight column does not lend well to orthogonal grouping. Also, in our experience, seeking a QC-LDPC structure to aggressively reduce the layer number eventually degrade the performance when pushed too far.

It is for exactly this performance consideration that for higher CRs, we do not consider row orthogonality amongst these rows.

For medium CRs, we propose to design the base matrix in which rows are orthogonal to each other within the same group excluding the puncturing VN blocks. This relaxed version of row orthogonal base matrix we refer to as a quasi-row orthogonal (qRO) base matrix. There is one criterion on the qRO base matrix which is that no cycle is allowed within the punctured columns in the same orthogonal group. The reason here is that a cycle within punctured columns in the same orthogonal group would result in memory access problems in the decoder implementation when the CN blocks of the same layer are decoded simultaneously. The memory access problem arises because different shift values correspond to different memory addresses.

Take the following shift-coefficient table with as an example where there are cycles within the conflict column as in Figure 3. Assume the decoder is designed to parallel process 256 CNs within one cycle. The first step is to decompose the QC-LDPC code into the representation of as in Figure 6. The rule of decomposition is introduced in [2]. In the following discussion, the VN/ CN index would be based on the decomposed shift-coefficient table.

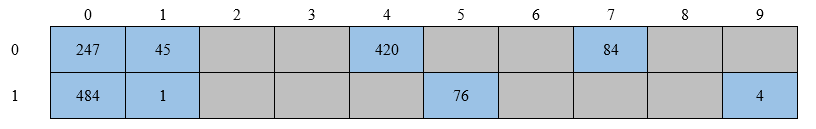


Figure 3: QC-LDPC code with a cycle inside the punctured columns

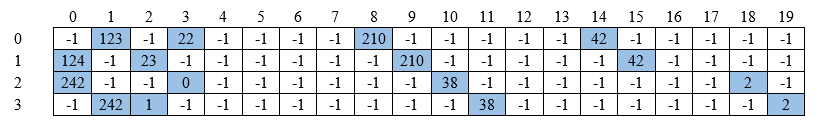


Figure 4: Decomposed QC-LDPC code with cycle inside the punctured columns

For memory efficiency, it is likely to aggregate the VNs which won’t be access simultaneously into one memory bank. So the VN 2n and VN 2n+1 of the decomposed shift-coefficient table are put in memory bank n. If we want to process CN0 and CN2 simultaneously in the decomposed shift-coefficient table of Figure 4, we will have an access conflict in the simultaneous reading of VN0 and VN1 because they both reside in the same memory bank. Likewise, if we instead want to process the CN0 and CN3 simultaneously, we will have an access conflict on the simultaneous reading of VN2 and VN3, or if we instead want to process the CN0 and CN1 simultaneously, we will have an access conflict on the simultaneous reading of VN0 and VN1 as well as VN2 and VN3.

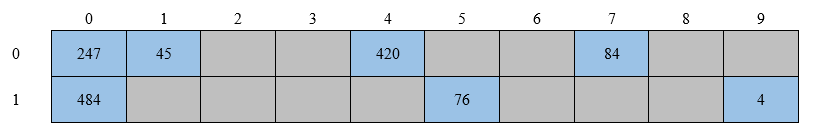
Now take the example where there are no cycles within the conflict column as in Figure 5.

