CT-216

LDPC DECODING FOR 5G NR

Group: 24

Professor : Yash Vasavad
TA : Darpan Lunagariya

HONOR CODE

We declare that:

- The work that we are presenting is our own work.
- We have not copied the work (the code, the results, etc.) that someone else has done.
- Concepts, understanding and insights we will be describing are our own.
- We make this pledge truthfully. We know that violation of this solemn pledge can carry grave consequences.

Team Members:

Rishik Yalamanchili (202301258)

Chirag katkoriya (202301259)

Mahek Jikkar (202301260)

Patel Nakul (202301261)

Priyanka garg (202301262)

Yug Savalia (202301263)

Krish Patel (202301264)

Jalu Risabh (202301265)

Vansh Vora (202301266)

Arav Vaitha (202301267)

Siddhant Shekar (202301268)

Signatures

INTRODUCTION

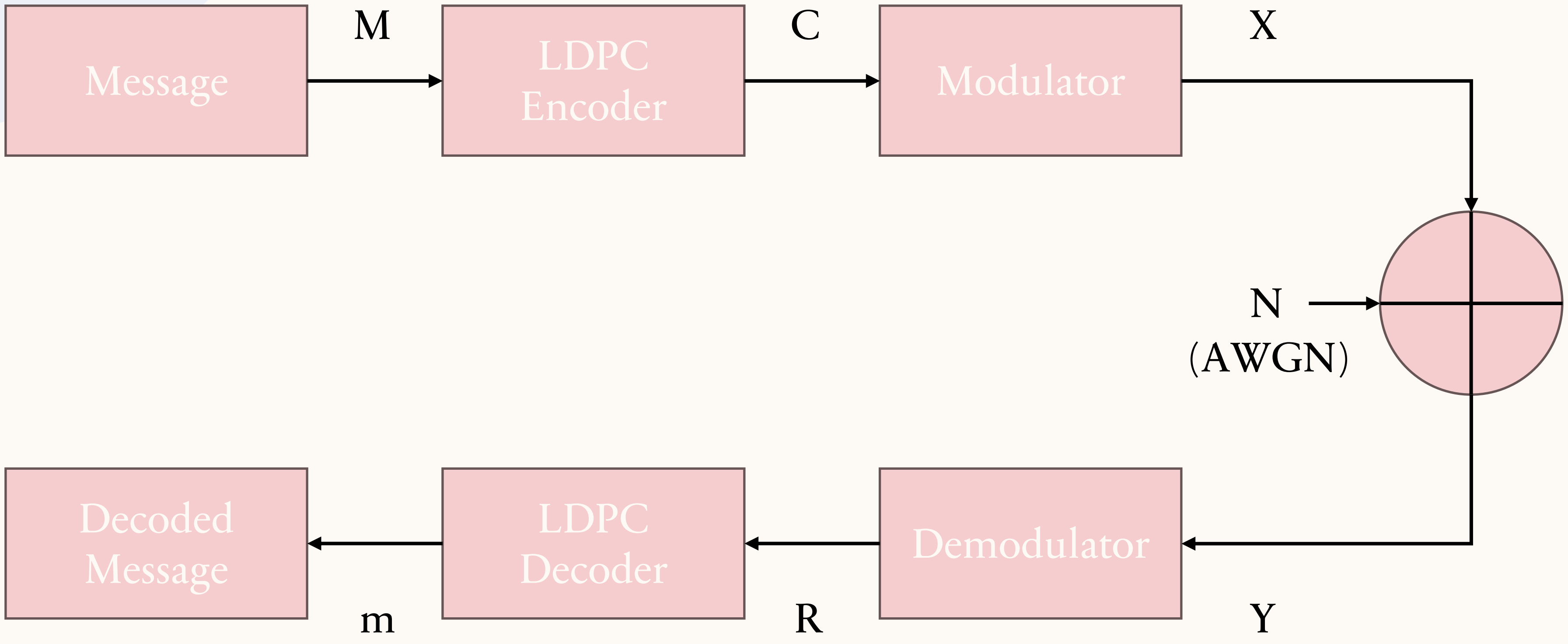
ABOUT LDPC CODES

- LDPC codes can be used in error correction of transmission errors which may occur through varies stages of communication, mainly reducing the errors brought about by noise by the channel.
- The LDPC codes can provide a channel capacity close to the theoretical Shannon's Channel Capacity Limit.
- These codes were proposed by R.G. Gallager in 1962, but were not applied since the decoding techniques were too complex for practical implementation.
- Since the rediscovery of the codes by David MacKay, they were introduced in a wide range of communication devices, both wired and wireless.
- Currently, the most significant use of LDPC codes is in the new generation of wireless communications standards called 5G New Radio (5G-NR).

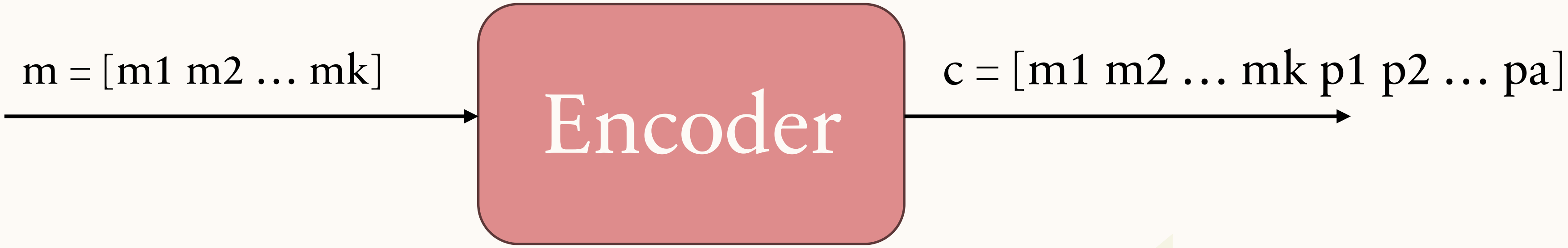
INTRODUCTION

- Low Density Parity Check (LDPC) codes are a class of linear block codes with a sparse parity check matrix H i.e. it contains small number of ones per row or column.
- LDPC code has a block length n with a mxn parity check matrix where each column contains a small fixed number of ones (wc = parity constraints) and each row contains (parity bits) associated with each Wc
- Wc >= 3 and each row contains Wr >= Wc number of ones.
- Low Density implies that Wc << m and Wr << n.
- Number of ones in the parity check matrix H= Wc*n = Wr*m.

FLOW DIAGRAM



ENCODING


m = [m1 m2 … mk]
Encoder
c = [m1 m2 … mk p1 p2 … pa]

Encoding Introduction

For encoding first we convert the non-binary BG to binary H matrix through **blow up** or **expansion.**

- ➢ Expansion Factor (Zc)
  - iLS : 0,1,2,3,4,5,6,7
  - a : 2,3,5,7,9,11,13,15
  - Ja : 7,7,6,5,5,5,4,4
  - Zc : a x 2^j , j=0,1,2,...Ja
- ➢ For each base matrix entry specified -1, 0, 1, ..., Zc -1.
  - -1 : Zc x Zc Null Matrix
  - I in [0, Zc -1] : Identity shifted right i times.

