Protograph example

$$B = \begin{bmatrix} 1 & -1 & 3 & 1 & 0 & -1 \\ 2 & 0 & -1 & 0 & 0 & 0 \\ -1 & 4 & 2 & 1 & -1 & 0 \end{bmatrix}$$

Expansion factor: 5

$$H = \left[\begin{array}{ccccc|ccccc|ccccc|ccccc|ccccc|ccccc}
0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
\hline
0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0\\
0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0\\
0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0\\
1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0\\
0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1\\
\hline
0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{array}\right]$$

For base graph 1,

$$K = 22Z_c$$

For base graph 2,

$$K = 10Z_c$$

Table 3.1: Sets of LDPC lifting size[1]

Set index (i_{LS})	Set of lifting sizes (Z_c)
0	2, 4, 8, 16, 32, 64, 128, 256
1	3, 6, 12, 24, 48, 96, 192, 384
2	5, 10, 20, 40, 80, 160, 320
3	7, 14, 28, 56, 112, 224
4	9, 18, 36, 72, 144, 288
5	11, 22, 44, 88, 176, 352
6	13, 26, 52, 104, 208
7	15, 30, 60, 120, 240

5G NR base matrices

- Two base matrices
  - BG1: 46 x 68 and BG2: 42 x 52
- Block structure of base matrices

  A E O

  B C I
- BG 1
  - A: 4 x 22, E: 4 x 4, O: 4 x 42 all zero
  - B: 42 x 22, C: 42 x 4, I: 42 x 42 identity
- BG 2
  - A: 4 x 10, E: 4 x 4, O: 4 x 38 all zero
  - B: 38 x 10, C: 38 x 4, I: 38 x 38 identity

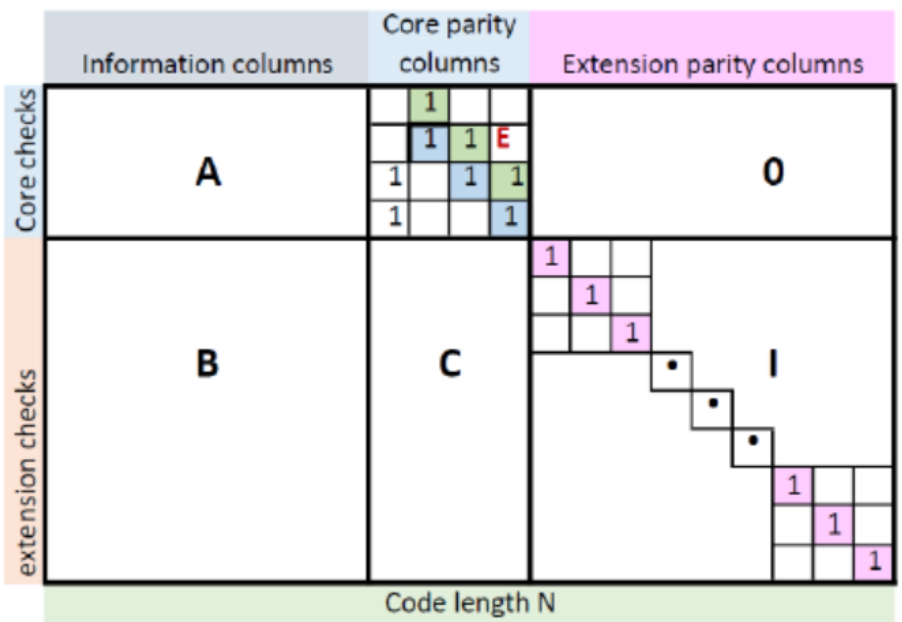


Figure 3.1: Base graphs block structure

A+E: high-rate base submatrix, information bit+ core parity bits

B: extension region, support IR-HARQ (incremental redundancy hybrid automatic repeat request)