Figure 5: QC-LDPC code without cycle inside the punctured columns

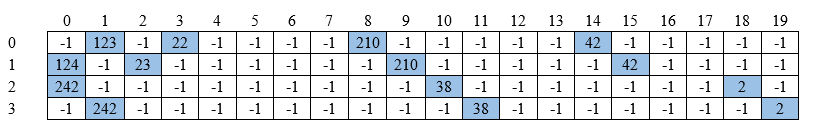


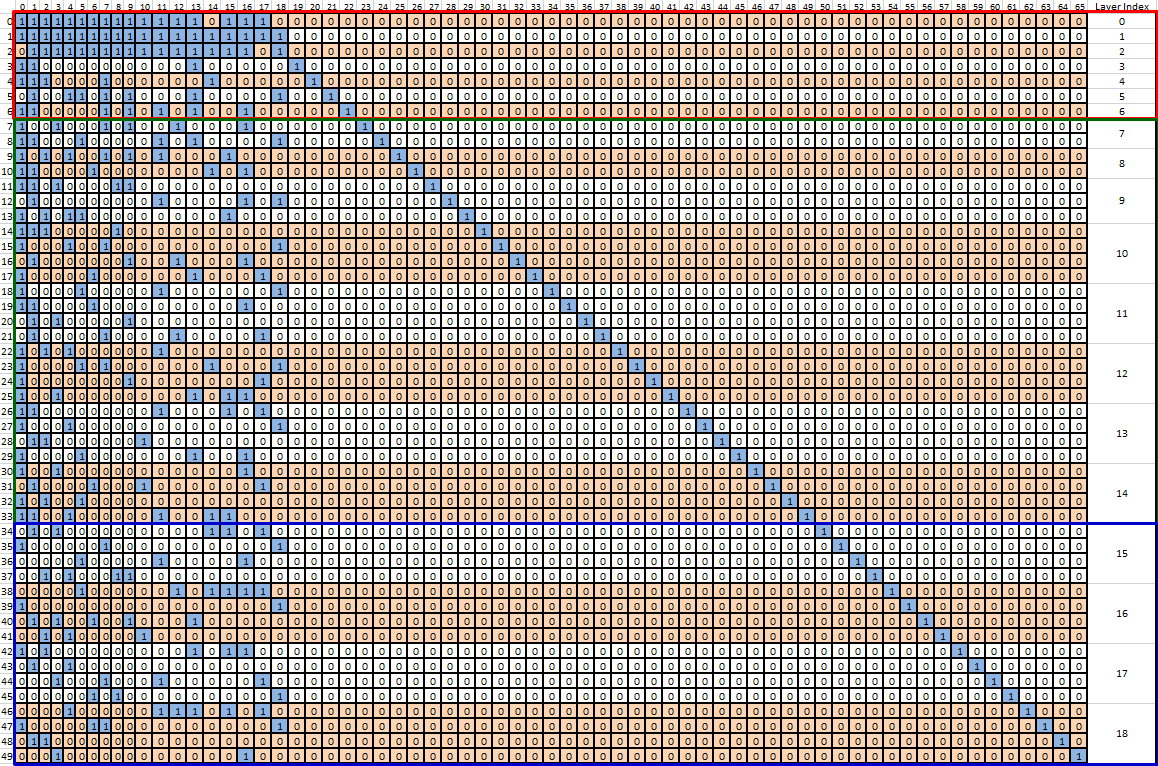
Figure 6: Decomposed QC-LDPC code without cycle inside the punctured columns

Then we can process CN0 and CN3 within one cycle and process CN1 and CN2 within the other cycle. No memory access conflicts exist here.

For lower CRs, we observed that the speed up in convergence of punctured VNs is not as critical. Therefore, we turn back to support pure row orthogonal (pRO) for lower rows of the base matrix.

Our design criterion is that the performance must not to be degraded due to the support of row orthogonality in any form, pure- or quasi-orthogonality. With this design concept, a row parallel LDPC decoder supporting qRO can fully leverage a small layer count to achieve extremely high throughput. A row parallel LDPC decoder supporting pRO can also achieve high throughput due to the support of the pRO. The proposed hybrid base matrix which combines qRO and pRO is shown in Figure 7.

**Proposal 3:** NR eMBB data channel coding should down-select to the proposals with row orthogonality to reduce layer number and therefore increase the decoding throughput when the performance is comparable.



Quasi-RO with 8 layers

Non-RO with 7 layers

Pure-RO with 4 layers

Figure 7: Base matrix A with hybrid row orthogonality

## Shift-coefficient Design

In this proposed QC-LDPC codebook, we define the sets of lifting factors (Z) as

The corresponding shift values are represented by 8 shift coefficient tables which correspond to shift coefficients of. For any lifting factor of within the set, the corresponding shift coefficient can be obtained by

,

where is the shift coefficient of the (-th element in the shift coefficient tables for . The shift-coefficient table supporting down to CR=0.33 of are shown in Figure 8 and Figure 9.