5G NR Base Graph Matrix And Protograph

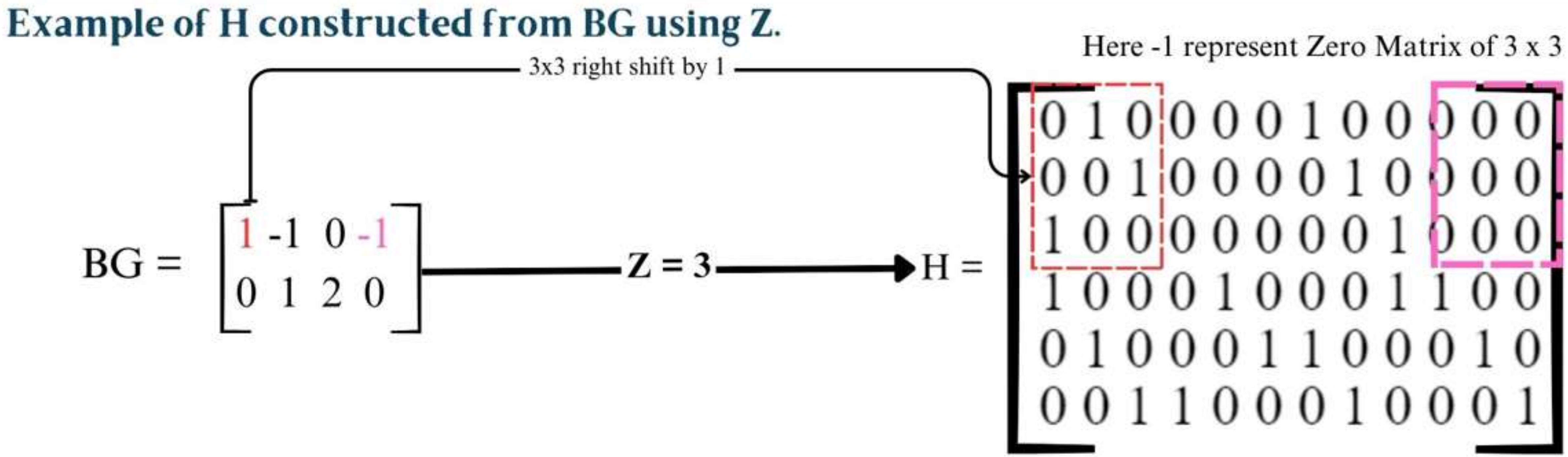
« **Protograph** : LDPC Codes parity-check matrix **H** is constructed from **BG** (base graph matrix)by using lifting size (expansion factor (Z)).

« The following steps are performed to construct the code.

« Base Graph Matrix (BG)

« Lifting size / Expansion factor (Z)

« Rate Matching

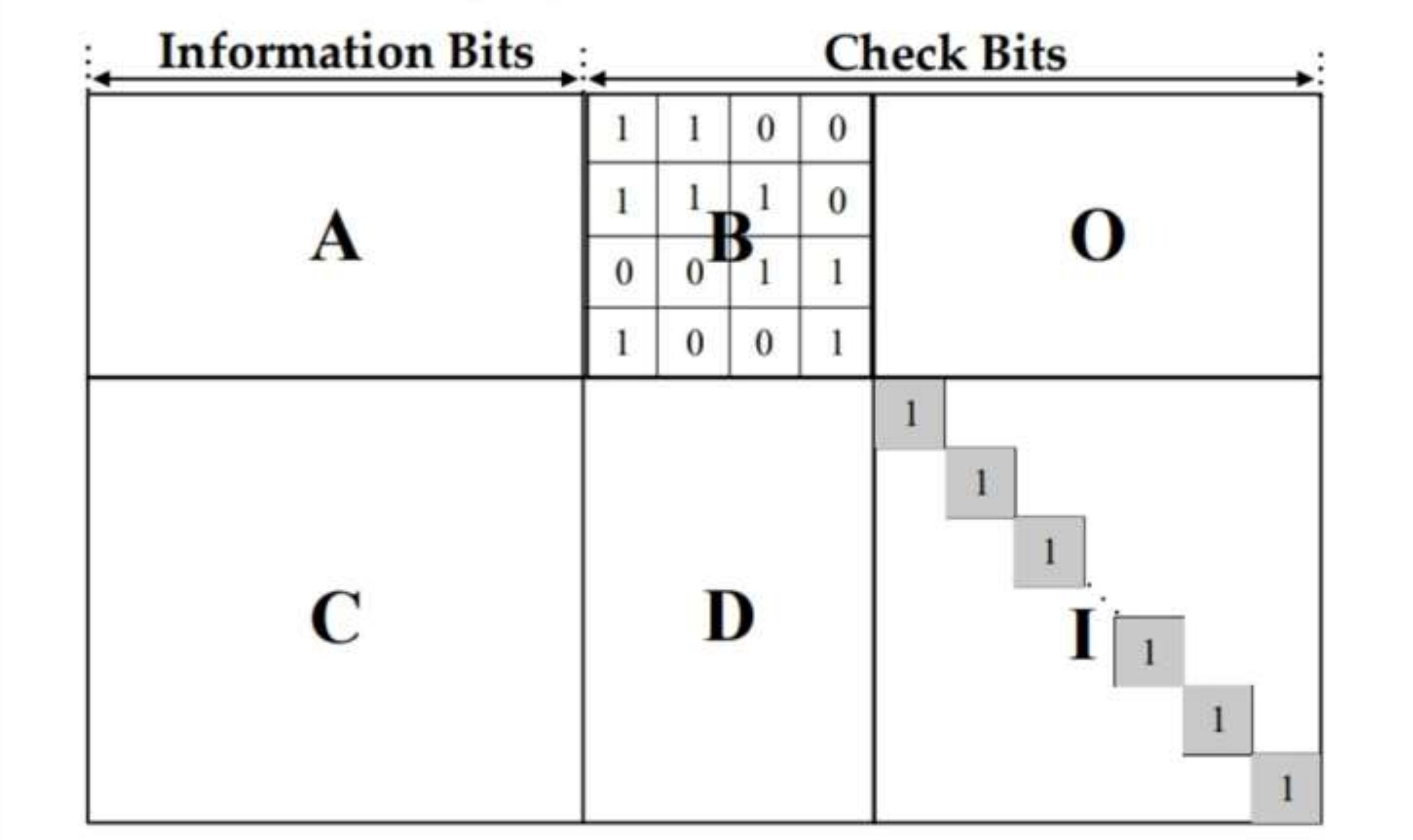


Base Graph Matrix

Let's take two non-binary base matrices

- BG1 : 46x 68
- BG2 : 42x 52

Base Matrix is a proper Block Like Structure



Characteristics of Block

- **A** : High dense part
- **B** : Double diagonal structure
- **O** : Zero matrix
- **C** : Low dense part
- **D** : Low dense part
- **I** : Identity matrix

For BG1,

A:4x22 , B:4x4 , O:4x42,

C:42x22 , D:42x4 ,I: 42x42

For BG2,

A:4x10 , B:4x4 , O:4x38,

C:38x10 , D:38x4 , I: 38 x 38

*This example is taken from, Andrew Thangaraj. Ldpc and polar codes in the 5g standard, 2019.