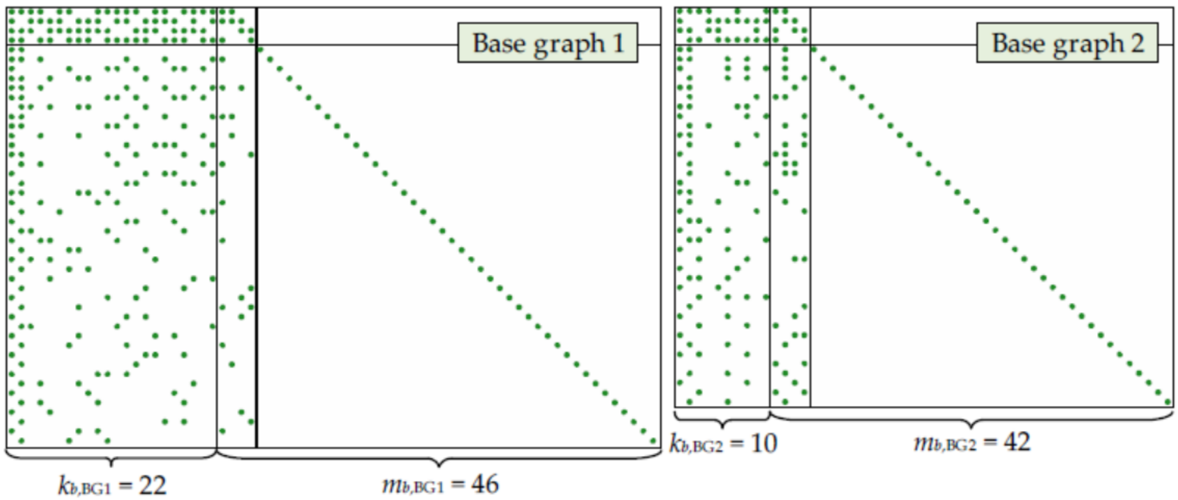
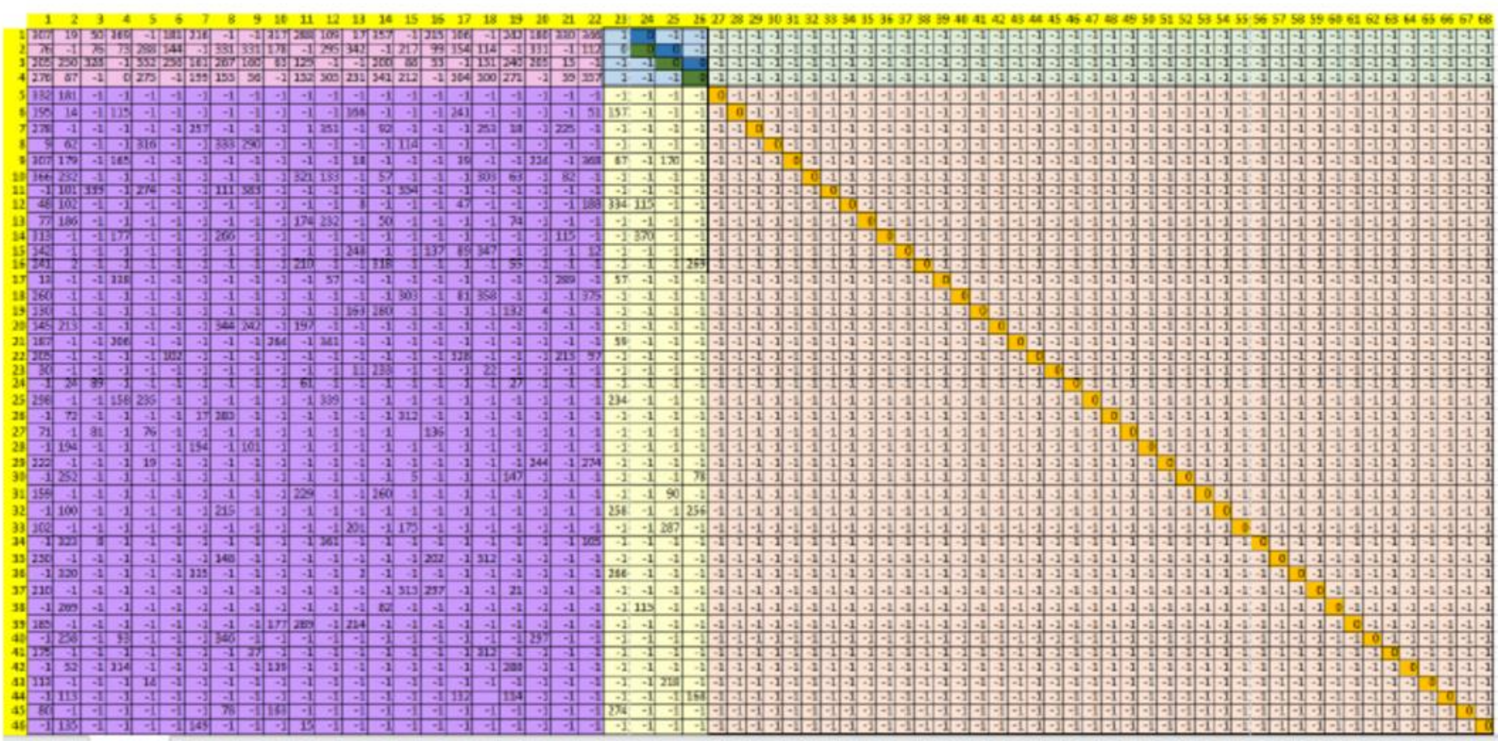


Figure 1. Base graph matrix structures for codes in 5G NR.