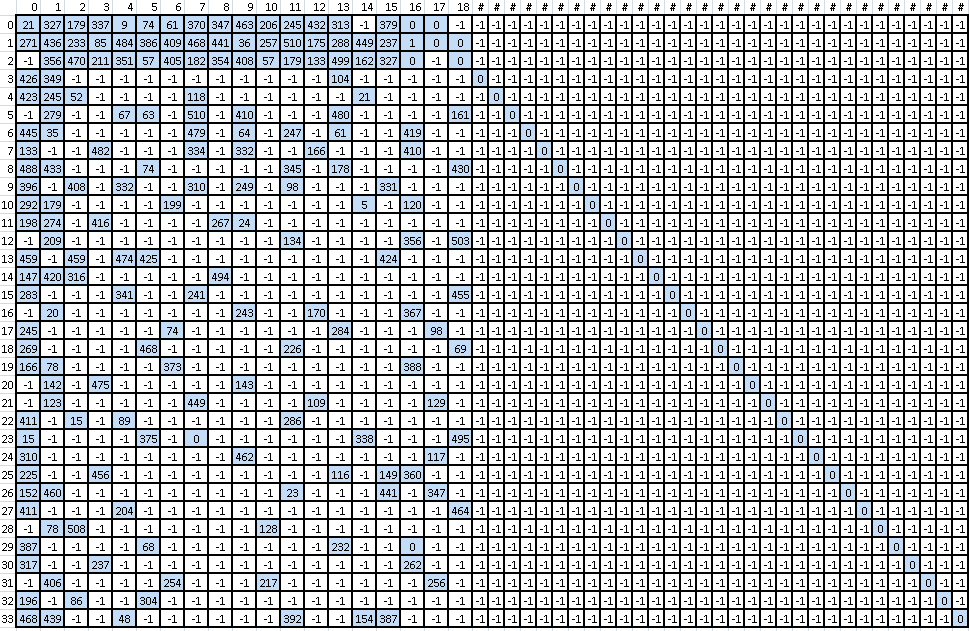


Figure 8: Shift coefficient table A for Z=512

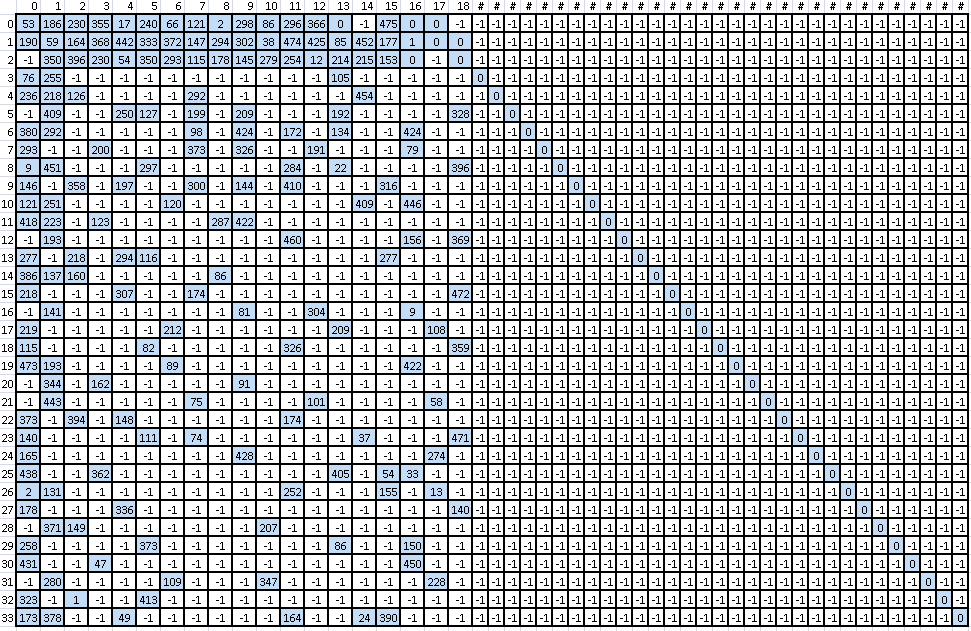


Figure 9 Shift coefficient table A for Z=480

## Lifting Factor and Zero-Padding Design

## In this proposed LDPC code, byte granularity or even 1-bit granularity of message size (K) is considered. Due to the limited number of lifting factors desired to be supported, shortening based on zero-padding of some information bits before encoding is required and proposed by many companies.