Base Matrix to H Matrix

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

Expansion factor: 5

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

*This example is taken from, Andrew Thangaraj. Ldpc and polar codes in the

Encoding Algorithm

First we expand the Base matrix

For all entries of B[i][j]:

if(B[i][j] == -1)

Replace B[i][j] by all zeros matrix of Zc x Zc

Else

Replace B[i][j] by shifted identity matrix of Zc x Zc by B[i][j]

- a = row(H) = n-k
- c = [m1 m2 ... mk p1 p2 ... Pa]
- We want to find p1p2 … Pa
- We multiply with H matrix $\mathbf{H} \times \begin{bmatrix} m^T \\ p^T \end{bmatrix} = \mathbf{0}$
- After that using double diagonal structure we find parity bits

Using this double diagonal structure it is easy to find parity.

I_k: identity matrix column shifted k times

Message m = [m1 m2 m3 m4]

where every mi has 5 bits

Codeword c = [m1 m2 m3 m4 p1 p2 p3 p4]

where each pi and mi has 5 bits

p1 p2 p3 and p4 are unknown

$$\left[\begin{array}{cccc|cccc} I_1 & 0 & I_3 & I_1 & I_2 & I & 0 & 0 \\ I_2 & I & 0 & I_3 & 0 & I & I & 0 \\ 0 & I_4 & I_2 & I & I_1 & 0 & I & I \\ I_4 & I_1 & I & 0 & I_2 & 0 & 0 & I \end{array}\right]$$

H matrix with Z_c: 5

*This example is taken from, Andrew Thangaraj. Ldpc and polar codes in the 5g standard, 2019.

- Equations obtained:
  - $I_1v_1 + I_3v_3 + I_1v_4 + I_2v_5 + Iv_6 = 0$ -Equation 1
  - $I_2v_1 + Iv_2 + I_3v_4 + Iv_6 + Iv_7 = 0$ -Equation2
  - $I_4v_2 + I_2v_3 + Iv_4 + I_1v_5 + Iv_7 + Iv_8 = 0$ -Equation 3
  - $I_4v_1 + I_1v_2 + Iv_3 + I_2v_5 + Iv_8 = 0$ -Equation 4
- Adding all four equations:
  - $I_1v_5 = I_1v_1 + I_2v_1 + I_4v_1 + Iv_2 + I_4v_2 + I_1v_2 + I_3v_3 + I_2v_3 + Iv_3 + I_1v_4 + I_3v_4 + Iv_4$
- We know that $[v_1\ v_2\ v_3\ v_4\ v_5\ v_6\ v_7\ v_8\ v_9] = [m_1\ m_2\ m_3\ m_4\ m_5\ p_1\ p_2\ p_3\ p_4]$
- So, rewriting the equation, we get:
- $I_1p_1 = I_1m_1 + I_2m_1 + I_4m1 + Im_2 + I_4m_2 + I_1m_2 + I_3m_3 + I_2m_3 + Im_3 + I_1m_4 + I_3m_4 + Im_4$
- From here we can get p_1.
- Substitute it in equation1 to get p_2 and solve so on to get all the 4 parity bits
- The encoded message must satisfy the equality H*c = 0.

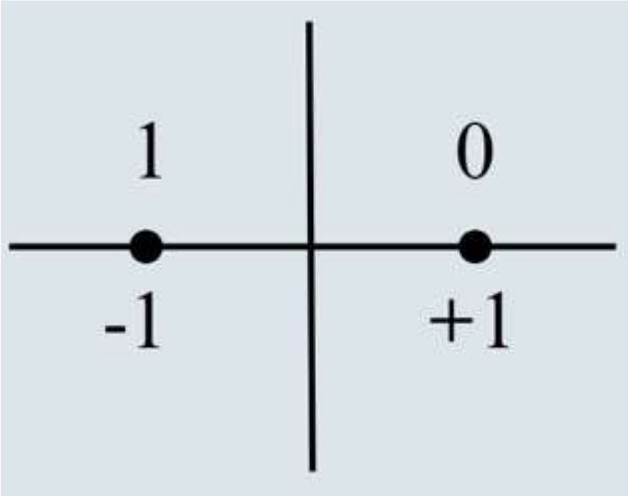
$$\begin{bmatrix} I_1 & 0 & I_3 & I_1 & I_2 & I & 0 & 0 \\ I_2 & I & 0 & I_3 & 0 & I & I & 0 \\ 0 & I_4 & I_2 & I & I_1 & 0 & I & I \\ I_4 & I_1 & I & 0 & I_2 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \end{bmatrix}$$

*This example is taken from, Andrew Thangaraj. Ldpc and polar codes in the 5g standard, 2019.

MODULATION

BPSK MODULATOR

- In **BPSK (Binary Phase Shift Keying)** , encoded bits are transformed into **BPSK** symbols before transmission. This can be achieved by representing the bits (ones and zeros) by shifting the phase of the carrier wave (codeword bits).
- The encoded codeword bit is mapped to a **BPSK** symbol using the mapping, **s =1-2c**, where s represents the **BPSK** symbol and c represents the codeword bit.
- Specifically, a bit is represented by shifting the phase of the carrier wave by 180° (pi radians) for a binary 1, and keeping it unchanged for a binary 0.

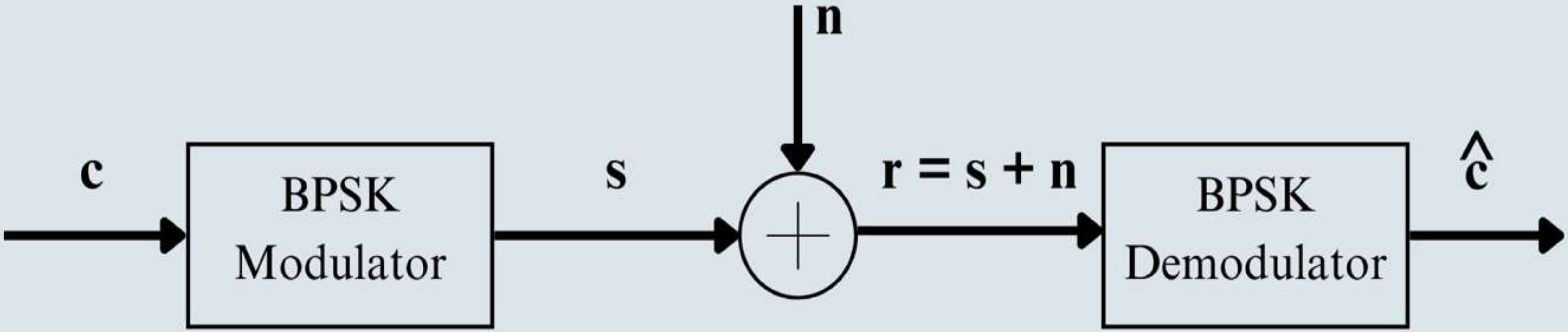


AWGN (ADDITIVE WHITE GAUSSIAN NOISE) CHANNEL

- The **BPSK** symbols are transmitted through an **AWGN** channel, which adds noise to the signal. The noise power is determined by the **SNR** (Signal-to-Noise Ratio) parameter, represented by y = Es/N0.
- Since **BPSK** symbols have energy, Es=1 Joule, the noise power is 1/y . The noise, n is generated as a Gaussian random variable with zero mean and variance is calculated per **SNR**.
- Thus, the received bits are represented by, **r =s + n**.

