

5G NR LDPC Encoder and Decoder

YADATI KRISHNA

CONTENTS

1	Introduction	1
2	5G NR LDPC Encoder	1
2.1	5G-NR LDPC Channel coding	1
2.2	LDPC Base Graphs	2
2.3	LDPC Base Graph selection procedure	2
2.4	Encoding procedure	4

1. INTRODUCTION

5G NR (New Radio) is the latest wireless communication standard that supports higher data rates, lower latency, and increased reliability. LDPC (Low-Density Parity-Check) is a forward error correction (FEC) technique used in 5G NR to improve the reliability of data transmission.

5G NR (New Radio) uses Low-Density Parity-Check (LDPC) codes for forward error correction (FEC) to improve the reliability of data transmission over wireless channels. LDPC codes are linear block codes that have a sparse parity-check matrix. The parity-check matrix is designed to have a low density of 1's, which means that only a small fraction of the bits in the matrix are 1's.

The 5G NR LDPC codes are designed to have a high coding gain and low error rates while maintaining low latency and complexity. The coding gain is the ratio of the signal-to-noise ratio (SNR) required for a non-coded system to achieve a certain bit error rate (BER) compared to the SNR required for a coded system to achieve the same BER. The higher the coding gain, the more robust the code is against channel noise and interference.

The 5G NR LDPC codes are specified by the 3GPP (Third Generation Partnership Project) and include several code rates ranging from 1/5 to 5/6. The code rates determine the amount of redundancy added to the original data to create the codeword. Higher code rates add more redundancy, which results in a more robust code but also increases the latency of the transmission.

The 5G NR LDPC codes have a block size of 8448 bits and are designed to be flexible and adaptable to different channel conditions. The code rate and the number of iterations used in the decoding process can be adjusted based on the channel conditions to optimize the trade-off between coding gain, latency, and complexity.

Overall, the 5G NR LDPC codes are a crucial component of the 5G NR communication system, and their performance

has a significant impact on the reliability and efficiency of data transmission over wireless channels.

2. 5G NR LDPC ENCODER

The 5G NR (New Radio) encoder is responsible for encoding information bits into a larger codeword to improve the reliability of data transmission. The 5G NR standard uses a specific type of forward error correction (FEC) technique called LDPC (Low-Density Parity-Check) for error correction. The 5G NR LDPC encoder adds redundant bits to the original information bits to create the codeword.

The 5G NR LDPC encoder uses a parity-check matrix to determine the redundant bits to add to the information bits. The matrix is designed to have a low-density of 1's, which means that only a small fraction of the bits in the matrix are 1's. This reduces the complexity of the encoder and decoder while still providing high coding gain and low error rates.

The 5G NR LDPC encoder can operate at different code rates, which determines the amount of redundancy added to the information bits. The higher the code rate, the more redundancy is added, which results in a more robust codeword but also increases the latency of the transmission.

The 5G NR encoder also includes other features such as rate matching and code block segmentation. Rate matching is used to adjust the size of the codeword to match the modulation and channel coding scheme used for transmission. Code block segmentation is used to divide the information bits into smaller code blocks, which can be processed and encoded separately.

Overall, the 5G NR encoder is a critical component of the communication system, and its performance impacts the reliability and efficiency of data transmission over 5G networks.

A. 5G-NR LDPC Channel coding

Based on my interpretation from [3GPP Specification 38.212](#) Rel 15 (Multiplexing and channel coding), I had put together the procedure on how LDPC Base Graph selection and coding happens.

- 1) For transmission of a DL transport block, a transport block CRC is first appended to provide error detection, followed by a LDPC base graph selection.
- 2) NR supports two LDPC base graphs, one for small transport blocks and one for larger transport blocks.
- 3) Then transport block is segmented into code blocks and code block CRC attachment is performed.

- 4) Each code block is individually LDPC encoded. The LDPC coded blocks are then individually rate matched.
- 5) Finally, code block concatenation is performed to create a codeword for transmission. Up to 2 code words can be transmitted simultaneously.

