5G NR LDPC Encoder and Decoder

YADATI KRISHNA

Contents

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 Zc.

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

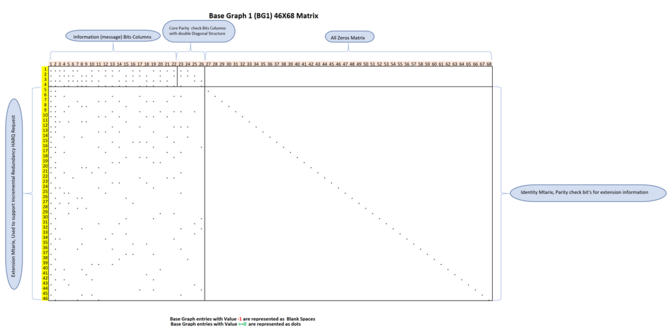

Fig. 1: High Level Visualization of fully Populated BG1 Matrix

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

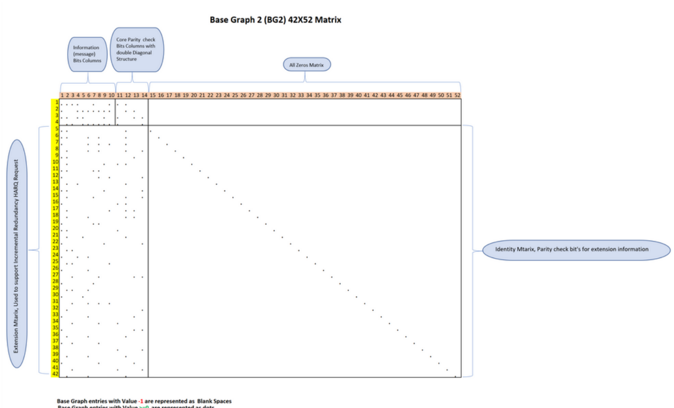

Fig. 2: High Level Visualization of fully Populated BG2 Matrix

C. LDPC Base Graph selection procedure

I have put together the following example of constructing the LDCP 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<=3824 and R<=0.67 then BG2 is selected.

   b) If K<= 292 then BG2 is selected

   c) if R<=0.25 then BG2 is selected.

   d) Else BG1 is selected

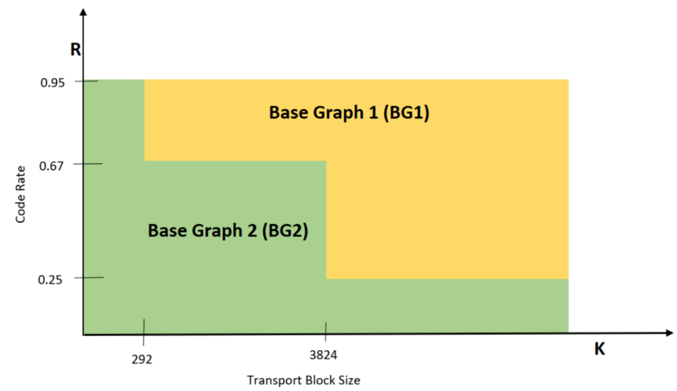

Fig. 3: Graphical Representation of Base Graph Selection

2) Determine the value of Kb for the given K (Transport Block) and R (Code Rate) Ref 3GPP Specification 38.212, 5.2.2
   Kb denotes the number of information bit columns for the lifting size Zc.

   As per the Specification

   For LDPC BG1:

   a) Kb = 22

   For LDPC BG2:

   a) if K is between 640 < K <= 3824 then Kb = 10

   b) if K is between 560 < K <= 640 then Kb = 9

   c) if K is between 192 < K <= 560 then Kb =8

   d) If K is <= 192 then Kb is = 6

3) Determine the base matrix expansion factor Zc by selecting the minimum Zc value in below Table,such that Kb*Zc >=K.Sets of LDPC lifting size Zc in the specification I have populated below Zc 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
j 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 >= 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.