BPSK Demodulator

- To recover the original encoded bits, the **BPSK** demodulator makes decision by mapping received values to bits (ones and zeros) by drawing threshold at zero.
- This works as: If the received value is positive, then it will be mapped to 0, and negative received value will be mapped to 1.



Rate Matching

What is Rate Matching and Why is it needed?

- Rate matching is the process of selecting a specific number of bits from the parity matrix in order to meet the needs of transmmision over a channel with fixed resources.
- Generally there are two ways to perform rate-matching: Shortening (or Zero Padding) and Puncturing of bits.

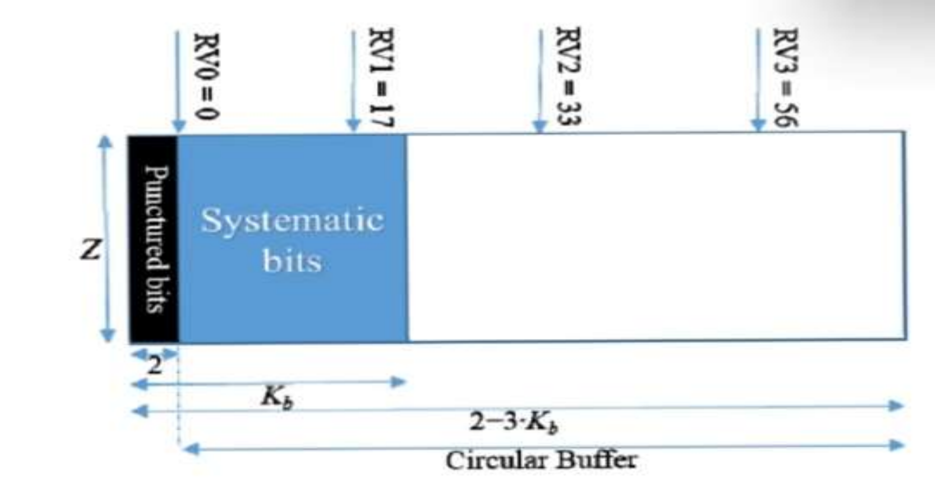
Rate Matching in LDPC 5G-NR Codes:

- For NR LDPC codes, rate matching is done by selecting bits from a Circular Buffer based on a Redundancy Version index.

Rate Matching: Redundancy Versions

In NR, the fixed locations of Redundancy versions {0,1,2,3} are defined at

Multiples of Z	BG1	BG2
RV0	0	0
RV1	17	13
RV2	33	25
RV3	56	43



- RV0 is self-decodable (decoding over a single transmission) for the largest range of code-rates, compared to all other RVs.
- Here RV3's are placed non-uniformly (towards the end of the matrix) to allow more overlap with the Information Bits.
- The values of RV's follow 3GPP (3rd Generation Partnership Project) Standards.

*This example is taken from, Andrew Thangaraj. Ldpc and polar codes in the 5g standard, 2019.

Rate Matching: Circular Buffer

- A storage mechanism where information and parity bits are stored and arranged temporarily before transmission.
- Arrangement (Reading) of bits during transmission is done on basis of RV's (the starting point for reading of bits).
- The term circular means that if the end of the buffer is reached during transmission, the reading of bits can wrap around to the beginning (continue form 2z+1 th bit).

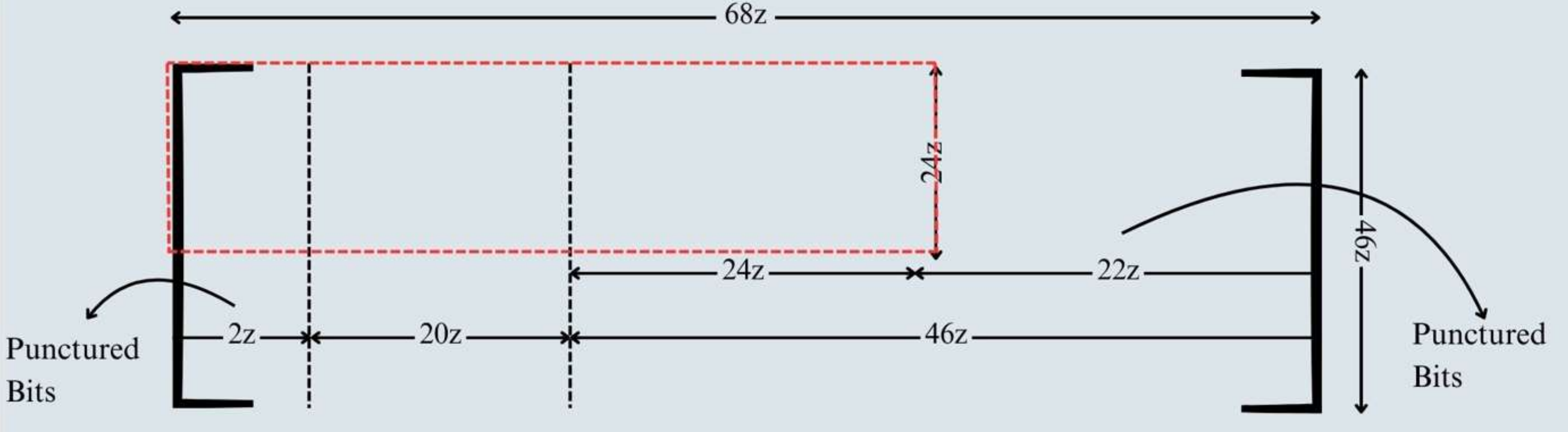
Rate Matching : An illustration for RV0 and BG1

- To achieve a code rate of r=1/2

We have 22z information bits, and 44z (excluding the first 2z bits) need to be transmitted.