B. LDPC Base Graphs

There are two types of Base Graphs standardized in the specification, [3GPP Specification 38.212](#) (Multiplexing and channel coding). Base Graph is a Matrix where each of the entries can be further expanded based on the expansion factor Z_c .

- 1) Base Graph 1 (BG1) : With Matrix size 46X68 entries For Large Transport Block.

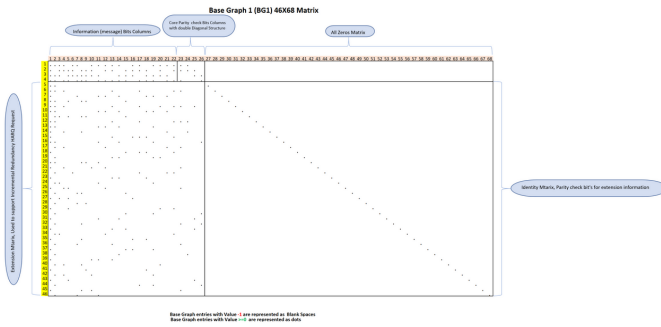


Fig. 1: High Level Visualization of fully Populated BG1 Matrix

- 2) Base Graph 2 (BG2) : With matrix size 42X52 entries For Smaller Transport Block.

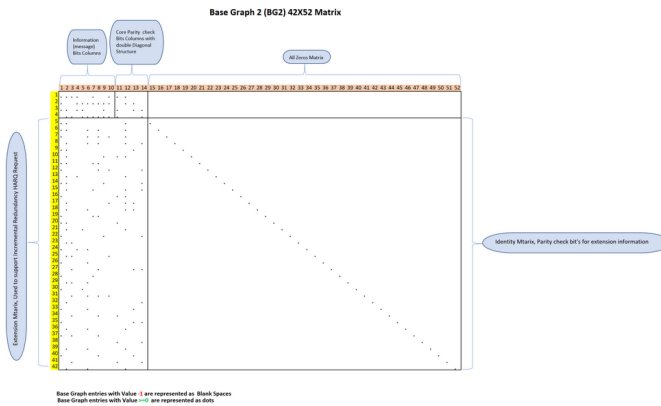


Fig. 2: High Level Visualization of fully Populated BG2 Matrix

C. LDPC Base Graph selection procedure

I have put together the following example of constructing the LDPC parity check matrix for a given information block size K and code rate $R = K/N$.

For simplicity I have considered a small TBS of size 20bits to illustrate below example, $K=20$ and $R=0.25$

- 1) Obtain the base graph BG1 or BG2 for the given K (Transport Block)and R (Code Rate), Refer [3GPP Specification 38.212](#) for LDPC base graph selection. As per the specification
 - a) if $K \leq 3824$ and $R \leq 0.67$ then BG2 is selected.
 - b) If $K \leq 292$ then BG2 is selected
 - c) if $R \leq 0.25$ then BG2 is selected.
 - d) Else BG1 is selected

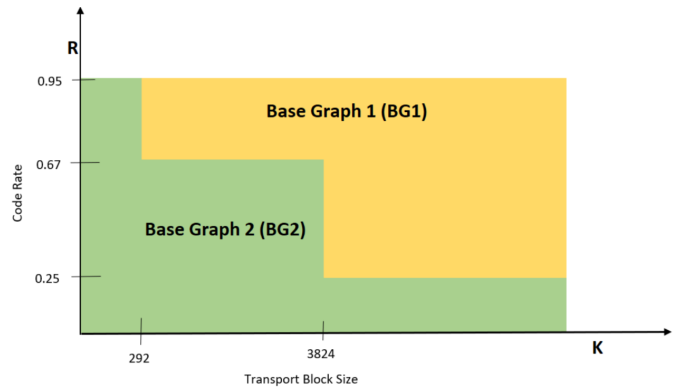


Fig. 3: Graphical Representation of Base Graph Selection

- 2) Determine the value of K_b for the given K (Transport Block) and R (Code Rate) Ref [3GPP Specification 38.212, 5.2.2](#)

K_b denotes the number of information bit columns for the lifting size Z_c .