$$P(i,j)= f(Vi,j\ ,\ Zc)= mod(Vi,j,\ Zc)$$

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

3GPP TS 38.212 version 15.3.0 Release 15 21 ETSI TS 138 212 V15.3.0 (2018-10)

$\mathbf{H}_{BG}$ Row index i	Column index j	$V_{i,j}$ Set index i_{LS} 0	1	2	3	4	5	6	7
0	0	250	307	73	223	211	294	0	135
	1	69	19	15	16	198	118	0	227
	2	226	50	103	94	188	167	0	126
	3	159	369	49	91	186	330	0	134
	5	100	181	240	74	219	207	0	84
	6	10	216	39	10	4	165	0	83
	9	59	317	15	0	29	243	0	53
	10	229	288	162	205	144	250	0	225
	11	110	109	215	216	116	1	0	205
	12	191	17	164	21	216	339	0	128
	13	9	357	133	215	115	201	0	75
	15	195	215	298	14	233	53	0	135
	16	23	106	110	70	144	347	0	217
	18	190	242	113	141	95	304	0	220
	19	35	180	16	198	216	167	0	90
	20	239	330	189	104	73	47	0	105
	21	31	346	32	81	261	188	0	137
	22	1	1	1	1	1	1	0	1
	23	0	0	0	0	0	0	0	0

$\mathbf{H}_{BG}$ Row index i	Column index j	$V_{i,j}$ Set index i_{LS} 0	1	2	3	4	5	6	7
15	1	96	2	290	120	0	348	6	138
	10	65	210	60	131	183	15	81	220
	13	63	318	130	209	108	81	182	173
	18	75	55	184	209	68	176	53	142
	25	179	269	51	81	64	113	46	49
	37	0	0	0	0	0	0	0	0
16	1	64	13	69	154	270	190	88	78
	3	49	338	140	164	13	293	198	152
	11	49	57	45	43	99	332	160	84
	20	51	289	115	189	54	331	122	5
	22	154	57	300	101	0	114	182	205
	38	0	0	0	0	0	0	0	0
17	0	7	260	257	56	153	110	91	183
	14	164	303	147	110	137	228	184	112
	16	59	81	128	200	0	247	30	106
	17	1	358	51	63	0	116	3	219
	21	144	375	228	4	162	190	155	129
	39	0	0	0	0	0	0	0	0
	1	42	130	260	199	161	47	1	183

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

3GPP TS 38.212 version 15.3.0 Release 15 24 ETSI TS 138 212 V15.3.0 (2018-10)

$\mathbf{H}_{BG}$ Row index i	Column index j	$V_{i,j}$ Set index i_{LS} 0	1	2	3	4	5	6	7
0	0	9	174	0	72	3	156	143	145
	1	117	97	0	110	26	143	19	131
	2	204	166	0	23	53	14	176	71
	3	26	66	0	181	35	3	165	21
	6	189	71	0	95	115	40	196	23
	9	205	172	0	8	127	123	13	112
	10	0	0	0	1	0	0	0	1
	11	0	0	0	0	0	0	0	0
1	0	167	27	137	53	19	17	18	142
	3	166	36	124	156	94	65	27	174
	4	253	48	0	115	104	63	3	183
	5	125	92	0	156	66	1	102	27
	6	226	31	88	115	84	55	185	96
	7	156	187	0	200	98	37	17	23
	8	224	185	0	29	69	171	14	9
	9	252	3	55	31	50	133	180	167
	11	0	0	0	0	0	0	0	0
	12	0	0	0	0	0	0	0	0

$\mathbf{H}_{BG}$ Row index i	Column index j	$V_{i,j}$ Set index i_{LS} 0	1	2	3	4	5	6	7
16	26	0	0	0	0	0	0	0	0
17	1	254	158	0	48	120	134	57	196
	5	124	23	24	132	43	23	201	173
	11	114	9	109	206	65	62	142	195
	12	64	6	18	2	42	163	35	218
	27	0	0	0	0	0	0	0	0
18	0	220	186	0	68	17	173	129	128
	6	194	6	18	16	106	31	203	211
	7	50	46	86	156	142	22	140	210
	28	0	0	0	0	0	0	0	0
19	0	87	58	0	35	79	13	110	39
	1	20	42	158	138	28	135	124	84
	10	185	156	154	86	41	145	52	88
	29	0	0	0	0	0	0	0	0
20	1	26	76	0	6	2	128	196	117
	4	105	61	148	20	103	52	35	227
	11	29	153	104	141	78	173	114	6
	30	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(Vi,j , z) = mod(Vi,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.