## For large message sizes, the lifting factor that results in the smallest number of zero-padding bits is selected, e.g.. Therefore, for the larger message size, the lifting factor is selected as

where for the outer codebook (CRs from 0.33 to 0.89) and for the inner codebook (CRs from 0.2 to 0.67).

For small message sizes, the lifting factor gets smaller and therefore the number of short cycles in the lifted Tanner graph gets larger. If nothing is done to mitigate this then the performance of the QC-LDPC code for small message sizes degrades accordingly. Therefore, the lifting factors for small message sizes must be carefully designed. In this proposed QC-LDPC code, a special design of lifting factors is required for. Some examples of lifting factors for some K of the outer codebook are listed in Table 1.

Table 1: Example code block sizes, K, to lifting factors, Z, mappings for First HARQ thread

|  |  |
| --- | --- |
| K | Z(K) |
| 40~47 | 12 |
| 48~63 | 10 |
| 64~95 | 16 |
| 96~400 | 28 |
| 1000 | 64 |
| 2000 | 128 |
| 4000 | 256 |
| 6000 | 384 |
| 8000 | 512 |

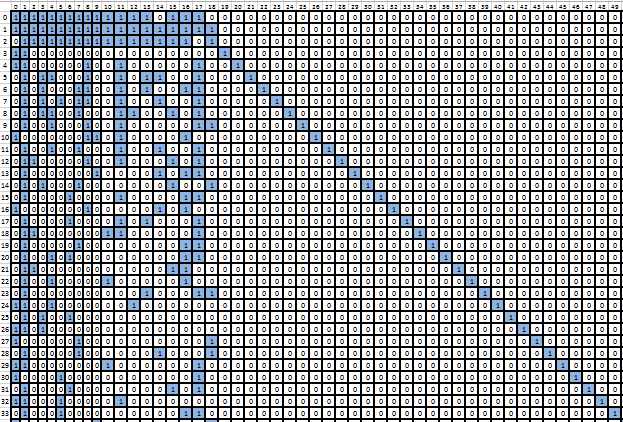
**Proposal4:** For small message sizes, the lifting factors need to be explicitly specified for better performance. For large message sizes, the lifting factor is selected such that the number of zero padded bits is minimized.

# QC-LDPC Code B

## Base Matrix Design

The proposed single-edge base matrix B of Check Nodes (CNs) 0~33 is shown in Figure 10. This proposal follows the working agreement of R1-88. The blocks of Variable Nodes (VNs) 0~15 correspond to information bits and the other VN blocks correspond to parity bits. Blocks of VN0 and VN1 are punctured in the beginning of the initial transmission. For flexible message sizes, the zero-padded bits are allocated from right to left starting from VN15. For rate-matching, the parity bits are punctured from right to left. Therefore, for CR=8/9, the sub-matrix on the upper left corner of the base matrix is used, and for CR=1/3, the sub-matrix on the upper left corner of the base matrix is used. For the CRs down to 1/5, the entire base matrix is used with 4 zero-padding blocks. The base matrix is attached in the excel file.

This base matrix has good performance and is designed to be compact with quasi-RO in the lower code rate extensions which enables efficient implementations for both block-parallel LDPC decoders and row-parallel LDPC decoders. The performance comparison are discussed in [1].



Puncturing columns

Figure 10: Proposed LDPC base matrix B of CN0~CN33

## Multi-codebook Embedded Base Matrix

This proposed QC-LDPC code B also embedded two codebooks as the proposed LDPC code A in Section 2.2. The first QC-LDPC codebook uses with limited zero-padding blocks to support CRs from 0.33 to 0.89 and the second QC-LDPC codebook use (with 4 or more zero-padding blocks) to support CRs from 0.2 to 0.67. The benefit of multi-codebook embedded design can be found in Section 2.2.

## Row Orthogonality

This proposed QC-LDPC code B integrates the quasi-row orthogonality characteristics. The qRO layer partition of the proposed QC-LDPC code B is shown in Figure 11. The design concept of row orthogonality can be found in Section 2.3.