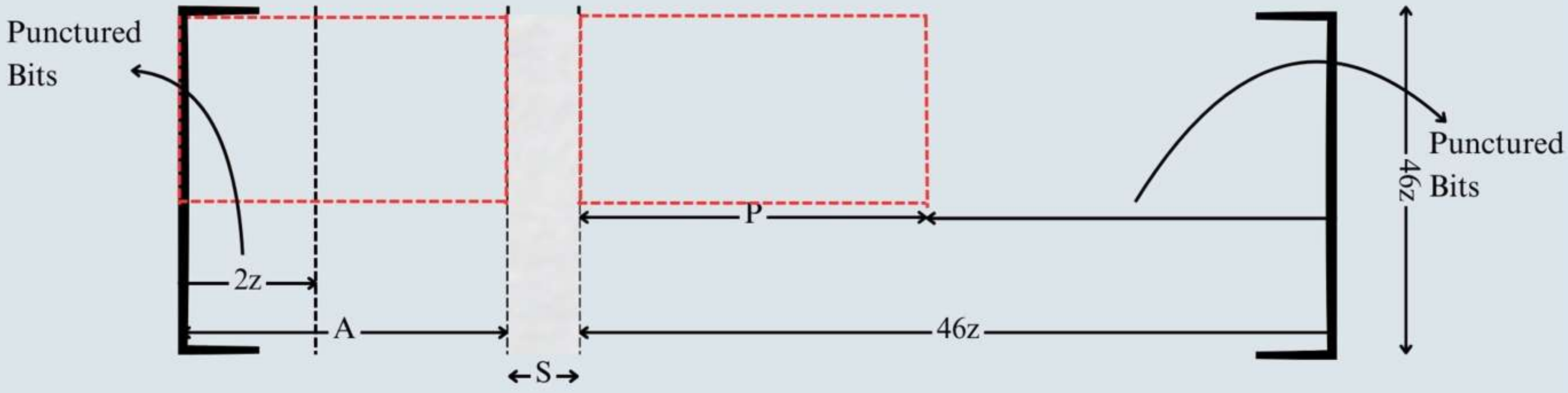
Hence we need to transmit : 20z message bits + 24z Parity Bits .

Therefore, the final dimensions of the H matrix will be 24x46 (Dotted Box) Punctured and Shortened bits are simply disregarded during encoding.

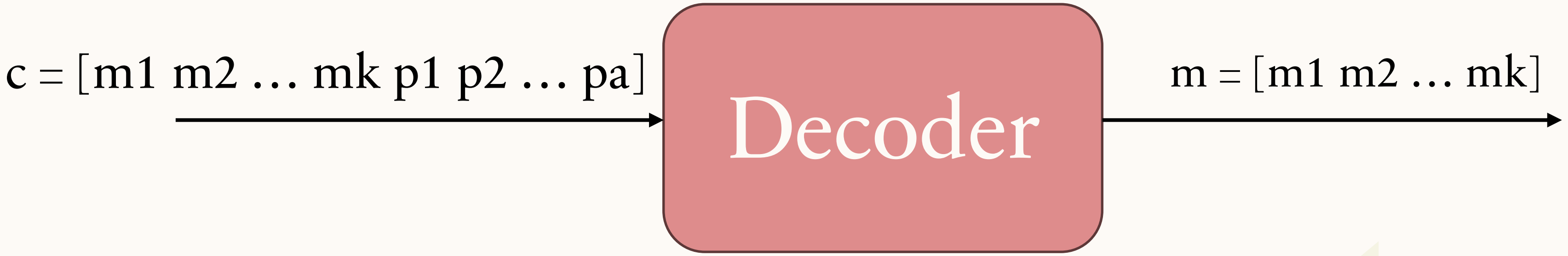


Rate Matching : Illustration for shortening of bits

- If only A out of 22z information bits are available, then the rest **S= 22z-A** bits are set to zero(Shortened).
- If we need to transmit a total of E bits, then **P = E-(A-2Z)** parity bits will be selected.
- The punctured parity bits and shortened information bits are simply disregarded in the decoding process.



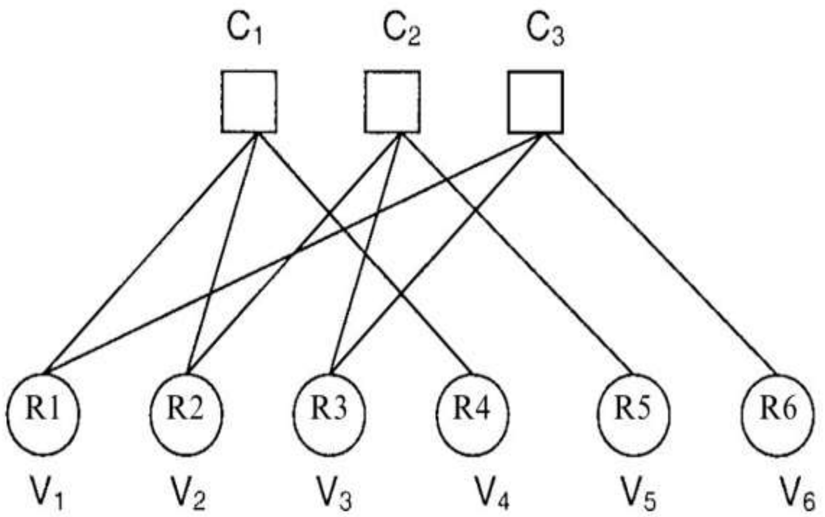
DECODING


c = [m1 m2 … mk p1 p2 … pa]
Decoder
m = [m1 m2 … mk]

Decoding Using Tanner Graph

This parity check matrix H can be represented by a Tanner Graph, which is a Bipartite Graph:

$$
H = \begin{array}{c} \begin{array}{cccccc} v1 & v2 & v3 & v4 & v5 & v6 \end{array} \\ \left[\begin{array}{cccccc} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right] \end{array} \begin{array}{c} \\ c1 \\ c2 \\ c3 \end{array}
$$



*This example is taken from, Andrew Thangaraj. Ldpc and polar codes in the 5g standard, 2019.