$\mathbf{H}_{BG}$ Row index i	Column index j	$V_{i,j}$ Set index i_{LS} 0	1	2	3	4	5	6	7
0	0	9	174	0	72	3	156	143	145
	1	117	97	0	110	26	143	19	131
	2	204	166	0	23	53	14	176	71
	3	26	66	0	181	35	3	165	21
	6	189	71	0	95	115	40	196	23
	9	205	172	0	8	127	123	13	112
	10	0	0	0	1	0	0	0	1
	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 Vi,j value that corresponds to Row Index "i" (0)& Column Index "j" (0) under SetIndex "0" = 9 , now using the formula P(i,j) = mod(Vi,j , Zc) = 9 mod 4 = 1
2nd entry of Row1 = Select a Vi,j value that corresponds to Row Index "i" (0)& Column Index "j" (1) under SetIndex "0" = 117 , now using the formula P(i,j) = mod(Vi,j , Zc) = 117 mod 4 = 1
3rd entry of Row1 = Select a Vi,j value that corresponds to Row Index "i" (0)& Column Index "j" (2) under SetIndex "0" = 204 , now using the formula P(i,j) = mod(Vi,j , Zc) = 204 mod 4 = 0
4th entry of Row1= Select a Vi,j value that corresponds to Row Index "i" (0)& Column Index "j" (3) under SetIndex "0" = 26 , now using the formula P(i,j) = mod(Vi,j , Zc) = 26 mod 4 = 2
5th entry of Row1 = Select a Vi,j value that corresponds to Row Index "i" (0)& Column Index "j" (not available) under SetIndex "0" = -1
6th entry of Row1= Select a Vi,j value that corresponds to Row Index "i" (0)& Column Index "j" (not available) under SetIndex "0" = -1
7th entry of Row1 = Select a Vi,j value that corresponds to Row Index "i" (0)& Column Index "j" (6) under SetIndex "0" = 189 , now using the formula P(i,j) = mod(Vi,j , Zc) = 189 mod 4 = 1
8th entry of Row1= Select a Vi,j value that corresponds to Row Index "i" (0)& Column Index "j" (not available) under SetIndex "0" = -1
9th entry of Row1= Select a Vi,j value that corresponds to Row Index "i" (0)& Column Index "j" (not available) under SetIndex "0" = -1
10th entry of Row1= Select a Vi,j value that corresponds to Row Index "i" (0)& Column Index "j" (9) under SetIndex "0" = 205 , now using the formula P(i,j) = mod(Vi,j , Zc) = 205 mod 4 = 1
11th entry of Row1= Select a Vi,j value that corresponds to Row Index "i" (0)& Column Index "j" (10) under SetIndex "0" = 0 , now using the formula P(i,j) = mod(Vi,j , Zc) = 0 mod 4 = 0
12th entry of Row1= Select a Vi,j value that corresponds to Row Index "i" (0)& Column Index "j" (11) under SetIndex "0" = 0 , now using the formula P(i,j) = mod(Vi,j , Zc) = 0 mod 4 = 0
13th entry of Row1= Select a Vi,j value that corresponds to Row Index "i" (0)& Column Index "j" (not available) under SetIndex "0" = -1
14th entry of Row1= Select a Vi,j value that corresponds to Row Index "i" (0)& Column Index "j" (not available) under SetIndex "0" = -1
.
.
.
52nd entry of Row1= Select a Vi,j value that corresponds to Row Index "i" (0)& Column Index "j" (not available) under SetIndex "0" = -1