Example: base graph 1, iLS=1, Zc=384

Example: Base matrix entries

- 10 rows and 20 columns of BG2
  - iLS = 3, j = 4, Zc = 48

24	14	23	37	−1	−1	47	−1	−1	8	1	0	−1	−1	−1	−1	−1	−1	−1	−1
5	−1	−1	12	19	12	19	8	29	31	−1	0	0	−1	−1	−1	−1	−1	−1	−1
8	35	−1	46	47	−1	−1	−1	43	−1	0	−1	0	0	−1	−1	−1	−1	−1	−1
−1	41	6	−1	36	28	28	14	12	37	1	−1	−1	0	−1	−1	−1	−1	−1	−1
8	16	−1	−1	−1	−1	−1	−1	−1	−1	−1	5	−1	−1	0	−1	−1	−1	−1	−1
41	42	−1	−1	−1	26	−1	27	−1	−1	−1	1	−1	−1	−1	0	−1	−1	−1	−1
27	−1	−1	−1	−1	7	−1	31	−1	30	−1	17	−1	−1	−1	−1	0	−1	−1	−1
−1	7	−1	−1	−1	13	−1	9	−1	−1	−1	6	−1	37	−1	−1	−1	0	−1	−1
3	43	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	8	−1	−1	−1	−1	−1	0	−1
−1	2	−1	−1	−1	−1	−1	−1	30	−1	40	35	−1	−1	−1	−1	−1	−1	−1	0

Example: Double-diagonal

$$
H \overset{4}{=} \begin{array}{c}
\begin{array}{cccccccc} & & & & 8 & & & \end{array} \\
\begin{bmatrix}
I_1 & 0 & I_3 & I_1 & I_2 & I & 0 & 0 \\
I_2 & I & 0 & I_3 & 0 & I & I & 0 \\
0 & I_4 & I_2 & I & I_1 & 0 & I & I \\
I_4 & I_1 & I & 0 & I_2 & 0 & 0 & I
\end{bmatrix}
\end{array}
$$

Expansion: 5

- I_k : identity matrix column-shifted k times
- Message: [m1 m2 m3 m4]
  - m1, m2, m3, m4: 5 bits each
- Codeword: [m1 m2 m3 m4 p1 p2 p3 p4]
  - p1, p2, p3, p4: 5 bits each

Double-diagonal encoding

- H [m1 m2 m3 m4 p1 p2 p3 p4]^T = 0
  - 1: I_1 m1 + I_3m3 + I_1 m4 + I_2p1 + I p2 = 0
  - 2: I_2m1 + I m2 + I_3 m3 + I p2 + I p3 = 0
  - 3: I_4m2 + I_2m3 + I m4 + I_1p1 + I p3 + I p4 = 0
  - 4: I_4m1 + I_1m2 + I m3 + I_2p1 + I p4 = 0
- Adding all 4
  - I_1p1 = I_1 m1 + I_3m3 + I_1 m4 + I_2m1 + I m2 + I_3 m3 + I_4m2 + I_2m3 + I m4 + I_4m1 + I_1m2 + I m3
  - Find p1 from above
- p2: use p1 in 1, p3: use p2 in 2, p4: use p3 in 3

Example: 5G base matrix

24	14	23	37	−1	−1	47	−1	−1	8	1	0	−1	−1	−1	−1	−1	−1	−1	−1
5	−1	−1	12	19	12	19	8	29	31	−1	0	0	−1	−1	−1	−1	−1	−1	−1
8	35	−1	46	47	−1	−1	−1	43	−1	0	−1	0	0	−1	−1	−1	−1	−1	−1
−1	41	6	−1	36	28	28	14	12	37	1	−1	−1	0	−1	−1	−1	−1	−1	−1
8	16	−1	−1	−1	−1	−1	−1	−1	−1	−1	5	−1	−1	0	−1	−1	−1	−1	−1
41	42	−1	−1	−1	26	−1	27	−1	−1	−1	1	−1	−1	−1	0	−1	−1	−1	−1
27	−1	−1	−1	−1	7	−1	31	−1	30	−1	17	−1	−1	−1	−1	0	−1	−1	−1
−1	7	−1	−1	−1	13	−1	9	−1	−1	−1	6	−1	37	−1	−1	−1	0	−1	−1
3	43	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	8	−1	−1	−1	−1	−1	0	−1
−1	2	−1	−1	−1	−1	−1	−1	30	−1	40	35	−1	−1	−1	−1	−1	−1	−1	0

- Message: [m1 m2 … m10], each 48 bits
- Parity: [p1 p2 p3 p4 p5 p6 …]
- First four rows: use double-diagonal encoding to find p1, p2, p3, p4
- Row 5: p5, Row 6: p6, and so on