As per the Specification

For LDPC BG1:

- a) $K_b = 22$

For LDPC BG2:

- a) if K is between $640 < K \leq 3824$ then $K_b = 10$
- b) if K is between $560 < K \leq 640$ then $K_b = 9$
- c) if K is between $192 < K \leq 560$ then $K_b = 8$
- d) If K is ≤ 192 then $K_b = 6$

- 3) Determine the base matrix expansion factor Z_c by selecting the minimum Z_c value in below Table, such that $K_b * Z_c \geq K$. Sets of LDPC lifting size Z_c in the specification I have populated below Z_c table.

a

Zc	2	3	5	7	9	11	13	15
0	2	3	5	7	9	11	13	15
1	4	6	10	14	18	22	26	30
2	8	12	20	28	36	44	52	60
3	16	24	40	56	72	88	104	120
4	32	48	80	112	144	196	208	240
5	64	96	160	224	288			
6	128	192	320					
7	256	384						

$$Zc = a * 2^j$$

For K=20 , Zc = 4, this satisfies the condition $Kb * Zc \geq K$, $6 * 4 = 24$, $24 > 20$ and this is the minimum Zc value from the above table that satisfies this condition.

- 4) After Zc is determined, the corresponding shift coefficient matrix set need to be selected from below Table., Ref for this table is [3GPP Specification 38.212,5.3.2](#)

Set Index(iLS)	Set of Lifting Sizes(Zc)
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	11,22,44,88,196
6	13,26,52,104,208
7	15,30,60,120,240

Since Zc = 4, Set Index (iLS) "0" is considered.

- 5) Determine the entries values in the base matrix based on the Zc, Calculate the shifting coefficient value P(i,j) by the modular Zc operation.

$$P(i,j) = f(V_{i,j}, Zc) = \text{mod}(V_{i,j}, Zc)$$

Referral tables to calculate P(i,j) are available in the specification [3GPP Specification 38.212,5.3.2](#)

H _{BG}	V _{i,j}	H _{BG}	V _{i,j}
Row index i	Column index j	Row index i	Column index j
	Set index i _{LS}		Set index i _{LS}
0	0 1 2 3 4 5 6 7	0	0 1 2 3 4 5 6 7
1	260 307 73 223 211 284 0 135	1	96 2 280 120 0 348 6 138
2	69 19 15 16 198 118 0 227	2	65 210 60 131 163 15 81 220
3	220 60 103 84 188 167 0 156	3	63 318 130 209 108 81 182 173
4	159 369 49 91 186 330 0 134	4	76 65 184 209 68 176 53 142
5	100 181 240 74 219 207 0 84	5	179 269 51 81 64 113 46 40
6	10 216 39 10 4 166 0 83	6	0 0 0 0 0 0 0 0
7	59 317 15 0 29 243 0 53	7	64 13 69 164 270 190 88 78
8	228 288 162 205 144 250 0 225	8	49 336 140 164 13 263 198 152
9	110 109 215 216 119 1 0 205	9	49 57 45 43 69 332 160 84
10	191 17 164 21 216 339 0 128	10	51 289 115 189 54 331 122 5
11	3 357 133 215 115 201 0 75	11	154 57 300 101 0 114 182 205
12	185 215 288 14 233 53 0 136	12	0 0 0 0 0 0 0 0
13	23 108 110 70 144 347 0 217	13	0 7 260 257 66 163 110 91 183
14	190 242 113 141 95 304 0 220	14	164 303 147 110 137 228 184 112
15	35 180 16 198 216 167 0 90	15	59 61 128 200 0 247 30 106
16	239 330 189 104 73 47 0 105	16	1 358 51 63 0 116 3 219
17	31 346 32 81 261 188 0 137	17	144 375 228 4 162 190 155 129
18	1 1 1 1 1 1 0 1	18	0 0 0 0 0 0 0 0
19	0 0 0 0 0 0 0 0	19	42 130 260 199 161 47 1 183