Row#1 entries = 1 1 0 2 -1 -1 1 -1 -1 1 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 -1 -1 -1 -1 -1 -1 -1 -1 -1

Fig. 7: modulus operation

Using the above principal remaining all entries of the matrix can be populated based on the Row index i and Column Index j and SetIndex(iLS)
I populated the below full base graph2 matrix with Zc=4

1	1	0	2	-1	-1	1	-1	-1	1	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	-1	-1	-1	-1	-1	-1	-1	-1	-1
3	-1	-1	2	1	1	2	0	0	0	-1	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	-1	-1	-1	-1	-1	-1	-1	-1
1	2	-1	0	0	-1	-1	-1	0	-1	1	-1	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	-1	-1	-1	-1	-1	-1	-1
-1	0	2	-1	2	0	1	2	2	0	0	-1	-1	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	-1	-1	-1	-1	-1	-1	-1
3	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	3	-1	-1	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	-1	-1	-1	-1	-1	-1
3	1	-1	-1	-1	2	-1	3	-1	-1	-1	3	-1	-1	-1	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	-1	-1	-1	-1	-1
3	-1	-1	-1	-1	0	-1	1	-1	0	-1	2	-1	-1	-1	-1	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	-1	-1	-1	-1
-1	1	-1	-1	-1	3	-1	0	-1	-1	-1	3	-1	0	-1	-1	-1	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	-1	-1	-1
2	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	2	-1	-1	-1	-1	-1	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	-1	-1
-1	3	-1	-1	-1	-1	-1	-1	1	-1	1	3	-1	-1	-1	-1	-1	-1	-1	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	-1
3	1	-1	-1	-1	-1	0	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	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
3	-1	-1	-1	-1	-1	-1	0	-1	2	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	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	3	-1	3	-1	-1	-1	-1	-1	-1	-1	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	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
3	2	-1	-1	-1	-1	-1	-1	2	-1	-1	-1	-1	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	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	3	-1	-1	-1	-1	1	-1	-1	-1	-1	3	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	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
3	-1	-1	-1	-1	-1	-1	-1	-1	-1	3	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	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	3	-1	-1	-1	-1	-1	-1	-1	2	-1	0	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	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	2	-1	-1	-1	0	-1	-1	-1	-1	-1	2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	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
0	-1	-1	-1	-1	-1	2	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
3	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	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	2	-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	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	-1	-1	-1	-1	-1	-1	-1	2	-1	-1	-1	-1	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	2	3	-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	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
3	-1	-1	3	-1	2	-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	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	2	3	-1	-1	-1	-1	-1	-1	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	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	-1	-1	-1	-1	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	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	1	-1	-1	-1	-1	3	-1	-1	-1	-1	0	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	-1	-1	-1	-1	-1	3	-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	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	2	1	-1	-1	3	-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	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
2	-1	-1	-1	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	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	3	-1	-1	0	-1	1	-1	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	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	2	-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	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
2	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	2	-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	0	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	0	-1	-1	-1	-1	0	-1	-1	3	-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	-1	0	-1	-1	-1	-1	-1	-1	-1	-1
3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	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	0	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	0	-1	-1	-1	-1	-1	3	-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	-1	-1	0	-1	-1	-1	-1	-1	-1
0	-1	2	-1	-1	-1	-1	2	-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	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	3	-1	-1	3	-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	-1	-1	0	-1	-1	-1	-1
-1	3	-1	-1	-1	2	-1	-1	-1	-1	-1	2	-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	-1	-1	-1	-1	-1	0	-1	-1	-1
3	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	2	-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	-1	-1	-1	-1	-1	0	-1	-1
-1	-1	0	-1	-1	-1	-1	-1	-1	-1	3	-1	-1	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	-1	-1	-1	-1	-1	0	-1
-1	1	-1	-1	-1	1	-1	-1	-1	-1	-1	2	-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	-1	-1	-1	-1	-1	-1	-1	-1	0

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 posible base graphs and are available in BaseGraphs section .