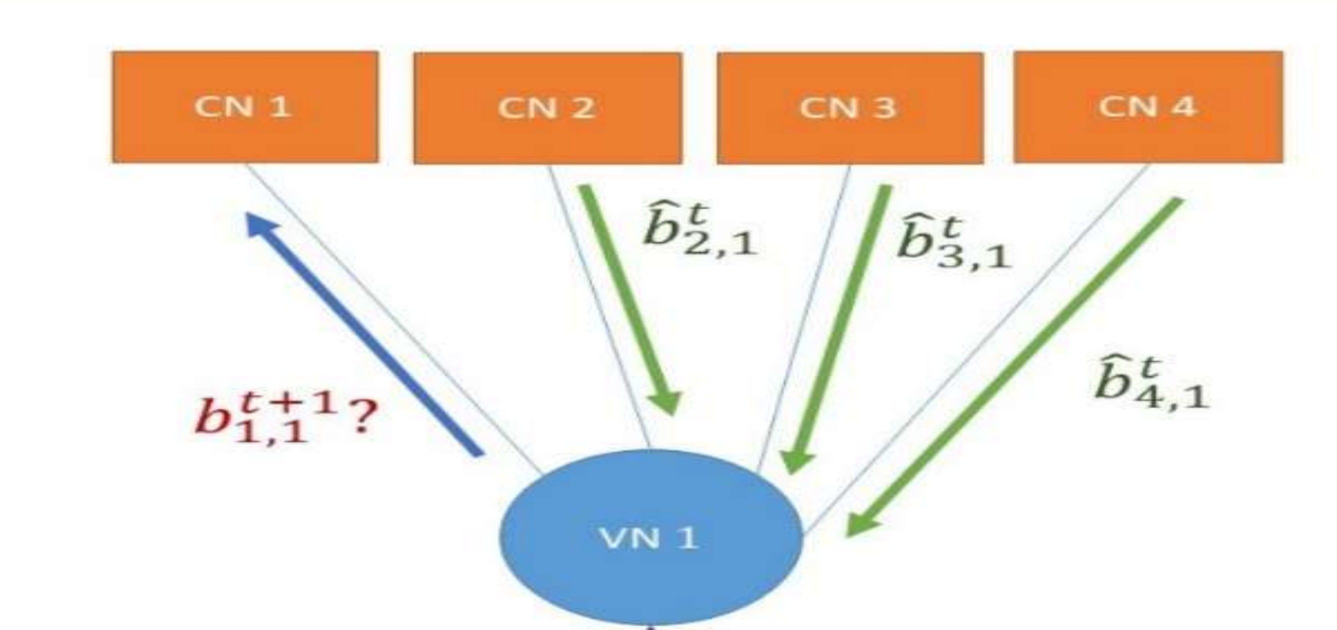
Hard Decision Decoding

Message Passing Iterative Decoding Algorithm

- The received bits are used to compute initial messages from check nodes (CNs) to variable nodes (VNs).
- Iteratively, messages are exchanged between CNs and VNs
- Each CN computes its outgoing messages based on incoming messages from connected VNs
- Similarly, each VN computes its outgoing messages based on incoming messages from connected CN.

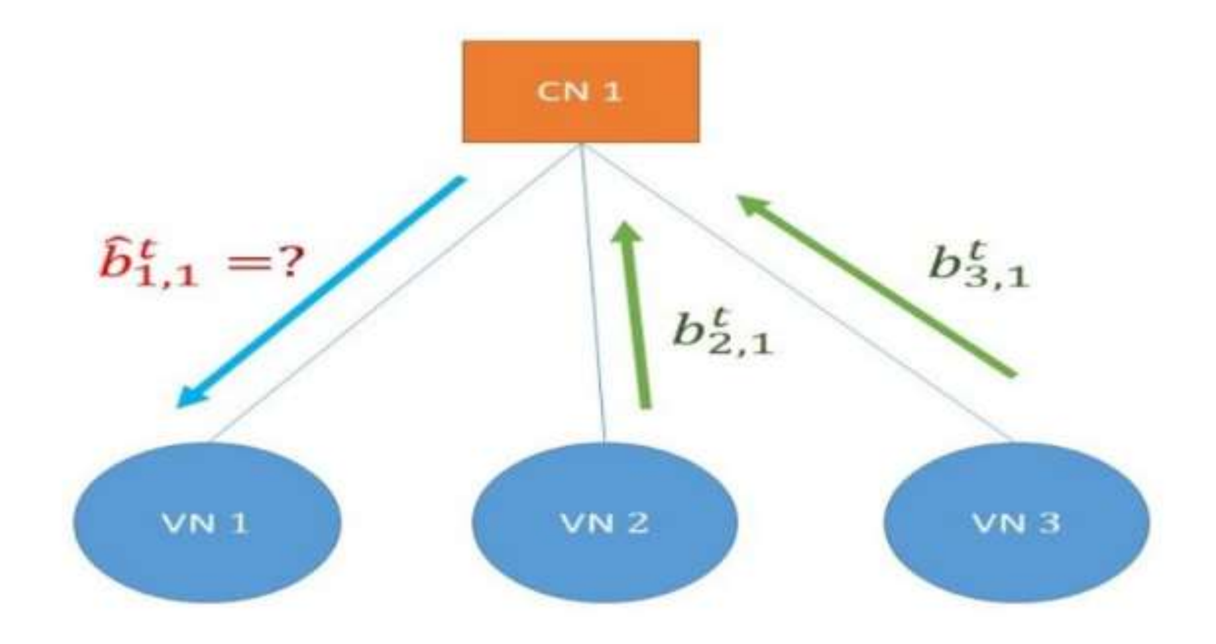
VN To CN

- First we load all the received values into the VN.
- Then, for first iteration:
  - We directly send the received value to all connected CNs of respective VNs.
- For all other iterations:
  - To send message to CN i, we take all the messages received to VN j and the bit it had originally, except value sent by CN i. Then we perform majority voting of these messages and send it to CN i.

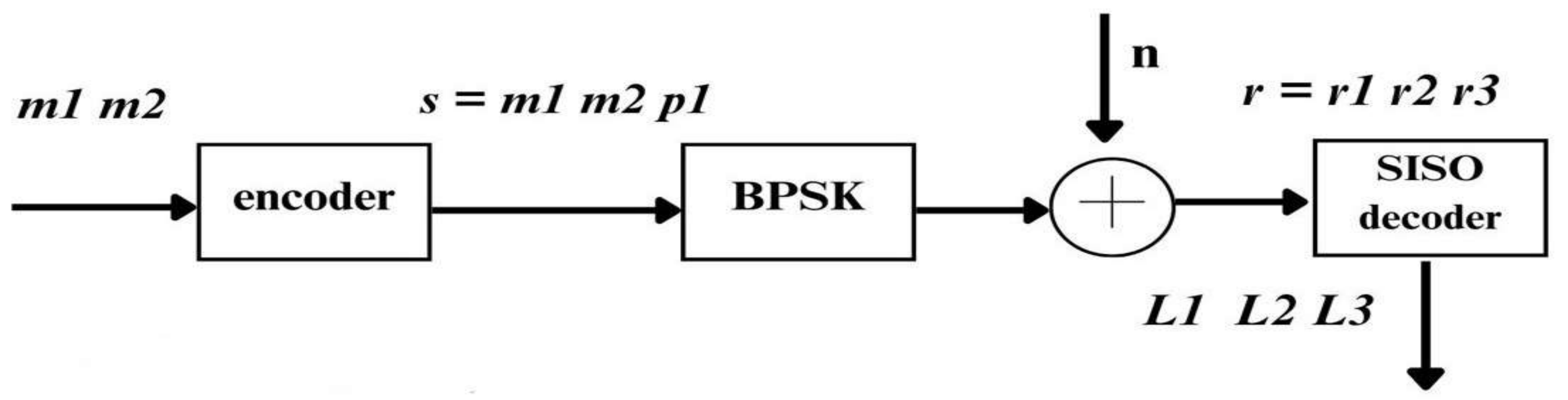


CN To VN

- Since each CN is a SPC code, the algorithm to send the message from CN to VN is simple performing an XOR operation .
- We take all the values received from the connected VNs except that sent by VN j. Then we take the xor of these values and send the message to VN j.
- To make the code efficient, we initially took XOR of all the messages received from each VN. The, to send the message to VN j, we again took the xor of the total with the message sent by VN j, and sent back that value.