Fig. 4: LDPC base graph 1 (BG 1) and its parity check matrices

H _{BG}	V _{i,j}	H _{BG}	V _{i,j}
Row index i	Column index j	Row index i	Column index j
	Set index i _{LS}		Set index i _{LS}
0	0 1 2 3 4 5 6 7	16	0 1 2 3 4 5 6 7
1	0 9 174 0 72 3 156 143 145	17	26 0 0 0 0 0 0 0
2	117 97 0 110 26 143 19 131	18	264 158 0 48 120 134 57 198
3	204 166 0 23 53 14 176 71	19	5 124 23 24 132 43 23 201 173
4	26 66 0 181 35 3 165 21	20	11 114 9 109 206 85 62 142 195
5	189 71 0 95 115 40 196 23		12 64 6 18 2 42 163 35 218
6	205 172 0 8 127 123 13 112		27 0 0 0 0 0 0 0
7	0 0 0 0 0 0 0 0		0 220 186 0 88 17 173 129 128
8	167 27 137 53 19 17 18 142		6 194 5 19 16 105 31 203 211
9	166 36 124 156 64 65 27 174		28 0 0 0 0 0 0 0
10	253 48 0 116 104 63 3 183		0 57 58 0 35 79 13 110 39
11	125 92 0 156 66 1 102 27		1 20 42 158 138 28 135 124 84
12	226 31 88 115 84 55 185 96		10 185 156 154 86 41 145 52 88
13	158 187 0 200 98 37 17 23		29 0 0 0 0 0 0 0
14	224 185 0 29 69 171 14 6		1 26 76 0 6 2 128 196 117
15	252 3 55 31 50 133 180 167		4 105 61 148 20 103 62 35 227
16	0 0 0 0 0 0 0 0		11 29 153 104 141 78 173 114 6
17	0 0 0 0 0 0 0 0		30 0 0 0 0 0 0 0
18	0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0

Fig. 5: LDPC base graph 2 (BG 2) and its parity check matrices

For K=20 , Base Graph = 2, Zc= 4 and SetIndex iLS = 0, from above Table LDPC base graph 2 Using the equation $P(i,j) = f(V_{i,j}, z) = \text{mod}(V_{i,j}, z)$ all the possible base graph matrix entries with the shifting coefficient are determined.

Below I have illustrated how the Base Matrix entries P(i,j) are populated, I have considered only the first row for below illustration, like wise the full matrix is built.

H _{BG}	V _{i,j}
Row index i	Column index j
	Set index i _{LS}
0	0 1 2 3 4 5 6 7
1	0 9 174 0 72 3 156 143 145
2	117 97 0 110 26 143 19 131
3	204 166 0 23 53 14 176 71
4	26 66 0 181 35 3 165 21
5	189 71 0 95 115 40 196 23
6	205 172 0 8 127 123 13 112
7	0 0 0 0 0 0 0 0
8	0 0 0 0 0 0 0 0
9	0 0 0 0 0 0 0 0
10	0 0 0 0 0 0 0 0
11	0 0 0 0 0 0 0 0