D. *Encoding Algorithm*

Let the codeword be

C=[s_1,s_2,....s_{k_b},p_{b_1},p_{b_2},p_{b_3},p_{b_4},p_{c_1},p_{c_2},.....,$p_{c_{m_b-4}}$]

where each element of C is a vector of length Zc.
The following figure represents the structure of base graph.

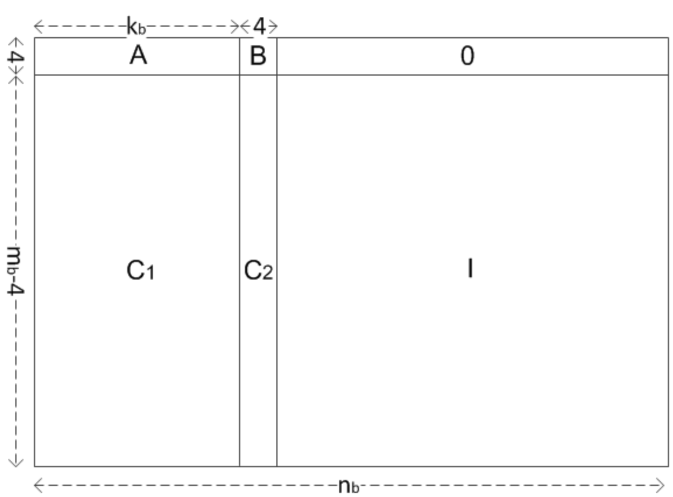

Fig. 9: Structure of Base-Garph

where

$$A=\begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,k_b} \\ a_{2,1} & a_{2,2} & \dots & a_{2,k_b} \\ a_{3,1} & a_{3,2} & \dots & a_{3,k_b} \\ a_{4,1} & a_{4,2} & \dots & a_{4,k_b} \end{bmatrix}$$

$$C_1=\begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,k_b} \\ c_{2,1} & c_{2,2} & \dots & c_{2,k_b} \\ c_{3,1} & c_{3,2} & \dots & c_{3,k_b} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m_b-4,1} & c_{m_b-4,2} & \dots & c_{m_b-4,k_b} \end{bmatrix}$$

$$C_2=\begin{bmatrix} c_{1,k_b+1} & c_{1,k_b+2} & c_{1,k_b+3} & c_{1,k_b+4} \\ c_{2,k_b+1} & c_{2,k_b+2} & c_{2,k_b+3} & c_{2,k_b+4} \\ c_{3,k_b+1} & c_{3,k_b+2} & c_{3,k_b+3} & c_{3,k_b+4} \\ \vdots & \vdots & \vdots & \vdots \\ c_{m_b-4,k_b+1} & c_{m_b-4,k_b+2} & c_{m_b-4,k_b+3} & c_{m_b-4,k_b+4} \end{bmatrix}$$

There are two types of B i.e.,
$B \in \{H_{BG1\_B1}, H_{BG1\_B2}, H_{BG2\_B1}, H_{BG2\_B2}\}$ in both BG1 and BG2.

$$H_{BG1\_B1}=\begin{bmatrix} 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & -1 \\ -1 & -1 & 0 & 0 \\ 1 & -1 & -1 & 0 \end{bmatrix}$$

$$H_{BG1\_B2}=\begin{bmatrix} 0 & 0 & -1 & -1 \\ 105 & 0 & 0 & -1 \\ -1 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 \end{bmatrix}$$

H_{BG1_B1} is for Zc set index iLS=(0,1,2,3,4,5,7), $H_{BG1_{B}2}$ is for iLS=(6).

$$H_{BG2\_B1}=\begin{bmatrix} 0 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 \end{bmatrix}$$

$$H_{BG2\_B2}=\begin{bmatrix} 1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 \\ 1 & -1 & -1 & 0 \end{bmatrix}$$

H_{BG2_B1} is for Zc set index iLS=(0,1,2,4,5,6), $H_{BG2_{B}2}$ is for iLS=(3,7).

The encoding of LDPC codes is carried out as follows:

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

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

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

$$\boxed{\text{Zc} * \text{Column Size}}$$