SISO Decoder For Repetition Code



- Li = belief that bit si is zero
- L1 = computed using r1,r2,r3
- L1 = r1+r2+r3
- for L1, r1 = intrinsic, r2+r3 = extrinsic

*This example is taken from, Andrew Thangaraj. Ldpc and polar codes in the 5g standard, 2019.

SISO Decoder For Repetition Code

Calculations for L1:

$$L_1 = l_1 + l_{ext,1}$$

$$l_i = \frac{2}{\sigma^2} r_i \text{ (derived in the class)}$$

$$l_{ext,1} = log\left(\frac{P(c_1 = 0 \,|r_2 r_3)}{P(c_1 = 1 \,|r_2 r_3)}\right) = log\left(\frac{p_{e1}}{1 - p_{e1}}\right)$$

$$p_{e1} = \beta p_2 p_3$$

$$(1 - p_{e1}) = \beta(1 - p_2)(1 - p_3)$$

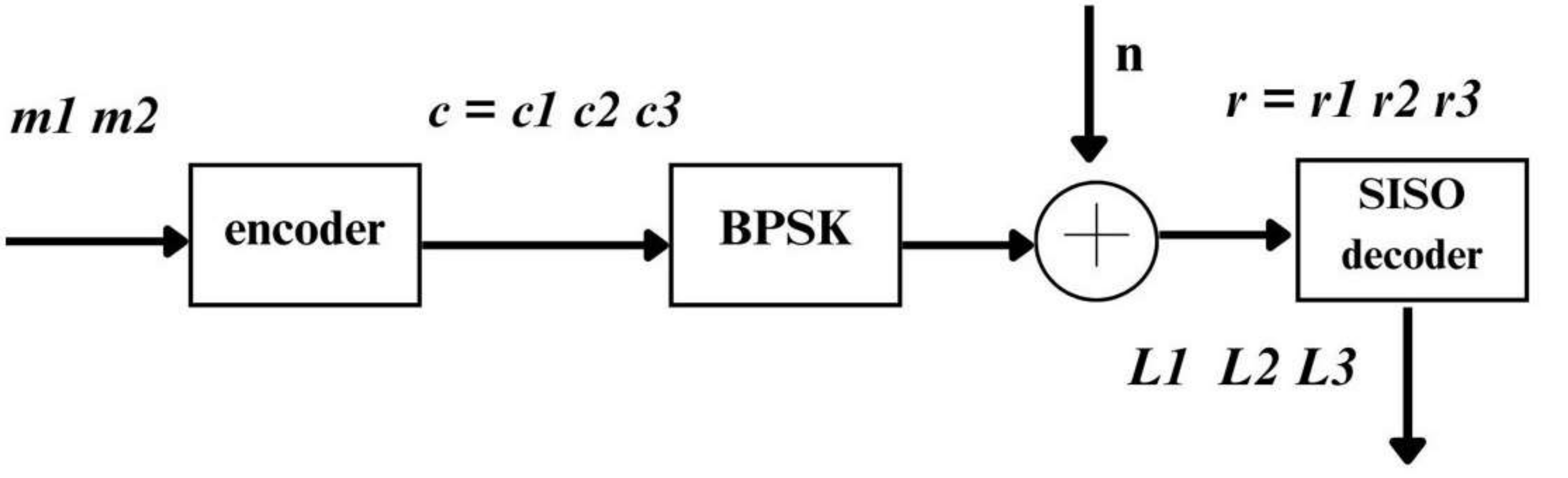
(derived in the class)

$$l_{ext,1} = log\left(\frac{p_2}{1 - p_2}\right) + log\left(\frac{p_3}{1 - p_3}\right)$$

$$l_{ext,1} = l_2 + l_3$$

*This example is taken from, Andrew Thangaraj. Ldpc and polar codes in the 5g standard, 2019.

SISO Decoder For SPC Code



- L1 =11+1ext, 1
- channel LLR 11= (2/(sigma)^2)*r1
- 1ext,1 = sign(r2)*sign(r3)* min(r2,r3)

*This example is taken from, Andrew Thangaraj. Ldpc and polar codes in the 5g standard, 2019.

SISO Decoder For SPC Code

Calculations for L1:

$$L_1 = l_1 + l_{ext,1}$$

$$l_i = \frac{2}{\sigma^2} r_i$$ (derived in the class)

$$l_{ext,1} = log\left(\frac{P\left(c_1 = 0\,|r_2 r_3\right)}{P\left(c_1 = 1\,|r_2 r_3\right)}\right) = log\left(\frac{p_{e1}}{1 - p_{e1}}\right)$$

As we know c1 = c2 $\oplus$ c3

$$p_{e1} = p_2 p_3 + (1 - p_2)(1 - p_3)$$

where
$$p_i = log\left(\frac{c_i = 0\,|r_i}{c_i = 1\,|r_i}\right)$$

from given equation we can derive

$$\frac{(p_{e1} - (1 - p_{e1}))}{(p_{e1} + (1 - p_{e1}))} = \frac{(p_2 - (1 - p_2))}{(p_2 + (1 - p_2))} \frac{(p_3 - (1 - p_3))}{(p_3 + (1 - p_3))}$$

$$\frac{\left(1 - \frac{(1-p_{e1})}{p_{e1}}\right)}{\left(1 + \frac{(1-p_{e1})}{p_{e1}}\right)} = \frac{\left(1 - \frac{(1-p_2)}{p_2}\right)}{\left(1 + \frac{(1-p_2)}{p_2}\right)} \frac{\left(1 - \frac{(1-p_3)}{p_3}\right)}{\left(1 + \frac{(1-p_3)}{p_3}\right)}$$

$$\frac{(1 - e^{-l_{ext,1}})}{(1 + e^{-l_{ext,1}})} = \frac{(1 - e^{-l_2})}{(1 + e^{-l_2})} \frac{(1 - e^{-l_3})}{(1 + e^{-l_3})}$$

$$tanh\left(\frac{l_{ext,1}}{2}\right) = tanh\left(\frac{l_2}{2}\right) tanh\left(\frac{l_3}{2}\right)$$

$$log\left(tanh\left(\frac{l_{ext,1}}{2}\right)\right) = log\left(tanh\left(\frac{l_2}{2}\right)\right) + log\left(tanh\left(\frac{l_3}{2}\right)\right)$$

*This example is taken from, Andrew Thangaraj. Ldpc and polar codes in the 5g standard, 2019.