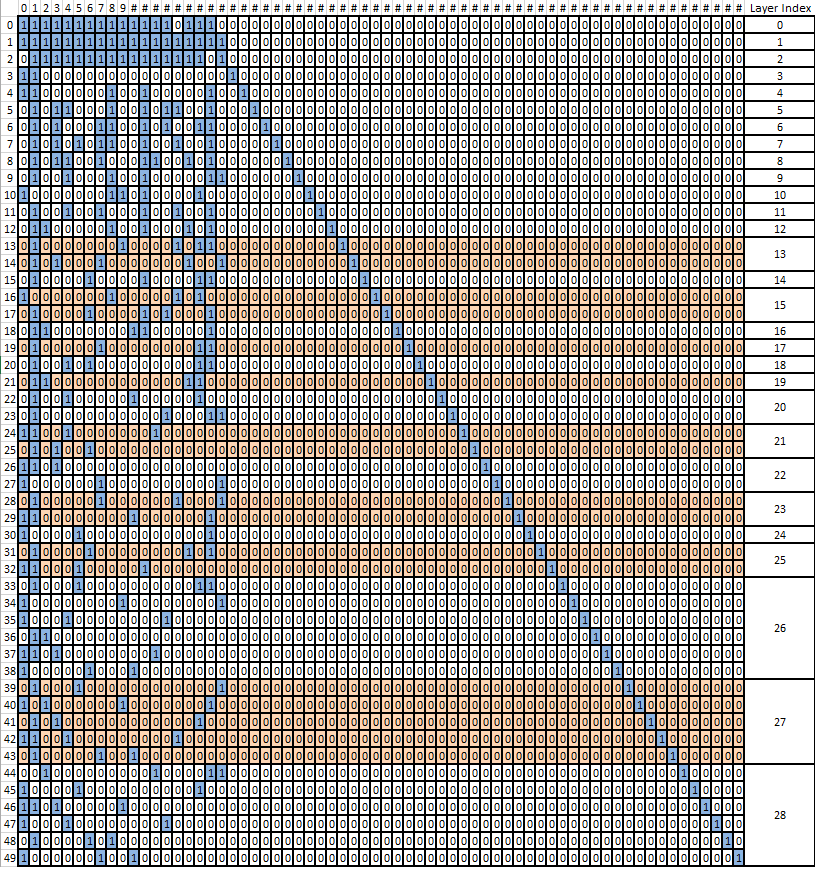


Figure 11: Layer partition of proposed QC-LDPC code B

## Shift coefficient Design

The proposed QC-LDPC code B has similar shift-coefficient design as QC-LDPC code A. The sets of lifting factors (Z) are defined as

The corresponding shift values are represented by 8 shift coefficient tables which correspond to shift coefficients of. For any lifting factor of within the set, the corresponding shift coefficient can be obtained by

,

where is the shift coefficient of the (-th element in the shift coefficient tables for .

## Lifting Factor and Zero-Padding Design

The proposed QC-LDPC code B has similar lifting factor and zero-padding design as QC-LDPC code A.

## For large message sizes, the lifting factor that results in the smallest number of zero-padding bits is selected, e.g.. Therefore, for the larger message size, the lifting factor is selected as

where for the outer codebook (CRs from 0.33 to 0.89) and for the inner codebook (CRs from 0.2 to 0.67).

For small message sizes, the lifting factor gets smaller and therefore the number of short cycles in the lifted Tanner graph gets larger. If nothing is done to mitigate this then the performance of the QC-LDPC code for small message sizes degrades accordingly. Therefore, the lifting factors for small message sizes must be carefully designed. In this proposed QC-LDPC code, a special design of lifting factors is required for. Some examples of lifting factors for some K of the outer codebook are listed in Table 1.

# Conclusion

The following summarizes the observations and proposals in this contribution. The corresponding performance and throughput analysis can be found in [1] and [2].

**Proposal 1:** The most compact as possible base matrix should be considered provided its performance meets or exceeds that of other competing proposals.

**Proposal 2:** NR eMBB data channel coding should down-select to the proposals with only one base matrix to reduce the control overhead, description overhead and routing complexity of the decoder when the performance is comparable.

**Proposal 3:** NR eMBB data channel coding should prioritize the proposal with row orthogonality to reduce layer number and therefore increase the decoding throughput when the performance is comparable.

**Proposal 4:** For small message sizes, the lifting factors need to be explicitly specified for better performance. For large message sizes, the lifting factor is selected such that the number of zero padded bits is minimized.

# References

1. R1-1706176, QC-LDPC codes performance comparison
2. R1-1706177, QC-LDPC codes throughput comparison