Fig. 6: LDPC parity check matrix selection

From Step 3 and 4 Set Index iLS = 0 and Zc =4

1st entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "0" (0) under SetIndex "0" = 9, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 9 \text{ mod } 4 = 1$
2nd entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "1" (1) under SetIndex "0" = 117, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 117 \text{ mod } 4 = 1$
3rd entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "2" (2) under SetIndex "0" = 204, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 204 \text{ mod } 4 = 0$
4th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "3" (3) under SetIndex "0" = 26, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 26 \text{ mod } 4 = 2$
5th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "4" (4) under SetIndex "0" = 189, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 189 \text{ mod } 4 = 1$
6th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "5" (5) under SetIndex "0" = 8, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 8 \text{ mod } 4 = 0$
7th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "6" (6) under SetIndex "0" = 127, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 127 \text{ mod } 4 = 3$
8th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "7" (7) under SetIndex "0" = 13, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 13 \text{ mod } 4 = 1$
9th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "8" (8) under SetIndex "0" = 205, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 205 \text{ mod } 4 = 1$
10th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "9" (9) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
11th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "10" (10) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
12th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "11" (11) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
13th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "12" (12) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
14th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "13" (13) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
15th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "14" (14) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
16th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "15" (15) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
17th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "16" (16) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
18th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "17" (17) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
19th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "18" (18) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
20th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "19" (19) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
21st entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "20" (20) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
22nd entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "21" (21) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
23rd entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "22" (22) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
24th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "23" (23) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
25th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "24" (24) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
26th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "25" (25) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
27th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "26" (26) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
28th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "27" (27) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
29th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "28" (28) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
30th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "29" (29) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
31st entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "30" (30) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
32nd entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "31" (31) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
33rd entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "32" (32) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
34th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "33" (33) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
35th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "34" (34) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
36th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "35" (35) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
37th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "36" (36) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
38th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "37" (37) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
39th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "38" (38) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
40th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "39" (39) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
41st entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "40" (40) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
42nd entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "41" (41) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
43rd entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "42" (42) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
44th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "43" (43) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
45th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "44" (44) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
46th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "45" (45) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
47th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "46" (46) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
48th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "47" (47) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
49th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "48" (48) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
50th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "49" (49) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
51st entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "50" (50) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
52nd entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "51" (51) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
53rd entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "52" (52) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
54th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "53" (53) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
55th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "54" (54) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
56th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "55" (55) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
57th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "56" (56) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
58th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "57" (57) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
59th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "58" (58) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
60th entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "59" (59) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
61st entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "60" (60) under SetIndex "0" = 0, now using the formula $P(i,j) = \text{mod}(V_{i,j}, Zc) = 0 \text{ mod } 4 = 0$
62nd entry of Row1 = Select a V_{i,j} value that corresponds to Row Index "0" (0) & Column Index "61" (61) under SetIndex "0" = 0, now using the

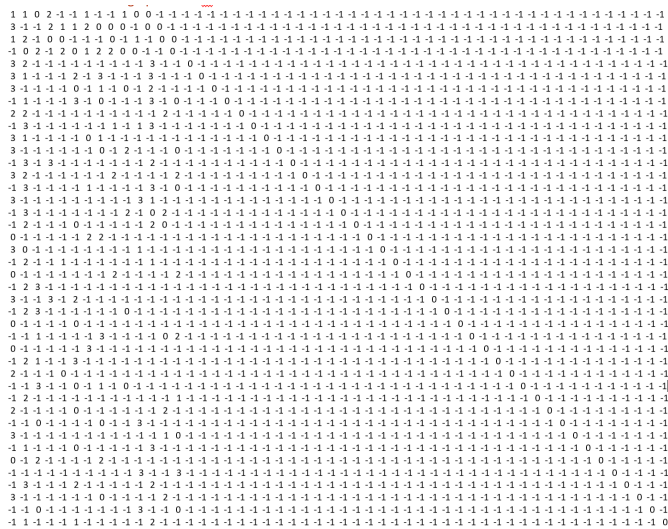


Fig. 8: Base Graph2 for iLS=0 and Zc=4

The above mentioned steps are used in the following program to generate base graphs based on K and N.

ldpcbasegraphgen.c

I have generated all the possible base graphs and are available in [BaseGraphs](#) section .

D. Encoding procedure

- a) First step is to generate any random message of size equal to lifting size (Z_c) which has obtained during the selection of base graph.
- b) Then Z_c should be multiplied with the original information bits i.e., difference between rows and columns for the selected base graph.

$$Z_c * (\text{Column Size} - \text{Row Size})$$

- c) After that generate the codeword. Which should be of size given below

$$Z_c * (\text{Column Size})$$