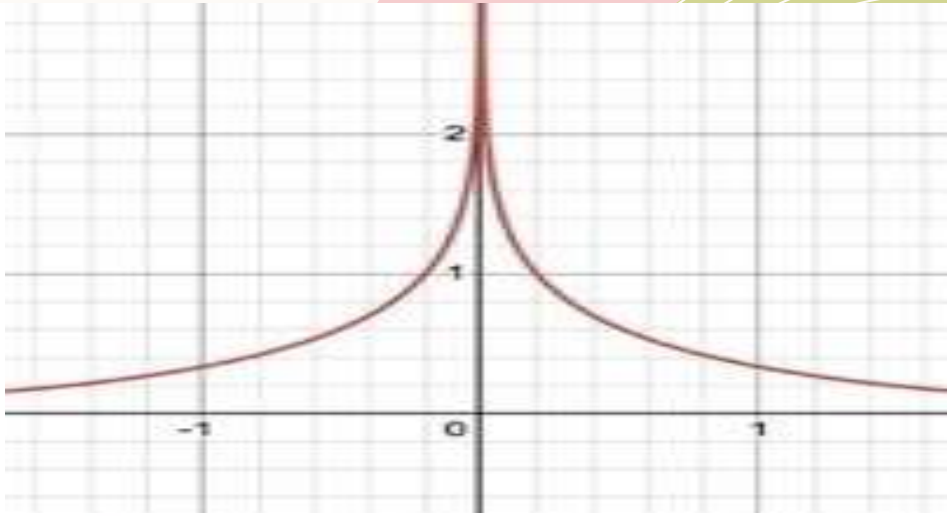
SISO Decoder For SPC Code

Calculations for L1:

we can separate the magnitude and sign of lext,1

$$log\left(tanh\left(\frac{|l_{ext,1}|}{2}\right)\right) = log\left(tanh\left(\frac{|l_2|}{2}\right)\right) + log\left(tanh\left(\frac{|l_3|}{2}\right)\right)$$

$$sgn\left(l_{ext,1}\right) = sgn\left(l_2\right)sgn\left(l_3\right)$$ because tanh is odd function



let $f = \left|log(tanh\left(\frac{|x|}{2}\right))\right|$ we can prove that f is self inverse function.

$$\left|l_{ext,1}\right| = f\left(f\left(l_2\right) + f\left(l_3\right)\right)$$

Minsum Approximation

$$f\left(l_2\right) + f\left(l_3\right) \approx f\left(min\left(|l_2|, |l_3|\right)\right)$$

$$\left|l_{ext,1}\right| \approx f\left(f\left(min\left(|l_2|, |l_3|\right)\right)\right) \approx min\left(|l_2|, |l_3|\right)$$

Soft Decision Decoding

MinSum decoder :

- Storage matrix L : sparse matrix of same dimensions as parity check matrix.
- L(i,j) = 0 if H(i,j) = 0
- L(i,j) can be non-zero only if H(i,j) = 1

$$\boldsymbol{r} = [\, r_1 \quad r_2 \quad r_3 \quad r_4 \quad r_5 \quad r_6 \quad r_7 \,]$$

$$L = \begin{bmatrix} r_1 & r_2 & r_3 & 0 & r_5 & 0 & 0 \\ 0 & r_2 & r_3 & r_4 & 0 & r_6 & 0 \\ r_1 & r_2 & 0 & r_4 & 0 & 0 & r_7 \\ r_1 & 0 & r_3 & 0 & r_5 & r_6 & r_7 \end{bmatrix}$$

*This example is taken from, Andrew Thangaraj. Ldpc and polar codes in the 5g standard, 2019.

For each Row

- **Magnitude**
  - min1 = minimum absolute value of all non-zero entries in row
  - min2 = next higher absolute value
  - set magnitude of all values

Set magnitude of all values except minimum = min1

Set magnitude of minimum value = min2

- **Sign**
  - Parity = Product of signs of entries in row
  - New sign of entry = (old sign) (parity)

Example:

Initialization

$$r = [\,0.2 \quad -0.3 \quad 1.2 \quad -0.5 \quad 0.8 \quad 0.6 \quad -1.1\,]$$

$$L = \begin{bmatrix} 0.2 & -0.3 & 1.2 & 0 & 0.8 & 0 & 0 \\ 0 & -0.3 & 1.2 & -0.5 & 0 & 0.6 & 0 \\ 0.2 & -0.3 & 0 & -0.5 & 0 & 0 & -1.1 \\ 0.2 & 0 & 1.2 & 0 & 0.8 & 0.6 & -1.1 \end{bmatrix}$$

Row Operation on Row 1

$$L = \begin{bmatrix} -0.3 & 0.2 & -0.2 & 0 & -0.2 & 0 & 0 \\ 0 & -0.3 & 1.2 & -0.5 & 0 & 0.6 & 0 \\ 0.2 & -0.3 & 0 & -0.5 & 0 & 0 & -1.1 \\ 0.2 & 0 & 1.2 & 0 & 0.8 & 0.6 & -1.1 \end{bmatrix}$$

After Row Operation

$$L = \begin{bmatrix} -0.3 & 0.2 & -0.2 & 0 & -0.2 & 0 & 0 \\ 0 & -0.5 & 0.3 & -0.3 & 0 & 0.3 & 0 \\ -0.3 & 0.2 & 0 & 0.2 & 0 & 0 & 0.2 \\ -0.6 & 0 & -0.2 & 0 & -0.2 & -0.2 & 0.2 \end{bmatrix}$$

*This example is taken from, Andrew Thangaraj. Ldpc and polar codes in the 5g standard, 2019.

For Each Column

- **New Values**
  - Sum_j = r_j + sum of all entries in column j.
  - New Entry = Sum - (Old Entry)
  - Sum is the new belief generated about values after the ith iteration
- **Decoding Values**
  - If sum_j > 0, decision on bit j = 0
  - If sum_j <0, decision on bit j = 1

Example:

$$r = [\;0.2 \quad -0.3 \quad 1.2 \quad -0.5 \quad 0.8 \quad 0.6 \quad -1.1\;]$$

$$L = \begin{bmatrix} -0.3 & 0.2 & -0.2 & 0 & -0.2 & 0 & 0 \\ 0 & -0.5 & 0.3 & -0.3 & 0 & 0.3 & 0 \\ -0.3 & 0.2 & 0 & 0.2 & 0 & 0 & 0.2 \\ -0.6 & 0 & -0.2 & 0 & -0.2 & -0.2 & 0.2 \end{bmatrix}$$

$$\text{Sum} = [\;-1 \quad -0.4 \quad 1.1 \quad -0.6 \quad 0.4 \quad 0.7 \quad -0.7\;]$$

Column Operation

$$\text{Sum} = [\;-1 \quad -0.4 \quad 1.1 \quad -0.6 \quad 0.4 \quad 0.7 \quad -0.7\;]$$

$$L = \begin{bmatrix} -0.7 & -0.6 & 1.3 & 0 & 0.6 & 0 & 0 \\ 0 & 0.1 & 0.8 & -0.3 & 0 & 0.4 & 0 \\ -0.7 & -0.6 & 0 & -0.8 & 0 & 0 & -0.9 \\ -0.4 & 0 & 1.3 & 0 & 0.6 & 0.9 & -0.9 \end{bmatrix}$$

After Column Operation

*This example is taken from, Andrew Thangaraj. Ldpc and polar codes in the 5g standard, 2019.

RESULTS

RESULTS

RESULTS

THANK YOU